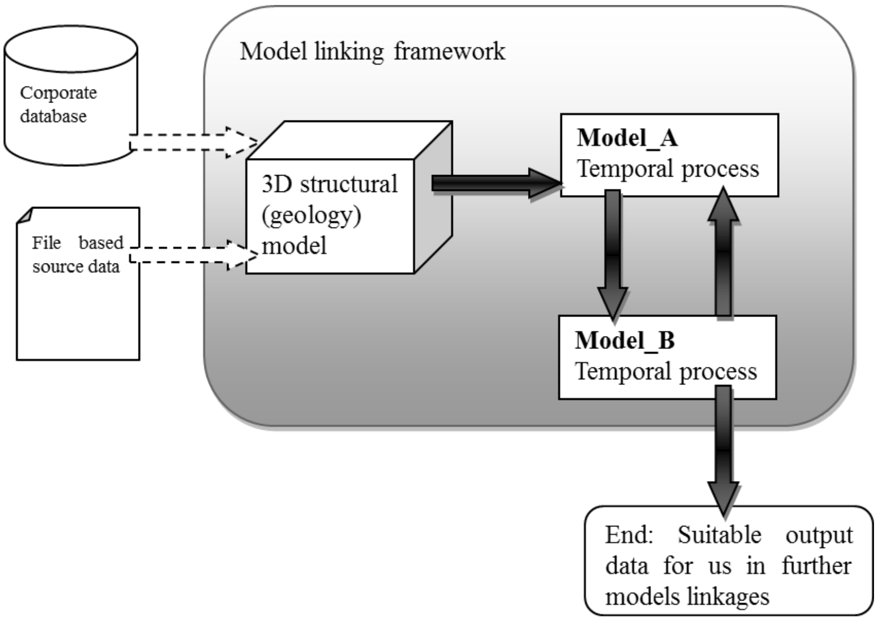
British Geological Survey
NATURAL ENVIRONMENT RESEARCH COUNCIL

# Couplers for linking environmental models: Scoping study and potential next steps

Environmental Modelling Programme

Internal Report OR/14/022



BRITISH GEOLOGICAL SURVEY

ENVIRONMENTAL Modelling PROGRAMME

INTERNAL REPORT OR/14/022

# Couplers for linking environmental models: Scoping study and potential next steps

Barkwith A.K.A.P., Pachocka M., Watson C. and Hughes A.G.



*Keywords*

Report; keywords.

Cover picture details, delete if no cover picture.

*Bibliographical reference*

BARKWITH A.K.A.P, PACHOCKA M., Watson C., Hughes A.G.. 2014. Couplers for linking environmental models: Scoping study and potential next steps. *British Geological Survey Internal Report*, OR/14/022. 49pp.



Keyworth, Nottingham British Geological Survey 2014

## BRITISH GEOLOGICAL SURVEY

The full range of our publications is available from BGS shops at Nottingham, Edinburgh, London and Cardiff (Welsh publications only) see contact details below or shop online at www.geologyshop.com

The London Information Office also maintains a reference collection of BGS publications, including maps, for consultation.

We publish an annual catalogue of our maps and other publications; this catalogue is available online or from any of the BGS shops.

*The British Geological Survey carries out the geological survey of Great Britain and Northern Ireland (the latter as an agency service for the government of Northern Ireland), and of the surrounding continental shelf, as well as basic research projects. It also undertakes programmes of technical aid in geology in developing countries.*

*The British Geological Survey is a component body of the Natural Environment Research Council.*

*British Geological Survey offices*

**BGS Central Enquiries Desk**

Tel 0115 936 3143 Fax 0115 936 3276
email enquiries@bgs.ac.uk

**Environmental Science Centre, Keyworth, Nottingham NG12 5GG**

Tel 0115 936 3241 Fax 0115 936 3488
email sales@bgs.ac.uk

**Murchison House, West Mains Road, Edinburgh EH9 3LA**

Tel 0131 667 1000 Fax 0131 668 2683
email scotsales@bgs.ac.uk

**Natural History Museum, Cromwell Road, London SW7 5BD**

Tel 020 7589 4090 Fax 020 7584 8270
Tel 020 7942 5344/45 email bgslondon@bgs.ac.uk

**Columbus House, Greenmeadow Springs, Tongwynlais, Cardiff CF15 7NE**

Tel 029 2052 1962 Fax 029 2052 1963

**Maclean Building, Crowmarsh Gifford, Wallingford OX10 8BB**

Tel 01491 838800 Fax 01491 692345

**Geological Survey of Northern Ireland, Colby House, Stranmillis Court, Belfast BT9 5BF**

Tel 028 9038 8462 Fax 028 9038 8461

www.bgs.ac.uk/gsni/

*Parent Body*

**Natural Environment Research Council, Polaris House, North Star Avenue, Swindon SN2 1EU**

Tel 01793 411500 Fax 01793 411501
www.nerc.ac.uk

Website www.bgs.ac.uk
Shop online at www.geologyshop.com

# Acknowledgements

Thanks to Bryan Lawrence who freely gave an early draft of his paper entitled “Bridging Communities: Technical Concerns for Integrating Environmental Models” and which helped in developing the ideas.

# Contents



**FIGURES**



**TABLES**

# Summary

This report scopes out what couplers there are available in the hydrology and atmospheric modelling fields. The work reported here examines both dynamic runtime and one way file based coupling. Based on a review of the peer-reviewed literature and other open sources, there are a plethora of coupling technologies and standards relating to file formats. The available approaches have been evaluated against criteria developed as part of the DREAM project. Based on these investigations, the following recommendations are made:

- The most promising dynamic coupling technologies for use within BGS are OpenMI 2.0 and CSDMS (either 1.0 or 2.0)
- Investigate the use of workflow engines: Trident and Pyxis, the latter as part of the TSB/AHRC project "Confluence"
- There is a need to include database standards CSW and GDAL and use data formats from the climate community NetCDF and CF standards.
- Development of a "standard" composition which will consist of two process models and a 3D geological model all linked to data stored in the BGS corporate database and flat file format. Web Feature Services should be included in these compositions.

There is also a need to investigate other approaches in different disciplines: The Loss Modelling Framework, OASIS-LMF is the best candidate.

# 1 Introduction

## 1.1 BACKGROUND

To make Integrated Modelling (IM) work, a way of passing data between models is required and to facilitate this, standards are necessary. Two sets of standards are required: data standards for one way, static transfer of data and model runtime standards for dynamic coupling. For the former, BGS environmental modellers use some basic standards: dxf, CSV, etc. However, it would be useful to identify internationally accepted standards that could be used for data exchange, particularly for gridded data. In terms of exchange of data during model runtime, the current standard and implementation used at BGS is OpenMI. OpenMI was designed with the solution of the problem posed by the Water Framework Directive, that of simulating catchment processes in a holistic manner. Therefore, the main aim of the OpenMI as it is currently implemented is for flexibility. It may not be appropriate in a high performance computing environment. Given that BGS' requirements may change then it is necessary to identify and understand other standards or even approaches for linking models at runtime.

This report, therefore, focuses on the data standards for static and runtime coupling of numerical models used in the hydrological and atmospheric sciences. Included in this process are workflow engines, but approaches for other disciplines such as risk in the insurance industry and human health are not included.

## 1.2 THE NEED FOR COUPLERS

The need for interdisciplinary environmental modelling has become clear over the last decade as the evidence of the climate change has been growing stronger. Such modelling provides the means to study complex dynamics of the Earth system and thus aids finding ways to mitigate the impacts of the environmental change. In the year 2000, the Water Framework Directive was enacted, which recognised the need to implement integrated management strategies to address ever more rising and conflicting demands for water resources in a catchment. This problem is best addressed by adopting sound modelling approaches. Integrated modelling requires sharing and coupling models simulating different parts of the Earth system. The approach used to link such models is called "a coupler". While a large number of different couplers are currently in use by scientists, their basic functions remain the same, namely: coordinating the execution of the coupled models and managing data transfer between them (Valcke et al., 2012).

The technologies used for coupling models vary in the level of "intrusiveness", which can be defined as the amount of work required to make a component "couplable" (Lawrence et al., Manuscript). The coupling technologies can be divided into: monolithic, component-based, communication-based, and scheduled (Dunlap et al., 2013). The monolithic approach requires combining code from multiple models into one code (Dunlap et al., 2013). The component-based approach introduces the concept of standard interfaces. In this approach each model, called a "component", has: an interface to communicate with other models, a structure in compliance with predefined criteria, and performs a distinct function (Dunlap et al., 2013, Lu 2011). In communication and scheduled approaches models are independent (Dunlap et al., 2013, Lu 2011). The communication approach requires embedding library calls within the model's code for sending and receiving data (Dunlap et al., 2013). In the scheduled approach the output from one model is used as an input to the next one, thus the models do not affect each other during the execution (Dunlap et al., 2013).

The coupling technologies can be formally divided into: coupling libraries, coupling frameworks, and workflows (Lawrence et al., Manuscript, Dunlap et al., 2013). Libraries provide concrete solution fragments (Lawrence et al., Manuscript); they minimise the amount of code changes required to make a model couplable, typically allowing it to act as independent executable and merely to exchange data at appropriate locations and times (Dunlap et al., 2013). Frameworks use standard interfaces for communication with the components, which must comply with the interfaces' calling conventions (Dunlap et al., 2013). Consequently that components must be structured in accordance with a predefined architectural design (Dunlap et al., 2013). Workflow engines are non-intrusive tools that allow components to remain independent, solely coordinating the exchange of data (Lawrence et al., Manuscript). There are significant overlaps between the technologies and they are often used in tandem (Lawrence et al., Manuscript). Based on the level of integration between the components, the coupling can be defined as either "tight" or "loose" (Goodall et al., 2011). Summarising, while all couplers have the same basic functions they differ in the level of component standardisation, the way the components are called and exchange data, and the degree to which they are integrated.

A large number of coupling technologies were developed up to date, which seemingly appears to be a redundant effort. However, this is not the case as different approaches address different, often conflicting demands, like: generality, flexibility, ease of use, accuracy, and performance (Jagers 2010).

## 1.3 COUPLER USE CASES AND REQUIREMENTS GATHERED FROM BGS STAFF

In 2010 the BGS produced the Dream Scoping study report (Giles et al., 2010), as part of the research for this report a wide range of BGS scientists, responsible for answering questions raised by clients, were asked what they required from a model linkage solution, below are a selection of those responses:

> *"As a geologist focussed on the urban environment I want an environmental modelling platform to act as an effective communication tool, perhaps through visual representations of processes, so that others, including non-geologists, can better understand the model."*
>
> *"As a geoscience standards and property team member I want to be able to calculate the financial implications of varying sub-surface project options, for example 'where is the cheapest place to dig this tunnel?', so that our customers (& potential customers) understand the significance and benefits of sub-surface knowledge."*
>
> *"As a geophysicist I want an environmental modelling platform to handle high volumes of data traffic on a regular and ongoing basis, so that I can process real time data from the field or sensors, automatically model it and I & customers can view the results and identify trends."*
>
> *"As a flood analyst, I want to predict possible flood scenarios for the village over the next 24hours using various inputs such as rain fall, groundwater, water table levels, so that decision makers can be given the info necessary to decide whether the village should be evacuated."*

At the time of capturing these use cases the imagined solution was referred to as an environmental modelling platform and opinions varied greatly on how much functionality would be delivered through the new platform and what existing components would be re-used. Despite significant differences in opinion it was possible to identify a common set of desirable attributes that any solution should exhibit.

## 1.4 COMMONLY DESIRABLE MODEL COUPLING TECHNOLOGY ATTRIBUTES

There is an almost bewildering choice of methodologies, technologies and tools available to integrated environmental modelling (IEM) practitioners, however there are some concepts which we regard as desirable.

The IEM technologies used by the BGS should incorporate the following attributes:

- Ability to link models in a modular way, rather than developing a single piece of code (model) that incorporates data manipulation and scientific logic we should encourage developers to separate out these functions so that they can be used in more than one scenario.
- Visual workflow builders open up the world of linked model development to users with little to no programming experience. Although care should be taken to ensure that any assessment of the performance of a linked model solution fully considers the impact of technological implementation as well as scientific logic, this becomes difficult when the user does not fully understand how a technology works behind the scenes.
- It should be simple to capture the metadata required to describe scientific models, the data they require and any data outputs generated, in order to support model discovery and provide guidance on how to use the model(s).
- Coupling technologies which exhibit a low degree of invasiveness tend to have less of a negative impact on the performance of existing models, extensive alterations can lead to code divergence and may adversely affect the original model design or purpose. In addition, alterations made for one technology can limit model re-use in alternative technologies.
- Technologies with significant community support provide potential users with a confidence that help is at hand should it be needed. The BGS should pay particular attention to the technologies favoured by communities who specialise in those areas of science we wish to integrate with.
- And finally, a 'stable' or clearly versioned technology provides the user with a certain degree of certainty that doesn't exist with rapidly changing environments. Models and linked models can be assessed for their scientific value without the added confusion of a transient informatics platform. Although the technology should be stable, it is also desirable that there is an active, albeit separate, development path which helps to improve the technology in response to community needs.

## 1.5 STRUCTURE OF REPORT

The following sections of the report describe in detail the dynamic (run-time) approaches for atmospheric and hydrological approaches (Section 2), which is followed by a summary of data standards for one-way, static transfer of data (Section 3). Section 4 compares the different approaches and the findings of the report are summarised in Section 5 along with providing recommendations for the next stage of work.

# 2 Description of dynamic (run-time) approaches

## 2.1 ATMOSPHERIC

Runtime coupling of environmental models is important, to capture the many feedbacks that exist between Earth systems. This section of the report details coupling software used in the atmospheric sciences. Where the software has been used within a project, the coupling component tends to be formed from two distinct sections; the coupler, which communicates with different model components; and the modelling framework, the architecture in which the coupler can operate. As atmospheric systems are tightly coupled with the Earth surface, many of the coupling frameworks encompass land and ocean modelling components.

There is a commonality of the data transfer methods for many of the approaches to produce coupled systems. In general, an active component needs data from (get or pull), and provides data to (set or put), the coupler, while data driven components read data during runtime and then provide that data to the coupler. Set (put) is typically a non-blocking communication implying that the calling code does not wait for a set to complete before proceeding. Get (pull) is blocking, so the receiver may have to wait until a sender puts the requested data. Initialise, Run Finalise (IRF) is used to describe the life-cycle of a model component within the modelling framework (Figure 1). Initialise describes the internal state of a component (eg, opening a file for reading, or a creating a database connection), Run provides the implementation logic of the component where input is being transformed to output, and Finalise provides the notion of a final cleanup after model execution. Dynamic data exchange between model components usually occurs during the run phase. The Message Passing Interface (MPI) is another standardised method commonly employed in dynamic model coupling. MPI is a language-independent communications protocol used to program parallel computers, which supports point-to-point and collective communication.

