PhD Thesis: Using spatio-temporal feature type structures for coupling environmental numerical models to each other and to data sources

Redacted Version: Some content has been removed and replaced by references where online publication permission was not granted.

ARC PhD by Publication Covering Paper

Candidate: Quillon Harpham, MSc Numerical Analysis

Personal Identifier: F7616989

Discipline: Hydroinformatics

Date of Submission: 20th March 2019

The Open University

Affiliated Research Centre: HR Wallingford Ltd

Quillon Harpham (Corresponding Author)(a)

a. HR Wallingford, Howbery Park, Wallingford, Oxfordshire, OX10 8BA. United Kingdom. Tel : +44(0)1491 822380 Email : q.harpham@hrwallingford.co.uk. Fax : +44(0)1491 835381

Contents

Glossary and Abbreviations

Terms used in this paper are defined in the context of this discussion as indicated in the glossary table below. This is followed by a second table listing the abbreviations.

GLOSSARY

Adaptors	Computer applications whose function it is to translate between datasets where there are differences between then such as units, spatial distances or temporal scales.
e-Infrastructure	A computing configuration tailored to specific applications and purposes.
e-Science	Science which is enabled by information and communications technology.
Feature types	Data structures based around identified and commonly repeating patterns, which are usually representative of physical objects described by the data (e.g. naturally occurring features such as rivers or constructed features such as bridges).
FerryBox	A through-flow measurement system installed on a ship.
File-based	Stored in a common file format.
GeoServer	An open source software server for enabling the sharing of spatial data.

GRID computing	A shared network of computing resources.
Ingesting data	Incorporating data taken from external sources.
Nowcast	Prediction of the very near future and calculation of the present and very near past.
Metadata	Additional contextual information about a dataset
Spatio-temporal structures	Defined arrangements of data whose values vary with both space and time.
Strongly typed	Highly prescriptive in structure and detail with little flexibility.
Structured (modelling) environments	Technical platforms provided to allow modellers to run and integrate models and supporting data.
Wave Sentry	A software product for enhancing ensemble model output using measured data sources.
Weakly typed	To a low degree of prescription in structure and detail yielding high flexibility.

ABBREVIATIONS

BMI	Basic Model Interface
Cb-TRAM	Cumulus Tracking and Monitoring
CF	Climate and Forecasting
CSML	Climate Science Modelling Language
DCI	Distributed Computing Infrastructure
DRIHM	Distributed Research Infrastructure for

	Hydro-Meteorology
ECOOP	European Coastal Ocean Observing Platform
GIS	Geographic Information System
gml	Geography Markup Language
GNSS	Global Navigation Satellite System
GUI	Graphical User Interface
gUSE	Grid and Cloud User Support Environment
HPC	High Performance Computing
HM	Hydro-Meteorology
ICT	Information and Communications Technology
INSPIRE	Infrastructure for Spatial Information in Europe
Model MAP	Model Metadata, Adaptors and Portability
NetCDF	Network Common Data Form
OGC	Open Geospatial Consortium
OpenMI	Open Modelling Interface
SDK	Software Development Kit
TimeSeriesML	Time Series Markup Language
UML	Unified Modelling Language
WFS	Web Feature Server
WMS	Web Map Server
XSLT	eXtensible Stylesheet Language

	Transformations

1. *Introduction, Aims and Objectives*

Our environment is increasingly being understood as a complex system of interacting processes; an interconnected system where changes to one phenomena at one location can have an impact on different phenomena at another location. Simulating such a system accurately therefore depends on at least some understanding of these interactions. Moore (2010) observes that an online mapping system such as Google Maps is the product of many years of standardising and processing the underlying geospatial information such that it can be seamlessly displayed to the user. If a similar system for observing environmental phenomena is to occur, then it will, in turn, depend on considerable effort in standardising and processing environmental data and numerical models to develop them into a platform capable of providing a similar user experience. Unfortunately, environmental numerical models have tended to be written to address highly specific questions and yet the need for an holistic approach, at least within topic domains, has been sought for many years.

Recent decades have seen a number of solutions being developed which tended to adopt the assumptions and characteristics of their associated topic domains. For example, the OpenMI standard (OGC OpenMI 2.0, 2014) was the result of over a decade of research with the intention of enabling interfacing between numerical models which may have initially been produced as such siloed solutions. Although more widely applicable, it arose from the hydraulic and hydrologic modelling community with a strong dependence on deterministic time-stepping models of physical phenomena based around varied spatial structures. The idea was to be able to re-use the established and calibrated solutions from the associated modelling domains without the investment necessary to re-write them from scratch. At the same time, technologies such as web services were being combined with standards such as GML (OGC GML, 2012) to offer methods of seamlessly displaying data from independent sources to user communities.

Accordingly, the overall aim of the present research at the outset was to facilitate the interaction of data between different numerical models and between numerical models and observed data sources. This was intended to give rise to interoperable and extensible simulations involving a variety of environmental phenomena from a variety of modelling domains. The specific emerging objectives were:

- To create a new abstract representation of environmental modelled data, typed to a level where it could be applied universally to numerical modelling. If such a representation were too strongly typed, then its applicability would be too narrow; too weakly typed and it would offer insufficient direction.
- To provide or draw out the accompanying elements necessary for such a representation to have practical use and to understand any limitations and boundaries.
- To demonstrate the use of such a representation in a variety of cases.

These objectives developed throughout the process of the present research and were shaped by the opportunities provided by related projects and initiatives which arose. These allowed the ideas to be explored and distilled as they were applied to the situations offered.

2. *Overview of Included Publications*

The publications included in this thesis document the developments which took place to achieve the above objectives.

Overall Context:

A paper giving an overall context for integrated environmental modelling:

- "*From integration to fusion: the challenges ahead*" observes the new field of integrated environmental modelling including consolidation given in a set of topics including metadata for data and models; supporting information; and linking (or interface) technologies.

Implementing Technologies:

A set of papers describing technologies developed to integrate data and numerical models.

- "*An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data*" is an early example of the use of spatio-temporal structures and web services to display independent datasets together. Identifying the underlying spatio-temporal structures for the datasets which function as the input and output datasets for environmental numerical models may be useful when considering the wider interoperation of these models.
- The OpenMI standard achieves coupling between numerical models by breaking down the spatial structures into their basic elements and considering the time dimension separately. *"The FluidEarth 2 implementation of OpenMI 2.0"* describes an implementation of this standard passing data between models in spatio-temporal structures aggregated from lower level component parts. This leads to

consideration of whether there is any value in spatio-temporal structures at a more aggregated level than basic components.

- Picking up the earlier theme of metadata for numerical models, *“Towards standard metadata to support models and interfaces in a hydro-meteorological model chain”* extends the ISO19115 standard to provide aspects related to model coupling including a vocabulary of spatio-temporal feature types – at this more aggregated level – derived from experiences using OpenMI and CSML.
- *“Using a Model MAP to prepare hydro-meteorological models for generic use”* formulates a set of necessary concepts for models to be coupled: **M**etadata (including that describing spatio-temporal structures at interfaces), **A**daptors to bridge between these structures and a measure of technical **P**ortability: together a Model MAP. This concept is tested through an hydro-meteorological model chain.

An e-Infrastructure for integrating data and models:

A series of papers describing the DRIHM e-Infrastructure for research into hydro-meteorology. They focus on the meteorological aspects; the user interface; the back-end computing infrastructures; and two key integrating technologies with an applied example.

- One journal paper, *“DRIHM(2US): an e-Science Environment for Hydro-meteorological research on high impact weather events”* and two conference papers, *“Setup an hydro-meteo experiment in minutes: the DRIHM e-Infrastructure for HM research”* and *“The DRIHM project: A flexible approach to integrate HPC, GRID and Cloud resources for hydro-meteorological research”* describe the motivations and high level outputs from two projects which sought to build an open and flexible architecture for model coupling, based on spatio-temporal structures at interfaces.

- *"Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy"* applies the concept of a Model MAP to a specific modelling chain where standardised data sources are also incorporated into the modelling architecture using the same spatio-temporal feature types to incorporate the driving data.

A further application of spatio-temporal feature types:

A paper giving another example of the use of spatio-temporal feature type structures as part of a new numerical modelling technology.

- *"A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members"* is another example of the use of spatio-temporal feature types to drive a modelling architecture which integrates data sources with a numerical model.

3. Summary of Each Publication Submitted in this Thesis including Journal Standing, Citations and Reviews

3.1 OVERALL CONTEXT

The first included paper gives an overall context to the field of integrated environmental modelling, motivating much of the subsequent work.

3.1.1 From integration to fusion: the challenges ahead

Sutherland, J., Townend, I.H., Harpham, Q.K. and Pearce, G.R., 2014. From integration to fusion: the challenges ahead. Geological Society, London, Special Publications, 408, pp.SP408-6.

The increasing complexity of numerical modelling systems in environmental sciences has led to the development of different supporting architectures. Models have become more and more detailed, representing more and more processes and, with increasing computer power, being solved using larger and larger geo-spatial structures. The past decade has seen the development of the new field of integrated environmental modelling where compositions of linked models exchange data at run-time. The application of systemic knowledge management to integrated environmental modelling indicates that we are at the onset of the norming stage, where gains will be made from the hierarchical organization of the competing options. This implies that there will be consolidation in the range of approaches that have proliferated in recent years, which is likely to become manifest in the predominance of a limited number of standards (covering ontologies, metadata, model interfaces, data formats and so on). An open software architecture (consisting of a user interface with published interfaces to a range of models, data and data processing routines) will be a key enabler to this, the use of open source software is likely to increase and a community must develop that values openness and the sharing of models and data as much as its publications and citation records.

Consolidation is proposed in six topics: metadata for data and models; supporting information; Software-as-a-service; linking (or interface) technologies; diagnostic or reasoning tools; and the portrayal and understanding of integrated modelling. Consolidation in these topics will develop model fusion: the ability to link models, with easy access to information about the models, interface standards (such as OpenMI) and software tools to make integration easier. This paper explores many of the current issues that need to be overcome to promote the concept of model integration by the dynamic linking of models. It starts by introducing two frameworks that have been used to assess the progression in numerical modelling (largely in hydraulics) and considers their application to the modelling of integrated environmental systems. It then discusses the increasingly blurred line between observations(data) and models, before describing the evolution of the OpenMI standard.

Google Scholar: cited by 10 other papers. Available in 4 versions.

3.2 IMPLEMENTING TECHNOLOGIES

The next papers discuss technologies which have been applied to integrated environmental modelling, integrating both measured data and numerical models.

3.2.1 An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data

Gemmell, A.L., Barciela, R.M., Blower, J.D., Haines, K., Harpham, Q., Millard, K., Price, M.R. and Saulter, A., 2011. An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data. Ocean Science, 7, pp.445-454. ISSN 1812-0792. DOI: 10.5194/os-7-445-2011

This paper describes the development of a web portal for the display and comparison of model and in-situ marine data under the European coastal operational oceanography project (ECOOP).

Marine scientists use highly diverse sources of data, including in situ measurements, remotely-sensed information and the results of numerical simulations. The ability to access, visualize, combine and compare these datasets is at the core of scientific investigation. The distributed model and in situ datasets are accessed via a Web Map Service (WMS) and Web Feature Service (WFS) respectively from the Open Geospatial Consortium (OGC) which has been instrumental in developing and promoting standards for representing and exchanging geospatial data. Many of its standards are mandated by INSPIRE[1], notably the aforementioned Web Map Service[2] for map imagery and the Web Feature Service[3] for geospatial data. These services were developed independently and readily integrated, illustrating the ease of interoperability resulting from adherence to international standards. These standards

[1] Infrastructure for spatial information in Europe, https://inspire.ec.europa.eu/
[2] WMS, http://www.opengeospatial.org/standards/wms
[3] WFS, http://www.opengeospatial.org/standards/wfs

have evolved from the domain of Geographic Information Systems (GIS), which have historically been concerned mainly with two-dimensional land-based data (Rahim et al., 1999; Guney et al., 2003). However, scientific description or modelling of the environment usually involves 4-D data (3D data evolving in time) – or even 5D including ensembles – as needed to describe the atmosphere or ocean properties. The key feature of the portal is the ability to display co-plotted time series of the *in situ* and model data and the quantification of misfits between the two. By using standards-based web technology we allow the user to quickly and easily explore over twenty model data feeds and compare these with dozens of *in situ* data feeds without being concerned with the low level details of differing file formats or the physical location of the data. Scientific and operational benefits to this work include model validation, quality control of observations, data assimilation and decision support in near real time. In these areas it is essential to be able to bring different data streams together from often disparate locations. A working multiple data provider system is demonstrated, delivered through a single web portal displaying real time model and in situ marine data from 20 modelling groups across Europe and from 45 different in situ observation monitoring stations in 24 different countries. The system has used OpenSource software and standards compliant methods wherever possible. Several applications requiring multi-data input have been given as examples and the authors believe this kind of service, built on the back of standards based data serving, will become critical for monitoring the marine and wider environment and environmental change on a national and international basis into the future.

Google Scholar: cited by 4 other papers. Available in 19 versions. Ocean Science journal impact factor for 2011 is 2.73.

3.2.2 The FluidEarth 2 implementation of OpenMI 2.0

Harpham, Q., Cleverley, P. and Kelly, D., 2014. The FluidEarth 2 implementation of OpenMI 2.0. Journal of Hydroinformatics, 16(4), pp.890-906.

The Open Modelling Interface (OpenMI) is a standard for coupling numerical models with data exchanged between modelling components at run time. Following the successful version 1.4, version 2.0 of OpenMI was released in December 2010. The standard consists of a set of object interfaces, which can be represented by a set of Unified Modelling Language (UML) diagrams. These interfaces are necessary to enable numerical model developers to easily adapt their models to become OpenMI compliant and to allow modellers to easily assemble and run compositions of them. Following the release of the OpenMI 2.0 standard, a set of tools – including a Software Development Kit (SDK) and Graphical User Interface (GUI) – is expected to accompany it. FluidEarth 2 is an HR Wallingford initiative providing these open source tools for the .net 4.0 Framework together with training, community support and sample models, focusing on openness, flexibility and usability. They are the only such open source tools available so in this sense they act as a reference SDK and GUI for OpenMI 2.0 with .net. To this end, a series of components were successfully constructed and compositions built. These included training models designed to demonstrate different aspects of model coupling, moving to industry strength model codes simulating dam-break bathymetry updates. The FluidEarth 2 tools have been designed to be cross-platform and have been tested under Windows and Linux (using

Mono). Usage is successfully demonstrated, providing an environment for integrated modelling with OpenMI 2.0. The purpose of this paper is to outline the FluidEarth 2 SDK and GUI and, with reference to the training material, document a set of examples introducing the reader to using OpenMI 2.0 with FluidEarth 2. Although not restricted to hydrology and environmental modelling, FluidEarth 2 is driven from these disciplines and the examples listed all derive from this subject area.

Google Scholar: cited by 19 other papers. Available in 3 versions. According to "omicsonline" the impact factor for this journal in 2014 was 1.388. Since its release as an open source product on SourceForge, FluidEarth has been downloaded over 1000 times from 55 countries, with ten or more downloads from 19 countries. Most active interest has been from the UK (242), China (150), USA (116), Germany (79) and India (44). Downloads of OpenMI 2.0 itself now exceed 5,500.

Notwithstanding requests to restructure early drafts of the paper and amendments to a number of details, peer reviews of the paper were positive:

- *"In general, the contribution can be considered as timely, especially in light of the growing need for an integrated approach to holistic water resources management, and the need of promoting, practicing and sharing 'open source' developments. Besides, I am fan of such 'open source' developments. I believe that the FluidEarth 2 implementation of the OpenMI standard v2.0 would contribute to the easy and feasible integration of environmental models."*
- *"This is an important software contribution to the water resources community given that it is one of the (if not the) first systems to implement the OpenMI 2.0*

standard. It should be of wide interest to readers of this journal given the popularity of the OpenMI 1.4 paper published in this journal."

3.2.3 Towards standard metadata to support models and interfaces in a hydro-meteorological model chain

Harpham, Q. and Danovaro, E., 2015. Towards standard metadata to support models and interfaces in a hydro-meteorological model chain. Journal of Hydroinformatics 17.2 260-274, IWA Publishing. doi: 10.2166/hydro.2014.061.

This paper seeks to move towards an un-encoded metadata standard supporting the description of environmental numerical models and their interfaces with other such models. Building on formal metadata standards and supported by the local standards applied by modelling frameworks, the desire is to produce a solution that is as simple as possible yet supports validation of model interfaces together with basic discovery and use requirements that support model coupling processes. The purpose of metadata is to provide supporting information to allow what it is describing to be found, correctly interpreted and utilised. In environmental modelling use cases such as the hydro-meteorological model chain discussed in this paper, the utilisation aspects increasingly depend on the ability to interface models with each other (and, indeed, other supporting datasets). Formal standards for model coupling are now also coming to the fore building on formal metadata standards and supported by the local standards applied by modelling frameworks.

The purpose of this metadata is to allow environmental numerical models, with a first application for a hydro-meteorological model chain, to be discovered and then an initial evaluation made of their suitability for use, in particular for integrated model compositions. Indeed, across all appropriate disciplines, metadata describing numerical models is clearly required to support any kind of automation or semi-automation of the model coupling process. The method applied is to begin with the ISO19115 standard and add extensions suitable for environmental numerical models in general. Further extensions are considered pertaining to model interface parameters (or phenomena) together with spatial and temporal characteristics supported by feature types from the Climate Science Modelling Language (CSML). Successful validation of parameters depends heavily on the existence of controlled vocabularies. The metadata structure formulated has been designed to strike the right balance between simplicity and supporting the purposes drawn out by interfacing the Real-time Interactive Basin Simulator hydrological model to meteorological and hydraulic models and, as such, successfully provides an initial level of information to the user.

Google Scholar: cited by 13 other papers. Available in 4 versions. At the time of writing, according to the IWA publishing website, impact factor for this journal is 1.180.

Peer review comments were in general positive, making some constructive challenges:

- *"In general this new set of keywords sounds reasonable and should help discovering model components faster. However, the question is how to make sure that these keywords will be standardized and supported by the different environmental modeling communities?"*

- *“I agree strongly that there is need for standard model metadata in environmental modeling. The author presents some interesting ideas and the work would be of interest to the community.”*
- *“In the conclusions, Page 34, line 38 "If the metadata is too comprehensive then there is a risk that suppliers will not provide it." I disagree and believe instead there needs to be levels of metadata with "core" metadata clearly distinguished from optional metadata that is clearly defined and consistent and available for use.”*

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

Harpham, Q., Cleverley, P., Danovaro, E., D’Agostino, D., Galizia, A., Delogu, F. and Fiori, E. 2015. Using a Model MAP to prepare hydro-meteorological models for generic use. Environmental Modelling & Software, 73, pp.260-271.

Structured environments for executing environmental numerical models are becoming increasingly common, typically including functions for discovering models, running and integrating them. 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. 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. Interoperability issues can play a major role in model integration when the models are developed in different programming languages, platforms and operating systems. The model MAP can be considered as a checklist of requirements designed at a level such that it spans the functional and technical diversity of environmental numerical models. 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. 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.

Google Scholar: cited by 5 other papers. Available in 3 versions. According to the Elsevier website the 'Environmental Modelling and Software Journal' has a 5-year impact factor of 4.528.

Reviews of earlier drafts were positive and suggested a more comprehensive referencing of current material, as a result the comparison with the Elag & Goodall ontology was included:

- *"The paper addresses a relevant topic within the context of integrated environmental modeling, that is, the concept of interoperability and how models from various science domains can be prepared to more easily share data and information at runtime. The paper is well organized and the logic flows fairly well."*
- *"I would recommend that the author review Elag et al, for example, who provide a detailed organization of metadata for numerical models."*
- *"I am in full agreement with the paper's overall message and recommend publication with minor to medium revisions."*

3.3 AN E-INFRASTRUCTURE FOR INTEGRATING DATA AND MODELS

The next series of papers builds on the implementing technologies and combines these with other innovations to construct an e-Infrastructure to facilitate research into hydro-meteorological model chains. This e-Infrastructure is entitled DRIHM – the Distributed Research Infrastructure for Hydro-Meteorology. The first paper focuses on the meteorological aspects, the second focuses on the user interface and its underlying technologies, the third focuses on the three different back-end computing infrastructures used and the fourth picks out two key integrating technologies in an applied example.

3.3.1 DRIHM(2US): an e-Science environment for hydro-meteorological research on high impact weather events

This and the following two conference papers are from the Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) and Distributed Research Infrastructure for Hydro-Meteorology to the United States (DRIHM2US) projects. They have wide authorship from across the project teams and have broad scope, drawing out major functional and technical aspects.

Parodi, A., Kranzlmueller, D., Clematis, A., Danovaro, E., Galizia, A., Garrote, L., Llasat, M., Caumont, O., Richard, E., Harpham, Q., Siccardi, F., Ferraris, L., Rebora, N., Delogu, F., Fiori, E., Molini, L., Georgiou, E. and D'Agostino, D. 2017. DRIHM(2US): an e-Science environment for hydro-meteorological research on high impact weather events. Bull. Amer. Meteor. Soc. Doi:10.1175/BAMS-D-16-0279.1.

From 1970 to 2012, about 9000 high impact weather events were reported globally causing the loss of 1.94 million lives and damage of US$ 2.4 trillion. The scientific community often struggles to help with improving resilience to such events or handling their impact. At the heart of these research challenges lies the ability to have easy access to hydrometeorological data and models, and to facilitate the necessary collaboration between meteorologists, hydrologists, and Earth science experts to achieve accelerated scientific advances. Two EU funded projects, DRIHM and

DRIHM2US, sought to help address this by developing a prototype e-Science environment providing advanced end-to-end services (models, datasets and post-processing tools), with the aim of paving the way to a step change in how scientists can approach studying these events, with a special focus on flood events in complex topography areas. This paper describes the motivation and philosophy behind this environment together with certain key components, focusing on meteorological aspects which are then illustrated through enabled research into flash flood events in Liguria, Italy.

Google Scholar: Cited by 4 other papers. According to Journals Impact Factor Lists – 2016 Citation Reports Ranking (www.omicsonline.org/Impact/Factors), the impact factor for the Bulletin of the American Meteorological Society for 2015 is 7.929.

3.3.2 SetUp an Hydro-Meteo Experiment in Minutes: The DRIHM e-Infrastructure for HM Research

Due to the fast moving nature of computational science, the Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) project was asked specifically to submit this material to high profile ICT conferences rather than to journals. This is the first of two such papers giving an overview of the infrastructure created by the project and it is featured in the conference proceedings.

Danovaro, E., Roverelli, L., Zereik, G., Galizia, A., D'Agostino, D., Quarati, A., Clematis, A., Delogu, F., Fiori, E., Parodi, A., Straube, C., Felde, N., Harpham, Q., Jagers, B., Garrote, L., Dekic, L., Ivkovic, M., Richard, E. and Caumont, O. Setup

an hydro-meteo experiment in minutes: the DRIHM e-infrastructure for hydro-meteorology research, proceedings of e-Science 2014: 10th IEEE International Conference on e-Science, Guarujá, SP, Brazil, October 20-24, 2014.

Predicting weather and climate and its impacts on the environment, including hazards such as floods and landslides, is still one of the main challenges of the 21st century with significant societal and economic implications. Understanding and addressing this challenge can be supported by a distributed and heterogeneous infrastructure, exploiting several kinds of computational resources: HPC, Grids and Clouds. This can help researchers speed up experiments, improve resolution and accuracy, and simulate with different numerical models and model chains. Such numerical models are typically complex with heavy computational requirements, huge numbers of parameters to tune, and not fully standardized interfaces. Hence, each research entity is usually focusing on a limited set of tools and hard-wired solutions to enable their interaction. The Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) project aimed at setting the stage for a new way of doing hydro-meteorological research (HMR) combining scientific expertise in this field with recent achievements in Grids, Clouds and High Performance Computing (HPC). The DRIHM approach is based on strong standardization, well defined interfaces, and an easy to use web interface for model configuration and experiment definition. A researcher can easily compare outputs from different hydrological models forced by the same meteorological model, or compare different meteorological models to validate or improve their research.

This paper presents the benefit of a web-based interface for HMR through a detailed analysis of a portal as developed by the DRIHM project, relying on the gUSE technology for submitting to a distributed computing infrastructure (DCI). gUSE is extended by a set of custom portlets to support hydro-meteorological researchers in experiment definition and execution. Effectiveness of the DRIHM approach is based on three pillars: wide effort in interface standardization based on adoptions of the NetCDF-CF and WaterML 2 OGC standards, meta-tagging of each model instance to perform workflow compatibility check prior to the actual execution of the jobs, and the availability of an heterogeneous computing infrastructure. The advantages and benefits of integrating these models, tools and data are briefly discussed by way of showing their potential importance in helping researchers to design more efficient early-warning flash flood prediction systems.

Google Scholar: cited by 10 other papers. Available in 3 versions.

3.3.3 The DRIHM Project: A Flexible Approach to Integrate HPC, Grid and Cloud Resources for Hydro-Meteorological Research

Due to the fast moving nature of computational science, the Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) project was asked specifically to submit this material to high profile ICT conferences rather than to journals. This is the second of two such papers giving an overview of the infrastructure created by the project and it is featured in the conference proceedings.

D'Agostino, D., Clematis, A., Galizia, A., Quarati, A., Danovaro, E., Roverelli, L., Zereik, G., Kranzlmuller, D., Schiffers, M., gentschen Felde, N., Straube, C., Caumont, O., Richard, E., Garrote, L., Harpham, Q., Jagers, B., Dimitrijevic, V.,

Dekic, L., Parodi, A., Fiori, E. and Delogu, F., 2014 The DRIHM project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological research, SC ’14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. pp 536-546, IEEE Press Piscataway, NJ, USA © 2014. doi 10.1109/SC.2014.49.

The distributed research infrastructure for hydrometeorology (DRIHM) project focuses on the development of an e-Science infrastructure to provide end-to-end hydro-meteorological research (HMR) services (models, data, and post processing tools) by exploiting HPC, Grid and Cloud facilities. In particular, the DRIHM infrastructure supports the execution and analysis of high-resolution simulations through the definition of workflows composed by heterogeneous HMR models in a scalable and interoperable way, while hiding all the low level complexities. Computational earth and atmospheric sciences such as HMR, play a key role in guiding the design and implementation of prediction tools devoted to the safety and prosperity of humans and ecosystems from highly urbanized areas to coastal zones and agricultural landscapes. This contribution gives insights into best practices adopted to satisfy the requirements of an emerging multidisciplinary scientific community composed of earth and atmospheric scientists. Forecasting severe storms and floods could be considered as one of the main challenges of the 21st century. Meeting this challenge requires improvements in the way predictions are obtained. At the heart of this challenge lies easy access to hydro-meteorological data repositories, models and computing resources and facilitating collaboration between meteorologists, hydrologists, and earth scientists. To this end, DRIHM supplies innovative services leveraging high performance and distributed computing resources. Hydro meteorological requirements shape this IT infrastructure through an iterative "learning-by-doing" approach that permits tight interactions between the application community and computer scientists, leading to the development of a flexible, extensible, and interoperable framework.

This paper presents the e-Science infrastructure developed for HMR within the DRIHM European project, an initiative that tackles these issues enabling the proper management of different kinds of software and hardware resources, from models and data to newly deployed services and infrastructures.

Google Scholar: cited by 16 other papers. Available in 4 versions.

3.3.4 Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modelling of Genoa, Italy

Harpham, Q., Lhomme, J., Parodi, A., Fiori, E., Jagers, B. and Galizia, A., 2016. Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy. JAWRA Journal of the American Water Resources Association (2016).

Extreme hydro-meteorological events such as flash floods have caused considerable loss of life and damage to infrastructure over recent years. An analysis carried out by the FLASH project calculated that flood events in the Mediterranean region between 1990 and 2006 caused over 4,500 fatalities and cost over €29 billion in damage, with Italy one of the worst affected countries. The Distributed Computing Infrastructure for Hydro-Meteorology (DRIHM) project is a European initiative aiming at providing an open, fully integrated eScience environment for predicting, managing, and mitigating the risks related to such extreme weather phenomena. DRIHM (http://www.drihm.eu) is an example of a structured research environment offering users easy access to numerical models and supporting data sources, together with the computing resources required to run workflows incorporating them. Incorporating both modelled and observational data sources, it enables seamless access to a set of computing resources with the objective of providing a collection of services for performing experiments with numerical models in meteorology, hydrology, and hydraulics. The purpose of this article is to demonstrate how this flexible modelling architecture has been constructed using a set of standards including the NetCDF and WaterML2 file formats, in-memory coupling with OpenMI, controlled vocabularies such as CF Standard Names, ISO19139 metadata, and a Model MAP (Metadata, Adaptors, Portability) gateway concept for preparing numerical models for standardized use. Hydraulic results, including the impact to buildings and hazards to people, are given for the use cases of the severe and fatal flash floods, which occurred in Genoa, Italy in November 2011 and October 2014. The modeling architecture outlined in this article is designed to be interoperable and extensible within modeling domains (meteorology, hydrology, and hydraulics) as well as between modeling domains. This has been achieved, but is limited by the nature of the model output and input: as long as the input/output spatio-temporal feature types (e.g., grid-series, point-series) are the same, numerical models can be incorporated into this simple structure. It is then also possible to utilize an ensemble of equivalent models and observational data from each domain – not restricted to those given here – as well as incorporating new domains. The DRIHM portal allows these models to be executed against a variety of resources, enabled by the gUSE science gateway with the interfaces based around the different spatio-temporal feature types using different file standards.

Google Scholar: Cited by 4 other papers. According to the Wiley website, The impact factor for the Journal of the American Water Resources Association for 2015 is 1.659 with ISI Journal Citation Reports © Ranking: 2015: 28/50 (Engineering Environmental); 34/85 (Water Resources); 93/184 (Geosciences Multidisciplinary).

Initial reviews for the paper indicated support for the approach taken and changes were requested to improve the structure, context and flow:

- *"I think this is a promising paper on a subject of increasing interest to the water resources community. Integrating computational resources in a more robust way and developing interoperable standards and data formats for modeling across disciplines is essential to mitigating flood risks, and more work needs to be published on these themes in water resources journals."*
- *"Overall the paper could benefit from improved transitions between topics, and a greater discussion of alternative methods."*
- *"I enjoyed reading this paper which I found a useful contribution to the science and implementation of integrated modelling. I have suggestion some fairly minor changes to improve the paper, mainly consistency, clarification and improving a couple of the figures."*

3.4 A FURTHER APPLICATION OF SPATIO-TEMPORAL FEATURE TYPES

The final paper introduces a new numerical modelling technology and considers another example of the use of spatio-temporal feature type structures. This example is drawn from a different field of environmental modelling.

3.4.1 A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members

Harpham, Q., Tozer, N., Cleverley, P., Wyncoll, D. and Cresswell, D., 2016. A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members. Environmental Modelling & Software 84 (2016): 482-493.

New innovations are emerging which offer opportunities to improve forecasts of wave conditions. Such forecasts are required for planning of a wide range of weather sensitive maritime operations from construction and maintenance to decommissioning. Traditional wave forecasts provide a single estimate of conditions with a typical outlook of 5 to 7 days, giving parameters such as significant wave height, maximum wave height, wave period and direction. Such deterministic forecasts provide limited or no information on the potential uncertainty in a given forecast. Probabilistic forecasts, in contrast, such as those based on an ensemble of multiple predictions, not only extend the range of the forecasts often out to 14 days, but also provide a measure of the uncertainty at any given time-step. With increasing computing power, probabilistic forecasts are becoming increasingly common and will no doubt become the norm. These include probabilistic modelling results, such as those based on an ensemble of multiple predictions which can provide a measure of the uncertainty, and new sources of observational data such as GNSS reflectometry and FerryBoxes (allowing sensors to be mounted on moving vessels collecting ocean data parameters), which can be combined with an increased availability of more traditional static sensors.

This paper outlines an application of the Bayesian statistical methodology which combines these innovations. The WaveSentry system is a set of components for harvesting observed data sources with different identified characteristics and implementing an application of the Bayesian statistical methodology that modifies the probabilities of ensemble wave forecasts based on recent past performance of individual members against these observations. Each data source is harvested and mapped against a set of spatio-temporal feature types and then used to post-process ensemble model output. A prototype user interface is given with a set of experimental results testing the methodology for a use case covering the English Channel.

Google Scholar: Cited by 1 other paper. According to the Elsevier website the 'Environmental Modelling and Software Journal' has a 5-year impact factor of 4.528.

Comments from the reviewers were positive, requesting that certain aspects be clarified and brought-to-the-fore, whilst others were deprecated:

- *"This is an interesting and clearly-written paper that integrates a number of concepts together in an application to improve probabilistic wave forecasting."*
- *"The paper is overall well written and interesting, and the application of the technique seems to be appropriate and useful."*
- *"I am also not fully convinced that this paper is the right place to discuss the GUI development. I do however agree that it is crucial to explain how the*

ensemble forecasting would work and who would use it, and how it would fit into a decision support system.”

4. *Interrelationship between the Publications*

This series of papers makes a progression in the candidate's research over the period between 2011 and 2017 towards the use of a set of spatio-temporal feature type structures for coupling environmental numerical models with each other and with data sources.

The paper "*From integration to fusion: the challenges ahead*" provides a context to the field of integrated numerical models and data sources, observing the new field of integrated environmental modelling where compositions of linked models exchange data at run-time and suggests that gains will be made from the hierarchical organization of the emerging approaches. With open software architectures acting as a key enabler, consolidation is given in a set of topics including metadata for data and models; supporting information; Software-as-a-service; and linking (or interface) technologies.

The paper "*An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data*" uses web services to display and compare a set of model and in-situ marine data sources, based on a set of spatio-temporal feature types from CSML. The CSML feature type set was designed to cover all aspects of observed environmental data. This architecture shows that standardising in this way allows different data streams to be brought together from often disparate locations, without use of a central repository. It is highly extensible since new data streams merely have to follow the same feature-type implementation of these web services.

Alongside these developments, initiatives were being undertaken to enable standardised coupling of legacy model codes. One such initiative resulted in the development of the OpenMI standard, with its highly successful FluidEarth implementation as outlined in the paper "*The FluidEarth 2 implementation of OpenMI 2.0*". OpenMI 2.0 overcomes spatial differences between numerical model domains by breaking down the spatial structures into their basic elements and FluidEarth overcomes temporal differences by interpolating between differing timesteps. Together these approaches demonstrate that common Feature Types used in model coupling (such as Grid and Point Series) can be aggregated from such lower level components and supported by standards.

The idea of standard metadata for models and supporting information is explored by the paper "*Towards standard metadata to support models and interfaces in a hydro-meteorological model chain*". The ISO19115 standard is extended to provide aspects related to model coupling including spatio-temporal feature types as suggested by CSML and earlier demonstrated in "*An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data*". Experiences in implementing OpenMI version 2.0 with FluidEarth helped derive these extensions and the environment in which they operate through the idea of adapting inputs and outputs with separate components. The paper "*Using a Model MAP to prepare hydro-meteorological models for generic use*" collects these concepts together in the formulation of a Model MAP and tests this implementation through an hydro-meteorological model chain involving meteorological, hydrological and hydraulic models where interfaces between modelling domains are marked by different spatio-temporal structures. The Model MAP acts as a checklist of requirements for numerical models seeking to be implemented on coupling architectures.

The Model MAP concepts, together with two others from the paper "*From integration to fusion: the challenges ahead*", software-as-a-service and linking (or interface) technologies, are implemented by the DRIHM infrastructure described in the papers "*Setup an hydro-meteo experiment in minutes: the DRIHM e-infrastructure for hydro-meteorology research*" and "*The DRIHM project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological research*" and a later journal paper also including the accompanying DRIHM2US project, "*DRIHM(2US): an e-Science Environment for Hydro-meteorological research on high impact weather events*". The hydro-meteorological modelling chain was applied in practice as described by the paper "*Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy*" where standardised data sources are also incorporated into the modelling architecture using the same spatio-temporal feature types which drive the numerical modelling interfaces. These interfaces are supported by established standards for storing (environmental) data.

Spatio-temporal feature types also drive the architecture for the WaveSentry system as described in the paper "*A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members*". This time, a variety of measured data sources are used to post-process a set of ensemble members thereby changing their relative weighting which is an indication of the probability that they are correct.

5. *Further Impact*

Further evidence of the impact of this work is given as follows:

- A number of these papers acknowledge the DRIHM and DRIHM2US projects. DRIHM (EC 7th Framework Programme, Grant Number 283568) was evaluated as an EC 'success story' and was supported by a follow up article. The DRIHM2US project (EC 7th Framework Programme, Grant Number 313122) was rated 'Excellent' by the Review Panel.
- The WaveSentry project (part funded by the UK Technology Strategy Board, now InnovateUK, under the call: "Harnessing Large and Diverse Sources of Data", grant number 100940) was also deemed a 'success story' resulting in the production of an associated article. The candidate also presented this work at the iEMSs conference in Toulouse 2016 with a very positive reception from delegates from US and European private and public sectors.
- The paper "*The FluidEarth 2 implementation of OpenMI 2.0*" has become a standard reference for a course at the University of Virginia.
- In his capacity as Chairman of the OpenMI Association, the candidate gave a keynote talk at the ICHE 2014 conference in Hamburg entitled "Integrated Environmental Modelling: What is the Vision? Is it achievable?". This presentation articulated a vision for environmental modelling using a metaphor of a musical orchestra, collecting together concepts from these papers.

6. Critical Review and Candidate Contribution

The increased availability of environmental data from both the diverse range of monitoring devices and from numerical models is creating opportunities for it to be combined and integrated. Scientists are continually seeking the benefits of comparing and contrasting data sources including formulating combinations of numerical models and their supporting datasets. Initiatives such as the EU INSPIRE directive have embraced this and, in an attempt to provide a framework for mitigation, move to mandate the use of international standards for structuring and disseminating public data. The Climate Science Modelling Language (CSML) sought to describe the natural structure of measured environmental data from a variety of sources. CSML version 2.0 introduces a set of ten Feature Types, structuring them around a variety of natural and commonly occurring forms of environmental data (Woolf et al., 2006). With the exception of 'observation' they have been defined to be specialisations of the Observations and Measurements (O&M) model (ISO19156, 2011). It has been stable since the release of version 3 in 2011 (Lowe, D., 2011). The paper "*An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data*" demonstrates how formal feature types selected from a defined collection – in this case CSML – can be combined with web service standards to integrate these data sources with others of similar (and different) character in a single portal. **Building on prior work developing XSLT to encode CSML feature types, the candidate's contribution was to create a relational database optimised for point series data and create a WFS service encoding directly into the CSML Point Series feature type, thereby demonstrating the success of the approach in integrating with data from other sources. This work contributed to the development and proving of**

CSML as a feature type approach as well as early adoption of the Geoserver technology.

File-based standards for the most common feature types such as PointSeries and GridSeries are well established. The evolution has seen community standards develop into more formalised articulations with subsequent ratification by standards bodies. Originating from hydrology, the WaterML[4] standard was devised for use cases such as the collection of readings from static river flow meters. Readings vary in time, but are fixed at a point in space. Having gone through two major versions, WaterML2 (Taylor et al., 2014) was further developed into the more generic TimeSeriesML for storing point series data (Arctur et al., 2015) which is now also ratified as an OGC standard.

In parallel with these developments, OpenMI, an in-memory model coupling standard, was developed by the hydraulic modelling community (Gregersen et al., 2007). Data is passed between numerical models, in memory, at run time. Processes can be allowed to influence each other as they progress through their successive timesteps. Version 2.0 of OpenMI (Moore et al., 2010) was, itself, adopted as an OGC standard in 2013[5]. The FluidEarth 2 implementation is described in the paper "*The FluidEarth 2 implementation of OpenMI 2.0*". OpenMI is now one of the leading numerical model coupling standards. The spatio-temporal structures are broken down into components which are at a lower level than the Feature Types catalogued in CSML (and similar frameworks) and yet, they are assembled to support these higher level Feature Types easily. This paper demonstrates that common Feature Types used in model coupling

[4] http://www.opengeospatial.org/standards/waterml
[5] http://www.opengeospatial.org/standards/openmi

(such as Grid and Point Series) can be aggregated from such lower level components and supported by standards such as OpenMI with its FluidEarth implementation. **As project lead, the candidate's contribution to the work included producing compositions ingesting point series data during development, specifying and testing the FluidEarth components against user requirements, deriving test and demonstration examples using a variety of spatio-temporal structures including Grid and Line Series and assisting with the derivation of the accompanying training website. The candidate is the current chair of the OpenMI Association**.

Supported by the candidate's experience of OpenMI and model coupling, the paper "*From integration to fusion: the challenges ahead*" places the OpenMI and FluidEarth developments into a wider context meeting a more general requirement to support coupling of numerical models. At minimum this involves passing the results of one numerical model to another, to allow it to influence those 'downstream'. This process can be as simple as taking the output file from one model and making it the input file to another – usually with some re-formatting, interpolation and interpretation in between. For example, a meteorological model predicting rainfall will typically drive a downstream model calculating the drainage of a river catchment. The output file of the meteorological model will include a measure of rainfall which is extracted and used as the input to the catchment drainage model. Most numerical models are also coupled to their supporting data sources. For example, the same catchment drainage model can be driven from measured, raingauge data taken from devices rather than the meteorological model. As such, standards for structuring observed data become relevant to the model coupling discussion.

The paper "*Towards standard metadata to support models and interfaces in a hydro-meteorological model chain*" uses this example to explore an accompanying aspect – the metadata necessary to support such a coupling process. **The candidate's contribution was to formulate the new metadata elements, extending the ISO19115 standard; specify a test numerical model chain; populate the model chain with example data and evaluate the feasibility and performance of the metadata structures with respect to this scenario**. Observing that many spatio-temporal feature types naturally lend themselves to describing numerical model outputs, the metadata standard and the model chain represented pick out interfaces between models and define them according to certain of these feature types. The paper explores the feasibility of automated or semi-automated model coupling – using these feature types – by way of assessing whether metadata capable of supporting such coupling can be created. It was concluded that the metadata derived could support a level of temporal and spatial validation, but not full automation. Moreover, validating the parameters used between datasets was dependent on the adoption of controlled vocabularies for expressing these parameters.

To support integrated environmental modelling, structured environments for running and coupling numerical models are beginning to proliferate. For example, the Community Surface Dynamics Modelling System (CSDMS) from the USA, focuses, as its name would suggest, on modelling earth's surface systems and includes a model repository supported by a metadata structure, similar to that described in "*Towards standard metadata to support models and interfaces in a hydro-meteorological model chain*". An ICT infrastructure is provided together with workflow facilities to allow models to be coupled together. The base design for CSDMS is described by Peckham et al. (2013). Numerical models are offered to the infrastructure through adherence to the Basic Model Interface (BMI) which implements a set of simple rules for structuring the model code and accessing base functions which must be present – a set of controls and descriptive information required for a component to be deployed in a typical modelling framework. Following a similar logic, the paper "*Using a Model MAP to prepare hydro-meteorological models for generic use*" outlines the Model MAP gateway concept, a necessary collection of accompanying aspects, for preparing environmental numerical models for structured model coupling environments. **The candidate's contribution was to devise the Model MAP concept from elements being explored by the project and place it in an appropriate context with respect to numerical modelling infrastructures, as well as performing the comparison with the Component Based Water Resource Model Ontology. Also, this included the details of the metadata and adaptors, which are both built around the concept of spatio-temporal feature types. The candidate also contributed to applying the Model MAP on its originating project, the Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) as a major enabling factor, standardising the way models were offered and packaged**.

Structured environments for model coupling tend to emerge from a particular use case and then branch out into other areas. Inevitably, they carry through assumptions about the technical and functional nature of their roots. For example, the CSDMS BMI assumes that the output of a model will be represented spatially by a grid structure (Peckham, et al., 2013); most OpenMI implementations assume that timestepping will be used, even though timestepping is actually an extension to the base standard version 2.0 (OGC OpenMI 2.0, 2014). The model MAP seeks to minimise these inherent assumptions in providing an open gateway for all environmental models. As such, it offers a looser definition than that of the BMI. Moreover, the paper "*An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data*" observes that the fundamental lack of harmonisation across data products continues to hinder adoption, resulting in the under-use of many data products, yet it can give way to increased interoperability through the use of an underlying encoding of spatio-temporal feature types.

This is applied to the DRIHM eInfrastructure, described in papers "*Setup an hydro-meteo experiment in minutes: the DRIHM e-infrastructure for hydro-meteorology research*", "*The DRIHM project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological research*" and the paper, "*DRIHM(2US): an e-Science Environment for Hydro-meteorological research on high impact weather events*" which also took material from the accompanying DRIHM2US project. DRIHM seeks to enable researchers to more easily run experiments simulating hydro-meteorological events with a particular focus on flash flooding. It incorporates a structured model chain beginning with a small ensemble of meteorological models passing data through to hydrological models which, in turn, pass their output data to hydraulic models. The passage of data is one-way down the chain where the user is able to select a meteorological model which will run on an HPC infrastructure; an hydrological model which will be run on a Grid and then a hydraulic model composition which will run on the cloud. Model coupling takes place between each of these scientific domains (meteorology, hydrology, hydraulics) as well as within the hydraulic composition. **The candidate's contribution was to lead the large 'Application Services' work package which devised the information architecture used for the**

DRIHM eInfrastructure, to contribute the high level design and many associated details. This architecture is both extensible and interoperable within modelling domains as well as between modelling domains. The result was a model ensemble partitioned by model domain and integrated through file-based standards. This allowed a variety of computing infrastructures – as appropriate for the characteristics of each model domain – to be employed to run the models, forming a chain linked together using a workflow engine. Model coupling in the hydraulic domain was achieved through OpenMI. The paper "*Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy*" outlines this modelling architecture including the use of common standards and also gives some experimental results.

The paper "*A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members*" illustrates another use of spatio-temporal feature types (including Track, PointSeries and GridSeries) in coupling data sources to numerical models as part of the WaveSentry project. The feature types form the basis of the harvesting and storage of data sources in a similar way to that described in "*An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data*", although web services are not used in this case. **The candidate's contribution was to lead the WaveSentry project, producing the high level design and many of the implementation details. This included the feature-type based data harvesting scheme and the database schema optimised for feature-type structures, as well as the data comparison strategy**. This method was not considered full data ingestion since the original model results were not updated by the method, which confined itself to updating the weighting given to each ensemble member. The results were promising, giving results closer to the measured data.

Collectively and in addition to the innovations expressed in each, this series of papers points towards a structured representation of the data supporting environmental numerical models in terms of the underlying spatio-temporal characteristics. The author suggests the adoption of a new vocabulary for describing the data in this regard. This is given in Table 1. In this context, the suffix ‘Set’ indicates multiple instances of the spatial structure (point, polygon or polyline) for a single time instance or duration; the suffix ‘Series’ indicates multiple time instances or durations for a single spatial instance; the suffix ‘Track’ indicates variation in both space and time which follows a path, for example the path of a moving vessel.

Table 1: Encapsulated structure for a new representation of environmental modelled data

	Point	Polyline	Polygon
Spatial Variation	PointSet	PolylineSet	Grid, Mesh, PolygonSet
Temporal Variation	PointSeries	PolylineSeries	GridSeries, MeshSeries, PolygonSeries
Temporal and Spatial Variation	PointTrack, PointSeriesSet	PolylineTrack, PolylineSeriesSet	PolygonTrack, GridSeriesSet, MeshSeriesSet, Adaptive Grid, Adaptive Mesh

Overall, the use of feature types in model coupling is implicit and assumed in many cases and the underlying assumptions cause limitations in interoperability with numerical models from other disciplines. The significance of this work lies, in part, with the observation that overtly describing numerical model inputs and outputs using a set of spatio-temporal feature type structures will create a common vocabulary. Defining the terminology for communication in this way will foster the acceleration of the use of standards in model coupling. This will open access to a large variety of environmental numerical models.

Members of this structure demonstrated in a coupling context within the accompanying papers are as follows:

- A PointSeries is a very common spatio-temporal structure for both measured and modelled data. The catchment drainage model ‘RIBS’ produces this output for the point ‘Fereggiano’ in “*Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy*” and if this output is taken together with the PointSeries produced at ‘Stadium’ – indeed it is produced by the same model instance – then this output is an example of a PointSeriesSet.
- The PointSeriesSet from RIBS is coupled to an instance of the MASCARET model producing a PolylineSeries simulation of river levels as described in “*Using a Model MAP to prepare hydro-meteorological models for generic use*” through the WaterML2 standard (OGC WaterML2, 2012) which has since been generalised to TimeSeriesML (OGC TimeSeriesML, 2016). This PolylineSeries is then coupled to the RFSM model which itself produces flood spreading

results as a GridSeries with the Impact Calculator model producing results for damage and hazard to people, presented as a Grid.

- A GridSeries of results is also produced by the ensemble of meteorological models offered by the DRIHM e-Infrastructure as outlined in "*DRIHM(2US): an e-Science Environment for Hydro-meteorological research on high impact weather events*". This results set is passed to hydrological models through the netCDF standard (OGC CF-netCDF 1.0 standard, 2013).
- PointTrack and PointSeriesSet output is incorporated into ensemble results from an hydro-meteorological model, as described in "*A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members*". The ensemble results are structured as a GridSeriesSet.

Other members of this structure are typical to common numerical models. TELEMAC[6] is an example of a numerical model which produces results in a MeshSeries and, were it to be run in ensemble mode, it would produce results as a MeshSeriesSet. Coastline evolution models such as BeachPlan plot the movement of the coastline as a PolylineTrack. Indeed, Sutherland et al. (2013) describe the incorporation of this model into an OpenMI composition to ease coupling with other models which are also OpenMI components. The empirical wave overtopping calculation tool[7], an implementation of the empirical methods within the European Overtopping Manual[8], gives results for specific coastline profiles as they experience defined wave conditions. The results are given as scalar values which can be applied anywhere along the coast where the profile and conditions are applicable, forming a PointSet. More complex output is given

[6] http://www.opentelemac.org/

[7] http://www.overtopping-manual.com/calculation_tool.html

[8] http://www.overtopping-manual.com/manual.html

by adaptive grids and meshes where the spatial structure representing the data varies throughout the time interval covered by the model run. Another numerical model which was part of the DRIHM project, but not featured in any of the accompanying papers is Cb-TRAM (Zinner, et al., 2008) which simulates the position of thunderstorms. Figure 1 shows the output from this model, which is an example of a PolygonTrack. Several thunderstorm cells appear within the northward flow of moist air over the western Mediterranean. A large cell is located just over the city of Marseille. The nowcast indicates that it will move further inland after one hour. In the region of Genoa further east along the coastline Cb-TRAM marks the thunderstorm cell over the area which was affected by heavy flooding.

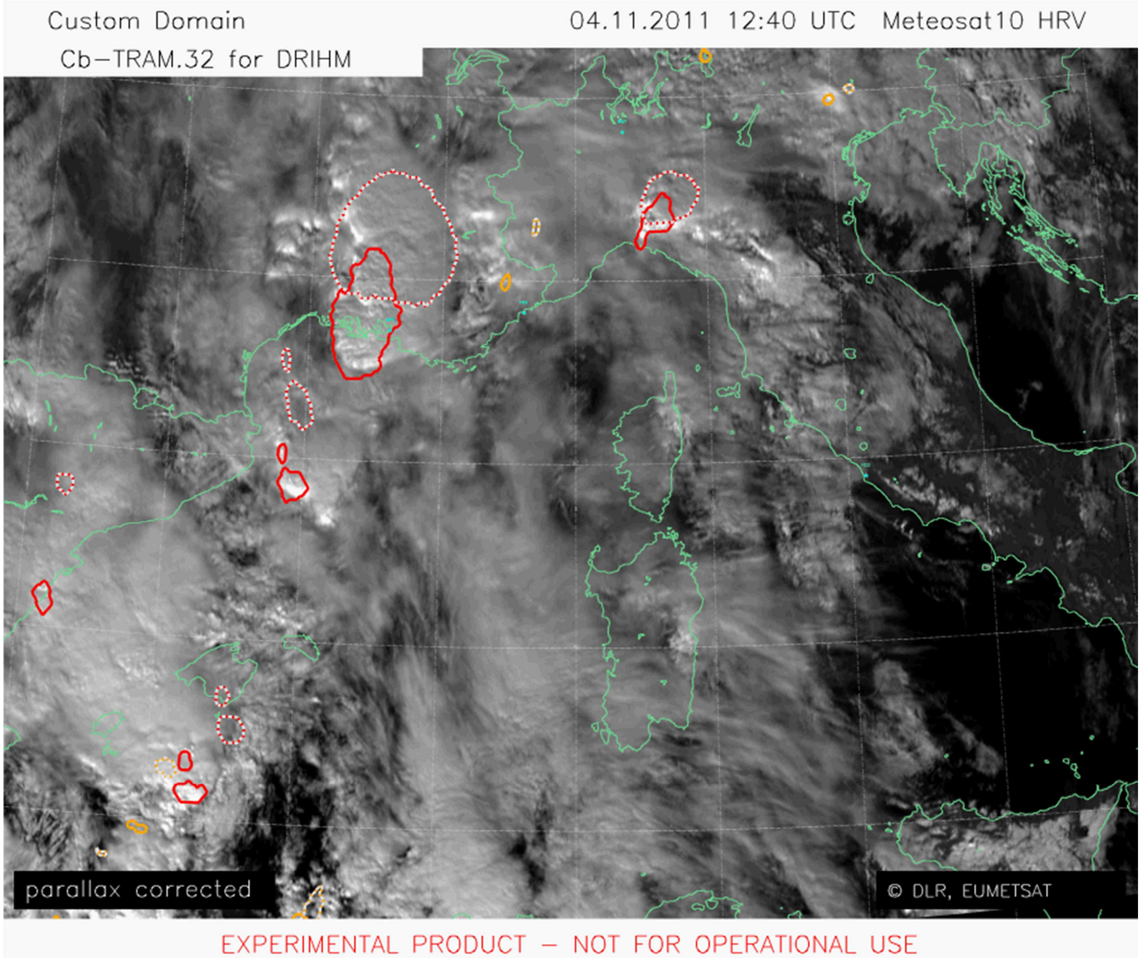

Custom Domain
Cb–TRAM.32 for DRIHM
04.11.2011 12:40 UTC Meteosat10 HRV
parallax corrected
© DLR, EUMETSAT
EXPERIMENTAL PRODUCT – NOT FOR OPERATIONAL USE

Figure 1: Cb-TRAM Thunderstorm Cells as detected (full red contours) and nowcast after one hour (dashed) by Cb-TRAM over the Western Mediterranean for 4 November 2011 12:40 UTC. Image reproduced by kind permission from Arnold Tafferner.

