DRIHM2US

DISTRIBUTED RESEARCH INFRASTRUCTURE FOR HYDRO-METEOROLOGY TO UNITED STATES OF AMERICA

D3.2: Interoperability Experiment Report

Abstract: This document reports on the interoperability experiments undertaken as part of the project.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 313122.

DRIHM2US

Document Information Page

Contract Number	313122
Project Name	Distributed Research Infrastructure for Hydro-Meteorology to United States of America
Project Acronym	DRIHM2US
Deliverable Number	3.2
Deliverable Name	Interoperability experiment report
Work Package Number	3
Work Package Name	Joint Prototypes and Expert Networking Sessions
Deadline	31/03/2015
Version	2.0
Dissemination Level	PU
Nature	R
Lead Beneficiary	HRW

Document History

Date	Version	Description
27th Mar 2014	0.1	Initial write up for review by the project team.
2nd Apr 2014	0.2	Issued for final project team review.
4th Apr 2014	1.0	First version for release.
16th Jan 2015	1.1	Update for contributions by project team prior to final release.
24th Apr 2015	2.0	Final version for release.

DRIHM2US

Table of Contents

DRIHM2US

1 Executive Summary

This report is DRIHM2US project deliverable D3.2, Interoperability Experiment Report. Its purpose is to document a set of experiments to test the interoperability of hydro-meteorological research (HMR) ICT architectures being developed in the EU and USA. The set of interoperability experiments has been devised to cover a variety of component parts of any potential e-Science infrastructure including data / modelling standards, controlled vocabularies, metadata standards and use of numerical models and ICT infrastructures.

A broad agreement on an exact definition of the term "interoperability" is non-trivial, because of the different contexts and expectations in a time where science in general and computer science in particular is constantly changing. For the purposes of this project we define interoperability as "the native ability of e-Science technologies and infrastructures to exchange, understand, and use information directly via common open standards-based message exchanges".

Using this definition of interoperability, the experiments completed have shown that: two well-defined and widely adopted data / modelling standards are strongly interoperable; notwithstanding potential requirements for re-engineering, components developed separately can be successfully coupled; inherent strength lies in the use of controlled vocabularies; metadata standards can provide a strong degree of interoperability to a certain level of detail and the ability to access XSEDE resources in the USA from a European developed and hosted workflow engine. In general but in particular for the experiments based on international standards, interoperability has been demonstrated to be more of an issue between scientific domains (e.g. meteorology, hydrology, hydraulics) than it is between Europe and the USA.

2 Introduction

This report is DRIHM2US project deliverable D3.2, Interoperability experiment report. It is one of the key deliverables from work package 3 [1] "Joint Prototypes and Expert Networking" and is part of task T3.1 "Interoperability test beds", with the objective of executing a set of experiments to test the interoperability of architectures being developed in the EU and USA. Its purpose is to document these experiments as they progress, report results and outline issues arising.

It is supported by two other reports, the 'Terms of reference for the domain expert meetings' (D3.1) and the 'Domain expert networking report' (D3.3). These two reports set out and document the four domain expert networking meetings, the only planned face-to-face meetings with all of the DRIHM2US [1] partners, together with the partners from the US companion project, SCIHM [2]. As part of the joint programme the DRIHM2US and SCIHM organisations undertake a set of experiments to test the interoperability of different aspects of ICT HMR infrastructures in Europe and the USA. These experiments are introduced and set out in D3.1 and D3.3 and fully documented here, in D3.2.

The Terms of reference for domain expert meetings, D3.1, was scheduled as a single delivery in month 2 of the project (December 2012); the first release of D3.3 at the end of month 9 (July 2013) and this report due first in month 17 (March 2014) alongside an update of D3.3 with final versions of both due at the end of the project in month 24 (October 2014).

3 Interoperability Experiments

The collection of experiments executed as part of the collaboration between DRIHM2US [1] and its US counterpart, SCIHM [2], is designed to span a wide range of potential interoperability areas identified by the hydro-meteorological and ICT communities. The experiments range from assessing data and metadata standards to utilising ICT infrastructures and numerical models.

Unfortunately, most deployed technologies, standards and methodologies as they relate to DRIHM2US/SCIHM are essentially not interoperable. This is mainly because of a limited adoption of a common reference model and being unable to agree on a common process of how interoperability can be established and then sustained. This lack of interoperability is a hindrance for the growing interest in the coordinated use of more than one infrastructure or standard.

In order to avoid ongoing fundamental discussions, the term "interoperability" needs to be precisely defined. However, the difficulty in coming to such a definition is best reflected in the fact that even IEEE offers four definitions of interoperability in 2000 [3]. According to IEEE [3], interoperability stands for:

- The ability of two or more systems or elements to exchange information and to use the information that has been exchanged.
- The capability for units of equipment to work together to do useful functions.
- The capability, promoted but not guaranteed by joint conformance with a given set of standards, that enables heterogeneous equipment, generally built by various vendors, to work together in a network environment.
- The ability of two or more systems or components to exchange information in a heterogeneous network and use that information.

Hence, a broad agreement on an exact definition of this term is non-trivial, because of the different contexts and expectations in a time where science in general and computer science in particular is constantly changing. New functions and capabilities of systems influence this definition. That, in turn, is another reason for the creation of a wide variety of community-specific definitions of this term. One such community-specific definition is provided in [4]

("native ability of Grids and Grid technologies to interact directly via common open standards"). For the purpose of this project we extend this definition to reflect aspects raised by IEEE and define "interoperability" as "the native ability of e-Science technologies and infrastructures to exchange, understand, and use information directly via common open standards-based message exchanges".

Interoperability is not synonymous with "interoperation": that which needs to be done to get e-Science infrastructures to work together as a fast short-term achievement using as much existing tuned (i.e. hacks, workarounds, tweaks, etc.) technologies as available today – as opposed to "interoperability" as the long-term "perfect solution".

3.1 Ingesting WaterML2 into an OpenMI Composition

The purpose of this interoperability experiment is to assess the feasibility of juxtaposing two different standards, devised for two different purposes, but which should be applicable to the same modelling study. 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) [5]. OpenMI is a software component interface definition originally conceived to allow the integration of interacting (environmental) processes which are usually numerical models, especially those associated with the movement and impact of water [6]. Both are recognised international standards which originate from the water domain with WaterML2 developed within the US run Open Geospatial Consortium (OGC) Hydrology Domain Working Group (DWG) and OpenMI through the European based OpenMI Association (OA).

3.1.1 Method

The HR Wallingford implementation of OpenMI is called FluidEarth [7]. It consists of two important tools. The first is a software development kit 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 [8]. The result of this process is the creation of an 'OpenMI Component' consisting of the model code as originally written (perhaps with minor modifications) together with an OpenMI wrapper. This wrapper enables input and output exchange items, as links, to be connected to

other OpenMI components. The second FluidEarth tool is a graphical user interface (GUI), called Pipistrelle, which allows OpenMI components to be assembled into linked compositions. For example, an OpenMI component modelling 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.

As outlined in more detail in D3.3 [9], the CUAHSI Hydrologic Information System is an online distributed system to support the sharing of hydrologic data from multiple repositories and databases. As part of the parent DRIHM project [10] point timeseries data is being held in the CUAHSI HIS system [11] 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 method applied was to construct FluidEarth OpenMI components to access WaterML2 files, interpret them and prove their usage in an OpenMI context.

Two OpenMI components were created for this experiment:

- A "WaterML Client Service" capable of i) retrieving WaterML files, either from a locally stored file or via the CUAHSI-HIS service where the project data was held and ii) of reading this file and passing the outputs as OpenMI output exchange items to other OpenMI components.
- An Hrw Locum Component which would i) act as an example OpenMI component to receive data across a link, ii) to prove the passage of the data by writing out a text file and iii) to be linked to a trigger to control the composition (data is 'pulled' between components with ultimate control pulling from the trigger).

Figure 3.1 shows the completed composition assembled in Pipistrelle.