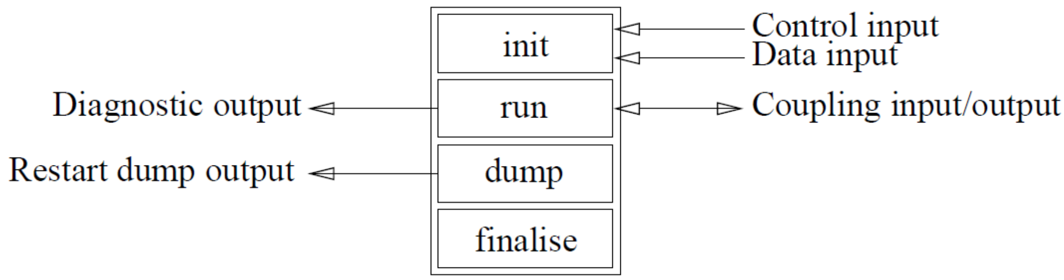



**Figure 1 – A typical dynamic interaction between an ensemble component using the IRF method.**

### 2.1.1 CESM -CPL 7 (Framework and Coupler)

#### 2.1.1.1 Overview

The Community Earth System Model (CESM) framework is used by researchers at the University Corporation for Atmospheric Research (UCAR) and the National Center for Atmospheric Research (NCAR) to couple land, sea, ice and atmospheric models using the CESM coupler CPL7 (Figure 2). The CESM replaces the previous Community Climate System Model (CCSM) modelling framework. CPL7 is designed to synchronise component time-stepping within the framework, manage component data communication, conservatively map data between component grids, and compute fluxes between components. While the processor configuration is relatively flexible and components can be run sequentially or concurrently, the sequencing of components in the driver (main CESM program) is fixed and independent of the processor layout. CESM components are called via the standard IRF method. The framework description used in this report is modified from Craig (2011).

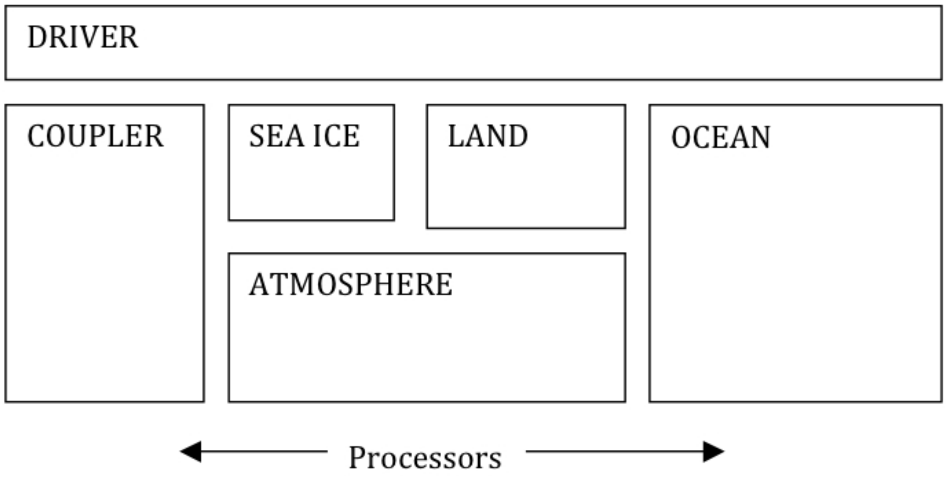



**Figure 2 – The basic CCSM framework with the CPL coupler timing controlled by the driver**

The CESM architecture is composed of a single executable with a high-level driver (Figure 2). The driver handles coupler sequencing, model concurrency, and communication of data between components. The driver directly calls the CPL7 coupler methods (for re-gridding, rearranging, merging, an atmosphere-ocean flux calculation, and diagnostics), which are run on a subset of processors essentially as a model component.

The standard CESM component model interfaces are based upon the ESMF design (See Section 2.2.12). Each component provides an IRF method with consistent arguments. As part of initialisation, an MPI communicator is passed from the driver to the component, and grid and decomposition information is passed from the component back to the driver. The driver and coupler acquire information about resolution, configurations, and processor layout at run-time from either a file or from communication with components.

In CESM, parts of the Model Coupling Toolkit (MCT) have been adopted at the driver-level, where they are used directly in the component IRF interfaces. In addition, MCT is also used for all data rearranging and re-gridding (interpolation) executed by the coupler.

The CESM driver manages the main clock in the system. That clock advances at the shortest coupling period and uses alarms to trigger component coupling and other events. In addition, the driver maintains a clock that is associated with each component. The standard implementation for grids in CESM has been that the atmosphere and land models are run on identical grids and the ocean and sea ice model are run on identical grids. An ocean model mask is used to derive a complementary mask for the land grid, such that for any given combination of atmosphere-land and ocean-ice grids there is a unique land mask. This approach for dealing with grids is still used a majority of the time in CESM, however it is possible to separate the atmosphere and land grids.

#### 2.1.1.2 Process

CESM consists of both data driven and active components. In general, an active component needs data from (*get* or *pull*), and provides data to (*set* or *put*), the coupler, while data driven components read data during runtime and then provide that data to the coupler. There are seven basic processor groups in the CESM framework associated with; the atmosphere, land, ocean, sea ice, land ice, coupler, and the global group. Each of the seven processor groups can be distinct, but that is not a requirement of the system.

System initialisation is relatively straight-forward. Firstly, the seven MPI communicators are computed in the driver. Then the atmosphere, land, ocean, sea ice, and land ice model initialisation-methods are called on the appropriate processor sets, an MPI communicator is sent and grid and decomposition information are passed back to the driver. Once the driver has all the grid and decomposition information from the components, various re-arrangers and re-gridding routines are initialised that will move data between processors, decompositions, and grids as needed at the driver level. The driver derives all MPI communicators at initialisation and passes them to the component models for use. There are two issues related to whether the component models run concurrently. The first is whether unique chunks of work are running on distinct processor sets. The second is the sequencing of this work in the driver. CESM driver sequencing has been implemented to maximize the potential amount of concurrency of work between different components. However, the active atmosphere model cannot run concurrently with the land and sea-ice models.

#### 2.1.1.3 Data Exchange

Active data exchange within the CESM may only occur through the coupler. Typically two dimensional gridded datasets are passed. Exchanged data must conform to a specific unit convention. A list of time variant and time invariant data exchange items may be found in Kauffman et al., (2004). Exchanged items are passed to the coupler as a set of output fields, where fluxes may be calculated. The coupler then provides a set of input fields for the receiving system component to read at the following timestep. Input flux fields handled by the system components are understood to fall into a set interval, otherwise the conservation of fluxes is lost. For example, if the atmospheric component communicates once per hour, but takes four internal time steps, the hourly precipitation received by the atmospheric component needs to be averaged internally over the four hours.

### 2.1.2 OASIS3-MCT_2.0 (Framework and Coupler)

#### 2.1.2.1 Overview

The framework description for OASIS3-MCT_2.0 is modified from Valcke et al., (2013). In 1991, CERFACS started the development of a software interface to couple existing ocean and atmosphere numerical General Circulation Models. OASIS3-MCT_2.0 is interfaced with the MCT, developed by the Argonne National Laboratory in the USA. MCT implements fully parallel re-gridding and parallel distributed exchanges of the coupling fields based on pre-computed re-gridding weights and addresses. MCT has proven parallel performance and is also the underlying coupling software used in the CESM.

Low model component intrusiveness, portability and flexibility were key concepts when designing OASIS3-MCT_2.0. The software itself may be envisaged as a coupling library that needs to be linked to the component models, the main function of which is to interpolate and exchange the coupling fields between them to form a coupled system. OASIS3-MCT_2.0 supports coupling of 2D logically-rectangular fields but 3D fields and 1D fields expressed on unstructured grids are also supported using a one dimension degeneration of the structures.

#### 2.1.2.2 Process

The employment of the MCT allows all transformations, including re-gridding, to be executed in parallel. All couplings are executed in parallel directly between the components via MPI. In addition to this, OASIS3-MCT_2.0 also supports file input and output (I/O) using the NetCDF file standard. To communicate with another model, or to perform I/O actions, a component needs to include specific calls to the OASIS3-MCT_2.0 coupling library. Information, about the resolution, configurations, and processor layout at run-time, may be gathered from either a file or from communication between components.

With OASIS3-MCT_2.0, time transformations are supported more generally with use of the coupling restart file. The coupling restart file allows the partial time transformation to be saved at the end of a run for exact restart at the start of the next run.

#### 2.1.2.3 Data Exchange

Using the OASIS3-MCT_2.0 coupling library, the user has the ability to use differing coupling algorithms. In the components, the set and get routines can be called at each model timestep, with the appropriate date argument giving the actual time at the beginning of the timestep. This time argument is automatically analysed by the coupling library and, depending on the coupling period and lag value chosen by the user, for each coupling field, different coupling algorithms can be reproduced without modifying the component model codes themselves.

The lag value tells the coupler to modify the time at which that data is sent (set) by the amount of lag. The lag can be positive or negative, but should never be larger than the coupling period of any field due to problems with restartability and dead-locking. When a component model calls set, the value of the lag is automatically added to the value of the date argument and the set is actually performed when the sum date+lag is a coupling time; in the target component, this set will match a get for which the date argument is the same coupling time. The lag only shifts the time data is sent and cannot be used to shift the time data is received.

The order of coupling operations in the system is determined solely by the order of calls to send (set) and receive (get) data in the models in conjunction with the setting of the lag. Data that is received (get) is always blocking while data that is sent (set) is non-blocking with respect to the model making that call. It is possible to deadlock the system if the relative orders of puts and gets in different models are not compatible. With OASIS3-MCT provides the coupling layer with an ability to detect a deadlock before it happens and exit. It does this by tracking the order of get and set calls in models.

### 2.1.3 FLUME (Framework)

#### 2.1.3.1 Overview

The UK Meterological Office’s Flexible Unified Model Environment (FLUME) project created a coupling framework for the Unified Model (UM) system. The framework separates infrastructure and scientific code, where scientific code is modularised and infrastructure code generated during the project.

Components, such as an ocean model or a particular sea-ice model, and support systems, such as those providing for restart and diagnostic output, are composed to form a set of communicating processes which combine to create a weather or climate simulation. The coupled components communicate through the FLUME communications interface using the set-get method. The remainder of the framework description is modified from Ford and Riley (2003).

#### 2.1.3.2 Process

The sequencing and execution rates of components and couplers must be specified. Data from a number of components may have to be combined, with the appropriate coupler, in order to satisfy the requirements of the receiving component. In addition the definition of the coupling intervals between components is required. Couplers are called from the high-level framework driving code and therefore are similar in many aspects to the scientific components. The allocation of component implementation and coupler functionality to executable files, and their deployment on a set of available computing resources, must also be provided.

The layered framework approach for the coupling system is shown in Figure 3. The control layer invokes model components at a rate consistent with the coupling intervals defined in the composition environment. The control code implements the sequencing of the models both sequentially and concurrently depending on requirements.

In Figure 3, the intra-component communication, which is a consequence of models exploiting parallel implementation, is shown at the bottom of the layered architecture. This reflects the current implementation choice for Met. Office models, where such communication takes place from within a component. Inside the top level call, each component and coupler perform the exchange before and after calls to the component implementation routines.

#### 2.1.3.3 Data Exchange

There are a couple of options available for the inter-model communication mechanisms to implement coupling exchanges. Arbitrary placement of communications use asynchronous set and get functions, which may be placed anywhere within a model. The alternative method is to layer the placement of communications. Under this method the model should be implemented as a subroutine and communication should only occur through an argument list. In this scenario, communication is through a higher layer function placed in the control (driving) layer.

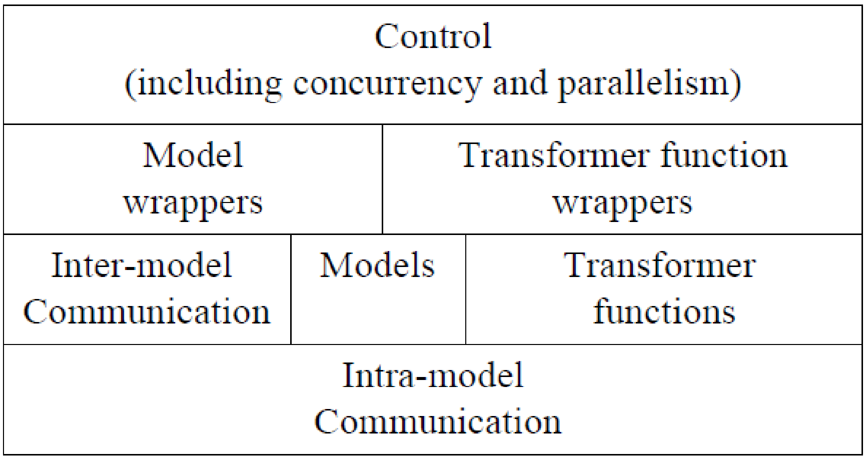



**Figure 3. Layered architecture for FLUME**

FLUME defines five types of input and output data:

- **Initial input control data** - this data is used to configure a model i.e. set its "knobs" and "switches".
- **Initial input data** - this data is used to provide initial conditions to prognostic fields (fields which are internally calculated by a model and whose state is maintained across timesteps) and to initialise any constant data.
- **Coupling input and output data** - this input data is produced externally to the model and changes over timesteps; this output data provides external data to other models which also changes over timesteps.
- **Diagnostic output data** - this data is used by scientists to determine the behaviour of the model.
- **Restart dump (checkpoint) output data** - this data is used to store the models state at intermediate steps in a simulation so that if an error occurs the simulation can be re-started from the latest checkpoint rather than from its initial conditions.

Data required to start a model must be specified in a models initialisation phase and by association the same data must also be specified in the dump phase. However, whether this data includes coupling data or not is a design choice. This document suggests (and makes the assumption that) coupling data is not specified as input or output in the init and dump phases respectively. Two reasons for this are 1: it reduces the number of "get" calls that need to be maintained 2: coupling get calls always return data (in the alternate case the first coupling get call after initialisation may need to "silently" return without modifying the data).

### 2.1.4 OpenPALM (Coupler)

#### 2.1.4.1 Overview

PALM is a coupler designed to combine dynamically different components into a high performance application. PALM was originally developed for operational oceanographic data assimilation in the framework of the French MERCATOR project. The PALM driver supports the dynamic launching of the coupled components, while its coupling library ensures the parallel data exchanges between the components. PALM also provides pre-defined algebra units. This PALM coupler description is modified from Valcke and Morel (2006).

In 2003 the final version of the PALM coupler, PALM_MP, was released. PALM_MP, which allows independent programs to work together, dealing with different data and different parts of the algorithm. The use of MPI2 for the passing of data makes this possible. In PALM_MP, components can be fully independent programs or, for optimization issues, subroutines of higher level entities called blocks. These recent developments allow the PALM coupler to operate on massively parallel architectures as well as integrate advanced interpolation methods. The latter are considered important as surface and volume interpolation models are needed to pass information between solvers at differing spatio-temporal scales

A PALM application can be described as a set of computational units arranged in a coupling algorithm. The different units are controlled by conditional and iterative constructs and belong to algorithmic sequences called computational branches. A branch is structured like a program in a high level programming language: it allows the definition of sequential algorithms. Inside a branch, the coupled component are invoked as if they were subroutines of the branch program.