This representation of the data outputs from and inputs to environmental numerical models is typed to a level which coincides with many of the emerging standards for efficient storage, interrogation and clear description of spatio-temporal data. Interfacing using this level of typing would typically require analysis at a lower spatial level as demonstrated in "*A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members*", where the data is ingested by querying results from these feature-types through the points that are used to construct them. However, this vocabulary gives a strong direction as to the nature of the datasets used or produced by the models to a level where efficient, common standards are applicable. This is demonstrated by the ‘P’ (Precipitation) and ‘Q’ (Flow) interfaces used in the DRIHM e-Infrastructure as described in “*DRIHM(2US): an e-Science Environment for Hydro-meteorological research on high impact weather events*” and the two additional conference papers.

7. Conclusions and Further Work

Overall, it is given that a new representation of environmental modelled data can be attained through a set of spatio-temporal feature type structures. These together form a vocabulary to describe the data structures at interfaces between environmental numerical models with each other and with data sources. The set is formulated by applying the philosophy used within OpenMI (see "*The FluidEarth 2 implementation of OpenMI 2.0*") where time and space are handled separately, together with three basic constructs for describing spatial data within GIS: point, polyline and polygon. This structure then picks out many of the spatio-temporal data interfaces between models and data sources seen in the accompanying publications. The result is summarised in the aforementioned Table 1.

So if the vocabulary in Table 1 is used as a framework for describing potential interfaces to and from numerical models, it offers opportunity to be incorporated into standardised metadata describing the numerical models, as given in "*Towards standard metadata to support models and interfaces in a hydro-meteorological model chain*". This allows semi-automation of model interfacing. Another key enabler in this regard is the use of common vocabularies for parameter names and units, where if adopted universally – at least within modelling domains – would facilitate a new level of validation of model interfaces. Both of these vocabularies (spatio-temporal feature types and parameter names) are featured in the model MAP concept as applied in "*Using a Model MAP to prepare hydro-meteorological models for generic use*". Here, the Metadata incorporates the vocabularies, the Adaptors plot the passage of data across the interfaces and the Portability provides the technical flexibility necessary to deploy coupled models efficiently.

Adoption of this vocabulary and new representation of environmental modelled data will speed the uptake of coupling strategies and create demand for established and new standards to support integrated environmental modelling. Framing the discussion in this way allows sets of community and formal standards to be adopted for different spatio-temporal feature types thereby limiting the number of possibilities when mappings are applied between them. Moreover, a more limited – but nevertheless large – set of adaptors can be defined to apply these mappings between pairs of feature typed datasets created using these standards. Further work is required to devise appropriate standards for many of the spatio-temporal feature types offered and to examine coupling schemes between them. Some standards do exist to each cover one (or more) of these spatio-temporal feature types such as netCDF and TimeSeriesML, constructed without model coupling specifically in mind, yet with a high degree of metadata capable of supporting this activity. Controlled vocabularies for phenomena names with reference to standard catalogues are also beginning to be included.

However, even when such standards do exist, the structures and content which would support data storage or discovery is not always sufficient for use in numerical model coupling. In their study of model coupling – in particular with reference to OpenMI and high performance computing – Buahin and Horsburgh (2016) report a lack of topological information available in standardised model coupling which is specifically required for successful use of many numerical models. For example, a set of model results for a river network may be considered as a PolylineSeries. However, in common representations of results and across coupling, aspects of the network topology (e.g. direction) can be lost. This indicates that merely expressing base spatio-temporal data such as position and time in feature type standards is not sufficient and thinking has to extend to the functional nature of the target numerical models. Another example concerns the Mesh and MeshSeries structures where topological interfaces for polyhedral and triangular irregular network surfaces using the Quad-Edge data structure, as described by Guibas and Stolfi (1985), require knowledge of mesh elements to the left and right of each element edge as well as their origin and destination.

Moreover, once any two spatio-temporal feature type structures have been sufficiently well defined, a variety of adaptations are possible when coupling them together. For example, if a parameter value is given for an element of a Grid, is this value to be taken as constant across the Grid element or only at its centroid? If this Grid element is mapped to a Polyline, then which value taken by the Grid element is mapped to the points making up the Polyline? A clear understanding is therefore necessary and sufficient information must be available to avoid errors when values are passed.

Fully automated model coupling is possible, even without much of this information, but this can happen only if assumptions are made about the underlying feature types and appropriate adaptations. A greater understanding of the pattern and impact of these assumptions is required before generic, universally automated coupling schemes can be attempted. Indeed, were the innovations described in this paper to be adopted in full, further work is required to establish technical implementations suitable for a wide variety of cross-domain use cases. Even though all three of HPC, Grid and Cloud have been incorporated, more consideration is required to fully understand the variety of requirements demanded by models tailored to each of these and how the implementation may vary for each. Even though standards applicable to high and low volumes of data have been incorporated, more consideration is required to achieve optimisation for different data volumes both required by and produced by environmental numerical models.

Returning to the analogy of the development of web mapping services given in the introduction to this paper, we observe that the innovations discussed here are key elements of the journey that environmental numerical modelling is taking towards an integrated, seamless and high-usability future. This progress is following a similar pattern to that of modern web map services which have benefitted from many years of patient standardisation and integration. Whilst online mapping has developed to the point of seamless display, overlay with optical imagery and many layers of features to a suitable degree of precision for many use cases, environmental numerical modelling is still far from this level of interoperability and extensibility. Within domains progress is being made as documented by some of the papers discussed here and, although the innovations have the potential to speak across domains, a universal set of underlying principles would have considerable traction.

References

Arctur, D.K., Taylor, P., Lowe, D., Tomkins, J., Teng, W.L. and Ames, D.P. 2015 From WaterML to TimeseriesML: Evolution and implications for cross-domain data interoperability. American Geophysical Union, Fall Meeting 2015, abstract #IN23D-1750.

Buahin, C.A., and Horsburgh, J.S., 2016 From OpenMI to HydroCouple: Advancing OpenMI to Support Experimental Simulations and Standard Geospatial Datasets. International Congress on Environmental Modelling and Software. Paper 11. http://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/11.

Gregersen, J.B., Gijsbers, P.J.A. & Westen, S.J.P. 2007 OpenMI: Open modelling interface. Journal of Hydroinformatics 9(3), 175–191.

Guibas, L. and Stolfi, J., 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. ACM Transactions on Graphics, 4(2):pp. 74–123.

Guney, C., Duman, M., Uylu, K., Avci, O. and Celik, R. N. 2003 Multimedia supported GIS in the internet (Case study: two Ottoman fortresses and a cemetery on the Dardanelles), CIPA 2003 XIXth International Symposium, 30 September – 4 October 2003, Antalya, Turkey, 2003.

ISO19156 2011, *ISO 19156:2011 Geographic information – Observations and measurements*, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32574, accessed 2nd May 2014.

Lowe, D. 2011 Climate Science Modelling Language v3.0 British Atmospheric Data Centre. Available at: http://csml.badc.rl.ac.uk/ (accessed 2 May 2014).

Moore, R.V. 2010. From Google Maps to Google Models. AGU Fall Meeting, 13-17 December, San Francisco, CA.

Moore, R.V., Gijsbers, P., Fortune, D., Gregersen, J., Blind, M., Grooss, J. & Vanecek, S. 2010. OpenMI Document Series: Scope for the OpenMI (Version 2.0). Butford Technical Publishing Ltd., Pershore, UK.

OGC CF-netCDF 1.0 standard, 2013, OGC network Common Data Form (netCDF) standards suite, http://www.opengeospatial.org/standards/netcdf, accessed 21st November 2014.

OGC GML 3.3 standard, 2012, Geography Markup Language, https://www.opengeospatial.org/standards/gml, accessed 15th March 2019.

OGC OpenMI 2.0 2014, OGC Open Modelling Interface (OpenMI) Interface Standard, Open Geospatial Consortium Interface Standard, http://www.opengeospatial.org/standards/openmi, accessed 28th August 2014.

OGC TimeSeriesML, 2016, TimeseriesML 1.0 – XML Encoding of the Timeseries Profile of Observations and Measurements, http://www.opengeospatial.org/standards/tsml, accessed 23rd May 2017.

OGC WaterML 2.0, 2012 OGC WaterML 2.0 Part 1 – Timeseries. Open Geospatial Consortium Implementation Standard, http://www.opengeospatial.org/standards/waterml, accessed 2 May 2014.

Peckham, S., Hutton, E. and Norris, D. 2013. A component-based approach to integrated modeling in the geosciences: The design of CSDMS, Computers & Geosciences, V.53, 3-12. DOI: 10.1016/j.cageo.2012.04.002.

Rahim, S. T., Zheng, K., Turay, S. and Pan, Y. 1999 Capabilities of multimedia GIS, Chinese Geogr. Sci., 9(2), 159-165. 1999.

Sutherland, J., Harper, A. and Bolster, M., 2013. Beachplan as an Open-MI composition. In Proceedings of the 35th IAHR World Congress, Chengdu, China.

Taylor, P., Cox, S., Walker, G., Valentine, D. and Sheahan, P. 2014 WaterML2.0: Development of an open standard for hydrological time-series data exchange. Journal of Hydroinformatics, 16.2. 425-446.

Woolf, A., Lawrence, B., Lowry, R., Kleese van Dam, K., Cramer, R., Gutierrez, M., Kondapalli, S., Latham, S., Lowe, D., O'Neill, K. and Stephens, A. Data integration with the Climate Science Modelling Language, Adv. Geosci., 8, 83-90, doi:10.5194/adgeo-8-83-2006, 2006.

Zinner, T., Mannstein, H. and Tafferner, A., 2008. Cb-TRAM: Tracking and monitoring severe convection from onset over rapid development to mature phase using multi-channel Meteosat-8 SEVIRI data. Meteorology and Atmospheric Physics, 101(3), pp.191-210.

Appendix I: From integration to fusion: the challenges ahead

Available from: http://dx.doi.org/10.1144/SP408.6

Appendix II: An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data

Ocean Sci., 7, 445–454, 2011
www.ocean-sci.net/7/445/2011/
doi:10.5194/os-7-445-2011

Ocean Science

An ECOOP web portal for visualising and comparing distributed coastal oceanography model and in situ data

A. L. Gemmell[1], R. M. Barciela[2], J. D. Blower[1], K. Haines[1], Q. Harpham[3], K. Millard[3], M. R. Price[2], and A. Saulter[2]

[1]Environmental Systems Science Centre, Harry Pitt Building, 3 Earley Gate, Whiteknights, University of Reading,
Reading, RG6 6AL, UK
[2]The Met Office, Fitzroy Road, Exeter, Devon, EX1 3PB, UK
[3]HR Wallingford, Howbery Park, Wallingford, Oxfordshire, OX10 8BA, UK

Received: 31 December 2010 – Published in Ocean Sci. Discuss.: 28 January 2011
Revised: 17 May 2011 – Accepted: 17 June 2011 – Published: 4 July 2011

Abstract. As part of a large European coastal operational oceanography project (ECOOP), we have developed a web portal for the display and comparison of model and in situ marine data. The distributed model and in situ datasets are accessed via an Open Geospatial Consortium Web Map Service (WMS) and Web Feature Service (WFS) respectively. These services were developed independently and readily integrated for the purposes of the ECOOP project, illustrating the ease of interoperability resulting from adherence to international standards.

The key feature of the portal is the ability to display coplotted timeseries of the in situ and model data and the quantification of misfits between the two. By using standards-based web technology we allow the user to quickly and easily explore over twenty model data feeds and compare these with dozens of in situ data feeds without being concerned with the low level details of differing file formats or the physical location of the data.

Scientific and operational benefits to this work include model validation, quality control of observations, data assimilation and decision support in near real time. In these areas it is essential to be able to bring different data streams together from often disparate locations.

1 Introduction

Marine scientists use highly diverse sources of data, including in situ measurements, remotely-sensed information and the results of numerical simulations. The ability to access, visualize, combine and compare these datasets is at the core of scientific investigation, but such tasks have hitherto been hindered by a fundamental lack of harmonization across data products and the lack of fast efficient online tools to exploit marine datasets available through the internet. As a result, much valuable data remains underused. As models become larger and increasingly complex, and sources of observed data become more numerous, it is important to be able to access and compare this growing amount of data efficiently, to ensure cross-checking and consistency between models and observations.

The advent of easy-to-use, consumer-focused tools such as Google Earth and Google Maps has transformed the way that geospatial data is presented on the internet (Peterson, 2008; Gibin et al., 2008) and there has been increasing interest from the scientific community to develop similar fast easy tools for exploring data. Meanwhile the EU has issued the INSPIRE directive (INfrastructure for SPatial InfoRmation in Europe initiative, http://inspire.jrc.it), which mandates the use of international standards in the dissemination of public geospatial data. The Open Geospatial Consortium (OGC) has been instrumental in developing and promoting standards for representing and exchanging geospatial data, and many of its standards are mandated by INSPIRE, notably the Web Map Service (WMS, http://www.opengeospatial.org/standards/wms) for map imagery and the Web Feature Service (WFS, http://www.opengeospatial.org/standards/wfs) for geospatial data. These standards have evolved from the domain of Geographic Information Systems (GIS), which have historically been concerned mainly with two-dimensional land-based data (Rahim et al., 1999; Guney et al., 2003). However scientific description or modelling of the environment usually involves 4-D data (3-D data evolving in time) as needed to describe the atmosphere or ocean properties. The NetCDF file format (http://www.unidata.ucar.edu/software/netcdf/) has become a widely used

Correspondence to: A. L. Gemmell
(a.l.gemmell@reading.ac.uk)

Published by Copernicus Publications on behalf of the European Geosciences Union.

standard for storing such dense multi-dimensional data, along with the Climate and Forecast (CF) metadata convention for describing the content of NetCDF files in their file headers (Blower et al., 2009a). There is much current research interest in bringing together the worlds of GIS and 4-D environmental data to develop "4-D GIS" systems. Groups have therefore developed OGC-based systems for encoding environmental data (e.g. CSML – Climate Science Modelling Language, Woolf et al., 2006; Marine Markup Language, http://www.ercim.eu/publication/Ercim_News/enw57/matthews.html), serving data (e.g. Best et al., 2007; de La Beaujardiere, 2009) and visualizing data (e.g. Blower et al., 2009b; Wei et al., 2009; Huang, 2003).

The Godiva2 project (Blower et al., 2009b) provides the starting point for the work presented here. It provides an efficient means of exploring 4-D environmental model data by generating 2-D maps or 3-D map-movies from data in CF-NetCDF files for remote viewing on an interactive interface based upon OpenLayers, an open-source browser-based map visualization library. The project uses the ncWMS software (http://ncwms.sf.net/) which generates 2-D maps fast enough for use in real-time interactive data browsing of large datasets. This software has been widely adopted by research institutes, government agencies and private industry for presenting operational marine forecasts (e.g. at the UK Met Office) and satellite imagery (e.g. NEODAAS, Plymouth Marine Laboratory). The software has also been adopted as the basis of the viewing interface for the GMES Marine Core Services project MyOcean (http://www.myocean.eu).

The aim of the European COastal sea Operational Observing and forecasting system Project (ECOOP, www.ecoop.eu) was to "build up a sustainable pan-European capacity in providing timely, quality assured marine services (including data, information products, knowledge and scientific advices) in European coastal-shelf seas". A key requirement was to develop a web portal that visualises and compares physical and biological marine data from both numerical models and in situ observations. We discuss the technical choices for viewing and interacting with the in situ data and relating it to the gridded model data. We also showcase the achievements of the ECOOP portal in its final form and go on to discuss the lessons learned and the further developments required in order to improve the system for future scientific applications requiring the viewing of combined model and observational datasets.

Section 2 discusses the datasets available through the ECOOP project as examples of the challenges required to be overcome. Section 3 first discusses the technical options for incorporating point data into the Godiva2 map viewing tool. We then discuss the architecture chosen and finally the scope of the ECOOP portal that was operational at the end of the project. Section 4 discusses some of the scientific uses this portal has been put to and Sect. 5 provides further discussion and conclusions particularly on the strengths and weaknesses of the current system and the plans for future developments.

2 Marine dataset distribution within Europe

2.1 Model forecast data

Many groups in Europe are now involved in operational ocean modelling and are able to provide daily or weekly forecasts of marine conditions in local European coastal sea areas. The ECOOP project provided 23 different model data feeds to the web portal from 14 different ECOOP partners. In order to view all these data in a single web portal the model data need to be rendered into map images. These images could be rendered at each data provider using a Web Map Service such as ncWMS (a federated approach), or they could be rendered at the central server at the University of Reading (a centralized approach), with the data being accessed from the data providers via the OPeNDAP protocol. The use of OPeNDAP to serve data was mandatory in the project and so the easiest solution from the point of view of the data providers was the centralized approach, with most data feeds being sent to a single WMS for rendering into images. However, two partners (UK Met Office and Plymouth Marine Laboratory) were able to run their own ncWMS servers and therefore provide imagery directly to the web portal. In future the federated approach is preferred, as this avoids a data bottleneck at the central server; this approach will be taken in the MyOcean project. The latest version of THREDDS (v4.2) includes an OPeNDAP service together with an embedded ncWMS service, therefore data providers will in future be able to provide both types of service using the same software. The complete list of model forecast providers can be found in Table 1 and Fig. 1 shows the data regions of model output provided. Note that there is a very low barrier to entry for the serving of additional CF-compliant model data using our system. If the data providers provide data through an OPeNDAP server, then it is a trivial matter to incorporate these new data feeds into the portal.

2.2 In situ observational data

The range of institutes involved in observational monitoring of European coastal seas is very large and it would not have been possible to set up data feeds for all providers. Fortunately the EU-SEPRISE project (Sustained, Efficient Production of Required Information Services, http://www.seprise.eu) which ran from 2004 to 2006 already gathered many such observational timeseries together and provides an ongoing single FTP point of delivery from SMHI (the Swedish Meteorological and Hydrological Institute). In total SEPRISE provides data from 45 institutes in 24 countries throughout Europe with data updated on a daily basis. FTP is not a convenient mechanism for incorporating data into the ECOOP web portal (for example it does not provide the ability to intelligently filter data), so the data were copied and served via a Web Feature Service (see Sect. 3) at HR Wallingford, with data formatted as CSML FeatureTypes.

Table 1. Full list of ECOOP partner institutes providing model data to the portal. Numbers match with model areas shown in Fig. 1.

Institute	Country
1. Bundesamt für Seeschifffahrt und Hydrographie (BSH)	Germany
2. Danish Meteorological Institute (DMI)	Denmark
3. Mercator	France
4. Previmer	France
5. Maretec	Portugal
6. The Marine Institute	Ireland
7. UK Meteorological Office (UKMO)	UK
8. Plymouth Marine Laboratory (PML)	UK
9. Istituto Nazionale di Geofisica e Vulcanologia (INGV)	Italy
10. University of Athens	Greece
11. Institute of Oceanology (IO-BAS)	Bulgaria
12. National Institute for Marine Research and Development (NIMRD)	Romania
13. Marine Hydrophysical Institute (MHI)	Ukraine
14. Mediterranean Institute for Advanced Studies (IMEDEA)	Spain

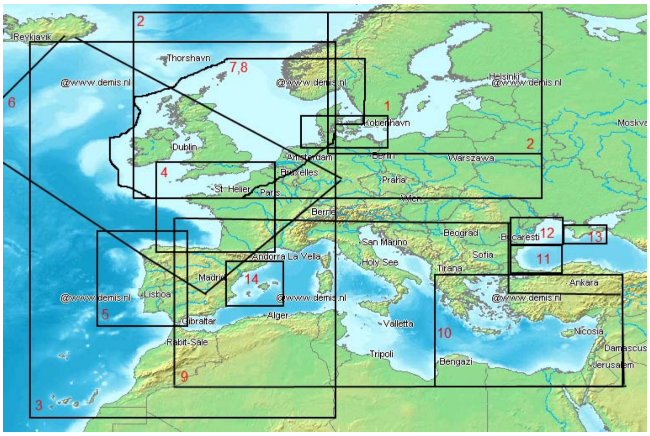


Fig. 1. The model regions available in the portal. Numbers represent the institutes serving the model data as per Table 1.

SEPRISE data only contain physical ocean variables such as temperature, salinity and current data. CEFAS (Centre for the Environment, Fisheries and Aquaculture Science), collect data from around the UK coasts using “SmartBuoys” which monitor both physical and biogeochemical variables which are of interest for an increasing number of applications. These SmartBuoy data were also obtained daily by HR Wallingford and served to the ECOOP portal via WFS in the same uniform CSML format as the SEPRISE data. The full range of model and in situ data which may be compared using the portal is given in Table 2.

3 Technical approach

The architecture of the system is illustrated in Fig. 2. The ncWMS software has been described elsewhere (Blower et al., 2009b) therefore in the following sections we focus on the other elements of the system – the use of WFS for serving in situ data (Sect. 3.1) and the web portal itself (Sect. 3.2).

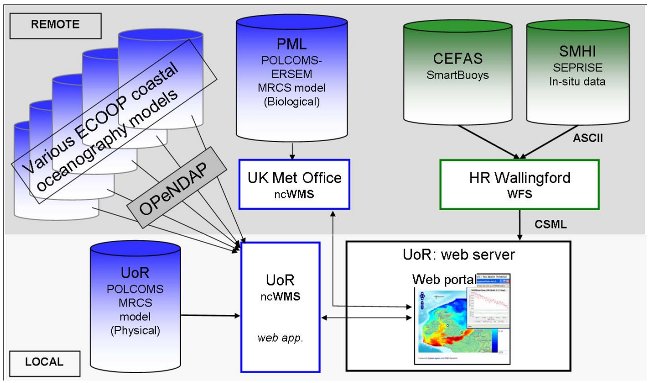


Fig. 2. Architecture of the system. Model data (blue) are ingested into the portal via an instance of ncWMS at the University of Reading (UoR), and an instance of ncWMS at the UK Met Office. in situ data (green) are ingested into the portal via the WFS at HR Wallingford serving CSML XML.

The WFS for serving the portal with in situ point data, and the WMS for serving the portal with gridded model data were developed independently, and only integrated later for the purposes of the work presented here. This modular approach increased flexibility and allowed both sides of the system to be developed and upgraded independently. Integration of the two data streams at the portal was facilitated by support for WMS layers and point features in the OpenLayers library (see Sect. 3.2).

3.1 Standards-based serving of in situ data

An OGC-compliant Web Feature Service was chosen as the means of standardising the in situ data ready for ingestion

Table 2. Model data types and their corresponding in situ data types, sources, and nature of comparison. Where the comparison is given as qualitative this indicates that a dual axis plot of both data are shown, and qualitative correlations can be expected.

Model data type	in situ data type	in situ data source	Comparison
Sea water temperature (Celsius)	Sea water temperature (Celsius)	SEPRISE, SmartBuoys	Exact match
Sea water salinity (P.S.U)	Sea water salinity (P.S.U)	SEPRISE, SmartBuoys	Exact match
Sea water velocity ($\mathrm{m\,s^{-1}}$)	Sea water velocity ($\mathrm{m\,s^{-1}}$)	SEPRISE	Exact match
Chlorophyll-*a* ($\mathrm{mg\,m^{-3}}$)	Fluorescence (Arbitrary Unit)	SmartBuoys	Qualitative
Dissolved Oxygen Conc. ($\mathrm{m\,mol\,m^{-3}}$)	Oxygen saturation (%)	SmartBuoys (Liverpool Bay, Warp Estuary only)	Converted according to Weiss (1970)
Photosynthetically active radiation ($\mathrm{W\,m^{-2}}$)	Irradiance ($\mathrm{E} \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$)	SmartBuoys	Qualitative
Suspended particulate matter (35 µ) ($\mathrm{kg\,m^{-3}}$)	Turbidity (F.T.U.)	SmartBuoys	Qualitative
Suspended particulate matter (2 µ) ($\mathrm{kg\,m^{-3}}$)	Turbidity (F.T.U.)	SmartBuoys	Qualitative

into the portal owing to its adherence to recognised international standards, complementing the existing OGC Web Map Service used for the model data, and increasing the reusability of existing code and tools. The OGC WFS standard specifies that data should be encoded as XML adhering to a GML application schema. For this we chose to use CSML, and in particular the PointSeries FeatureType. CSML is specifically tailored to represent features of relevance to the climate sciences and comprises 13 FeatureTypes, of which the PointSeriesFeature represents a timeseries at a fixed location.

There are two different queries which are made in order to retrieve the in situ data. The first step is a query to determine which of the geographically static in situ stations were actively measuring data for the dates and parameter of interest. The response is a set of locations of in situ stations. When one of these stations is interrogated the second query is to request the data from that station for a period of time. These two queries both correspond to GetFeature requests within the OGC WFS standard. In the case of the first query an example query string (minus this initial server URL and port number) is:

wfs?request=GetFeature&service=WFS&version=1.1.0&
typeName=hrw:ECOOPTimeSeries.

In the case of the second query an example query string (minus the initial server URL and port number) is:

wfs?request=GetFeature&service=WFS&version=1.1.0&
typeName=ldip:ECOOPSmartBuoyTimeSeries&filter=FILTER

where "FILTER" is a placeholder for an XML string to filter the results which adheres to the OGC Filter Encoding Implementation specification (http://www.opengeospatial.org/standards/filter). For example, the filter is often used to select only a single FeatureType based on its ID and the parameter being measured. In this case, the required start and end times of the in situ data were also used in the filter.

However, here we are extending the WFS specification somewhat. The logical Features in question are CSML PointSeries features. Each Feature is an entire timeseries, which may be very long. For our application, we need to access subsets of these features, which are themselves PointSeriesFeatures. There is no support in version 1 of the WFS standard for subsetting a feature (features must be served whole), and hence our WFS implementation (GeoServer) was not able to support this requirement in a straightforward manner. The system was therefore designed so that each individual measurement in each timeseries was stored as a point measurement in the database that sits behind the WFS. The request for a particular time range leads to the extraction of a number of these individual point features, which are produced by GeoServer as a single XML document. This document (containing several point features) is then transformed by an XSLT transformation into a CSML document which contains a single PointSeriesFeature. The net effect is that the user of the WFS can request subsets of logical PointSeriesFeatures; this extends the standard but was necessary to fulfil the requirements of the project. We discuss in Sect. 5 below possible alternatives to this architecture.

Figure 3 is a sequence diagram illustrating the actors, requests and responses involved in the system. The first step in ingesting the in situ data is a query to HR Wallingford to

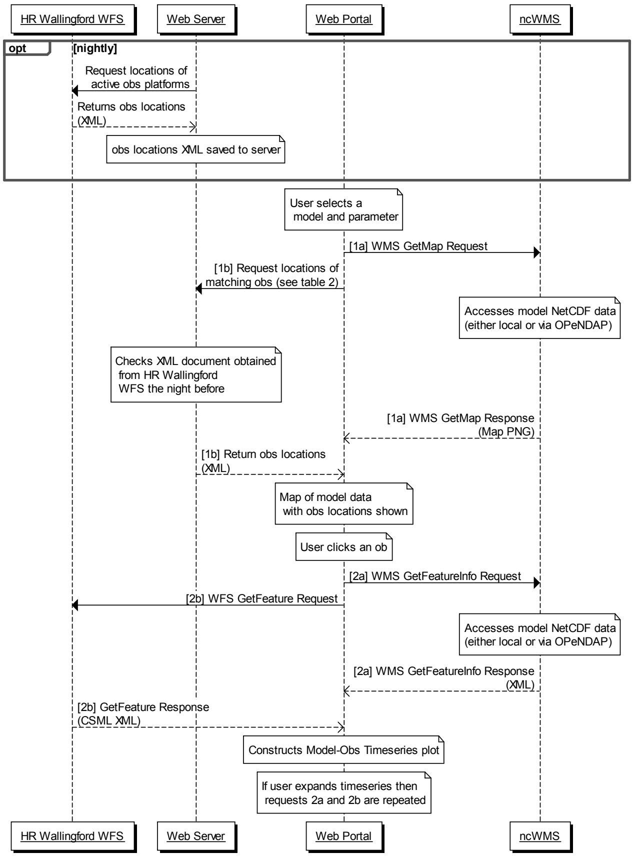

HR Wallingford WFS
Web Server
Web Portal
ncWMS
opt
[nightly]
Request locations of active obs platforms
Returns obs locations (XML)
obs locations XML saved to server
User selects a model and parameter
[1a] WMS GetMap Request
[1b] Request locations of matching obs (see table 2)
Accesses model NetCDF data (either local or via OPeNDAP)
Checks XML document obtained from HR Wallingford WFS the night before
[1a] WMS GetMap Response (Map PNG)
[1b] Return obs locations (XML)
Map of model data with obs locations shown
User clicks an ob
[2a] WMS GetFeatureInfo Request
[2b] WFS GetFeature Request
Accesses model NetCDF data (either local or via OPeNDAP)
[2a] WMS GetFeatureInfo Response (XML)
[2b] GetFeature Response (CSML XML)
Constructs Model-Obs Timeseries plot
If user expands timeseries then requests 2a and 2b are repeated
HR Wallingford WFS
Web Server
Web Portal
ncWMS

Fig. 3. Sequence diagram showing the calls made from the Godiva2 web portal (running in the browser client) to request the model and obs data, as well as the responses returned.

```
<PointSeriesFeature gml:id="LIVBAYFluorescence">
 <gml:description>
  ECOOP Smart Buoy Liverpool Bay Coastal Observatory Measuring Fluorescence
 </gml:description>

 <csml:location srsName="urn:EPSG:geographicCRS:4326">-3.3578 53.5345</csml:location>
 <csml:value>
  <csml:PointSeriesCoverage gml:id="ECOOPSmartBuoy.cov">
   <csml:pointSeriesDomain>
    <csml:TimeSeries gml:id="ECOOPSmartBuoy.cov.times">
     <csml:timePositionList>
      2008-06-03T01:59:06 2008-06-03T03:59:06 2008-06-03T05:59:06 … 2008-06-10T21:59:06
     </csml:timePositionList>
    </csml:TimeSeries>
   </csml:pointSeriesDomain>

   <gml:rangeSet>
    <gml:QuantityList uom="arbitrary">
     8.61375702075546 7.94905128204984 6.49687423687306 … 1.83973382173349
    </gml:QuantityList>
   </gml:rangeSet>
  </csml:PointSeriesCoverage>
 </csml:value>

 <csml:parameter>
  <swe:Phenomenon gml:id="Fluorescence">
   <gml:identifier codeSpace="http://www.cfconventions.org">Fluorescence</gml:identifier>
  </swe:Phenomenon>
 </csml:parameter>
</PointSeriesFeature>
```

Fig. 4. Example of CSML XML returned from the WFS, representing sea water temperature data from the Liverpool Bay SmartBuoy. Note that the 10-day list of times and values as been truncated for brevity.

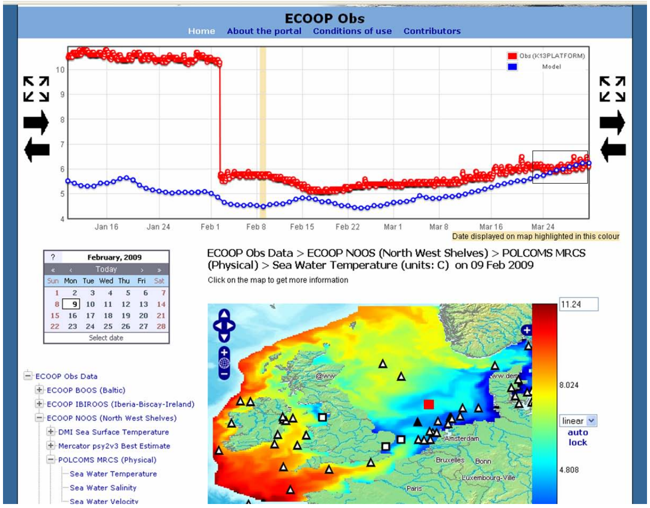
determine what data are present for the dates and parameter of interest being currently displayed as a model field in the ECOOP portal. It was important to make the process of returning the positions of the available in situ data as efficient as possible to avoid a prolonged wait before the locations of the platforms can be displayed in the portal. However, it was taking several minutes for the WFS to filter the more than 800 FeatureTypes being served to return the locations of relevant data. This bottleneck was eventually avoided because the in situ data are only updated once per day so it was possible to query the WFS server each night and save the resulting XML document of CSML FeatureTypes locally. This caching reduces the time to query the data from several minutes to seconds. Work is ongoing to increase the WFS efficiency (see Sect. 5) and preliminary results indicate that server caching of data will no longer be necessary in the next generation of the HR Wallingford WFS.

In Fig. 3 the request 2b is only enacted when the client clicks on one of the in situ data icons on the portal, at which point that actual in situ data item is retrieved for display. The result of such a query is a CSML document, e.g. Fig. 4 representing sea water temperature data from the Liverpool Bay SmartBuoy. Note that the 10-day list of times and values has been truncated for brevity.

3.2 The web portal

It can be seen from Fig. 3 that the web portal coordinates the delivery and integration of data from the various sources. The web portal runs entirely in the browser client (http://www.resc.reading.ac.uk/ecoop_obs_portal) and its primary roles are to (a) respond to the user's requests and to delegate these requests to the relevant actor and (b) receive the respective responses and present these to the user including combining of multiple responses where appropriate. The portal employs two Javascript libraries in order to fulfil these roles – OpenLayers (openlayers.org) and Flot (http://code.google.com/p/flot).

The OpenLayers mapping library is used for displaying the map images of gridded model data and the location markers for the in situ data. In order to display the in situ data locations and points superimposed on the model map images we make use of OpenLayers support for layers of points (known as markers in OpenLayers). These markers can be configured to respond to mouse events allowing the user to click on an observation and request the observed and model data from that location (requests 2a and 2b in Fig. 3). The use of OpenLayers markers is suitable in this situation as there are of the order of 100 markers. As each one is a distinct object in the browser's Document Object Model (DOM), and therefore takes a certain amount of memory, the solution does not scale to very large numbers (thousands or tens of thousands) of observations without the browser becoming unresponsive. In this situation an alternative would be to use the OpenLayers



ECOOP Obs Data > ECOOP NOOS (North West Shelves) > POLCOMS MRCS (Physical) > Sea Water Temperature (units: C) on 09 Feb 2009

Click on the map to get more information

Fig. 5. Portal screenshot showing an extended timeseries of sea water temperature (red line, two-hourly data feed) from the K13 Platform in the North Sea (solid black triangle) and from the POLCOMS MRCS model (blue line, daily average). The elevated temperatures present in the observed data for the first third of the timeseries are erroneous, and the correction of this problem is clearly seen as later temperatures are in much closer agreement with the model data. The black rectangle on the right of the timeseries plot denotes the area of the zoomed-in timeseries in Fig. 7, which also contains a timeseries from the OysterGround SmartBuoy to the North East of the K13 platform (red square).

cluster strategy or to render the observed locations onto an image overlay.

When in situ data are requested they are displayed above the map in a timeseries, along with an equivalent timeseries sampled from the model data being concurrently displayed. These timeseries plots, which can be seen in Figs. 5–8, are produced by the Flot graphing library. This graph can be zoomed for more detail in a particular region, and the user can mouse-over the data points to reveal their precise time and value. The start and end of the timeseries can be incrementally expanded or contracted. Each time this happens the data are cached, meaning that the timeseries can be contracted and expanded again without unnecessary calls to the server. Figure 5 illustrates a view of the portal after requests 1a, 1b, 2a, and 2b have been successfully executed. In addition, the default timeseries of 10 days has been expanded a number of times. One benefit of using a dynamic plotting library as opposed to static plots is the ability to correct on-the-fly for rogue values that distort the y-axis scaling as illustrated in Fig. 6.

4 Scientific and operational applications

There are many benefits that may be derived from the ability to quickly and easily compare model and in situ data over

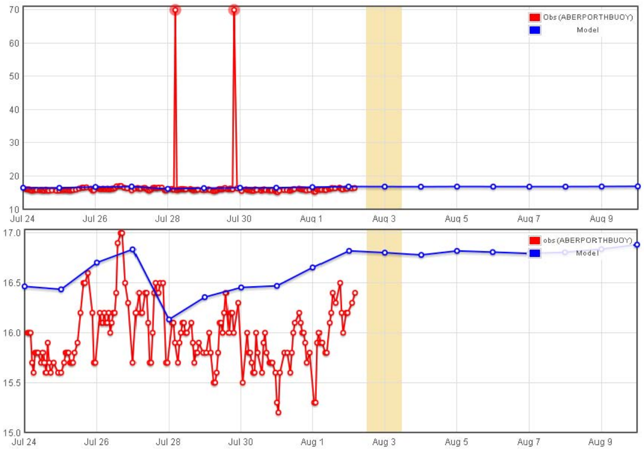


Fig. 6. Sea water temperature (°C) from the AberporthBuoy platform. One of the benefits of using a dynamic plotting library in the web portal is that if rogue data points are disturbing the y-axis scaling, they may be highlighted (top panel) and removed, resulting in auto-scaling of the y-axis to the correct range for the genuine data values (bottom panel).

the web. Calibrating observations against model background data in order to detect biases and other gross errors is one application. Once the observations are calibrated they can be used for testing the detailed accuracy of the numerical models. The best analysis should come from combining model and observations in a data assimilation process and this can benefit greatly from displaying the results before and after assimilation along with either assimilated or independent data, or displaying the success of a forecast made using assimilated initial conditions. Decision support systems will benefit from the display of multiple model and data streams together to add confidence to the decision making process. In the following sections we describe examples of the benefits which the current work brings to these areas.

4.1 Observational quality control

It is often assumed that in situ observations represent the “truth” to which model and remotely-sensed data should be compared in order to improve their accuracy. This is not the case, as in situ instruments are simply attempting to measure the true state of the system, and are subject to errors and biases in doing so. One can consider two major categories of errors in in situ measurements:

1. Accuracy errors inherent in the instrument, e.g. a tide gauge may be capable of measuring sea surface height only to within 1 cm.

2. Gross errors and biases due to instrument or retrieval failures. For example, an instrument becomes fouled by debris and records incorrect values of suspended matter.

Type 1 errors sometimes appear as quantizations in timeseries plots – the observed parameter only takes up certain values resolved by the instrument. Type 2 errors are often unexpected and may be identified through routine comparisons with other data sources, such as model background data. Figure 5 shows one of the SEPRISE platforms initially displaying erroneous temperature data, which was noticed upon comparison with POLCOMS MRCS temperature fields in the portal, prior to 1 February 2009. The data acquisition and transfer process was checked by staff at SMHI who discovered the error and rectified the problem (the data feed had erroneously been coming from another buoy entirely) after 1 February. Viewed in isolation, prior to comparison with model data within the portal, the excessively high observed temperatures had not been noticed.

Figure 7 shows a zoom of the model data matchup for both the SEPRISE data in Fig. 5 (upper panel) and for the SmartBuoy data about 100 km to the North East of the SEPRISE location in Fig. 5 (lower panel). The zoomed-in plot shows up diurnal and tidal timescales for the same period. It is noticeable that, particularly in the early phase of the timeseries, the SEPRISE data illustrate the quantization phenomenon described above, whereas the SmartBuoy data show a more reliable picture of diurnal variation in sea surface temperatures. It is clear from both timeseries that there is an increase in the diurnal and tidal signal during and after 28 March. In the case of the SEPRISE data this signal is dominated by an approximately 24 h cycle, with a weaker approximately 12 h cycle, whereas the two frequencies are more equally represented in the SmartBuoy data. This type of matchup across instruments and nearby locations demonstrates how the ability to cross reference observational and model datasets provides interesting and useful calibration information.

4.2 Validation of ocean models

The ability to compare models with in situ observations is critical to the model validation and improvement process. The present work ensures that there is a low barrier to model validation against in situ observations by bringing the two datasets together in timeseries plots. In Fig. 5 during the initial portion of the timeseries the model acts as a test for the observed data (Sect. 4.1), while during the later period, after correction, the observed data acts as a test for the model data. One can see that there is overall agreement between observation and model for this portion of the timeseries, with both datasets starting to show a slow warming into Spring. Although the model starts off too cold by about 1 °C, this deficit disappears by the end of March. This model is not run with a diurnal cycle forcing and so we are unable to test the difference in diurnal behaviour noted in Fig. 7.

Another example of model validation is shown in Fig. 8 in which ocean velocity output from the University of Athens Aegean and Levantine Eddy Resolving MOdel (ALERMO, Korres and Lascaratos, 2003) is compared with current meter data from the E1M3ACRETANSEA in situ station. ALERMO has a horizontal resolution of 1/30° × 1/30° and a vertical resolution of 25 logarithmically distributed sigma levels. ALERMO is one-way coupled with the SKIRON weather forecasting system (Kallos, 1997) which provides air temperature and relative humidity at 2 m above the sea surface, wind velocity at 10 m, sea level atmospheric pressure, net shortwave radiation at the sea surface, downward longwave radiation and precipitation rate. Here we can see that the model daily average output for velocity is underrepresenting the daily mean velocities that we could infer from the in situ station. This is not entirely surprising given the model resolution and forcing used, but it does give an immediate quantitative evaluation of the model discrepancies. This example also illustrates a potential pitfall when comparing model and in situ velocities with differing time resolution. The model velocity is a daily mean, whereas the in situ velocities are instantaneous. If the latter were averaged to create a daily mean then this could be lower than a simple numerical average of the velocity magnitudes owing to potential changes in current direction.

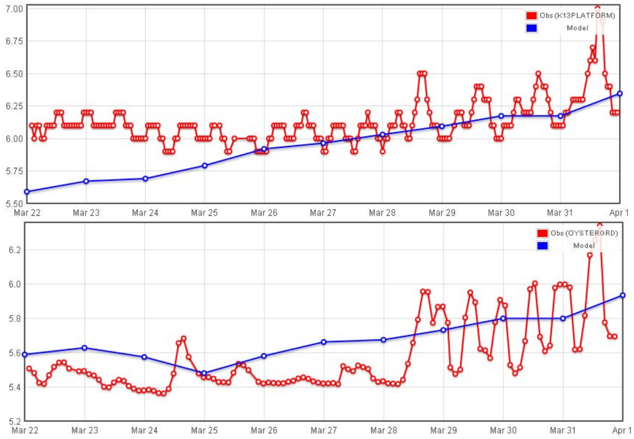
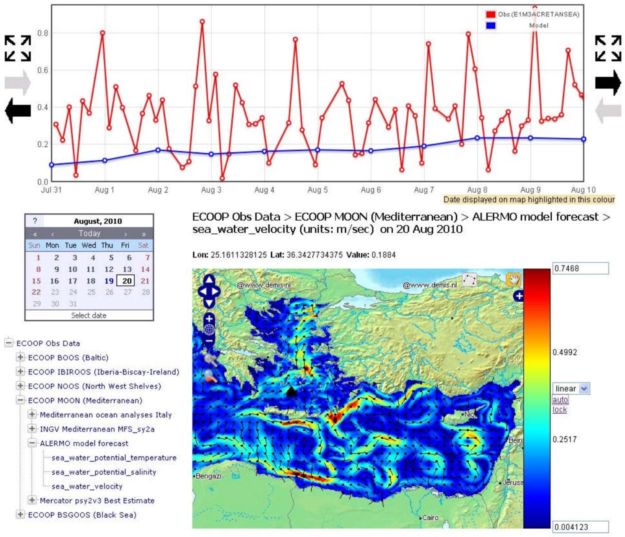


Fig. 7. Zoomed in timesereies from Fig. 5 of sea water temperature (°C) showing diurnal and tidal variation in the K13 observing platform and MRCS model (upper panel). Also comparing with the OysterGround SmartBuoy to the north east of the K13 platform (lower panel).

4.3 Data assimilation systems

Figure 9 shows a timeseries of sea water temperature from the SEPRISE platform Frederica (solid black triangle to the east of Denmark on the map) superimposed on the Mercator psy2v3 model surface temperature. The lower timeseries shows the observations in red from the Frederica buoy up till 25 November 2009 along with the model background data in blue, prior to assimilation of observations from the previous 7 days. The upper timeseries shows the revised model best estimate and seven day forecast out to 2 December, made on



ECOOP Obs Data > ECOOP MOON (Mediterranean) > ALERMO model forecast > sea_water_velocity (units: m/sec) on 20 Aug 2010

Lon: 25.1611328125 Lat: 36.3427734375 Value: 0.1884

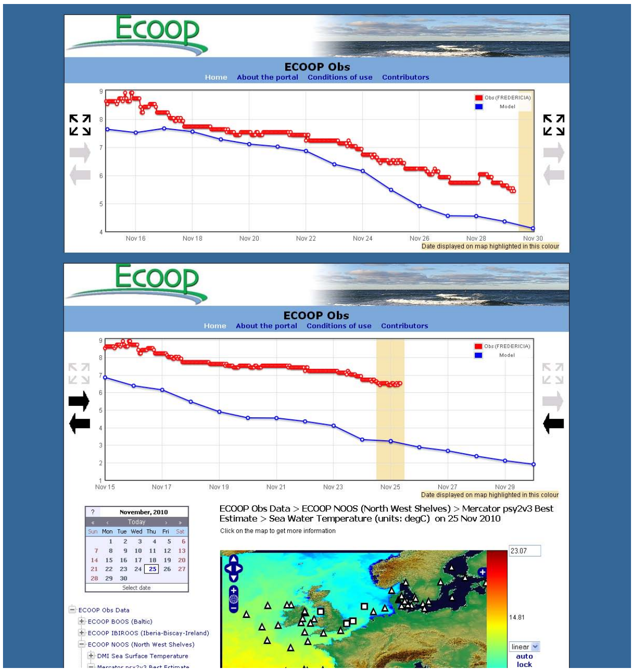
Obs (E1M3ACRETANSEA)
Model
0.0
0.2
0.4
0.6
0.8
Jul 31
Aug 1
Aug 2
Aug 3
Aug 4
Aug 5
Aug 6
Aug 7
Aug 8
Aug 9
Aug 10
Date displayed on map highlighted in this colour
?
August, 2010
Today
Sun Mon Tue Wed Thu Fri Sat
Select date
ECOOP Obs Data
ECOOP BOOS (Baltic)
ECOOP IBIROOS (Iberia-Biscay-Ireland)
ECOOP NOOS (North West Shelves)
ECOOP MOON (Mediterranean)
Mediterranean ocean analyses Italy
INGV Mediterranean MFS_sy2a
ALERMO model forecast
sea_water_potential_temperature
sea_water_potential_salinity
sea_water_velocity
Mercator psy2v3 Best Estimate
ECOOP BSGOOS (Black Sea)
@www.demis.nl
Bengazi
Cairo
0.7468
0.4992
0.2517
0.004123
linear
auto
lock

Fig. 8. Sea water velocity output from the ALERMO model from the University of Athens compared to current meter data from the E1M3ACRETANSEA in situ station (black triangle).

25 November utilising all the observations up till that date. The observations in this case are taken further forward until 2 December to show the level of agreement. Note that the SEPRISE buoy data being used here is independent and will not have been included in the assimilation procedure. It can be seen that although these buoy data have not been assimilated, the actual variations in SST through the period are better reproduced from the best estimate products. Comparisons such as this shown in a screenshot from our web portal allow scientists and users to interpret and use the model forecast and analysis results much more easily without specialist knowledge of the data sets. They can compare for themselves the various models and forecasts with the in situ data both before and after data have been assimilated into the model output.

4.4 Decision support in near real time

Operational and pre-operational physical-biogeochemical models routinely generating ecological products now exist (Siddorn et al., 2007; Brasseur et al., 2009) and their products, including forecasts, are being disseminated in near-real time to end-users (www.myocean.eu). The South West Algal Pilot Project (SWAPP) and its successor, the AlgaRisk project (www.npm.ac.uk/rsg/projects/algarisk, Barciela et al., 2009), assessed and demonstrated the feasibility of this approach for forecasting algal blooms affecting the coastal waters of the UK, through the combination of satellite observations, model and meteorological data (Mahdon et al., 2010). Other initiatives, such as the European Marine Strategy Framework Directive, are likely to benefit from incorporating web technology, such as the web portal described here, as part of a wider set of monitoring tools required to establish a successful environmental monitoring programme. It is important that these tools use standards to guarantee interoperability of different national components at larger pan-European scales..



ECOOP Obs Data > ECOOP NOOS (North West Shelves) > Mercator psy2v3 Best Estimate > Sea Water Temperature (units: degC) on 25 Nov 2010

Click on the map to get more information

Ecoop
ECOOP Obs
Home About the portal Conditions of use Contributors
Obs (FREDERICIA)
Model
4
5
6
7
8
9
Nov 16
Nov 18
Nov 20
Nov 22
Nov 24
Nov 26
Nov 28
Nov 30
Date displayed on map highlighted in this colour
Ecoop
ECOOP Obs
Home About the portal Conditions of use Contributors
Obs (FREDERICIA)
Model
1
2
3
4
5
6
7
8
9
Nov 15
Nov 17
Nov 19
Nov 21
Nov 23
Nov 25
Nov 27
Nov 29
Date displayed on map highlighted in this colour
?
November, 2010
Today
Sun Mon Tue Wed Thu Fri Sat
Select date
ECOOP Obs Data
ECOOP BOOS (Baltic)
ECOOP IBIROOS (Iberia-Biscay-Ireland)
ECOOP NOOS (North West Shelves)
DMI Sea Surface Temperature
Mercator psy2v3 Best Estimate
23.07
14.81
linear
auto
lock

Fig. 9. Portal screenshot showing a timeseries of sea water temperature (red line, two-hourly data feed) from the Frederica platform off the East coast of Denmark (solid black triangle) and from the Mercator psy2v3 model (blue line, daily average). The lower timeseries is from 25 November, and the upper timeseries is from 30 November. Note that the more recent timeseries represents a best estimate based on more available observed data which have been assimilated into the model, and hence shows a better fit with the Frederica in situ data feed.

This web portal provides an important step in this direction. It was originally conceived within the ECOOP project as a technology demonstrator to help validate such model predictions by combining the multiple datasets used in decision support for the monitoring of the ecological state of the ocean, including the early warning and prediction of potentially harmful algal blooms in the North Sea. It was then extended to demonstrate the pan-European potential of identifying and forecasting the risk of algal blooms, offering the potential to reduce costs associated with intense monitoring programmes which cannot otherwise have human resources on permanent standby or afford to deploy specialist instrumentation 24 h a days, 7 days a week, 365 days a year.

5 Technical discussion

The comparison of in situ and model data from disparate sources is a common problem in the Earth and environmental sciences. This project has employed a number of tools and standards to address this problem. It is important to develop tools such as this in a manner consistent with international standards such as INSPIRE, and those of the OGC in order to facilitate greater interoperability and reusability. Neither the WFS standard, nor the CSML data format, are widely used in ocean science and there is little previous work in this area to guide the development of the portal described here. We therefore discuss WFS and CSML in this section, and compare them with other options for the serving of in situ data.

There is no support in version 1 of the WFS standard for subsetting a feature, and hence our method of returning a specific portion of a CSML PointSeriesFeature had to be achieved through serving features from the WFS which were single points in space and time, and then amalgamating these into the required portions of timeseries in the form of a late-stage transformation to a CSML PointSeries feature. This is not a very efficient method, but was chosen to allow the re-use of the GeoServer software with only relatively minor modifications.

Work is ongoing to increase the efficiency of serving data from the WFS by using a relational database to store the in situ data and by upgrading to the latest version of GeoServer. In addition, the final step in returning the in situ data to the user – conversion of the basic GML Point Features from the WFS into a CSML PointSeries Feature – is a known bottleneck and may be omitted in future versions of the WFS. This will result in the quicker return of data to the user, whilst still maintaining standards-compliance. In this scenario there is a trade-off between increased efficiency and the loss of the more application domain tailored CSML PointSeries Features.

In contrast to WFS, the Sensor Observation Service, SOS, has explicit support for the time dimension, and allows for the request of observations from a specific instant, multiple instances or periods of time and we would now consider SOS to be a good alternative standard for serving in situ timeseries data. (SOS was not published as a standard at the time the work described in this paper was started.) In addition to the support for the time dimension, SOS also has explicit support for the observed property being measured. The OGC Oceans Interoperability Experiment (OGC OIE, http://www.opengeospatial.org/projects/initiatives/oceansie) recommended the use of SOS over WFS for in situ marine data citing the above benefits, as well as others such as increased potential for interoperability and schema and functional maintenance.

The recently published OGC Web Feature Service 2.0 Interface Standard (http://portal.opengeospatial.org/files/?artifact_id=39967) is distinct from the WFS version 1 in a number of ways including richer queries and support for splitting features. This new iteration of the WFS standard may thus prove to be a strong contender for the type of work described here, and it will be interesting to monitor take-up in future projects.

The method of data encoding is somewhat orthogonal to that of the standard for serving the data, and in the present study the use of CSML was successful in meeting the needs of the project. It has the advantage of being semantically precise and tailored for the climate sciences. However, there are not yet many clients capable of correctly parsing CSML, and depending on the nature of the problem the users and their client software, there remains a place for looser formats such as CSV (Comma Separated Value) as recommended in the SeaDataNet (www.seadatanet.org) project. Note that formats such as CSV are semantically weaker than CSML, for example, there is no clean separation between the "domain" and "range" (i.e. the independent and dependent variables). It is a more free-form format that relies to some extent on human interpretation, although an advantage of CSV is that it is more easily ingested into common tools such as spreadsheets. CSML is more precise but, as an XML format, is relatively verbose and hence inefficient to parse in browsers. As a hierarchical format, it is harder to ingest into spreadsheets than column-based formats such as CSV.

An alternative format is ObsJSON (http://code.google.com/p/xenia/wiki/ObsJSON) which is more compact and efficient format for communication between the web server and the browser. In practice however web portal developers have limited control over the service types and data formats used by data providers, and thus must accommodate them as effectively as possible. We anticipate that more data will be made available in ObsJSON format through the IOOS initiative (http://www.ioos.gov/).

