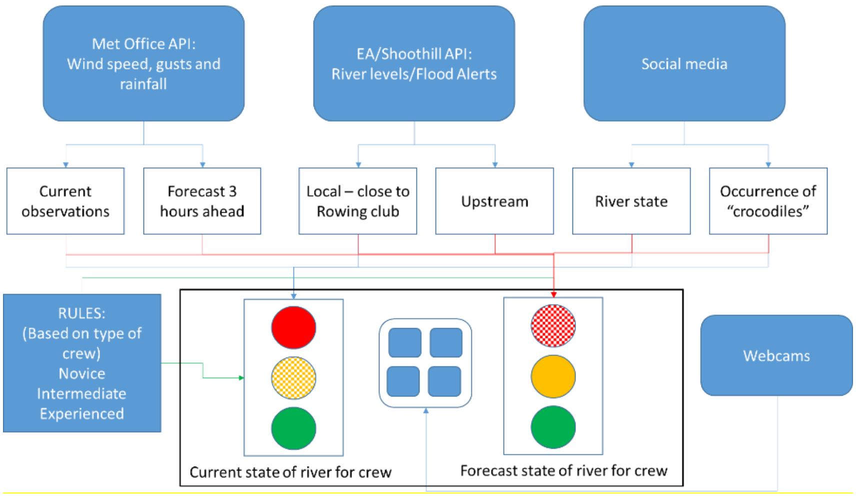
British Geological Survey
NATURAL ENVIRONMENT RESEARCH COUNCIL

York City Environmental Observatory: WP4 Model Integration

Groundwater Programme

Commissioned Report CR/17/022



BRITISH GEOLOGICAL SURVEY

GROUNDWATER PROGRAMME

COMMISSIONED REPORT CR/17/022

York City Environmental Observatory: WP4 Model Integration

A G Hughes and A Powell

Keywords

York; YCEO; models; integration; pilot.

Front cover

Diagram showing workflow for rowing use case

Bibliographical reference

HUGHES A. G.. 2017. York City Environmental Observatory: WP4 Model Integration. *British Geological Survey Commissioned Report,* CR/17/022. 43pp.

Keyworth, Nottingham British Geological Survey 2017

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, Department of Enterprise, Trade & Investment, Dundonald House, Upper Newtownards Road, Ballymiscaw, Belfast, BT4 3SB
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

The assistance of colleagues on the York ULP project is grateful acknowledged, namely Alistair Boxall, Lorraine Youdis, Ian Cunningham and Claudio Rodriguez-Nuno. Jon Timmis and colleagues at SimOmics are also gratefully acknowledged for discussions.

Andrew Allen, Nottingham University and Borkur Sigurbjornsson, Tombolo project Future Cities Catapult are acknowledged for spending the time to explain their projects. Finally colleagues at BGS: Steph Bricker, Carl Watson and Jon Ford are gratefully acknowledged for their input to the report and David Macdonald for his review.

Contents

FIGURES

TABLES

Summary

This work is produced as part of the York Urban Living Pilot (ULP) project an RCUK funded project to develop the York City Environmental Observatory (YCEO). The project is led by York University with York City Council as lead partners and other organisations including BGS. Split into a number of Work Packages (WPs), WP4 aims to set out the design for the implementation phase of the YCEO. This report forms part of the deliverables of WP4 and is concerned with model integration and aims to determine which models are available for the York area and how they could be integrated into the YCEO.

The information was obtained by internet searches, literature review, and investigating known projects to determine: a) what models are out there and b) how model integration could be undertaken

Evidence showed limited metadata for models of the York area, even though there is knowledge of models available. There is also limited examples of model integration platforms in the "smart city" context or even urban environments. However, exemplars exist including predictive analytics built on open data platforms and workflow approaches such as the Innovate UK funded Tombolo project. Application programming Interfaces (APIs) also offer possibilities in the UK with the Met Office and the Environment Agency making both observed data and model forecasts available via this method. A simple example is presented using APIs to determine when it is safe to row ont eh River Ouse, York.