#### 2.1.4.2 Process

PALM introduced the dynamic coupling approach where a coupled component can be launched and can release resources upon termination at any moment during the simulation. The originality of this coupler resides in the ability to describe complex coupling algorithms. Programs, parallel or not, can be executed in loops or under logical conditions. Computing resources such as the required memory and the number of concurrent processors, are handled by the PALM coupler. A component of the coupled system is only initialised when needed, reducing memory and processor use when inactive. With a static coupler, all the coupled programs would have to start simultaneously at the beginning of the simulation, occupying memory and CPU resources from the beginning to the end of the application. The concept of dynamic coupling came from the observation that different data assimilation algorithms can be obtained with different execution sequences of the same basic units and operators. In PALM, a dynamic coupling algorithm is composed of basic pieces of code, the components themselves and assembled components in different execution sequences (branches). Simulation maybe be started or stopped dynamically during the run.

The user defines and provides the elementary units, thereby fixing the scale of the coupling. Each component is a piece of code that must be instrumented by the user with a PALM wrapper. Each unit can consume and/or produce data, which are called objects, via the implementation of the get-set primitives. All the objects that a component can request or provide must be described in the component code by comment lines following a pre-defined syntax, which contain the object metadata. Modularity is ensured by the end-point communication principle: i.e., there is no reference to the origin of the input or to the destination of the output in the component code.

#### 2.1.4.3 Data Exchange

The execution of the coupled components is driven by a scheduler that allocates the computational resources according to the algorithm flow, the priorities and the limitations set by the user. At run time, the PALM driver ensures the execution and synchronisation of the different components, compiled by the user, following the sequence of actions defined in PrePALM.

The PrePALM package allows users to choose the elementary components to be coupled, which appear as individual boxes on the PrePALM GUI, and defines their execution sequences (branches). PrePALM analyses different component codes and clearly identifies the potential data input and output. To establish an exchange of information between components, the user links the output of one component to the input of another component; a pop-up appears on the link which allows the user to specify the different exchange parameters, such as the times of exchange. PrePALM also provides supervision tools such as a performance analyser and a runtime monitoring.

### 2.1.5 Summary

Atmospheric modelling frameworks for the coupling of Earth system components provide an attractive option for integrated modelling within the BGS. Many contain a land surface component as part of a coupled atmosphere-land-ocean coupling. However, these frameworks have little flexibility in terms of linking components within the land surface, as is often required in the coupled environmental modelling research we undertake. The coupling technology for the majority of these models is based on the MCT (model coupling toolkit), a set of open-source software tools for creating coupled models. MCT is fully parallel and can be used to couple message-passing parallel models to create a parallel coupled model. The passing of data is most commonly performed using the MPI (message passing interface) standard, where data is moved from the address space of one process to that of another process through cooperative operations. Due to the complexity of atmospheric modelling frameworks, the ability to restart model composition runs from a saved point is highly desirable. As integrated environmental modelling within the BGS advances and becomes increasingly complex, this ability to restart model compositions will also benefit future modelling. If BGS were to further develop a model coupling system, the Met Office FLUME project would be of interest, as the process of development and background research is freely available.

## 2.2 HYDROLOGICAL

While a large number of couples exist, clearly a few of them emerge as the most prominent. We will take a closer look at these couplers, also mentioning those that have a potential for linking different modelling frameworks. The report is primarily concerned with technologies that can be used to couple models from the same realm. However, web services that can be used to link hydrology and climate models or to link model and databases are also considered. The section on couplers is split into two parts: the first part describing couplers that can be deployed on lower level computing platforms such as desktops, and the second part describing these that are specifically designed for high performance computing (HPC).

### 2.2.1 Software suitable for desktop applications

#### 2.2.1.1 OpenMI

Open Modelling Interface (OpenMI) Standard was established by a consortium of 14 organisations from seven countries, in the course of the HarmonIT project co-funded through the European Commison’s Fifth Framework programme (Moore et al., 2010). It was originally developed to address the Water Framework Directive's call for integrated water resources at the catchment level (Moore and Tindall 2005), however, its application was later extended to other domains of environmental management (OATC 2010a). OpenMI is maintained and promoted by the OpenMI Association (OpenMI 2013), and is supported by the FluidEarth initiative of HR Wallingford (FluidEarth 2013), which provides tools for robust model integration, e.g.: FluidEarth2 Toolkit. OpenMI is equipped with GUI (OpenMI Configuration Editor), which facilitates creating and running compositions (Goodall et al., 2011).

Components in OpenMI are called 'Linkable Components' (Lu 2011) and their architectural design follow initialise/run/finalise cycle (Lawrence et al., Manuscript). They must be accompanied by metadata provided in the form of XML files (OATC 2010a) and encoded using either VB.Net or C# (Lu 2011). Models written in other languages (e.g.: Fortran, C, C++, F#, Matlab, etc.) can be integrated in OpenMI after implementing appropriate wrappers (OATC 2010a). A number of tools are available to assist users in developing their applications, including wrappers, which are provided in the form of code libraries (Software Development Kits or SDKs) (OATC 2010a). A set of interfaces need to be implemented to make a component OpenMI-compliant (OATC 2010a), with the central one being 'ILinkableComponent' (OATC 2010b).

The primary data structure is the 'ExchangeItem', which can be of two different types: 'InputExchangeItem' and 'OutputExchangeItem' (Saint and Murphy 2010). The ExchangeItems can be either 'Quantities' or 'Elementsets' (Lu 2011). A Quantity contains metadata of a variable, while an Elementset provides its spatial information (Lu 2011). To enable linking of data expressed in different units, each Quantity is provided with a conversion formula to standard SI system units (OATC 2010b). Elementsets contain references to the coordinate system used, which allows mapping between different systems (OATC 2010b).

The OpenMI was designed to exchange data on the time basis (i.e.: time stamp or time span), however, the exchange of data between temporal and non-temporal components (e.g: databases, data analysis tools) is also possible (OATC 2010a). The communication mechanism is based on request-reply mechanism ('pull driven' approach) (Lu 2011, OATC 2010a). A component only progresses if other component requests data from it via 'GetValues' method (OATC 2010a). Data request invokes 'Update' function on the called component, which triggers next time step computation. The produced output may have to be modified before returning to the calling component, to provide for differing grids (regridding) or time steps (interpolation, extrapolation) (OATC 2010b). Essentially, "components in OpenMI are connected in a chain and invoking the Update method on the last component in the chain triggers the entire stack of data exchange" (OATC 2010b).

OpenMI is a very popular standard for linking hydrologic models. The fact that a significant number of prominent water resources models (e.g.: MIKE SHE, MODFLOW, SWAT, ISIS, HEC-RAS) have been made OpenMI compliant (Graham et al., 2006, Gijsbers et al., 2010, ISIS 2013) proves that it is the industry standard of choice for integrated modelling.

#### 2.2.1.2 OMS

Object Modelling System (OMS) is an open-source software for linking components by means of annotations (David et al., 2013, OMS 2013). It was developed to support research within agricultural and natural resources management programmes administered by the US Department of Agriculture (USDA) (David et al., 2010). OMS originates from Modular Modelling System (MMS) - one of the first coupling frameworks, a hybrid between stand-alone model and a component-based modelling system (Lu 2011, David et al., 2013). OMS employs new advances in software framework design and is described as lightweight and non-invasive. It supports implicit multi-threading, implicit scaling to cluster and cloud, domain specific languages, and interoperability with other frameworks (David et al., 2013). Web services are enabled through specific annotations on the components (David et al., 2013). Simulations are described using a mini-language called Domain Specific Language (DSL) (David et al., 2010); the simulation file lists all model components, define connectivity, and provide parameter definitions (David et al., 2013). A number of pre-defined simulation types are available, including: Shuffled Complex Evolution global search algorithm (for model calibration), Fourier Amplitude Sensitivity Test, Dynamically Dimensioned Search parameter estimation, and Ensemble Streamflow Prediction (David et al., 2013). Models can be executed in a number of different platforms, e.g.: PC, cluster, or cloud (David et al., 2010)

OMS is based on Java, however, it is interoperable with C, C++ and Fortran. Therefore, models written in these languages do not need to be changed (David et al., 2010). The integration of components in OMS3 is achieved through the use of metadata annotations, encoded as declarations within XML files (Lu 2011), which "specify and describe points of interest amongst data fields and class methods of the model" (David et al., 2013). The initialise/run/finalise cycle is maintained merely by tagging methods with the corresponding annotations, e.g.: the compute method is tagged with '@Execute' (David et al., 2013). Data exchange is described using '@In' and '@Out' annotations (David et al., 2013). Components can be hierarchical and composed of progressively finer components (David et al., 2013). Annotation approach facilitates capturing modelling metadata (e.g.: units, ranges) and automatic generation of component's documentation (David et al., 2013).

In case of incompatible data types, units, resolution, or time step, the data can be transformed using a service provider interface (SPI) (David et al., 2013).

Execution is multithreaded by design; no explicit definition of execution order is needed as it is defined by the flow of data (David et al., 2013). Components are executed in parallel if all their input data is available (David et al., 2010).

There are several hydrologic applications of OMS3 up to date. The National Water and Climate Centre of the USDA Natural Resources Conservation Service (NRCS) used OMS3 to develop a modelling system for short term stream flow forecasting. The system is based on distributed physical process models and the Ensemble Steamflow Prediction (ESP) methodology. It provides capabilities for displaying selected ESP output traces, performing frequency analysis on the peaks/ volumes, or weighting output traces based on climate signals (e.g.: El Nino, La Nina, and Pacific Decadal Oscillation) (David et al., 2013). Another example of OMS application is Agro-Ecosystem-Watershed model (AgES-W) - a fully distributed model that simulates hydrology of a large watershed. It consist of above 80 Java-based components derived from a number of models, namely: J2K-S, SWAT, RZWQM2, and WEPP, which are integrated using OMS (David et al., 2013). OMS is also used in Northern and Central Africa for groundwater modelling studies using isotope tracing (David et al., 2013).

In recent years USDA-NRCS has initiated the Cloud Services Innovation Platform (CSIP). CSIP employs OMS3 and various databases to support environmental modelling within the cloud environment. CSIP development is still ongoing but it already runs watershed scale models (David et al., 2013).

#### 2.2.1.3 TIME

The Invisible Modelling Environment (TIME) is a metadata-based framework developed within the Catchment Modelling Toolkit project in the Cooperative Research Centre for Catchment Hydrology (CRCCH) (Rahman et al., 2003). CRCCH is currently a part of the eWater Cooperative Research Centre (CRC) - an organisation responsible for implementation of the Australian Government's National Hydrological Modelling Strategy (eWater CRC 2013).

TIME architecture is based on as a number of interacting layers, with each layer consisting of a number of components and a framework supporting the specific layer's function (Rahman et al., 2003). The central layer is the Kernel, which contains definitions of metadata tags, the parent classes for models and data, and mechanisms for performing IO operations (Rahman et al., 2003). The other layers include: the Model layer, which consists of all the modelling components; the Tools layer, which includes components for data and model processing and parameter optimisation; and the Visualisation and User Interface layer, which contains tools for data visualisation and user interaction (Rahman et al., 2003).

Components can be encoded in one of the several .NET languages, e.g.: Visual Basic, Fortran 95, C#, C++, Visual J#; modelling systems can be composed of components written in different languages (Rahman et al., 2003). All models are implemented as child classes inheriting from the Kernel's parent classes. Fields for inputs, outputs, parameters, and state variables are defined and documented using metadata tags (Rahman et al., 2003).

TIME supports a number of data types, e.g.: rasters, time series, points, lines, polygons, node link networks (e.g.: river systems), cross sections, arrayed data (Rahman et al., 2003). Most data types are represented by two classes: a class storing the data values, and a class storing its spatial/temporal context (Rahman et al., 2003). Along generic processing tools that act on all data types (e.g.: adding two objects together, statistics, and rule-based processing), a number of data type specific tools are available, e.g.: terrain analysis of rasters (Rahman et al., 2003). Unit conversions are provided by the Unit component (Rahman et al., 2003).

TIME was used to design a large number of integrated catchment modelling tools, mostly within the Catchment Modelling Toolkit project (Argent et al., 2009). A prominent example of TIME application is a decision support system (DSS), called E2 (Argent et al., 2009). E2 offers a tailored approach to conceptualisation of catchment dynamics, providing for flexible representation of different processes, through easily exchangeable model components (Argent et al., 2009). A catchment in E2 is represented by sub-catchments, each of which can contain one or more Functional Units (FU) – a portion of the sub-catchment displaying distinct characteristics and thus modelled using different models or parameterisation than the other parts of the sub-catchment (Argent et al., 2009). TIME features a sophisticated calibration tool, which provides a number of unique capabilities, e.g.: parameters varying in proportion between FUs can be scaled during the calibration to maintain the proportions (Argent et al., 2009). E2 software, a part of the Catchment Modelling Toolkit, has been used to construct over 20 water and environmental management DSSs (Argent et al., 2009). An advanced version of the catchment hydrology and water quality DDS, built upon E2, was released in 2008 under the name 'WaterCAST' (Argent et al., 2009).

#### 2.2.1.4 KEPLER

Kepler is an open-source desktop application for creating scientific workflows, which emerged from Ptolemy II (Kepler 2013). Ptolemy II is a framework allowing for a number of different modes of execution, which was developed at the University of California at Berkley and originally targeted at bioinformatics, computational chemistry, ecoinformatics, and geoinformatics (Kepler 2013a, Kepler 2013b). Ptolemy II and Kepler are characterised by separation of workflow components from the workflow orchestration, which enables direct reusability of components (Kepler 2013b). Workflows can be executed either from the GUI or from a command line (Kepler 2013). Each component is represented graphically in the GUI by an icon reflecting its function (Kepler 2013a). Kepler is featuring a library of above 530 ready components (Kepler 2013b), which facilitate a number of tasks, among others: remote data access, processing, analysis and visualization; transformations for syntactically incompatible components; GIS processing; execution of command line applications; statistical analysis using R or Matlab; web services invocation; cluster and grid computing, execution and monitoring (Goodall et al., 2011, Kepler 2013b, Kepler 2013). Kepler is maintained for Windows, OSX, and Linux operating systems (Kepler 2013).

Kepler workflow is composed of components, called actors, each performing a different function. A director is a special type of an actor that controls (directs) the execution of a workflow. Workflows can have a number of sub-workflows (also called composite actors), each comprised of a collection of actors performing complex embedded task and each controlled by its own director (Kepler 2013a). Kepler is developed in Java, however, components written in other language can be adopted by using wrappers (Kepler 2013b).

Workflows pipe output of one component to an input of another component. Library actors facilitate data transformations for syntactically incompatible components. Data is exchanged via ports; there are three types of ports: input, output, and input/output. Ports are configured to specify the type of data that they accept and to indicate if they are 'singular' or 'multiple'. A single port can only be connected to one actor, whereas a multiple port can be connected to many actors. In the latter case, data can be sent to a number of different places in the workflow, e.g.; a different actor for further processing and a display actor to visualise the data at a specific reference point (Kepler 2013a).