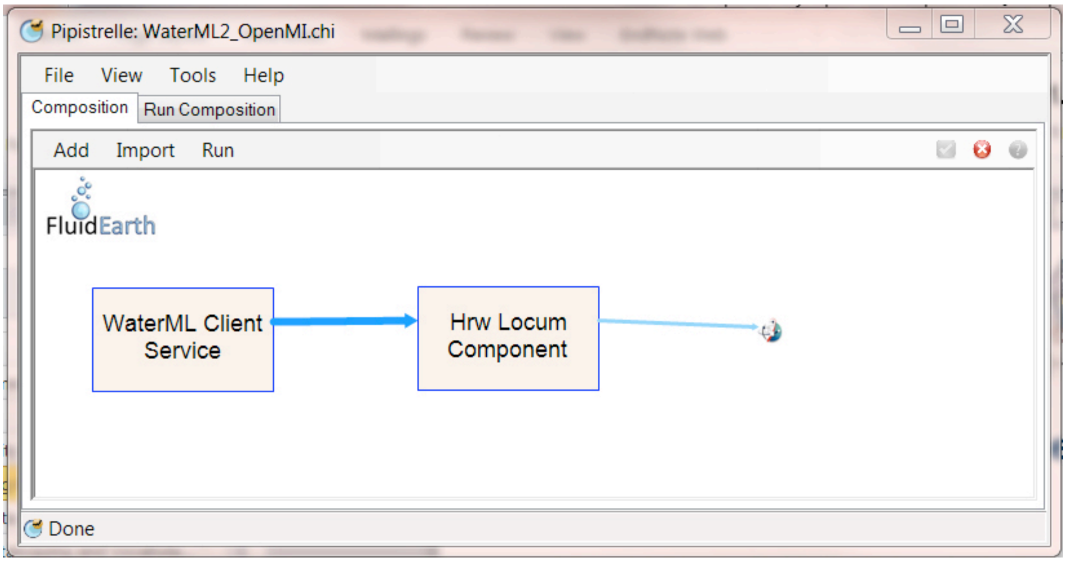


Figure 3.1: OpenMI composition for ingesting WaterML2 files

The WaterML Client Service component has a one way link to the Hrw Locum Component, which is attached to the trigger 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 things, the arguments to be used in the query to CUAHSI-HIS. Figure 3.2 gives this file with these arguments highlighted.

```
<?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 3.2: WaterML Client Service OMI file

The OpenMI composition was then run in Pipistrelle on a laptop at HR Wallingford.

3.1.2 Results

Following the creation of firewall rules to allow access to the CUAHSI-HIS web service, the composition ran without error. Pipistrelle creates a log file and shows this to the user as the composition completes. Figure 3.3 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 seconds.

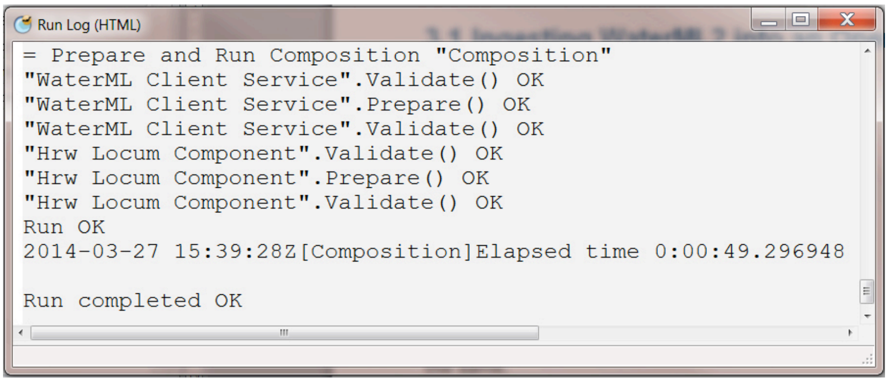


Figure 3.3: WaterML2 composition log file

The CUAHSI-HIS service issues WaterML2 files similar to that given in Appendix I. This data is read by the WaterML2 Client Service and written by the Locum component to csv files, as indicated earlier. Figure 3.4 gives two screenshots of this csv file displayed with TextPad. The first shows the top of the file with numerous 0 returns and the second, later in the file where some non-zero data values appear.

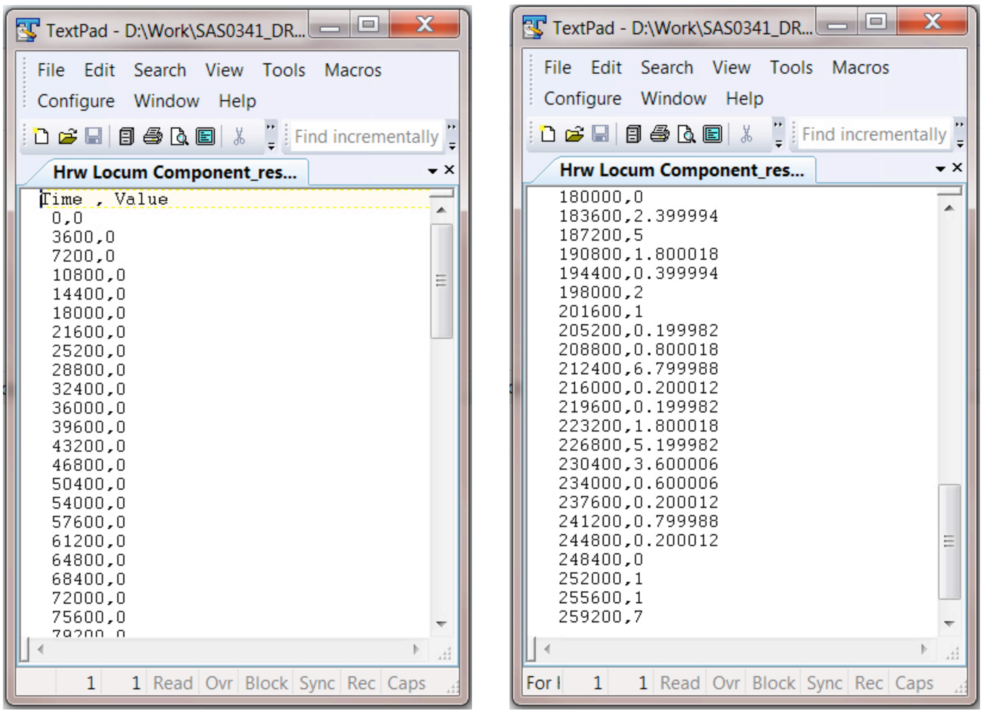


Figure 3.4: Results returned from CUAHSI-HIS written out as csv

3.1.3 Conclusions

Overall the experiment was successful indicating that it is quite possible to ingest WaterML2 data into an OpenMI composition. The two standards, although they were developed for different purposes and by different groups, allow their mutual juxtaposition in modelling studies. The details of the implementation, particularly from the OpenMI side, yield a number of valuable learning points.

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 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 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 (see Appendix I). The

performance hit would be considerable. The strategy used in the experiment 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 seconds, 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 3.2 were hard-coded into the omi file through the Pipistrelle GUI by the modeller. A better, more flexible and generic approach would be to allow these to be passed to the component as data is required. However, modellers often prefer to check all data sources as part of the composition assembly and have more control over data coming from external sources.

The work done in assembling this set of OpenMI components was further developed on the DRIHM project, contributing to the hydraulic and impact model composition used there for ingesting WaterML2 data.

3.2 Coupling WRF with Continuum

The purpose of this interoperability experiment was to test the feasibility of coupling Continuum, a continuous distributed hydrological model developed in Europe, with Weather Research and Forecasting (WRF), a mesoscale numerical weather prediction system from the USA. This model coupling follows the same pattern as that successfully attempted on the DRIHM project – that of coupling a meteorological model to an hydrological model – although this experiment would result in the initial stages of construction of a WRF-Continuum hybrid. It addresses functional concerns and technical issues which arise from considering the applications themselves, rather than underlying infrastructures or standards.

The WRF-Continuum experiment follows the WRF-Hydro pattern, that is, as a framework designed to facilitate easier coupling between the Weather Research and Forecasting model

and components of terrestrial hydrological models.

The WRF-Continuum coupling experiment was also designed to result in a prototype stand-alone hydrological modeling architecture as well as an architecture for coupling of hydrological models with atmospheric models. It is intended to be developed to be fully-parallelized to enable its usage on clusters and high performance computing systems. Like the WRF model itself, it does not attempt to prescribe a particular or singular suite of physics but, instead, is designed to be extensible to new hydrological parameterizations. Although the resultant work will be able to be to be used within the WRF model, it can evolve over time to include:

- Multi-scale functionality to permit modelling of atmospheric, land surface and hydrological processes on different spatial grids.
- Modularized component model coupling interfaces for many typical terrestrial hydrological processes such as surface runoff, channel flow, lake/reservoir flow, sub-surface flow, land-atmosphere exchanges.
- Parallel code development for application on commodity cluster and higher performance computing systems.
- Stand-alone capabilities for hydrological prediction and research uncoupled to atmospheric models.
- Efficient coupling architecture so that it can be embedded within (or coupled to) other types of Earth system models.
- Utilization of many standard data formats for efficient job construction and evaluation.

3.2.1 Method

In software development, coupling refers to the degree to which software components are dependent upon each other. For instance, in a tightly-coupled architecture, each component and its associated components must be present in order for code to be successfully executed or compiled. In a loosely-coupled architecture, components can remain autonomous and allow middleware software to manage communication between them. In a decoupled architecture, the components can operate completely separately and independently.

The method used in the WRF-Continuum coupling experiment can be defined as a loosely-

coupled system: interconnecting the components in a system so that those components depend on each other, but to the least extent practicable. The main goal of this is to reduce the risk that a change made within one element will create unanticipated changes within other elements. Limiting interconnections can help isolate problems when things go wrong and simplify testing, maintenance and troubleshooting procedures.

A loosely coupled system can be easily broken down into definable elements. The extent of coupling in a system can be measured by mapping the maximum number of element changes that can occur without adverse effects. Examples of such changes include adding elements, removing elements, renaming elements, reconfiguring elements, modifying internal element characteristics and rearranging the way in which elements are interconnected. As such, an interface system was developed to link the WRF and Continuum models so that forcing and state variables can be used in a complete hydro-meteorological framework. If this interface structure is switched off, Continuum and WRF can be run and compiled separately.

Overall, this coupling of these model codes implies:

- exchange and transform of information at the code interface;
- management of the execution and synchronization of the codes;

with the following constraints:

- physical conservation i.e. energy conservation at the interfaces;
- the coupling algorithm as dictated by the science;
- the coupling itself should be easy to implement, flexible, efficient and portable;
- the starting point from existing and independently developed codes;
- feasible performance and load balancing;
- varied computing platform and OS characteristics.

3.2.2 Results

A large part of the experiment was taken up with re-structuring the Continuum model code to be consistent with the prescribed method and coupled with the WRF framework. Originally, Continuum was written in Fortran 77/90 using an old fashioned style of programming that utilized single subroutines and common declaration of variables. The Input/Output system

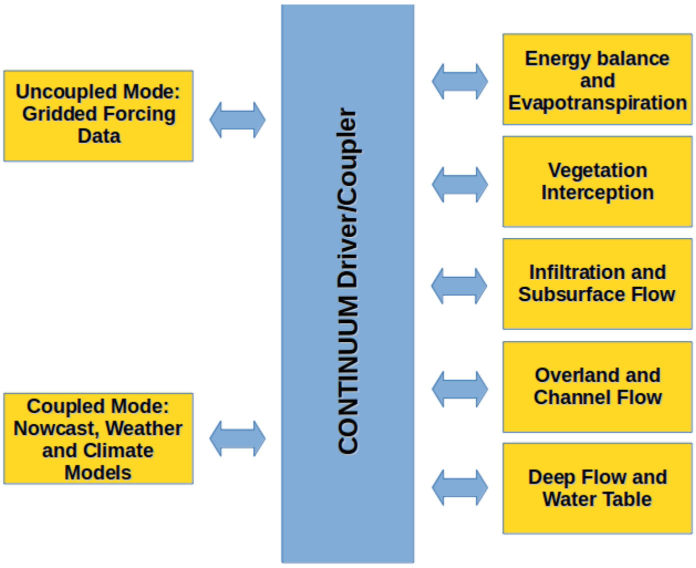
forced data and results in a binary format, static input files in an ascii grid and parameter files in a text format. The code was sequential and no parallel libraries were used. As it stands, the model can be used in a 1-way coupling with a meteorological model. Data is output from the meteorological model and passed to the hydrological model in a file-based interface.

From the physics point of view, Continuum provides both energy and mass balance. Energy balance is described using the so-called "force-restore equation" that evaluates the land surface temperature dynamic and consequently the energy fluxes (sensible heat and latent heat) and soil moisture. Mass balance is computed with infiltration and subsurface flow using a semi empirical, but quite detailed, methodology based on a modification of the Horton algorithm and focuses especially on exploiting land use and climatology information, which are easily available, to set the infiltration parameters. The overland runoff is distributed with differentiation between hill slope and channel flow. Vegetation interception and water table flow have been also schematized.

The physics is managed by a main procedure that sequentially runs each part of the model. Energy and mass balance state variables and processes are not clearly divided and this structure does not allow each to be easily identified. Accordingly, the re-engineered version of the Continuum model, following the best practices of modern Fortran 90/95, modules, derived data types and namelists are largely used. In fact, in typical engineering programming applications, it is often the case that there are parameters, variables, and subprograms that must be shared by several program units. Fortran 90/95 provides a special program unit known as a module that conveniently packages collections of declarations and subprograms so that they may be imported into other program units.

A derived data type is also called a structure and it can consist of data objects of different types. Namelists and sequential read statements translate data from external to internal form by using the data types of the objects in the corresponding namelist statement to determine the form of the data. The translated data is assigned to the specified objects in the namelist group in the order in which they appear, from left to right.

As such, energy balance and relative subroutines have been collected in a module named LSM (land surface model) and mass balance processes in another module named RT (routing). The re-engineered version of the Continuum model is represented by the combination of LSM and RT modules. The uncoupled version has three computational steps and the coupled version five: (1) initialize state variables and model, (2) disaggregate state variables from meteorological model (coupled version only), (3) execute model (LSM + RT), (4) collect data



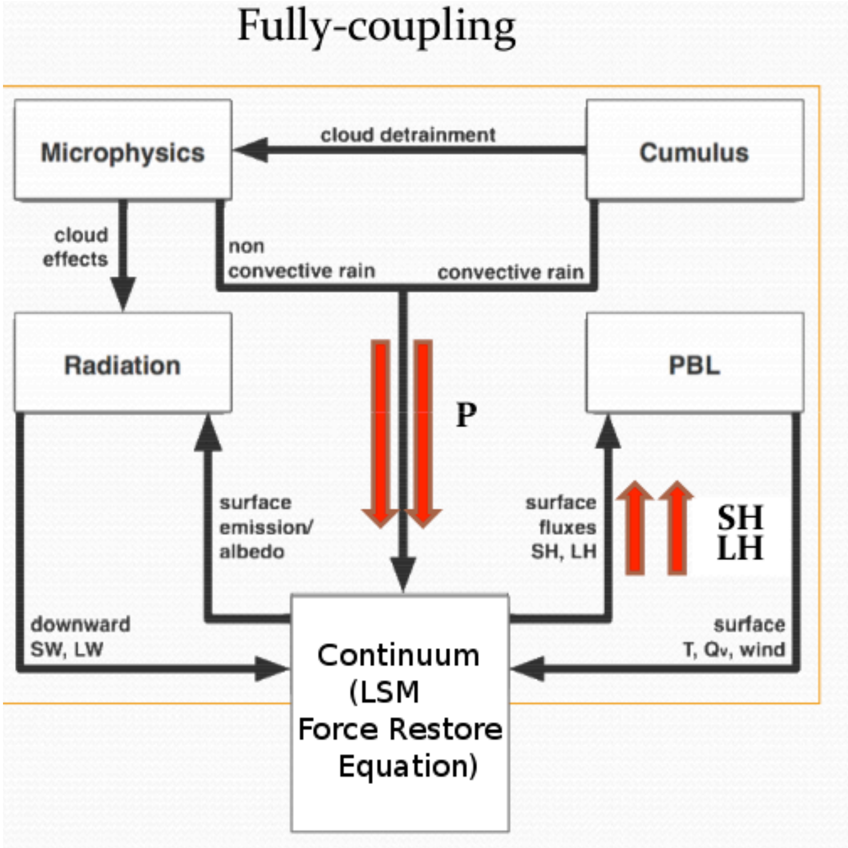
n the global memory of the WRF model and framework (coupled version only).

been developed to drive the LSM and RT input (uncoupled versions) or a fully coupled utputs are in NetCDF format (version 3 or 4) Metadata Conventions. The library is linked into modules that use it to read/write NetCDF files.

Figure 3.5: Example of Driver/Coupler system; two configuration are available for uncoupled and coupled model.

This structure is now suitable for linkage to the WRF framework because it is possible to select only a part of the hydrological model (for example the routing part RT) and call it directly from the WRF side. Moreover, to replace the Continuum LSM part, for example, it could be possible to use one of the land surface models available on WRF system (for example NOAHMP land surface model). As mentioned previously, it is also possible to execute the complete model (LSM + RT modules). The differences between these configurations only arise during the compilation phase where, according with the simulation settings, only the necessary parts for the selected application will be compiled.

For the first WRF-Continuum model configuration, all the hydrological model parts were used and called from the WRF surface driver module (WRF3/phys/surface_driver.F). This



ne as given in figure 3.6.

Figure 3.6: An example of a fully coupled system of WRF and Continuum. Energy fluxes are calculated in the hydrological model and updated in the meteorological framework.

In the Fortran code, after the WRF call, all interactions between WRF and Continuum are managed by the driver/coupler interface. The first phase consists of loading all the Continuum settings and parameters, using a specific namelist, into the common variables of the WRF framework. Secondly, all the forcing and state variables useful for the hydrological part are disaggregated from the meteorological coarse grid to the hydrological fine mesh. All the state variables, from the disaggregation phase, are passed to the LSM and RT modules to calculate energy and mass balance variables.

At the end of each Continuum computational step, energy fluxes and soil moisture are aggregated from the fine mesh to the coarse grid and are used to update the WRF state variables. The same procedure is performed for each step of the coupled simulations; the