Model integration is problematic, but it can take many forms and Tombolo offers the opportunity to enable pre-formed solutions to be made available. The YCEO model platform should be based on pre-formed solutions using the Tombolo approach with semantic reasoning at its basis.

1 Background and Introduction

1.1 BACKGROUND

This report forms part of the outputs for the York City Environmental Observatory (YCEO) project. This project, funded by RCUK aims to pilot the development of an Environmental Observatory for York City. The project is organised into four main Work Packages (WPs):

1. Characterising stakeholder perceptions of the importance of York city environment for health, well-being and economy (led by Stockholm Environment Institute - SEI)
2. Development of a York city environment open data and model platform (led by York City Council)
3. Evaluation of data availability and data needs for answering key questions around how the quality of York's City Environment links to health , well-being and economic parameters (led by University of York)
4. Design of the YCEO (led by SimOmics jointly with the University of York)

WP4 aims to take the findings of WP1-3 and make recommendations for the next phase of development of the Environmental Observatory. BGS role within WP4 is to describe the availability of models, how they can be integrated and what the modelling part of the YCEO should look like. This work is reported here.

1.2 INTRODUCTION

Questions and their answer form an important part of everyday life. This is the case for decision-makers from the householder through to governmental level. These questions, which are often easy to articulate, but difficult to solve, include:

- Given that I suffer from asthma should I walk this way to work today?
- Should I grant planning permission for a new retail development?
- What will the effect of this national policy have on flood resilience?

Given that increasing numbers of people live in cities then questions related to city life are becoming increasingly important to address. This should be the case for a range of decision-makers. The vision for the YCEO is that users from a range of backgrounds and with differing technical skills should be able to address questions similar to those outlined above.

1.3 REPORT STRUCTURE

This report describes the models instances identified as existing within the York area and related nationally available services that are available; it details how they may be integrated, presents national and international exemplars of such integration and finally sets out how modelling can be incorporated into the YCEO.

2 Current models available

2.1 WHAT HAS BEEN LOOKED AT?

The aim of the YCEO is to be able to address user-driven questions. To underpin this the availability of environmental models that support or describe the flow of fluids and pollutant mass around the York city area has been investigated. This could be the supply of water, flood risk or the air pollution caused by cars within the city environs. Specifically, this is driven by the following likely use cases:

- Rowing
- Flood
- Air quality
- Heritage and culture

The latter three use cases are currently under development in WP2. The assumption has been made that these will need to be addressed by an understanding of fluid and pollutant mass flow in and around York City.

These need to answer “What-if” scenarios and predictions based on changes to the current status quo. Therefore, it is necessary to include not just data and data analytics but models as well. The latter enable changes to be made to a system and the impacts of these changes examined.

2.2 HOW WAS THIS UNDERTAKEN?

The following was used to extract knowledge of models:

- Organisational knowledge from the partners involved in the YCEO consortium
- web searches
- Searches in the peer-reviewed literature

Keywords: York, UK, flooding, groundwater, air quality, pollution, models

Basic metadata was collected to enable the model to be defined, described and incorporated into the YCEO.

2.3 WHAT HAS BEEN DISCOVERED?

Generally speaking, there is difficulty discovering suitable models: model metadata are limited and there are no suitable environmental portals for providing models. Whilst this is likely to improve in the next five years as the recognition increases that models need to be described and made available, this doesn’t change the availability for the YCEO project.

A summary of those models found using the basic search approach described above are presented in Table 1.

However there are nationally available models and datasets that provide relevant information for the project. These are typically made available as Application Programming Interfaces (APIs) and provide data and model outputs or a combination of the two. As described below APIs are a way by which the results from models can be made available to any user that has access to the internet and the relevant information. The examples presented in Table 1 are APIs provided by the Met Office (observations from weather station and forecasts) and Shoothill (flood extent, river and groundwater levels on behalf of the Environment Agency).

Whilst this is not providing a direct link to the model itself, the API provides access to model results that are stored on a central server and made available via a standard web interface.