Finally, we note that the use of the WMS standard for comparing disparate data can provide a challenge in terms of the choice of colour scales when gridded data are coming from different providers. In the present work, all the model data accessible in the portal are viewed by the ncWMS software, which enables a choice of colour scales. In a scenario where data were also coming from other WMS implementations, careful consideration would have to be given to the choice of colour scale. One possible option in this case is to consider the use of Styled Layer Descriptors (http://www.opengeospatial.org/standards/sld) – another OGC standard.

In this paper we have demonstrated a working multiple data provider system delivered through a single web portal displaying real time model and in situ marine data from 20 modelling groups across Europe and from 45 different in situ observation monitoring stations in 24 different countries. The system has used OpenSource software and standards compliant methods wherever possible. Several applications requiring multi-data input have been given as examples and we believe this kind of service, built on the back of standards based data serving, will become critical for monitoring the marine and wider environment and environmental change on a national and international basis into the future.

Acknowledgements. This project was supported through the EU ECOOP project and the BERR Public Sector Research Exploitation Fund: Third Round Capacity Building Funding for the National Centre for Ocean Forecasting. We would like to thank Patrick Gorringe for his strong support and help to use the SEPRISE dataset, CEFAS for their cooperation in making the SmartBuoy data available for comparison, and the Plymouth Marine Laboratory for making Met Office model output available including biological parameters. We would also like to thank the two anonymous reviewers for their helpful feedback.

Edited by: G. Korotaev

References

Barciela, R., Mahdon, R., Miller, P., Orrell, R., and Shutler, J.: AlgaRisk 08. A pre-operational tool for identifying and predicting the movement of nuisance algal blooms, Innovation for efficiency science programme, Environment Agency, 32 pp., SC070082/SR1, 2009.

Best, B. D., Halpin, P. N., Fujioka, E., Read, A. J., Qian, S. S., Hazen, L. J., and Schick, R. S.: Geospatial web services within a scientific workflow: Predicting marine mammal habitats in a dynamic environment, Ecol. Inform., 2(3), 210–223, 2007.

Blower, J. D., Blanc, F., Clancy, R., Cornillon, P., Donlon, C., Hacker, P., Haines, K., Hankin, S. C., Loubrieu, T., Pouliquen, S., Price, M., Pugh, T., and Srinavasan, A.: Serving GODAE data and products to the ocean community, Oceanography, 22(3), 70–79, 2009a.

Blower, J. D., Haines, K., Santokhee, A., and Liu, C.: Godiva2: Interactive visualization of environmental data on the web, Philos. T. Roy. Soc. A., 367, 1035–1039, 2009b.

Brasseur, P., Gruber, N., Barciela, R., Brander, K., Doron, M., El Moussaoui, A., Hobday, A. J., Huret, M., Kremeur, A., Lehodey, P., Matear, R., Moulin, C., Murtugudde, R., Senina, I., and Svendsen, E.: Integrating Biogeochemistry and Ecology Into Ocean Data Assimilation Systems, Oceanography, 22(3), 206–215, 2009.

de La Beaujardiere, J.: Serving ocean model data on the cloud, OCEANS 2009, MTS/IEEE Biloxi – Marine Technology for Our Future: Global and Local Challenges, 1–10, 2009.

Gibin, M., Singleton, A., Milton, R., Mateos, P., and Longley, P.: An exploratory cartographic visualization of London through the Google Maps API, Applied Spatial Analysis and Policy, 1(2), 85–97, 2008.

Guney, C., Duman, M., Uylu, K., Avci, O., and Celik, R. N.: Multimedia supported GIS in the internet (Case study: two Ottoman fortresses and a cemetery on the Dardanelles), CIPA 2003 XIXth International Symposium, 30 September–4 October 2003, Antalya, Turkey, 2003.

Huang, B.: Web-based dynamic and interactive environmental visualization, Comput. Environ. Urban, 27, 623 pp., 2003.

Kallos, G.: The regional weather forecasting system SKIRON: an overview. Proceedings of the symposium on regional weather prediction on parallel computer environments, University of Athens, 1997.

Korres, G. and Lascaratos, A.: A one-way nested eddy resolving model of the Aegean and Levantine basins: implementation and climatological runs, Ann. Geophys., 21, 205–220, doi:10.5194/angeo-21-205-2003, 2003.

Mahdon, R., Barciela, R., Edwards, K., Miller, P., Shutler, J., Roast, S., Jonas, P., Murdoch, N., and Wither, A.: Advances in Operational Ecosystem Modelling and the Prediction of Nuisance Algal Blooms at the UK Met Office, submitted the ICES Conference Proceedings, 2010.

Peterson, M.: International Perspectives on Maps and the Internet: An Introduction, Geoinformation and Cartography Part A, 3–10, 2008.

Rahim, S. T., Zheng, K., Turay, S., and Pan, Y.: Capabilities of multimedia GIS, Chinese Geogr. Sci., 9(2), 159–165. 1999.

Siddorn, J. R., Allen, J. I., Blackford, J. C., Gilbert, F. J., Holt, J. T., Holt, M. W., Osborne, J. P., Proctor, R., and Mills, D. K.: Modelling the hydrodynamics and ecosystem of the North-West European continental shelf for operational oceanography, J. Marine Syst., 65, 417–429. 2007.

Wei, Y., Santhana-Vannan, S., and Cook, R.: Discover, visualize, and deliver geospatial data through OGC standards-based WebGIS system, 17th International conference on geoinformatics, 1–16, 2009.

Weiss, R. F.: The solubility of nitrogen, oxygen and argon in water and seawater, Deep-Sea Res., 17, 721–735, 1970.

Woolf, A., Lawrence, B., Lowry, R., Kleese van Dam, K., Cramer, R., Gutierrez, M., Kondapalli, S., Latham, S., Lowe, D., O'Neill, K., and Stephens, A.: Data integration with the Climate Science Modelling Language, Adv. Geosci., 8, 83–90, doi:10.5194/adgeo-8-83-2006, 2006.

Appendix III: The FluidEarth 2 implementation of OpenMI 2.0

Adapted from J. Hydro volume 16, issue number 4, pages 890-906, with permission from the copyright holders, IWA Publishing.

Journal of Hydroinformatics

The FluidEarth 2 implementation of OpenMI 2.0

Quillon Harpham, Paul Cleverley and David Kelly

ABSTRACT

Following the release of the OpenMI 2.0 standard for model coupling with reference object classes (interfaces) in C# and Java, a set of tools including a Software Development Kit (SDK) and Graphical User Interface (GUI) is expected to accompany it. These are necessary to enable numerical model developers to easily adapt their models to become OpenMI compliant and to allow modellers to easily assemble and run compositions of them. FluidEarth 2 is an HR Wallingford initiative providing these open source tools for the .net 4.0 Framework together with training, community support and sample models. They are the only such open source tools available so in this sense they act as the reference SDK and GUI for OpenMI 2.0 with .net. The purpose of this paper is to outline these and demonstrate a set of examples. A series of components were successfully constructed and compositions built. These include training models designed to demonstrate different aspects of model coupling, moving to industry strength model codes simulating dam-break bathymetry updates. The FluidEarth 2 tools have been designed to be cross-platform and have been tested under Windows and Linux (using Mono). Usage is successfully demonstrated, providing an environment for integrated modelling with OpenMI 2.0.

Key words | environment, hydraulic, integrated, interface, modelling, OpenMI

Quillon Harpham (corresponding author)
Paul Cleverley
David Kelly
HR Wallingford,
Howbery Park,
Wallingford,
Oxfordshire,
OX10 8BA,
United Kingdom
E-mail: *q.harpham@hrwallingford.co.uk*

ACRONYMS AND ABBREVIATIONS

2DH	2-Dimensional Horizontal
BIA	Bilinear Interpolation Adaptor
GIS	Geographic Information System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IPC	Interprocess Communication
NLSW	Non-linear Shallow Water
OA	OpenMI Association
OpenMI	Open Modelling Interface
SDK	Software Development Kit
TCP	Transmission Control Protocol
UML	Unified Modelling Language

INTRODUCTION

OpenMI (Open Modelling Interface) is a standard for coupling numerical models with data exchanged between modelling components at run time. Following the successful version 1.4, version 2.0 of OpenMI was released in December 2010. The standard consists of a set of object interfaces, which can be represented by a set of Unified Modelling Language (UML) diagrams. The governing OpenMI Association (OA) also produces two sets of reference classes in C# and Java. Developers may construct their own classes from the UML diagrams, but use of the reference classes is encouraged. Following release of the base OpenMI standard, the OA also pledges to release two tools to allow the standard to be used more easily: a Software Development Kit (SDK) to assist construction of OpenMI compliant components and a Graphical User Interface (GUI) to assist the assembly and execution of compositions of OpenMI compliant components. Version 1.4 of OpenMI was accompanied by an associated SDK and GUI for C#, similar applications were also required for version 2.0 of OpenMI.

doi: 10.2166/hydro.2013.190

FluidEarth 2 is an HR Wallingford initiative providing an SDK and GUI for OpenMI 2.0 for the Microsoft .net Framework with C#. They use the C# OpenMI reference classes and are the only such open source tools available so in this sense they act as the reference SDK and GUI for OpenMI 2.0 with the .net Framework. Accompanying these tools is an extensive training website and a set of example models together with a portal for community interaction. The purpose of this paper is to outline the FluidEarth 2 SDK and GUI and, with reference to the training material, document a set of examples introducing the reader to using OpenMI 2.0 with FluidEarth 2. Although not restricted to hydrology and environmental modelling, FluidEarth 2 is driven from these disciplines and the examples listed all derive from this subject area.

Motivation

It is becoming increasingly recognised that many modern environmental questions cannot be answered by modelling physical, chemical or biological parameters in isolation. Environmental systems couple many natural processes and simulating them accurately demands modelling them in a similar fashion, that is, taking into account their interactions (Moore 2010). The environment is an interconnected system and what happens in one location can have repercussions both far away (Meiburg 2008) or at a single location if multiple, dependent parameters interact. This being the case, the only way to successfully answer these questions is to employ integrative approaches, often spanning disciplines, to complement the traditional single discipline methods (Anastas 2010).

In recognising that the actions of these complex environmental systems often produce dramatic and severe consequences, it is clear that one single numerical model cannot be sufficient to represent all of the details needed for decision making and planning (Voinov 2010). Incorporating all necessary environmental processes in a single model eventually becomes unwieldy, difficult to develop and support and ultimately becomes vulnerable by its dependence on certain key individuals. One solution to this is to simulate complex systems by integrating multiple, smaller models that collectively simulate the larger problem in question. That is, to build an integrated composition of previously independent numerical models and run them together allowing them to influence each other as they proceed through their respective time steps. The authors consider that any such solution should meet four key requirements:

(i) Allowing two-way exchange of results between the independent models in the composition as they proceed through their formulation.
(ii) Each component remaining sufficiently independent so that experts can remain in their disciplines, yet are able to communicate model outputs clearly where necessary at the interfaces between their coupled models.
(iii) Allowing the model interoperability to be undertaken flexibly and in a standardised fashion.
(iv) Enabling easy extensibility of the integrated composition to incorporate new parameters and to exchange similar numerical engines where appropriate.

OpenMI and FluidEarth have been undertaken to meet these challenges.

Background

Developed through considerable cooperation and joint working from leading hydraulic centres across Europe and part funded by the European Commission, OpenMI is a software component interface for the computational core (the engine) of a numerical model (Gregersen *et al.* 2007). Model engines are designed or modified to be 'OpenMI Compliant', thus enabling their inclusion in OpenMI integrated compositions. Previous versions of OpenMI have been used for many purposes including:

- river basin management (Makropoulos *et al.* 2010; Safiolea *et al.* 2011);
- dike seepage under transient boundary conditions (Becker & Schüttrumpf 2011);
- integrating agriculture, groundwater and economic models (Bulatewicz *et al.* 2010);
- water quality modelling (Shrestha *et al.* 2012);
- real time control of hydraulic structures (Becker *et al.* 2012); and
- beach plan-shape modelling (Sutherland *et al.* 2013).

As a response to European Union (EU) Water Framework Directive calls for integrated water management, OpenMI itself was originally developed as a means for coupling existing models which would typically consider the interactions of environmental processes, in particular involving water (Moore *et al.* 2010). It has since been realised to be considerably more flexible, evolving into activities such as decomposing large models into smaller model components. It is now considered an interface standard between software components which can be applied to linking any combination of models, databases and associated tools (Lu & Piasecki 2012; OpenMI Association website 2012a).

OpenMI has been designed to allow two-way exchange of data between compliant components as they run, as explored by Elag & Goodall (2011). This would typically occur between two simultaneously running, timestepping model components which would send and/or receive data at specific timesteps as they proceed through their respective time intervals. In this way, the two model components can both influence the results produced by the other. The linked components may run asynchronously with respect to these timesteps or proceed through together. OpenMI also supports one-way passing of data from a driving component to a second, set up only to receive.

As an upgrade from the previous version 1.4, OpenMI 2.0 was released in December 2010 at a specially convened reception during an EU-US summit in Washington, DC (OpenMI Website 2012b). It incorporated a set of new features, some to build on the base from version 1.4 and others to replace or enhance the standard. These included the following:

- **Base Interfaces and Extensions** – A set of minimum 'base interfaces' for compliance, plus the addition of extensions (including an extension covering time and space dependent components). The essential OpenMI component is no longer forced to be time and space dependent, making the standard considerably more flexible and extensible. This allows different types of components to be incorporated e. g. those which vary in time and not in space; those which vary in space, but not in time or those which vary in both time and space (OpenMI Association Website 2010c).
- **Adaptors** – Taking over from the role of 'Data Operations' in OpenMI 1.4, 'Adapted Outputs' allow multiple, distinct adaptations, separate from the components themselves and the link, to take place. Again, this makes the standard more flexible and allows outputs and adapted outputs to be re-used by more than one OpenMI component (OpenMI Association Website 2010c).

IMPLEMENTATION

HR Wallingford's FluidEarth 2 is an implementation of OpenMI 2.0 consisting of a set of tools which provide an environment for the standard to be used. It uses the C# reference classes (OpenMI SourceForge project 2010). The tools are Open Source (FluidEarth SourceForge project 2012) and FluidEarth 2 also comes with a training website and examples, both ready for use and to act as templates for the user's own components. FluidEarth began as an implementation of OpenMI 1.4 and has been upgraded to FluidEarth 2 to meet the specification of this new OpenMI standard. It meets the OA (OpenMI Association) pledge to accompany each release of OpenMI with two tools:

- An SDK allowing model *developers* to easily make their model engines OpenMI compliant. FluidEarth 2 includes such an SDK to cover this requirement as a follow-up to that provided under OpenMI 1.4.
- A GUI allowing model *users* to build and run compositions of OpenMI compliant components. The FluidEarth 2 GUI, 'Pipistrelle', is a follow-up to the OpenMI 1.4 version of Pipistrelle and the OpenMI 1.4 Configuration Editor.

In addition to these and from its inception, FluidEarth incorporates three other aspects:

- A community of model providers and users.
- A set of websites/repositories: the FluidEarth portal at http://fluidearth.net including document libraries, news, discussion, case studies and community contact details; a model catalogue allowing the listing of model engines and their instances at http://catalogue.fluidearth.net; a source forge repository with open source application code at http://sourceforge.net/projects/fluidearth/ and an extensive training website at http://training.fluidearth.net.
- A library of models available for compositions.

FluidEarth has been in operation since 2008, originally supporting OpenMI version 1.4. OpenMI 2.0 is the current supported standard, although compatibility with version 1.4 has been maintained. The tools and training for OpenMI 1.4 are still available from the FluidEarth portal and Pipistrelle includes a facility to allow OpenMI 1.4 components to be automatically made OpenMI 2.0 compatible.

The philosophy behind the FluidEarth 2 SDK and Pipistrelle follows that of OpenMI version 2.0 itself: openness and flexibility. OpenMI 2.0 includes a base standard and extensions; Pipistrelle gives rich base functionality and also plugins. The tools themselves are designed with a high-degree of usability and are supported by clear and simple training, leading the user into the principles of use of OpenMI 2.0 with FluidEarth 2 with highly straightforward use cases designed to demonstrate the capabilities of the toolset. Templates are provided with detailed explanations allowing users to adapt these simple examples into real use cases.

Compositions can be set-up and loaded into Pipistrelle using the menus, although it is also possible to configure a composition by editing the configuration files (xml) directly and running through the console. The tool runs the compositions using a 'pull' approach. The most common use case involves timestepping components which proceed through time in a series of intervals calculating values at each. One of the components is assigned a 'trigger' which controls the composition. Components then demand data from one another, waiting while the requested component runs timesteps until it can meet the request. The 'adaptors' concept from OpenMI version 2.0 is implemented directly through an additional set of menus, allowing connections between components to apply functions to the data passed along the linkages. These adaptors are independent of both components; a departure from the OpenMI 1.4 concept where this functionality resided within one or other of the components linked.

With usability and interoperability seen as specific goals of the FluidEarth 2 implementation, three features have been built into Pipistrelle and the FluidEarth SDK since its original release in 2012. The first of these is the ability to save a FluidEarth OpenMI 1.4 composition (containing components prepared under the previous version of FluidEarth using OpenMI 1.4) as a FluidEarth 2 composition. This facility is supplied with downloads of Pipistrelle from July 2013 (it existed in previous versions but did not stand up to full testing) and works as long as the 'data operations' aspect of the standard has not been used in these components. Figure 1 gives a screenshot of this facility in Pipistrelle.

The second feature built into versions of Pipistrelle available from the summer of 2013 is the ability to view spatial datasets. This facility is added as a plug in and gives the user a two-dimensional view of the node sets related to the models in the composition. This can be particularly

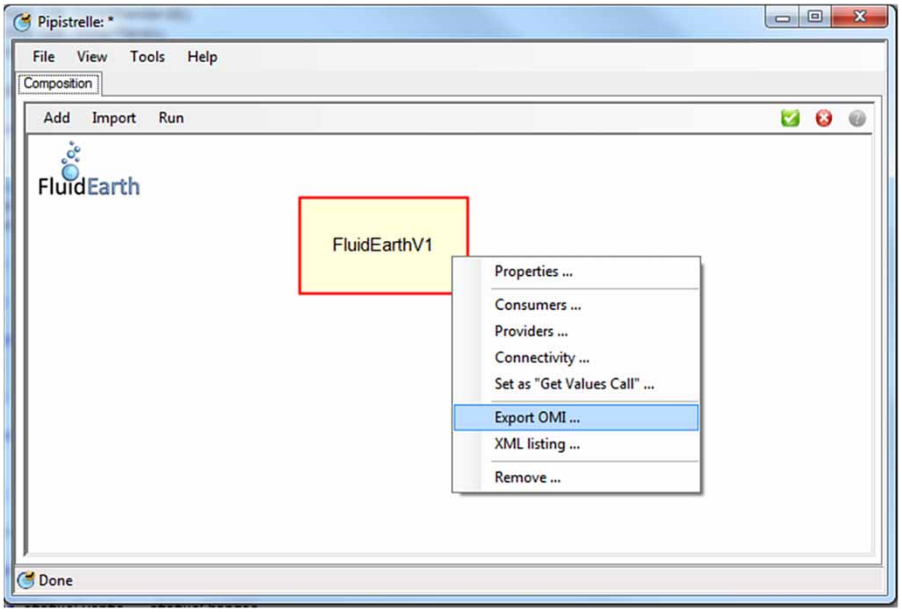


Figure 1 | Using Pipistrelle to save an OpenMI 1.4 component as an OpenMI 2.0 component.

important when different meshes (or even different spatial structures) are required to be connected. Nodes that are required to pass data between one another can be more easily identified and adaptations followed. Figure 2 shows a screenshot of the 'Spatial View' plug-in which is built around the open source DotSpatial geographic information system library (http://dotspatial.codeplex.com/) giving base Geographic Information System (GIS) functions to the user. It shows spatial layers being loaded and viewed.

The third feature is called 'Component Builder', again available from summer 2013. This facility is designed to avoid any FORTRAN programmers having to write code in C#. The C# shell code is built automatically by component builder, based on certain knowledge provided about the FORTRAN module(s) to be wrapped. Figure 3 shows a screenshot of the component builder plug-in at the point in the process where the user is selecting a spatial definition to match that of a FORTRAN model which is undergoing the wrapping process to form an OpenMI 2.0 component.

Native language development

The FluidEarth implementation of OpenMI version 2.0 has been achieved through the use of two development languages: C# and FORTRAN. Visual Basic (.net) is expected to be a corollary of this approach, but the perceived lack of potential components to be implemented in Visual Basic resulted in full and complete testing of Visual Basic counterparts of all the components offered in C# to be omitted in favour of some simple verifications. This can be completed at a later date should demand for components written in Visual Basic be demonstrated. Components written in other languages such as Python and C++ are feasible, but this has not been tested to date.

At run time the FluidEarth Pipistrelle GUI connects components in a composition using .net Framework connectivity to load and execute objects and their methods. Components which are developed in a native compiled language (that is code that is unmanaged by the Common Language Runtime) such as FORTRAN or C, are wrapped in a thin C# (or other .net language) wrapper class which performs the work of communicating with the native code itself. The native code base must implement a defined set of functions (provided by a template) to allow the managed code wrapper to control the running of the native code. These are given in Table 1. More detailed descriptions are available in the FluidEarth Help documentation (FluidEarth SourceForge Project 2012).

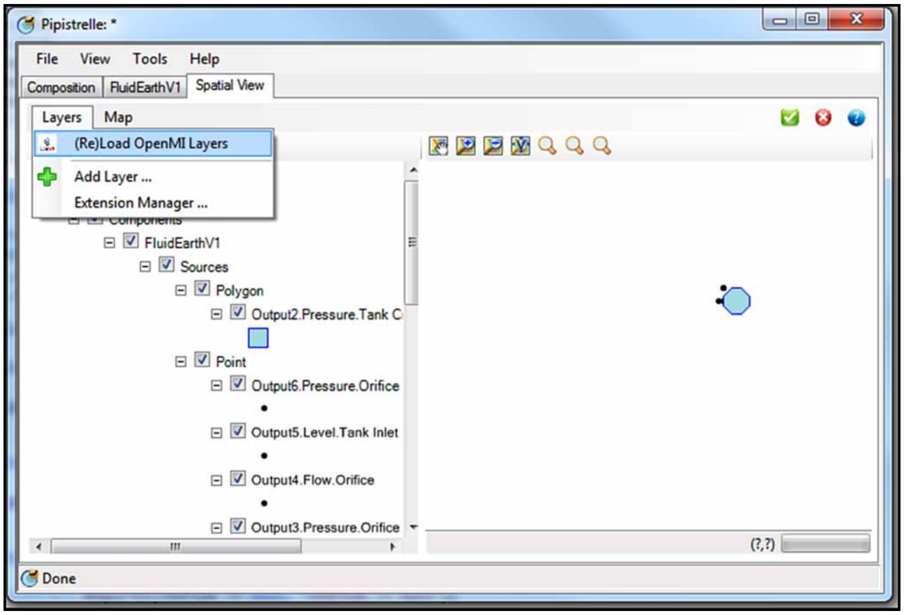
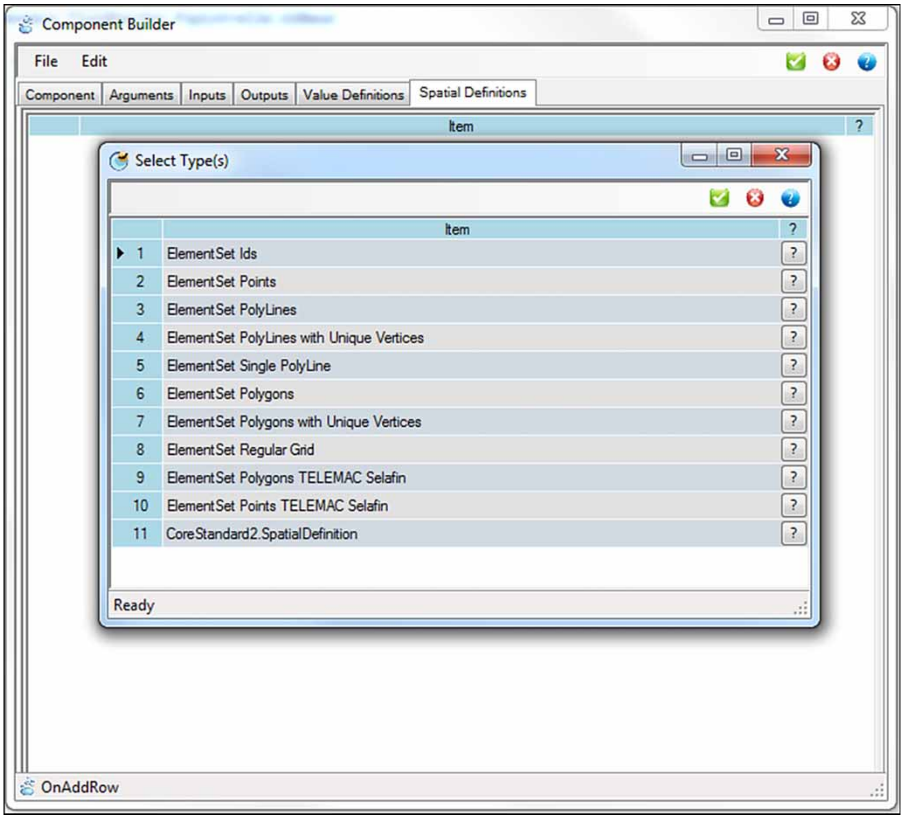


Figure 2 | Pipistrelle Spatial View plug-in screenshot.



	Item	?
1	ElementSet Ids	?
2	ElementSet Points	?
3	ElementSet PolyLines	?
4	ElementSet PolyLines with Unique Vertices	?
5	ElementSet Single PolyLine	?
6	ElementSet Polygons	?
7	ElementSet Polygons with Unique Vertices	?
8	ElementSet Regular Grid	?
9	ElementSet Polygons TELEMAC Selafin	?
10	ElementSet Points TELEMAC Selafin	?
11	CoreStandard2.SpatialDefinition	?

Figure 3 | The Pipistrelle Component Builder plug-in.

Native code is unmanaged. It is not controlled by the Common Language Runtime of .net nor the Java Virtual Machine environment; its memory allocation is handled directly; run time type checking and reference checking are also uncontrolled by .net/Java. As the native code is unmanaged it is essential it is either written as thread-safe (that is, it runs in a manner which guarantees that other executing threads will not be destructively interfered with) or, ideally, run in a completely separate process via the FluidEarth remoting protocols.

Remoting is a .net technology for handling communication between objects across computer and network boundaries. If a component in a composition is to be run outside of the managed Pipistrelle application process (because it is not guaranteed to be thread-safe like a native code engine) then the remoting option of the component will need to be set to something other than 'InProcess'. Once selected a communications technology can be selected from one of IPC (Inter-process communication), AutoIPC (Inter-process communication where the identifiers are managed automatically), TCP (Transmission Control Protocol) and HTTP (Hyper Text Transfer Protocol). This can be configured per component in the Pipistrelle GUI which offers the options for executing a component in Table 2. The approach used does not exclude future development of other protocols, methods of communication and runtime control.

EXAMPLE CASES

The objective sought by the FluidEarth 2 toolkit (SDK and GUI) is to easily enable models (and other valid components) to be made OpenMI compliant and to provide a user-friendly graphical interface to allow users to assemble and run compositions. A set of examples was put together to progressively test the functionality of the toolkit, from a low level of complexity to a level expected by a typical 'real world' example of hydraulic modelling. This technical progression and the

Table 1 | Template functions for native code to implement; Mandatory (*M*), Optional (*O*)

	Function/Subroutine signature
M	function FLUIDEARTH2_ENGINE_PING() Used to establish that the engine has been successfully instantiated.
M	function FLUIDEARTH2_ENGINE_SUCCESSMESSAGE(success_code, message) Returns a message determined by the success_code parameter
M	subroutine FLUIDEARTH2_ENGINE_INITIALISE(args, success_code) Initialises FluidEarth2.Sdk.BaseEngine: Arguments from the supplied text XML and then uses argument helper functions to set specific engine parameters.
M	subroutine FLUIDEARTH2_ENGINE_SETARGUMENT(k, v, success_code) Initialises the argument specified by k with the value v returning the result of the operation in success_code.
M	subroutine FLUIDEARTH2_ENGINE_SETINPUT(engine_variable, element_count, element_value_count, vector_count, success_code) Called for each active IBaseInput and informs this code that the engine variable specified by engine_variable has been activated for this run the so engine must use these values when they arrive; element_count is the number of elements in the corresponding element set; element_value_count is the number of values per element; vector_count is the size of the vector for each value.
M	subroutine FLUIDEARTH2_ENGINE_SETOUTPUT(engine_variable, element_count, element_value_count, vector_count, success_code) Called for each active IBaseOutput and informs the code that the engine variable specified by engine_variable has been activated for this run so engine must use these values when they arrive; element_count is the number of elements in the corresponding element set; element_value_count is the number of values per element; vector_count is the size of the vector for each value.
M	subroutine FLUIDEARTH2_ENGINE_PREPARE(success_code) The place to dynamically allocate memory for arrays and other requirements.
O	subroutine FLUIDEARTH2_ENGINE_SETINT32S(engine_variable, missing_value, values_size, values, success_code) Called for each active IBaseInput before a call to Update(); this is used to set the values of a 32 bit integer type variable prior to the Update();
O	subroutine FLUIDEARTH2_ENGINE_SETDOUBLES(engine_variable, missing_value, values_size, values, success_code) Called for each active IBaseInput before a call to Update(); this is used to set the values of a double precision type variable prior to the Update();
O	subroutine FLUIDEARTH2_ENGINE_SETBOOLS(engine_variable, missing_value, values_size, values, success_code) Called for each active IBaseInput before a call to Update(); this is used to set the values of a Boolean type variable prior to the Update();
M	subroutine FLUIDEARTH2_ENGINE_UPDATE(success_code) Called to perform a calculation and modify the component's time step.
O	subroutine FLUIDEARTH2_ENGINE_GETINT32S(engine_variable, missing_value, values_size, values, success_code) Called for each active IBaseOutput after a call to Update(); this is used to retrieve the values of a 32 bit integer variable immediately after a call to Update().
O	subroutine FLUIDEARTH2_ENGINE_GETDOUBLES(engine_variable, missing_value, values_size, values, success_code) Called for each active IBaseOutput after a call to Update(); this is used to retrieve the values of a double precision variable immediately after a call to Update().
O	subroutine FLUIDEARTH2_ENGINE_GETBOOLS(engine_variable,missing_value,values_size, values,success_code) Called for each active IBaseOutput after a call to Update(); this is used to retrieve the values of a Boolean variable immediately after a call to Update().
M	subroutine FLUIDEARTH2_ENGINE_FINISH(success_code) Called to dispose of any dynamically allocated resources – as may have been allocated in Prepare().
M	function FLUIDEARTH2_ENGINE_GETCURRENTTIME(success_code) Called to return a double precision number representing the current time as known to the component.

testing examples used has been built into a training website (Cleverley 2012), giving a comprehensive introduction to FluidEarth and its underlying concepts. First, a simple, stand-alone model is constructed. It has no geospatial attributes, offers a single parameter as output and receives a single identical parameter as input. Using just this component, it is then possible to construct the very simplest compositions: a single OpenMI 2.0 component running

Table 2 | FluidEarth 2 component execution options

Component remoting option	Description
In Process	Runs the component 'in process' - the usual mechanism for .net managed components. Each instance of a given managed component will have its own memory space but each instance of unmanaged components will share a single memory space so the danger is that each instance of a given unmanaged component might overwrite data from another instance of the same unmanaged component.
IPC Auto	Runs the component in a separate process using inter-process communication protocols whilst automatically assigning port and object identifiers – this will ensure that, in a given composition, instances of the same unmanaged component will run with its own memory space.
IPC	Runs like IPC Auto but requires the user to specify the object and port identifiers (not used in Cleverley (2012)).
TCP	Runs the component via TCP protocol (not used in Cleverley (2012)).
HTTP	Runs the component via the HTTP protocol (not used in Cleverley (2012)).

alone and the natural corollary of another simple composition linking two identical pond instances together – one pond drains into its twin. Further examples then add an adaptor between the two components (in this case to perform a unit translation where one pond drains in centilitres into another which requires input in millilitres), geospatial structures (linear boundaries) and geospatial interpolation (between these linear boundaries, but with differing numbers of nodes). This series of examples is performed using models and adaptors written in both C# and FORTRAN.

When defining suitable examples to demonstrate the development of Open MI 2.0 compliant FluidEarth components the FluidEarth 'engine pattern' was used. This results in a component interface class which is separate from its related engine class. An interface class is a 'class that primarily defines a protocol, but does not provide an implementation. This means they only describe the expected behaviour … ' (Google Web Definitions 2013). This separation allows changes to the engine operation whilst maintaining the interface exposed to the Pipistrelle GUI. The engine class itself is implemented using one of the interfaces given in the FluidEarth 2 SDK *FluidEarth2.Sdk. Interfaces* – either *IEngine* or *IEngineTime*.

The term 'Engine' refers to the executable code that the user wishes to 'wrap' or develop to make it 'OpenMI compliant'. The Engine Pattern simply means Engine code that can be used via the interfaces FluidEarth2.Sdk.Interfaces.IEngine or FluidEarth2.Sdk.Interfaces.IEngineTime. OpenMI does not require the use of the Engine Pattern to make code compliant. It is a simplification which, if possible, then allows SDK providers to provide libraries of code to simplify the compliancy task. Hence, if a user's code can be reformulated to use one of these engine interfaces it can be easily implemented using the FluidEarth2_SDK.

Time stepping components, that is models or data providers which offer data over a timeline divided into timesteps each of which holds data values, have been used throughout since most current requirements fall into this category. In this case, the results of the engine change over time and the component exposes a time 'horizon' as an argument. Hence the selected component uses the *FluidEarth2.Sdk. Interfaces.IEngineTime* interface.

The simple pond

This Simple Pond example, taken from the training material (Cleverley 2012), illustrates a simple form of OpenMI component. It allows the user to grasp some basic aspects of using OpenMI with FluidEarth 2 as well as offering some template code. In addition to the necessary default OpenMI required arguments, this component (a Pond) has the arguments **capacity**, **currentlevel** and **flow**. These are denoted in Figure 4, below.

flow indicates the amount of water which overflows out of the component each time step; **currentlevel** gives the current water level of the Pond at any given time (at the

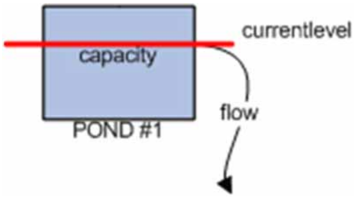


Figure 4 | Simple Pond OpenMI component schematic view.

beginning, during the run and at the end of the run); **capacity** gives the amount of water contained in the component which must be exceeded before any overflows. In addition, the component has one input: **inFlow** and one output: **outFlow**, both variables whose value may change at run time. The base functionality of the FluidEarth SDK and Pipistrelle has been demonstrated through the most simple composition with this model, stand alone, in both C# and FORTRAN. In addition, another simple composition is possible by connecting two identical copies of this pond, connecting the inFlow of the first pond into the outFlow of the second.

Adaptors are required when the output of one model cannot be directly connected to the input of another. To demonstrate the basic adaptor definition and function in FluidEarth an adaptor was developed to perform a simple unit transformation against a single input, multiplying it by 10 and giving a single output. This generic function is used in this case to convert centilitres to millilitres. In this way the user building the composition can focus on the details of building an adaptor rather than the function of the adaptation itself.

The next composition, depicted in Figure 5 and again taken from the training material (Cleverley 2012), gives an instance of the Pond component (Pond #1) overflowing at a certain rate when its capacity is exceeded. The adaptor modifies its input from centilitres to millilitres and passes that value onto the second component (Pond #2) as the inFlow input. Pond #2 fills up until it too begins to overflow.

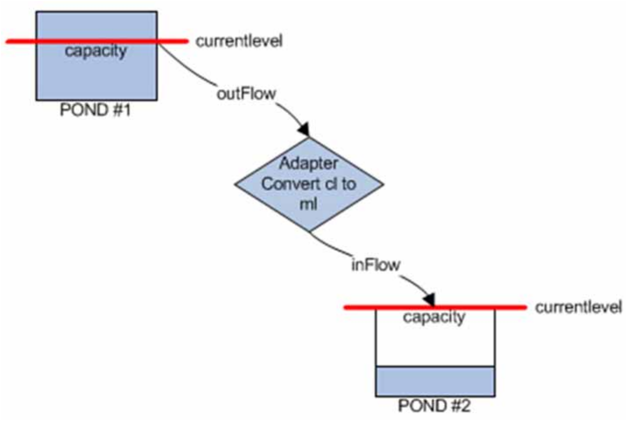


Figure 5 | FluidEarth 2 composition involving a simple adaptation.

The two-dimensional pond

In moving towards a more typical modelling solution, the pond theme is continued, but a geospatial structure is added to the components. The two-dimensional pond example is designed to begin to illustrate usage of spatial structures and provide template code for users. It is also taken from the training material (Cleverley 2012) and presents a straightforward use case. Pond II offers output across arrays at each boundary, evenly spread across each length to represent water transfer across the entire length of each pond edge (see Figure 6).

When two such pond components are joined in a composition the action is similar – fluid will flow from one part of the pond to another as it drains into a second, identical pond component along a boundary. The nodes of the eastern boundary of the first pond match to the nodes on the western boundary of the second pond one-to-one, with values passed directly between the two.

In removing the restriction of the connected boundaries being of the same array dimension an adaptor is required to interpolate between boundaries of different sizes. In Figure 7 the 'ten-node' eastern boundary of Pond #1 needs to be connected to the 'five-node' western boundary of Pond #2.

Without this one-to-one mapping of outputs to inputs, the 2d pond adaptor provides an interpolation to allow values to be passed between components. Of course, any conversion between these two node-sets is possible, the

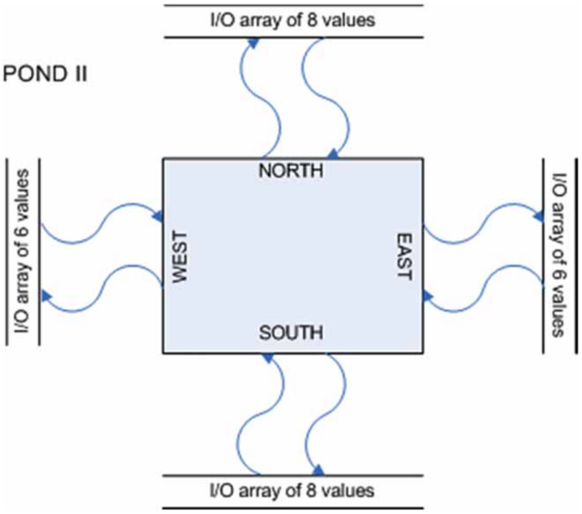


Figure 6 | The two-dimensional pond component with water transfer across pond edges.

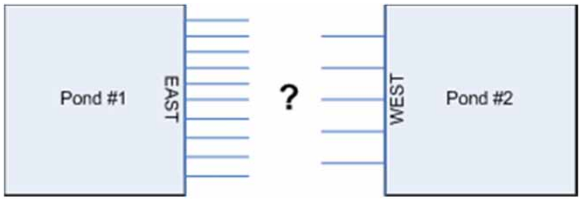


Figure 7 | Connecting ponds of different boundary dimensions.

adaptor concept is given as a placeholder for the implementation of any chosen algorithm.

In Cleverley (2012), simple examples have been chosen, since the emphasis is on learning how to build and incorporate an adaptor and not on working with complex data aggregation algorithms. This allows the student to learn how to build an adaptor which handles two-dimensional data rather than showing the behaviour of a more realistic example. A two-dimensional composition connects the East boundary Pond #1 to the West boundary of Pond #2 using a two-dimensional aware adaptor to interpolate the values where necessary.

Coupling two timestepping models with a simple dam-break test case

A more involved coupling scenario is now considered. The composition comprises two timestepping models coupled together in Pipistrelle via an adapter. The model OTT2D is a 2DH (2 dimensional horizontal) NLSW (non-linear shallow water) solver. The OTT2D solver employs a collocated (cell-centred) finite volume scheme. A detailed description of the OTT2D solver is beyond the scope of this paper and, as such, a full description of the OTT2D model can be found in Hubbard & Dodd (2002). The Exner solver solves the sediment continuity equation employing the simple, first-order accurate, node based 'upstream' finite-difference scheme of Perdreau & Cunge (1971). The sediment continuity (Exner) equation relates bathymetric evolution to sediment flux divergence via a sediment transport formula and can be written in vector notation according to Equation (1),

$$\frac{\partial B}{\partial t} = -\nabla \cdot \vec{q} \qquad (1)$$

where $B = B(x, y, t)$ is the bed height relative to a datum level and $\vec{q} = \vec{q}(\vec{u}, h)$ is the vector of sediment fluxes.

These models are coupled together in order to allow a subsection of the OTT2D model bathymetry to evolve based on the hydrodynamic conditions. The OTT2D mesh is a different spatial representation than that of Exner, in fact, in the example below the Exner mesh is nested within the OTT2D mesh (see Figure 8). An adaptor is therefore required to allow the models to pass data. A simple spatial, BIA (Bilinear Interpolation Adaptor) is used. It is effectively a linear interpolation based on a triangulation of the input point set. The point set comprising the OTT2D mesh is first Delaunay triangulated and the bounding triangle (i.e. the triangle that encloses the interpolation point) for each interpolation point on the Exner sub-mesh is identified. A weighted average based on splitting the bounding triangle into three sub-triangles with the interpolation point as a common vertex is used. Areas of the bounding triangle and three sub-triangles are computed using the general formula given by Braden (1986). Values at each vertex of the original bounding triangle are then weighted according to the relative weights of each the three-sub triangles to the bounding triangle to give the value at the interpolation point. The two models that comprise the composition, OTT2D and Exner, are run on distinct meshes; the OTT2D mesh is cell centred whilst the Exner mesh is node centred so the meshes are not coincident. The Exner mesh comprises a sub-domain of the larger OTT2D mesh as illustrated in Figure 8, which clearly shows the node centred finite difference Exner mesh (depicted in white) nested within the cell centred finite volume mesh of OTT2D (depicted in grey), with the inset section illustrating that

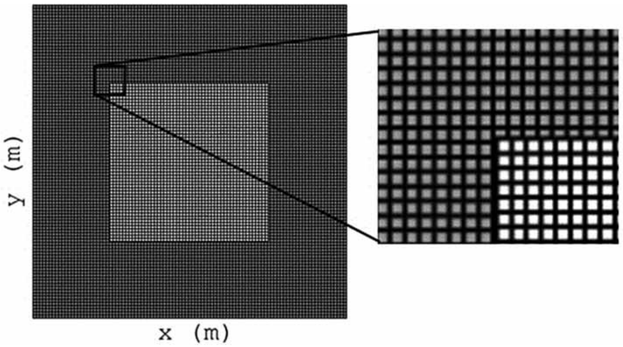


Figure 8 | The experimented configuration with the Exner Mesh within the OTT2D mesh.

the meshes are not coincident. The OTT2D solver solves the 2DH NLSW equations to give the time evolution of the water depth, h, and depth-averaged velocity components u and v. Velocities output from OTT2D are interpolated at each point on the Exner mesh using the adapter described above and the associated sediment fluxes are computed according to the Grass (1981) formula in Equation (2),

$$\vec{q} = A(u|\vec{u}|^2, v|\vec{u}|^2) \qquad (2)$$

where A is a dimensional transport constant (dimensions $s^2\ m^{-1}$) whose value can be related to sediment density and grain size (d_{50}) (see e.g. Hudson 2001).

The Grass formula is used to close the Exner equation in this illustrative example as it is the simplest total load sediment transport formula available. Test specific details on the mesh dimensions are provided in the test case section.

A test case is now considered which moves towards that typical of 'real world' usage of these models. It consists of a simple wet-wet dam-break in a closed box and is used purely to illustrate the coupling of two-different timestepping models via an adaptor. The initial conditions for the simulation are shown in Figure 9 and comprise still water of depth 1 m with a 1 m high 20 m × 20 m block, or reservoir, of water centred at $x = 55$ m, $y = 55$ m. The OTT2D mesh has a uniform mesh spacing of $\Delta x = 1$ m and $\Delta y = 1$ m.

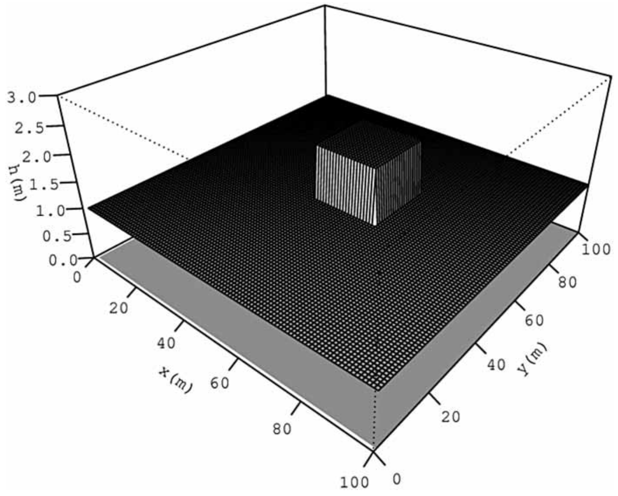


Figure 9 | Initial dam-break test case conditions.

The Exner mesh has its origin at $x = 25.25$ m, $y = 25.25$ m and also uses a uniform mesh spacing of $\Delta x = 1$ m and $\Delta y = 1$ m (see Figure 9). All of the water is initially at rest and at time $t = 0$ the dam 'walls' are assumed to vanish instantaneously creating a shock wave, or bore, that propagates outwards towards the domain boundaries.

We note here that the coupling is one-way with OTT2D passing data to the Exner solver; the water movement deforms the bathymetry but changes in the bathymetry are not fed back to OTT2D. This limitation has been applied due to modelling complexities with this example: when running a model that updates the bed at a different timestep to the flow, instabilities are difficult to avoid and water depth must be corrected to account for bed change. Addressing these issues is beyond the scope of this simple example. An output of the Exner solver is the total bed evolution since the beginning of the simulation *E(i, j)* which is computed according to Equation (3),

$$E(i, j) = \int_0^T \Delta B(i, j)\ dt\ \forall\ i, j \qquad (3)$$

where T is the frame time for the simulation, and *i, j* are the x and y indices, respectively, of the finite difference mesh employed by the Exner solver.

Figures 10–12 show the results of the simulation paused after 4 s of simulation time; these include the water surface, velocity vectors and bed evolution.

Figure 11 shows the outputs of OTT2D as velocity vectors which are passed to Exner through the BIA adaptor at the same timestep.

Time variant two-way data exchange

We now consider a two-way exchange of data between two OpenMI components in a single composition as given in the training material (Cleverley 2012). This common requirement of OpenMI compositions allows two models to influence each other as they run. Components pass data to each

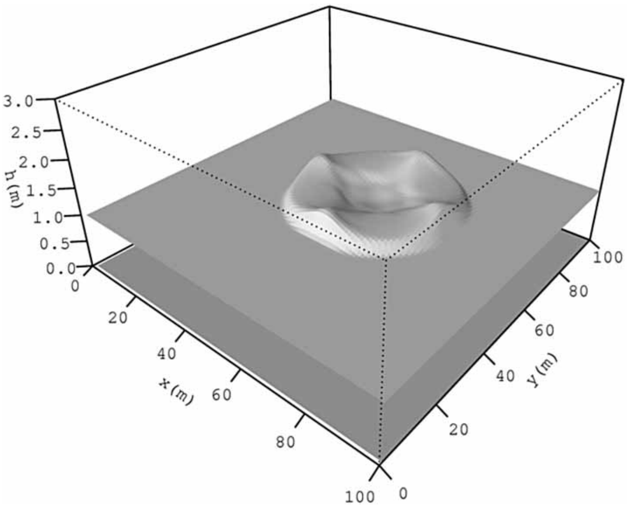


Figure 10 | A snapshot of the water surface and bathymetry (bottom) at $t = 4$ s for the dam-break in a box test problem.

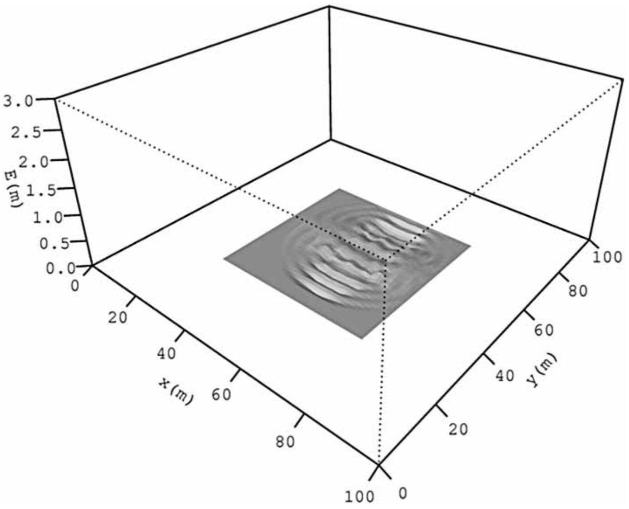


Figure 12 | The total bed evolution at $t = 4$ s computed in the subdomain of the main mesh that is used by the Exner solver.

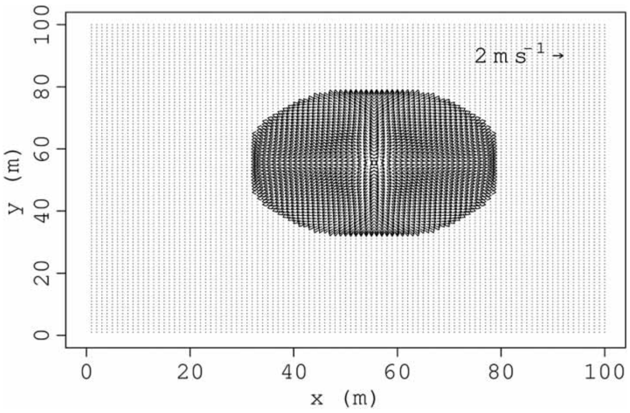


Figure 11 | Velocity vectors at $t = 4$ s as output from the OTT2D solver to the BIA adapter.

other on demand as the composition runs, with each model advancing its internal time. Component A requests data from Component B which runs through sufficient internal timesteps until it can fulfil Component A's request. Similarly Component B may reach a point where it needs to request data from Component A. Component A then runs through sufficient timesteps until it, in turn, can fulfil Component B's request. One component will be the prime driver of this composition (connected to the run trigger) and its completion will signal the completion of the composition itself.

Such a bi-directional exchange of data between components may result in deadlock: Component A is waiting for Component B to fulfil its request for data, but Component B cannot do so until it receives data from Component A. Neither component can proceed and the composition fails to complete successfully. Pipistrelle provides a solution to prevent such deadlock situations occurring: if a component is asked for information that it cannot provide by computation (for example because it would be relying on data supplied from the requesting component) then the component is forced to provide a value, even if it has to approximate.

Morita & Yen (2002) is an example of a model coupling where the value for the previous timestep is used. Pipistrelle, however, is also designed to cover situations where the timestep values of the two models may differ considerably. If the default within Pipistrelle is to supply the previously computed value and a coarse timestep is being used in one component and a finer timestep in another component, then the supplied value may be considerably out-of-date. Accordingly, Pipistrelle uses an extrapolation from previously computed values – a polynomial interpolation based on a defined number of previously calculated results. The request-reply mechanism in use is described in the OpenMI Standard 2 Specification document (OpenMI Association Website 2010d).

By way of example, consider two reservoirs of liquid, A and B. They are connected to each other by two independent channels. One channel only allows water to be pumped from reservoir A to reservoir B and the other channel allows only the reverse, from B to A.

We wish to apply the following rules to the system:

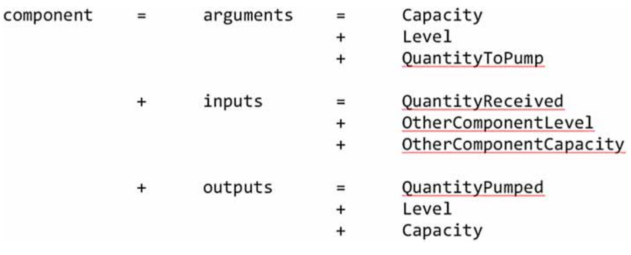
- At a given time, if reservoir A contains more liquid than B then A will pump a certain quantity of liquid (QAB) to B. The quantity pumped (QAB) will be calculated so that, when added to the current level of B, it will not exceed the capacity of B, nor exceed an arbitrary maximum value, nor allow the level of A to drop below zero.
- Equally, if B contains more liquid than A then B will pump a quantity of liquid (QBA) to A. Again, this amount will be calculated so that the level of A won't then exceed its capacity, nor an arbitrary maximum value, nor allow the level of B to drop below zero.

As such:

- No component should ever be allowed to overflow.
- No component should ever pump more than the minimum of its current level and an arbitrary maximum.

Each component will require variables representing its current level and the amount it can pump. Each component will also require an 'input exchange item' representing the quantity of water received and an 'output exchange item' representing the quantity of water pumped. In the composition quantity of water received (A) is linked to quantity of water pumped (B) and quantity of water received (B) is linked to quantity of water pumped (A). So the composition is set up to pass water both ways between the components. Furthermore, as each component will need information from the other component before calculating the amount it is about to pump, each component will require 'output exchange items' exposing its own current level and capacity, and input exchange items indicating the current level of the other component and its capacity.

As illustrated in Figure 13, each component will comprise arguments (Capacity, Level and QuantityToPump), inputs (QuantityReceived, OtherComponentLevel and OtherComponentCapacity) and outputs (QuantityPumped, Level and Capacity).



```
component   =   arguments   =   Capacity
                            +   Level
                            +   QuantityToPump

            +   inputs      =   QuantityReceived
                            +   OtherComponentLevel
                            +   OtherComponentCapacity

            +   outputs     =   QuantityPumped
                            +   Level
                            +   Capacity
```

Figure 13 | Reservoir OpenMI Component arguments, inputs and outputs.

We follow the pull driven approach favoured by the FluidEarth Pipistrelle GUI and so reservoir B will request liquid from reservoir A and reservoir A will request liquid from reservoir B. Each request for data will advance the time step for each component. Figure 14 shows a screenshot of Pipistrelle with this composition loaded.

The initial conditions represent the composition in an unbalanced state. Reservoir A begins with a level of 59 and B with 40. B then requests liquid from A and in this manner the composition progresses from the unbalanced state, where A has more liquid than B, to a state with a degree of equilibrium as seen in Table 3. Units are arbitrary in this notional example but consistent across the composition. Note, also, that it takes time for water to proceed from one reservoir to another.