expected result is an improving of the estimation of the exchange state variables (such as energy fluxes and soil moisture).

Additionally, to apply Continuum into the WRF system a set of Python scripts to pre-process static data has been developed. In particular, these procedures carry out geographical re-projections, grid interpolations and retrieving of digital elevation model (DEM) files from the USGS HydroSHEDS database. Preprocessing tools create a NetCDF file that contains all the necessary static information for running the Continuum model.

In the case of the uncoupled version, dynamic data is prepared using a Python script, starting from observed point time-series and generating gridded fields of forcing data or alternatively starting from meteorological offline fields and interpolating them onto the Continuum domain.

3.2.3 Conclusions

The coupling between the US developed WRF modelling suite and the European Continuum model was shown to be feasible. This was not possible however, without re-engineering the legacy Continuum model to use more modern programming techniques. The overall scheme used, although derived independently, follows the same pattern as that of the FluidEarth implementation of OpenMI [7]. In order for the two-way coupling to function successfully, modules within Continuum were made accessible for calling by external components. This is analogous to the input and output exchange items within an OpenMI component passing data to (and from) other components as the whole composition is controlled from an external entity. In the case of the OpenMI composition, this entity is the component agnostic Pipistrelle application which allows users to build and run compositions, in the case of the WRF-Continuum coupling this entity is the WRF modelling system itself.

This application also reinforces the concept of adaptors between components to handle any necessary transformations such as aggregation or dis-aggregation between grids and meshes, geospatial coordinate transformations or temporal interpolation.

This first prototype of the WRF-Continuum coupled system does not have a parallel running mode and so cannot be scaled up to very large volumes or handle ensemble scale outputs. It is speculated, at this stage, that a reasonably efficient coarse grain parallelism will be possible to gain significant efficiencies with any future parallel implementation.

3.3 Standard naming of parameters for interfacing from meteorological models

This experiment tests the strength of the Climate and Forecasting Standard Names parameter dictionary by applying it across a variety of meteorological models and exploring its extension into the hydrological domain. The process and use case was conducted in conjunction with technical interface definition work undertaken on the DRIHM project [10].

3.3.1 Method

The method applied was to go through a semantic analytical process with meteorologists from Europe and the USA who run a variety of meteorological models in an attempt to define a 'reasonable' set of parameter names (taken from the output of the models, but defined in the CF Standard Names dictionary), used by meteorologists to be used in downstream, one-way interfaces to other models. The idea is to bring together the typical parameters which would be used by other models and to communicate them to non-meteorologists in a way that these experts inheriting this data from the meteorologists would understand.

The analysis proceeded as follows:

- Select a single meteorological model.
- Draw a list of meteorological parameters from this model, which, in the eyes of the model specialists are commonly required for interfaces elsewhere.
- Compare this parameter list to those produced by other meteorological models and rationalise together.

In this way a set of well defined and universally understood environmental parameters would be built and offered to other modellers and could be crafted into a set of common interface definitions (see DRIHM description of work (DoW) [10]). A corollary of this study extended this idea into the hydrological domain by considering a similar parameter interface further down the model chain.

3.3.2 Results

The initial meteorological model selected was the Weather Research and Forecasting (WRF) ARW model. It was developed in the USA but is also in common use in Europe. An assessment of the parameters relevant to downstream interfaces was given as follows. This analysis produced a set of parameters which are defined in the CF Standard Names definition [12] but also with additional, WRF ARW specific, metadata qualification used in the output datasets.

Table 3.1: WRF ARW Parameters potentially relevant to downstream interfaces

Parameter
lwe_thickness_of_total_precipitation_amount;
lwe_thickness_of_precipitation_amount: model_description = "Total grid scale precipitation" ;
lwe_thickness_of_precipitation_amount: model_name = "RAINNC" ;
lwe_thickness_of_convective_precipitation_amount: model_description = "Total cumulus precipitation" ;
lwe_thickness_of_convective_precipitation_amount: model_name = "RAINC" ;
lwe_thickness_of_total_precipitation_amount: model_description = "" ;
lwe_thickness_of_total_precipitation_amount:model_name = "RAINNC + RAINC" ;
air_temperature:model_description = "Temp at 2 m" ; air_temperature:model_name = "T2" ;
specific_humidity:model_description = "Qv at 2 m" ; specific_humidity:model_name = "Q2" ;
surface_net_downward_longwave_flux: model_description = "Downward long wave flux at ground surface" ;
surface_net_downward_longwave_flux: model_name = "GLW" ;
eastward_wind:model_description = "U at 10 m" ;
eastward_wind:model_name = "U10" ;
northward_wind:model_description = "V at 10 m" ;
northward_wind:model_name = "V10" ;
surface_air_pressure:model_description = "Sfc Pressure" ;
surface_air_pressure:model_name = "PSFC" ;
surface accumulated precipitation

This list of parameters was then rationalised to the list given below by comparing to two other meteorological models, WRF NMM (USA) and Meso-NH (Europe). The parameter definition was then separated out from the vertical position at which it is to be applied (and also the unit). Certain of these parameters are calculated at the surface, others at 2m above the surface and others at 10m 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.

Table 3.2: 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;	2m	K
specific_humidity;	2m	1
surface_net_downward_longwave_flux;	surface	$W\ m^{-2}$
eastward_wind;	10m	ms^{-1}
northward_wind;	10m	ms^{-1}
surface_air_pressure;	surface	Pa

Analysis was conducted of a second interface further down the model chain used on the DRIHM project [10]. This interface passes data from an hydrological model to an hydraulic model: drainage of river catchment into river channel. Although the geospatial definitions at the interface were different, the same semantic issues between scientific model domains had to be solved. This resulted in the definition of an additional parameter to serve this interface as follows.

Table 3.3: 'river_discharge' parameter definition

Standard Parameter	Definition	Unit
river_discharge	"river_discharge" is the instantaneous volumetric flux of water flow through a river cross-sectional area.	m^3s^{-1}

3.3.3 Conclusions

The process concluded in defining what was considered to be a reasonable candidate list of meteorological parameters to be offered to non-meteorologists in standard interfaces. This includes the usual typical meteorological parameters expected 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, which were developed by experts including those from the USA and Europe, was pivotal in this process. As a result the meteorological semantic issues were minimal, communication clear and definitions agreed and common. Metadata definitions for parameters do exist for the hydrology / hydraulics domain but they are not as universally agreed or defined. For example, the "river_discharge" parameter defined corresponds to the "Discharge.stream" item in the CUAHSI-HIS ontology service controlled vocabulary [13]. Moreover, parameter definitions from one scientific domain may be inappropriate for another. Sometimes the same phenomenon is expressed in two different ways, sometimes specialised details are brought out by one community and not another, sometimes terminology simply doesn't exist across all disciplines. For example, a generic 'crop' definition in WRF (describing crops grown in fields) may not be sufficient for the requirements of an hydrological model. Mappings such as the one given in figure 3.7 illustrate the approximations which can occur across interfaces.

HydrologicModel.maize=WRF.crop; HydrologicModel.wheat=WRF.crop

Figure 3.7: Mapping from the WRF.crop parameter to more specialised definitions in hydrologic models

Fixing the vertical dimension to that which would commonly be required in typical output, such as taking parameter "lwe_thickness_of_precipitation_amount" just at the surface or parameter "air_temperature" at 2m above the surface allows results for each parameter to expressed as a 2-dimensional grid series. This is simpler to process and results in considerably less data being transferred at interfaces. This strategy will not apply universally since 3D results or 2D results at different vertical levels will, of course, be required, but many use cases will be satisfied. It is also important to note that the vertical levels indicated do not necessarily correspond to the levels in the 3D model grid; they need to be derived as required.

3.4 Metadata structures for cataloguing and interfacing models

This experiment works in conjunction with the parameter naming experiment, above. The parameter names adopted in that experiment are now incorporated as part of a more comprehensive metadata strategy for describing environmental numerical models [14]. The purpose is to determine how well such metadata supports the 'discovery' and 'use' of numerical models, particularly with respect to model interfacing (passing data between them). Options for validating interfaces using just the supporting metadata will be considered. These metadata structures are tested by considering an interface between a model developed in the US and a model developed in Europe.

3.4.1 Method

Once again an analytical process was followed which maximises the synergy between the DRIHM2US [1], DRIHM [10] and SCIHM [2] projects. A structured but unencoded metadata standard was applied to two numerical models and assessed:

- Select a metadata standard used to describe numerical models;
- Select two numerical models that would validly interface (preferably one from the USA and one from Europe);
- Apply the metadata standard to the models;
- Analyse the metadata representation, particularly with respect to the interface characteristics.

3.4.2 Results

The metadata standard selected to describe models is that adopted by the DRIHM project [10]. It is based on ISO 19115 (a standard devised for describing geospatial datasets), but with extensions to allow it to be used to describe numerical models [14]. In particular, three important aspects are included:

i) An inheritance relationship between model engines (core code) and model instances (which occur when the engine is applied to a time and place with supporting configuration files). Each model instance is a child of the engine, inheriting all metadata from the engine with added elements to describe the instance.

ii) The incorporation of a separate element to describe each input required by the model and each output produced by it.

iii) The adoption of a set of 'feature types' describing the geo-temporal datasets required by, or output from, the numerical models. These are taken from those described by CSML [15].

An example metadata segment for a meteorological model instance from the USA (WRF-ARW) is given in Appendix II with only certain inputs and outputs shown. The precipitation parameter, lwe_thickness_of_precipitation_amount, adopted by the experiment above is given here and provides an example of key 'output' interface metadata as follows:

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: Grid Series
  Position: 8.0,44.0;8.0,45.0;9.0,45.0;9.0,44.0
  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
  Maximum Timestep Interval: 900s
  Minimum Timestep Interval: 3600s

Figure 3.8: Precipitation output metadata for WRF-ARW

A corresponding 'input' metadata element from a European hydrological model (RIBS) is given below. The WRF-ARW precipitation output is to be input into RIBS across an interface described by these two elements.

Input
  Name: Precipitation
  Description: Spatially distributed fields of rainfall
  Format: NetCDF CF
  Mandatory: true
  Feature Type: Grid Series

Position: 8.8,44.3;8.8,44.4;9.0,44.4;9.0,44.3
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: 1800s
Maximum Timestep Interval: 3600s

Figure 3.9: Precipitation input metadata for an hydrological model

So how well do these 'output' and 'input' metadata elements describe this interface and what options exist for validating the connection just using the metadata?