Table 1. Summary of model availability

Name	Short name	Purpose	Organisation	Contact	Spatial Extent		Time periods		Code	Further information
					Bottom Left	Top right	Start	End		
York-Haxby 3D geological model	York 3D geo	3D Framework model to enhance geological understanding of superficial deposits	BGS	enquiries@bgs.ac.uk	460000, 450000	465000, 460000	n/a	n/a	GSI3D	Report: http://nora.nerc.ac.uk/501836/
York-Ripon Trias groundwater model	York GW model	Improve understanding of Triassic Sherwood Sandstone aquifer between Ripon and York, its role and water resources	JBA	info@jbaconsulting.com	??	??	??	??	MODFLOW	http://www.jbaconsulting.com/project/conceptual-and-numerical-models-york-sherwood-sandstone-aquifer
Upper Ouse flood model	Ouse flood model	Model flood extent and depth for particular return periods	??	??	??	??	??	??	??	https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/289228/River_Ouse_Catchment_Flood_Management_Plan.pdf

Air quality model York	ADMS-York		CERC		??	??	??	??	ADMS	http://www.cerc.co.uk/
QUESTOR water quality model	QUESTOR-WQ	Simulating riverine water quality	CEH	See further information	??	??	??	??	QUESTOR	catalogue.ceh.ac.uk/documents/53f3d4fd-8f62-4d35-a232-053a4e961ba0
Shoothill API		API providing observed hydrometric data and flood forecasts	Developed on behalf of the Environment Agency	support@shoothill.com	UK-wide		Depends on data availability		n/a	www.shoothill.com/our-api/
Met Office Datapoint API	MO API	API providing observed meteorological variables and met forecasts	Met Office	enquiries@metoffice.gov.uk	UK-wide		3 hours ahead forecast		n/a	http://www.metoffice.gov.uk/datapoint

3 Integration approaches

3.1 WHY UNDERTAKE INTEGRATION?

Single instance modelling aims to provide the solution to a range of problems. However, there are limitations and so model integration aims to provide a better "end product". By combining different models together the end result can answer the question posed better than by using a single existing model. The type of questions described above can often only be addressed by integrated modelling.

To address the questions relevant to urban living / YCEO the single "silo" discipline approach within biophysical disciplines, e.g. groundwater and surface water is not sufficient. Therefore, a cross disciplines: socio-economic and biophysical is required. Models are developed by different organisations and need to be linked together. The models can be thought of as unit operations requiring data and some form of processing. Figure 1 describes the model as a "unit operation", e.g. a single definable step in any process: data is fed in, some form of operation occurs and the output is produced.