Figure 15 gives the water levels of both reservoirs as the composition runs. Figure 16 shows the water pumped from each reservoir. Instability occurs at the equilibrium point causing the composition to oscillate and neither reservoir is able to find a stable level.

Time invariant two-way data exchange

The time variant two-way data exchange described above represents a typical two-way OpenMI 2.0 composition; one for which Pipistrelle was originally conceived and has been designed to address. However, it is also possible for two components to require exchanging data with each

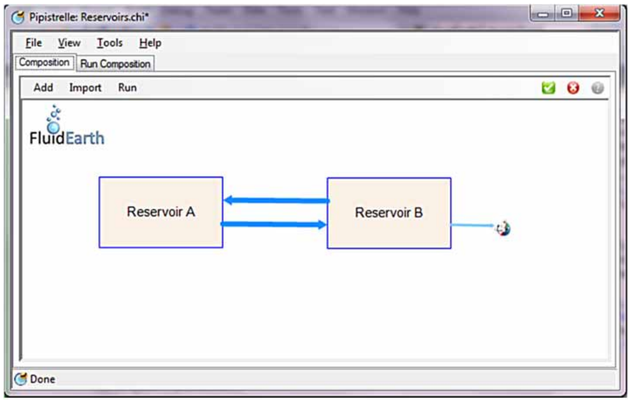


Figure 14 | A basic two-way exchange composition.

Table 3 | Two-way reservoir composition results

	Reservoir A			**Reservoir B**		
Current Time	**Level**	**Received**	**Pumped**	**Level**	**Received**	**Pumped**
2013-05-31 23:00:00Z	59	0	1	40	0	0
2013-05-31 23:05:00Z	58	0	1	41	1	0
2013-05-31 23:10:00Z	57	0	1	42	1	0
2013-05-31 23:15:00Z	56	0	1	43	1	0
2013-05-31 23:20:00Z	55	0	1	44	1	0
2013-05-31 23:25:00Z	54	0	1	45	1	0
2013-05-31 23:30:00Z	53	0	1	46	1	0
2013-05-31 23:35:00Z	52	0	1	47	1	0
2013-05-31 23:40:00Z	51	0	1	48	1	0
2013-05-31 23:45:00Z	50	0	1	49	1	0
2013-05-31 23:50:00Z	49	0	1	50	1	0
2013-05-31 23:55:00Z	49	0	0	50	1	1
2013-06-01 00:00:00Z	50	1	0	49	0	1
2013-06-01 00:05:00Z	50	1	1	49	0	0
2013-06-01 00:10:00Z	49	0	1	50	1	0
2013-06-01 00:15:00Z	49	0	0	50	1	1
2013-06-01 00:20:00Z	50	1	0	49	0	1
2013-06-01 00:25:00Z	50	1	1	49	0	0
2013-06-01 00:30:00Z	49	0	1	50	1	0
2013-06-01 00:35:00Z	49	0	0	50	1	1
2013-06-01 00:40:00Z	50	1	0	49	0	1
2013-06-01 00:45:00Z	50	1	1	49	0	0
2013-06-01 00:50:00Z	49	0	1	50	1	0
2013-06-01 00:55:00Z	49	0	0	50	1	1
2013-06-01 01:00:00Z	50	1	0	49	0	1
2013-06-01 01:05:00Z	50	1	1	49	0	0

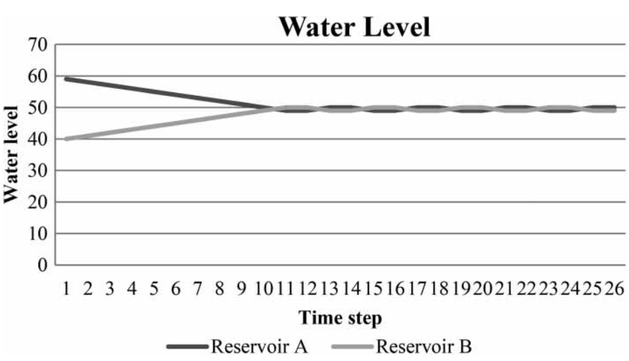


Figure 15 | Two-way reservoir composition water levels.

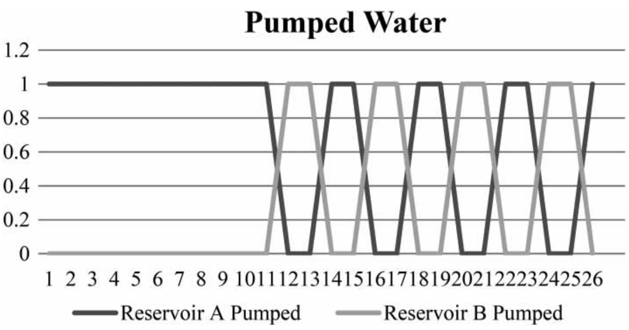


Figure 16 | Two-way reservoir composition pumped water.

other before completing an individual composition timestep. Such an example could be considered a 'time invariant' two-way data exchange. So the bi-directional exchange of information needs to take place within a single timestep. This could be required in compositions where components need to assess the status of other components without advancing their computation. It is possible that a component will wish to gather information about the state of other components before making decisions about its own computation. A time invariant version of the exchange items can be used to ensure that certain exchanges won't progress the clock of the component from which the information is requested.

Consider, as an example the simple FluidEarth Pond model, refactored as 'ConditionalPond'. ConditionalPond only allows water to flow from itself to a downstream component if the current level of the downstream component plus the flow it would receive does not exceed its own capacity. This condition must be determined in such a way that the downstream component's time remains unaffected. This requires developing a TimeInvariant set of ValueSetConverters which allow input and output items to be exchanged without influencing the component's time step.

We connect two independent instances of ConditionalPond together in a two-way connection so that ConditionalPond is type of both upstream and downstream components. Then the ConditionalPond class must implement the appropriate behaviour for the condition where it may be both up and downstream of other similar components. Hence it has both input and output exchange items corresponding to Capacity, CurrentLevel and Flow. Naming our upstream component 'Upstream' and our downstream component 'Downstream' we can represent the model composition as in Figure 17.

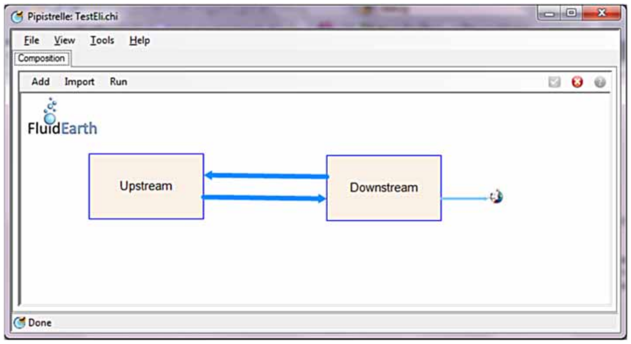


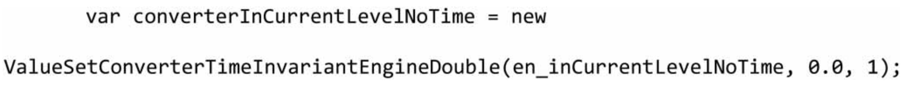
Figure 17 | Two-way connections using the same 'ConditionalPond' class.

The arrow pointing from Upstream to Downstream represents the exchange item Flow. The arrow pointing from Downstream to Upstream represents the exchange items CurrentLevel and Capacity.

When the composition begins, the trigger will request Flow from Downstream. Once Downstream has satisfied the request, its current time will increment. However, in order to satisfy the request from the trigger, Downstream will request Flow from Upstream. Critically, within the same time step, and in order to determine the value of Flow, Upstream will request CurrentLevel and Capacity from Downstream and will modify Flow so that CurrentLevel + Flow <= Capacity.

This request for CurrentLevel and Capacity will not result in the Downstream current time being incremented, hence the nomenclature 'TimeInvariant' applied to the relevant ValueSetConverter (see Figure 18). An IValueSetConverter is a FluidEarth implementation interface that factors out the runtime specific implementation details of a specific OpenMI IBaseExchangeItem interface. Typically this is where the logic for implementing the data transfer resides. Thus a TimeInvariant version results in no increment in the timestep for the component when the exchange item is updated.

This capability allows the building of compositions where component interchange can depend on the state of other components at a given time step and, effectively, makes the state of the components in a given composition available to all other components at run time. This method facilitates such approaches as agent-based modelling where a population of individual agents is modelled by



```
      var converterInCurrentLevelNoTime = new
ValueSetConverterTimeInvariantEngineDouble(en_inCurrentLevelNoTime, 0.0, 1);
```

Figure 18 | TimeInvariant ValueSetConverter code snippet.

each agent being aware of the state of other agents at any given time. The bi-directional passing of data allows them to undertake tests of each other's attributes within timesteps before each makes any decisions about their next exchange.

SUMMARY AND OUTLOOK

FluidEarth 2 is an implementation of OpenMI version 2.0 which seeks openness, flexibility and usability. The successfully executed examples using the FluidEarth 2 SDK and Pipistrelle given above, range from simple one-way compositions to those more typical of real industry or academic requirements. These examples have been built in C# and FORTRAN with VB usage seen as a corollary. The model coupling process is improved and more accessible to less technical users. Using the Pipistrelle GUI, compositions can be built utilising compatible components from different suppliers in a high usability environment. The same GUI is used for executing the compositions offering the same desired level of usability. It has been necessary to apply a detailed level of instruction for building components using the SDK since this is the most involved procedure, tending to be the most esoteric. Usability is higher with the natural C# development language (that of Pipistrelle and the SDK) although FORTRAN compositions are also readily accessible.

The most involved of the compositions, that of coupling together OTT2D and Exner via a simple adapter, from a user's point of view yielded a generally positive experience, indeed a strong improvement from the FluidEarth implementation of OpenMI 1.4. The introduction of adapters as a concept has dramatically improved the usability of implementations of OpenMI 2.0 such as FluidEarth 2 for the linking together of two or more area-type models running on two or more distinct meshes. Responsibility for the adaptation now lies independently from the two model components. These can then remain the same for a variety of compositions with adaptors coded independently and applied as required. For a user who is familiar with the model(s) to be wrapped, the conversion of an existing model to a FluidEarth 2 component is a relatively straightforward task. Both models that were wrapped for this composition had FORTRAN as the native language and the FORTRAN template provided in the FluidEarth 2 download (Harper *et al.* 2012) greatly facilitated the wrapping procedure. It is noted that the simple adapter presented here cannot be expected to conserve quantities that should be conserved. To this end it precedes the generation of a library of adapters that are suitable for adapting between model types based on different numerical algorithms, i.e. from a node-centred finite difference scheme to a cell-centred finite volume scheme. This will require careful consideration. Moreover, care must be taken when using adapters as, if many adaptations are involved, simple linear interpolation could lead to a smoothing of artefacts to such an extent that artefacts that were initially present are lost as the simulation evolves in time. An example of this problem could be specific topographic features in a two-way coupling between OTT2D and Exner.

FluidEarth 2 is targeted at Windows and Linux using .net 4.0 and Mono. Testing has been most extensive on Windows 7 but a medium level of testing has been undertaken to run Pipistrelle on Mono with a composition that has been built on Windows 7. Some cosmetic issues were found within the user interface and, at the time of writing, remain to prevent trouble-free usage on Mono. However, the main elements held up well for the compositions tried. The Mono testing environment was a virtual machine comprising: 1 CPU, 1GB RAM, 40GB HDD. The operating system installed was Linux Ubuntu 12.04.2 LTS Desktop x64. Mono was installed from the default package repository mono-complete version 2.10.8.1-1ubuntu2 monodevelop version 2.8.6.3 + dfsg-2.

The FluidEarth 2 toolkit (Pipistrelle and the FluidEarth SDK) are open source developments available on SourceForge (FluidEarth SourceForge Project 2012). The initial, 2012, versions of the code and the 2013 updates described here were developed for HR Wallingford by Adrian Harper of Innovyze. FluidEarth 2 was co-funded by the European Commission (EC) 7th Framework Programme DRIHM Project, Grant Number 283568.

REFERENCES

Anastas, P. 2010 *Agency Priority*. EPA Office of Research and Development, Washington, DC.

Becker, B. P. & Schüttrumpf, H. 2011 An OpenMI module for the groundwater flow simulation programme Feflow. *Journal of Hydroinformatics* **13** (1), 1–12.

Becker, B. P. J., Schwanenberg, D., Schruff, T. & Hatz, M. 2012 Conjunctive real time control and hydrodynamic modelling in application to rhine river. In: *10th International Conference on Hydroinformatics, HIC 2012*, Hamburg, Germany.

Braden, B. 1986 The surveyor's area formula. *The College Mathematics Journal* **17** (4), 326–337.

Bulatewicz, T., Yang, X., Peterson, J. M., Staggenborg, S., Welch, S. M. & Steward, D. R. 2010 Accessible integration of agriculture, groundwater, and economic models using the Open Modeling Interface (OpenMI): methodology and initial results. *Hydrology and Earth System Sciences* **14**, 521–534.

Cleverley, P. 2012 FluidEarth Training Website, http://eLearning.fluidearth.net (accessed December 2012).

DotSpatial Geographic Information System Library for .net4 website, http://dotspatial.codeplex.com/ (accessed October 2013).

Elag, M. & Goodall, J. L. 2011 Feedback loops and temporal misalignment in component-based hydrologic modeling. *Water Resources Research* **47** (12), W12520.

FluidEarth SourceForge Project 2012 Open Source Development for FluidEarth, http://sourceforge.net/projects/fluidearth/ (accessed 18th October 2013).

Grass, A. J. 1981 Sediment Transport by Waves and Currents. SERC London Centre of Marine Technology, Report FL29.

Gregersen, J. B., Gijsbers, P. J. A. & Westen, S. J. P. 2007 OpenMI: open modelling interface. *Journal of Hydroinformatics* **9** (3), 175–191.

Harper, A., Cleverley, P. & Kelly, D. 2012 Source Forge FluidEarth 2 Download, http://sourceforge.net/projects/fluidearth/files/latest/download (accessed December 2012).

Hubbard, M. E. & Dodd, N. 2002 A 2D numerical model of wave run-up and overtopping. *Coastal Engineering* **47**, 1–26.

Hudson, J. 2001 Numerical Techniques for Morphodynamical Modelling. PhD Thesis, Department of Mathematics, University of Reading.

Lu, B. & Piasecki, M. 2012 Community modeling systems: classification and relevance to hydrologic modelling. *Journal of Hydroinformatics* **14** (4), 840–856.

Makropoulos, C., Safiolea, E., Baki, S., Douka, E., Stamou, A. & Mimikou, M. 2010 An integrated, multi-modelling approach for the assessment of water quality: lessons from the Pinios River case in Greece. In: *Proceedings of International Environmental Modelling and Software Society (iEMSs) 2010 International Congress*, Fifth Biennial Meeting, Ottawa, Canada.

Meiburg, S. in EPA 2008 *Integrated Modeling for Integrated Environmental Decision Making.* EPA-100-R-08-010. US Environmental Protection Agency, Office of the Science Advisor, Washington, DC.

Moore, R. V. 2010 *From Google Maps to Google Models*. AGU Fall Meeting, 13–17 December, San Francisco, CA.

Moore, R. V., Gijsbers, P., Fortune, D., Gregersen, J., Blind, M., Grooss, J. & Vanecek, S. 2010 *OpenMI Document Series: Scope for the OpenMI (Version 2.0)*. Butford Technical Publishing Ltd, Pershore, UK.

Morita, M. & Yen, B. 2002 Modeling of conjunctive two-dimensional surface-three-dimensional subsurface flows. *Journal of Hydraulic Engineering* **128** (2), 184–200.

OpenMI Association Website 2012a What is OpenMI? The OpenMI Association, http://www.openmi.org/new-to-openmi#TOC-What-is-OpenMI- (accessed 15th August 2012).

OpenMI Association Website 2012b Short History? The OpenMI Association. http://www.openmi.org/new-to-openmi#TOC-Short-history (accessed 21st August 2012).

OpenMI Association Website 2010c What's New in OpenMI 2.0?, The OpenMI Association, http://www.openmi.org/learning-more#TOC-OpenMI-manuals-and-guidelines (accessed 22nd August 2012).

OpenMI Association Website 2010d OpenMI Standard 2 Specification. The OpenMI Association, https://sites.google.com/a/openmi.org/home/learning-more/OpenMIStandard2InterfaceSpecification.pdf?attredirects=0 (accessed 24th October 2013).

OpenMI SourceForge Project 2010 OpenMI, The OpenMI Association, http://sourceforge.net/projects/openmi/ (accessed 18th October 2013).

Perdreau, N. & Cunge, J. A. 1971 Sedimentation dans les estuaries et les embouchures bouchon marin et bouchon fluvial. In: *14th Congress of the IAHR*, Paris.

Safiolea, E., Baki, S., Makropoulos, C., Deliege, J. F., Magermans, P., Everbecq, E., Gkesouli, A., Stamou, A. & Mimikou, M. 2011 Integrated modelling for river basin management planning. *Proceedings of the ICE, Water Management* **164** (8), 405–419.

Shrestha, N. K., Leta, O. T., de Fraine, B., Van Griensven, A., Garcia-Armisen, T., Ouattara, N. K., Servais, P. & Bauwens, W. 2012 Integrated modelling of river Zenne using OpenMI. In: *Proceedings of 10th International Conference on Hydroinformatics, HIC 2012*, Hamburg, Germany.

Sutherland, J., Bolster, M. & Harper, A. 2013 Beachplan as an Open-MI composition. In: *Proceedings of the 35th IAHR World Congress*, Chengdu, China. In press.

Voinov, A. 2010 Model integration and the role of data. *Environmental Modelling & Software* **25**, 965–969.

First received 30 July 2013; accepted in revised form 19 November 2013. Available online 16 December 2013

Appendix IV: Towards standard metadata to support models and interfaces in a hydro-meteorological model chain

Adapted from J. Hydro volume 17, issue number 2, pages 260-274, with permission from the copyright holders, IWA Publishing.

Journal of Hydroinformatics

Towards standard metadata to support models and interfaces in a hydro-meteorological model chain

Quillon Harpham and Emanuele Danovaro

ABSTRACT

This paper seeks to move towards an un-encoded metadata standard supporting the description of environmental numerical models and their interfaces with other such models. Building on formal metadata standards and supported by the local standards applied by modelling frameworks, the desire is to produce a solution, which is as simple as possible yet meets the requirements to support model coupling processes. The purpose of this metadata is to allow environmental numerical models, with a first application for a hydro-meteorological model chain, to be discovered and then an initial evaluation made of their suitability for use, in particular for integrated model compositions. The method applied is to begin with the ISO19115 standard and add extensions suitable for environmental numerical models in general. Further extensions are considered pertaining to model interface parameters (or phenomena) together with spatial and temporal characteristics supported by feature types from climate science modelling language. Successful validation of parameters depends heavily on the existence of controlled vocabularies. The metadata structure formulated has been designed to strike the right balance between simplicity and supporting the purposes drawn out by interfacing the Real-time Interactive Basin Simulator hydrological model to meteorological and hydraulic models and, as such, successfully provides an initial level of information to the user.

Key words | controlled vocabulary, environmental numerical modelling, ISO19115, metadata standard, model interface

Quillon Harpham (corresponding author)
HR Wallingford,
Howbery Park, Wallingford,
Oxfordshire OX10 8BA,
UK
E-mail: *q.harpham@hrwallingford.co.uk*

Emanuele Danovaro
CNR-IMATI,
Via De Marini 6 (11th Floor),
16149 Genova,
Italy

ACRONYMS AND ABBREVIATIONS

DRIHM2US	Distributed Research Infrastructure for Hydro-Meteorology to the United States of America
DTM	Digital terrain model
EPA	Environmental Protection Agency
NetCDF	Network Common Data Form
WaterML	Water markup language

INTRODUCTION

It is common practice to pass data between environmental numerical models. A typical one-way connection would consist of part of the output of one model becoming part of the input to the next model down the chain. Building on early incarnations of this process supported by bespoke scripts and file types, many frameworks designed to reduce the effort in achieving such couplings now exist. Johnston *et al.* (2011) describe a US EPA integrated modelling framework for environmental assessment using the Framework for Risk Analysis of Multi-Media Environmental Systems (FRAMES) system; Weerts *et al.* (2010) demonstrate these processes in operational forecasting with the Delft – Flood Early Warning System (FEWS) forecasting platform using published interfaces between models encoded in extensible markup language (XML) and utilising adaptors to handle any differences between outputs produced and inputs required; the Earth System Modelling Framework (ESMF)

doi: 10.2166/hydro.2014.061

is building a flexible software infrastructure to increase interoperability and reuse in numerical weather prediction and other environmental applications (Hill *et al.* 2004). Peckham *et al.* (2013) describe the design of a component-based approach to integrated modelling in the geosciences and Peckham & Goodall (2013) build on this further by demonstrating interoperability between two independently developed frameworks for models and data. Formal standards for model coupling are now also coming to the fore. Following the earlier open modelling interface (OpenMI) 1.4 (Gregersen *et al.* 2007), OpenMI 2.0 has been ratified by the Open Geospatial Consortium (OGC). OpenMI allows a two-way exchange of data between model components so that they may influence each other as they run (OpenMI Association Website 2014). OpenMI is itself supported by software tools allowing models to be adapted and coupled more easily. One such implementation is HR Wallingford's FluidEarth (Harpham *et al.* 2014) giving a software development kit (SDK) and graphical user interface (GUI) environment together with other supporting material and training.

By definition, the object interfaces defined within the OpenMI specification point the way to metadata describing the model components adapted to be OpenMI compatible. For example, 'output exchange items' are derived to pass data out of the model into another model's 'input exchange items'. Indeed, across all appropriate disciplines, metadata describing numerical models is clearly required to support any kind of automation or semi-automation of the model coupling process. Geller & Melton (2008) look forward to studying the impacts of climate change using a model web where data are passed between models using web services, which would, by definition, be supported by a set of such standards.

Nativi *et al.* (2013) emphasise the need for a clear information model for accommodating the components supporting environmental modelling including model engines and model services. This is supported by FluidEarth's model cataloguing component, configured to describe models as engines (core code) and instances (configured applications). Furthermore, Voinov *et al.* (2014) challenge the very basic processes underpinning common approaches to modelling and recommend a participatory approach, which challenges the traditional approach to modelling itself as a process beginning with a problem formulation and finishing with a product such as a decision support system. Such thinking would surely demand greater flexibility and more accurate representation from a typical modelling framework.

Given these drivers and building on formal metadata standards supported by the local standards applied by modelling frameworks, this paper seeks to derive an un-encoded metadata structure supporting the description of environmental numerical models with particular attention to the construction of model compositions by interfacing independent model components. The desire is to produce a solution that is as simple as possible yet supports validation of model interfaces together with basic discovery and use requirements.

METHODS

Formulating model engine metadata

Beginning with the model engine, that is the core model code before it has been configured to apply to a particular use case, a number of formally ratified or community standards exist from which to build. In atmospheric science, Murphy *et al.* (2009) describe two such metadata structures incorporated in the Earth System Grid (ESG) and European Common Metadata for Climate Modelling Digital Repositories (METAFOR) projects and characterise a finite volume dynamical core as having 'Basic properties', 'Technical properties', 'Scientific properties', 'Components' and 'Outputs'. The Community Surface Dynamics Modelling System (CSDMS) focuses, as its name would suggest, on modelling earth's surface systems and includes a model repository supported by a metadata structure with 'Summary', 'Contact', 'Technical specs', 'Input/output', 'Process', 'Testing', 'Other' and 'Component info' elements. This community seeks to create metadata for cataloguing earth surface dynamics models in building a catalogue of those available (CSDMS Model Repository 2014). The result is a community standard derived from a sensible set of descriptive fields and implemented in an online repository. ISO19115 (2003) offers an ISO ratified metadata standard for describing spatial datasets, the typical input to and output from environmental models. This standard offers a formal definition covering many similar fields to those required by CSDMS. Another ISO standard, ISO15836 (2009) gives the Dublin

Core Metadata Element Set, a more generic set of elements describing cross-domain resources. Once again, there are many similarities to the more specific ISO19115 and CSDMS community standards. For example, each includes an element providing a general description of the resource ('Abstract' in ISO19115, 'Description' (including an abstract construct) in ISO15836 and 'Extended model description' in CSDMS). The desire in this case is to formulate a candidate metadata structure, which supports the assembly of environmental model chains or compositions. In addition to the usual discovery and to use metadata requirements, particular attention must be paid to the interfaces between the model components. Ideally (and increasingly typically), these interfaces are governed by standards such as OGC OpenMI 2.0 (2014) or OGC WaterML 2.0 (2012) (see, for example, D'Agostino *et al.* 2014). Users must be able to analyse outputs coming from one model for suitability to use as inputs into another. The attributes associated with these inputs and outputs take particular importance and need to refer, where relevant, to the standards governing the interfaces. As such, ISO19115 was chosen as the starting point for the metadata formulation due to its specific design supporting spatial datasets (Hughes *et al.* 2013). Drawing from ISO19115 also allows use of a mature set of flexible cataloguing tools implementing the standard together with bespoke extensions such as the FluidEarth Catalogue (2011).

Initially, the approach of CSDMS and Murphy *et al.* (2009) was followed in drawing together the typical metadata elements required to describe a model engine. It has already been observed that a good proportion of these (such as a title, an abstract, owning organisation or contact details) are present in ISO19115 and more generically in ISO15836. Table 1 gives a base set of model engine metadata elements, their ISO19115 representation and application of each to a hydrological model.

A principal driver for this metadata formulation is to logically extend this description of environmental numerical models to that of their results datasets. Again, elements similar to those adopted by CSDMS (CSDMS Model Repository 2014) and Murphy *et al.* (2009) are applied as an extension to formulate the complete set of model engine metadata elements and ISO15836 offers a more generic approach including 'format' and 'coverage'. This extension was first applied as part of the FluidEarth model catalogue (FluidEarth Catalogue 2011) in describing model engines. Table 2 documents the FluidEarth extension to ISO19115 with a continuation of the hydrological model example.

Formulating base model instance metadata

When an environmental numerical model engine is applied to a particular situation, a place and a time, it becomes a model instance, which is an instance of that model engine. There is a natural inheritance relationship here where the model instances inherit all of the metadata from their parent model engine. This approach is followed in HR Wallingford's FluidEarth catalogue (FluidEarth Catalogue, 2011) with each model instance being directly associated with just one model engine thereby inheriting all of its metadata.

A further extension to the metadata elements defined above is required to give all of the metadata needed as a minimum to reasonably describe such a model instance. We begin with the spatial aspects with a view to discovering the model instance through a search of spatial extents. Indeed, this is part of the base functionality of the GeoNetwork cataloguing tool for spatial metadata (GeoNetwork 2014). Again, since they have been defined to describe spatial datasets, ISO19115 can provide these spatial elements. Table 3 gives two additional spatial elements used in this extension and shows how they are applied to the hydrological model example used previously.

Formulating interface driven model instance metadata

Further metadata is required to describe model instance outputs and inputs if the metadata set is to have any value in assessing the validity of interfaces to other models. If this metadata is to take a structured form across a large set of models, then the nature of the interfaces will need to be characterised in some way. Three aspects of the model inputs and outputs are singled out as having particular importance in evaluating model interfaces: the spatial characteristics, the temporal characteristics and the environmental parameters (or phenomena) described. These must be defined for each input and output.

The climate science modelling language (CSML) gives a set of 10 spatial feature types describing environmental data (Lowe 2011). Given in Table 4, they have been defined to be

Table 1 | Model engine metadata elements taken from ISO19115

Title, ISO19115 representation and description	Hydrological model example
Title (CI_Citation.title): the title of the dataset (model engine)	RIBS
Dataset Reference Date (CI_Citation.date) and DateType: the date marking the 'creation' of the dataset describing the model engine	2011-05-04: CI_DateTypeCode = creation
Abstract (MD_DataIdentification.abstract): description of the model engine	The Real-time Interactive Basin Simulator (RIBS) model is a distributed hydrological rainfall–runoff model that simulates the basin response to an event of spatially distributed rainfall. This model was designed for real-time application in medium-size basins. The model follows the structure of the grid of a DTM in a matrix form. The data are stored in layers of raster-type information, which are combined to obtain the model parameters
Point of Contact (Organisation) (CI_ResponsibleParty. organisationName): the organisation responsible for the model engine	Technical University of Madrid
Point of Contact (Online Resource) (CI_Contact.onlineResource): URL where more information can be obtained	www.upm.es
Point of Contact (Role): the precise role that the point of contact organisation plays identified as 'custodian'	CI_RoleCode = custodian
Point of Contact (Individual) (CI_ResponsibleParty.individualName): a person who can be contacted regarding this model engine	Luis Garrote
Point of Contact (Organisation) (CI_ResponsibleParty. organisationName): the organisation the individual point of contact belongs to	Technical University of Madrid
Point of Contact (Position) (CI_ResponsibleParty.positionName): the role occupied by the individual point of contact	
Point of Contact (Address and Email) (CI_Contact.address): the postal address of the individual point of contact including their email address	l.garrote@upm.es
Descriptive Keywords (MD_DataIdentification.descriptiveKeywords): a list of keywords describing the model engine	Rainfall, runoff, model
Topic Category Code (MD_TopicCategoryCode): the topic category to which the model belongs, most commonly 'Environment'	Geoscientific information
Date Stamp (MD_Metadata.dateStamp): the date (and time) stamp when the metadata file was created	2011-12-02T12:11:08

specialisations of the observations and measurements (O&M) model (ISO19156 2011) with the exception of 'observation' which is a direct usage. Crucially, these feature types are not only spatial representations, but also incorporate a temporal aspect.

This set of feature types is derived principally from considering earth observations from sensors of various kinds. However, a strong subset can be applied directly to numerical model output: PointSeries, ProfileSeries and GridSeries in particular. As such, the CSML feature types are adopted here as a controlled vocabulary for describing environmental numerical model inputs and outputs. In addition to this spatial and temporal description, a measure of the precise position of each input/output in space and time is required. The spatial aspect is given through a bounding box for each input and output (in addition to the bounding box representing the model instance as a whole); the temporal aspects are covered similarly by considering the time range covered by each input and output, as well as elements describing their associated timesteps.

Syvitski *et al.* (2014) highlight the need for precise description of model output and input parameter, units

Table 2 | FluidEarth extension to ISO19115 used to describe model engines

Title and description	Hydrological model example
Programming Language: the programming language(s) used to develop the model engine	C + +
Supported Platforms: the technical platform(s) supported by the model engine	Windows
Spatial Dimension: the spatial dimension of the model results	2
Source Code URI: a URI from which the source code of the model can be obtained	None supplied
Executable URI: a URI from which the model executable can be obtained	None supplied
Documentation URI: a URI from which the model documentation can be obtained	None supplied
Supported Model Standard: description of the model engine's compatibility with standards such as OpenMI and BMI (Peckham *et al.* 2013)	None
Supported Model Standard Version: the version of the compatible supported model standard	None
Number of Processors: the number of processors needed to run the model	1
Typical Run Time (and Time Unit): an estimate of the elapsed time for a typical run of the model. Although this may vary, it is included to give a 'ballpark' estimate	100 s
Input: input(s) to the model (Name, Description, Format, whether it is mandatory)	Name: DTM Description: digital terrain model of the basin Format: ESRI shapefile Mandatory: true
Output: output(s) from the model (Name, Description, Format, whether it is mandatory)	Name: hydrograph Description: discharges in time at selected locations Format: WaterML2 Mandatory: false

Table 3 | Additional spatial model instance metadata elements from ISO19115

Title, ISO19115 representation and description	Hydrological model example
Reference System (MD_ReferenceSystem.referenceSystemIdentifier): the coordinate reference system used	urn:ogc:def:crs:EPSG::3857
Extent (EX_GeographicBoundingBox): a geographic two-dimensional bounding box describing the extent of the model instance. The coordinates of the north, south, east and west bounds are given	8.8,44.3;8.8,44.4;9.0,44.4;9.0,44.3

and other attributes at interfaces between models. A set of standard parameter names, CSDMS standard names (CSDMS Standard Names 2013), is given as an extension to the well-established climate and forecasting standard names (CF Standard Names 2003), itself an extension to the Cooperative Ocean/Atmospheric Research Data Service standards (COARDS Conventions 1995). The metadata described here simply uses such standard naming conventions (which often produce very long parameter names) giving space for the precise parameter name and the unit used against each input and output.

The additional metadata elements given to support model interfaces are given in Table 5 with application to the hydrological model.

RESULTS AND DISCUSSION

General applicability

Further to the snippets given as the full metadata structure outlined above, a full example metadata set is given in

Table 4 | Climate science modelling language feature types

CSML feature type	Description	Example
Point	A single observation at a point	A single raingauge measurement
PointSeries	A series of 'Point' observations, varying in time, but not space	A stream of raingauge measurements
Profile	An observation along a vertical line in space	Air temperature at a varying height above sea level
ProfileSeries	A time-series of 'Profile' measurements	A set of air temperature profiles taken at a set of timesteps
Grid	Results given across a set of defined points in space	Two-dimensional high frequency (HF) Radar current output at a single time instant
GridSeries	A time-series of 'Grid' measurements from the same defined grid	Two-dimensional HF Radar current outputs at multiple time instants against the same set of grid points
Trajectory	An observation along a discrete path, varying in time and space	Water quality measurements taken from a moving ship
Section	A series of 'Profiles' from a 'Trajectory'	Marine CTD measurements taken from a moving ship
Swath	A 'Trajectory' but with two spatial dimensions resulting in a 'Grid' output but varying also in time	AVHRR satellite imagery taken from a satellite fly-past
ScanningRadar	Backscatter profiles along a look direction at fixed elevation but rotating in azimuth	Weather radar output

Table 6. It represents the metadata given by the Technical University of Madrid for a hydrological model called RIBS, the Real-time Interactive Basin Simulator (Garrote & Bras 1995), as part of the Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) project (Danovaro *et al*. 2014).

The result is a human readable metadata set giving the model engine elements together with the three inputs to the model and one output produced by it. The purpose of this metadata set is two-fold: (i) to allow the model to be found (discovery metadata) by potential users, and (ii) to allow potential users to evaluate whether the model is appropriate for their needs (use metadata). In general, the base ISO19115 metadata fields have been designed for these purposes for geospatial datasets, yet their extension into environmental models (in this case, a hydrological model) is equally as effective. The standard topic category code of 'Geoscientific Information' (itself from a keyword list) is generic and high level, but appropriate. Sensible search fields are present including abstract, keywords and point of contact details. The technical information added allows a rudimentary evaluation of the model yielding language and platform details together with a runtime estimate and uniform resource identifiers (URIs) where executables, documentation and source code can be found if they are available.

Evaluating interface feasibility using the RIBS model

We now consider whether it is possible to evaluate the feasibility of using output data from one model as input data to another using just the metadata for the two models. The RIBS model was selected, because it lies in the centre of a hydro-meteorological model chain. Precipitation predictions are provided as input to RIBS from meteorological models. RIBS calculates the catchment drainage and provides hydrographs into hydraulic models. These two file-based, one-way interfaces are denoted the 'P Interface' (or 'Precipitation Interface') and 'Q Interface' (or 'Flow Interface'). The P Interface is an example of passing gridded data between models where RIBS is the 'receiving model' and the Q Interface concerns point data where RIBS is the 'providing model'. This is illustrated in Figure 1. We consider each interface in turn.

The 'P' or 'Precipitation' Interface

The 'P' or 'Precipitation' Interface is the interface between the meteorological model and the hydrological model. The meteorological model produces a series of parameters, in particular precipitation, over the catchment to be drained. The meteorological model sequence can include

Table 5 | Additional model instance input and output metadata elements

Title and description	Hydrological model example
Feature Type: a description of the spatial/temporal structure of the data. Valid values from CSML feature type controlled vocabulary	GridSeries
Position: the two-dimensional geospatial position of the data given as a rectangular bounding polygon	8.8,44.3;8.8,44.4;9.0,44.4;9.0,44.3
Time Range: the timestamp of the first (earliest) and last (latest) reading in the time-series in ASCII format, i.e., YYYY-DD-MMThh:mm:ss + hh (e.g., 2014-01-31T15:46:51 + 01) defining the time interval of the data	2011-11-04T01:00:00 + 01, 2011-11-04T15:00:00 + 01
Timestep Type: indicator of 'regular' or 'irregular' timestep interval. Regular timestep types indicate a fixed interval or set of fixed intervals in the result dataset	Regular
Maximum Timestep Interval: the length of the largest timestep represented in the data and its unit of measurement. Used to allow validation of the temporal stability of interfaces	3,600 s
Minimum Timestep Interval: the length of the smallest timestep represented in the data and its unit of measurement. Used to allow validation of the temporal stability of interfaces	1,800 s
Parameter Name and Unit: the name and unit of measurement of the physical parameter/phenomenon represented	lwe_thickness_of_precipitation_amount m

downscaling routines and also the generation of ensembles. In all these cases, the interface to the hydrological drainage model is the same. The meteorological models produce results, which are usually represented as a three-dimensional terrain following GridSeries, as shown in Figure 2, with results being produced over a set of levels.

One of these three-dimensional results cubes is produced at each timestep. A wide variety of atmospheric parameters (or phenomena) are usually described, ranging from precipitation to wind to air pressure. Precipitation is applicable to the 'P Interface' and the parameter 'lwe_thickness_of_precipitation_amount' (CF Standard Names 2003), calculated at the surface only, is expected to be passed to the hydrological model as a two-dimensional GridSeries.

We now consider evaluating the feasibility of connecting a meteorological model (in this case, Weather Research and Forecasting – Advanced Research (WRF-ARW) model (Michalakes *et al.* 2004)) to RIBS using just metadata expressed using this structure. Table 7 shows the metadata element for an example output from WRF-ARW and Table 8 the counterpart input element, which describes what is expected by RIBS. Both model instances refer to a flash flood event that took place in Genoa, Italy in 2011 (Silvestro *et al.* 2012; Rebora *et al.* 2013; Fiori *et al.* 2014).

As previously discussed, the validation of this potential interface (i.e., whether it is valid to pass such data between the two models) should primarily concern the spatial characteristics, the temporal characteristics and the environmental parameters. The parameter matching is straightforward and depends on correct use of the controlled vocabulary used to describe the parameter and its unit of measurement. The output parameter 'Name' and 'Unit' needs to be compared to the input parameter 'Name' and 'Unit'. In this example, there is a direct match with 'lwe_thickness_of_precipitation_amount' in 'm' supplied by WRF-ARW as output and expected by RIBS as input. If there is not an exact match between the two, the interface may still be valid if there is a formula for translating between the different parameters or units, but it is suggested that such adaptation into common standards be applied within the model suite (albeit as a separate module) and reflected in the metadata in the standard forms.

The temporal characteristics are evaluated by a direct comparison of 'Feature Type' elements (in this example, both 'GridSeries'), 'Timestep Type' (in this example, both 'Regular' but with result data containing more than one interval), the maximum and minimum 'Timestep Interval' and the 'Time Range'. An interface may be deemed valid if the input Time Range does not fall outside the output

Table 6 | Hydrological model example of a model instance metadata set

- Citation
  - Title: RIBS
  - Creation Date: 2011-05-04
- Abstract: The Real-time Interactive Basin Simulator (RIBS) model is a distributed hydrological rainfall–runoff model that simulates the basin response to an event of spatially distributed rainfall. This model was designed for real-time application in medium-size basins. The model follows the structure of the grid of a DTM in a matrix form. The data are stored in layers of raster-type information, which are combined to obtain the model parameters
- Point of Contact
  - Custodian Organisation Name: Technical University of Madrid
  - Custodian Online Resource: www.upm.es
  - Responsible Individual
    - Name: Luis Garrote
    - Organisation: Technical University of Madrid
    - Position:
    - Address and Email: l.garrote@upm.es
- Descriptive Keywords: rainfall, runoff, model
- Topic Category Code: geoscientific information
- Date Stamp: 2011-12-02T12:11:08
- Reference System: urn:ogc:def:crs:EPSG::3857
- Extent: 8.88,44.37; 8.88,44.50; 9.09,44.50; 9.09,44.37
- Programming Language: C + +
- Supported Platforms: Windows
- Spatial Dimension: 2
- Source Code URI:
- Executable URI:
- Documentation URI:
- Supported Model Standard: none
- Supported Model Standard Version: none
- Number of Processors: 1
- Typical Run Time
  - Duration: 100
  - Unit: second
- Input
  - Name: DTM
  - Description: digital terrain model of the basin
  - Format: ESRI shapefile
  - Mandatory: true
  - Feature Type: Grid
  - Position: 8.88,44.37; 8.88,44.50; 9.09,44.50; 9.09,44.37
  - Parameter
    - Name: height above sea level
    - Unit: m
  - Time Range: none
  - Timestep Type: regular/irregular

(*continued*)

Table 6 | continued

Maximum Timestep Interval: none

Minimum Timestep Interval: none

Input

Name: soil type

Description: spatially distributed map of soil types, according to a local soil type categorisation

Format: ESRI Shapefile

Mandatory: true

Feature Type: Grid

Position: 8.88,44.37; 8.88,44.50; 9.09,44.50; 9.09,44.37

Parameter

Name: soil type

Unit: local categorisation

Time Range: none

Timestep Type: regular/irregular

Maximum Timestep Interval: none

Minimum Timestep Interval: none

Input

Name: precipitation

Description: spatially distributed fields of rainfall

Format: NetCDF 1.6

Mandatory: true

Feature Type: GridSeries

Position: 8.88,44.37; 8.88,44.50; 9.09,44.50; 9.09,44.37

Parameter

Name: lwe_thickness_of_precipitation_amount

Unit: m

Time Range: 2011-11-04T01:00:00 + 01,2011-11-04T15:00:00 + 01

Timestep Type: regular/irregular

Minimum Timestep Interval: 1,800 s

Maximum Timestep Interval: 3,600 s

Output

Name: hydrograph

Description: discharges in time at selected locations

Format: WaterML2

Mandatory: false

Feature Type: PointSeries

Position: 8.9538,44.4108; 8.9538,44.4109; 8.9539,44.4109; 8.9539,44.4108

Parameter

Name: River_Discharge

Unit: m^3s^{-1}

Time Range: 2011-11-04T01:00:00 + 01,2011-11-05T12:00:00 + 01

Timestep Type: regular

Maximum Timestep Interval: 300 s

Minimum Timestep Interval: 300 s

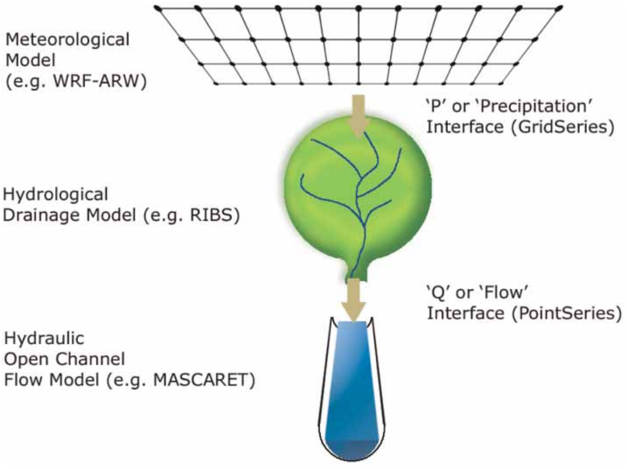

Meteorological Model (e.g. WRF-ARW)
'P' or 'Precipitation' Interface (GridSeries)
Hydrological Drainage Model (e.g. RIBS)
'Q' or 'Flow' Interface (PointSeries)
Hydraulic Open Channel Flow Model (e.g. MASCARET)

Figure 1 | The 'P' (Precipitation) and 'Q' (Flow) interfaces between the meteorological model, the hydrological drainage model and the hydraulic, open channel flow model.

Time Range and the Timestep Intervals between the two models are within a defined tolerance. These conditions may not always be necessary however, and this largely depends on how each model operates.

A comparison of spatial characteristics also depends on the Feature Type due to the dual spatial and temporal nature of this descriptor. Otherwise, the spatial validation consists solely of a comparison of 'Position'. Position consists of a bounding box (or polygon) expressed in the coordinate system defined once for the model instance. Usually, it would be expected that the input bounding box not lie outside that of the output model so that the spatial coverage required by the input model is guaranteed. If the bounding boxes are both rectangular, axis aligned and expressed in the same coordinate system then this comparison is simple, otherwise spatial functions to compare polygons and transform coordinate systems are required. Assuming the same coordinate system, in this example, it can be seen that the RIBS input bounding box lies within the WRF-ARW output bounding box sitting on its northern boundary, both expressing the boundary of the model grid supporting their respective GridSeries.

There are two remaining metadata elements to be considered when validating model interfaces: 'Mandatory' and 'Format'. Clearly, if an output from one model is not mandatory then the input model cannot expect to receive it – any interface between the models must have such output guaranteed. Also, the Format element is largely informational giving certain technical information, in this case, a NetCDF 1.6 file is passed by WRF-ARW and expected by RIBS. However, a direct match of a loosely typed structure such as this does not guarantee that the interface will operate without the need for interpolation between the two files, and moreover, a controlled vocabulary does not exist to allow direct text matching in this field.

The 'Q' or 'Flow' Interface

The Q Interface (the letter Q given to represent flow, or discharge) is the interface between the hydrological drainage model, RIBS and the hydraulic open channel model. RIBS calculates the drainage into the river channel and produces a hydrograph giving the flow at a certain point on the river network. Wherever hydraulic modelling is required, a hydrograph needs to be present. That is, for every reach of the river that requires open channel modelling, a flow-time boundary

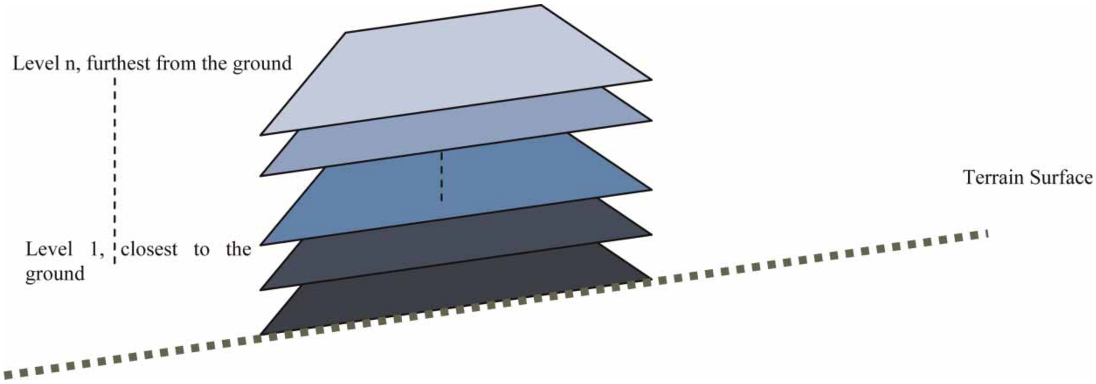

Level n, furthest from the ground
Level 1, closest to the ground
Terrain Surface

Figure 2 | Three-dimensional results from meteorological models.

Table 7 | Output metadata element from WRF-ARW

Output

- Name: precipitation
- Description: liquid water equivalent thickness of precipitation amount at the surface, defined as lwe_thickness_of_stratiform_precipitation_amount + lwe_thickness_of_convective_precipitation_amount
- Format: NetCDF 1.6
- Mandatory: false
- Feature Type: GridSeries
- Position: 8.50,44.25; 8.50,44.50; 9.25,44.50; 9.25,44.25
- Parameter
  - Name: lwe_thickness_of_precipitation_amount
  - Unit: m
- Time Range: 2011-11-04T01:00:00 + 01,2011-11-05T12:00:00 + 01
- Timestep Type: regular
- Minimum Timestep Interval: 900 s
- Maximum Timestep Interval: 3,600 s

Table 8 | Input metadata element from RIBS

Input

- Name: precipitation
- Description: spatially distributed fields of rainfall
- Format: NetCDF 1.6
- Mandatory: true
- Feature Type: GridSeries
- Position: 8.88,44.37; 8.88,44.50; 9.09,44.50; 9.09,44.37
- Parameter
  - Name: lwe_thickness_of_precipitation_amount
  - Unit: m
- Time Range: 2011-11-04T01:00:00 + 01,2011-11-04T15:00:00 + 01
- Timestep Type: regular
- Minimum Timestep Interval: 900 s
- Maximum |Timestep Interval: 1,800 s

condition must be supplied at the top of the stretch to be modelled. This information is passed to the hydraulic, open channel flow model, as illustrated in Figure 1.

We now consider evaluating the feasibility of passing the output from RIBS into a hydraulic open channel model (in this case, MASCARET (Goutal & Maurel 2002; Goutal *et al.* 2012)), using just metadata expressed in this structure.

Table 9 | Output metadata element from RIBS

Output

- Name: hydrograph
- Description: discharges in time at selected locations
- Format: WaterML2
- Mandatory: false
- Feature Type: PointSeries
- Position: 8.9538,44.4108; 8.9538,44.4109; 8.9539,44.4109; 8.9539,44.4108
- Parameter
  - Name: River_Discharge
  - Unit: m^3s^{-1}
- Time Range: 2011-11-04T01:00:00 + 01,2011-11-05T12:00:00 + 01
- Timestep Type: regular
- Maximum Timestep Interval: 300 s
- Minimum Timestep Interval: 300 s

Table 9 shows the metadata element for an example output from RIBS and Table 10 the counterpart input element, which describes what is expected by MASCARET. Again, both model instances refer to the same Genoa flash flood from 2011 and together with the WRF-ARW model instance constitute a viable model chain.

The metadata design leads to performing the same validation of this potential interface as for the example P

Table 10 | Input metadata element from MASCARET

Input

- Boundary Conditions
- Description: discharge or level hydrograph, rating curve
- Format: WaterML2
- Mandatory: true
- Feature Type: PointSeries
- Position: 8.95388,44.41083; 8.95388,44.41084; 8.95389,44.41084; 8.95389,44.41083
- Parameter
  - Name: River_Discharge
  - Unit: m^3s^{-1}
- Time Range: 2011-11-04T01:00:00 + 01,2011-11-05T12:00:00 + 01
- Timestep Type: regular
- Maximum Timestep Interval: 300 s
- Minimum Timestep Interval: 300 s

Interface, above. This time, the output parameters Name and Unit refer to a parameter called 'River_Discharge' measured in m^3s^{-1}. This parameter does not exist in CF Standard Names (CF Standard Names 2003). It has been defined as a candidate addition to such controlled vocabularies and corresponds to the 'Discharge, stream' item in the Consortium of Universities for the Advancement of Hydrological Science Incorporated – Hydrologic Information System (CUAHSI-HIS) ontology (Zaslavsky *et al.* 2012). A similar parameter, 'channel_outflow_end_water_discharge', also exists in the draft CSDMS standard names controlled vocabulary (CSDMS Standard Names 2013).

The temporal validation is the same as that explored above and gives the same outcome. A direct comparison of 'Feature Type' elements (in this example, both 'PointSeries') , 'Timestep Type' (both 'Regular'), the maximum and minimum 'Timestep Interval' and the 'Time Range' proceeds in the same way and yields the same uncertainty over validating timestep intervals and time ranges. However, the implications of a bounding box (or polygon) around a GridSeries feature type are somewhat different to that of a PointSeries. In this example, RIBS produces data as a single PointSeries and MASCARET is expecting to receive a PointSeries. Geospatially, this is represented by a single point and sensible validation would ensure that the point used by RIBS is in the same place as that expected by MASCARET. It is reasonable to assume that there will be rounding errors in each representation or that each model has expressed the point in a slightly different position (the point given in this example is on the Bisagno river above Genoa (see Silvestro *et al.* 2012)). As such, a tight bounding box is given to represent the RIBS output (instead of a single point) and another for the MASCARET input. If the same validation is used as in the P Interface, then the MASCARET bounding box must lie inside the RIBS bounding box for the interface to pass this validation.

Table 11 | Candidate set of model interface validation conditions

Condition	Pseudo-code
Parameter Name and Unit: the providing model output parameter name and unit must match with the receiving model input parameter name and unit	receivingModel.input.parameterName = providingModel.output.parameterName AND receivingModel.input.parameterUnit = providingModel.output.parameterUnit
Feature Type: the providing model output feature type must match the receiving model input feature type	receivingModel.input.featureType = providingModel.output.featureType
Timestep Type: if the providing model output has an irregular timestep, check that the receiving model can accept it	If providingModel.output.timestepType = 'irregular' then receivingModel.input.timestepType must = 'irregular'
Time Range: warn if the time range of the receiving model input lies outside the time range of the providing model output	receivingModel.input.timeRange.minimumTime > =providingModel.output.timeRange.minimumTime AND receivingModel.input.timeRange.maximumTime < =providingModel.output.timeRange.maximumTime
Timestep Interval: warn if the minimum timestep interval of the receiving model input is less than a defined multiplier of the maximum timestep interval of the providing model output	receivingModel.input.maximumTimestepInterval < = tolerance*providingModel.output.minimumTimestepInterval 'for an appropriate tolerance'
Position: the bounding box of the receiving model input has to be contained entirely within the bounding box of the providing model output	providingModel.output.position contains receivingModel.input.position 'or if geospatial functionality is not available, for rectangular bounded grids only and ignoring wrapping from 0 to 360 (or −180 to 180)': greatest providingModel.output.y-coordinate > =greatest receivingModel.input.y-coordinate AND smallest providingModel.output.y-coordinate < = smallest receivingModel.input.y-coordinate AND greatest providingModel.output.x-coordinate > =greatest receivingModel.input.x-coordinate AND smallest providingModel.output.x-coordinate < = smallest receivingModel.input.x-coordinate
Mandatory: warn if the providing model output is not mandatory	providingModel.output.Mandatory = false

As with the P Interface, above, if the output from RIBS is not mandatory, then MASCARET is not guaranteed to receive any data and the same issues arise with a comparison of the 'Format' element.

'P' and 'Q' Interface validation summary

Accordingly, a candidate set of validation conditions with pseudo-code supporting both the P and Q Interfaces (as examples of a typical file-based GridSeries-to-GridSeries and PointSeries-to-PointSeries interfaces) can be summarised in Table 11.