The authors consider that there are three fundamental aspects which need to be considered when validating such potential interfaces:

Parameter Validation

Is the parameter offered by the supplying model the same as that expected by the receiving model? Parameters can appear to be describing the same phenomenon, yet can be derived very differently. In this case, there is a direct corollary from the previous interoperability experiment. The well defined and understood parameter, lwe_thickness_of_precipitation_amount, is that provided and expected by the two models and is easily incorporated into the metadata. Here we depend upon the ground work which has provided the CF Standard Names controlled vocabulary.

However, the *name* of the input and output elements is 'precipitation'. This term is ambiguous and, although would be valid in this example, cannot be used to validate such a connection in general. The precision of the controlled vocabulary is not necessarily applied to this metadata element.

Spatial Validation

Do the two models represent phenomenon against the same geographical region? Moreover, do the datasets describe the data according to the same geospatial structure? The metadata elements relevant here are 'feature type' and 'position'. In this example, both datasets are

described as having a 'grid series' feature type, but it is not clear what is being described by 'position'. More metadata is necessary to perform anything other than a rudimentary assessment of the spatial aspects of this connection and that is assuming that 'position' is understood for each different feature type expressed.

Temporal Validation

Do the two models represent phenomenon against the same time period? Are the timesteps compatible between the models? Four metadata elements provide a reasonably rich description of this aspect: Time Range, Timestep Type, Minimum Timestep Interval and Maximum Timestep Interval. It can be easily checked that the same time range is being described as long as a standard representation of a time instant is used, as in this example. The other elements provide an overall understanding of the timestepping strategy in each model, but are not specific enough for definite validation without further information. For example, does the receiving model require an input timestep that matches its own internal timestep? Is the output of the receiving model expected to go on beyond that of the supplying model?

3.4.3 Conclusions

Overall, this experiment demonstrates that the metadata standard adopted provides good quality 'discovery' metadata. Users can discover models that would be of interest to them by searching general metadata items as well as assessing the parameters provided and the spatial and temporal coverages.

The performance of the metadata standard to support 'use' of the models in this experiment and, in particular, interfacing between the models can be summarised as follows:

- **Parameter validation** can be performed with precision using controlled vocabularies such as CF Standard Names. Validation should be performed against the 'parameter name' metadata element (and, indeed, the unit) and not against the input or output 'name' which would not be subject to the controlled vocabulary.
- **Spatial validation** can be performed to a rudimentary level as long as the use of 'feature type' and 'position' is understood and correctly applied.
- **Temporal validation** can also be performed to a rudimentary level, particularly assessing the temporal intervals used, with the metadata elements provided.

- It should also be noted that inputs and outputs can be denoted mandatory or optional using this metadata standard. If an interfaced output is given as optional and the associated input mandatory then, clearly, this should be identified and flagged.

Further spatial and temporal validation could be achieved with more extensive metadata or by analysing data files directly, although the latter would depend on the existence of the data files which is not assumed by the metadata. For example, interoperability of grids could be achieved through denoting, as part of the metadata, the position of each grid node or cell. This would lead to potentially very large and complex metadata files whose usability would deteriorate exponentially with each level of complexity.

The authors consider that full validation and application of interfaces should be performed by modellers who are familiar with the functional (and not necessarily the technical) requirements of their models. Metadata standards such as that analysed by this experiment can strongly support discovery of models and support use, particularly with respect to interfaces, to a rudimentary level.

3.5 Using gUSE against XSEDE and PRACE

The purpose of this experiment is to test the interoperability between XSEDE and PRACE by testing job submission through the workflow tool SCI-BUS [16] / gUSE [17].

3.5.1 Method

The SCI-BUS project aims at creating a new science gateway customisation methodology based on the general purpose gUSE/WS-PGRADE portal family (see figure 3.10 below). The SCI-BUS project provides both a portlet repository and an application repository. As a first interoperability experiment between XSEDE and PRACE, Andrew Grimshaw (University of Virginia, visiting professor at LMU) and the Hungarian Laboratory of Parallel and Distributed Systems (MTA SZTAKI) attempted to access XSEDE resources from the European scientific gateway SCI-BUS.

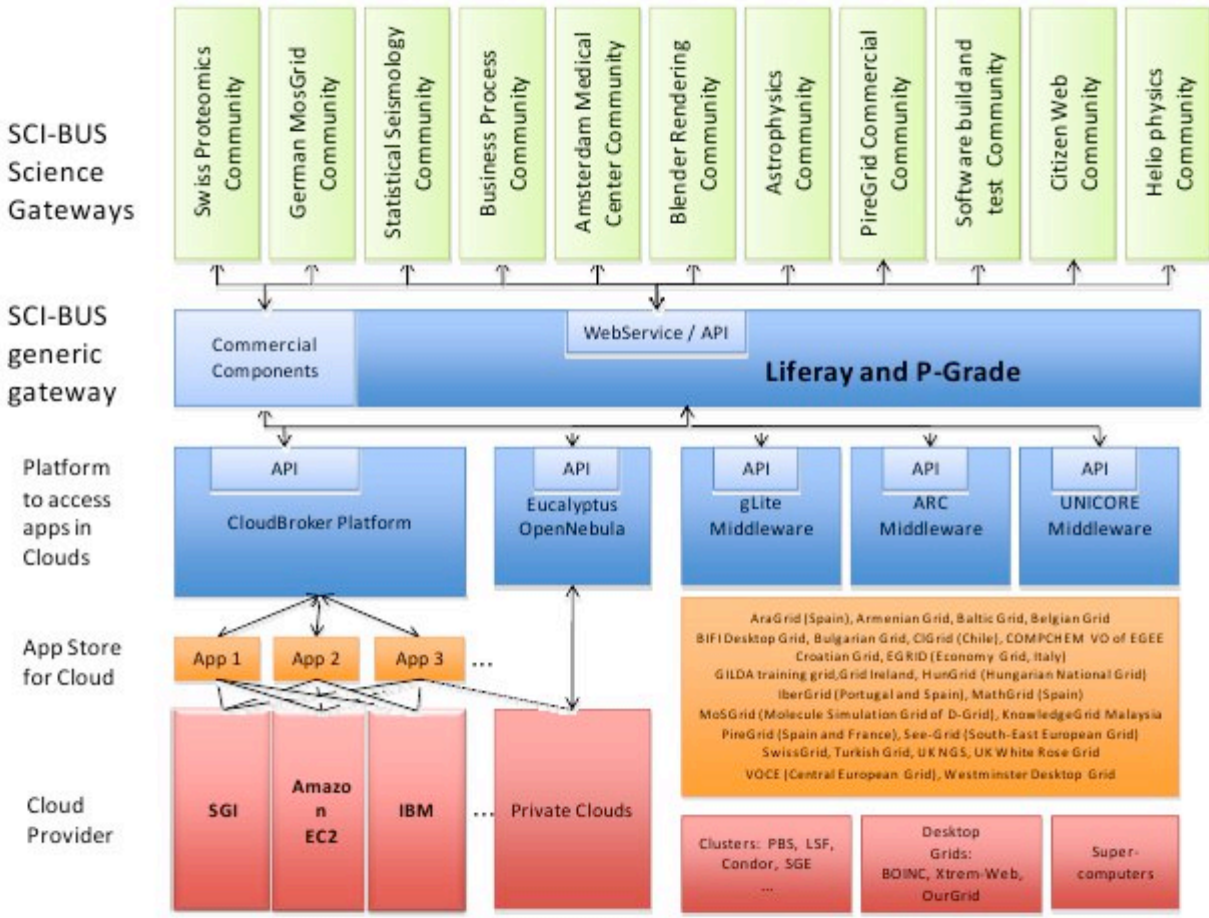


Figure 3.10: SCI-BUS Architecture

Access to XSEDE resources was performed by developing a simple job submission plugin to gUSE that would interact with the XSEDE Execution Management Services (BESs and gridqueues). As a first step a simple experiment was conducted to have the European site (Hungary) submit jobs to the XSEDE EMS and test the basic functionality. These tests were executed on March 26, 2014. In particular, the steps were the following:

1. The European site gets an account on the Global Federated File System (GFFS) GFFS.eu at /users/xcg.virginia.edu
2. The European site downloads the GFFS.eu installer onto their Linux desktop.
3. The European site installs GFFS.eu on their desktop.
4. The European site logs in using their identity.
5. Create a "hostname" job and run from the GUI, staging output into the GFFS directory.

6. Run a "hostname" JSDL job from the command line.
7. Perform a parameter sweep.

Further to this initial set of experiments, further discussions took place throughout the course of the project between DRIHM2US project members and representatives from SCI-BUS and XSEDE. This resulted in the specification of an ad-hoc gUSE plugin to allow European infrastructure access to XSEDE resources. In this way, the usage of resources belonging to both sides of the Atlantic can be made feasible from a single workflow source.

This plugin was made available within gUSE version 3.6.7, released on August 15th 2014. Performance of the experiment, therefore, required an upgrade from the current version in use 3.6.3, released in March 2014. Following completion of the upgrade, access to XSEDE resources was arranged to allow the commencement of the experiments.

3.5.2 Results

The initial tests all succeeded except step 7, the parameter sweep. The European site successfully achieved a GFFS account which was downloaded and installed. The log in proceeded successfully. It was also possible to create a "hostname" job and run it from the command line. However the parameter sweep failed either due to typing errors (later corrected) or because of certificate problems.

It was decided to incorporate one XSEDE resource into the portal which had been configured for the companion DRIHM project [10]. Accordingly, a job was submitted between these two infrastructures successfully: the DRIHM portal was able to access the XSEDE resource to complete the processing.

3.5.3 Conclusions

This experiment shows that, in principle, XSEDE resources can be accessed from the SCI-BUS gateway as it has been deployed on the DRIHM2US parent project, DRIHM [10]. The reverse, that of accessing PRACE resources from within XSEDE science gateways, still has to be demonstrated and more complex workflow experiments have to be conducted to show transparent, two-way access between these environments.

4 Summary

The set of interoperability experiments has been devised to cover a variety of component parts of any potential e-Science infrastructure. Each experiment acts as a typical use case and is intended to provide specific depth in the area to which it relates. Using the definition of interoperability agreed here, the experiments completed to-date have shown:

- That two well-defined and widely adopted data / modelling standards are strongly interoperable out-of-the-box, but not without adaptation at the interface between them.
- That numerical modelling applications developed separately can be successfully coupled, however this has not been shown to be feasible without re-engineering at least one of the components.
- The strength of controlled vocabularies as applied to a specific scientific domain (i.e. meteorology) but that each is not immediately transferrable to another scientific domain (i.e. hydrology).
- That metadata standards can provide a strong degree of interoperability, but only to a certain level of detail.
- The ability to access XSEDE resources in the USA from a European developed and hosted workflow engine.

In general and in particular for the completed experiments based on international standards, interoperability has been demonstrated to be more of an issue between scientific domains (e.g. meteorology, hydrology, hydraulics) than it is between Europe and the USA. This is in no small measure due to the fostering of interoperability standards though the international standards bodies and domain specific groupings.

DRIHM2US

5 Acronyms and References

5.1 Acronyms and Abbreviations

Acronym / Abbreviation	Definition
CF	Climate and Forecasting
CUAHSI	Consortium of Universities for the Advancement of Hydrologic Science, Inc.
DoW	Description of Work
DRIHM	Distributed Research Infrastructure for Hydro-Meteorology
DRIHM2US	Distributed Research Infrastructure for Hydro-Meteorology to United State of America
DWG	Domain Working Group
EMS	Execution Management Services
GFFS	Global Federated File System
GUI	Graphical User Interface
gUSE	grid and cloud Science Gateway
HIS	Hydrologic Information System
HMR	Hydro-Meteorological Research
ICT	Information and Communications Technology
JSDL	Job Submission Description Language
LMU	Ludwig Maximilians Universitat
Meso-NH	Mesoscale Non-hydrostatic Model
OGC	Open Geospatial Consortium
OpenMI	Open Modelling Interface
PRACE	Partnership for Advanced Computing in Europe
RIBS	Real-time Interactive Basin Simulator
SCIHM	Standards-Based Cyberinfrastructure for Hydro-meteorology
SCI-BUS	SCIentific gateway Based User Support
WaterML	Water Markup Language

WRF ARW	Weather Research and Forecasting Advanced Research
WRF NMM	Weather Research and Forecasting Nonhydrostatic Mesoscale Model
XSEDE	Extreme Science and Engineering Discovery Environment

5.2 References

[1]. Parodi, A. et al.: DRIHM2US Description of Work (DoW), 2012.

[2]. Hooper, R. et. Al.: About SCIHM, http://www.scihm.org/, accessed 4th April 2014.

[3]. IEEE: IEEE 100 - The Authoritative Dictionary of IEEE Standard Terms, 7th Edition. IEEE Standards Information, Network, ISBN 0-7381-2601-2, 2000

[4]. M. Riedel et al.: Interoperation of World-wide Production e-Science Infrastructures. In Concurrency and Computation: Practice and Experience, volume 21, pages 961–990, 2009.

[5]. WaterML2 A Global Standard for Hydrological Time Series, http://www.waterml2.org/ , accessed 28th March 2014.

[6]. OpenMI Association: What is OpenMI https://sites.google.com/a/openmi.org/home/new-to-openmi#TOC-What-is-OpenMI-, accessed 28th March 2014.

[7]. Harpham, Q. et al.: Welcome to FluidEarth, http://fluidearth.net/default.aspx , accessed 28th March 2014.

[8]. OpenMI Association: Source Forge OpenMI Project, http://sourceforge.net/projects/openmi/ , accessed 28th March 2014.

[9]. Harpham, Q. et al.: DRIHM2US D3.3, Domain Expert Networking Report, 2014.

[10]. Parodi, A. et al.: DRIHM Description of Work (DoW), 2011.

[11]. CUAHSI: CUAHSI Water Data Centre, http://his.cuahsi.org/ , accessed 28th March 2014.

[12]. CF Standard Name Table, http://cf-pcmdi.llnl.gov/documents/cf-standard-names/, Accessed 2013.

[13]. CUAHSI: Hydrologic Ontology for Discovery, http://his.cuahsi.org/ontologyfiles.html, accessed 1st April 2014.

[14]. Hughes, A., Harpham, Q., Riddick, A., Royse K., Singh A.: Meta-model: Ensuring the widespread access to metadata and data for environmental models - Scoping Report. British Geological Survey External Report, OR/13/042. 47pp, 2013.

[15]. Lowe, D. et. al: Climate Science Modelling Language v3.0, http://csml.badc.rl.ac.uk/, accessed 4th April 2014.

[16]. SCI-BUS Project: http://www.sci-bus.eu/de/home, accessed 28th March 2014.

[17]. Balasko, A. and Farkas, Z.: gUSE grid and cloud science gateway, http://sourceforge.net/projects/guse, accessed 26th March 2014.

Appendix I – Sample WaterML2 File