Workflow execution can be synchronous or parallel, depending on the type of director used. A small set of directors come pre-packaged with Kepler, including: Synchronous DataFlow (SDF), Process Networks (PN), Dynamic Dataflow (DDF), Continuous Time (CT), and Discrete Events (DE). (Kepler 2013a, Kepler 2013b). SDF director is used to oversee simple, sequential workflows, in which data consumption and production rate is constant and declared (Kepler Project 2013b). PN director is used for workflows that are driven by data availability. Actor is executed once it collects all the required inputs. Being loosely coupled, this kind of workflows are good candidates for parallel and distributed computing. DE director oversees workflow where events occur at discrete times and is well suited for modelling time-oriented systems. CT director is designed to oversee workflows that predict how systems evolve as a function of time. Rates of change in such systems are described by differential equations and each workflow execution is simply one time step of a numerical integration. Similarly to SDF director, DDF director executes a workflow in a single thread. However, data production and consumption rates can change as workflow executes. It is a good choice for workflows that use Boolean switches, if-then-else statements, branching, or that require data-dependent iterations (Kepler 2013b)

There do not seem any hydrological applications of Kepler in the open literature. However, Kepler was suggested to perform web services orchestration of water resources models (Goodall et al., 2011), and to replace OpenMI in a two-way coupled system, developed by Goodall et al., (2013), which links a hydrological model with a climate model.

#### 2.2.1.5 TAVERNA

Taverna is an open-source software, composed of a set of tools written in Java, which facilitates discovery, design, and execution of scientific workflows (Taverna 2013). It automates multi-step and repetitive tasks involving invocation of several applications, largely web services-based (Deelman et al., 2009), by defining the flow of data and performing format conversions (Taverna 2013). Taverna has been developed within myGrid project and funded through OMII-UK - an organisation supporting development of open source software for the UK research community (Taverna 2013). The rationale behind Taverna development was providing scientists, that only have basic understanding of programming, with a straightforward environment for assembling and executing workflows (Sroka et al., 2010). Scientific collaboration and reuse of workflows is encouraged through partnership with myExperiment portal, a social networking and workflow sharing environment for scientists, where the existing workflows can be discovered and downloaded from (Taverna 2013, De Roure and Goble 2009).

A range of different types of services are supported within Taverna, e.g.: WSDL, RESTful, BioMart, BioMoby and SoapLab web services; R scripts on a R server (Rshell scripts); local Java services (Beanshell scripts); data import from Excel or csv spreadsheets (Taverna 2013). Users can access over 3500 ready applications and analysis tools; BioCatalogue, accessible through Taverna website, provides details of the services that are currently available (Taverna 2013). External tools, scripts, or Java libraries can be easily incorporated as plug-ins or via ssh calls (Taverna 2013).

Tools for workflow validation (debugging) during the composition and detection of service's interface changes and off-line times are included in the suite (Taverna 2013). Execution can be monitored and paused, and workflows can be debugged at run time (Taverna 2013). Workflows are run from within the desktop application, called Workbench, which provides a graphical user interface for the selection of the services (Taverna 2013, De Roure and Goble 2009); Command Line Tool for the execution of workflows from a terminal is also provided (Taverna 2013). Workflow execution is data-driven and parallel; the number of the concurrent threads is configurable (Sroka et al., 2010, Taverna 2013). A trace of a workflow is recorded, providing information on the executed services, inputs, and outputs (Taverna 2013). Taverna supports remote deployment of workflows, e.g.: on a grid or on a cloud, and editing and running workflows on the Web (Taverna 2013).

Although Taverna was originally designed for bioinformatics, it is domain independent and can be applied in a number of different disciplines (Taverna 2013). Currently, more than 350 organisations around the world employ Taverna and its use has spanned a large number of different fields, e.g.: bioinformatics, astronomy, chemistry, engineering, geoinformatics, biodiversity, social sciences, data mining, education, arts (Taverna 2013). An example of a hydrology-related application of Taverna is the development of the Environmental Virtual Observatory (EVO) (Taverna 2013), environmental monitoring and decision making system based on web services (EVO 2013).

#### 2.2.1.6 FrAMES

Framework for Aquatic Modelling of the Earth System (FrAMES), developed at the University of New Hampshire, is software used for simulating biogeochemical processes as water is routed through an aquatic system to a coastal zone. It allows assessing contaminant removal and attenuation from its source to the river's outlet, and permits studying process kinetics, role of different stream orders, impact of water withdrawals, spatial distribution of contaminant inputs, and factors controlling contaminant removal (Wollheim 2006). The modelling system is composed of gridded terrestrial and aquatic components, and can be applied at both local and global scales using gridded river networks of varying resolutions depending on the application (Wollheim 2006). FrAMES runs on Linux/Unix operating systems and requires very little knowledge of coding for its implementation (Wollheim 2006).

Building on FrAMES, Next Generation Framework for Aquatic Modelling of the Earth System (NextFrAMES) is being developed. It uses an eXtensible Markup Language (XML) for describing a model structure (Fekete et al., 2009, Lu 2011) and a declarative language to integrate components (Lu 2011). It is characterised by a high level of abstraction; most of the services are hidden behind the platform to offer more straightforward model development environment (Fekete et al., 2009).

#### 2.2.1.7 FRAMES

Framework for Risk Analysis of Multi-Media Environmental Systems (FRAMES) is a piece of software developed and used by the US Environmental Protection Agency. It is composed of 17 modules (called 3MRA) collectively simulating release, fate and transport, and exposure and risk to human and environment associated with contaminants originating from landfills, waste piles etc (Jagers 2010). As the results are based on ten thousand simulations, to shorten the total run time, the modules use highly simplified representation of processes (Jagers 2010). The communication method is one-way and file-based, which is planned to be replaced by two-way in-memory communication based on OpenMI (Jagers 2010).

### 2.2.2 Software developed for HP computing

#### 2.2.2.1 CSDMS

CSDMS is an international initiative, funded by US National Science Foundation (NSF), which promotes sharing, reusing, and integrating Earth-surface models (Peckham et al., 2013). CSDMS implements CCA Common Component Architecture (CCA) standard for model coupling, which is adopted by many US federal agencies. CCA development started in 1998 to address the demand for technology standards in high-performance scientific computing (Peckham et al., 2013). CCA is distinguished by its capacity to support language interoperability, parallel computing and multiple operating systems (Peckham et al., 2013). Three fundamental tools underpin CSDMS, namely: Babel, Ccaffeine, and Bocca (Peckham et al., 2013). CSDMS is equipped with GUI, called Ccafe-GUI, in which components are represented as boxes that can be moved from a palette into a workspace. Connections between components are made automatically by matching "uses ports" to "provides ports" (Peckham and Goodall 2013). Results of simulations can be visualised and analysed during and after the model run using a powerful visualisation tool (VisIt) (Peckham and Hutton 2009), which features, among others, the ability to make movies from time-varying databases (Peckham et al., 2013). A light-weight desktop application is provided, called CSDMS Modelling Tool (CMT), which runs on a PC but communicates with the CSDMS supercomputer to perform simulations (Peckham and Goodall 2013, CSDMS 2013).

CCA components' must be split into initialise, update, and finalise sections. CSDMS provide a tool called Bocca that helps creating, editing and managing CCA-compliant components (Peckham et al., 2013). Models can be written in a number of different languages, i.e.: C, C++, Fortran (77, 90, 95, and 2003), Java, and Python. The communication between such disparate pieces of code is achieved thanks to implementation of the language interoperability tool called Babel, which automatically generates the "glue code", enabling models to exchange data (Peckham et al., 2013). For Babel to do its work, it only needs the descriptions of the component's interface, written either in XML (eXtensible Markup Language) or SIDL (Scientific Interface Definition Language), including information on the data types and the return values of the methods (Peckham et al., 2013).

Data transformations between components are enabled through the use of the utility components, which provide services such as: spatial regridding, time interpolation, unit conversion, variable name matching, or writing outputs to a standard or NetCDF file formats (Peckham et al., 2013).

To allow communication between components they have to be wrapped with two interfaces. The first level interface called Basic Model Interface (BMI), must be implemented by a model developer and provide a set of basic functions, namely: initialise, update, and finalise. These functions allow communication with the underlying wrapped model and enable model to "fit into a second-level wrapper" (Peckham and Goodall 2013). A model that has the BMI interface is converted to a CSDMS component by providing it with the second level interface, called the Common Model Interface (CMI), using the CSDMS automated tools. CMI allows CSDMS components to communicate and exchange data (Peckham et al., 2013). Runtime environment is provided through the third fundamental CSDMS tool called Ccaffeine, which enables "component instantiation and destruction, connecting and disconnecting ports, handling of input parameters, and control of Message Passage Interface (MPI) communicators" (Peckham et al., 2013).

CSDMS maintains a large database of contributed models from a variety of Earth's surface dynamics disciplines, e.g.: hydrology, sediment transport, landscape evolution, geodynamics, glaciology, coastal and marine, and stratigraphy. The current number of hydrological model in the repository exceeds 50 (CSDMS 2013).

CSDMS have been used in a number of hydrologic studies. Ashton et al., (2013) coupled hydrological transport model HydroTrend with Coastline Evolution Model (CEM) to study how fluctuations in sediment input due to climate change may affect delta morphology and evolution (Ashton et al., 2013). An ongoing PhD study employs CSDMS to improve representation of the physiographic distribution of snow water equivalent and timing and volume of simulated stream flows (CSDMS 2013). Examples of other applications include: studying the consequences of past and future climate changes on water resources, water storage, and the expansion of the desert in the eastern watersheds of Jordan; or investigating the effects of terrain and vegetation structure on soil moisture, hydrological flow, and snowmelt (CSDMS 2013).

2.2.2.2 BFG

Bespoke Framework Generator (BFG) is software developed at the Centre for Novel Computing (CNC) in the School of Computer Science at the University of Manchester. The rationale for its development was creation of a framework that imposes minimal number of requirements on component's architecture and thus allows for straightforward and flexible model integration (Henderson 2006). BFG only needs metadata, in the form of XML files, in order to generate the required wrapper code, which then can be used with a coupling system of the user's choice (Henderson 2006, Warren et al., 2008). A component must comply with a small set of rules, i.e.: it must be a subroutine or a function, and use 'put' to provide data and 'get' to receive data (Warren et al., 2008). XML files must be entered manually by a user; it is planned that in the future they will be generated automatically from a GUI (Henderson 2006).

The process of model integration is characterised by "separation of concerns", which can be summarised by terms: Define, Compose, and Deploy (DCD) (Warren et al., 2008). These terms correspond to three XML files containing interface, composition, and deployment information (Henderson 2006). The interface metadata describes which fields component requires and which it provides, and includes information about the module's time step (Warren et al., 2008). Composition metadata describes how fields are connected between different models. Fields can be connected using either 'inplace I/O' or 'argpass I/O'. In the case of the former, the output fields are connected with the corresponding input fields using “point-to-point notation”. In the case of the later, the connections between fields are made by grouping together the subroutines that use a particular field (Henderson 2006). Deployment metadata defines scheduling information, that is: a number of executables and MPI processes, a number of threads, and a sequence in which model functions are to be called (Henderson 2006). Using an XSLT processor metadata is converted to a source code capable of controlling and coupling the models (Henderson 2006).

BFG supports complex control representation, e.g.: inner loops or convergence based loops (Henderson 2006). On the other hand, it allows for control to be handled within the source code of the models. Such models are referred to as having “minimal compliance” and they must only provide one entry point subroutine that BFG can call to start the model (Henderson 2006).

BFG can generate wrapper code for: "models with Fortran entry points running in sequence on a single machine in a single executable communicating through shared buffers; models with Fortran entry points running concurrently, generated as a single executable communicating through MPI; models with Fortran entry points running concurrently, with a configurable number of executables, communicating through MPI; models with Fortran entry points running concurrently using a TDT sockets implementation; models with Fortran entry points running concurrently using a TDT SSH implementation; models with Fortran 90 entry points running concurrently using OASIS3" (BFG 2013).

At the moment BFG has no built-in capability for carrying out unit, spatial, and temporal transformations. When BFG is used with OASIS, these transformations are carried out by OASIS itself (Henderson 2006).

A prominent example of BFG-facilitated model integration is the Flexible Unified Model Environment (FLUME) - UK Met Office Earth System Modelling system (Henderson 2006). Another example is GENIE Earth System Modelling Framework - IGCM atmosphere and GOLDSTEIN ocean models coupled using OASIS4 and BFG (Henderson 2006). It is also worth mentioning Community Integrated Assessment System (CIAS) – a system used for studying relationships between the economy and the climate change and composed of models distributed across different institutions (Warren et al., 2008). Owing to BFG, models in CIAS can be easily exchanged, allowing for different policy variants and the modelling uncertainty to be readily assessed (Warren et al., 2008).

#### 2.2.2.3 ESMF

The Earth System Modelling Framework (ESMF) is a software for building complex Earth system modelling applications and is typically used to couple models of large physical domains (ESMF 2013). ESMF originates in the Common Modelling Infrastructure Group (CMIWG), which comprised major US weather and climate modelling organisations. It was developed in response to the NASA Earth Science Technology Office (ESTO) Cooperative Agreement Notice, entitled "Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life and Microgravity Sciences", which called for it creation (ESMF 2013). ESMF implements methods, which allow separate components to operate as a single executable, multiple executables or web services (Valcke et al., 2012). It supports parallel computing on Unix, Linux, and Windows HPC platforms (Lu 2011, Jagers 2010).

ESMF is based on two types of components: 'Gridded Components' (ESMF_GridComp) and 'Coupler Components' (ESMF_CplComp) (ESMF 2013). Gridded Components represent the physical domain being modelled while Coupler Components enable data transformation and transfer (ESMF 2013). Coupler Component's operations include: time advancement, data redistribution, spectral and grid transformations, time averaging, and unit conversions (ESMF 2013). Coupler Components need to be written in Fortran on case by case basis using ESMF classes (ESMF 2013). Gridded Components need to be split into one or more initialise, run, and finalise sections callable as subroutines (Goodall et al., 2013, ESMF 2013). ESMF allow for nested components, with "progressively more specialised processes or refined grids" (ESMF 2013).

The user is required to write a wrapper code that will connect component's native data structures to ESMF data structures (ESMF 2013). There are two ways to do it: either using the 'ESMF_Array' class to represent the data structures in an index-space, or using the 'ESMF_Field' class to represent them it in a physical space (ESMF 2013). In the latter case interpolation weights can be calculated using coordinate information stored in the 'ESMF_Grid' class; bilinear and higher order interpolation calculations in up to three dimensions are supported (ESMF 2013). User is also required to write 'SetServices' routine, which associates the ESMF initialise/run/finalise methods with their corresponding user code methods (ESMF 2013).

Data is passed using container classes called 'States' (Goodall et al., 2013); each Gridded component has an import State, containing its inputs, and an export State, containing its outputs (ESMF 2013). States can hold different data classes, including Arrays, ArrayBundles, Fields, or FieldBundles (ESMF 2013). "Arrays store multidimensional data associated with an index space. Fields include data Arrays along with an associated physical grid and a decomposition that specifies how data points in the physical grid are distributed across computing resources. ArrayBundles and FieldBundles are groupings of Arrays and Fields, respectively" (Goodall et al., 2013).