CONCLUSIONS

The purpose of metadata is to provide supporting information to allow what it is describing to be found, correctly interpreted and utilised. In environmental modelling use cases such as the hydro-meteorological model chain discussed here, the utilisation aspects increasingly depend on the ability to interface models with each other (and, indeed, other supporting datasets). Standards such as ISO19115 and ISO15836 provide formal patterns for establishing such metadata sets. The effectiveness of any metadata structure and its resulting encoding lies in achieving the right level of complexity for the common requirements to be placed on it. If the metadata is too comprehensive, then there is a risk that suppliers will not provide it, or that provided metadata sets will be of low quality and not maintained. If the metadata is not comprehensive enough, then it will not be fit for its intended purpose.

The purpose of the metadata outlined here is to allow environmental numerical models to be discovered (discovery metadata) and then an initial evaluation made of their suitability for use (use metadata), in particular with reference to interfacing with other numerical models, with a first application for a hydro-meteorological model chain. As such, ISO19115 provides the important base elements as constructed for geospatial datasets, and a small number of additions extend its usage into environmental numerical models. Further extensions describing environmental parameters (or phenomena), temporal and spatial attributes have been added to allow analysis of potential interfaces using inputs and outputs as follows:

- Successful validation of parameters depends heavily on the existence of controlled vocabularies. The interfaces to and from the hydrological RIBS model example demonstrate that these controlled vocabularies are more mature when interfacing to meteorological models than to hydraulic models.
- A level of temporal validation can be achieved by considering a limited number of attributes, most importantly the time range covered by the model.
- Use of a bounding box (or polygon) to describe spatial coverage is satisfactory for all of the CSML defined feature types and is particularly simple to apply if rectangular and in a common coordinate system. Precise validation is not possible without providing metadata including complete and comprehensive descriptions of the geo-temporal structures supporting the data.

The metadata structure formulated has been designed to strike the right balance between simplicity and supporting the purposes drawn out by the hydro-meteorological model chain and, as such, successfully provides an initial level of validation. It is easy to establish a base knowledge of the model functions and technology, the temporal and spatial coverage and the environmental parameters handled. This extends to individual interfaces with metadata attribution added to model inputs and outputs. However, a more comprehensive analysis and, in particular, precise confirmation that a model interface is valid would only be possible with considerably more information. Attempting to provide this with metadata, which must be available before the datasets are produced by the models, risks construction of an unwieldy metadataset, which would unnecessarily duplicate supplementary and essential model documentation and subsequent results datasets represented in self-describing file types such as NetCDF (OGC NetCDF 2011) and WaterML2 (OGC WaterML 2.0 2012).

ACKNOWLEDGEMENTS

This research was co-funded by the European Commission (EC) 7th Framework Programme DRIHM Project, Grant

Number 283568 and DRIHM2US Project, Grant Number 313122. Metadata for the RIBS model was adapted from that supplied by Luis Garrote of the Universidad Politecnica de Madrid. Examples of models catalogued using a version of the metadata structure outlined in this paper can be found in the DRIHM Model Catalogue (2014).

REFERENCES

CF Standard Names 2003 CF Metadata NetCDF CF Metadata Conventions. http://cf-convention.github.io/index.html (accessed 29 April 2014).

COARDS Conventions 1995 Conventions for the Standardization of NetCDF Files. http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html (accessed 29 April 2014).

CSDMS Model Repository 2014 All Models. http://csdms.colorado.edu/wiki/Model_download_portal (accessed 28 February 2014).

CSDMS Standard Names 2013 Standard Name Examples (Version 0.7.1). http://csdms.colorado.edu/wiki/CSN_Searchable_List (accessed 29 April 2014).

D'Agostino, D., Clematis, A., Galizia, A., Quarati, A., Danovaro, E., Roverelli, L., Zereik, G., Kranzlmuller, D., Schiffers, M., Gentschen Felde, N., Straube, C., Parodi, A., Fiori, E., Delogu, F., Caumont, O., Richard, E., Garrote, L., Harpham, Q., Jagers, B., Dimitrijević, V. & Dekic, L. 2014 The DRIHM project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological research. To be published in SC '14: In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, November 16–21, 2014* New Orleans, LA, USA.

Danovaro, E., Roverelli, L., Zereik, G., Galizia, A., D'Agostino, D., Quarati, A., Clematis, A., Delogu, F., Fiori, E., Parodi, A., Straube, C., Felde, N., Harpham, Q., Jagers, B., Garrote, L., Dekic, L., Ivkovic, M., Richard, E. & Caumont, O. 2014 Setup an hydro-meteo experiment in minutes: the DRIHM e-infrastructure for hydro-meteorology research. To be published in the proceedings of e-Science 2014: In: *10th IEEE International Conference on e-Science, October 20–24, 2014* Guarujá, SP, Brazil.

DRIHM Model Catalogue 2014 DRIHM Model Catalogue. http://drihmcatalogue.fluidearth.net/, in publication.

Fiori, E., Comellas, A., Molini, L., Rebora, N., Siccardi, F., Gochis, D., Tanelli, S. & Parodi, A. 2014 Analysis and hindcast simulations of an extreme rainfall event in the Mediterranean area: the Genoa 2011 case. *Atmos. Res.* **138**, 13–29.

FluidEarth Catalogue 2011 Welcome to the FluidEarth Catalogue. http://catalogue.fluidearth.net (accessed 2 May 2014).

Garrote, L. & Bras, R. L. 1995 A distributed model for real-time forecasting using digital elevation models. *J. Hydrol.* **167**, 279–306.

Geller, G. N. & Melton, F. 2008 Looking forward: applying an ecological model web to assess impacts of climate change. *Biodiversity* **9** (3–4), 79–93.

GeoNetwork 2014 GeoNetwork Open Source. http://geonetwork-opensource.org/ (accessed 2 May 2014).

Goutal, N. & Maurel, F. 2002 A finite volume solver for 1D shallow-water equations applied to an actual river. *Int. J. Numer. Methods Fluids* **38**, 1–19.

Goutal, N., Lacombe, J.-M., Zaoui, F. & El-Kadi-Abderrezzak, K. 2012 MASCARET: a 1-D open-source software for flow hydrodynamic and water quality in open channel networks. In: *River Flow* (Rafael Murillo Muñoz ed.). Taylor & Francis Group, London, pp. 1169–1174.

Gregersen, J. B., Gijsbers, P. J. A. & Westen, S. J. P. 2007 OpenMI: open modelling interface. *J. Hydroinform.* **9** (3), 175–191.

Harpham, Q. K., Cleverley, P. & Kelly, D. 2014 The Fluid Earth 2 implementation of OpenMI 2.0. *J. Hydroinform.* **16** (4), 890–906.

Hill, C., DeLuca, C., Balaji Suarez, M. & da Silva, A. 2004 The architecture of the earth system modeling framework. *Comput. Sci. Eng.* **6**, 18–28.

Hughes, A. G., Harpham, Q. K., Riddick, A. T., Royse, K. R. & Singh, A. 2013 *Meta-Model: Ensuring the Widespread Access to Metadata and Data for Environmental Models: Scoping Report*. British Geological Survey, Nottingham, UK, 39 (OR/13/042).

ISO15836 2009 ISO15836:2009 Information and Documentation – The Dublin Core Metadata Element Set. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=52142 (accessed 28 August 2014).

ISO19115 2003 ISO 19115:2003 Geographic Information – Metadata. http://www.iso.org/iso/catalogue_detail.htm?csnumber=26020 (accessed 2 May 2014).

ISO19156 2011 ISO 19156:2011 Geographic Information – Observations and Measurements. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32574 (accessed 2 May 2014).

Johnston, J. M., McGarvey, D. J., Barber, M. C., Laniak, G., Babendreier, J., Parmar, R., Wolfe, K., Kraemer, S. R., Cyterski, M., Knightes, C., Rashleigh, B., Suarez, L. & Ambrose, R. 2011 An integrated modeling framework for performing environmental assessments: application to ecosystem services in the Albemarle-Pamlico basins (NC and VA, USA). *Ecol. Modell.* **222**, 2471–2484.

Lowe, D. 2011 Climate Science Modelling Language v3.0 British Atmospheric Data Centre. http://csml.badc.rl.ac.uk/ (accessed 2 May 2014).

Michalakes, J., Dudhia, J., Gill, D., Henderson, T., Klemp, J., Skamarock, W. & Wang, W. 2004 The weather research and forecast model: software architecture and performance. In: *Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology* (Vol. 25, p. 29). World Scientific.

Murphy, S., Cinquini, L., Chastang, J., DeLuca, C., Middleton, D. & Balaji, V. 2009 Collaborative Model Metadata Development with ESG/Curator/Metafor. White paper.

ESMF website. http://www.earthsystemmodeling.org/publications/ (accessed 28 February 2014).

Nativi, S., Mazzetti, P. & Geller, G. N. 2013 Environmental model access and interoperability: the GEO ModelWeb initiative. *Environ. Modell. Softw.* **39**, 214–228.

OGC NetCDF 2011 OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0. Open Geospatial Consortium. http://www.opengeospatial.org/standards/netcdf (accessed 2 May 2014).

OGC OpenMI 2.0 2014 OGC Open Modelling Interface (OpenMI) Interface Standard. Open Geospatial Consortium Interface Standard. http://www.opengeospatial.org/standards/openmi (accessed 28 August 2014).

OGC WaterML 2.0 2012 OGC WaterML 2.0 Part 1 – Timeseries. Open Geospatial Consortium Implementation Standard. http://www.opengeospatial.org/standards/waterml (accessed 2 May 2014).

OpenMI Association Website 2014 What is OpenMI? https://sites.google.com/a/openmi.org/home/new-to-openmi#TOC-What-is-OpenMI- (accessed 28 February 2014).

Peckham, S. & Goodall, J. 2013 Driving plug-and-play models with data from web services: a demonstration of interoperability between CSDMS and CUAHSI-HIS. *Comput. Geosci.* **53**, 154–161.

Peckham, S., Hutton, E. & Norris, D. 2013 A component-based approach to integrated modeling in the geosciences: the design of CSDMS. *Comput. Geosci.* **53**, 3–12.

Rebora, N., Molini, L., Casella, E., Comellas, A., Fiori, E., Pignone, F., Siccardi, F., Silvestro, F., Tanelli, S. & Parodi, A. 2013 Extreme rainfall in the Mediterranean: what can we learn from observations? *J. Hydrometeorol.* **14**, 906–922.

Silvestro, F., Gabellani, S., Giannoni, F., Parodi, A., Rebora, N., Rudari, R. & Siccardi, F. 2012 A hydrological analysis of the 4 November 2011 event in Genoa. *Natl. Hazard. Earth Syst. Sci.* **12**, 2743–2752.

Syvitski, J. P., Hutton, E., Piper, M., Overeem, I., Kettner, A. & Peckham, S. 2014 Plug and play component modeling – the CSDMS2.0 approach. In: *International Environmental Modelling and Software Society (iEMSs), 7th International Congress on Environmental Modelling and Software* (D. P. Ames & N. Quinn, eds), San Diego, CA, USA. http://www.iemss.org/society/index.php/iemss-2014-proceedings.

Voinov, A., Seppelt, R., Reis, S., Nabel, J. E. M. S. & Shokravi, S. 2014 Values in socio-environmental modelling: persuasion for action or excuse for inaction. *Environ. Modell. Softw.* **53**, 207–212.

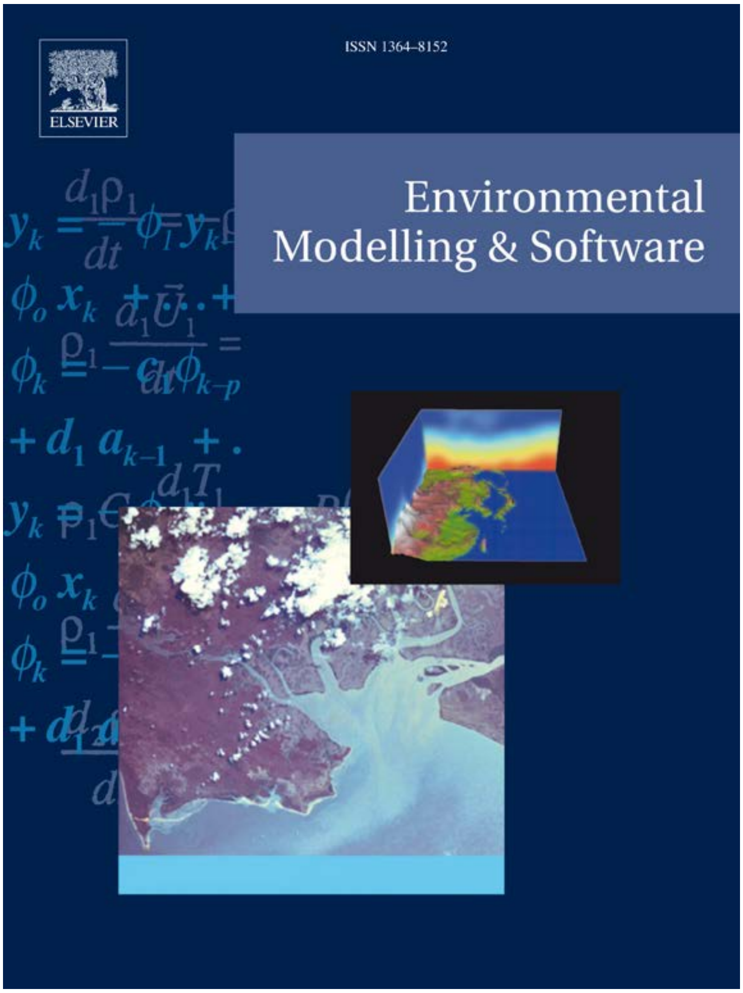
Weerts, A. H., Schellekens, J. & Weiland, F. S. 2010 Real-time geospatial data handling and forecasting: examples from Delft-FEWS forecasting platform/system. *IEEE J. Sel. Top. Appl.* **3** (3), 386–394.

Zaslavsky, I., Valentine, D., Hooper, R., Piasecki, M., Couch, A. & Bedig, A. 2012 Community practices for naming and managing hydrologic variables. In: AWRA 2012 Spring Speciality Conference, New Orleans, LA.

First received 13 May 2014; accepted in revised form 10 October 2014. Available online 26 November 2014

Appendix V: Using a Model MAP to prepare hydro-meteorological models for generic use

This article was published in Environmental Modelling & Software, 73, Harpham, Q., Cleverley, P., Danovaro, E., D'Agostino, D., Galizia, A., Delogu, F. and Fiori, E, Using a Model MAP to prepare hydro-meteorological models for generic use, pp.260-271, Copyright Elsevier (2015).


ISSN 1364-8152
ELSEVIER
Environmental Modelling & Software

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 xxx

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/© 2015 Elsevier Ltd. All rights reserved.

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.

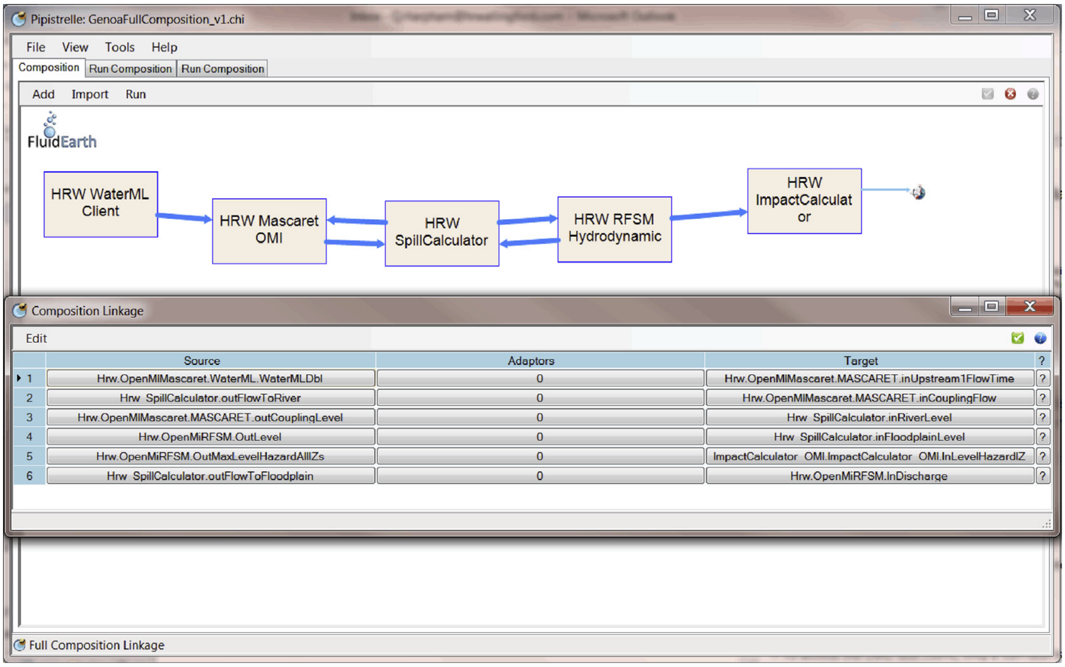
OpenMI (OGC OpenMI, 2014) is an accredited standard for model interoperability designed to enable the exchange of data between modelling components at run time. The first releases appeared in around 2004 with the latest version, 2.0 having been released in 2010. The specification for OpenMI consists 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 interface called 'Pipistrelle' for assembling compositions. Fig. 1 shows Pipistrelle being used as part of the DRIHM project to assemble a model composition designed to study flash flooding in Genoa. Each OpenMI component is represented by a box with the links between them given as arrows. This composition includes one and two-way connections between components so that numerical models can influence each other as the composition proceeds through its time-steps. This is to allow for the possibility that water leaving the river channel and proceeding onto the floodplain could subsequently re-enter the river or that flows from the river to the floodplain could change direction.

In addition to these tools FluidEarth also includes a model catalogue which implements an extension to the ISO19139 standard through GeoNetwork (Ožana and Horáková, 2008). This catalogue stores xml records of models and allows keyword and geospatial searching.

The Community Surface Dynamics Modelling System (CSDMS) has been in existence for a similar timeframe to OpenMI and FluidEarth and also includes a library of compatible models set out in a standard template. An ICT infrastructure for executing them is provided together with workflow facilities to allow models to be coupled together. The base design for CSDMS is described by Peckham et al. (2013). Numerical models are offered to the infrastructure through adherence to the Basic Model Interface (BMI) which implements a set of simple rules for structuring the model code and accessing base functions which must be present – a set of controls and descriptive information required for a component to be deployed in a typical modelling framework. Certain aspects of this approach are similar to that giving rise to the "GetCapabilities" function demanded by web service standards such as OGC WFS (OGC WFS 2.0.2, 2014) and OGC WMS (OGC WMS 1.3.0, 2006); a 'describe yourself' request where the web service outlines its makeup within that expected by the standard. Indeed, Peckham and Goodall (2013) demonstrate interoperability between the CSDMS 'plug-and-play' approach with the CUAHSI-HIS system (Tarboton et al., 2009) for storing and serving point series data. The models are driven directly with data from web services. This demonstrates that it is possible to interoperate standards-based data repositories with standards-based modelling infrastructures. CSDMS standard names (CSDMS Standard Names, 2013; Peckham, 2014) seeks to derive a directory of phenomena names including surface dynamics with the intention of being domain independent and avoiding domain-specific jargon in favour of broadly understood quantity names (such as "time_derivative" instead of "tendency"). Both version 1.6 of the CF Standard Names conventions (CF Standard Names, 2003) and version 0.8.3 of CSDMS Standard Names now total around 2500, with many CF Standard Names referring to chemicals occurring in the atmosphere, whereas CSDMS Standard Names covers a broader range concentrating on ease of parsibility and structuring around a natural alphabetical ordering.

Before we proceed further in defining a high level framework for numerical models, it is necessary to consider precisely what is meant by the term 'environmental numerical model', which is a little ambiguous. Environmental numerical models are usually written to have a set of core code modules. Together these are referred to as the 'model engine'. It is usually possible to apply the same model engine to different time periods in different locations. When this happens, configuration files are added to the model engine to allow it to be run. The model engine plus the configuration files is known as a 'model instance'. For example, RIBS (the Real-time Interactive Basin Simulator, Garrote and Bras (1995)) is an hydrological model engine which, given precipitation, calculates drainage into an open channel. It is possible to set up an instance of RIBS to study the Genoa flash flood of 4th November 2011. This involves providing a set of supporting files including rainfall, soil type and the topography of the Genoa region being studied. A calibration process may also be required. The RIBS model engine plus the final set of supporting (or configuration) files makes the model instance of RIBS for this flood event. As such, the term 'environmental numerical model' could refer to the model engine or the model instance. Moreover, the distinction between these is often blurred, particularly when geometry is less of an immediate concern. For example, meteorological models can include a



	Source	Adaptors	Target	?
1	Hrw.OpenMIMascaret.WaterML.WaterMLDbl	0	Hrw.OpenMIMascaret.MASCARET.inUpstream1FlowTime	?
2	Hrw SpillCalculator.outFlowToRiver	0	Hrw.OpenMIMascaret.MASCARET.inCouplingFlow	?
3	Hrw.OpenMIMascaret.MASCARET.outCouplingLevel	0	Hrw SpillCalculator.inRiverLevel	?
4	Hrw.OpenMiRFSM.OutLevel	0	Hrw SpillCalculator.inFloodplainLevel	?
5	Hrw.OpenMiRFSM.OutMaxLevelHazardAllIZs	0	ImpactCalculator OMI.ImpactCalculator OMI.InLevelHazardIZ	?
6	Hrw SpillCalculator.outFlowToFloodplain	0	Hrw.OpenMiRFSM.InDischarge	?

Fig. 1. The FluidEarth Pipistrelle user interface.

topography which is global in scale and so each model instance is more strongly defined by the set up required to study local conditions and the timescale to which it refers, rather than the topography to which it applies. In exploring a set of five environmental modelling catalogues, of which CSDMS is one, Zaslavsky et al. (2014) find that the emphasis is very much on recording model engines with almost all fields referring to this aspect. Some measure of local application is, however, mentioned in three out of the five. The FluidEarth catalogue lists both model engines and model instances with the model instances as specialisations of the model engines. The model instance records then inherit all of the metadata attributes from their parent model engine record. Given this ambiguity, whenever an aspect of the framework is applied to a 'numerical model' it must be made clear whether this refers to the model engine or instance.

We proceed by drawing three sets of observations from this discussion. A set of necessary information elements about model engines and instances is becoming clear: summary information such as name, description, type and version; owner or developer name and contact information; scientific background such as processes modelled, assumptions and limitations; technical specification such as languages and platforms; implementation details such as geospatial and temporal coverages. Zaslavsky et al. (2014) also record that, of the five model catalogues examined, model documentation and instructions is included in only two and licence in only one (CSDMS). Such aspects have clearly been assumed as provided elsewhere (or defaulted to local standard) since it is extremely difficult to use an unfamiliar model without any documentation and this should not be done without permission. Documentation, instructions and licence are taken here to be essential for interoperability of model components from a variety of providers.

Another key aspect is a need to standardise the form of model outputs and inputs, usually given as properties of the model engine. This is necessary for interfacing to other numerical models and also for post-processing functions such as visualisation. In OpenMI this is governed by the input and output exchange items, which as part of the standard, can be modified with an independent 'adaptor', provided separately from the modelling components themselves (OGC OpenMI 2.0, 2014). To avoid ambiguity and wherever possible, the parameters passed between models should be defined through controlled vocabularies such as the aforementioned CF Standard Names (2003) and newly formulated CSDMS standard names (2013).

Abstracting from the notion of a linkable component from OpenMI and the model core which is described by the BMI gives a generic portable model component, that is, a numerical model instance which can be executed independently: forced by one of its inputs as driving data to produce a set of coherent outputs. Coupling to this model is the equivalent of either replacing one of the regular inputs with that from another model, passing one of the outputs to another model or both of these simultaneously. In concept we wish to make no technical dependencies here, simply describing what is deemed necessary to ensure that the model component can exist independently of the environment in which it was created. Indeed, in outlining the GEO Model Web initiative, Nativi et al. (2013) comment that modelling frameworks often impose constraints on model developers by requiring specific technology or platforms.

Thus we are able to structure these three sets of observations into a candidate, high level framework giving what is necessary and sufficient for defining model interoperability with the initial use case being application to the DRIHM eInfrastructure. It is characterised as a Model MAP where each of the letters M, A, and P represent one of these aspects as follows:

- M – Metadata: Each model instance must be supplied with a descriptive metadata file, appropriate documentation and a licence to use it.
- A – Adaptors: Adaptors (or Bridges) must be provided, which translate the model inputs and outputs from and to common standards.
- P – Portability: Each model must be made portable, that is, not tied strongly to a local infrastructure or the environment in which it was created.

The Model MAP is not intending to replace or duplicate that which might already be present, it is simply a checklist of defined characteristics that a numerical model is required to follow.

2.1. M – Metadata (Documentation and Licence)

Each model instance must be supplied with a package of appropriate metadata, documentation and a licence to use it:

- A metadata file with sufficient information to allow the model to be discovered (see, for example O' Neill, 2004 and Weibel et al., 1998) and a preliminary evaluation undertaken for its use (see also Harpham and Danovaro, 2015). The metadata elements should include basic information such as name, description (or abstract) and version; scientific information such as input and output parameters (Whelan et al., 2014), spatial coverage and temporal coverage; technical information such as supported languages and operating systems.
- A licence which permits use of the model in the context supplied. An open source licence (such as LGPL or New BSD) – insisted upon by CSDMS – is preferred here over free-to-use or fully commercial licences.
- Documentation which adequately supports the model in the context supplied. This includes installation instructions, usage instructions, background information as well as a description of any changes which have been made to adhere to the Model MAP itself. The documentation requirements have been devised to be as simple and lightweight as possible whilst still achieving fitness for purpose.

2.2. A – Adaptors

The ambition of model engine inputs and outputs adapted to standard formats is worthy, but easier said than done. The requirement operates on the premise that there are appropriate target standards available for use. Indeed, if such standards do exist then it cannot be assumed that they will be sufficiently strongly typed to achieve interoperability with model components or other tools out-of-the-box without further modification. A detailed discussion of file (and memory) based standards will not be attempted here, however, as this aspect will be investigated as part of the results and discussion later.

2.3. P – Portability

Portability is also a property of a numerical model engine. It refers to generally good practice and housekeeping of the model code modules:

- The model engine does not expect any libraries other than native system libraries to be installed;
- The model engine package provides all binaries and libraries it depends on apart from the native system libraries assumed above;

- The model engine does not make any assumptions about the full directory structure of all files on which it depends and uses environment variables to create full paths;
- Each necessary file is given a unique version number and the model engine package contains an inventory of all dependent files with their versions listed. This does not include files which are created when a model instance is formed from the model engine, but only the files required to form the model engine itself.

The objective here is to be able to execute the model engine on an infrastructure other than that on which it was created. This requirement does not demand universal portability on any infrastructure which may exist, but an infrastructure similar to the native infrastructure of the model engine. For example, if a model engine has been written using C#.NET then it is unreasonable to assume that it be able to run on native Linux (or even Mono). However it is reasonable to assume that it will run on another.NET environment without additional non-standard libraries.

Having derived this set of gateway characteristics, we now compare them – and in particular their encapsulation in the associated metadata file – against the list of classes in the Component-Based Water Resources Model Ontology developed by Elag and Goodall (2013) (see also Nogueras-Iso et al., 2004). Developed with reference to concepts and properties used to describe models in Earth System Curator (Dunlap et al., 2008) and CSDMS (Peckham et al., 2013), this ontology is derived principally for water resources, but attempts to provide a conceptualized knowledge construct for defining model components across disciplinary boundaries. Comparison with this ontology is, therefore, particularly appropriate in this case since the Model MAP concept is motivated strongly from a multi-disciplinary context. This comparison is given in Table 1 and arranged according to the structure of superclasses within the high level ontology.

Notwithstanding the fact that domain comparisons are only partially possible, overall the ontology and the Model MAP yield a similar set of information elements. Most ontology superclasses are represented in the Model MAP, particularly for the resource, technical and coupling layers. This reveals a primary purpose of the Model MAP in facilitating use of model components out of native context in tandem with others and indicates a convergence of thinking on model components and coupling, even for a more pragmatic attempt to incorporate legacy models with potentially long histories and varying functional natures. The layer least represented in the Model MAP is the scientific layer where the MAP draws heavily on the legacy model documentation. This highlights a potential weakness in the Model MAP in not explicitly describing the underlying simulation equations, but thorough description of the I/O makes this less necessary for coupling and portability. The idea of assigning a component a "Development Level" (Argent, 2004) is adopted by the ontology in the resource layer as a means of describing the maturity of the code and its typical usage in simple terms. Development levels range from I (developed for research purposes) to IV (used in planning policy analysis). Such an evaluation is missing from the Model MAP and the author considers that it would make a highly appropriate addition to future Model MAP implementations.

The high level structures are similar in having many information elements (e.g. Organisation, Programming Language) associated closely with the component itself, however the Model MAP associates many of the other, common elements strongly with the input and output data. Indeed, elements attributed to the inputs and outputs are given in the ontology as belonging to both the coupling and scientific layers. The Model MAP structure is primarily to allow variation across a set of inputs and outputs with a view to coupling to or from any in that set. It is also a function of beginning with metadata standards for describing datasets (ISO19115 and 19139) where, invariably, focus will be given to the data structures surrounding the inputs and outputs to the model components. Elag and Goodall (2013) remark that 'other layer groupings such as model engines and model instances would not result in any major changes to the ontology concepts'. This is strongly supported through comparison of the ontology to the Model MAP which has been devised from this alternative layer grouping and has yielded similar results.

As such, a simple checklist for the Model MAP can be expressed as given in Table 2.

3. Results and discussion

3.1. Using the Model MAP on DRIHM

We begin by applying the MAP framework to the overall DRIHM eInfrastructure and then consider particular models within it. DRIHM is structured around three experiment suites as illustrated in Fig. 2.

Experiment Suite 1 involves running meteorological models to simulate atmospheric parameters such as precipitation, Experiment Suite 2 involves running hydrological models to simulate catchment drainage and produce discharge hydrographs at points in a river channel and Experiment Suite 3 involves running hydraulic model compositions to simulate open channel flow, flood spreading and its impact on infrastructure and people. Together, these three experiment suites are used to simulate a set of flash flooding use cases. Fig. 3 shows the modelling architecture for these experiment suites. The models are depicted as boxes on the diagram together with their adaptors (or bridges). Each numerical model is supplied with its own adaptor, but these are built to a common pattern and share code elements. The arrows show the flow of data between these components. The RainFarm and Meso-NH models can produce an ensemble of results, as indicated by the '+' signs.

Three interfaces are also shown on the figure: the 'P' or 'Precipitation' interface between the meteorological Experiment Suite 1 and the hydrological Experiment Suite 2; the 'Q' or 'Flow' interface between the hydrological Experiment Suite 2 and the hydraulic Experiment Suite 3; the OpenMI or 'O' interfaces between the hydraulic and impact models of Experiment Suite 3. The P and Q interfaces are file based, using the following standards:

- For P Interface grid-series data: NetCDF-CF 1.6, that is, CF-NetCDF 1.0 plus version 1.6 of the Climate and Forecasting naming conventions (Eaton et al., 2011) (OGC CF-netCDF 1.0 standard, 2013). This provides a compact format for the large meteorological datasets together with an appropriate controlled vocabulary for defining the data parameters. A set of additional rules were set out governing its use in this context including a smaller set of data parameters which would be supplied to downstream models.
- For Q Interface point-series data: WaterML 2.0 Part 1 – Timeseries (OGC WaterML2, 2012). This provides a metadata rich, xml encoded format designed exclusively for point-series datasets.

The O Interfaces are memory based, using OpenMI:

- Open Modelling Interface (OpenMI) Interface Standard 2.0 (OGC OpenMI 2.0, 2014) with the FluidEarth 2 implementation (Harpham et al., 2014).

Controlled vocabularies have been adopted throughout

Table 1
Comparison of Classes from the Component-Based Water Resource Model Ontology with the Model MAP as applied to the DRIHM eInfrastructure.

Component-Based Water Resource Model Ontology	Model MAP as applied to the DRIHM eInfrastructure 1 = from ISO19115/19139, 2 = from FluidEarth extension (mim = model instance metadata)
Resource Layer	
Developer	Attributed to the model engine using the following metadata elements, but individual responsibility is more commonly given as 'Custodian' rather than 'Developer': CI_Contact.onlineResource (1); CI_RoleCode = custodian (1) CI_ResponsibleParty.individualName (1)
Organisation (University, Company)	Attributed to the model engine with the following elements, but again offered as 'Custodian': CI_ResponsibleParty.organisationName (1); CI_Contact.address (1)
Project (Research, Teaching, Commercial)	No explicit reference to the originating project is given due to an assumption of a potentially long development history involving many initiatives. The following elements are attributed to the model engine as high level summary: CI_Citation.title (1); CI_Citation.date (1); MD_DataIdentification.abstract (1); MD_DataIdentification.descriptiveKeywords (1)
Development Level	Not given, but recommended as a potential addition.
Data (Data File (Geospatial (Vector, Raster), XML, Tabular, TimeSeries (WaterML)), Data Value)	Given for each model instance input and output featuring separate file standards for each geo-temporal feature type and also in memory coupling. Uses element mim:featureType (2) to describe the geo-temporal structure and mim:format (2) to describe the data format (e.g. WaterML2.0).
Scientific Layer	
Symbol (Universal Constant, Parameter, Variable (Dependent, Independent))	Assumption of use of parameters only, given in controlled vocabularies and as a property of each model instance input/output. Expressed by mim:parameterName (2).
Units	A property of each model instance input/output and subject to controlled vocabularies. Expressed by mim:parameterUnit (2).
Mathematical Classification (Deterministic, Stochastic)	Reference to source model documentation for model engine with mim:documentationUri (2).
Equation (Assumption, Initial Condition, Boundary Condition, Equation Type (Integral, Algebraic, Differential), Numerical Simulation (TimeDifference Scheme, Numerical Technique))	Tacit assumption of time-stepping scheme. Reference to source model documentation with mim:documentationUri (2).
Domain, applied as WaterResource Domain including Hydrology, Evapotranspiration and Ground Water.	Direct comparison only possible with hydrologic models, but this aspect is covered with reference to source model documentation with mim:documentationUri (2).
Technical Layer	
Programming Language	Attributed to model engine with mim:programmingLanguage (2), mim:sourceCodeUri (2), mim:documentationUri (2) and mim:executableUri (2).
Operating System	Attributed to model engine with mim:supportedPlatform (2).
Number of Processors	Attributed to model engine with mim:numberOfProcessors (2) and supported by MD_TopicCategoryCodemim:typicalRunTime (2).
Memory Requirements	Not given, but would be a useful addition.
Coupling Layer	
Modelling Framework(Concurrent; Sequential)	No direct reference to a modelling framework in addition to that offered by the input and output standards.
Architecture	Coupling architecture assumed to be standards-based at interfaces.
Standard Interface	Currently attributed to the model engine with mim:openMiStatus (2), although since generalised to cover all interface standards with SupportedModelStandard.
Computational Resolution (Spatial Resolution (Spatial Extent and Spatial Dimension); Temporal Resolution)	Spatial coverage attributed to the model engine with MD_ReferenceSystem (1); EX_GeographicBoundingBox (1) and mim:spatialDimension (2). Spatial coverage also attributed to the inputs and outputs of each model instance with EX_GeographicBoundingBox (1). Temporal coverage attributed to the inputs and outputs of each model instance with mim:timeStart (2); mim:timeEnd (2). Temporal resolution attributed to the inputs and outputs of each model instance with mim:maximumTimestep (2); mim:minimumTimestep (2) and mim:timestepCategory (2) (regular, irregular).

following the model set out by Climate and Forecasting 1.6.

These three standards act as the target standards for the MAP 'Adaptors'. As such, a typical model component prepared to run on the DRIHM architecture includes these adaptors, which may be written using the same technology as the model itself. The overall intention is to simply adapt each model's inputs and outputs to the standards, not necessarily to provide a library of adaptors for general use – as is the intention with OpenMI, for example – although re-use of common modules is encouraged where technically practical. The model instance and its adaptors can be considered as a single entity: The DRIHM Model Package. This is represented in Fig. 4 for file-based interfaces and Fig. 5 for OpenMI compositions with OpenMI adaptors. The OpenMI composition as a whole can be considered the model package but, of course, the composition can consist of a single model. Adaptors can be applied as per the OpenMI standard definition or as separate OpenMI components in the composition.

With the DRIHM model package defined in this way, the metadata, documentation and licence applies to the whole model package including any adaptors, although this can present complications if the adaptor has been written under a different licence than the numerical model engine (see German and Hassan, 2009). The model owner has the right to choose any licence they prefer, but integrating different licences is considerably simpler with use of permissive open source licences. As such, use of the open source permissive (BSD) or copyleft (GPL/LGPL) licence is preferred. The

Table 2
Model MAP checklist.

Metadata, documentation and licence	
Model Instance Metadata file provided	For the model engine Custodian: online resource, individual name, organisation name, contact address; High level summary: Title, citation date, abstract, descriptive keywords; Development Level (I–IV); Documentation: Documentation URI; Technical: Programming language, source code URI, executable URI, supported platform, number of processors, typical run time, memory requirements; Coupling: Supported model standard, coordinate reference system, geographic bounding box, spatial dimension; For each input and output Feature type, format, parameter name, parameter unit, geographic bounding box, time start, time end, maximum timestep, minimum timestep, timestep category.
Documentation provided	Referred to under “documentation URI” in metadata file Scientific: Mathematical Classification, Equation, Domain Technical: Installation, Code Structure and Functionality, Architecture, Coupling Framework Use: User Guide, Set up and calibration
Licence provided	Referred to under “documentation URI” in metadata file Licence to use, (optional) open source licence
Adaptors	
Adapted inputs/outputs provided	Referred to under “Supported model standard” and “format” in metadata file All inputs and outputs which are to be made available for coupling standardised to coupling standards.
Portability	
Model must be portable (not tied strongly to a local infrastructure or the environment in which it was created)	The model engine package: Requires only native system libraries; Provides all binaries and libraries it depends on (apart from the native system libraries); Makes no assumptions about the full directory structure of all files on which it depends and uses environment variables to create full paths; Contains an inventory of all dependent files with their unique versions listed (not including files which are created when a model instance is formed).

documentation must also include information about supporting functions and tools including the adaptors/bridges, a description of MAP modifications, a technical diagram showing what is necessary to allow the DRIHM infrastructure to run the model and handle the files it produces, and information on how to set up (and, if necessary, calibrate) an instance of the model package.

In addition, each DRIHM Model Package must be supplied with a single metadata file for each instance created. This file must adhere to the DRIHM specified standard given in Harpham and Danovaro (2015). In addition to being a compact and sensible way of

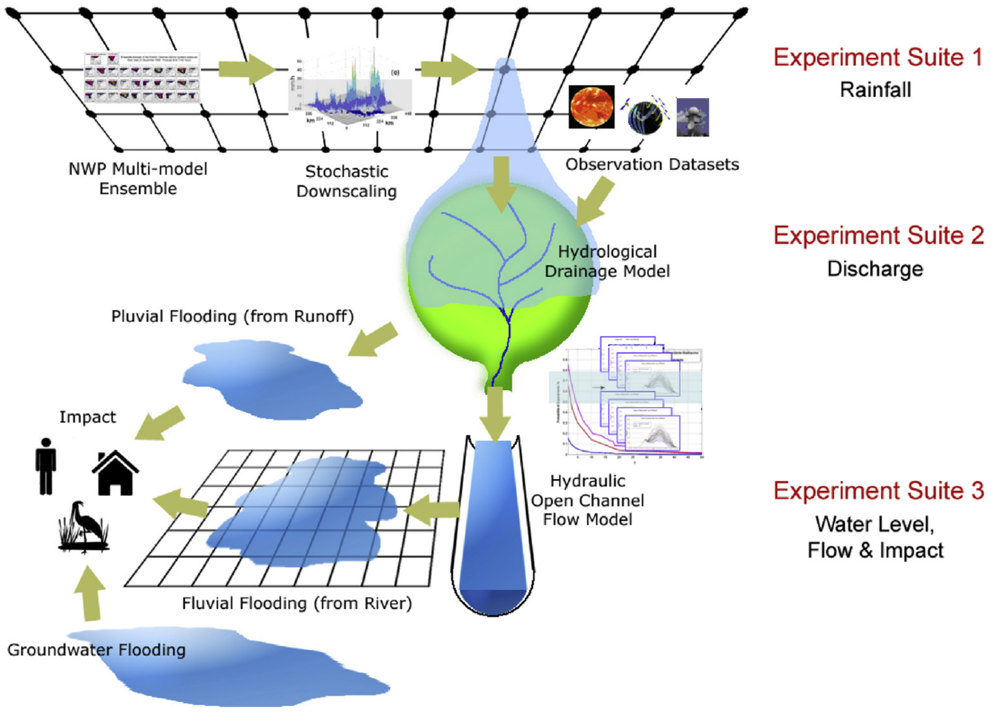


Fig. 2. The DRIHM experiment suites.

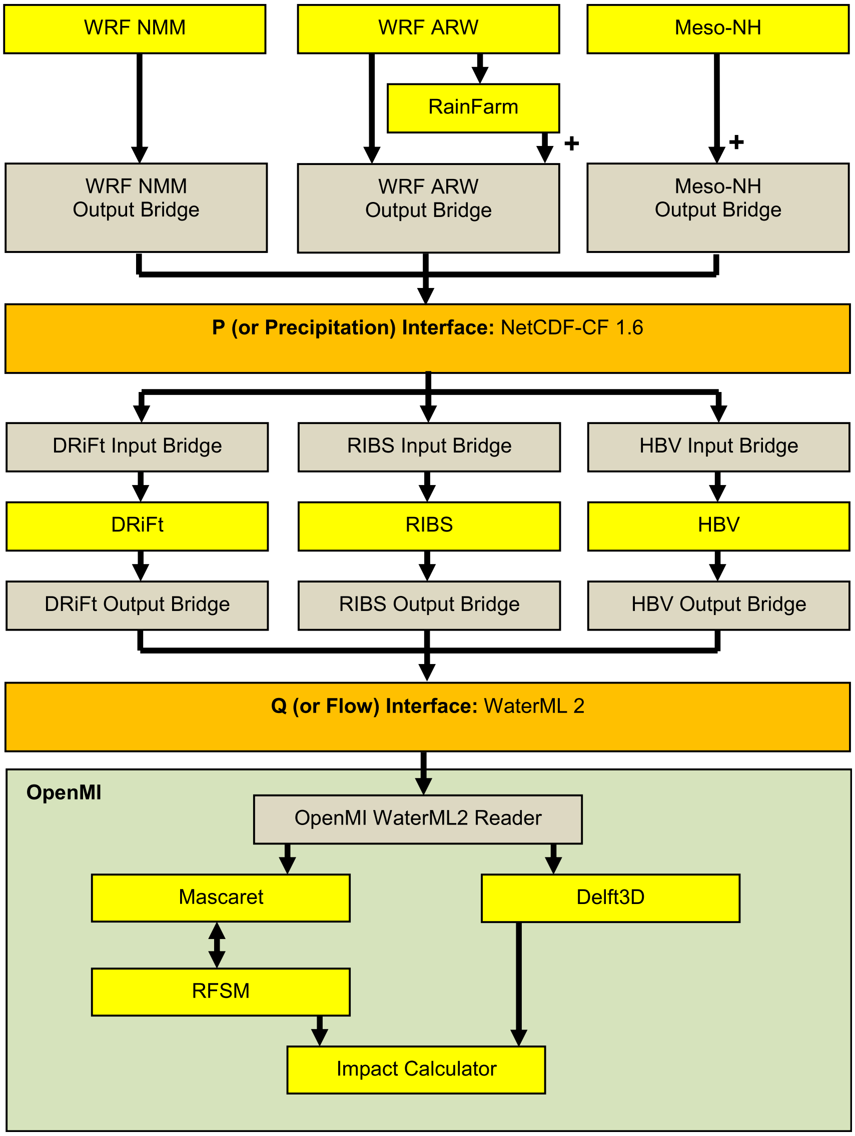


Fig. 3. The DRIHM model architecture.

describing a numerical model, this community standard serves two specific purposes: firstly to allow metadata for each model instance to be stored in the DRIHM Model Catalogue (2014) allowing the models to be found by keyword or geographical bounding box searches and secondly to facilitate linking models together. The concept adopted by FluidEarth of storing both instances and engines with the model instances as specialisations of the model engines is rejected in this case. This is due to the complexity of maintaining this relationship when subtle modifications of any aspect of the potentially complex DRIHM Model Package could result in a new version of a model engine. It is necessary to store metadata fields at the instance level in order to support validation

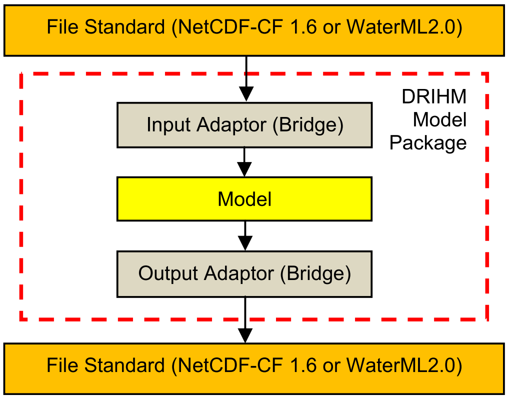

File Standard (NetCDF-CF 1.6 or WaterML2.0)
DRIHM Model Package
Input Adaptor (Bridge)
Model
Output Adaptor (Bridge)
File Standard (NetCDF-CF 1.6 or WaterML2.0)

Fig. 4. The DRIHM model package with file-based interfaces.

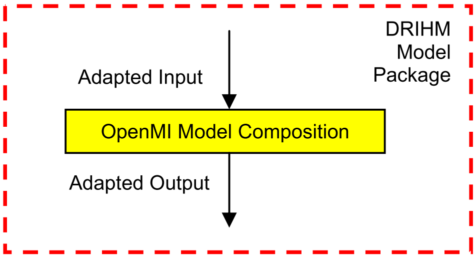


Fig. 5. An OpenMI Composition DRIHM Model Package with adapted inputs and outputs.

of potential linkages to other model instances. The DRIHM model metadata standard is based on the established ISO19115 metadata standard for describing spatial datasets. Specific elements of this have been used (such as Title, Abstract, Point of Contact, Bounding Box) together with certain extensions (mainly describing technical details and inputs/outputs) to allow DRIHM Model Packages to be represented in a human readable as well as machine readable form (Harpham and Danovaro, 2015). Each model record can be found by accessing the underlying xml encoding directly or displayed in human readable form by searching the catalogue through the user interface.

DRIHM is a distributed research infrastructure, that is, it depends on a set of different back-end resources including HPC, Grid and Windows Cluster. Successfully placing a model engine onto DRIHM, therefore, depends entirely on the integrity of its portability.

We now consider application of the Model MAP to two of the modelling components within DRIHM. Since this eInfrastructure offers a variety of components, we select two components that are quite different in nature: the meteorological model WRF-ARW (from the USA) and an hydraulic OpenMI composition (from Europe).

3.2. DRIHM WRF-ARW MAP

Developed at the National Center for Atmospheric Research (NCAR), the Advanced Research Weather Research and Forecasting model (WRF-ARW) (WRF Website, 2014) is an extensive and well-established numerical weather prediction (NWP) system. It has a large community of many thousands of users in a wide variety of countries. The documentation aspect is strongly supported, including descriptions of the underlying equations, the physics options available and installation instructions – many versions of the user guide have been released and many associated publications have been produced. It is licenced as an open source development managed by a number of groups including a Developers Committee and a Release Committee, each with appropriate terms of reference. It is made available under the WRF Public Domain Notice (2008).

The execution of WRF-ARW in the DRIHM Distributed Computing Environment requires that model instances and model engines be run not only on one particular machine but also on diverse machines under different execution environments. Following a check on the local presence of the NetCDF library, WRF-ARW was able to be installed using the standard installation procedure including that of the WRF Pre-processing System (WPS). WRF-ARW produces outputs in NetCDF format out-of-the-box, but this standard is too loosely typed for the purposes required here, in particular to allow semi-automated coupling to hydrological models downstream, supported by the instance metadata record. To meet this requirement, a WRF-ARW output bridge was created to take care of transforming the particular hydrological fields required by Experiment Suite 2 into the NetCDF-CF1.6 format required by the 'P' Interface – including specific definitions for time coordinates; default time zone; definition of horizontal coordinates as in latitude/longitude; enforcing the CF-conventions recommendation for variable dimensions to appear in order time, vertical, latitude, longitude; enforcing the OGC recommendation for variable long names; enforcing certain optional metadata such as title, history and institution. The full specification is given in Appendix III of DRIHM Consortium (2015).

When run, the model is supported by a shell script that sets the main environment variables (like input and output directories and variable names) and runs the model executable. The encapsulation of workflows under DRIHM follows this and has also been achieved through shell scripts. The workflow scripts create temporary working directories, generate the initial and boundary condition files to the temporary working directories, execute the model in sequential or parallel model, generate WRF-ARW output files in NetCDF-CF 1.6 format and delete working directories. Local DRIHM aspects not covered by the base WRF documentation have been documented separately.

As a test case, an instance of WRF-ARW was configured to study the severe flash flood which hit the city of Genoa, Italy in November 2011 (Silvestro et al., 2012; Rebora et al., 2013; Fiori et al., 2014). A metadata record (Parodi, 2014) was inserted into the DRIHM catalogue describing this model package instance including the temporal and spatial coverage and details of inputs and outputs which could be passed to downstream models. The output phenomena are described in terms of the Climate and Forecasting vocabulary (CF Standard Names, 2003).

The mature, user tested documentation provided the necessary supporting information and the open licence allowed the model to be used as required in conjunction with the additional adaptor software. The adaptation itself was facilitated by the provision of NetCDF output in the base model requiring further processing only as a refinement into the tighter local NetCDF-CF 1.6 definition required by the 'P' interface, which provided that target standard definition. The well-established track record of the model ensured no major portability issues, notwithstanding the local configuration described. The pre-requisite installation of the NetCDF library would probably not hold for most infrastructures, however this aspect is covered adequately in the documentation.

The most problematic aspect of applying the MAP framework to this model was in the creation of the metadata file representing the model instance. An important purpose of this file is to outline the model engine inputs and outputs including their spatial and temporal coverages in order to support validation of interfaces to other models. Meteorological models tend to have a large number of these making the metadata file unwieldy. Moreover, the concept of a model engine and a model instance sits more naturally with hydrological or hydraulic models since they have an important, small-scale geospatial aspect to their model configuration. Meteorological models would, on deployment, typically include a global topography which would not change on local implementation removing the geospatial variation in each instance.

The Model MAP created a package for WRF-ARW to be used in a model chain alongside other numerical models from hydrology and hydraulics, with each represented in the same manner. The configured model instance can be discovered in the catalogue and a preliminary coupling assessment undertaken against the outputs advertised. In particular the temporal and spatial output coverages can be matched against required input coverages of downstream models, together with matching of parameter names from controlled vocabularies. Adaptation to a refined NetCDF standard reduces the effort required in coupling to downstream models and increases the level of automation possible due to a reduction of uncertainties in file formatting.

3.3. DRIHM hydraulic composition MAP

A composition of hydraulic and impact models is the final package in the DRIHM model chain as outlined in Fig. 3. It is driven from point-series hydrographs from hydrologic drainage models via the Q Interface. The composition, shown in the Pipistrelle user interface in Fig. 1, consists of a reader to translate the Q Interface WaterML2.0 input into that required by OpenMI input exchange items; MASCARET, a 1-dimensional open channel flow model; RFSM-EDA, a 2-dimensional flood spreading model; a spill calculator to govern the exchange of data between MASCARET and

Table 3
Bisagno Flood Event Hydraulic model composition metadata records in the DRIHM Model Catalogue.

Description	URL
DRIHM Model Catalogue	http://drihmcatalogue.fluidearth.net/
Hydraulic Model Composition	http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=22
MASCARET ID Open Channel Flow Model	http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=10
Spill Calculator	http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=20
RFSM-EDA Flood Spreading Model	http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=17
Impact Calculator	http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=9

RFSM-EDA; an impact calculator to evaluate damage to buildings and injuries to people. The file received by the reader from the Q-Interface includes a number of restrictions to the WaterML2 standard including: a level one element structure consisting of exactly gml:description, wml2:metadata, wml2:temporalExtent, wml2:localDictionary, wml2:samplingFeatureMember, and wml2:observationMember; a restriction to one spatial point per file to allow data to be identified by file name and parameter name; specific definitions for coordinate system (WGS84), timestamp formats, parameter vocabularies and null values. The full definition is given in Appendix IV of DRIHM Consortium (2015).

Documentation has been supplied for each of these models according to the requirements specified by DRIHM. This includes general descriptions of the models, the files required to set them up and instructions for running the composition as a whole. Documentation and training for the OpenMI aspects is comprehensive from the FluidEarth and OpenMI Association websites. Of these models, only MASCARET is available under an open source licence (GNU GPL and LGPL for one library), the others are licenced as free to use. The FluidEarth implementation of OpenMI is available open source under the new BSD licence.

OpenMI 2.0 components are, by definition, interoperable with other OpenMI 2.0 components out-of-the-box. Adaptation to the standard required by the Q Interface is obtained using the WaterML2.0 reader as shown in Fig. 3. This means that each individual component within the composition adheres to standard inputs and outputs (as defined by OpenMI) and this is also true of the composition as a whole for inputs through the use of the reader. The composition produces outputs in ASCII grid format for loading into a GIS system using local phenomena names where the parameters are not within the current scope of existing controlled vocabularies.

Portability must be attained by each model individually as well as for the Pipistrelle tool which controls the interaction between the components as the composition runs. MASCARET and Pipistrelle are well established, both reaching the expected level of portability. The other models are not distributed to the same degree, indeed, the Spill Calculator evolved from an adaptor and was specifically developed for this composition. However, attaining the required level of portability was seen as minimum standard programming practice.

Since the entire OpenMI composition is given to be the DRIHM Model Package, discovery and use metadata needs can be met with one single catalogued metadata file as given in Table 2. This single file described the entire composition noting and referencing the models within it. Spatial coverage was given by a bounding box encompassing all of the bounding boxes for the individual models and temporal coverage was defined in the same way. The associated FluidEarth composition definition (.chi) file can be loaded into Pipistrelle showing the structure of the composition and the models within it. However, these models are interoperable entities within themselves and it is valid for each individual model to also be considered as a feasible DRIHM Model Package. Therefore, in addition to the metadata covering the whole composition, a metadata record pertaining to each model was also placed in the catalogue. A summary of this is given in Table 3.