```xml
<?xml version="1.0" encoding="utf-16"?>
<wml2:Collection xsi:schemaLocation="http://www.opengis.net/waterml/2.0
http://schemas.opengis.net/waterml/2.0/waterml2.xsd" gml:id="generated_collection_doc"
xmlns:wml2="http://www.opengis.net/waterml/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wml="http://www.cuahsi.org/waterML/1.1/" xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:om="http://www.opengis.net/om/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:op="http://schemas.opengis.net/op"
xmlns:sf="http://www.opengis.net/sampling/2.0" xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"
xmlns:sam="http://www.opengis.net/sampling/2.0" xmlns:wml1_0="http://www.cuahsi.org/waterML/1.0/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gmd="http://www.isotc211.org/2005/gmd"
xmlns:gco="http://www.isotc211.org/2005/gco" xmlns:gss="http://www.isotc211.org/2005/gss">
<wml2:metadata>
<wml2:DocumentMetadata gml:id="doc_md">
<wml2:generationDate>2011-01-01T00:00:00Z</wml2:generationDate>
<wml2:version xlink:href="http://www.opengis.net/waterml/2.0" xlink:title="WaterML 2.0 RFC" />
<wml2:generationSystem>XSLT Translation from WaterML1.1 response
document</wml2:generationSystem>
</wml2:DocumentMetadata>
</wml2:metadata>
<wml2:localDictionary>
<gml:Dictionary gml:id="phenomena">
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/waterml2/dictionaries/">phenomena</gml:identifier>
<gml:dictionaryEntry>
<gml:Definition gml:id="CIMA1.RF_1h">
<gml:description xlink:href="CIMA1:RF_1h" xlink:title="Precipitation" />
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/wml/variable">CIMA1:RF_1h</gml:identifier>
<gml:name codeSpace="http://hiscentral.cuahsi.org/ontology/">Precipitation</gml:name>
<gml:name codeSpace="http://hiscentral.cuahsi.org/wml/vocabulary/CIMA1">Precipitation</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
</gml:Dictionary>
</wml2:localDictionary>
<wml2:localDictionary>
<gml:Dictionary gml:id="quality">
<gml:identifier codeSpace="http://www.cuahsi.org/waterml2/dictionaries/">quality</gml:identifier>
<gml:dictionaryEntry>
<gml:Definition gml:id="qclevel-0">
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/wml/qualityControlLevelCode">0</gml:identifier>
```