Although, ESMF is primarily aimed at high performance climate/weather/atmospheric computations, its developers seek cooperation with hydrological modellers and have been looking into ways to achieve cross-domain integration between ESMF and water resources modelling systems (Deluca et al., 2008).

#### 2.2.2.4 OASIS

Ocean Atmosphere Sea Ice Soil coupler (OASIS) is a software used for coupling models representing different components of the Earth system (OASIS 2013). It was developed at The

European Centre for Research and Advanced Training in Scientific Computation (CERFACS) in the framework of the EU FP5 Programme for Integrated Earth System Modelling (PRISM) (Valcke et al., 2006). The main purpose of PRISM was development of the infrastructure for European climate research and it involved 17 European climate research centres and a number of computer software companies (DKRZ 2013). OASIS is characterised by low intrusiveness; "components remain almost unchanged with respect to their standalone mode" (Valcke et al., 2012). In a coupled system components act as separate executables, while the main function of the coupler is to interpolate and exchange data between the components (Caubel et al., 2005). OASIS is based on Fortran and C (Valcke and Morel 2006). Currently three versions of the coupler exist: OASIS3, OASIS4, and OASIS3-MCT (Caubel et al., 2005, OASIS 2013). Since OASIS3 only supports 2D coupling fields, a fully parallel OASIS4 was developed, which supports higher number of coupling fields and targets high resolution climate simulations (Caubel et al., 2005). OASIS3-MCT, is the OASIS coupler interfaced with Model Coupling Toolkit (MCT). This version provides capabilities for parallel execution of data transformations and exchanges (OASIS 2013).

To implement data exchange at run time, the components are linked to the OASIS coupling interface library (PSMILe), which enables sending data requesting and data passing calls. The characteristics of the exchanges are defined outside of the model code, in an external user-written configuration file (Valcke et al., 2012).

Due to its flexibility and low intrusiveness, OASIS have been very popular and is currently used by about 35 different climate modelling groups in Europe, Australia, Asia and North America (Valcke et al., 2012). An example of hydrology-related application of OASIS is the study of impacts of climate change on the water cycle in the Mediterranean using the coupled system composed of REgional atmosphere MOdel (REMO), the Max-Planck-Institute for Meteorology Ocean Model (MPI-OM) and the Hydrological Discharge Model (HD model) (Arellano 2011).

## 2.3 WEB SERVICES

Applications operating as web services are based on components that are independent, distributed, loosely-coupled and exchange data over a computer network. In the hydrological domain web services are used in a number of ways, e.g.: to integrate hydrologic data from heterogeneous sources; to link modelling frameworks with databases; to connect models, databases, and analysis tools into water resources decision support systems; or to join modelling systems from different domains (e.g.: hydrology and climate).

There are a number of examples of successful use of service-oriented technology for environmental data integration. One such example is Hydrologic Information System (HIS), created by the Consortium of Universities for the Advancement of Hydrological Science Inc. (CUAHSI) – an organisation of more than 100 US universities aimed at developing infrastructure and services for the advancement of the hydrologic sciences (Peckham and Goodall 2013). HIS is composed of hydrologic databases and servers connected through web services (Peckham and Goodall 2013). It employs WaterOneFlow web service interface and Water Markup Language (WaterML) for data transmission to enable integration of hydrologic data from heterogeneous data sources into one “virtual database” (Goodall et al., 2011).

Research efforts focus also on ways to integrate data and modelling systems. HydroDesktop is open source GIS-enabled software developed by CUASHIU HIS, which allows accessing HIS services from a personal computer. It not only provides capabilities for data querying, downloading, visualisation, editing, graphing, analysis, and exporting to different formats but also supports integrated model development and use of the retrieved data in simulations (HydroDesktop 2013). HydroModeler is a HydroDesktop plug-in, based on OpenMI Configuration Editor, which provides functionality for building and executing model compositions from within HydroDesktop (HydroDesktop 2013). Another example of data and modelling systems integration stems from the partnership between CSDMS and HIS. As a result of this cooperation a novel system was developed, which allows accessing HIS data through web services calls from within the CSDMS modelling environment (Peckham and Goodall 2013). This functionality was achieved by incorporating an additional component, called DataHIS, within a CSDMS model composition. It is planned that CSDMS web services are further developed, provided that other environmental databases employ standardised interfaces for data retrieval and integration. It is envisioned that in the future CSDMS components could become web services themselves, potentially available to client applications such as HydroDesktop and HydroModeler (Peckham and Goodall 2013).

Building water resources modelling systems using web services is certainly more challenging than using them for data integration. However, it offers an advantage of keeping models independent thus allowing for continuous maintenance and development (Goodall et al., 2011). Goodall et al., (2011) proposed interface design for exposing models as web services and presented a prototype of service-oriented water resources decision support system. The interface was designed combining ideas from two standards: OGS Web Processing Service, and the Open Modelling Interface (Goodall et al., 2011). OpenMI ExchangeItem object was used as a starting point for developing data exchange standard. However, more work is needed to standardise the vocabulary of variables, unit names and geographical referencing systems, possibly adopting NetCDF Climate and Forecast Metadata Conventions (Goodall et al., 2011). For web services integration, OpenMI Configuration Editor was selected, as it already includes conventions specific for water resources modelling. However, since OpenMI does not support web services, a web service component was created that enables incorporation of this functionality within OpenMI (Goodall et al., 2011). To demonstrate the successful implementation of the system, a model simulating rainfall/runoff was assembled (Goodall et al., 2011).

Another technology that could potentially be harnessed for building decision support systems is cloud computing. Environmental Virtual Observatory (EVO) pilot project, sponsored by the UK's Natural Environment Research Council (NERC), employs cloud computing to integrate datasets, models and tools for cost-effective, efficient and transparent environmental monitoring and decision making (EVO 2013). EVO works with other international partners (e.g.: CUAHSI, NeON) to develop consistent standards for exchanging data and models (EVO 2013). The project activities include developing cyber infrastructure, cloud-enabled environmental models, and a number of exemplar web-based services concerning soil and water management at both local and national scales (EVO 2013). Exemplars developed within the course of the two year pilot project focus on a range of environmental problems, which directly affect the well-being of people in the UK, e.g.: studying national-scale nutrient fate using linked hydrogeological and biochemical models, developing a system to assess the effects of different land management practices on reducing diffuse pollution from agriculture, advancing modelling capabilities for drought and flood predictions to address and mitigate the effects of climate change, or establishing technologies for studying biodiversity and ecosystem service sustainability (EVO 2013). EVO aims to provide different groups of users, from scientists to local stakeholders, with free and easy access to expert knowledge by combining assets from various sources with novel tools for data analysis and visualisation (Gurney et al., 2011). The system is designed to promote feedback, ownership, community involvement, and better communication between technical ad non-technical users (EVO 2013). An example of a community tool established within EVO is The Local Landscape Visualisation Tool, developed by engaging stakeholders in three catchments in the UK: the Afon Dyfi, the River Tarland, and the River Eden (Wilkinson et al., 2013). The tool is accessed via a web portal and communicates flood risk in the local impacted communities. It is based on a number of services, i.e.: catchment datasets, hydrological models, and visualisation tools. Users can access real time data concerning river levels, rainfall, weather, and water quality, which is additionally supported by webcam images, or can use cloud-based models to explore how different land management strategies might affect the risk of flooding (Wilkinson et al., 2013).

Last but not least, web services can be used to link different modelling frameworks. Hydrologic studies traditionally did not consider bi-directional interactions between atmosphere and water bodies. However, as the scale of the models increase, the assumption about the lack of feedback between the land surface and the atmosphere may no longer hold and bi-directional coupling becomes important (Goodall et al., 2013). Up to date coupling of hydrological and climate models has been hindered by discrepancies between both technologies, namely climate models run on high performance computers while hydrologic models run on personal computers (Goodall et al., 2013, Saint and Murphy 2010). Additionally, there is a lack of established techniques for transferring data between differing spatial scales of climate and hydrologic models (Goodall et al., 2013). Hydrological Modelling for Assessing Climate Change Impacts at different Scales project (HYACINTS) coupled climate model HIRHAM and physically distributed hydrological model MIKE SHE for the whole of Denmark by migrating both models into the OpenMI standard (HYACINTS 2013). Method based on statistical downscaling and bias-correction was developed to enable data transfer across different grids (HYACINTS 2013). While the project achieved integration of models from different domains, this required migrating them to the same standard. Goodall et al., (2013) proposed a novel approach to loosely couple climate and hydrologic models using web services, which enabled integration of different modelling frameworks. The researchers did not address the problem of data scalability between climate and hydrologic models but merely aimed to develop technically feasible strategy for coupling such models. In the proposed approach web services are used to pass data between a hydrologic model running on desktop computer and a climate/weather model running in HPC environment (Goodall et al., 2013). The prototype developed in the study was a two-way coupled system composed of the Community Atmosphere Model (CAM) and the Soil and Water Assessment Tool (SWAT) (Goodall et al., 2013). CAM implemented with ESMF was made available as a web service. SWAT was provided as an OpenMI compliant model and CAM model was wrapped with an OpenMI interface (Goodall et al., 2013). The execution was controlled and implemented by OpenMI’s Configuration Editor (Saint and Murphy 2010). This study proved that coupling of two disparate modelling systems is feasible while still maintaining the models' original structure and purpose (Goodall et al., 2013). The study provided a technical solution for coupling models running on different computing platforms, e.g.: PC and HPC, different HPCs, or cloud (Goodall et al., 2013). Bridging the gap between OpenMI and ESMF was possible due to features that both standards provide, namely: ESMF supporting web services and OpenMI supporting a wrapper for accessing external services (Goodall et al., 2013). Both frameworks are widely used within their respective communities and their integration is an important milestone in modelling coupled hydrology-climate systems (Saint and Murphy 2010).

# 3 Data standards for one way, static transfer of data

## 3.1 GENERAL

The way data is organised, formatted and transferred within groundwater and other numerical process modelling teams within BGS has historically been controlled by the individual carrying out the research and influenced by the technologies used. This often results in a mass of loosely controlled text files stored on local and networked computer drives. These files contain source data, metadata on the methodology used to create the model, metadata on the model outputs and the resultant outputs. By learning lessons from the BGS corporate software team, the process modelling teams could improve model and data management, reduce duplication of effort and enable greater data reuse.

The BGS has invested a vast amount of money and time into the professionalisation of information management, specialising in the storage of geological data from a wide range of sources and standardising digital formats to maximise opportunities for data reuse. Through these efforts the BGS have built up a robust digital infrastructure and staff expertise in the fields of relational databases, applications design and web based communications. To date much of this knowledge has not been applied to the field of process modelling in the BGS, but there are ongoing efforts to rectify this, for example adapting international spatial metadata standards for use in process models or use through the introduction of the source code repository and versioning system, Subversion.

Whilst the BGS aim to improve how static data relating to process models is managed there remains a wider issue of how such data is incorporated into the model coupling technologies. The most popular coupling technology in the BGS to date has been the FluidEarth software development kit (SDK) for the OpenMI 1.4 standard, which does not support the linking of process model components to static data sources in a model workflow (referred to as a composition). The OpenMI 2.0 standard does include support for the linking of static data sources but this functionality is yet to be tested by BGS staff.

There are three data source types which are most likely to be used in a linked model composition, namely text files, relational databases and web services. Each of these data source types can be used in an indiscriminate or standardised way; the following lists provide an overview of the key standards, technologies and organisations that relate to the storage and transfer of gridded data, the most common spatial representations in mathematical process models.

There are a number of organisations that publish standards for spatial data structure, these include:

- **ISO**, traditionally focussed on the logical data models required to describe phenomena, these tend to be published in the form of UML models
- **OGC**, the Open Geospatial Consortium aim to gain consensus on standards by building upon existing real world implementations, therefore, it could be argued, more useful in applied use cases than ISO.
- **W3C**, the world wide web consortium, is the main standards organisation for the WWW, set up by Tim Berners Lee. It aims to ensure compatibility and agreement between the industry leaders behind the web. W3C standards that may relate to IEM technologies include HTML, SOAP, SPARQL, XML and WSDL

Other more proprietary organisations such as ESRI, Microsoft and Oracle define file formats and interfaces which often relate to international standards or become standards in their own right, simply because these technologies are so widely used.

Specific standards that relate to the datasets which are likely to be involved in linked models include:

- **CSW**, Catalog Service for the Web is one part of the OGC Catalog Service specification that they describe as follows *"Catalogue services support the ability to publish and search collections of descriptive information (metadata) for data, services, and related information objects. Metadata in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software. Catalogue services are required to support the discovery and binding to registered information resources within an information community."*
- **GML**, Geographic Markup Language is an OGC XML standard for geographic systems, it describes features, geometries, coordinate reference systems and more. One of the primary purposes for GML is to help connect various geographic databases
- **WCS**, Web Coverage Service: provides access, sub setting, and processing on a 'coverage' (a spatio-temporal feature conveying different values at different locations)
- **WCPS**, Web Coverage Processing Service is maintained by the OGC and provides a languages for querying raster data over the web.
- **WFS**, Web Feature Service from the OGC, provides an interface which allows clients to query and access geographical features across the web.
- **WMS**, Web Mapping Service is a specification published by the OGC and defines a protocol for serving of georefenced map images over the internet. As the images themselves tend not to be analysed in quite the same way as the data received via a WFS call this service may be less relevant to challenge of linking models.

Various technologies and libraries have been created to support the management of spatial data, noteworthy examples include:

- **GDAL**, Geospatial Data Abstraction Library is, according to gdal.org *"a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It also comes with a variety of useful commandline utilities for data translation and processing."*
- **Oracle Spatial**, although a less generic solution than most of those mentioned in this section, Oracle Spatial is particularly relevant to the BGS as the corporate database is hosted on an Oracle 11g server. The BGS corporate database contains a wealth of spatial data that could theoretically be consumed by process models, not least the Geological Object Store of modelled objects.
    - Oracle Spatial has an implementation of CSW
    - Through ArcSDE it is possible to access and edit Oracle Spatial data in a GIS environment
    - GDAL is able to read and write raster data in Oracle Spatial GeoRaster format

Direct database connections provide powerful ways to store and access spatio-temporal data and metadata. Connection technologies include:

- **ADO**, a Microsoft middleware layer that sits between a programming language and OLE DB

- **ODBC**, is the Open Database Connectivity standard API for accessing data from a wide range of database platforms. Drivers exist for all major database management systems and many other sources such as Microsoft Excel and CSV files.
- **OLE DB**, another Microsoft solution is an API that allows access to data in a variety of formats, including non-relational database data sources. It is now a legacy technology that has been superseded by ODBC.