The DRIHM Model MAP was applied to the hydraulic composition, by applying the framework to each model individually and then aggregating. This collection was handled through an additional metadata item representing the entire composition. The documentation for each of the model engines within the composition would not typically reference an individual composition context, but as instances are created for a particular composition then the instance metadata records should reference this documentation. The concept of a model engine and instance sits very comfortably with hydraulic models due to the geospatial dependencies of setting up instances. Combining models into compositions presents potential licence conflicts, but was not an issue in this case due to the open source or free to use licencing terms.

The Model MAP created a package consisting of a composition of hydraulic/impact models, represented in the same manner as the meteorological and hydrological models further up the chain. As with the other models, the configured model instance can be discovered in the catalogue and a preliminary coupling assessment undertaken against the inputs advertised. The temporal and spatial input coverages can be matched against that offered by potential suppliers, together with matching of parameter names from controlled vocabularies. Adaptation from a refined WaterML2 standard reduces the effort required in coupling to hydrographs from upstream models and increases the level of automation possible due to direct identification of the coupling data and, again, a reduction of uncertainties in file formatting (Harpham, 2015). The resultant composition was demonstrated to be compatible with any equivalent environment by installation and running on a Windows cloud infrastructure.

4. Conclusions

The Model MAP can be considered as a checklist of requirements designed at a level such that it spans the functional and technical diversity of environmental numerical models. It is not invasive and assumes very little about the nature of the models themselves. Is such a high level concept of any value? It would seem that simply requesting that adequate, contextual documentation be provided would be fatuous, as would suggesting that each component be issued with a licence to use it. However, it is surprising how often these simple items are missing or are of insufficient quality. Again, model portability would be assumed by many developers as standard programming practice, however, the Model MAP checklist prompts appropriate testing to take place. Of the models tested here, the additional framework specific developments required several rounds of testing before full portability was attained. Moreover, the requirement for all non-standard libraries to be part of the DRIHM Model Package led to useful discussions on the 'standard' installation that should be acceptable.

The Model MAP requirement for a metadata file for each model instance has more traction, but assumes that the model engine/model instance concept is applicable. It was demonstrated that this is more so for some types of models than others, but was applied successfully to a model chain including meteorological, hydrological, hydraulic and impact models. The DRIHM metadata standard used is based upon ISO19115 following the observation of the high level of overlap with spatial data and environmental model output and is given an extension to cover those elements applicable only for numerical models. However, Zaslavsky et al. (2014) demonstrate that, although many model catalogues and information standards exist, there is insufficient commonality to suggest a universal approach at present.

The Model MAP requirement for adaptation of inputs and outputs into common standards assumes that those common standards exist. These standards are not specified, only that they should be used. As such, this requirement has considerable utility if appropriate standards can be found, but is redundant otherwise. Certainly very few universal standards exist with take-up across the environmental modelling domain. OpenMI 2.0, as accredited by the OGC, serves this purpose for in-memory coupling of models for both one and two-way interfaces offering interoperability out-of-the-box for those that share a common wrapper (i.e. JAVA or.Net). The two file-based standards used here, WaterML2.0 for point-series data and NetCDF-CF 1.6 for grid-series data, were applied successfully although further specification within the limits of each of these standards was required to gain the level of automated interoperability – governed by rule sets rather than manual intervention – required for the model chain. These included restricting to separate files for each geo-temporal structure and decisions on usage of coordinate systems and controlled vocabularies.

Since DRIHM is a distributed research infrastructure built on a variety of back end resources and which also encompasses a varied model suite exhibiting numerical model engines with varied functional and technical natures, a framework which is valid for DRIHM would point strongly to being applicable more generically.

Acknowledgements

This research was co-funded by the European Commission (EC) 7th Framework Programme DRIHM Project, Grant Number 283568 and DRIHM2US Project, Grant Number 313122. The author would like to acknowledge the contributions of the European Commission and both project teams.

The WRF-ARW Model MAP was completed by Antonio Parodi of CIMA Research Foundation, Savona, Italy. The author would also like to acknowledge this important contribution.

References

Argent, R., 2004. An overview of model integration for environmental applications – components, frameworks and semantics. Environ. Model. Softw. 19 (3), 219–234.

Athanasiadis, I., Rizzoli, A., Janssen, S., Andersen, E., Villa, F., 2009. Ontology for Seamless Integration of Agricultural Data and Models. In: Sartori, F., Sicilia, M.A., Manouselis, N. (Eds.), In 3rd Intl Conf on Metadata and Semantics Research (MTSR'09). Springer-Verlag, pp. 282–293.

Bastin, L., Cornford, D., Jones, R., Heuvelink, G., Pebesma, E., Stasch, C., Nativi, S., Mazzetti, P., Williams, M., 2013. Managing uncertainty in integrated environmental modelling: the UncertWeb framework. Environ. Model. Softw. 39, 116–134. http://dx.doi.org/10.1016/j.envsoft.2012.02.008.

Balasko, A., Farkas, Z., 2011. gUSE Grid and Cloud Science Gateway (accessed 26.03.13.). http://sourceforge.net/projects/guse.

CF Standard Names, 2003. CF Metadata NetCDF CF Metadata Conventions. Available at: http://cf-convention.github.io/index.html (accessed 29.04.14.).

CSDMS Standard Names, 2013. Standard Name Examples (Version 0.8.3). Available at: http://csdms.colorado.edu/wiki/CSN_Searchable_List (accessed 29.04.14.).

D'Agostino, D., Clematis, A., Galizia, A., Quarati, A., Danovaro, E., Roverelli, L., Zereik, G., Kranzlmuller, D., Schiffers, M., gentschen Felde, N., Straube, C., Parodi, A., Fiori, E., Delogu, F., Caumont, O., Richard, E., Garrote, L., Harpham, Q., Jagers, B., Dimitrijevic, V., Dekic, L., 2014. The DRIHM project: a flexible approach to integrate hpc, grid and cloud resources for hydro-meteorological research", to be published in SC '14. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, New Orleans, LA, USA, November 16–21, 2014.

Danovaro, E., Roverelli, L., Zereik, G., Galizia, A., D'Agostino, D., Quarati, A., Clematis, A., Delogu, F., Fiori, E., Parodi, A., Straube, C., Felde, N., Harpham, Q., Jagers, B., Garrote, L., Dekic, L., Ivkovic, M., Richard, E., Caumont, O., October 20–24, 2014. Setup an hydro-meteo experiment in minutes: the DRIHM e-infrastructure for hydro-meteorology research. to be published in the proceedings of e-Science 2014. In: 10th IEEE International Conference on e-Science, Guarujá, SP, Brazil.

DRIHM Consortium, 2015. D6.2: Report on Application Services Delivery. DRIHM Website. http://www.drihm.eu/images/Deliverable/last/drihm-dwp6.2-20150228-4.0-HRW-Report_on_application_services_delivery.pdf (accessed 19.05.15.).

DRIHM Model Catalogue, 2014. DRIHM Model Catalogue. Available at: http://drihmcatalogue.fluidearth.net/ (accessed 20.11.14.).

Dunlap, R., Mark, L., Rugaber, S., Balaji, V., Chastang, J., Cinquini, L., DeLuca, C., Middleton, D., Murphy, S., 2008. Earth System Curator: metadata infrastructure for climate modelling. Earth Sci. Inf. 1, 131–149. http://dx.doi.org/10.1007/s12145-008-0016-1.

Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Caron, J., Signell, R., Bentley, P., Rappa, G., Höck, H., Pamment, A., Juckes, M., 2011. NetCDF Climate and Forecasting (CF) Metadata Conventions Version 1.6. http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html (accessed 21.11.14.).

Elag, M., Goodall, J.L., 2013. An ontology for component-based models of water resource systems. Water Resour. Res. 49, 5077–5091. http://dx.doi.org/10.1002/wrcr.20401.

Fiori, E., Comellas, A., Molini, L., Rebora, N., Siccardi, F., Gochis, D., Tanelli, S., Parodi, A., 2014. Analysis and hindcast simulations of an extreme rainfall event in the Mediterranean area: the Genoa 2011 case. Atmos. Res. 138, 13–29.

Garrote, L., Bras, R.L., 1995. A distributed model for real-time forecasting using digital elevation models. J. Hydrol. 167, 279–306.

German, D.M., Hassan, A.E., 2009. License integration patterns: addressing license mismatches in component-based development. In: Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on. IEEE, pp. 188–198.

Harpham, Q.K., Cleverley, P., Kelly, D., 2014. The Fluid Earth 2 implementation of OpenMI 2.0. J. Hydroinformatics. http://dx.doi.org/10.2166/hydro.2013.190, 16.4 890–906, IWA Publishing.

Harpham, Q.K., Danovaro, E., 2015. Towards standard metadata to support models and interfaces in a hydro-meteorological model chain. J. Hydroinformatics. http://dx.doi.org/10.2166/hydro.2014.061, 17.2 260–274, IWA Publishing.

Harpham, Q.K., 2015. DRIHM Report D6.2: Report on Application Services Delivery (accessed 07.05.15.). http://www.drihm.eu/images/Deliverable/last/drihm-dwp6.2-20150228-4.0-HRW-Report_on_application_services_delivery.pdf.

Michener, W.K., 2006. Meta-information concepts for ecological data management. Ecol. Inf. 1 (1), 3–7. http://doi.org/10.1016/j.ecoinf.2005.08.004.

Nativi, S., Mazzetti, P., Geller, G., 2013. Environmental model access and interoperability: the geo model web initiative. Environ. Model. Softw. 39, 214–228.

Nogueras-Iso, J., Zarazaga-Soria, F.J., Lacasta, J., Béjar, R., Muro-Medrano, P.R., 2004. Metadata standard interoperability: application in the geographic information domain. Computers. Environ. Urban Syst. 28 (6), 611–634. http://doi.org/10.1016/j.compenvurbsys.2003.12.004.

OGC CF-netCDF 1.0 standard, 2013. OGC Network Common Data Form (netCDF) Standards Suite (accessed 21.11.14.). http://www.opengeospatial.org/standards/netcdf.

OGC OpenMI 2.0, 2014. OGC Open Modelling Interface (OpenMI) Interface Standard. Open Geospatial Consortium Interface Standard (accessed 28.08.14.). http://www.opengeospatial.org/standards/openmi.

OGC WFS 2.0.2, 2014. OGC Web Feature Service 2.0 Interface Standard – with Corrigendum (accessed 18.11.14.). http://www.opengeospatial.org/standards/wfs.

OGC WMS 1.3.0, 2006. OGC Web Map Server Implementation Specification (accessed 18.11.14.). http://www.opengeospatial.org/standards/wms.

OGC WaterML 2.0, 2012. OGC WaterML 2.0 Part 1 – Timeseries. Open Geospatial Consortium Implementation Standard (accessed 02.05.14.). http://www.opengeospatial.org/standards/waterml.

O'Neill, K., 2004. A Specialised Metadata Approach to Discovery and Use of Data in the NERC DataGrid (accessed 19.11.04.). http://cedadocs.badc.rl.ac.uk/160/1/NDG_Poster.pdf.

Ožana, R., Horáková, B., 2008. Actual State in Developing GeoNetwork OpenSource and Metadata Network Standardization. GIS Ostrava 2008, Ostrava 27. – 30. 1. 2008.

Parodi, 2014. DRIHM Cataolgue WRF Bisagno Record (accessed 20.11.14.). http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=19.

Peckham, S.D., 2014. The CSDMS Standard Names: cross-domain naming conventions for describing process models, data sets and their associated variables. In: Ames, D.P., Quinn, N.W.T., Rizzoli, A.E. (Eds.), Proceedings of the 7th Intl. Congress on Env. Modelling and Software. International Environmental Modelling and Software Society (iEMSs), San Diego, CA. In: http://www.iemss.org/society/index.php/iemss-2014-proceedings.

Peckham, S., Goodall, J., 2013. Driving plug-and-play models with data from web services: a demonstration of interoperability between CSDMS and CUAHSI-HIS. Comput. Geosci. 53, 154–161. http://dx.doi.org/10.1016/j.cageo.2012.04.019.

Peckham, S., Hutton, E., Norris, D., 2013. A component-based approach to integrated modeling in the geosciences: the design of CSDMS. Comput. Geosci. 53, 3–12.

http://dx.doi.org/10.1016/j.cageo.2012.04.002.

Rebora, N., Molini, L., Casella, E., Comellas, A., Fiori, E., Pignone, F., Siccardi, F., Silvestro, F., Tanelli, S., Parodi, A., 2013. Extreme rainfall in the Mediterranean: what can we learn from observations? J. Hydrometeorol. 14, 906–922.

Silvestro, F., Gabellani, S., Giannoni, F., Parodi, A., Rebora, N., Rudari, R., Siccardi, F., 2012. A hydrological analysis of the 4 November 2011 event in Genoa. Nat. Hazard. Earth Syst. Sci. 12, 2743–2752.

Sutherland, J., Townend, I., Harpham, Q., Pearce, G., 2015. From Integration to Fusion: the Challenges Ahead. To Be Published in an Integrated Modelling Special Edition. Geological Society, London.

Tarboton, D.G., Horsburgh, J.S., Maidment, D.R., Whiteaker, T., Zaslavsky, I., Piasecki, M., Goodall, J., Valentine, D., Whitenack, T., 2009. Development of a community hydrologic information system. In: 18th World IMACS/MODSIM Congress, Cairns, Australia 13-17 July 2009. http://mssanz.org.au/modsim09.

Wang, W., Tolk, A., Wang, W., 2009. The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S. http://arxiv.org/abs/0908.0191.

Weibel, S., Kunze, J., Lagoze, C., Wolf, M., 1998. Dublin core metadata for resource discovery. Internet Eng. Task Force RFC 2413 (222), 132.

Whelan, G., Keewook, K., Pelton, M., Castleton, K., Laniak, G., Wolfe, K., Parmar, R., Babendreier, J., Galvin, M., 2014. Design of a component-based integrated environmental modeling framework. Environ. Model. Softw. 55, 1–24.

WRF Public Domain Notice, 2008. WRF Public Domain Notice (accessed 20.11.14.).

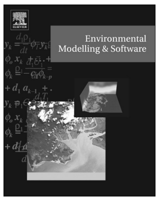
WRF Website, 2014. The Weather Research and Forecasting Model (accessed 20.11.14.). http://www.wrf-model.org/index.php.

Zaslavsky, I., Whitenack, T., Valentine, D., 2014. Exploring environmental model catalogs. In: Ames, D.P., Quinn, N.W.T., Rizzoli, A.E. (Eds.), Proceedings of the 7th International Congress on Environmental Modelling and Software, June 15-19, San Diego, California, USA, ISBN 978-88-9035-744-2.

Contents lists available at ScienceDirect

Environmental Modelling & Software

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



Corrigendum

Corrigendum to “Using a model MAP to prepare hydro-meteorological models for generic use” [Environ. Model. Softw. 73 (2015) 260–271]

Quillon Harpham

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

The authors regret that the co-author names were originally missed in the above published article:

The correct author details are as follows:

Full author list now given as follows:

Quillon Harpham[a,*], Paul Cleverley[a], Emanuele Danovaro[b], Daniele D'Agostino[b], Antonella Galizia[b], Fabio Delogu[c] and Elisabetta Fiori[c]

[a]HR Wallingford, Howbery Park, Wallingford, Oxfordshire, OX10 8BA, United Kingdom

[b]CNR – Institute of Applied Mathematics and Information Technologies, Italy

[c]CIMA Research Foundation, Italy

*Corresponding author. Tel.: +44 (0) 1491 822380; Fax: +44 (0) 1491 835381.

Correction to included reference as follows:

D'Agostino, D., Clematis, A., Galizia, A., Quarati, A., Danovaro, E., Roverelli, L., Zereik, G., Kranzlmuller, D., Schiffers, M., gentschen Felde, N., Straube, C., Caumont, O., Richard, E., Garrote, L., Harpham, Q., Jagers, B., Dimitrijevic, V., Dekic, L., Parodi, A., Fiori, E. and Delogu, F., 2014 The DRIHM project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological research”, SC ’14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. pp 536–546, IEEE Press Piscataway, NJ, USA © 2014. http://dx.doi.org/10.1109/SC.2014.49.

The authors would like to apologise for any inconvenience caused.

DOI of original article: http://dx.doi.org/10.1016/j.envsoft.2015.08.007.

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

http://dx.doi.org/10.1016/j.envsoft.2016.06.001
1364-8152/

Appendix VI: DRIHM(2US): an e-Science Environment for Hydro-meteorological research on high impact weather events

Available from https://journals.ametsoc.org/doi/full/10.1175/BAMS-D-16-0279.1

Appendix VII: Setup an hydro-meteo experiment in minutes: the DRIHM e-Infrastructure for HM research

Available from https://ieeexplore.ieee.org/abstract/document/6972248

Appendix VIII: The DRIHM project: A flexible approach to integrate HPC, GRID and Cloud resources for hydro-meteorological research

Available from https://ieeexplore.ieee.org/abstract/document/7013031

Appendix IX: Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy

Harpham, Q., Lhomme, J., Parodi, A., Fiori, E., Jagers, B. and Galizia, A., 2016. Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy. JAWRA Journal of the American Water Resources Association (2016). https://onlinelibrary.wiley.com/doi/abs/10.1111/1752-1688.12418,

JOURNAL OF THE AMERICAN WATER RESOURCES ASSOCIATION

AMERICAN WATER RESOURCES ASSOCIATION

USING OPENMI AND A MODEL MAP TO INTEGRATE WATERML2 AND NETCDF DATA SOURCES INTO FLOOD MODELING OF GENOA, ITALY[1]

Quillon Harpham, Julien Lhomme, Antonio Parodi, Elisabetta Fiori, Bert Jagers, and Antonella Galizia[2]

ABSTRACT: Extreme hydrometeorological events such as flash floods have caused considerable loss of life and damage to infrastructure over recent years. Flood events in the Mediterranean region between 1990 and 2006 caused over 4,500 fatalities and cost over €29 billion in damage, with Italy one of the worst affected countries. The Distributed Computing Infrastructure for Hydro-Meteorology (DRIHM) project is a European initiative aiming at providing an open, fully integrated eScience environment for predicting, managing, and mitigating the risks related to such extreme weather phenomena. Incorporating both modeled and observational data sources, it enables seamless access to a set of computing resources with the objective of providing a collection of services for performing experiments with numerical models in meteorology, hydrology, and hydraulics. The purpose of this article is to demonstrate how this flexible modeling architecture has been constructed using a set of standards including the NetCDF and WaterML2 file formats, in-memory coupling with OpenMI, controlled vocabularies such as CF Standard Names, ISO19139 metadata, and a Model MAP (Metadata, Adaptors, Portability) gateway concept for preparing numerical models for standardized use. Hydraulic results, including the impact to buildings and hazards to people, are given for the use cases of the severe and fatal flash floods, which occurred in Genoa, Italy in November 2011 and October 2014.

(KEY TERMS: computational methods; flooding; OpenMI; WaterML2; data management.)

Harpham, Quillon, Julien Lhomme, Antonio Parodi, Elisabetta Fiori, Bert Jagers, and Antonella Galizia, 2016. Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy. *Journal of the American Water Resources Association* (JAWRA) 1-17. DOI: 10.1111/1752-1688.12418

INTRODUCTION

Extreme hydrometeorological events such as flash floods have caused considerable loss of life and damage to infrastructure over recent years. An analysis carried out by the FLASH project calculated that flood events in the Mediterranean region between 1990 and 2006 caused 4,566 fatalities and cost over €29 billion in damage (Llasat *et al.*, 2010). Algeria recorded the highest number of total casualties (over 1,200) with Italy recording the most damage (almost €20 billion). The highest proportion of events took place in September, October, and November.

[1] Paper No. JAWRA-15-0074-P of the *Journal of the American Water Resources Association* (JAWRA). Received June 2, 2015; accepted February 9, 2016. © 2016 American Water Resources Association. **Discussions are open until six months from issue publication**.

[2] Principal Scientist (Harpham), Hydrodynamics and Metocean Group, formerly Senior Scientist (Lhomme), HR Wallingford, Howbery Park, Wallingford, Oxfordshire, OX10 8BA United Kingdom; now Senior Product Engineer (Lhomme), XP Solutions, Newbury, Berkshire, United Kingdom; Research Director (Parodi) and Researcher (Fiori), CIMA Research Foundation, Genoa, Italy; Technical Coordinator Software Development (Jagers), Deltares, Delft, Netherlands; and Researcher (Galizia), CNR—Institute of Applied Mathematics and Information Technologies, Genoa, Italy (E-Mail/Harpham: q.harpham@hrwallingford.co.uk).

The Distributed Computing Infrastructure for Hydro-Meteorology (DRIHM) project (D'Agostino *et al.*, 2014, 2015; Danovaro *et al.*, 2014) is a European initiative aiming at providing an open, fully integrated eScience environment platform for predicting, managing, and mitigating the risks related to such extreme weather phenomena. It enables seamless access to a set of computing resources with the objective of providing a collection of services for performing experiments with numerical models in meteorology, hydrology, and hydraulics. DRIHM (http://www.drihm.eu) is an example of a structured research environment offering users easy access to numerical models and supporting data sources, together with the computing resources required to run workflows incorporating them. Other examples of frameworks in the domain of integrated environmental modeling include the Framework for Risk Analysis of Multi-Media Environmental Systems (FRAMES) from the U.S. Environmental Protection Agency (USEPA), a desktop integrated modeling framework for environmental assessment (Johnston *et al.*, 2011) running highly simplified components using file-based data exchange; the Community Surface Dynamics Modeling System (CSDMS), which provides a web-based modeling tool (WMT) to configure and run geomorphological simulations across a wide range of time and space scales on a dedicated HPC platform, using a component-based approach (Peckham *et al.*, 2013) and the Earth System Modeling Framework (ESMF) that provides a flexible software infrastructure to support the development of integrated models building on high resolution parallelized numerical weather prediction (NWP) and other environmental components (Hill *et al.*, 2004). Compared to these other frameworks, DRIHM builds on a highly heterogeneous distributed infrastructure (composed of supercomputer, grid, and cloud architecture) with both file-based and in-memory data exchange.

The primary use case, demonstrating the functionality developed for DRIHM and offering a blueprint for other meteorology-driven use cases, is flash flooding. The user functions available are structured around three experiment suites: Experiment Suite 1: Meteorology (in particular rainfall); Experiment Suite 2: Hydrology (producing discharge hydrographs); and Experiment Suite 3: Hydraulics (in particular water level, flow and impact). This is illustrated in Figure 1.

The objective is to consider each experiment suite as an ensemble of broadly equivalent numerical models and data sources and allow users to perform a variety of experiments incorporating different combinations of each. The purpose of this article is to demonstrate how this flexible modeling architecture has been constructed using a set of standards including file formats, in-memory coupling, controlled vocabularies and metadata, and a Model MAP (Metadata, Adaptors, Portability) gateway concept for preparing numerical models for standardized use. A science gateway, the DRIHM portal, is used to invoke the models and pass results between them. Numerical modeling results, including the impact to buildings and hazards to people, are given for severe and fatal flash flooding, which occurred in Genoa, Italy.

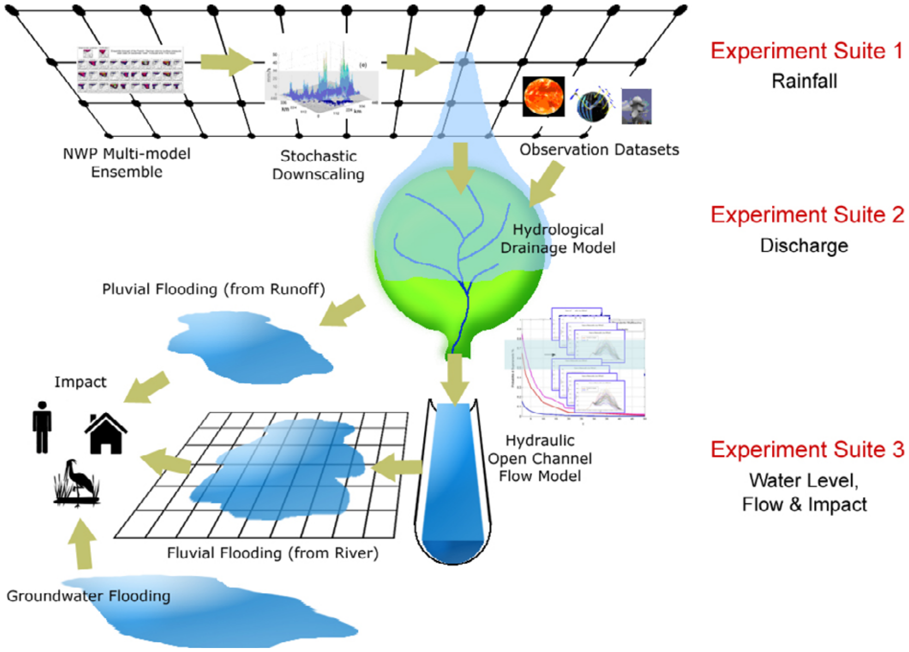


FIGURE 1. The DRIHM Experiment Suites: It Rains, It Drains, and It Floods. DRIHM, Distributed Computing Infrastructure for Hydro-Meteorology project; NWP, numerical weather prediction.

METHODS

The full suite of numerical model components and data sources is described by Figure 2. The overall structure mirrors that of the experiment suites illustrated in Figure 1. Notwithstanding the presence of an OpenMI composition at the hydraulic level, in contrast to model coupling approaches such as OpenMI, which takes a low level spatial structure approach combined with a separate temporal structure and the Basic Model Interface (CSDMS Basic Model Interface, 2012) which assumes a computational grid, the main model interfaces are based around spatio-temporal feature types, in particular grid series and point series. In addition to direct incorporation of rain gauge observations, three meteorological models, WRF-ARW, WRF-NMM (WRF Website 2014), and Meso-NH (Meso-NH Website, 2015) are included, together with RainFARM, a downscaling model (Rebora *et al.*, 2006). These models pass standardized data through a grid-series, file-based interface called the P-Interface (or Precipitation Interface) to a suite of hydrological models. In addition to direct incorporation of streamflow observations, three hydrological models, DRiFt (Giannoni *et al.*, 2000), RIBS (Garrote and Bras, 1995), and HBV (Bergström, 1995) are included. These models pass standardized point series data through a second file-based interface called the Q-Interface (or Flow Interface) to a suite of hydraulic models. These include an OpenMI (OGC OpenMI 2.0, 2014) composition incorporating MASCARET (Goutal and Maurel, 2002; Goutal *et al.*, 2012) and RFSM-EDA (Jamieson *et al.*, 2012a, b) and also Delft3D (Roelvink and Van Banning, 1995). The numerical models within each of the three domains are interoperable in the sense that they all can be interchanged and each set is extensible, readily admitting new models that perform the same (or similar) function. By the use of similar interfaces to the P- and Q-Interfaces, the whole numerical model architecture is extensible to new domains.

Formal standards and formulations leading to standards are used throughout to create the necessary interoperability and extensibility of the modeling architecture. These standards include file formats such as netCDF (OGC netCDF, 2013) and WaterML (OGC WaterML 2.0, 2012), semantic standards such as the Climate and Forecasting vocabulary for parameter naming and mediation between components (CF Standard Names, 2003) and memory-based model coupling with OpenMI (OGC OpenMI 2.0, 2014). The formulations are given in terms of a Model MAP gateway concept leading to numerical model interoperability (Harpham *et al.*, 2015).

The P-Interface: NetCDF and Controlled Vocabularies

The "P-" or "Precipitation" interface is a formalization, based on a set of standards, for passing data from meteorological models to hydrological models. The data that can be passed is not restricted to pre-

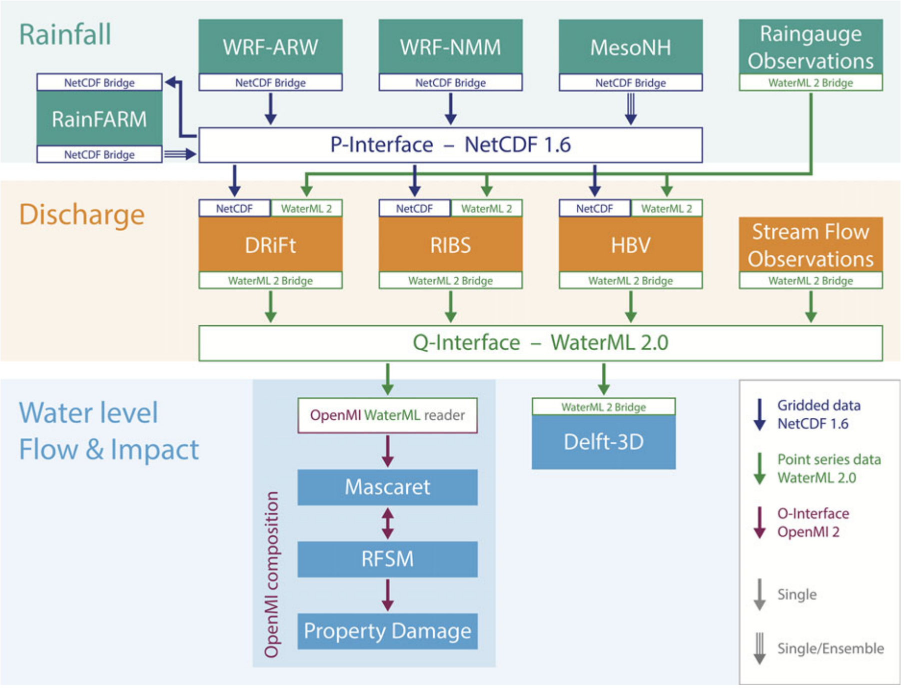


FIGURE 2. The DRIHM Numerical Modeling Architecture.

cipitation, indeed, any of the meteorological outputs can be used. However, the DRIHM use case (flash flooding) concentrates on passing precipitation data across this interface.

The P-Interface is a one-way, file-based interface. The output file(s) from the meteorological model is passed to the hydrological model(s) downstream. The data that passes across this interface is structured around a grid. As such, the data itself is represented by a grid or grid-series feature type and is given in netCDF-CF 1.6 format (OGC netCDF, 2013). The specification for the usage of the standard in this case is given in Appendix III of DRIHM Consortium D6.2 (2015). In particular, all data are represented in the WGS84 coordinate system (urn:ogc:def:crs:EPSG::4326). Use of Latitude and Longitude is required by the CF conventions, with WGS84 most commonly assumed. This has been emphasized here, as part of the DRIHM implementation, to keep geospatial transformations away from the logic implemented when models are chained and to keep consistency with their associated metadata records as stored in the catalog of available models (DRIHM Model Catalogue, 2014).

One file will be given for each geo-temporal feature type structure represented, in this case a grid series. This file can contain data from more than one parameter. If two different grid series are represented then they must be split into two separate netCDF-CF 1.6 files. For example:

- A meteorological model producing a grid series for the parameter "lwe_thickness_of_precipitation_amount" across a single grid will produce a single netCDF-CF 1.6 file.
- A meteorological model producing two grid series for the parameter "lwe_thickness_of_precipitation_amount" across two different grids will produce two NetCDF-CF 1.6 files, one for each grid.
- A meteorological model producing a grid series for the parameters "lwe_thickness_of_stratiform_precipitation_amount" and "lwe_thickness_of_convective_precipitation_amount" across a single grid will produce a single NetCDF-CF 1.6 file.
- A meteorological model producing two grid series for the parameters "lwe_thickness_of_stratiform_precipitation_amount" and "lwe_thickness_of_convective_precipitation_amount" across two different grids will produce two NetCDF-CF 1.6 files, one for each grid, each containing data for both parameters.

Although this limitation is not imposed by the NetCDF-CF 1.6 standard, it has been included so that unique files can be identified for transfer across interfaces by passing just the file and parameter names.

Controlled vocabularies for parameter (phenomena) names and units are used at all interfaces between independent sources of data and models. If these vocabularies are not adopted by the native code of each source then adaptors are used to undertake the vocabulary cross-walks as indicated in Figure 2. The primary function of the P-Interface is to pass precipitation data to downstream models to drive the flash flooding use case. However, in order to extend the approach to a wider variety of use cases, a selection of meteorological outputs is offered in the P-Interface files. The idea is to bring together the typical parameters which would be used by other models and to communicate them to nonmeteorologists in a way that the experts inheriting this data from the meteorologists would understand.

Many meteorological models use the Climate and Forecasting Standard Names controlled vocabulary for parameter names and units (CF Standard Names, 2003), a directory of around 2,500 parameters from meteorology, many referring to chemicals occurring in the atmosphere. As such this vocabulary has been adopted directly for this interface, although it is noted that CSDMS standard names (CSDMS Standard Names, 2013; Peckham, 2014) now covers a broader range of parameters including from fields outside meteorology such as surface dynamics and concentrates on ease of parsability.

The set of parameters identified and offered across the P-Interface is given in Table 1. Certain of these parameters are calculated at the surface, others at 2 m above the surface and others at 10 m above the surface. The vertical positions were taken from those advised to be "standard" or typical by the meteorological community. The important parameter "lwe_thickness_of_precipitation_amount" was taken from its definition as "lwe_thickness_of_stratiform_precipitation_amount" + "lwe_thickness_of_convective_precipitation_amount" in CF Standard Names.

This collection includes the usual typical meteorological parameters such as precipitation, wind, air temperature, humidity, and air pressure. The presence of a mature set of metadata standards such as the Climate and Forecasting conventions was pivotal in the process of identifying such a set to be relevant across a variety of numerical models. As a result, the meteorological semantic issues were minimal, communication clear, and definitions agreed and common. Fixing the vertical dimension to that which would commonly be required in typical output, such as taking data from parameter "lwe_thickness_of_precipitation_amount" at the surface only or parameter "air_temperature" at 2 m above the surface allows results for each parameter to be expressed as a two-dimensional grid series rather than the native three-dimensional output. This is simpler to process and

TABLE 1. Base Set of Meteorological Parameters Available to Downstream Model Interfaces.

Standard Parameter (from CF Standard Names)	Level	Unit
lwe_thickness_of_precipitation_amount (lwe_thickness_of_stratiform_precipitation_amount + lwe_thickness_of_convective_precipitation_amount)	Surface	m
lwe_thickness_of_stratiform_precipitation_amount	Surface	m
lwe_thickness_of_convective_precipitation_amount	Surface	m
air_temperature	2 m	K
specific_humidity	2 m	1
surface_net_downward_longwave_flux	Surface	W/m^2
eastward_wind	10 m	m/s
northward_wind	10 m	m/s
surface_air_pressure	Surface	Pa

results in considerably less data being transferred at interfaces. This strategy will not apply universally since three-dimensional results (or two-dimensional results at different vertical levels) will often be required. It is also important to note that the vertical levels indicated do not necessarily correspond to the levels in the three-dimensional model grid; they need to be derived as required.

The Q-Interface: WaterML2.0 and OpenMI

The Q-Interface passes flow data (hydrographs) from hydrologic data sources—either sensor data or numerical models—into hydraulic models. The file-based point series hydrograph is represented in WaterML2.0 format and acts as an upstream boundary condition for hydraulic models. As such, the Q-Interface is the second downstream, file-based interface bridging between two different numerical model types.

To illustrate how the Q-Interface works, we consider ingesting such WaterML2.0 data into an OpenMI composition. This is essentially a juxtaposition of two different standards, devised for two different purposes, both of which should be applicable to play an important role in the flash flooding use case explored here. WaterML2 is a file-based, xml encoded, standard devised for storing point-series data (data stored against a single point in 2D space which varies only with time, such as readings from a rain gauge) (OGC WaterML2.0, 2012). OpenMI allows data to be passed between numerical models (OGC OpenMI 2.0, 2014). It is a software component interface for the computational core (the engine) of the numerical model. Model engines are designed or modified to be “OpenMI Compliant,” thus enabling their inclusion in OpenMI integrated compositions. OpenMI has been designed to allow two-way exchange of data between compliant components as they run. This would typically occur between two simultaneously running, time stepping model components which would send and/or receive data at specific time steps as they proceed through their respective time intervals. In this way, the two model components can influence the results produced by the other. Both WaterML2 and OpenMI are recognized international standards which originate from the water domain with WaterML2 developed by a team including members of the Open Geospatial Consortium (OGC) Hydrology Domain Working Group (DWG) and OpenMI through the OpenMI Association (OA).

This interaction between standards was achieved using HR Wallingford’s FluidEarth implementation of OpenMI 2.0 (Harpham *et al.*, 2014) and the Consortium of Universities for the Advancement of Hydrological Sciences Incorporated (CUAHSI) Hydrologic Information System (Tarboton *et al.*, 2009), also used by Peckham and Goodall (2013) to demonstrate interoperability with CSDMS (Peckham *et al.*, 2013). This approach allowed the two standards to interoperate whilst also introducing the live dynamic offered by a web service. This aspect looks toward situations where the architecture is used by live flood forecasting systems to accurately simulate a flood, which is in progress whilst hydrographs are being registered by in situ instrumentation and rendered into WaterML2.0. A simple experiment was conducted with custom built components which was then integrated into the modeling chain.

FluidEarth consists of two important tools:

- SDK: a software development kit (SDK) allowing model developers to more easily make their (time-stepping) model components OpenMI compliant, that is, to use the OpenMI interface definitions as provided in the reference implementation of OpenMI 2.0 (OpenMI Association, 2013). The result of this process is the creation of an “OpenMI Component” consisting of the model code as originally written (perhaps with minor modifications such as conformance to the “Initialise-Timestep-Finalise” structure) together with an OpenMI wrapper. This wrapper enables input and output exchange items, as links, to be connected to other OpenMI components.
- Pipistrelle: a graphical user interface (GUI) and underlying functionality, which allows OpenMI components to be assembled into linked compositions and executed. For example, an OpenMI component modeling catchment drainage may output river flow to another OpenMI component,

which uses this flow as an upstream boundary condition to model the flow in a river reach.

The CUAHSI Hydrologic Information System (HIS) is an online distributed system to support the sharing of hydrologic data from multiple repositories and databases (Tarboton *et al.*, 2009). For the purposes of this experiment, point time series data was inserted into the CUAHSI HIS system and offered through web services in WaterML2 format. It is possible to query this system via http and extract WaterML2 files, using certain arguments in the http "query" string.

The Model MAP

When considering standards-based infrastructures for running numerical models and accessing the supporting data, new numerical models can be written to be directly compliant with the standards incorporated. If the infrastructure is to include existing models, then these must be made compliant to the necessary level. The DRIHM infrastructure is exclusively populated by legacy models, ranging from those common to their scientific domains with long development histories and large user bases to research standard code, which has been iterated many times at universities. In order to incorporate such a wide variety of models, a simple, gateway concept for numerical model compatibility was derived. Adherence to this would make a model compatible for implementation on the infrastructure and also point toward future, more formal standardization. The concept, called a Model MAP, is outlined in more detail in Harpham *et al.* (2015) and is a collection of established model coupling concepts brought together by the DRIHM project in a specific implementation. It contains both managerial and technical elements setting out a straightforward set of requirements so that scientists without any specialization in informatics can prepare their models for compatibility with eInfrastructures (such as DRIHM) and further to formal standards such as OpenMI and the Basic Model Interface. The Model MAP is summarized as follows:

M—**Metadata**, Documentation and Licence: Each model must be supplied with metadata according to a given standard, appropriate documentation and a licence for users to use it.

A—**Adaptors** (or bridges) must be provided, which translate the model inputs and outputs to and from common standards.

P—**Portability**. Each model must be made portable, that is, not tied strongly to local infrastructure.

Each numerical model—as set up to simulate for a given time and place—must be supplied with a package including a metadata record, documentation, and a licence for its use. The metadata record has been derived from sources including reference to a study of five model catalogs conducted by Zaslavsky *et al.* (2014) and by comparison with the Component-Based Water Resources Model Ontology from Elag and Goodall (2013). The metadata record must include the information included in Table 2.

Each model provided on DRIHM is accompanied by a metadata record, an extended ISO19139 dataset (Harpham and Danovaro, 2015) and stored in an accompanying catalog (DRIHM Model Catalogue, 2014).

Each model must also be supplied with a licence which permits its usage on the infrastructure. For compatibility to the DRIHM eInfrastructure, an open source licence is preferred but not demanded. Documentation must also be supplied that fully supports the usage of the model. This is not intended to overlap with existing documentation, rather existing documentation should be referenced and available. This includes installation instructions, usage, background information, and a description of any changes, which have been made to adhere to the Model MAP itself. The documentation requirements have been devised to be as simple and lightweight as possible whilst still achieving fitness for purpose. Ideally, URI references to the licence and documentation should be included in the metadata record.

Inputs to and outputs from each numerical model must, if they are intended to be coupled or passed to other models on the infrastructure, adhere to specific

TABLE 2. Model MAP Metadata for Model Instances.

Category	Elements
High level summary	Title, citation date, abstract, descriptive keywords, development level (see Argent, 2004)
Custodian	Online resource, individual name, organization name, contact address
Technical	Programming language, source code URI, executable URI, supported platform, number of processors, typical run time, memory requirements
Coupling	Supported model standard, coordinate reference system, geographic bounding box, spatial dimension; *For each input and output* Feature type, format, parameter name, parameter unit, geographic bounding box, time start, time end, maximum time step, minimum time step, time step category

standards. These may be file-based or in-memory. The assumption here is that appropriate target standards exist and that they are sufficiently strongly typed to achieve the required level of interoperability out-of-the-box. In the case of the DRIHM eInfrastructure, further specification was necessary when dealing with the interfaces across scientific domains at the P- and Q-Interfaces.

Each numerical model must be portable, that is, it must be possible to run the model on different (albeit equivalent) infrastructures from that on which it was created. The idea is to avoid having to hear the remark: “well, it works on my machine….” A set of good practices and housekeeping are sufficient to achieve this: no expectation of any libraries other than native system libraries; provision of all binaries and libraries it depends on in addition to the native system libraries; no assumption of full directory structure for all dependent files; use of environment variables to denote full paths; every necessary file (excepting files created on model setup) to be given a unique version number identified in an associated inventory.

The DRIHM Portal

The DRIHM portal (Danovaro *et al.*, 2014) is a science gateway supporting hydrometeorological researchers in experiment configuration and execution. It is constructed around the model architecture depicted in Figure 2, the models being the nodes and the directed connecting arrows representing the link between two models sharing the same interface. Each possible simulation chain is a path on the directed graph, thus the selection of a single model or a complex chain (e.g., WRF-NMM, RainFARM, RIBS and Delft3D), defines valid chains supported. This is enabled by a Model MAP for each of the numerical models featured including the standardized interfaces and data conversion adaptors (or bridges) (Harpham *et al.*, 2015).

The portal has been developed, using the gUSE/WS-PGRADE science gateway toolkit (Balasko *et al.*, 2013). It exposes user-friendly web-based interfaces that let the user chain the desired models, specify parameters and submit jobs. The main functional features of the portal are as follows: the management of model instances (which can be uploaded and configured by a restricted set of users); the experiment configuration (i.e., definition of the model instances involved in a simulation chain); and the model configuration (i.e., fine-tuning of the model instance parameters). Model configuration takes into account inter-model constraints, such as coherence of spatial and temporal domains. To ensure validity of interfaces down the model chain, two complementary techniques have been employed: unified storage of experiment parameters among model user interfaces (UIs), so that the generated configurations are shared among models and a set of metadata files as given in the model MAP (Harpham and Danovaro, 2015). Before experiment execution, consistency can be checked by comparing this metadata.

Given a valid model chain configuration, one of the chief functions of the portal is to trigger the execution of the numerical simulations. Each model is executed on the most suitable resource. These resources are not proprietary or predefined, but mainly leverage the existing European e-Infrastructures ecosystem, including the European Grid Infrastructure (EGI) (https://www.egi.eu) and Partnership for Advanced Computing in Europe (PRACE) (http://www.prace-ri.eu). Use of dedicated nodes to provide models with specific constraints, such as dependence on certain libraries or Operating Systems, is also included, together with a data repository to store or provide large datasets. In particular, the preprocessing of meteorological simulations is performed on a dedicated cluster (via web services), meteorological simulations are executed on high performance computing (HPC) clusters providing hundreds of cores, hydrological models are executed on high throughput resources (HTC) on the grid and hydraulic models are executed using Windows virtual machines. Access to HPC and HTC resources relies on the Open Grid Forum (OGF) Basic Execution Service (BES)—resources are accessed on EGI or PRACE depending on user’s permissions. Hydraulic simulations are executed on cloud resources hosted on the EGI Federated Cloud. This is performed by the Distributed Computing Infrastructure (DCI) Bridge, a specific component of gUSE (Kozlovszky *et al.*, 2014), which is able to manage this variety of computing environments.

Intermediate and final simulation results are hosted on a dedicated repository for data analysis and inspection; the latter tasks can be accomplished using ad hoc services available on the portal.

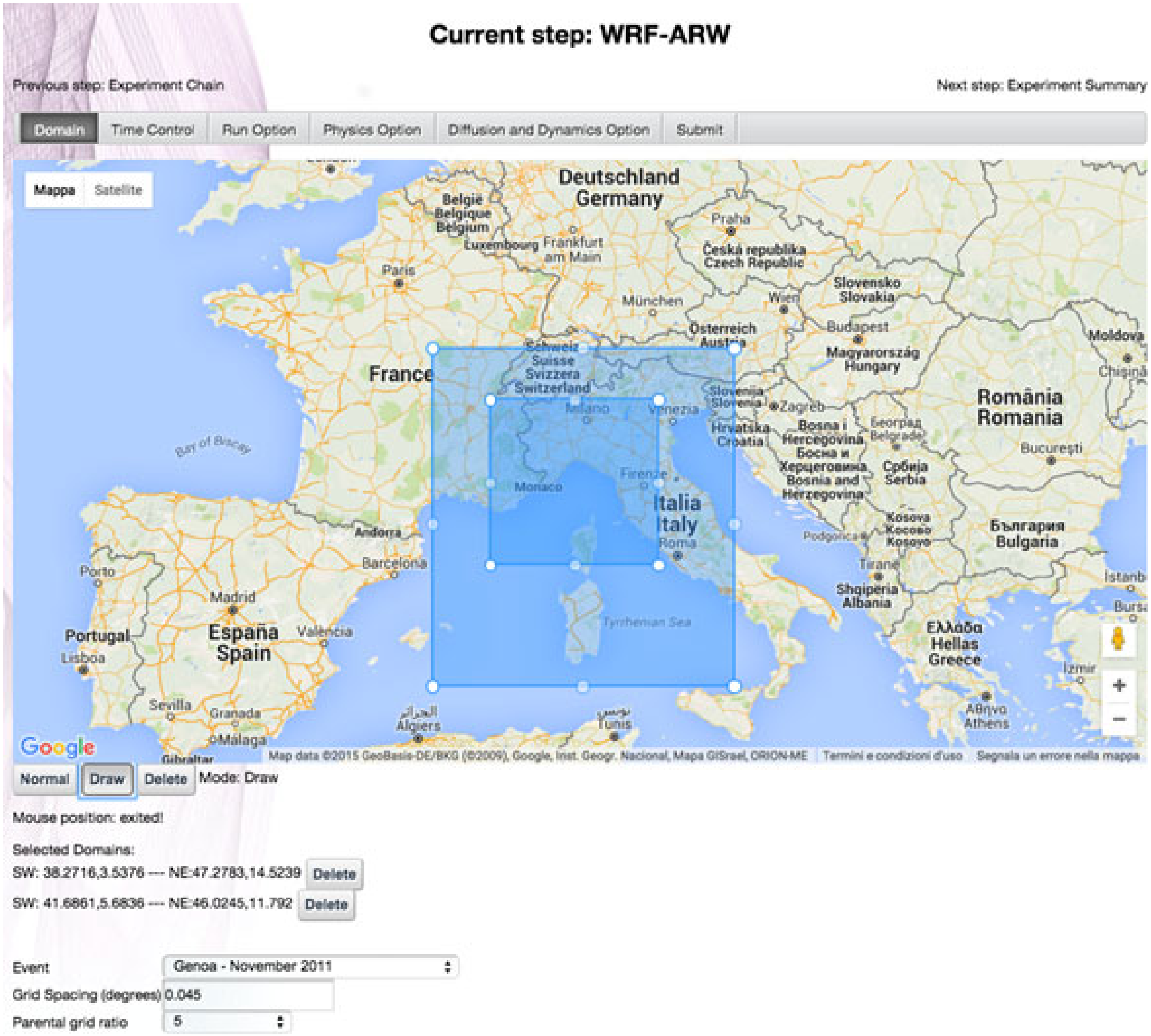
RESULTS AND DISCUSSION

Severe flash flood events in Genoa, Italy in 2011 and 2014 were taken as “critical case” events and used to test the principles behind the DRIHM eInfrastructure. Genoa lies on a narrow strip of land between the Tyrrhenian sea and the Apennine mountains. On November 4, 2011 about 450 mm of rain—a third of the average annual rainfall—fell in six hours. Six people were killed and the raging waters uprooted trees, swept cars away, shattered shops, and flooded the town center. This was the worst disaster in the town since October 7, 1970 when a similar flash flood killed 25 people. Just under three years after the flood event in 2011, it was repeated on October 9-10, 2014. On this occasion one person was killed.

The DRIHM numerical modeling architecture allows these events to be simulated using a variety of numerical models from meteorology, hydrology, and hydraulics. Model MAPs allow each of the numerical models from Figure 2 to be incorporated into model chains across the P- and Q-Interfaces. The metadata supporting each model instance allows workflow threads through these models to be validated to a certain level of detail.

We consider one such workflow thread, using WRF-ARW, RIBS, and the Hydraulic OpenMI composition shown in Figure 10. The WRF-ARW model instance fully adopts the two domains (5 and 1 km, Figure 3) setup and related physical parameterization choices described in Fiori *et al.* (2014).

The RIBS hydrological model sits in the middle of this model chain taking rainfall from the meteorological model (WRF-ARW) simulating river catchment drainage and providing hydrographs to drive the


Current step: WRF-ARW

Previous step: Experiment Chain

Next step: Experiment Summary

Mode: Draw

Mouse position: exited!

Selected Domains:

SW: 38.2716,3.5376 --- NE:47.2783,14.5239 Delete

SW: 41.6861,5.6836 --- NE:46.0245,11.792 Delete

Event: Genoa - November 2011

Grid Spacing (degrees): 0.045

Parental grid ratio: 5

FIGURE 3. WRF-ARW (5 and 1 km) Domains Setup via the DRIHM Portal.

hydraulic composition. The meteorological data is represented as a grid series and the hydrographs as point series. An instance of this model was created to study Bisagno River flash flood events in Genoa, Italy in 2011 and 2014. The overall RIBS Model MAP is represented in Figure 4.

The completed metadata file for the instance set up for Bisagno basin to support the Genoa flood event use case is given here in its xml encoding: http://drihmcatalogue.fluidearth.net/geonetwork/srv/eng/xml_iso19139?id=18. The human readable version can be found in the DRIHM Catalogue (DRIHM Model Catalogue, 2014) by entering "RIBS" into the simple search. Documentation is accessible in the RIBS section of the DRIHM Online Support Center: http://www.drihm.eu/index.php/support-centre/drihm-components/model-section/ribs. The documentation includes a technical description and user manual together with source code and the adaptors to the P- and Q-Interfaces. There are also publications, tutorials, and other training material. RIBS is licenced under a generic Berkeley Software Distribution (BSD) licence.

The package includes an adaptor (or bridge) to receive from the P-Interface. It takes as input the netCDF-CF file produced by meteorological models and produces the rainfall input files required by RIBS in RIBS-raster format. Gridded rainfall is read from the standard Climate and Forecasting variable "lwe_thickness_of_precipitation_amount." In particular, it follows the time reference described in the netCDF file (time origin and time units) and performs a spatial interpolation of the rainfall field to obtain rainfall intensity, using an algorithm based on an inverse distance weighted average of the rainfall in the grid cells of the netCDF file located within a radius of influence (see Appendix VII of DRIHM Consortium D6.2, 2015). The package also includes an adaptor (or bridge) to provide to the Q-Interface. RIBS output consists of a set of series composed of a time label (in RIBS format) and pairs of values (hydrograph in m^3/s and average rainfall over the basin rainfall in mm). This component takes these final hydrograph(s) produced in the RIBS simulation and produces discharge time series in WaterML2.0 format.

The complete model package (including the adaptors) obeys the portability conditions and is able to run on equivalent infrastructures through shell scripts incorporating a set of environment variables. Key metadata elements for the input to RIBS and the two outputs used to drive the hydraulic composition for the flood event of November 4, 2011 are given in Figure 5.

In order to take the output from RIBS and pass it through the Q-Interface, two OpenMI components were constructed to access WaterML2 files, interpret them, and prove their usage in an OpenMI context. Firstly, a "WaterML Client Service" capable of (1) retrieving WaterML files, either from a locally stored file or via the CUAHSI-HIS service where the project data was held and (2) of reading this file and passing the outputs as OpenMI output exchange items to other OpenMI components. Secondly, an "Hrw Locum Component" which would (1) act as an example OpenMI component to receive data across a link, (2) to prove the passage of the data by writing out a text file and (3) to be linked to a trigger to control the composition (data is "pulled" between components with ultimate control pulling from the trigger). Figure 6 shows this test composition assembled in Pipistrelle.