```
<gml:name codeSpace="http://hiscentral.cuahsi.org/wml/qualityControlLevelCode">Raw data</gml:name>
<gml:remarks>Raw and unprocessed data and data products that have not undergone quality control.  Depending on the variable, data type, and data transmission system, raw data may be available within seconds or minutes after the measurements have been made. Examples include real time precipitation, streamflow and water quality measurements.</gml:remarks>
</gml:Definition>
</gml:dictionaryEntry>
</gml:Dictionary>
</wml2:localDictionary>
<wml2:localDictionary>
<gml:Dictionary gml:id="censorCode">
<gml:identifier codeSpace="http://www.cuahsi.org/waterml2/dictionaries/">censorCode</gml:identifier>
<gml:dictionaryEntry>
<gml:Definition gml:id="censorCode-nc">
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/wml/censored">nc</gml:identifier>
<gml:name codeSpace="http://hiscentral.cuahsi.org/wml/censored">not censored</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
</gml:Dictionary>
</wml2:localDictionary>
<wml2:localDictionary>
<gml:Dictionary gml:id="method">
<gml:identifier codeSpace="http://www.cuahsi.org/waterml2/dictionaries/">method</gml:identifier>
<gml:dictionaryEntry>
<gml:Definition gml:id="methodCode-CIMA1-1">
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/wml/method">CIMA1-1</gml:identifier>
<gml:name codeSpace="http://hiscentral.cuahsi.org/wml/method">water precipitation measured through a 1000 cm? calibrated mouth flowing in a tipping-bucket container</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
</gml:Dictionary>
</wml2:localDictionary>
<wml2:samplingFeatureMember>
<wml2:MonitoringPoint gml:id="CIMA1-1003">
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/network/CIMA1">1003</gml:identifier>
<gml:name>S.Alberto</gml:name>
<sam:sampledFeature xlink:title="Unsepecified Sampled Feature" xlink:role="http://hiscentral.cuahsi.org/wml/site" xlink:href="urn:ogc:def:nil:OGC:unknown" />
<sam:parameter>
<om:NamedValue>
<om:name xlink:href="http://hiscentral.cuahsi.org/wml/siteProperty/elevation_m/" xlink:title="elevation in meters" />
<om:value xsi:type="xsd:string">637m</om:value>
```