## 3.2 ATMOSPHERIC

Atmospheric datasets tend to fall into three generic categories, Gridded Binary (GRIB), Network Common Data Form (netCDF) or the Hierarchical Data Format (HDF) system. All are intended for use with modern atmospheric datasets, which encompass information about the atmosphere, sea, and ocean. The same systems are used for observed and simulated data, as observational data is often used to initialise atmospheric models, particularly those adopted for short term weather prediction. Atmospheric datasets avoid the use of gridded ascii files, as the volume of data produced renders these file types unsuitable. The Climate and Forecasting (CF) standard for atmospheric datasets was conceived at the turn of the century and is increasingly gaining acceptance as the *de facto* convention. CF aims to distinguish quantities (descriptive, units, prior processing, etc) and to spatio-temporally locate data as a function of other independent variables, such as a coordinate system (Gregory, 2003). Each method for storing data for transfer has its own advantages and therefore if a method is selected it should be the most adequate for the data concerned

### 3.2.1 GRIB

The Gridded Binary (GRIB) format is commonly used to store meteorological datasets, both forecast and historical. The GRIB standard is described in detail in the World Meteorological Organisation (WMO) code manual (WMO, 1995). There have been three versions of the GRIB standard, however the first (GRIB 0) was only used on a limited number of projects. The second version has been used operationally for a number of years. Currently the third generation GRIB format (GRIB2) is used by some institutions at the operational level. Use of the third generation standard is expanding.

The GRIB file format is a set of self containing records, which when broken down retain their usability. They are composed of two main parts, the header and the data, the latter of which is in binary format.

### 3.2.2 HDF

Hierarchical Data Format (HDF, HDF4, or HDF5) is the name of a set of file formats and libraries designed to store and organise large amounts of numerical data. The HDF format, libraries and associated tools are available under a liberal, Berkeley Software Distribution (BSD)-like license for general use. HDF is supported by many commercial and non-commercial software platforms, including Java, MATLAB/Scilab, Octave, IDL, Python, and R. The freely available HDF distribution consists of the library, command-line utilities, test suite source, Java interface, and the Java-based HDF Viewer (HDFView).

HDF is self-describing, allowing an application to interpret the structure and contents of a file with no outside information. One HDF file can hold a mix of related objects which can be accessed as a group or as individual objects. Users can create their own grouping structures called *vgroups*. There currently exist two major versions of HDF; HDF4 and HDF5, which differ significantly in design and API.

HDF4 is the older version of the format, although still actively supported by The HDF Group. It supports a proliferation of different data models, including multidimensional arrays, raster images, and tables. Each defines a specific aggregate data type and provides an API for reading, writing, and organizing the data and metadata. New data models can be added by the HDF developers or users. The HDF4 format has many limitations. It lacks a clear object model, which makes continued support and improvement difficult. Supporting many different interface styles (images, tables, arrays) leads to a complex API. Support for metadata depends on which interface is in use; SD (Scientific Dataset) objects support arbitrary named attributes, while other types only support predefined metadata. Perhaps most importantly, the use of 32-bit signed integers for addressing limits HDF4 files to a maximum of 2 GB, which is unacceptable in many modern scientific applications.

The HDF5 format is designed to address some of the limitations of the HDF4 library, and to address current and anticipated requirements of modern systems and applications. HDF5 works well for time series data such as stock price series, network monitoring data, and 3D meteorological data. The bulk of the data goes into straightforward arrays (the table objects) that can be accessed much more quickly than the rows of a SQL database, but access is available for non-array data. HDF5 simplifies the file structure to include only two major types of object:

- Datasets, which are multidimensional arrays of a homogenous type
- Groups, which are container structures which can hold datasets and other groups

This results in a hierarchical filesystem-like data format. Metadata is stored in the form of user-defined, named attributes attached to groups and datasets. More complex storage APIs representing images and tables can then be built up using datasets, groups and attributes.

The latest version of NetCDF, version 4, is based on HDF5.

### 3.2.3 NetCDF

Net CDF is a set of interfaces for array-oriented data access and a distributed collection of data access libraries for C, Fortran, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data.

The NetCDF format is self-describing, whereby the file includes information about the data it contains. NetCDF files also exhibit some platform independence, so they can be accessed by computers with different ways of storing integers, characters, and floating-point numbers. One major advantage of the NetCDF format is its ability to handle large datasets that are otherwise unsuitable for other formats. The NetCDF libraries are designed to be backwards compatible, so data stored in old versions will always be accessible.

### 3.2.4 CF

The Climate and Forecast (CF) convention is intended for use with state estimation and forecasting data, in the atmosphere, ocean, and other physical domains. It is used by many atmospheric institutions and projects around the world. It was designed primarily to address gridded data types such as numerical weather prediction model outputs and climatology data in which data binning is used to impose a regular structure. However, the CF conventions are also applicable to many classes of observational data and have been adopted by a number of groups for such applications. CF originated as a standard for data written in netCDF, but its structure is general and it has been adapted for use with other data formats. For example, using the CF conventions with HDF data has been explored.

CF conventions are for the description of Earth sciences data, intended to promote the processing and sharing of data files. The metadata defined by the CF conventions are generally included in the same file as the data, thus making the file *self-describing*. The conventions provide a definitive description of what the data values found in each CF variable represent, and of the spatial and temporal properties of the data, including information about grids, such as grid cell bounds and cell averaging methods. This enables users of files from different sources to decide which variables are comparable, and is a basis for building software applications with powerful data extraction, grid remapping, data analysis, and data visualisation capabilities.

The CF conventions have been adopted by a wide variety of national and international programs and activities in the Earth sciences. For example, they were required for the climate model output data collected for Coupled Model Inter-comparison Projects (CMIP), which are the basis of Intergovernmental Panel on Climate Change assessment reports. They are promoted as an important element of scientific community coordination by the World Climate Research Programme. They are also used as a technical foundation for a number of software packages and data systems, including the Climate Model Output Rewriter (CMOR), which is post processing software for climate model data, and the Earth System Grid, which distributes climate and other data. The CF conventions have also been used to describe the physical fields transferred between individual Earth system model software components, such as atmosphere and ocean components, as the model runs.

# 4 Comparison of approaches

Table 1 is a first attempt at a high level summary of selected coupling technologies which serves as a means to quickly compare some of the key features associated with couplers. Dunlap *et al.,* (2013) describe an approach to assessing coupler features through feature analysis and the creation of feature diagrams, this approach may be considered as a subsequent, more detailed, analysis was required. Carrying out a feature analysis based approach would be much easier to achieve after the couplers described in the following matrix have been whittled down to a short list of candidates.

The technologies compared in the matrix were specifically identified as being relevant, or potentially relevant, to current BGS activities.

**Table 1. Comparison of coupling approaches**

| | | CSDMS 1.0 | CSDMS 2.0 | OpenMI 1.4 | OpenMI 2.0 | Trident [2] | CESM-CPL 7 | OASIS3-MCT_2.0 | FLUME |
|---|---|---|---|---|---|---|---|---|---|
| Background | Open Source | Yes | Yes | Yes | Yes | Yes<br>Community led since 2013, previously an MS initiative | Yes<br>Subject to the IPR rules of embedded software | Yes<br>(LGPL) | No |
| | Primary research community | Surface dynamics | Surface dynamics + ? | Water | Water<br>+ extra env. disciplines | Oceanography | Climate | Climate | Climate |
| | Central model repository | Yes<br>CSDMS portal | Yes<br>CSDMS portal | Yes but optional<br>FluidEarth | ?<br>Could use FluidEarth but no 2.0 models there yet | Some models held on a site called myExperiment and CSIRO have their own repository | No | No | ? |
| Functionality and implementation details | Visual workflow configuration interface | Yes<br>(CMT) | Yes<br>(CMT) | Yes<br>(FluidEarth) | Yes<br>(FluidEarth) | Yes | Yes (GUI) | Yes (GUI) | Yes (GUI) |
| | Visual 'programming' interface<br>Tools for creating model components that require relatively low level of programming experience, recommended by *Gou D et al.,, 2012* | ?<br>None identified | ?<br>None identified | Partially 3rd party tools e.g. Visual Studio | Partially 3rd party tools e.g. Visual Studio | Yes | ?<br>None identified | ?<br>None identified | ?<br>None identified |

| | | CSDMS 1.0 | CSDMS 2.0 | OpenMI 1.4 | OpenMI 2.0 | Trident [2] | CESM-CPL 7 | OASIS3-MCT_2.0 | FLUME |
|---|---|---|---|---|---|---|---|---|---|
| | Programming language for the framework SDK/wrapper code | BMI functions can be written in C, C++, Fortran (all years), Java and Python. CMT conversion done by CSDMS staff | Same as 1.0 except for the documentation no longer mentions Fortran and Java wrapping tool not available yet | C#,<br>Java?<br>And supported by XML | C#,<br>Java?<br>And supported by XML | .NET (C# & VB.NET) | Fortran | Fortran 77, Fortran 90 and C | ? |
| | Model Languages supported<br>This is a list of languages that pre-compiled models/components can be written in. | C,<br>Fortran (77, 95, 2003),<br>C++,<br>Java,<br>Python | C,<br>Fortran (77, 95, 2003),<br>C++,<br>Java,<br>Python | C#,<br>Java,<br>C, C++,<br>Fortran,<br>Pascal<br>(*And via 3rd party SSW:* MATLAB, Scilab, Python) | C#,<br>Java,<br>C, C++,<br>Fortran,<br>Pascal<br>(SSW for 2.0 planned) | R, Python, TIME<br>Also includes support for ArcGIS and related spatial functions | Fortran | Fortran 77, Fortran 90 and C | ? |
| | Invaseness<br>How much a model needs to be altered before it can be used in the framework<br>*(Jagers, 2010; Lloyd et al., 2011)* | Both OpenMI and CSDMS use similar methods to prepare components for use in each framework, namely implement methods such as initialise, run, describe and finalise. It was not clear from this initial investigation if one was much more invasive than the other. | | | | ? | High, this framework is designed for a set of fixed models representing the key earth systems | Low-intrusiveness, portability and flexibility are key design concepts | Low |

| | | CSDMS 1.0 | CSDMS 2.0 | OpenMI 1.4 | OpenMI 2.0 | Trident [2] | CESM-CPL 7 | OASIS3-MCT_2.0 | FLUME |
|---|---|---|---|---|---|---|---|---|---|
| | Time stepping | Yes | Yes | Yes | Yes | ? | Yes | Yes | Yes |
| | Two way model communication | ? | ? | Yes | Yes | ?<br>Most descriptions involve linear one direction workflows | Yes | Yes | Yes |
| | "Non-temporal data source" e.g. 3D model files or database | Yes<br>68 datasets available on the CSDMS portal 25/10/2013 | Yes | No | Yes | Yes | Yes<br>Typically two dimensional gridded datasets are passed | Yes | Yes |
| | Model metadata<br>The framework supports the capture of metadata, ideally at least partially automated | Yes<br>Via a model metadata file XML | | Yes<br>OMI XML file defines exchange items, more descriptive information can also be captured | | ? | Yes | Yes | Yes |
| | Qualitative model exchange items | ? | ? | No | Yes | ? | ?<br>None identified | ?<br>None identified | ?<br>None identified |
| Utilities | Spatial conversion | Yes<br>Grid based | | Yes | | Yes | Yes<br>Grid based | Yes<br>Grid based | Yes |
| | Temporal scale conversion | Yes | | Yes | | | ? | Yes | Yes |
| | Unit conversions | Yes | Yes | Yes | Yes | Yes | No, models need to use | Via external libraries? | Yes |

| | | CSDMS 1.0 | CSDMS 2.0 | OpenMI 1.4 | OpenMI 2.0 | Trident [2] | CESM-CPL 7 | OASIS3-MCT_2.0 | FLUME |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | standard units | | |
| | Semantic model attribution | ? | Yes | ? | ? | ? | No | ? | ? |
| | Scientific performance[1] | ? | ? | ? | ? | ? | ? | ? | ? |
| | Implementation ready?<br>(Eg. SDK available) | Yes | No? | Yes | Yes | Yes[2] | Yes although limited scope for the work we undertake | Yes | No |

[1] There is a danger that we confuse the evaluation of the technology and the scientific robustness of the solution, especially when the solution is relatively new or designed for another purpose.

[2] Project Trident is a now open source project, originally set up by Microsoft, it is described as 'a scientific workflow workbench'. The most readily available information on an implementation of the Trident software came from publications and website for the 'Hydrologists Workbench', an implementation developed by CSIRO, Australia. The Hydrologists Workbench was used as a proxy for the Trident software when carrying out feature analysis for the matrix, it is therefore possible that some features identified are not fully developed in the original version of the Trident code available via CodePlex.

# 5 Summary and recommendations

## 5.1 SUMMARY

### 5.1.1 Coupling Approaches

The benefits of integrated modelling are not limited to better understanding of complex coupled Earth system processes. Adopting integrated modelling technology means substantial saving in time and costs, since already developed codes can be repurposed and reused in new models.

The couplers described in this report are good exemplars of the types of technologies that are available for model integration. Due to rapid developments in IT, most of the current technologies allow components to communicate dynamically (Lu 2011). While a large number of couplers have been developed up to date, not all of them have been equally successful within the scientific community. Reasons for that could be attributed to specific coupler's features, or lack thereof, e.g.: lack of support for Windows operating system (e.g.: MMS, SME), use of less compatible languages (e.g.: ICMS using MickL, Tarsier using Borland C++), or lack of GUI and use of declarative statements to describe model structure (e.g.: SME, NextFRAMES) (Lu 2011).

OpenMI standard appears to be the most successful and widely accepted within the hydrological community. This does not come as a surprise as OpenMI was developed to specifically target water resources domain. OpenMI's particular feature is that it only sets standards based on interfaces and ensuring that these are implemented correctly is sufficient to make a component complaint (Knapen et al., 2009). The disadvantages include no support for web services (Goodall et al., 2011), and a sequential (pull-driven) communication mechanism, which only allows for single threaded execution (OATC 2010a, OATC 2010b).

CSDMS is suggested by some authors to have a broader hydrologic scope than OpenMI (Peckham 2007). An obvious advantage of CSDMS is its interoperability tool Babel, which, by automatically generating the "glue code", enables communication between models written in different languages (Peckham 2007). CSDMS is intended to be interoperable with ESMF and OpenMI (Peckham 2007); integration with these different frameworks opens opportunities for cross-domain environmental research.

While CCA is intended for high-performance computing applications, it does not provide "any automatic way for the software to take advantage of multiple processors" (Peckham 2007). ESMF, on the contrary, provides direct path to parallel computation through domain decomposition (Peckham 2007). While ESMF is considered rather intrusive (Lawrence et al., Manuscript), OASIS using “a concurrent multiple executable approach requires minimal modification to the existing component code" (Valcke et al., 2012)

Although certain aspects of frameworks are similar, for example interfaces of CCA, ESMF, OASIS, OMS, OpenMI, and TIME all use initialise, run, finalise, get, and set concepts, the amount of code needed to integrate models varies significantly (Jagers 2010). OMS 3.0 is a lightweight framework, which uses metadata approach to integrate models. In the study by Lloyd et al., (2011) it was shown to be the least invasive in comparison with other tested frameworks (OMS 2.2, ESMF 3.1.1C, ESMF 3.1.1Fortran, OpenMI 1.4, CCA 0.6.6), e.g.: OMS 3.0 required the least amount of code for implementation of the Thornthwaite model (Lloyd et al., 2011, David et al., 2013).