The WaterML Client Service component has a one way link to the Hrw Locum Component, given to be the trigger, which controls the composition time window and initializes time steps from downstream components. This is denoted by a small OpenMI circular logo. The bold arrow between the WaterML Client Service and the Hrw Locum Component indicates that an adaptor is used to perform necessary modifications to the data stream (such as spatial interpolation or unit conversion) between components. Associated with the WaterML Client Service component is an "omi" file defining, amongst other

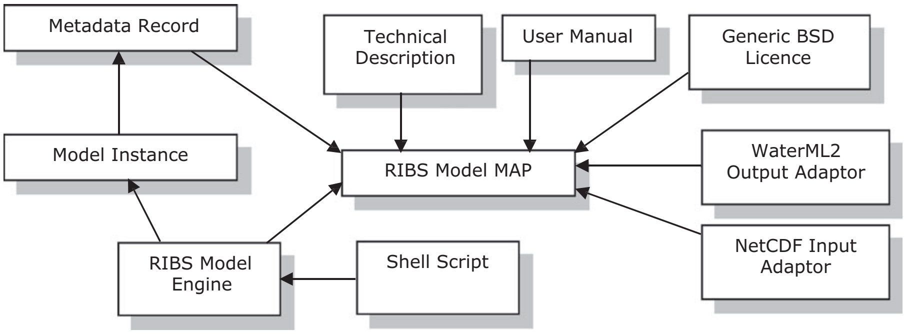


FIGURE 4. The RIBS Model MAP.

```
Input: Precipitation
Mandatory: True
Parameter: lwe_thickness_of_precipitation_amount (units m)
Start time: 2011-11-04T01:00:00+01:00
End time: 2011-11-05T12:00:00+01:00
Maximum timestep: 1 hour
Minimum timestep: 10 minutes
Bounding box (EPSG:4326): 8.88,44.5; 9.09,44.5; 9.09,44.37; 8.88,44.37

Output: Stadium Hydrograph
Mandatory: True
Parameter: river_discharge (units m$^3$s$^{-1}$)
Start time: 2011-11-04T01:00:00+01:00
End time: 2011-11-05T12:00:00+01:00
Maximum timestep: 30 minutes
Minimum timestep: 30 minutes
Bounding box (EPSG:4326): 8.948,44.417; 8.950,44.417; 8.950,44.419; 8.948,44.419

Output: Fereggiano Hydrograph
Mandatory: True
Parameter: river_discharge (units m$^3$s$^{-1}$)
Start time: 2011-11-04T01:00:00+01:00
End time: 2011-11-05T12:00:00+01:00
Maximum timestep: 30 minutes
Minimum timestep: 30 minutes
Bounding box (EPSG:4326): 8.963,44.416; 8.965,44.416; 8.965,44.418; 8.963,44.418
```

FIGURE 5. Key Metadata Elements from the Coupled Input to and Outputs from the Genoa Instance of the RIBS Hydrological Model.

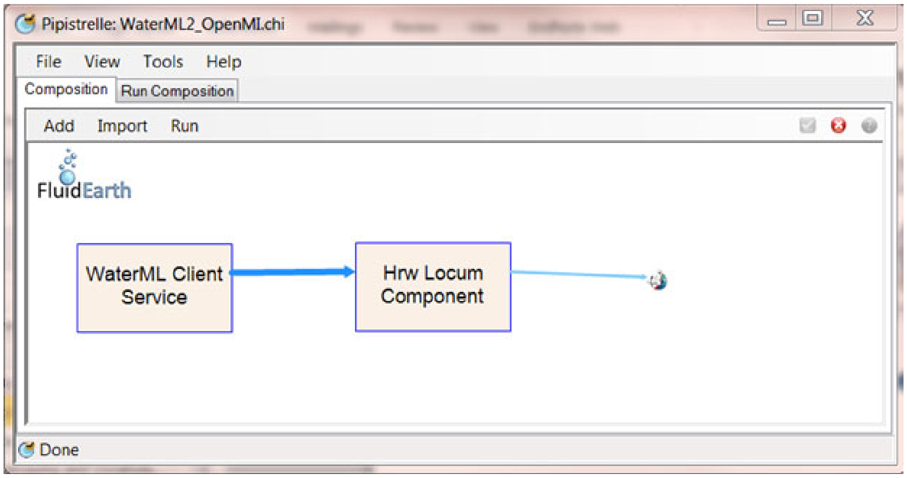


FIGURE 6. OpenMI Composition for Ingesting WaterML2.0 Files.

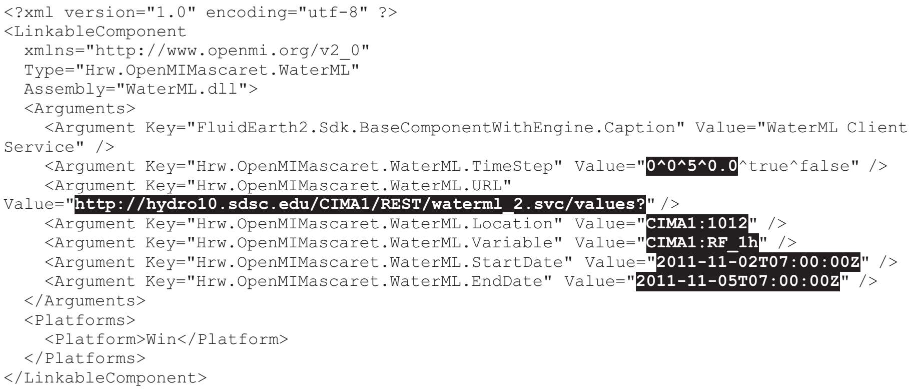
things, the arguments to be used in the query to CUAHSI-HIS. Figure 7 gives this file with these arguments highlighted.

Following the creation of firewall rules to allow access to the CUAHSI-HIS web service, the composition ran on a local laptop. Pipistrelle creates a log file and shows this to the user as the composition completes. Figure 8 shows the last lines of the log file for this composition as displayed with Pipistrelle indicating successful completion of the composition run. The penultimate line indicates the elapsed time of the composition, in this case over 49 s.

The CUAHSI-HIS service issues WaterML2 files which are read by the WaterML2 Client Service and written by the Locum component to csv files. Figure 9 gives two screenshots of this csv file displayed in a text editor. The first shows the top of the file giving column titles with numerous 0 returns and the second, later in the file where some nonzero data values appear.

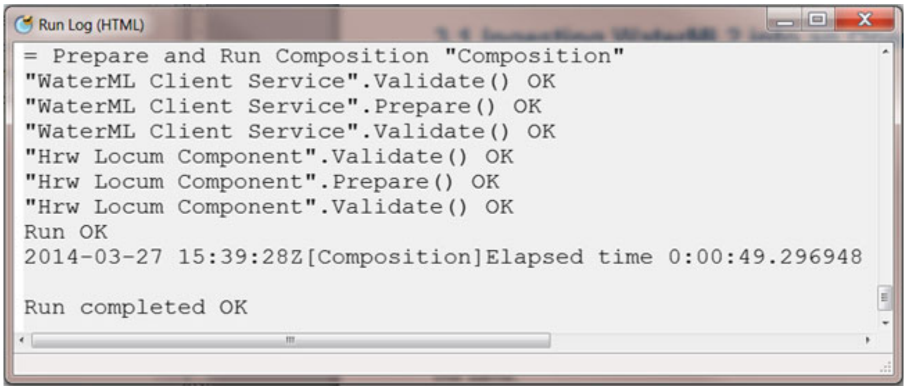
As such, the development of these two OpenMI components demonstrates that it is possible to read a WaterML2.0 file into an OpenMI composition. This is not surprising since both standards deal in time-stepping data and transferring the file-based data into the OpenMI composition is simply a matter of reformatting. However, when this is applied to real-world modeling compositions and through web services additional configuration is required.

The philosophy behind OpenMI is for data to be exchanged in memory with a particular ability to exchange data both ways between model components as they run. This allows each to influence the results



```
<?xml version="1.0" encoding="utf-8" ?>
<LinkableComponent
  xmlns="http://www.openmi.org/v2_0"
  Type="Hrw.OpenMIMascaret.WaterML"
  Assembly="WaterML.dll">
  <Arguments>
    <Argument Key="FluidEarth2.Sdk.BaseComponentWithEngine.Caption" Value="WaterML Client
Service" />
    <Argument Key="Hrw.OpenMIMascaret.WaterML.TimeStep" Value="0^0^5^0.0^true^false" />
    <Argument Key="Hrw.OpenMIMascaret.WaterML.URL"
Value="http://hydro10.sdsc.edu/CIMA1/REST/waterml_2.svc/values?" />
    <Argument Key="Hrw.OpenMIMascaret.WaterML.Location" Value="CIMA1:1012" />
    <Argument Key="Hrw.OpenMIMascaret.WaterML.Variable" Value="CIMA1:RF_1h" />
    <Argument Key="Hrw.OpenMIMascaret.WaterML.StartDate" Value="2011-11-02T07:00:00Z" />
    <Argument Key="Hrw.OpenMIMascaret.WaterML.EndDate" Value="2011-11-05T07:00:00Z" />
  </Arguments>
  <Platforms>
    <Platform>Win</Platform>
  </Platforms>
</LinkableComponent>
```

FIGURE 7. WaterML Client Service omi File.



```
= Prepare and Run Composition "Composition"
"WaterML Client Service".Validate() OK
"WaterML Client Service".Prepare() OK
"WaterML Client Service".Validate() OK
"Hrw Locum Component".Validate() OK
"Hrw Locum Component".Prepare() OK
"Hrw Locum Component".Validate() OK
Run OK
2014-03-27 15:39:28Z[Composition]Elapsed time 0:00:49.296948

Run completed OK
```

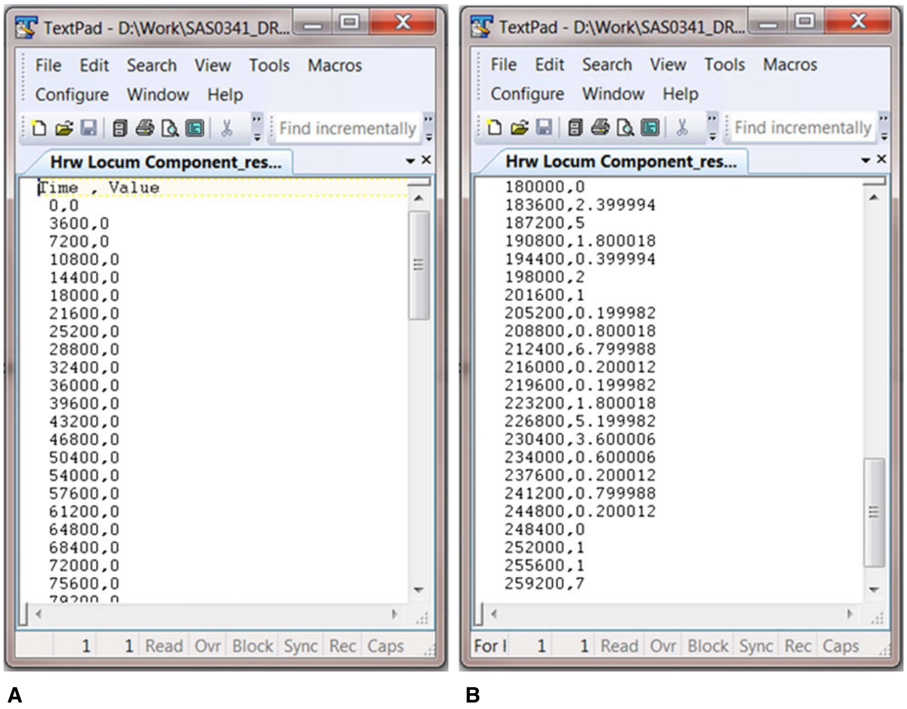
FIGURE 8. WaterML2 Composition log File.

of the other. As each respective component clocks run through their time horizons, data is provided piecemeal, on demand as components request it from each other. In many ways, this philosophy lies counter to that of drawing data files from web services and especially when a large file is generated, mostly by markup, as is often the case with xml. Ideally, the OpenMI component would request data multiple times, as it is required by the pulling component. This would mean making multiple separate requests to the web service and then each time passing, for just a few values, a file containing mostly markup. The performance hit would be considerable compared to the strategy used in this experiment which was to request the whole data file once containing all data required by the composition time horizon, at the start of the composition, pass it once and then draw data from it. Even this approach resulted in a run time of over 49 s, which is attributed to both the web service request (essentially running a query and dynamically creating a large xml file) and to passing the xml file across the Internet.

However, accepting the performance limitations of web services and large xml files, a switch was included in future implementations to allow single up-front requests and multiple requests to be made by the WaterML Client Service, which can also use locally stored files. The date and time arguments used in the http query string and illustrated in Figure 7 were hard-coded into the omi file through the Pipistrelle GUI by the modeler. Notwithstanding the fact that modelers often prefer to check all data sources as part of the composition assembly and have more control over data coming from external sources, a better, more flexible, and generic approach is to allow these arguments to be passed to the component as data is required. Indeed, any automated or semi-automated model chaining service incorporating these components would need this flexibility.

Returning to the model chain used to study the flooding in Genoa, the WaterML Client Service component was then used to read WaterML2.0 files across the Q-Interface (Figure 2) as part of a full OpenMI composition developed by HR Wallingford to simulate flash flood events. Figure 10 shows this composition in the Pipistrelle user interface.

In addition to the WaterML Client, the composition incorporates:


TextPad - D:\Work\SAS0341_DR...
File Edit Search View Tools Macros
Configure Window Help
Find incrementally
Hrw Locum Component_res...
1 1 Read Ovr Block Sync Rec Caps
For I 1 1 Read Ovr Block Sync Rec Caps

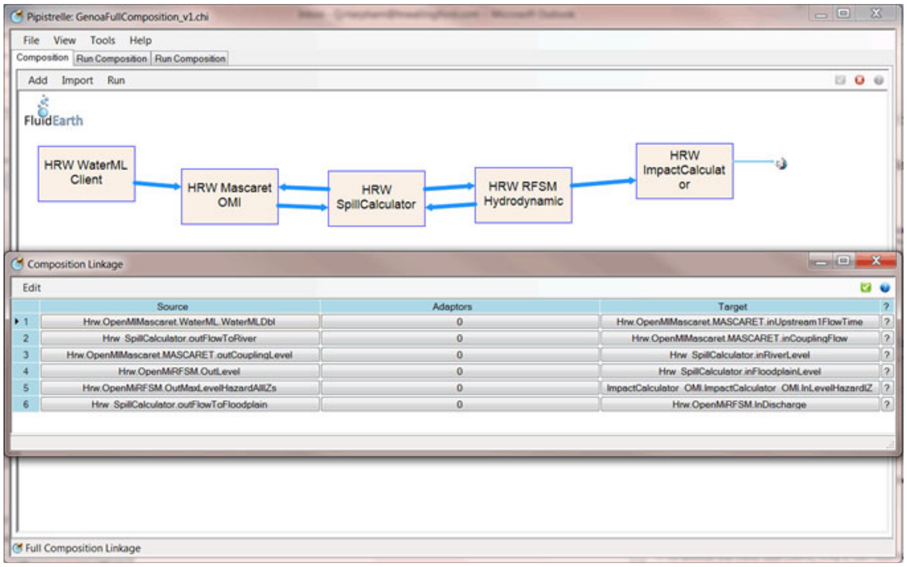
A

```
Time , Value
 0,0
 3600,0
 7200,0
 10800,0
 14400,0
 18000,0
 21600,0
 25200,0
 28800,0
 32400,0
 36000,0
 39600,0
 43200,0
 46800,0
 50400,0
 54000,0
 57600,0
 61200,0
 64800,0
 68400,0
 72000,0
 75600,0
 79200,0
```

B

```
180000,0
183600,2.399994
187200,5
190800,1.800018
194400,0.399994
198000,2
201600,1
205200,0.199982
208800,0.800018
212400,6.799988
216000,0.200012
219600,0.199982
223200,1.800018
226800,5.199982
230400,3.600006
234000,0.600006
237600,0.200012
241200,0.799988
244800,0.200012
248400,0
252000,1
255600,1
259200,7
```

FIGURE 9. Results Returned from CUAHSI-HIS Written Out as csv.



	Source	Adaptors	Target	?
1	Hrw.OpenMIMascaret.WaterML.WaterMLDbl	0	Hrw.OpenMIMascaret.MASCARET.inUpstream1FlowTime	?
2	Hrw SpillCalculator.outFlowToRiver	0	Hrw.OpenMIMascaret.MASCARET.inCouplingFlow	?
3	Hrw.OpenMIMascaret.MASCARET.outCouplingLevel	0	Hrw SpillCalculator.inRiverLevel	?
4	Hrw.OpenMiRFSM.OutLevel	0	Hrw SpillCalculator.inFloodplainLevel	?
5	Hrw.OpenMiRFSM.OutMaxLevelHazardAllIZs	0	ImpactCalculator OMI.ImpactCalculator OMI.InLevelHazardIZ	?
6	Hrw SpillCalculator.outFlowToFloodplain	0	Hrw.OpenMiRFSM.InDischarge	?

FIGURE 10. Schematic View of the OpenMI Composition Assembled to Study Flows and Impacts of Flash Floods, Shown in the Pipistrelle GUI.

- MASCARET: a one-dimensional open channel hydraulic model capable of simulating subcritical and trans-critical flows in channels and conduits together with a range of hydraulic structures either in the channel or connected to the channel (Goutal and Maurel, 2002; Goutal *et al.*, 2012);
- RFSM: a two-dimensional Rapid Flood Spreading Model, more specifically the EDA variant (Explicit Diffusion wave with Acceleration terms) (Jamieson *et al.*, 2012a,b) calculating floodplain flow using computational elements constructed as irregular polygons around the key topographic features (local crests and depressions);
- Impact Calculator: a tool used to estimate the impact of flooding on (1) buildings and agricultural land with depth-damage curves taken from

sources such as the Flood Hazard Research Centre “Multi Colour Manual” (Penning-Rowsell *et al.*, 2013) and (2) on people (predicted Loss of Life), using the Risk To People method published in the DEFRA report FD2317 (Ramsbottom *et al.*, 2003).

The connection between the MASCARET and RFSM-EDA components is two-way to allow water to pass from the river to the floodplain and vice versa. This is enabled by the Spill Calculator component which receives water levels from the two components and returns the calculated flow to both components. This is illustrated in Figure 10 where the Composition Linkage window is open showing the linkages between all components. Linkages 2, 3, 4, and 6 describe these connections. Essentially, the Spill Calculator is acting as a complex adaptor between MASCARET and RFSM-EDA.

The associated bounding boxes from the RIBS output described by the metadata in Figure 5 are given in Figure 11. The city of Genoa can be seen along the coastal strip between the mountains in the north and the sea to the south. The large plotted rectangle represents the bounding box of the precipitation input required by RIBS. The two smaller rectangles in the enlarged section represent the bounding boxes giving the approximate locations of the outputted Stadium hydrograph (next to the football stadium on the left of the inset map) and the Fereggiano hydrograph (on the right of the inset map).

The RIBS metadata elements from Figure 5 can be matched with their equivalents from the upstream and downstream models (output to input) to provide a level of validation of the model chain ensuring that:

- Mandatory inputs are served by mandatory outputs;
- The parameter name and unit expected matches that provided;
- The time window of the expected input lies inside that provided;
- The maximum time step provided is within a certain tolerance of that expected;
- The bounding box of the expected input lies inside that provided.

Note that the hydrograph outputs are given the parameter name “river_discharge” which does not come from a controlled vocabulary. This is due to the reach of CF Standard Names not extending to hydrology (although the more recently formulated CSDMS Standard Names does cover this). Also, bounding boxes are used since they can describe the approximate position of all possible feature type outputs and are applicable to general searching and plotting.

Hydraulic modeling using the OpenMI composition was performed for the 2011 and 2014 flood events with hydrographs located at Stadium and Fereggiano as driving data. Figure 12 shows the maximum depth throughout the flooded area for a preliminary calibration of the composition which studies the inundation around the railway line. The water travels down the river channel from the top boundary of the figure, past the stadium after which it is joined by water from the Fereggiano. As it attempts to go under the railway line in the center of the figure, inundation occurs first to the west side of the river and later to the east. The water travels through the four road and foot tunnels under the railway and inundates the area to the south.

The flood event in October 2014 was very similar to that of November 2011. Figure 13 shows the maximum hazard plot for the 2014 event, again for a preliminary calibration of the composition which

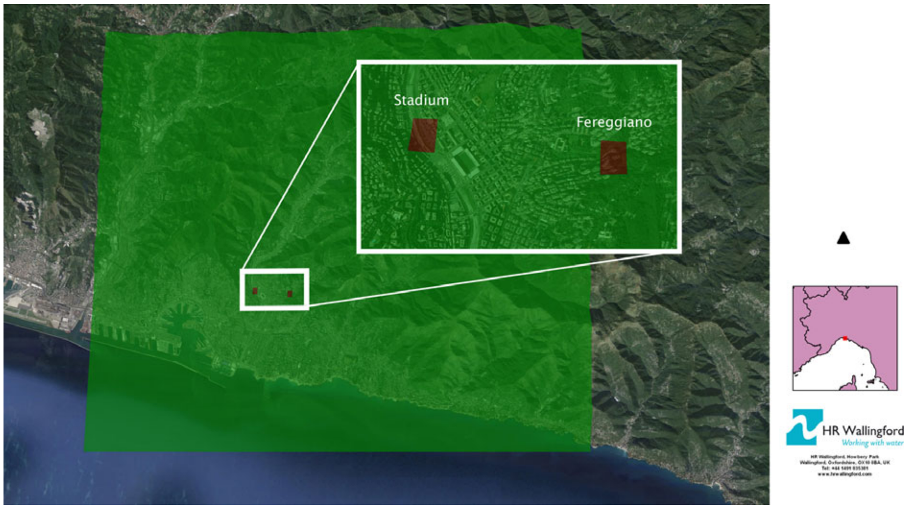


FIGURE 11. Input and Output Bounding Boxes for the RIBS Hydrological Model Set Up to Study Flash Flooding over the City of Genoa.

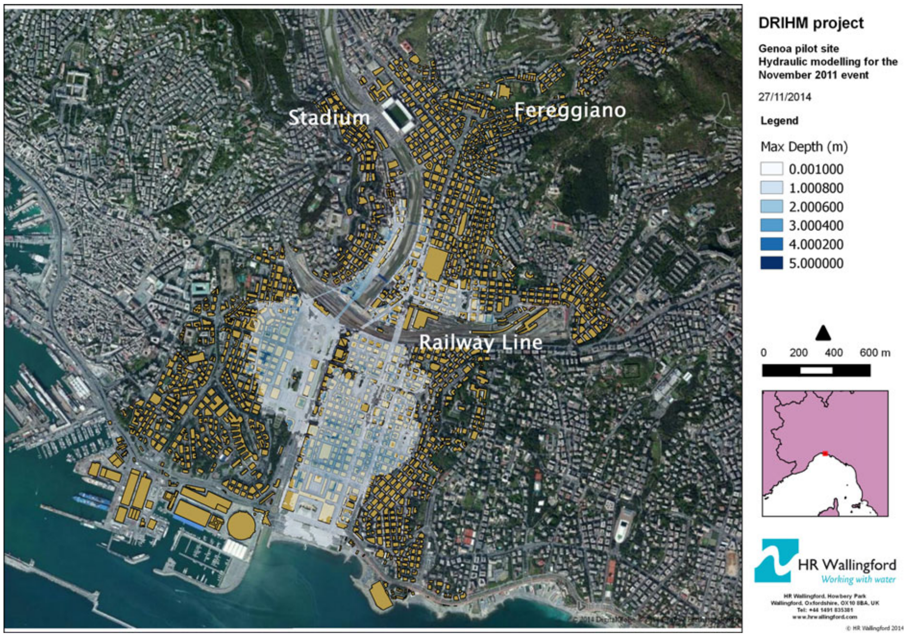


FIGURE 12. Maximum Depth Plot for the Hydraulic Model Composition Studying the Inundation around the Railway Line for the November 2011 Flood Event. Please note, the results are subject to the disclaimer given at the end.

studies the inundation around the railway line. The hazard is defined according to the table given in Figure 14 (see Ramsbottom *et al.*, 2003). It can be seen that the results show the event capable of causing danger to all people (i.e., people of all ages and of any physical strength) across most of the inundated area.

This composition also includes a calculation of damage to buildings. This was performed with a default uniform damage curve derived from sources such as the Flood Hazard Research Centre “Multi Colour Manual” (Penning-Rowsell *et al.*, 2013), rather than specific damage curves related to the particular infrastructure inundated in this case. As such, the damage was calculated at €205 million (2014 prices), agreeing strongly with unconfirmed, anecdotal estimates.

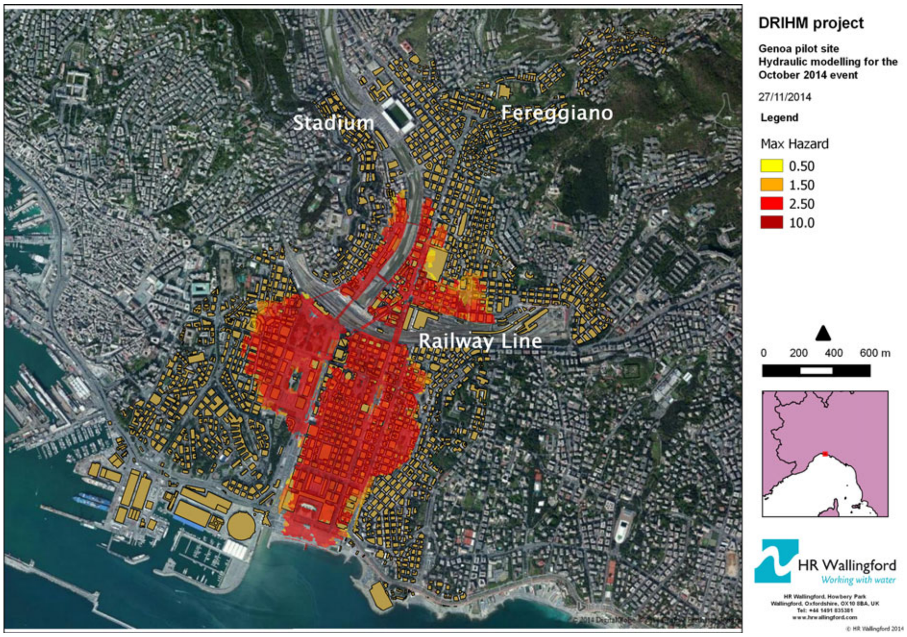
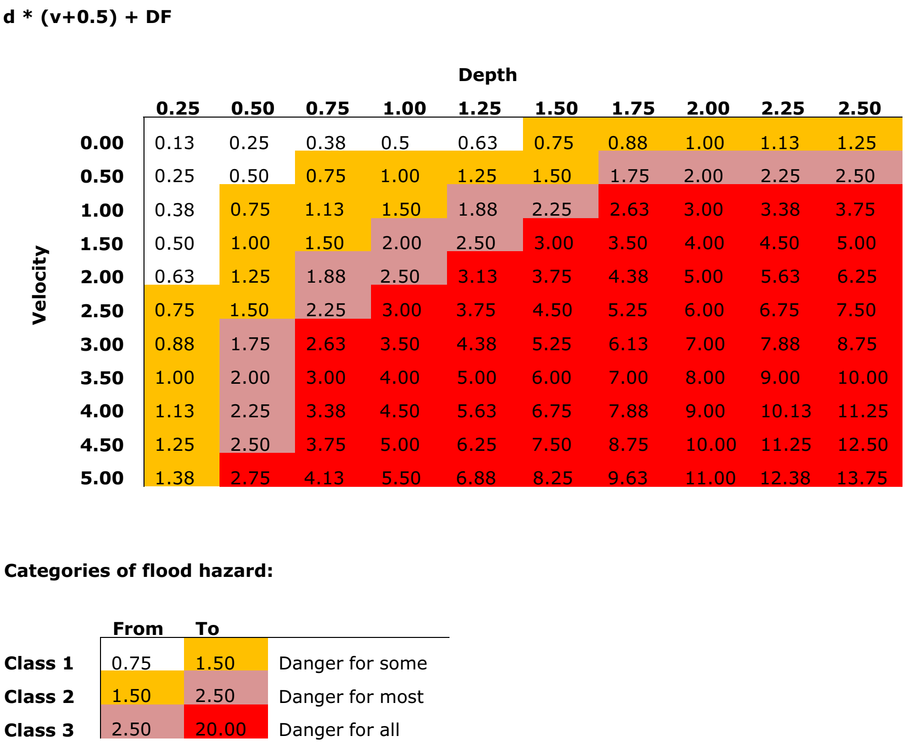


FIGURE 13. Maximum Hazard Plot for the Hydraulic Model Composition Studying the Inundation around the Railway Line for the October 2014 Flood Event. Please note, the results are subject to the disclaimer given at the end.



d * (v+0.5) + DF

	Depth									
Velocity	**0.25**	**0.50**	**0.75**	**1.00**	**1.25**	**1.50**	**1.75**	**2.00**	**2.25**	**2.50**
0.00	0.13	0.25	0.38	0.5	0.63	0.75	0.88	1.00	1.13	1.25
0.50	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
1.00	0.38	0.75	1.13	1.50	1.88	2.25	2.63	3.00	3.38	3.75
1.50	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00
2.00	0.63	1.25	1.88	2.50	3.13	3.75	4.38	5.00	5.63	6.25
2.50	0.75	1.50	2.25	3.00	3.75	4.50	5.25	6.00	6.75	7.50
3.00	0.88	1.75	2.63	3.50	4.38	5.25	6.13	7.00	7.88	8.75
3.50	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
4.00	1.13	2.25	3.38	4.50	5.63	6.75	7.88	9.00	10.13	11.25
4.50	1.25	2.50	3.75	5.00	6.25	7.50	8.75	10.00	11.25	12.50
5.00	1.38	2.75	4.13	5.50	6.88	8.25	9.63	11.00	12.38	13.75

Categories of flood hazard:

	From	To	
Class 1	0.75	1.50	Danger for some
Class 2	1.50	2.50	Danger for most
Class 3	2.50	20.00	Danger for all

FIGURE 14. Hazards as a Function of Water Depth and Velocity (see Ramsbottom *et al.*, 2003).

CONCLUSION

The modeling architecture outlined in this article and summarized in Figure 2 is designed to be interoperable and extensible within modeling domains (meteorology, hydrology, and hydraulics) as well as between modeling domains. This has been achieved, but is limited by the nature of the model output and input: as long as the input/output spatio temporal feature types (e.g., grid-series, point-series) are the same, numerical models can be incorporated into this simple structure. It is then also possible to utilize an ensemble of equivalent models and observational data from each domain—not restricted to those given here—as well as incorporating new domains. The DRIHM portal allows these models to be executed against a variety of resources, enabled by the gUSE science gateway with the interfaces based around the different spatio temporal feature types using different file standards (e.g., NetCDF-CF1.6 and WaterML2.0). These file standards would not allow this level of automation "out-of-the-box" and some local conventions were required.

The metadata structure adopted allows discovery of candidate numerical models for the user together with a preliminary evaluation of each. Even with a comprehensive and assertive review process, this handwritten metadata is capable of validation of interfaces only to a certain level of detail permitting up to semi-automatic formulation of model chains. At the end of the model chain, use of OpenMI 2.0 for the hydraulic modeling enabled one-dimensional and two-dimensional models to exchange data in a two-way connection as the models proceeded through their respective time-steps.

DISCLAIMER

The hydraulic model results presented here reflects preliminary calibration of models and should not be used for any purpose other than that intended, i.e., that of demonstrating the innovations of the DRIHM eInfrastructure. They are preliminary research results and do not necessarily reflect actual results of flooding, either past or future. HR Wallingford will accept no liability whatsoever for any direct, indirect, or consequential loss or damage to any third party who uses information contained in this presentation or source dataset. HR Wallingford retains copyright of all flood results except where the copyright belongs to a third party (as acknowledged or not) and neither the original recipient nor any subsequent user of the flood results can acquire any stake in the said copyright. Any queries should be addressed to: HR Wallingford, Howbery Park, WALLINGFORD, Oxfordshire, OX10 8BA, United Kingdom.

ACKNOWLEDGMENTS

This research was co-funded by the European Commission (EC) 7th Framework Programme DRIHM Project, Grant Number 283568 and DRIHM2US Project, Grant Number 313122. The authors would like to acknowledge the contributions of the European Commission and both project teams. With respect to the content given here, thanks to Paul Cleverley for assisting with the construction of the hydraulic modeling OpenMI composition at HR Wallingford and to Luis Garrote of the Universidad Politecnica de Madrid for the original RIBS Model MAP.

LITERATURE CITED

Argent, R., 2004. An Overview of Model Integration for Environmental Applications—Components, Frameworks and Semantics. Environmental Modelling & Software 19(3):219-234.

Balasko, A., Z. Farkas, and P. Kacsuk, 2013. Building Science Gateway By Utilizing the Generic WS-PGRADE/GUSE Workflow System. Computer Science 14(2):307-325.

Bergström, S., 1995. The HBV Model. *In*: Computer Models of Watershed Hydrology, V.P. Singh (Editor). Water Resources Publications, Colorado, pp. 443-476, ISBN: 0-918334-91-8.

CF Standard Names, 2003. CF Metadata NetCDF CF Metadata Conventions. http://cf-convention.github.io/index.html, *accessed* April 2014.

CSDMS Basic Model Interface, 2012. CSDMS Basic Model Interface (Version 1.0). http://csdms.colorado.edu/wiki/BMI_Description, *accessed* May 2015.

CSDMS Standard Names, 2013. Standard Name Examples (Version 0.7.1). http://csdms.colorado.edu/wiki/CSN_Searchable_List, *accessed* April 2014.

D'Agostino, D., A. Clematis, A. Galizia, A. Quarati, E. Danovaro, L. Roverelli, G. Zereik, D. Kranzlmuller, M. Schiffers, N. Felde, C. Straube, O. Caumont, E. Richard, L. Garrote, Q. Harpham, B. Jagers, V. Dimitrijevic, L. Dekic, A. Parodi, E. Fiori, and F. Delogu, 2014. The DRIHM Project: A Flexible Approach to Integrate HPC, Grid and Cloud Resources for Hydro-Meteorological Research. *In*: SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE Press, Piscataway, New Jersey, pp. 536-546, DOI: 10.1109/SC.2014.49.

D'Agostino, D., E. Danovaro, A. Clematis, L. Roverelli, G. Zereik, A. Parodi, and A. Galizia, 2015. Lessons Learned Implementing a Science Gateway for Hydro-Meteorological Research. Concurrency and Computation: Practice and Experience, DOI: 10.1002/cpe.3700.

Danovaro, E., L. Roverelli, G. Zereik, A. Galizia, D. D'Agostino, A. Quarati, A. Clematis, F. Delogu, E. Fiori, A. Parodi, C. Straube, N. Felde, Q. Harpham, B. Jagers, L. Garrote, L. Dekic, M. Ivkovic, E. Richard, and O. Caumont, 2014. Setup an Hydro-Meteo Experiment in Minutes: The DRIHM e-Infrastructure for Hydro-Meteorology Research, to be published in the Proceedings of e-Science 2014: 10th IEEE International Conference on e-Science, Guarujá, SP, Brazil, October 20-24, 2014.

DRIHM Consortium D6.2, 2015. Report on Application Services Delivery. http://www.drihm.eu/images/Deliverable/last/drihm-dwp6.2-20150228-4.0-HRW-Report_on_application_services_delivery.pdf, *accessed* May 2015.

DRIHM Model Catalogue, 2014. DRIHM Model Catalogue. http://drihmcatalogue.fluidearth.net/, *accessed* May 2015.

Elag, M. and J.L. Goodall, 2013. An Ontology for Component-Based Models of Water Resource Systems. Water Resources Research 49:5077-5091, DOI: 10.1002/wrcr.20401.

Fiori, E., A. Comellas, L. Molini, N. Rebora, F. Siccardi, D.J. Gochis, S. Tanelli, and A. Parodi, 2014. Analysis and Hindcast Simulations of an Extreme Rainfall Event in the Mediterranean Area: The Genoa 2011 Case. Atmospheric Research 138:13-29, DOI: 10.1016/j.atmosres.2013.10.007.

Garrote, L. and R. Bras, 1995. A Distributed Model for Real-Time Flood Forecasting Using Digital Elevation Models. Journal of Hydrology 167(1–4):279-306, DOI: 10.1016/0022-1694(94)02592-Y.

Giannoni, F., G. Roth, and R. Rudari, 2000. A Semi-Distributed Rainfall-Runoff Model Based on a Geomorphologic Approach. Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere 25(7–8):665-671, DOI: 10.1016/S1464-1909(00)00082-4.

Goutal, N., J.-M. Lacombe, F. Zaoui, and K. El-Kadi-Abderrezzak, 2012. MASCARET: A 1-D Open-Source Software for Flow Hydrodynamic and Water Quality in Open Channel Networks. *In*: River Flow 2012, R. Murillo Muñoz (Editor), Taylor & Francis Group, London, pp. 1169-1174.

Goutal, N. and F. Maurel, 2002. A Finite Volume Solver for 1D Shallow-Water Equations Applied to an Actual River. International Journal for Numerical Methods in Fluids 2002(38):1-19.

Harpham, Q.K., P. Cleverley, D. D'Agostino, A. Galizia, E. Danovaro, F. Delogu, and E. Fiori, 2015. Using a Model MAP to Prepare Hydro-Meteorological Models for Generic Use. Environmental Modelling and Software 73(2015):260-271.

Harpham, Q.K., P. Cleverley, and D. Kelly, 2014. The Fluid Earth 2 Implementation of OpenMI 2.0. Journal of Hydroinformatics 16(4): 890-906.

Harpham, Q.K. and E. Danovaro, 2015. Towards Standard Metadata to Support Models and Interfaces in a Hydro-Meteorological Model Chain. Journal of Hydroinformatics 17(2):260-274, IWA Publishing, DOI: 10.2166/hydro.2014.061.

Hill, C., C. DeLuca, V. Balaji, M. Suarez, and A. da Silva, 2004. The Architecture of the Earth System Modeling Framework. Computing in Science & Engineering 6:18-28.

Jamieson, S., J. Lhomme, G. Wright, and B. Gouldby, 2012a. Highly Efficient 2D Inundation Modeling with Enhanced Diffusion-Wave and Sub-Element Topography. Proceedings of the Institution of Civil Engineers-Water Management 165(10): 581-595.

Jamieson, S., G. Wright, J. Lhomme, and B. Gouldby, 2012b. Validation of a Computationally Efficient 2D Inundation Model on Multiple Scales. Proceedings Floodrisk 2012, Taylor & Francis Group, Rotterdam.

Johnston, J.M., D.J. McGarvey, M.C. Barber, G. Laniak, J. Babendreier, R. Parmar, K. Wolfe, S.R. Kraemer, M. Cyterski, C. Knightes, B. Rashleigh, L. Suarez, and R. Ambrose, 2011. An Integrated Modeling Framework for Performing Environmental Assessments: Application to Ecosystem Services in the Albemarle-Pamlico Basins (NC and VA, USA). Ecological Modelling 222:2471-2484.

Kozlovszky, M., K. Karóczkai, I. Márton, P. Kacsuk, and T. Gottdank, 2014. DCI Bridge: Executing WS-PGRADE Workflows in Distributed Computing Infrastructures. *In*: Science Gateways for Distributed Computing Infrastructures, P. Kacsuk (Editor). Springer International Publishing, Switzerland, pp. 51-67.

Llasat, M.C., M. Llasat-Botija, M.A. Prat, F. Porcú, C. Price, A. Mugnai, K. Lagouvardos, V. Kotroni, D. Katsanos, S. Michaleides, Y. Yair, K. Savvidou, and K. Nicolaides, 2010. High-Impact Floods and Flash Floods in Mediterranean Countries: The FLASH Preliminary Database. Advances in Geosciences 23:47-55. www.adv-geosci.net/23/47/2010/.

Meso-NH website, 2015. Meso-NH Mesoscale Non-Hydrostatic Model 5.1, http://mesonh.aero.obs-mip.fr/mesonh51, *accessed* May 2015.

OGC netCDF, 2013. OGC Network Common Data Form (NetCDF) Standards Suite. http://www.opengeospatial.org/standards/netcdf, *accessed* May, 2015.

OGC OpenMI 2.0, 2014. OGC Open Modelling Interface (OpenMI) Interface Standard, Open Geospatial Consortium Interface Standard. http://www.opengeospatial.org/standards/openmi, *accessed* August 2014.

OGC WaterML 2.0, 2012. OGC WaterML 2.0 Part 1—Timeseries. Open Geospatial Consortium Implementation Standard. http://www.opengeospatial.org/standards/waterml, *accessed* May 2015.

OpenMI Association, 2013. Source Forge OpenMI Project. http://sourceforge.net/projects/openmi/, *accessed* May 2015.

Peckham, S.D., 2014. The CSDMS Standard Names: Cross-Domain Naming Conventions for Describing Process Models, Data Sets and Their Associated Variables. *In*: Proceedings of the 7th International Congress on Environmental Modelling and Software, D.P. Ames, N.W.T. Quinn, and A.E. Rizzoli (Editors). International Environmental Modelling and Software Society (iEMSs), San Diego, California. http://www.iemss.org/society/index.php/iemss-2014-proceedings, *accessed* May 2015.

Peckham, S. and J. Goodall, 2013. Driving Plug-and-Play Models with Data from Web Services: A Demonstration of Interoperability between CSDMS and CUAHSI-HIS. Computers & Geosciences 53:154-161.

Peckham, S., E. Hutton, and D. Norris, 2013. A Component-Based Approach to Integrated Modeling in the Geosciences: The Design of CSDMS. Computers & Geosciences 53:3-12, DOI: 10.1016/j.cageo.2012.04.002.

Penning-Rowsell, E., S. Priest, D. Parker, J. Morris, S. Tunstall, C. Viavattene, J. Chatterton, and D. Owen, 2013. Flood and Coastal Erosion Risk Management: A Manual for Economic Appraisal. Routledge, Abingdon, UK.

Ramsbottom, D., P. Floyd, and E. Penning-Rowsell, 2003. Flood Risks to People Phase 1. R&D Technical Report FD2317, Defra-Flood Management Division, London, ISBN: 1844321355.

Rebora, N., L. Ferraris, J. von Hardenberg, and A. Provenzale, 2006. RainFARM: Rainfall Downscaling by a Filtered Autoregressive Model. Journal of Hydrometeorology 7:724-738, DOI: 10.1175/JHM517.1.

Roelvink, J.A. and G.K.F.M. Van Banning, 1995. Design and Development of DELFT3D and Application to Coastal Morphodynamics. Oceanographic Literature Review 42(11):925, ISSN: 0967-0653

Tarboton, D.G., J.S. Horsburgh, D.R. Maidment, T. Whiteaker, I. Zaslavsky, M. Piasecki, J. Goodall, D. Valentine, and T. Whitenack, 2009. Development of a Community Hydrologic Information System, 18th World IMACS/MODSIM Congress, July 13-17, Cairns, Australia. http://mssanz.org.au/modsim09, *accessed* May 2015.

WRF Website, 2014. The Weather Research and Forecasting Model. http://www.wrf-model.org/index.php, *accessed* May 2015.

Zaslavsky, I., T. Whitenack, and D. Valentine, 2014. Exploring Environmental Model Catalogs. *In*: D.P. Ames, N.W.T. Quinn, and A.E. Rizzoli (Editors), Proceedings of the 7th International Congress on Environmental Modelling and Software, June 15-19, San Diego, California, ISBN: 978-88-9035-744-2.

Appendix X: A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members

This article was published in Environmental Modelling & Software, 84, Harpham, Q., Tozer, N., Cleverley, P., Wyncoll, D. and Cresswell, D., A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members, pp. 482-493, Copyright Elsevier (2016).

Environmental Modelling & Software 84 (2016) 482–493

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A Bayesian method for improving probabilistic wave forecasts by weighting ensemble members

Quillon Harpham*, Nigel Tozer, Paul Cleverley, David Wyncoll, Doug Cresswell

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

ARTICLE INFO

Article history:
Received 29 January 2016
Received in revised form
22 July 2016
Accepted 25 July 2016

Keywords:
Bayesian
Ensemble
Feature types
Forecast
Observations
Waves

ABSTRACT

New innovations are emerging which offer opportunities to improve forecasts of wave conditions. These include probabilistic modelling results, such as those based on an ensemble of multiple predictions which can provide a measure of the uncertainty, and new sources of observational data such as GNSS reflectometry and FerryBoxes, which can be combined with an increased availability of more traditional static sensors. This paper outlines an application of the Bayesian statistical methodology which combines these innovations. The method modifies the probabilities of ensemble wave forecasts based on recent past performance of individual members against a set of observations from various data source types. Each data source is harvested and mapped against a set of spatio-temporal feature types and then used to post-process ensemble model output. A prototype user interface is given with a set of experimental results testing the methodology for a use case covering the English Channel.

Software/data availability

The software development was led by Paul Cleverley (paper co-author). Executable code for the extract, transform and load modules together with the statistical model was written in a windows environment with C# under. Net 4.0 using Visual Studio 2012 with a front end written in VB script by Naveed Hussain. Supporting scripts were written in Python using the Enthought EPD python implementation 7.2.1 (64 bit). Intermediate files were stored using CSV and XML. The relational database management system used was PostgreSQL 8.4 plus POSTGIS extensions. The execution environment was a windows server running Windows Server 2003 Standard Edition.

1. Introduction

Forecasts of wave conditions are required for planning of a wide range of weather sensitive maritime operations from construction and maintenance to decommissioning. There is an ever increasing requirement to minimise downtime in order to reduce costs and demand for increasingly more accurate forecasts is one aspect that can help planners make the most appropriate operational decisions. Two independent sets of innovations are now emerging, which offer new opportunities to improve forecasting services.

Traditional wave forecasts provide a single estimate of conditions with a typical outlook of 5–7 days, giving parameters such as significant wave height, maximum wave height, wave period and direction. Such deterministic forecasts provide limited or no information on the potential uncertainty in a given forecast. Probabilistic forecasts, in contrast, such as those based on an ensemble of multiple predictions, not only extend the range of the forecasts often out to 14 days, but also provide a measure of the uncertainty at any given time-step. With increasing computing power, probabilistic forecasts are becoming increasingly common and will no doubt become the norm.

Alongside the increase in availability of ensemble wave forecasts, new and innovative sources of observational data are emerging. Global Navigation Satellite System (GNSS) reflectometry (see for example Gleason et al., 2005) offers the potential for reflected signals from Global Positioning Satellites (GPS) to be used to interpret phenomena such as sea conditions (e.g. wave mean square slope from which other parameters can be inferred). The sensors must be situated in a low enough orbit to receive these signals and so cannot be geostationary, thereby producing a dataset of observations following the track of the satellite carrying the receiver. Back on the surface of the ocean, products such as the

* Corresponding author.
E-mail address: q.harpham@hrwallingford.co.uk (Q. Harpham).

http://dx.doi.org/10.1016/j.envsoft.2016.07.015
1364-8152/© 2016 Elsevier Ltd. All rights reserved.

Acronyms and abbreviations

ADCP	Acoustic Doppler Current Profiler
AR	Autoregressive
AWAC	Acoustic Wave and Current Profiler
AVHRR	Advanced Very High Resolution Radiometer
CRPS	Continuous Rank Probability Score
CTD	Conductivity, Temperature and Depth
DDM	Delayed Doppler Map
ECMWF	European Centre for Medium-range Weather Forecasts
GNSS	Global Navigation Satellite System
GPS	Global Positioning Satellites
NOC	National Oceanography Centre

“Ferrybox” (Chelsea Technologies 2012, Hydes et al., 2004; Hydes and Dunning, 2005; Dunning and Hand, 2005) allow sensors to be mounted on moving vessels collecting ocean data parameters, from which typical wave parameters (height, period and direction) can be interpreted. Like the GNSS receiver, such technology results in data along a track, but one far more variable and with a potentially much greater density of readings. As well as these moving devices, there are also an increasing number of static sensors producing oceanographic parameters in time series at fixed locations. Many of these static sensors offer real-time data streams (see for example the Channel Coast Observatory (2014)).

The use of ensembles aims to represent the uncertainty in a forecast using a population of individual ensemble members. Ensemble members may be perturbed instances of the same model – either global or downscaled – (e.g. Saetra and Bidlot (2004), Cao et al. (2009), Behrens (2015)), or of different models (e.g. Durrant et al., 2009) or a combination of the two (e.g. Alves et al., 2013). In weather forecasting historically, ensembles have often been focussed on the uncertainty at the tail end of the forecast window. The separation of ensemble members being relatively small at analysis time (e.g. Saetra and Bidlot (2004), Cao et al. (2009)) and growing as the forecasts progresses. Ensemble spread at analysis time can be achieved using Ensemble Transform techniques, e.g. Bunney and Saulter (2015), Alves et al. (2013), which is useful when the focus is on short term uncertainty.

Motivated by the potential spread of Wave Farms to harness the generating potential of the Ocean, Pinson et al. (2012) introduce a methodology for the probabilistic forecasting of wave energy flux. Using meteorological forecasts from the European Centre for Medium-range Weather Forecasts (ECMWF) and a log-Normal assumption for the shape of predictive densities, benchmarked improvements of between 6 and 70% are shown in terms of Continuous Rank Probability Score (CRPS). However, in studying the effectiveness of the spread of results presented by European ensemble wind speed forecasts, Saunders et al. (2014) concluded that “leading ensemble forecasts of European windspeed often represent uncertainty poorly” and, in particular, that “the mis-calibration is worst at shorter lead times and improves at longer forecast lead times”. They also observed that the probabilistic information was very likely to be “erroneous and inaccurate for users”.

The purpose of this paper is to describe the WaveSentry system: a set of components for harvesting observed data sources with different identified characteristics and implementing an application of the Bayesian statistical methodology that modifies the probabilities of ensemble wave forecasts based on recent past performance of individual members against these observations. Portrayal of the result set is also briefly indicated. A set of data sources are introduced followed by characterisation of each from a set of spatio-temporal feature types which facilitates interoperability and extensibility at this level. Components for data collection and incorporation are then described. An example prototype user interface is then given with a set of experimental results testing the methodology for a use case covering the English Channel. The proposed methodology allows for the incorporation of the various different types of observed and modelled data to create ensemble forecasts with potentially enhanced accuracy, in both best estimate and uncertainty, providing added value and confidence to end users. The systems built on the methodology are capable of using ensemble data that may be very time consuming and computationally expensive to produce, while reacting swiftly and efficiently when fresh observations bring in new information. Indeed, the ideas presented here are not confined to ensemble wave forecasts with supporting marine data, they are appropriate to any situation where ensemble model output can be post-processed in this manner.

2. Methods

2.1. Data sources

Applications were written to allow three independent types of measured wave data to be incorporated into post-processing ensemble wave forecasts: GNSS Reflectometry, FerryBox mounted accelerometers and static fixed position devices.

2.2. GNSS reflectometry data

GNSS reflectometry (Gleason et al., 2005) uses a comparatively low cost receiver to pick up backscatter from GPS signals from which sea state data parameters such as ‘mean square slope’ can be derived. With an increasing number of GPS satellite transmitters being deployed, mounting receivers to make this additional use of the GPS signal will lead to a considerable increase in the potential data coverage currently available from the constellation of traditional satellites currently fitted with wave sensing instruments. Near real time observations are available from existing satellites and this technology will translate to the GNSS receivers. This is dependent on full implementation and validation of fast analysis methods that transform the backscatter signals, via a Delayed Doppler Map (DDM), into a useful parameter set including mean square slope and surface wind speed. All parameters derived are presented together along a data track following the satellite receiver path. As different GNSS signals come in and out of range the data received can be patchy with large stretches of the tracks giving no data.

2.3. FerryBox accelerometer

The use of FerryBox vessel mounted instrumentation for observing water quality (i.e. physical, chemical and biological content) is fairly widespread (Chelsea Technologies 2012, Hydes et al., 2004; Hydes and Dunning, 2005; Dunning and Hand, 2005). Adding a low cost accelerometer device from which wave conditions can be derived has the potential to provide relatively wide geographical coverage (Dunning, 2011). By recording vessel motion, application of an inversion routine is required to compute the associated wave parameters. One difficulty is in the derivation of the inversion routine which is vessel specific and is more likely to work well for relatively small vessels that respond to a wide range of sea states. Methods for near real-time signal transmission are also problematic but expected to improve over the next few years with improved ship to shore communication systems. Trial installations were performed as part of WaveSentry on a Cross-Channel ferry and an NOC research vessel.

2.4. Fixed position wave measurements

There is a wide range of fixed position, in-situ, wave measurement devices. These include:

- Bed mounted Acoustic Wave and Current Profilers (AWAC) or Acoustic Doppler Current Profiler (ADCP or ADP),
- Sea surface mounted Pitch and roll wave rider buoys, and
- Land based wave radars e.g. Rex, HF and X-band.

These currently provide some of the lowest cost methods for carrying out wave measurements. The main advantage with these devices is that they provide dense time-series data at a constant location. The accompanying disadvantage is the relatively sparse coverage they provide. Signal transfer from bed mounted devices also provides an ongoing challenge largely due to battery life. During the WaveSentry project experiments were carried out using a bed mounted AWAC in the English Channel pilot study area located near to the existing Greenwich Light Ship, as shown in Fig. 1, in approximately 43 m water depth providing a source of data for validation. Fig. 2 shows the time-series of waves recorded during the period 3 April 2013–15 April 2013.