```
</om:NamedValue>
</sam:parameter>
<sams:shape>
<gml:Point gml:id="CIMA1-1003_pos">
<gml:pos srsName="">44.437222 9.106667</gml:pos>
</gml:Point>
</sams:shape>
<wml2:monitoringType xlink:href="http://www.cuahsi.org/waterml2/siteType/surfaceWater" xlink:title="Surface
Water" />
</wml2:MonitoringPoint>
</wml2:samplingFeatureMember>
<wml2:observationMember>
<om:OM_Observation gml:id="observation-1">
<om:metadata>
<wml2:ObservationMetadata>
<gmd:contact gco:nilReason="inapplicable" />
<gmd:dateStamp gco:nilReason="inapplicable" />
<gmd:locale>
<gmd:PT_Locale>
<gmd:languageCode>
<gmd:LanguageCode codeList="LanguageCode" codeListValue="EN-US">English-United
States</gmd:LanguageCode>
</gmd:languageCode>
<gmd:characterEncoding>
<gmd:MD_CharacterSetCode codeList="MD_CharacterSetCode" codeListValue="utf8">UTF
8</gmd:MD_CharacterSetCode>
</gmd:characterEncoding>
</gmd:PT_Locale>
</gmd:locale>
<gmd:identificationInfo>
<gmd:MD_DataIdentification id="ID0ELAAC">
<gmd:citation>
<gmd:CI_Citation>
<gmd:title>
<gco:CharacterString>Continuous monitoring data collected by CIMA Research Foundation on behalf of the Italian
Civil Protection Department</gco:CharacterString>
</gmd:title>
<gmd:date gco:nilReason="inapplicable" />
</gmd:CI_Citation>
</gmd:citation>
<gmd:abstract>
<gco:CharacterString>Continuous monitoring data collected by the Italian Civil Protection Department RAWS network
(remotely automated weather stations)</gco:CharacterString>
```