TIME, likewise OMS, uses metadata approach to integrate models. The primary difference it that annotations in TIME are embedded in the source code, while in OMS they are encoded as declarations in external XML files (Lu 2011). An evident advantage of TIME is its GIS functionality; a considerable disadvantage is its lack of support for non-TIME models and for interoperability with other frameworks (Fitch and Bai 2009). However, efforts have been undertaken to overcome this limitation by developing software based on web services, which would enable TIME models to interface with other applications (Fitch and Bai 2009).

The use of workflows to integrate hydrologic models is still rather limited (Lu 2011). The challenge comes in refactoring the existing codes into reusable workflow activities. Deciding on the right granularity and complexity of the individual activities is critical for constructing a good workflow (Cuddy and Fitch 2010). Although, still not a common practise, a few high profile projects are exploring ways to employ workflows for water resources modelling. Kepler was suggested to replace OpenMI Configuration Manager in the two-way coupled system linking hydrology and climate models (Goodall et al., 2013, Saint and Murphy 2010); the rationale for this being that Kepler is more extensive and versatile than OpenMI (Saint and Murphy 2010). EVO developers are looking into ways to increase customisation by implementing workflow execution such as that provided by Taverna (Elkhatib et al., 2013). Hydrologists' Workbench, employing Microsoft's TRIDENT, is being developed by The *Commonwealth Scientific and Industrial Research Organisation* (CSIRO) to help fulfil the Bureau of Meteorology’s legal obligation for producing monthly regional water situation reports based an integrated data and modelling system's output (Cuddy and Fitch 2010, CSIRO 2013). The main advantages of using a workflow are the automation of repetitive tasks, and the ability to document model runs and record the workflow sequence as a file, which guarantees repeatability, auditability, and transparency of scientific computations (Lu 2011, CSIRO 2013).

BFG offers a novel approach to model integration, which "isolates the science that a model performs from the code used to control and couple it with other models" (BFG 2013). When employing BFG, no changes to the component's code are needed, since a wrapper code is generated which enables it to fit within a framework of choice. Furthermore, models integrated using BFG are “resistant” to the framework’s modifications (Warren et al., 2008). BFG goes beyond a typical coupling technology that imposes architectural requirements on components, hence it allows for models to be easily exchanged.

Employing tight coupling enables "use of the most efficient algorithms to solve complicated numerical problems, for example fully-coupled systems of differential equations" (Goodall et al., 2011). However, an obvious disadvantage of tight coupling is the difficulty with integrating models that do not comply with the framework requirements (Goodall et al., 2011). "In contrast, a loosely-coupled approach requires only the standardisation of interfaces and data exchanges" (Goodall et al., 2011). The advantages of using loosely-coupled, service-oriented approach extend beyond the ability to integrate disparate models. The user does not have to be concerned with large computing resources or datasets needed. Each model operates in its own hardware environment and the system's functionality can be accessed through web services interfaces (Goodall et al., 2011). In the case of the cloud technology, the resources are available on demand, which reduces the computing equipment and run-time costs (e.g.: electricity, administration, etc.) (EVO 2013). Hence, using web services frees user from some of the technological concerns, allowing them to focus on the scientific aspect of their work (EVO 2013). Service-oriented technology, however, does not come without its challenges. The design of such a system need to consider potential performance, reliability and security issues (Goodall et al., 2011). The primarily concern is the performance associated with modelling fully-coupled processes with large data transfers and tasks with long execution times (Goodall et al., 2011). Reliability might be a problem as remote servers can become temporarily unavailable (Goodall et al., 2011). Additionally, security must be ensured to prohibit unauthorised use (Goodall et al., 2011).

All of the described advances in the scientific computing technology constitute a significant progress toward comprehensive and efficient modelling systems. Such systems are essential to address water resources management challenges that arise due to the climate change on one hand, and increasing and conflicting demands on the other.

### 5.1.2 One way file transfer formats

BGS has investigated significant resources in developing Information Management (IM) to serve data both internally and externally. The experience built up in this process as well as the relevant infrastructure is useful in developing any IEM solution. This experience is based around using Oracle databases and the standards associated with it and include:

- Catalog Service for the Web (CSW) is one part of the OGC Catalog Service specification that they describe as follows "Catalogue services support the ability to publish and search collections of descriptive information (metadata) for data, services, and related information objects. Metadata in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software. Catalogue services are required to support the discovery and binding to registered information resources within an information community."
- Web Feature Service (WFS) from the OGC provides an interface which allows clients to query and access geographical features across the web.
- Geospatial Data Abstraction Library (GDAL) is, according to gdal.org *"a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing."*

Alongside these standards BGS has adopted OpenMI 1.4 as a model linking standard. However, whilst this version isn't designed to exchange static data, the revision OpenMI 2.0 can and offers promise for linking with static datasets.

The climate community has adopted a number of standards for their data. These include Gridded Binary (GRIB), Network Common Data Form (netCDF) or the Hierarchical Data Format (HDF) system. All are intended for use with modern atmospheric datasets, which encompass information about the atmosphere, sea, and ocean and are used for modelled and observed data. These standards are supplemented by a recently conceived Climate and Forecasting (CF) standard which aims to distinguish quantities (descriptive, units, prior processing, etc) and to spatio-temporally locate data as a function of other independent variables, such as a coordinate system.

## 5.2 RECOMMENDATIONS

### 5.2.1 Couplers and workflow engines

Given the range of coupler technologies open to the BGS, and others, it is easy to spend a long time reading literature on the theory behind each and attempting to evaluate the relative value of one over another. It would be prudent to identify a shortlist of candidate technologies for hands on evaluation, the aim being to assess model performance over a range of desirable model coupling features.

Given the experience the BGS has with the OpenMI 1.4 standard and ongoing efforts to implement a composition in OpenMI 2.0, this report recommends the later is shortlisted for inclusion in the coupler evaluation process.

CSDMS provides an alternative approach to OpenMI in as much as the philosophy behind the technology is more related to the use of High Power Computing, something that the BGS has relatively little experience of. One of the key similarities between CSDMS and OpenMI is the use of the Initialise – Run – Finalise (IRF) principle raising the potential for code re-use across both technologies, allowing the modellers to select the best coupling option for the job without the need for extensive re-factoring. Therefore, the CSDMS technology is recommended for the shortlist, although it is unclear whether this should be CSDMS1.0 or CSDMS2.0, the later was launched in 2013 but relatively little information was found during this investigation about real world applications of the technology.

Given that the remit of the study was coupling technology within the hydrological and atmospheric sciences, it is necessary to extend the scope outside of these communities. A recently developed approach that shows promise is OASIS-LMF (Loss Modelling Framework) whose aim is to provide a methodology to provide risk assessments for the Insurance and re-insurance industry. It is suggested that the trial composition be tested using OASIS-LMF.

Finally this report recommends the evaluation of one or more 'workflow engines', the Trident project is open source and appears to have been successfully used by CSIRO to develop the 'Hydrologists Workbench', however, there does not seem to be a particularly strong community of users outside of CSIRO. It seems unlikely that this is a solution the BGS should spend too much time evaluating unless a strong contact can be established with members of the CSIRO team involved in the 'Hydrologists Workbench'.

Another workflow option is the Kepler project, it appears to have an active community, producing numerous peer reviewed publications, discussing topics such as environmental sensor networks, climate change and species distribution, due to time spent reviewing other options these papers were not studied in depth. (see https://kepler-project.org/publications?tags=keplerworkflow)

It may also be possible to use existing workflow tools within the BGS such as FME (http://www.safe.com/fme/fme-technology/). It is recommended that in-house experts in FME (e.g. Tony Myers) should be consulted on the capabilities of the system to see if this approach is worth taking any further.

This activity should be linked with the TSB-AHRC funded project Confluence. This project, led by HR Wallingford and undertaken in conjunction with Nottingham University aims to assess the use of the Pyxis workflow tool. The project will involve including BGS models in Pyxis and assessing how this improves the management of the overall workflow.

#### 5.2.1.1 Evaluation criteria

The shortlisted coupler technologies should be evaluated by directly comparing the scientific accuracy, ease of use and feature richness when applied to a single linked model. The exact nature of the linked model scenario should be designed in consultation with geologists, mathematical modellers and senior staff within the BGS Environmental Modelling Directorate to ensure the scenario being considered is consistent with current and anticipated future challenges.

Once a scenario has been defined it will be possible to identify the key resources and components required to answer the question, i.e. which datasets, models and conversion functions are required. The following diagrams show a possible scenario to use in our proposed bench test, the entities in Figure 4 are generic whereas Figure 5 provides a real world context containing BGS examples.

#### 5.2.1.2 Data file formats

To ensure that the compositions described use appropriate standards then it is necessary to define a set of internationally recognised ones to use. Given the reliance of data from BGS corporate databases then those used for the Geological Object Store should be used. These include CSW and GDAL. Alongside these, the use of NetCDF and CF for large datasets should be investigated.

Finally the use of WFS for data transfer between dynamic models should be included within one part of the composition.

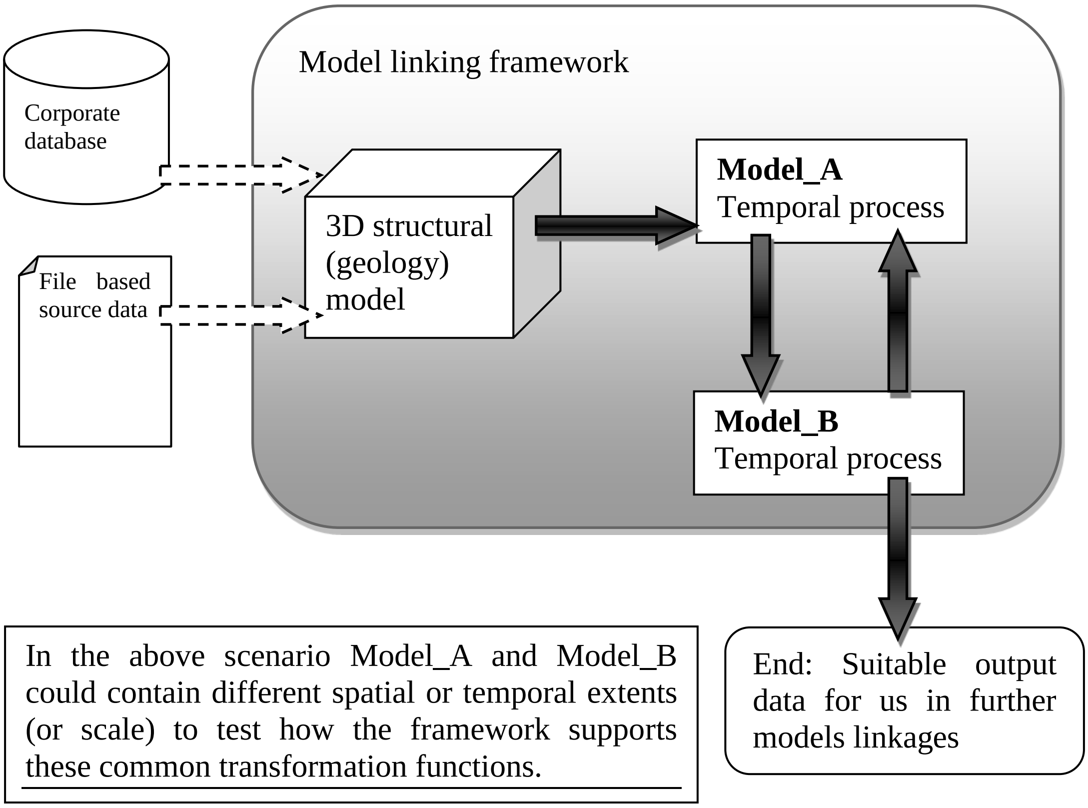



**Figure 4. Conceptual basis for a composition for testing different couplers**

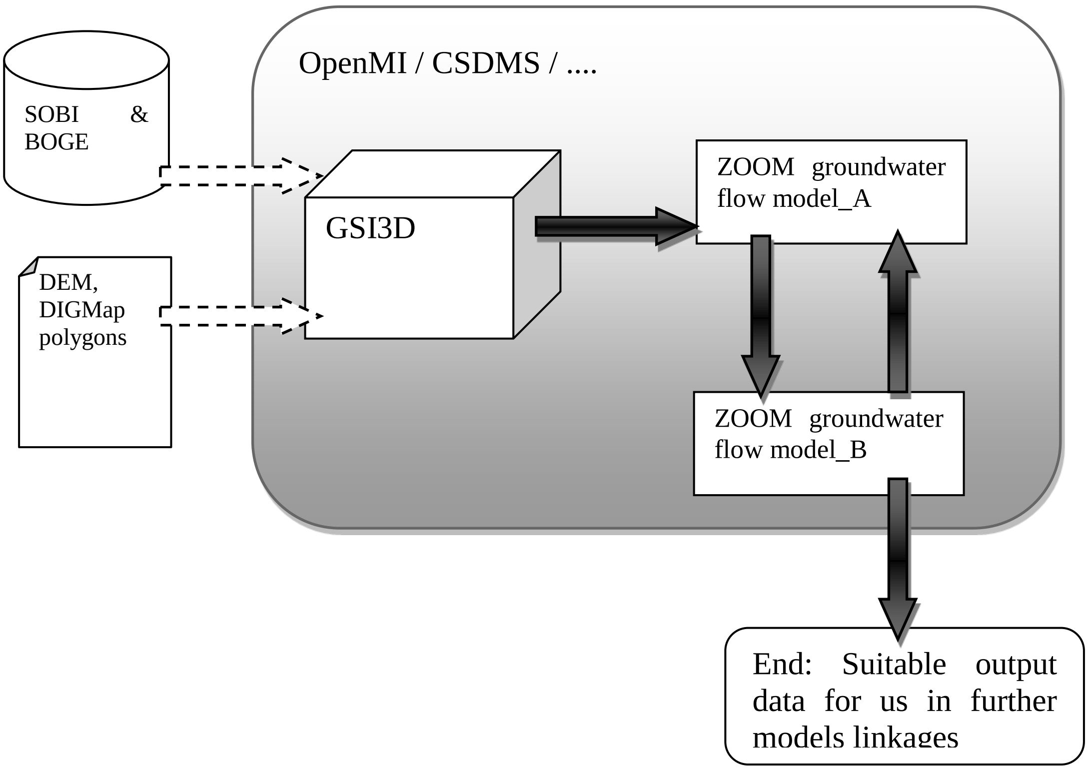



**Figure 5. Potential example composition for testing different couplers**

# References

ARELLANO, A E. 2011. *The Water Cycle in the Mediterranean Region and the Impacts of Climate Change.* PhD thesis, Max-Planck-Institute for Meteorology.

ARGENT, R M., PERRAUD, J M., RAHMAN, J M., GRAYSON, R B. & PODGER, G M. 2009. A new approach to water quality modelling and environmental decision support systems. *Environmental Modelling & Software 2*4**,** 809-818.

ASHTON, A D., HUTTON, E W H., KETTNER, A J., XING, F., KALLUMADIKAL, J., NIENHUIS, J. & GIOSAN, L. 2013. Progress in coupling models of coastline and fluvial dynamics. *Computers and Geosciences,* 53 21-29.