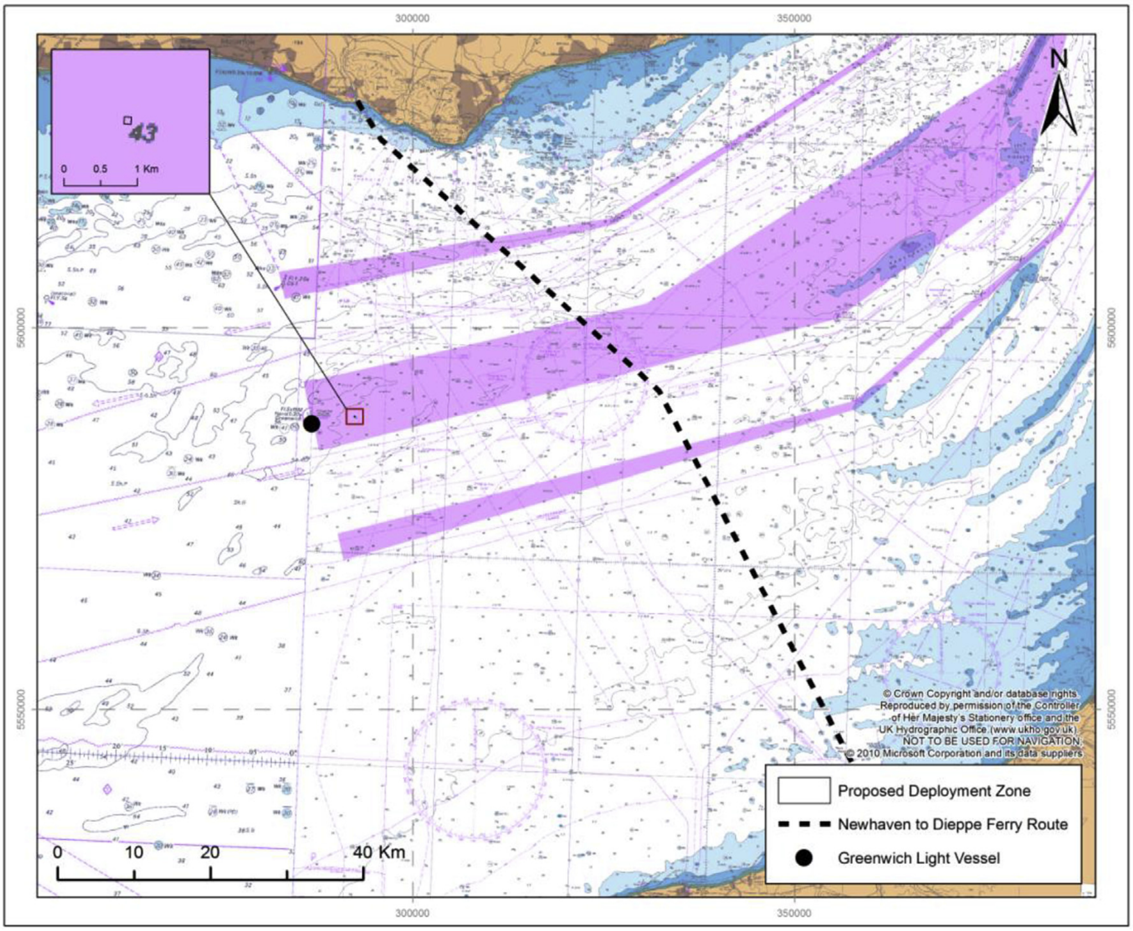
Such observations normally provide a source for off-line calibration and validation of numerical models. When near real-time data is available, this allows potential assimilation into operational forecast models.

3. Data characterisation

Considering these three data source types together with forecasting model output, different spatial and temporal characteristics are observed (see Harpham and Danovaro, 2015). These are summarised in Table 1 together with a note about the typical wave parameters measured by these devices.

It is necessary to allocate spatio-temporal structures, or features types, to these data types which reflect these characteristics. The Climate Science Modelling Language (CSML) offers a set of ten such spatio-temporal feature types intended to describe measured environmental data from devices (Lowe, 2011). Nine are specialisations of the ISO19156 Observations and Measurements (O&M) model (ISO19156, 2011) and one 'observation' is a direct usage of this standard. They are summarised in Table 2.

Three of these feature types can be directly adopted as a controlled structure for describing the data sources being considered here. Two of the data source types, GNSS Reflectometry and the FerryBox accelerometer represent the data in tracks where results vary in space and time. The spatial and temporal densities are very different but the same CSML feature type, a 'Trajectory', can be applied to both. Following a similar method to Harpham et al.



© Crown Copyright and/or database rights. Reproduced by permission of the Controller of Her Majesty's Stationery office and the UK Hydrographic Office (www.ukho.gov.uk). NOT TO BE USED FOR NAVIGATION. © 2010 Microsoft Corporation and its data suppliers

Fig. 1. Location of bed-mounted AWAC near the Greenwich light ship.

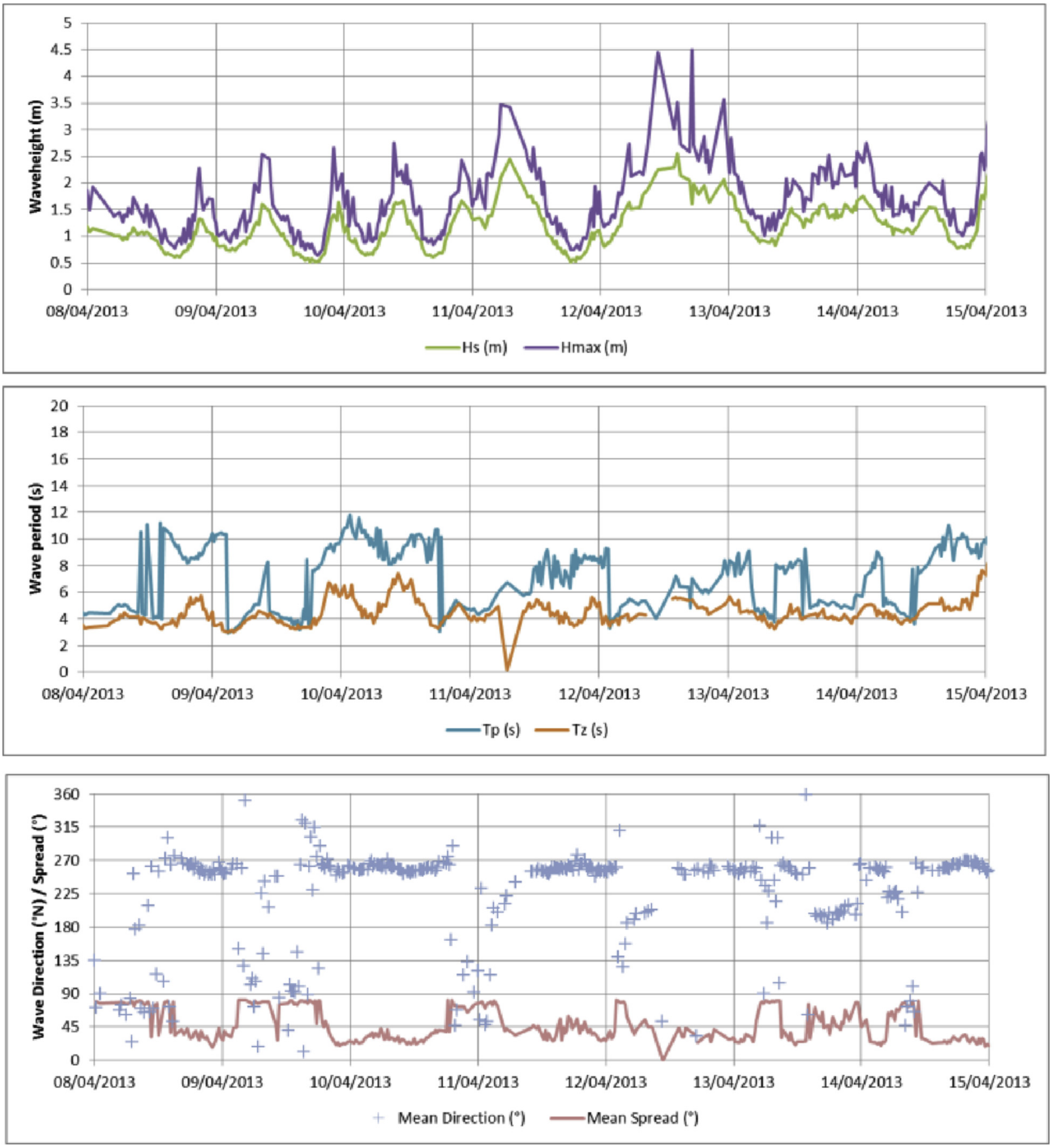


Fig. 2. Time-series of AWAC measured wave height, period and direction recorded between 3 and 15 April 2013.

(2015) (there applied in an hydro-meteorological context), Fixed position e.g. bed mounted ADCP and AWAC data can clearly be represented by a 'PointSeries', data that is varying in time but not space and measured at a single point. The regular grid output from Wave forecast models can clearly be represented by a CSML 'GridSeries'.

Fig. 3 shows each data source represented together in a notional example. On the extreme left of the figure two coasts are depicted at the top and bottom with an area of sea in between. Overlaid on this area of sea is a regular model grid (the method will also work with triangular meshes) with model predictions at each node for $t = t_{00}$, at the start of the forecast period. This model grid represents a single forecast. As time increases we move along the diagram to the right with more model results at t_{01}, t_{02}, t_{03} and t_{04} for the same grid, the first timesteps of a forecast with n timesteps. This single forecast is one member of an ensemble of similar forecast outcomes. So each ensemble member produces predictions across the same grid for timesteps t_{01}, …, t_n i.e. a GridSeries. For example t_{01} may be midday today with the ensemble producing many different possible forecast results ending at t_n, at midnight tomorrow.

Also depicted on the area of sea is the position of a wave rider buoy. The buoy is in a fixed position and produces readings that vary in time, but not space: i.e. a PointSeries. These readings are depicted on the dotted line proceeding from the buoy to the right

Table 1
Spatio-temporal characteristics of measured and modelled data sources with parameters typically measured by each device type.

Data source	Spatial characteristics	Temporal characteristics	Parameters
GNSS Reflectometry	Recorded on a track corresponding to the path of the satellite mounting the receiver. Sparse, intermittent observations. Exact track unlikely to be re-visited.	Burst of readings across study area as satellite passes. Re-visit time to vicinity depends on next satellite pass over region which could be measured in days, depending on the number of satellites.	Currently single parameter given (wave mean square slope).
Ferry Box Accelerometer	Recorded on track following vessel path. Dense observations. Exact track unlikely to be revisited.	High frequency readings (for example at 5 min intervals) when vessel is in study area. Revisit likely if vessel function requires local operation.	Measures vessel motion along 6 degrees of freedom from which wave height and period can be deduced with knowledge of vessel response characteristics.
Fixed position, in-situ, wave measurement devices (in particular, bed mounted ADCP and AWAC)	Readings produced at a single location.	High frequency readings (typically at 1 h intervals).	Multiple wave parameters measured including wave spectra, from which height, period and direction computed.
Wave Forecast Models	Data produced on a grid or irregular mesh of points covering an area. Spatial scale from global to metres.	Data produced at fixed time intervals, usually measured in minutes or hours.	The wave energy spectrum is typically the forecast model state variable from which a wide variety of parameters can be computed.

Table 2
Geo-temporal feature types from CSML.

CSML feature type	Description	Example
Point	A single observation at a point.	A single rain gauge measurement.
PointSeries	A series of 'Point' observations varying in time, but not space.	A stream of rain gauge measurements.
Profile	An observation along a vertical line in space.	Air temperature at a varying height above sea level.
ProfileSeries	A time-series of 'Profile' measurements.	A set of air temperature profiles taken at a set of time-steps.
Grid	Results given across a set of defined points in space.	2-dimensional HF Radar current output at a single time instant.
GridSeries	A time-series of 'Grid' measurements from the same defined grid.	2-dimensional HF Radar current outputs at multiple time instants against the same set of grid points.
Trajectory	An observation along a discrete path varying in time and space.	Water quality measurements taken from a moving ship.
Section	A series of 'Profiles' from a 'Trajectory'.	Marine Conductivity, Temperature and Depth (CTD) measurements taken from a moving ship.
Swath	A 'Trajectory' but with 2 spatial dimensions resulting in a 'Grid' output but varying also in time.	Advanced Very High Resolution Radiometer (AVHRR) satellite imagery taken from a satellite fly-past.
ScanningRadar	Backscatter profiles along a look direction at fixed elevation but rotating in azimuth.	Weather radar output.

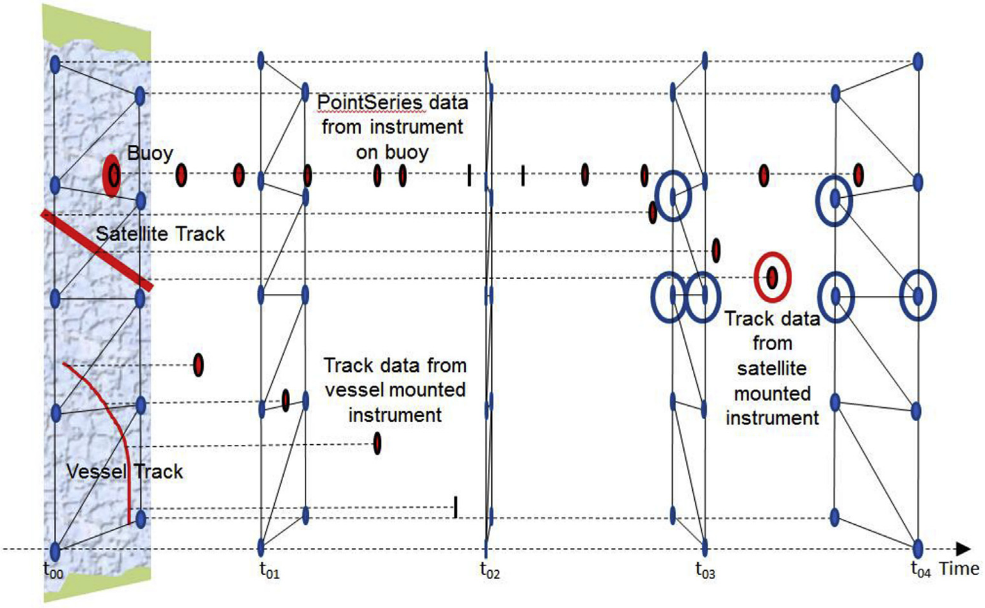

PointSeries data from instrument on buoy
Buoy
Satellite Track
Track data from vessel mounted instrument
Track data from satellite mounted instrument
Vessel Track
t_{00}
t_{01}
t_{02}
t_{03}
t_{04} Time

Fig. 3. Model output GridSeries data alongside PointSeries and tracks (Trajectories) from measuring devices.

across the figure. We cannot assume, indeed it is very unlikely, that these readings coincide in time with the timesteps of the model ensemble member (t_{01}, …, t_n) or in space with the nodes on the model grid. Immediately below the buoy a satellite track is depicted marking the line of readings taken by the satellite instrument as the satellite passes overhead. These readings occur successively, varying along the line in time and space: i.e. a Trajectory of measurements. Again, we cannot assume that any of these readings coincide in time or space with the model ensemble member results, although it is possible. Moreover, when the satellite passes this region again, we cannot assume that it will take the precise same path. In fact, typical satellites with these receivers housed will, at each circle of the earth, make their way along the region, forming a criss-cross or sequence of multiple tracks. Further to the south of the satellite track, a vessel track is depicted. Vessel mounted instruments will form Trajectories of readings varying in both time and space in the same way as the satellite, although they will, of course, follow the route that the vessel takes which can be considerably more changeable than that of the satellite resulting in a far less predictable spatial interval of readings, even with a reasonably constant timestep.

In WaveSentry it was never assumed that any of the data taken by the instruments has a fixed timestep between successive readings, although this may often be the case.

4. Data collection

Characterising the measurement data and the numerical model (ensemble forecast) results in terms of a feature type set allows the pipeline for processing each to be provided from a standard function set. Whilst not finely tuned for speed, this approach allows the flexibility to accommodate disparate data sources from different providers in different formats. The data collection pipeline for each data source consists of three steps: harvesting, transforming and loading.

4.1. Harvest

The raw/processed data from the measurement devices and ensemble forecast providers is collected using two approaches. The first is passive with scheduled jobs waiting for data to arrive on specified File Transfer Protocol (FTP) servers from a variety of data sources. Typically these are Comma Separated Values (.CSV files). Following detection of a new file it is transferred via FTP and stored in a file system structure. The second approach is active, with scheduled tasks extracting data proactively by connecting to other FTP servers to download relevant data files by arrangement with data providers. Each of these approaches is driven by configuration files in the INI file format and controlled via Python scripts. Following this approach, the emergence of new data sources is well defined and encapsulated keeping the amount of work required to bring such new sources online to a minimum. The data harvesting approach is summarised in Fig. 4.

4.2. Transform

The harvested data (both measurement and forecast) is then transformed into a consistent normalised structure built around the spatio-temporal feature type collection. Each local data source has a specific file format and internal structure; in addition each supplier may provide data for different parameters – some might include information about wind and temperature, others might include wave height, period and direction. Two canonical data structures have been used to encapsulate the three feature types characterising the data sources: a Track, directly adopting the CSML Trajectory and, observing that a PointSeries is a special case of a GridSeries (a GridSeries with one point), allows these two feature types to be covered by a single 'MeshPointSeries' structure, itself a generalisation of a GridSeries. Each disparate data source (and any similar future sources) can then be transformed into one or other of these two formats, conforming to their accompanying XML schema (MeshPointSeries.xsd and Track.xsd). This process is depicted in Fig. 5.

4.3. Load

Once the harvested data is transformed into either the Track or MeshPointSeries structure it is loaded into a spatially enabled relational database, in this case Postgres with PostGIS extensions (see Maidment, 2002 and Malinky et al., 2002 for similar examples). The database structure normalised the relationships between parameters, data sources, data series, and forecast ensembles and was optimised around the required feature types. Although not a pre-requisite by database design, for simplicity in implementing following functionality, each database instance carried only one forecasting GridSeries. Since the raw data forms were reduced to just two structures (MeshPointSeries and Track), only two standard loading applications were necessary. These are depicted in Fig. 6.

5. Data incorporation

With the data from the measurement devices and ensemble forecasts harvested and loaded into the same relational database structure derived from and optimised for the feature types used, the data incorporation process (a particle filtering method) can be undertaken. Primarily, this process is driven by a configuration file which identifies the devices and parameters of interest together with a parameter bias estimate for each device/parameter pair.

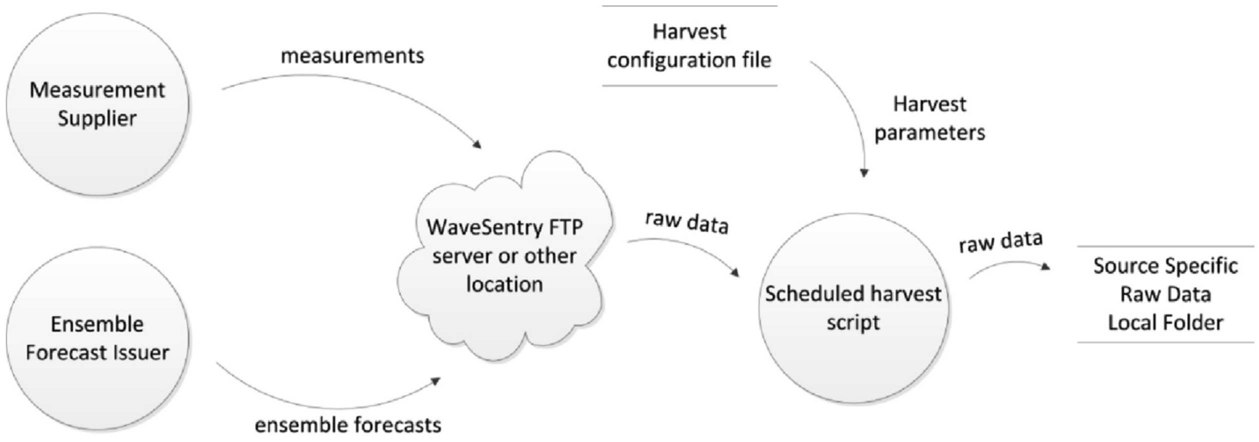


Fig. 4. Data harvesting approach.

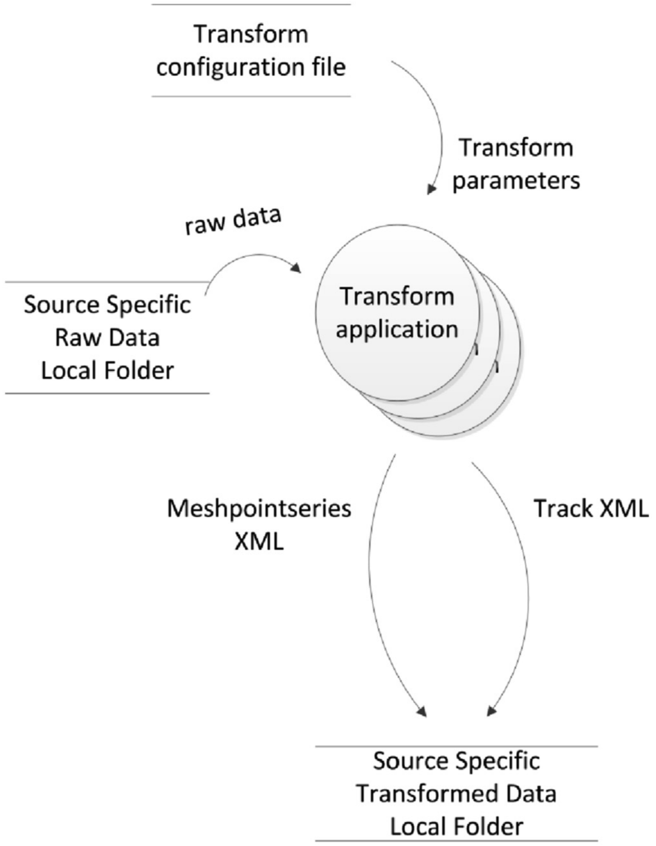

Transform configuration file
Transform parameters
raw data
Source Specific Raw Data Local Folder
Transform application
Meshpointseries XML
Track XML
Source Specific Transformed Data Local Folder

Fig. 5. Transformation of raw data into common formats.

Even though the data is mapped to the three different feature types (GridSeries, PointSeries, Trajectory) and reduced to two associated storage structures (MeshPointSeries and Track), it is important to note that the base construct supporting all of them is a point in 2-dimensional space. Each individual reading can therefore be considered in isolation as being registered against an independent point. This means that each ensemble member can be compared to the entire collection of available measured data by considering each independent point in turn. That is, the data points making up the GridSeries can be compared to the data points making up the Trajectories and the PointSeries. This process is illustrated in Fig. 3 where the circled reading from the satellite track is being compared to the six GridSeries readings bounding it, i.e. three from the preceding timestep at t_3 and three from the next timestep at t_4, also circled. Where the point readings to be compared are not coincident in time or space then a linear interpolation is performed. This is undertaken for each measured reading which is compared to the six readings bounding it. It is assumed that the grid will always be triangular so that there are six bounding readings. If the grid is rectangular then each rectangle is divided into two triangles by connecting two of the opposite vertices.

Several parameters may be recorded at each point. For example, at the point on the satellite track circled in Fig. 3 the wave mean square slope may have been measured from which a value for wave height may be inferred. The six surrounding points from the GridSeries (also circled) may have modelled results of three parameters, wave height, wave period and wave direction. A process was also introduced to filter very high frequency data (e.g. taking every *m*th value, where *m* is calculated as a function of the relative frequencies of values from the data sources) so that the observations could be considered as being statistically independent.

Using this method, the full collection of measured parameter values can be compared to each ensemble member in turn. Interpolating the ensemble forecast values to match the space and time of particular instrument reading from each of the PointSeries and Trajectories produces a set of observations y_i each with a corresponding forecast $\hat{y}_{i,j}$ for each ensemble member m_j. The aim of the proposed methodology is to improve the ensemble forecast by incorporating new measured data as it becomes available. Since the ensemble forecasts themselves cannot be easily modified, this additional information is incorporated by weighting the ensemble members and their corresponding forecasts. When the ensemble forecasts are first received the weight for each member is initialised to $w_i = 1/N$ where N is the number of members, i. e each member is equally weighted or likely to occur. These weights are then updated in light of the new measurements to non-equal values always ensuring that the weights continue to sum to one.

The weight updating strategy is motivated by posing the problem as a simple application of Bayesian data incorporation. This sets the strategy in a robust and rigorous theoretical framework. The true state of nature is unknown hence random in a Bayesian setting and approximated by the weighted ensemble of forecasts. The current ensemble weights before incorporating to a new set of

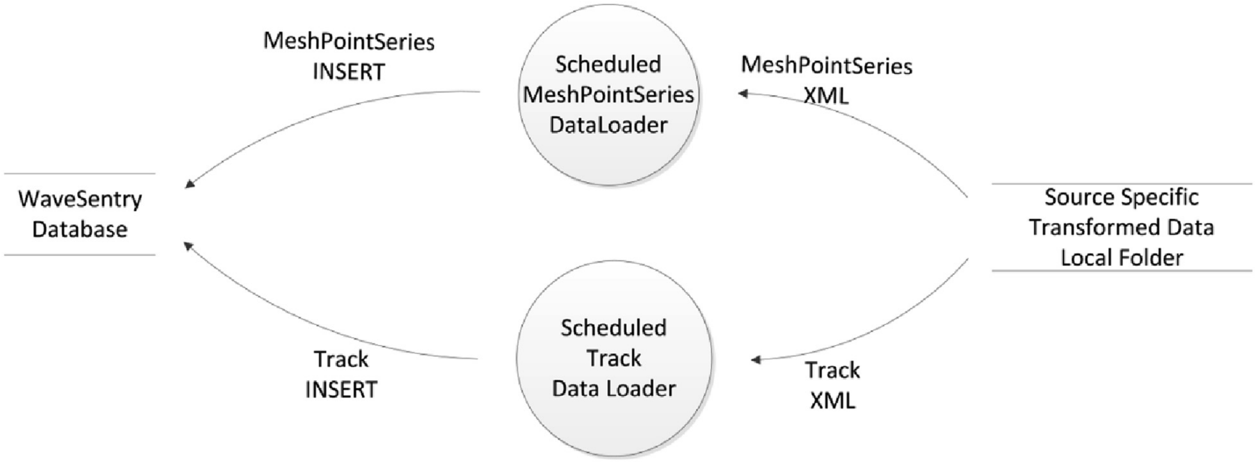

MeshPointSeries INSERT
Scheduled MeshPointSeries DataLoader
MeshPointSeries XML
WaveSentry Database
Source Specific Transformed Data Local Folder
Track INSERT
Scheduled Track Data Loader
Track XML

Fig. 6. Database loading functions for MeshPointSeries and Track.

instrumental data are treated as a prior probabilities on the ensemble members, that is $p(m_j)=w_j$.

Using Bayesian theory, these are updated in light of the new set of observations $\boldsymbol{y}=(y_1,\ldots, y_n)$ to produce the posterior ensemble weights $p(m_j|\boldsymbol{y})$. This is given by

$$p(m_j|\boldsymbol{y}) \propto p(m_j)p(\boldsymbol{y}|m_j) \tag{1}$$

that is, the posterior weights are proportional to the prior weights multiplied by the likelihood of the ensemble member. This is sufficient to define the posterior weights since they must always sum to one so may be calculated up to proportionality then normalised.

The likelihood term $p(\boldsymbol{y}|m_j)$ represents the probability of observing the new set of observations given knowledge of the ensemble member m_j. If the errors between the observations and the corresponding ensemble predictions can be considered statistically independent between observations, it follows that the likelihood is given by

$$p(\boldsymbol{y}|m_j) = \prod_{i=1}^{n} p(y_i|m_j) \tag{2}$$

that is the product of the individual observation likelihoods. Should this independence assumption not hold, it would be necessary to either model the dependence between observations or thin the observations until only independent values remain.

The single observation likelihood $p(y_i|m_j)$ defines a statistical model of the observation y_i given the ensemble m_j and in particular the corresponding forecast $\hat{y}_{i,j}$. A simple error equation is assumed taking the form

$$y_i = \hat{y}_{i,j} + \mu_{i,j} + \varepsilon_{i,j}$$

where $\mu_{i,j}$ is a systematic bias in the forecast and $\varepsilon_{i,j}$ is a random error assumed to follow the Normal distribution with zero mean and standard deviation $\sigma_{i,j}$.

The bias term represents the expected discrepancy between observation and forecast while the standard deviation governs the scale of the additional random error. This discrepancy is likely to be a combination of both forecast and measurement errors relative to the unobserved 'truth'. Rather than taking different values for every observation, the bias and error scale values are assumed to be constant over space and time for a particular parameter and instrumental device. It is also likely that the ensemble members are all of comparable accuracy hence there is no need for different values for each member, although potentially seasonal bias and error scales values could be considered. Both sets of terms are assumed to be known prior to applying the updating procedure and would typically be estimated by consideration of long-term sources.

These assumptions lead to a single observation likelihood of the form

$$p(y_i|m_j) = \frac{1}{\sqrt{2\pi\sigma_{i,j}^2}} \exp\left(-\frac{\left(y_i - \hat{y}_{i,j} - \mu_{i,j}\right)^2}{2\sigma_{i,j}^2}\right).$$

For directional observations, the squared difference should wrap around 360° as necessary to ensure that the smallest directional difference is squared. This equation can be applied with (1) and (2) to update the weight for each ensemble member, up to proportionality. These are then each divided by their total sum to rescale the weights to sum to one. These now represent the posterior ensemble probabilities which now account for the additional information provided by the latest set of observations $\boldsymbol{y}$.

If a new set of observations arrive after the weights have been updated, the same approach is used again taking the current weights as prior with respect to the latest observations. The properties of Bayesian theory ensure that the resulting weights are the same as would have been produced if all observations were incorporated at once or indeed if each were incorporated one at a time.

5.1. Using the weighted ensemble

As more information is accrued, it is likely that the weights will become increasingly uneven with a higher proportion of the probability mass focused on a smaller number of ensemble members. This has the same effect as reducing the number of ensemble members which increases the Monte Carlo error in the approximation of nature. The extent of sample degradation can be monitored by calculating the *effective sample size* (*ESS*) of Kong et al. (1994) given by

$$ESS = 1\Big/\sum_{j=1}^{N} w_j^2.$$

This is guaranteed to lie between the value 1 and the number of ensemble members N. It takes the value N only when the weights are all equal and the value 1 when all the probability is placed on a single member. The weighted ensemble can therefore be considered to have the same accuracy as an un-weighted ensemble with *ESS* members. When *ESS* is small there will be too few significantly-weighted members to produce a good Monte Carlo approximation hence the resulting forecasts will be unreliable until the model forecasts are updated and the weights reset.

Prior to weighting the ensemble, the expected value of a wave forecast $\hat{Y}$ of a particular time, location and parameter was estimated as the mean of the corresponding individual ensemble forecasts. With a weighted ensemble, this extends naturally to a weighted mean of the forecasts. Other summary statistics such as the standard deviation, confidence intervals and exceedance probabilities and be easily derived. However, these are only averaging the ensemble forecasts and neither take into account the potential bias nor the additional sampling error already assumed when updating the weights.

Statistical properties of potential future observations may be estimated by considering the posterior distribution of

$$Y = \hat{Y} + \mu + \varepsilon$$

that is the expected wave forecast plus the expected bias and observational error for the particular parameter and measurement device. For example, the expected value of a future observation is estimated by

$$E(Y) = \sum_{j=1}^{N} \left(\hat{y}_j + \mu_j\right) w_j$$

where $\hat{y}_j$ is the matching forecast from ensemble member m_j, μ_j is the bias for this type of parameter which potentially varies with member, and w_j is the current member weight. Statistics such as confidence limits can be estimated using Monte Carlo sampling of the observational error.

If the bias and standard deviation values used to fit the model are reapplied here, the resulting predictions will correspond to potential new observations of the same measurement device with their inherent measurement errors included. To instead make predictions of the unobserved 'true' value of each wave parameter, the bias and standard deviation values should correspond only to the expected discrepancy between the ensemble forecasts and the natural values, if this is known.

5.2. Potential extensions

The accuracy of the ensemble weights and their resulting predictions depends upon the validity of the statistical assumptions. The principal of these is the assumption of independence between observations. While this can to some extent be ensured by simply filtering the collected data, it may be preferable to extend the statistical model to account for known dependencies between observations. These may include dependencies in time but also correlations between multiple parameters recorded by a device.

The likelihood term (2) is a product of single-observation likelihoods only as a consequence of the assumption of independence between them. If this is no longer the case, the product of dependent observations is replaced by the joint distribution of the observations given a particular ensemble member's forecasts. Joint likelihoods of different sets of observations, for example from two separate instruments, may still be multiplied together if they can be considered independent of each other.

Since each measurement device is likely to record a time series of values on potentially short timesteps, the raw observations for each device and therefore the errors between this and each forecast are likely to be dependent in time. There are many ways this can be modelled but one option is to assume an autoregressive (AR) model for the discrepancy between observation and forecast. This assumes that the observation, or rather the discrepancy, at any timestep is a function of the p previous values plus white noise, where p is the order of the autoregressive process. This introduces p additional parameters which, as with the bias and error scale, must be estimated from a training set before use.

It may also be beneficial to account for dependencies between multiple parameters observed on a single measurement device, for example wave height and wave period. The dependencies between each set of parameters may be simply modelled by replacing the univariate Normal error by a multivariate Normal distribution. These dependencies are characterised by pairwise correlations which must again be estimated from a training dataset before application.

5.3. Portrayal of probabilistic forecasts

Once an evaluation has been conducted the results are stored in the database and transferred on a regular basis to an additional portrayal database (also a relational database with geospatial capabilities).

It was decided to keep these two databases separate to better balance the load between generating results and serving queries and improve the response time for any given query; real time ensemble forecast evaluation in response to a given query was unnecessary and also impractical with the computing resources available.

The portrayal system is a web site serving HTTP and making use of the Microsoft Bing mapping service to provide a map front end of the area of interest (in this case, The English Channel).

End users can select points of interest on a map, a time period of interest and one of three parameters (wave height, wave period and wave direction). The portrayal system will then display the calculated results from the portrayal database consistent with the parameters specified by the user.

6. Pilot application

In order to evaluate the proposed method, a pilot study application covering the English Channel between the UK and France was chosen. This area was selected because of the availability of an existing diverse set of wave data sources, some available in near real time. The area is also an important shipping route and has a wide variety of maritime users including cross channel ferries, offshore wind farm development, leisure users, etc. all in need of detailed forecast information. Measured wave data for this area was available from a range of on-line networks including:

- Channel Coast Observatory (www.channelcoast.org), UK Government funded countrywide scheme for coastal observations, including coastal waves.
- Wavenet (http://www.cefas.defra.gov.uk/our-science/observing-and-modelling/monitoring-programmes/wavenet.aspx) incorporating data from both local and central governments.
- Candhis (http://candhis.cetmef.developpement-durable.gouv.fr/) French government funded.
- GlobWave (www.globwave.org) providing data from traditional satellite observations of sea state.

For the purposes of the initial experiments described here, the sources of wave data were limited to data collected by the Met Office at: Channel Light Ship, Greenwich Light Ship and Sandettie Lightship which are all located centrally along the Channel, as shown in Fig. 7, and therefore providing a good source of measurements to assimilate.

Wind and wave ensemble forecasts from ECMWF were used for the pilot study. Fig. 7 shows the ECMWF operational wave model ensemble grid in the area of interest. Each 10 day forecast provides 50 ensemble members and a control (equivalent to the corresponding deterministic forecast). The ECMWF ensemble forecast is tuned to provide an accurate estimate of the uncertainty in forecast conditions with a lead time of 3 days or longer and a characteristic of this forecast is that there is no initial spread. So whilst this ensemble forecast dataset is useful for validating the general methodology, the proposed data incorporation approach may add more value if based on ensemble forecasts such as those that include initial perturbation i.e. uncertainty in the forecast.

By way of an example GUI, included here to show how the output can be portrayed to users in a decision support context, Fig. 8 shows the WaveSentry website home page map. As historical forecasts are stored, users can select a forecast date and an output type and the associated forecasts will appear as a figure that can be viewed or emailed. Three initial output plot types are provided:

- Box and Whisker;
- Time series probability of forecast conditions exceeding a given threshold, and
- Full ensemble member output.

Fig. 9 gives examples of the three primary plot types available from WaveSentry provided for illustrative purposes of the types of possible portrayal of ensemble forecasts.

The box and whisker time series graph (Fig. 9a) gives an indication of the uncertainty in forecast, the wide the error bars indicate higher uncertainty. The probability time series graph (Fig. 9b) shows the probability that the forecast wave height at a selected point will exceed the selected threshold of 2 m. The full ensemble time series plot (Fig. 9c) provides the end user with an indication of the spread of results from the set of ensemble members, mean and variance.

Initial results using the WaveSentry method are presented as time series plots of significant wave height (Hs (m)) for a 5 day forecast issued on 25 December 2012 at the Channel Light ship

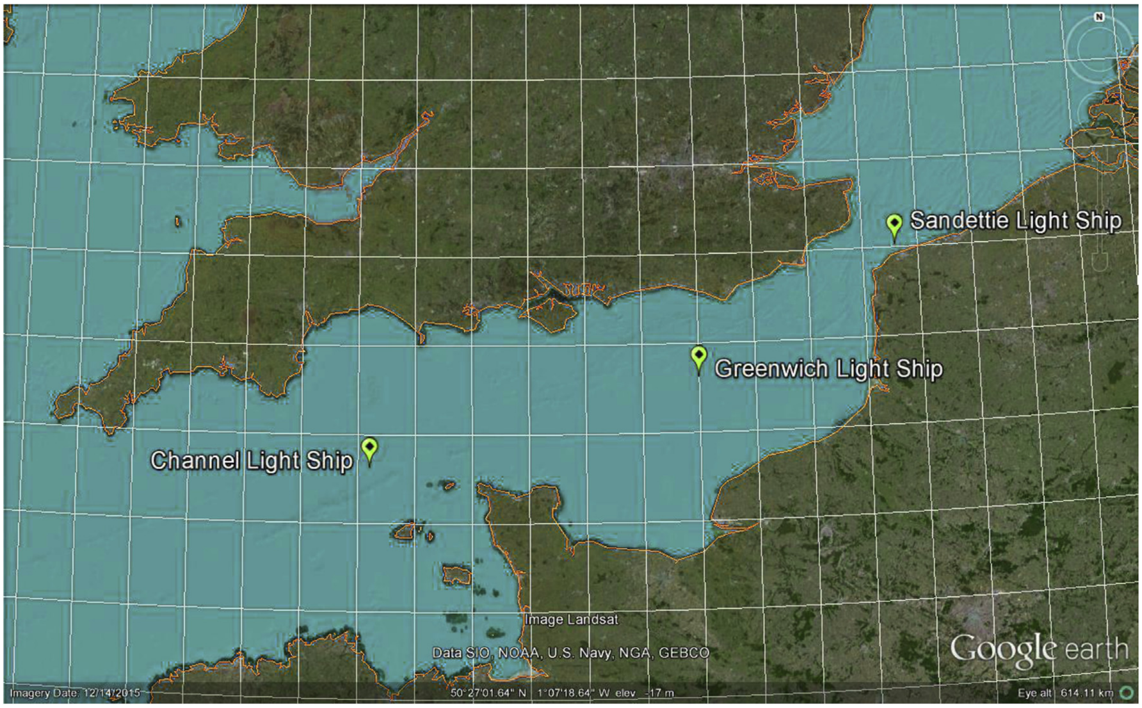
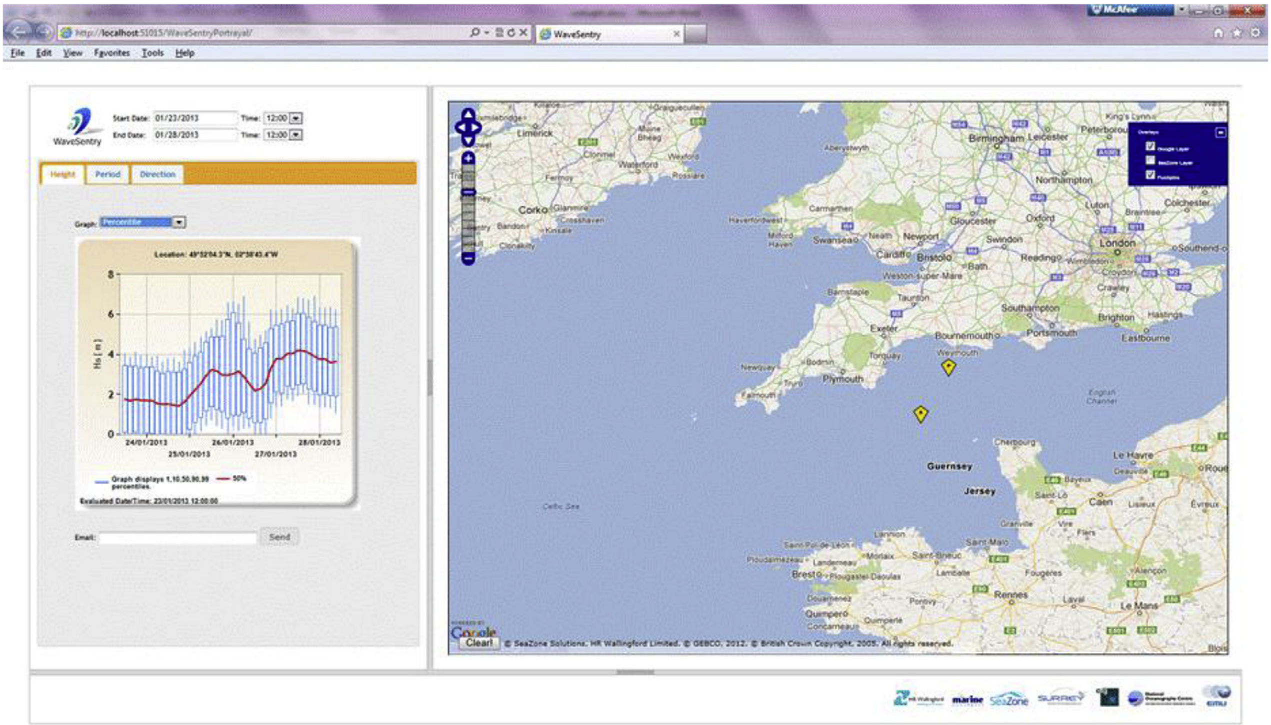


Fig. 7. Pilot area.


http://localhost:51015/WaveSentryPortrayal/

Start Date: 01/23/2013 Time: 12:00
End Date: 01/28/2013 Time: 12:00

Graph: Percentile

Location: 49°52'04.3"N, 02°38'43.4"W
Hs [m]
0
2
4
6
8
24/01/2013
25/01/2013
26/01/2013
27/01/2013
28/01/2013
Graph displays 1,10,50,90,99 percentiles.
50%

Evaluated Date/Time: 23/01/2013 12:00:00

Email:

Fig. 8. WaveSentry website homepage.

(49.9°N, 2.9°W) in the English Channel. Fig. 10 shows: the EMCWF forecast wave heights for each ensemble member; the control forecast (equivalent to the deterministic forecast); the ensemble mean; the WaveSentry ensemble mean computed after 12 h into the forecast; and the observations from the Channel Lightship.

Fig. 10 shows that compared with the raw ensemble mean, the WaveSentry ensemble mean was a better reproduction of the measured wave heights. This is largely attributed to the bias correction factor in the methodology and for this particular forecast holds true for approximately 36 h, after which improvement is no longer apparent. Table 3 summarises a sample of error statistics for this particular forecast. This shows that, as observed, the

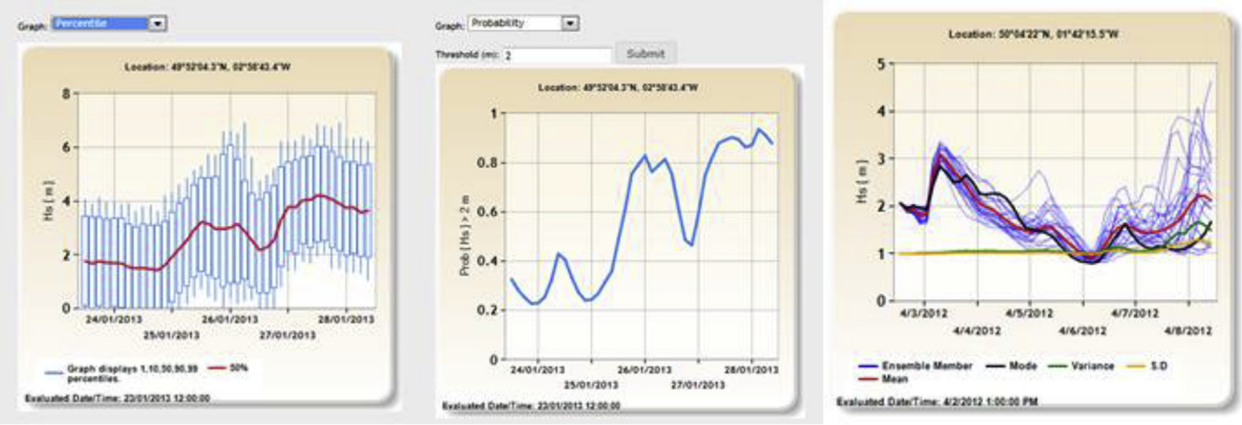


Fig. 9. WaveSentry Output Types (a. Box and Whisker, b. Probability above threshold, c. all member time series).

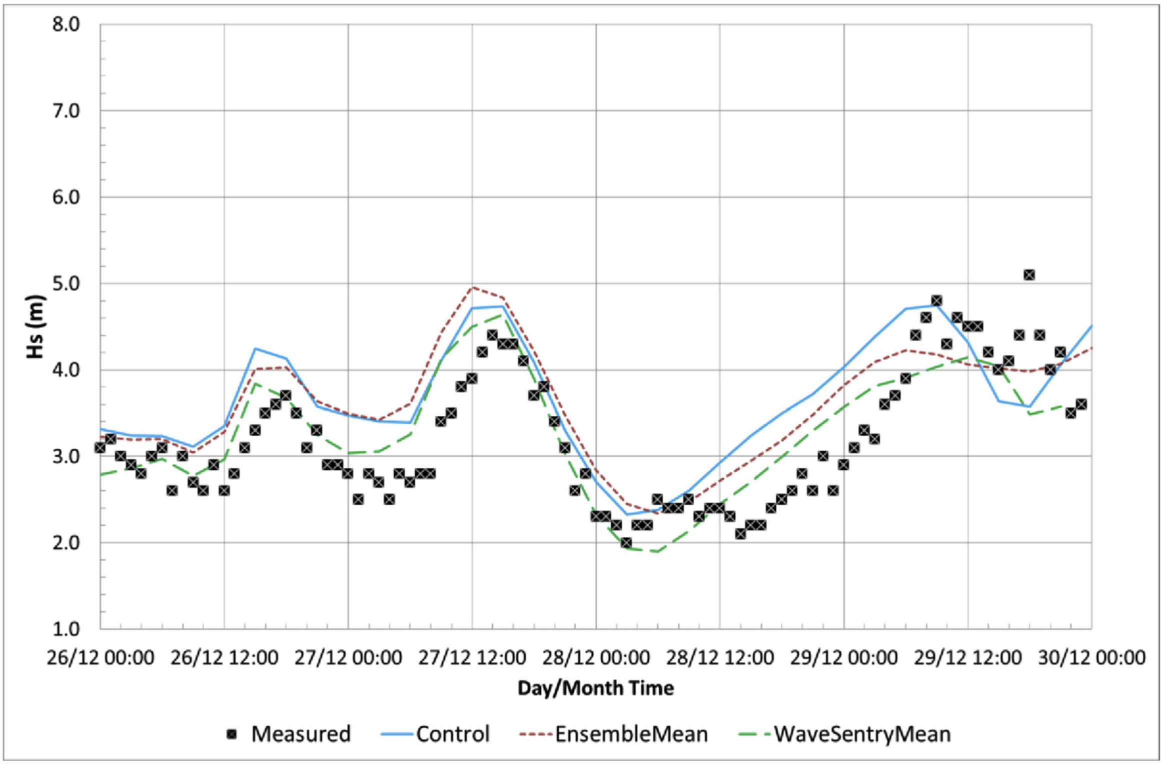


Fig. 10. Example forecast time series.

WaveSentry results in an overall lower, if slightly negative, bias, but also a lower root mean square (RMSE) error and slightly improved Scatter Index (SI).

Possible future iterations to explore the potential of the proposed method could, for example, explore a wider set of performance measures, for a wide range of parameters, test sites and forecast simulations. It is worth noting that in an operational forecast service, updates to the input forecasts would typically be issued at least twice daily, so it is expected that the WaveSentry approach is likely to only provide improved forecasts within each update.

Table 3
Error statistics for sample forecast significant wave height.

Forecast source	Bias (m)	Root mean square error (m)	Scatter Index
Control	0.43	0.68	0.21
Ensemble Mean	0.38	0.61	0.19
WaveSentry Mean	−0.05	0.51	0.16

7. Conclusions

A novel adaptive ensemble weighting scheme methodology for updating probabilistic wave forecasts based on recent observations has been developed. Enabled by harvesting data and converting into a set of standard feature types, the proposed implementation is capable of incorporating wave data from a range of sources including that from satellites, ships and in-situ devices. The results were presented to users on a web map enabled interface allowing selection of particular points for data retrieval and subsequent portrayal using a variety of scientific charts. Experiments using the

ECMWF ensemble wave forecast over the English Channel showed that the modified forecast compared more closely to the observations from the Channel Light Ship than the unmodified initial ensemble forecast.

Possible future iterations to explore the potential of the method include further assessment of the small positive impact observed on the forecasts as compared to the measured data and use of ensemble forecasts that have initial spread or measure of the initial uncertainty. This may also enable a longer backward looking analysis, which would open up the possibility for evaluation against more observations. Also, the method itself can be applied at a range of scales from local to global although validation of implementations at these scales would need to be undertaken.

Acknowledgements

This work was undertaken as part of the WaveSentry project, which was part funded by the UK Technology Strategy Board under the call: “Harnessing Large and Diverse Sources of Data”, grant number 100940. The authors would like to thank the other consortium members: Marine South East (assembling the consortium and project managing), EMU Ltd. (providing point series data), Chelsea Technologies Ltd. (providing vessel mounted track data) and Surrey Satellite Technologies Ltd. with the National Oceanography Centre (pioneering the GNSS Reflectometry techniques for interpreting wave height from satellite mounted receivers).

The authors would also like to thank ECMWF and the UK Met Office for the provision of ensemble wave forecasts.

The lead author would like to thank the John Radcliffe Hospital NHS Accident and Emergency Staff for repairing a broken wrist whist final edits of this paper were undertaken in the waiting room.

References

Lowe, D., 2011. *Climate Science Modelling Language v3.0* British Atmospheric Data Centre. http://csml.badc.rl.ac.uk/ (accessed 02.05.14.).

Alves, J.-H.G.M., Wittmann, P., Sestak, M., Schauer, J., Stripling, S., Bernier, N.B., McLean, J., Chao, Y., Chawla, A., Tolman, H., Nelson, G., Klotz, S., 2013. The NCEP–FNMOC Combined Wave Ensemble Product Expanding Benefits of Interagency Probabilistic Forecasts to the Oceanic Environment. Bulletin of the American Meteorological Society. December 2013.

Behrens, A., 2015. Development of an ensemble prediction system for ocean surface waves in a coastal area. Ocean. Dyn. 65, 469–486. http://dx.doi.org/10.1007/s10236-015-0825-y.

Bunney, C., Saulter, A., 2015. An ensemble forecast system for prediction of Atlantic–UK wind waves. Ocean. Model. 96 (part 1), 103–116. December 2015.

Cao, D., Tolamn, H., Chen, H.S., Chawla, A., Wittmann, P., 2009. Performance of the Ocean Wave Ensemble Forecast at NCEP. NOAA Marine Modelling and Analysis Branch (MMAB). Technical Note No.279. 2009.

Channel Coast Observatory, 2014. Strategic Regional Coastal Monitoring Programmes. http://www.channelcoast.org/data_management/real_time_data/charts/ (accessed 01.08.14.).

Dunning, J., 2011. Commercialisation of the FerryBox Concept; Finding new markets. In: 4th FerryBox Workshop 2011 Helmholtz-zentrum Geesthacht. http://www.hzg.de/imperia/md/content/ferryboxusergroup/presentations/fb-ws2011_dunning.pdf (accessed 29.07.14.).

Dunning, J., Hand, S., 2005. Design, manufacture, servicing and usage of FerryBox systems. In: Proceedings of the 4th EuroGoos Conference, Brest, June 2005.

Durrant, T.H., Woodcock, F., Greenslade, D.J.M., 2009. Consensus forecasts of modeled wave parameters. Wea. Forecast. 24, 492–503.

Gleason, S., Adjrad, M., Unwin, M., 2005. Sensing Ocean, Ice and Land Reflected Signals from Space: Results from the UK-DMC GPS Reflectometry Experiment. In: ION GNSS 18th International Technical Meeting of the Satellite Division, Long Beach, CA. http://spacejournal.ohio.edu/issue9/pdf/SensingOcean.pdf (accessed 03.07.14.).

Harpham, Q.K., Danovaro, E., 2015. Towards standard metadata to support models and interfaces in a hydro-meteorological model chain. J. Hydroinformatics 17 (2), 260–274. http://dx.doi.org/10.2166/hydro.2014.061. IWA Publishing.

Harpham, Q.K., Cleverley, P., D'Agostino, D., Galizia, A., Danovaro, E., Delogu, F., Fiori, E., 2015. Using a Model MAP to prepare hydro-meteorological models for generic use. Environ. Model. Softw. 73 (2015), 260–271.

Hydes, D., Dunning, J., 2005. FerryBox Celebrates first year of data collection. Mar. Sci. (10), 32–35, 1Q.

Hydes, D.J., Campbell, J., Dunning, J., 2004. Systematic Oceanographic Data Collected by FerryBox. Sea Technology Magazine. February 2004.

ISO19156, 2011. ISO 19156:2011 Geographic Information – Observations and Measurements. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32574 (accessed 02.05.14.).

Kong, A., Liu, J.S., Wong, W.H., 1994. Sequential imputations and Bayesian missing data problems. J. Am. Stat. Assoc. 89 (425), 278–288.

Maidment, D.R., 2002. Arc Hydro: GIS for Water Resources. ESRI, Redlands.

Malinky, S., Ruszovan, G., Funke, R., Stein, J., 2002. WISKI–A software package for acquisition, analysis and administration of time series data. Water Stud. 10, 463–472.

Pinson, P., Reikard, G., Bidlot, J.-R., 2012. Probabilistic forecasting of the wave energy flux. Appl. Energy 93, 364–370.

Saetra, O., Bidlot, J.-R., 2004. Potential Benefits of Using Probabilistic Forecasts for Waves and Marine Winds Based on the ECMWF Ensemble Prediction System. Weather Forecast. 19, 673–689.

Saunders, M., Lea, A., Chandler, R., 2014. How Well Do Ensemble Forecasts of European Windspeed Represent Uncertainty? PURE Research Blog Article. https://connect.innovateuk.org/web/pure-research-programme/article-view/-/blogs/pure-research-blog-how-well-do-ensemble-forecasts-of-european-windspeed-represent-uncertainty-by-prof-mark-saunders-ucl (accessed 03.07.14).

Chelsea Technologies, 2012. AquaLine FerryBox System: Autonomous Environmental Measurements from Ferries. http://cdn.chelsea.co.uk/images/Marine/Datasheets/aqualineferrybox/2271-063-PD-A-AquaLineFerryBoxSystem.pdf (accessed 03.07.14.).