```
</gmd:abstract>
<gmd:pointOfContact>
<gmd:CI_ResponsibleParty>
<gmd:individualName>
<gmd:LocalisedCharacterString locale="#EN-US">Luca Molini</gmd:LocalisedCharacterString>
</gmd:individualName>
<gmd:organisationName>
<gmd:LocalisedCharacterString locale="#EN-US">CIMA research Foundation</gmd:LocalisedCharacterString>
</gmd:organisationName>
<gmd:contactInfo>
<gmd:CI_Contact>
<gmd:phone>
<gmd:CI_Telephone>
<gmd:voice>
<gco:CharacterString>+39-019230271</gco:CharacterString>
</gmd:voice>
</gmd:CI_Telephone>
</gmd:phone>
<gmd:address>
<gmd:CI_Address>
<gmd:deliveryPoint>
<gco:CharacterString>Via Armando Magliotto, 2
,Savona, Italy 17100</gco:CharacterString>
</gmd:deliveryPoint>
<gmd:electronicMailAddress>
<gco:CharacterString>luca.molini@cimafoundation.org</gco:CharacterString>
</gmd:electronicMailAddress>
</gmd:CI_Address>
</gmd:address>
<gmd:onlineResource>
<gmd:CI_OnlineResource>
<gmd:linkage>
<gmd:URL>-</gmd:URL>
</gmd:linkage>
</gmd:CI_OnlineResource>
</gmd:onlineResource>
</gmd:CI_Contact>
</gmd:contactInfo>
<gmd:role>
<gmd:CI_RoleCode codeList="ISOTC211/19115"
codeSpace="http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#CI_RoleCode"
codeListValue="principalInvestigator" />
</gmd:role>
```

```xml
</gmd:CI_ResponsibleParty>
</gmd:pointOfContact>
<gmd:language>
<gco:CharacterString>EN-US</gco:CharacterString>
</gmd:language>
</gmd:MD_DataIdentification>
</gmd:identificationInfo>
<wml2:sampledMedium xlink:href="http://hiscentral.cuahsi.org/wml/sampledMedium" xlink:title="Air" />
</wml2:ObservationMetadata>
</om:metadata>
<om:phenomenonTime>
<gml:TimePeriod gml:id="phen_time-1">
<gml:beginPosition>2011-11-04T01:00:00</gml:beginPosition>
<gml:endPosition>2011-11-05T00:00:00</gml:endPosition>
</gml:TimePeriod>
</om:phenomenonTime>
<om:resultTime>
<gml:TimeInstant gml:id="eor-1">
<gml:timePosition>2011-11-05T00:00:00</gml:timePosition>
</gml:TimeInstant>
</om:resultTime>
<om:procedure>
<wml2:ObservationProcess gml:id="process-1">
<gml:description>water precipitation measured through a 1000 cm? calibrated mouth flowing in a tipping-bucket
container</gml:description>
<gml:identifier codeSpace="http://hiscentral.cuahsi.org/wml/method">1</gml:identifier>
<wml2:processType xlink:href="http://www.opengis.net/def/waterml/2.0/processType/Sensor" xlink:title="water
precipitation measured through a 1000 cm? calibrated mouth flowing in a tipping-bucket container" />
<wml2:aggregationDuration>PT1H</wml2:aggregationDuration>
<wml2:processReference xlink:href="http://www.cae.it/en/sens_pluviometro25.php" xlink:title="water precipitation
measured through a 1000 cm? calibrated mouth flowing in a tipping-bucket container" />
<wml2:parameter>
<om:NamedValue>
<om:name xlink:title="valueType" xlink:href="http://hiscentral.cuahsi.org/wml/valueType" />
<om:value xsi:type="xsd:string">Field Observation</om:value>
</om:NamedValue>
</wml2:parameter>
<wml2:parameter>
<om:NamedValue>
<om:name xlink:title="dataType" xlink:href="http://hiscentral.cuahsi.org/wml/dataType" />
<om:value xsi:type="xsd:string">Average</om:value>
</om:NamedValue>
</wml2:parameter>
```

```
<wml2:parameter>
<om:NamedValue>
<om:name xlink:title="noDataValue" xlink:href="http://hiscentral.cuahsi.org/wml/noDataValue" />
<om:value xsi:type="xsd:string">-9999</om:value>
</om:NamedValue>
</wml2:parameter>
<wml2:parameter>
<om:NamedValue>
<om:name xlink:title="http://hiscentral.cuahsi.org/wml/sampleMedium" xlink:href="" />
<om:value xsi:type="xsd:string">Air</om:value>
</om:NamedValue>
</wml2:parameter>
<wml2:parameter>
<om:NamedValue>
<om:name xlink:title="speciation" xlink:href="http://hiscentral.cuahsi.org/wml/speciation" />
<om:value xsi:type="xsd:string">Not Applicable</om:value>
</om:NamedValue>
</wml2:parameter>
</wml2:ObservationProcess>
</om:procedure>
<om:observedProperty xlink:href="#CIMA1.RF_1h" xlink:title="Precipitation" />
<om:featureOfInterest xlink:href="#CIMA1-1003" xlink:title="S.Alberto1003" />
<om:result>
<wml2:MeasurementTimeseries gml:id="_TS-1">
<wml2:metadata>
<wml2:MeasurementTimeseriesMetadata>
<wml2:temporalExtent xlink:href="#phen_time-1" />
<wml2:cumulative>true</wml2:cumulative>
</wml2:MeasurementTimeseriesMetadata>
</wml2:metadata>
<wml2:defaultPointMetadata>
<wml2:DefaultTVPMeasurementMetadata>
<wml2:quality xlink:href="http://www.opengis.net/def/waterml/2.0/quality/good" xlink:title="not censored" />
<wml2:qualifier>
<swe:Text definition="#methodCode-CIMA1-1">
<swe:value>water precipitation measured through a 1000 cm? calibrated mouth flowing in a tipping-bucket
container</swe:value>
</swe:Text>
</wml2:qualifier>
<wml2:processing xlink:href="#qclevel-0" xlink:title="Raw data" />
<wml2:source xlink:href="#ID0ELAAC" />
<wml2:uom code="" />
```

```
<wml2:interpolationType xlink:href="http://www.opengis.net/def/timeseriesType/WaterML/2.0/AverageSucc"
xlink:title=" Average in succeeding interval&#xA;                    ($dataTypeName) " />
<wml2:aggregationDuration>PT1H</wml2:aggregationDuration>
</wml2:DefaultTVPMeasurementMetadata>
</wml2:defaultPointMetadata>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2011-11-04T00:00:00</wml2:time>
<wml2:value>10.0</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2011-11-04T05:00:00</wml2:time>
<wml2:value>300.0</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2011-11-04T15:00:00</wml2:time>
<wml2:value>10.0</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2011-11-05T00:00:00</wml2:time>
<wml2:value>10.0</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
</wml2:MeasurementTimeseries>
</om:result>
</om:OM_Observation>
</wml2:observationMember>
</wml2:Collection>
```

Appendix II – WRF-ARW Metadata

An example of an un-encoded meteorological model instance metadata segment: **WRF ARW**

Citation

Title: WRF ARW

Creation Date: 2008-03-04

Abstract: The Weather Research and Forecasting (WRF) -Advanced Research WRF (ARW) model is a fully compressible, nonhydrostatic model (with a hydrostatic option). Its vertical coordinate is a terrain-following hydrostatic pressure coordinate. The grid staggering is the Arakawa C-grid. The model uses higher-order numerics. This includes the Runge-Kutta 2nd- and 3rd-order time integration schemes, and 2nd- to 6th-order advection schemes in both horizontal and vertical directions. It uses a time-split small step for acoustic and gravity-wave modes. The dynamics conserves scalar variables.

Point of Contact

Custodian Organisation Name: CIMA Research Foundation

Custodian Online Resource: www.cimafoundation.org

Responsible Individual

Name: Antonio Parodi

Organisation: CIMA Research Foundation

Position:

Address and Email: antonio.parodi@cimafoundation.org

Descriptive Keywords[1]: Rainfall, Runoff, Model

Topic Category Code: Climatology, meteorology, atmosphere

Date Stamp: 2011-12-10T13:10:52

Reference System: urn:ogc:def:crs:EPSG::3857

Extent: 8.0,44.0;8.0,45.0;9.0,45.0;9.0,44.0

Programming Language: Fortran 90 or 95, C

Supported Platforms: Linux

Spatial Dimension: 3

Source Code URI: http://box.mmm.ucar.edu/wrf/users/downloads.html

Executable URI:

Documentation URI: http://www.mmm.ucar.edu/wrf/OnLineTutorial/Introduction/index.html

OpenMI Status: Does not implement OpenMI

OpenMI Version: none

Number of Processors:

Typical Run Time

Duration: 1000

Unit: second

Input

Name: Pressure

Description:

Format:

Mandatory: Yes

Feature Type: Grid Series

Position:

Parameter

Name: *as defined in CF Standard Names*

Unit: *as defined in CF Standard Names*

Time Range:

Timestep Type: Regular

Maximum Timestep Interval:

Minimum Timestep Interval:

List other inputs here……

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: 8.0,44.0;8.0,45.0;9.0,45.0;9.0,44.0

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

Maximum Timestep Interval: 900s

Minimum Timestep Interval: 3600s

List other outputs here……