BFG 2013. *Bespoke Framework Generation Website.* [Online]. Centre for Novel Computing University of Manchester. [cited 14 November 2013]. Available: http://cnc.cs.man.ac.uk/projects/bfg.php.

CAUBEL, A., DECLAT, D., FOUJOLS, M.-A., LATOUR, J., REDLER, R., RITZDORF, H., SCHOENEMEYER, T., VALCKE, S. & VOGELSANG, R. 2005. The PRISM couplers: OASIS3 and OASIS4. *Geophysical Research Abstracts* [Online], 7.

Craig, T., 2011. CPL7 User's Guide. http://www.cesm.ucar.edu/models/cesm1.0/cpl7/cpl7_doc/ug.pdf (accessed 14.02.14).

CSDMS 2013. *The Community Surface Dynamics Modeling System Website* [Online]. [cited 14 November 2013]. Available: http://csdms.colorado.edu/wiki/Main_Page.

CSIRO 2013. *The Workbench (TWB) Website* [Online]. Last revised 8 February 2013. [cited 14 November 2013]. Available: ttps://wiki.csiro.au/pages/viewpage.action?pageId=305136129.

CUDDY, S M. & FITCH, P. 2010. Hydrologists Workbench - a hydrological domain workflow toolkit. *In:* SWAYNE, D A., YANG, W., VOINOV, A A., RIZZOLI, A. & FILATOVA, T. (eds.) *International Congress on Environmental Modelling and Software, Modelling for Environment's Sake, Fifth Biennial Meeting.* Ottawa, Canada: International Environmental Modelling and Software Society (iEMSs).

DAVID, O., ASCOUGH II, J C., LLOYD, W., GREEN, T R., ROJAS, K W., LEAVESLEY, G H. & AHUJA, L R. 2013. A software engineering perspective on environmental modeling framework design: The Object Modeling System. *Environmental Modelling and Software,* 39 201-213.

DAVID, O., CARLSON, J R., LEAVESLEY, G H., ASCOUGH II, J C., GETER, F W., ROJAS, K W. & AHUJA, L R. 2010. Object Modeling System v3.0 Developer and User Handbook.

DELUCA, C., OEHMKE, R., NECKELS, D., THEURICH, G., O'KUINGHTTONS, R., DE FAINCHTEIN, R., MURPHY, S. & DUNLAP, R. Enhancements for Hydrological Modeling in ESMF. American Geophysical Union Fall Meeting 2008.

DEELMAN, E., GANNON, D., SHIELDS, M. & TAYLOR, I. 2009. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems 2*5**,** 528-540.

DE ROURE, D. & GOBLE, C. 2009. Software Design for Empowering Scientists. *Ieee Software 2*6**,** 88-95.

DKRZ 2013. *PRISM Program for Integrated Earth System Modelling Website* [Online]. [cited 14 November 2013]. Available: http://www.dkrz.de/daten-en/wdcc/projects_cooperations/past-projects/prism.

DUNLAP, R., RUGABER, S. & LEO, M. 2013. A feature model of coupling technologies for Earth System Models. *Computers and Geosciences,* 53**,** 13-20.

ELKHATIB, Y., BLAIR, G S. & SURAJBALI, B. Experiences of Using a Hybrid Cloud to Construct an Environmental Virtual Observatory. 3rd International Workshop on Cloud Data and Platforms 2013 Prague Czech Republic.

ESMF 2013. *Earth System Modeling Framework Website* [Online]. [cited 14 November 2013]. Available: http://www.earthsystemmodeling.org/.

EVO 2013. *Environmental Virtual Observatory Website* [Online]. [cited 14 November 2013]. Available: http://www.evo-uk.org/.

eWater CRC 2013. eWater Cooperative Research Centre Website [Online]. Available: http://www.ewater.com.au/ [cited 2/12/2013].

FEKETE, B M., WOLLHEIM, W M., WISSER, D. & VÖRÖSMARTY, C J. 2009. Next generation framework for aquatic modeling of the Earth System. *Geosci. Model Dev. Discuss. 2* 279-307.

FITCH, P. & BAI, Q F. 2009. A standards based web service interface for hydrological models. *18th World Imacs Congress and Modsim09 International Congress on Modelling and Simulation: Interfacing Modelling and Simulation with Mathematical and Computational Sciences*, 873-879.

FLUIDEARTH 2013. *FluidEarth HR Wallingford Website* [Online]. [cited 14 November 2013]. Available: http://fluidearth.net/default.aspx.

Ford, R. W., and Riley, G. D., 2003. Towards the Flexible Composition and Deployment of Coupled Models. World Scientific: 2003: 189-195.

GIJSBERS, P., HUMMEL, S., VANECEK, S., GROOS, J., HARPER, A., KNAPEN, R., GREGERSEN, J., SCHADE, P., ANTONELLO, A. & DONCHYTS, G. From OpenMI 1.4 to 2.0. International Congress on Environmental Modelling and Software Modelling for Environment's Sake, Fifth Biennial Meeting 2010 Ottawa, Canada.

Giles, Jeremy RA, et al. "Data, and research for applications and models (DREAM): scoping study report." (2010).

GOODALL, J L., ROBINSON, B F. & CASTRONOVA, A M. 2011. Modeling water resource systems using a service-oriented computing paradigm. *Environmental Modelling and Software 26*, 573-582.

GOODALL, J L., SAINT, K D., ERCAN, M B., BRILEY, L J., MURPHY, S., YOU, H., DELUCA, C. & ROOD, R B. 2013. Coupling climate and hydrological models: Interoperability through Web Services. *Environmental Modelling and Software,* 46 250-259.

GRAHAM, D N., CHMAKOV, S., SAPOZHNIKOV, A. & GREGERSEN, J B. OpenMI coupling of Modflow and Mike 11. *In:* GOURBESVILLE, P., CUNGE, J., GUINOT, V. & LIONG, S Y., eds. 7th International Conference on Hydroinformatics 2006 Nice, France.

Gregory, J. M., 2003. The CF metadata standard. Technical Report 8, CLIVAR

GURNEY, R., EMMETT, B., MCDONALD, A., BLAIR, G., BUYTAERT, W., FREER, J E., HAYGARTH, P., REES, G., TETZLAFF, D. & EVO SCIENCE TEAM. The Environmental Virtual Observatory: A New Vision for Catchment Science. American Geophysical Union Fall Meeting 2011.

HENDERSON, I. 2006. *GENIE BFG. University of Bristol Geography Source Website.* [Online]. Last revised 26 August 2008. [cited 14 November 2013]. Available: https://source.ggy.bris.ac.uk/wiki/GENIE_BFG.

HYACINTS 2013. *Hydrological Modelling for Assessing Climate Change Impacts at different Scales Project Website* [Online]. Last revised 26 June 2009. [cited 14 November 2013]. Available: http://hyacints.dk/main_uk/main.html.

HYDRODESKTOP 2013. *HydroDesktop CUAHSI Open Source Hydrologic Data Tools Website* [Online]. Last revised 13 March 2012. [cited 14 November 2013]. Available: http://his.cuahsi.org/hdhelp/welcome.html.

ISIS 2013. *ISIS User Community Website* [Online]. [cited 14 November 2013]. Available: http://www.isisuser.com/.

JAGERS, H R A. Linking Data, Models and Tools: An Overview. International Congress on Environmental Modelling and Software Modelling for Environment's Sake, Fifth Biennial Meeting 2010 Ottawa, Canada.

Kauffman, B. G., Jacob, R., Craig, T., and Large, W. G., 2004: The CCSM Coupler version 6.0: User's guide, source code reference, and scientific description. http://www.ccsm.ucar.edu/models/ccsm3.0/cpl6/users guide/users guide.html (accessed 13.12.13).

KEPLER 2013. *The Kepler project* [Online]. [cited 14 November 2013]. Available: https://kepler-project.org/.

KEPLER. 2013a. Getting started with Kepler Manual [Online]. [cited 14 November 2013]. Available: https://kepler-project.org/.

KEPLER. 2013b. Kepler User Manual [Online]. [cited 14 November 2013]. Available: https://kepler-project.org/.

KNAPEN, M J R., VERWEIJ, P., WIEN, J E. & HUMMEL, S. 2009. OpenMI - The universal glue for integrated modelling? *18th World IMACS/MODSIM Congress.* Cairns, Australia.

LAWRENCE, B N., BALAJI, V., CARTER, M., DELUCA, C., EASTERBROOK, S., FORD, R., HUGHES, A. & HARDING, R. Manuscript. Bridging Communities: Technical Concerns for Integrating Environmental Models.

LLOYD, W., DAVID, O., ASCOUGH II, J C., ROJAS, K W., CARLSON, J R., LEAVESLEY, G H., KRAUSE, P., GREEN, T R. & AHUJA, L R. 2011. Environmental modeling framework invasiveness: Analysis and implications. *Environmental Modelling & Software 2*6**,** 1240-1250.

LU, B. 2011. *Development of A Hydrologic Community Modeling System Using A Workflow Engine.* PhD thesis, Drexel University.

MOORE, R., GIJSBERS, P., FORTUNE, D., GREGERSEN, J., BLIND, M., GROOSS, J. & VANECEK, S. 2010. OpenMI Document Series: Scope for the OpenMI (Version 2.0). *In:* MOORE, R. (ed.).

MOORE, R V. & TINDALL, C I. 2005. An overview of the open modelling interface and environment (the OpenMI). *Environmental Science & Policy,* 8 279-286.

OASIS 2013. *OASIS Coupler Website* [Online]. [cited 14 November 2013]. Available: https://verc.enes.org/oasis.

OATC 2010a. OpenMI Document Series: The OpenMI 'in a Nutshell' for the OpenMI (Version 2.0). The OpenMI Association Technical Committee. *In:* MOORE, R. (ed.).

OATC 2010b. OpenMI Document Series: OpenMI Standard 2 Specification for the OpenMI (Version 2.0). The OpenMI Association Technical Committee. *In:* MOORE, R. (ed.).

OMS 2013. *Object Modelling System Website* [Online]. [cited 14 November 2013]. Available: http://www.javaforge.com/project/oms.

OPENMI 2013. *The OpenMI Association Website* [Online]. [cited 14 November 2013]. Available: http://www.openmi.org/.

PECKHAM, S D. Evaluation of Model Coupling Frameworks for Use by the Community Surface Dynamics Modelling System (CSDMS). American Geophysical Union Fall Meeting 2007.

PECKHAM, S D. & GOODALL, J L. 2013. Driving plug-and-play models with data from web services: A demonstration of interoperability between CSDMS and CUAHSI-HIS. *Computers and Geosciences,* 53**,** 154-161.

PECKHAM, S D. & HUTTON, E. Componentizing, standardizing and visualizing: How CSDMS is building a new system for integrated modeling from open-source tools and standards. American Geophysical Union Fall Meeting 2009.

PECKHAM, S D., HUTTON, E W H. & NORRIS, B. 2013. A component-based approach to integrated modeling in the geosciences: The design of CSDMS. *Computers and Geosciences,* 53**,** 3-12.

RAHMAN, J. M., SEATON, S. P., PERRAUD, J.-M., HOTHAM, H., VERRELLI, D. I. & COLEMAN, J. R. It's TIME for a New Environmental Modelling Framework. Proceedings of MODSIM International Congress on Modelling and Simulation 2003 Townsville, Australia. Modelling and Simulation Society of Australia and New Zealand Inc., 1727-1732.

SAINT, K. & MURPHY, S. End-to-End Workflows for Coupled Climate and Hydrological Modeling. International Congress on Environmental Modelling and Software, Modelling for Environment's Sake, Fifth Biennial Meeting 2010 Ottawa, Canada.

SROKA, J., HIDDERS, J., MISSIER, P. & GOBLE, C. 2010. A formal semantics for the Taverna 2 workflow model. *Journal of Computer and System Sciences,* 76**,** 490-508.

TAVERNA. 2013. *Taverna Workflow Management System Website* [Online]. Available: http://www.taverna.org.uk/ [cited 2/12/2013].

VALCKE, S., BALAJI, V., CRAIG, A., DELUCA, C., DUNLAP, R., FORD, R W., JACOB, R., LARSON, J., O'KUINGHTTONS, R., RILEY, G D. & VERTENSTEIN, M. 2012. Coupling technologies for Earth System Modelling. *Geoscientific Model Development,* 5**,** 1589 - 1596.

VALCKE, S., BUDICH, R., CARTER, M., GUILYARDI, E., FOUJOLS, M A., LAUTENSCHLAGER, M., REDLER, R., STEENMAN-CLARK, L. & WEDI, N. The PRISM software framework and the OASIS coupler. *In:* HOLLIS, A J. & KARITO, A P., eds. The eighteenth annual BMRC Modelling Workshop "The Australian Community Climate and Earth System Simulator (ACCESS) - challenges and opportunities" 2006 Melbourne, Australia.

Valcke, S., Craig, T., and Coquart, L., 2013. OASIS3-MCT User Guide, OASIS3-MCT 2.0, Technical Report, TR/CMGC/13/17, CERFACS/CNRS SUC URA No 1875, Toulouse, France.

VALCKE, S. & MOREL, T. 2006. OASIS and PALM, CERFACS couplers. Technical Report TR/CMGC/06/38.

WARREN, R., DE LA NAVA SANTOS, S., ARNELL, N W., BANE, M., BARKER, T., BARTON, C., FORD, R., FÜSSEL, H M., HANKIN, R K S., KLEIN, R., LINSTEAD, C., KOHLER, J., MITCHELL, T D., OSBORN, T J., PAN, H., RAPER, S C B., RILEY, G., SCHELLNHÜBER, H J., WINNE, S. & ANDERSON, D. 2008. Development and illustrative outputs of the Community Integrated Assessment System (CIAS), a multi-institutional modular integrated assessment approach for modelling climate change. *Environmental Modelling and Software 2*3**,** 1215-1216.

WILKINSON, M., BEVEN, K., BREWER, P., EL-KHATIB, Y., GEMMELL, A., HAYGARTH, P., MACKAY, E., MACKLIN, M., MARSHALL, K., QUINN, P., STUTTER, M., THOMAS, N. & VITOLO, C. The Environmental Virtual Observatory (EVO) local exemplar: A cloud based local landscape learning visualisation tool for communicating flood risk to catchment stakeholders. EGU General Assembly 2013 Vienna Austria.

WMO, 1995. Manual on Codes. WMO Publication Number 306, Volume 1, Part B, 1995 Edition, plus Supplements.

WOLLHEIM, W. FrAMES - Framework for Aquatic Modeling of the Earth System. Denitrification Modeling Across Terrestrial, Freshwater and Marine Systems Workshop 2006 Millbrook, New York.

# Appendix 1 – Supplementary material

CSDMS
Community Surface Dynamics Modeling System
(2013) csdms.colorado.edu/wiki/Main_Page

The Workbench (TWB) Documentation
CSIRO Hydrologists Workbench online documentation
(2013) https://wiki.csiro.au/display/HWB/The+Workbench+%28TWB%29+Documentation

OASIS-LMF (Loss Modelling Framework) - http://www.oasislmf.org/

B.N. Lawrence et al 2012 "Bridging Communities: Technical Concerns for Integrating Environmental Models"