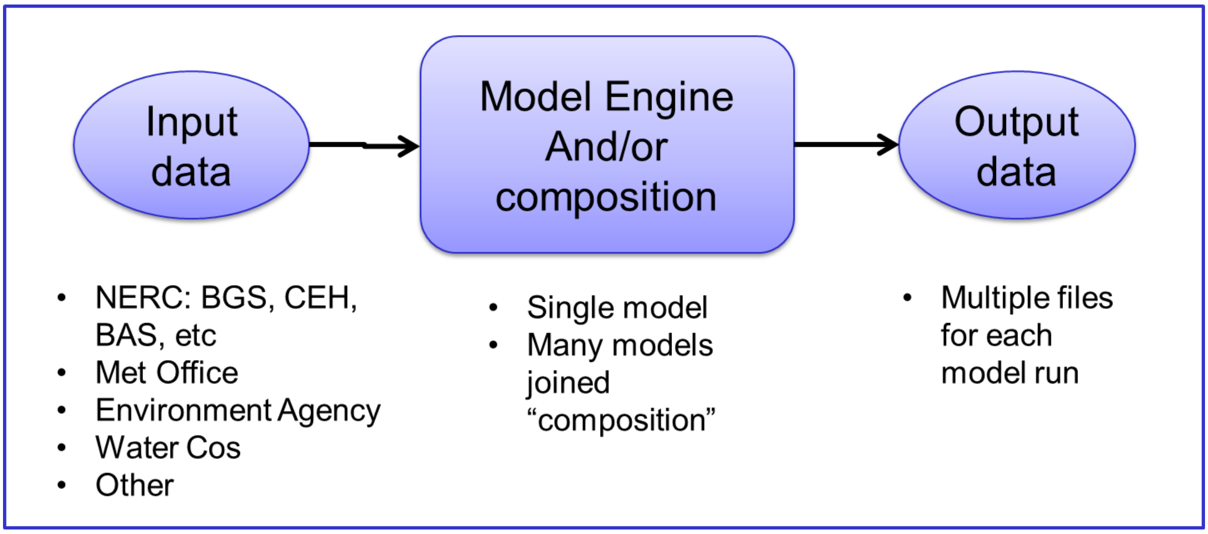


Figure 1. Model as a "unit operation"

The workflow, defined here as the movement of data around and between different models, is important. As in all modelling approaches matching the solution to the question is highly advantageous. The question posed by the stakeholder has to be translated into a modelling system described by a workflow that can address that question.

3.2 MAIN ELEMENTS OF INTEGRATION

The solution of complex questions require holistic solutions drawn from different disciplines (Laniak et al., 2013). Importantly the technology has progressed significantly within the last five years such that models from different disciplines can be linked and integrated assessments of problems can now be made (Kelly et al., 2013). Now it is possible to link models, consideration has to be given to the opportunities but also the science issues that are generated by studying interacting processes. These include:

- Understanding interactions within and across disciplines, for example how best to simulate the interaction between agricultural policy and farming practice (e.g the EU funded SEAMLESS project; Van Ittersum et al., 2008)

- Reconciling the differing functions used by different disciplines when evaluating similar interacting processes (e.g. sewers and river flooding, Van Assel et al., 2010 and sediments transport in rivers, Shrestha et al, 2013)
- Reconciling the different languages that different scientific disciplines use (e.g. soil scientists, ecologists and geologists all have subtly different definitions for what constitutes 'soil'), thereby reducing the opportunity for error in finding and linking models, e.g. Knapen et al. (2013).
- Defining model metadata so that developers can describe and users can find, evaluate and validly link the models needed to resolve their problem. Whilst no internationally recognised standard exists workers such as Harpham and Danovaro (2015) have suggested possible approaches based on existing standards such as ISO19115.
- Encouraging the development of standardisation both in input/output data and the modelling process itself, to make the process of creating a model "auditable". For the latter is the use of ontologies and "controlled vocabularies" to identify similarities in the concepts being modelled is required. Work has been undertaken on metadata standards to describe the variables within each code and to assist the coupling of models (e.g. Peckham et al., 2013).
- Understanding the propagation of uncertainty in model chains, and projects such as UncertWeb have proposed possible solutions (Bastin et al., 2013)
- Providing suitable computational resources to run linked model compositions. Here High Performance Computing holds the key as espoused by the CSDMS project (Peckham et al., 2013)

Finally models need to undergo quality assurance and ensure that they are auditable.

3.3 DIFFERENT APPROACHES

To enable the workflow to be delivered, then a number of different approaches can be adopted. The overall aim is to enable the functionality of the modelling system to be matched to the question under consideration. There are number of different ways of achieving this, but the most commonly used ones are as follows:

1. Hardwired – joining two models together by adding them in the same code; extending existing code (e.g. MODFLOW-SWAT; Sophocleous et al., 1999).
2. Passing data – "flat files" the output from one model is passed to another via asci or binary files.
3. Models as components – converting existing model into linkable components and passing data at runtime (OpenMI; DHRIM example – Harpham et al., 2016).
4. Models as a Service (Maas) – using web processing services to pass data via the internet; Model Web initiative (Nativi et al., 2013).

All have their advantages and disadvantages and, again, depend on the question being posed. All of them require a degree of technical knowledge. For example option (1) requires that the user has knowledge of the code being modified and can produce scientific code.

3.4 STANDARDS REQUIRED FOR INTEGRATION

There are a number of standards required for integration to be undertaken successfully. These include flat file (asci files), metadata standards for data and models and standards to pass data between models as components during runtime. For flat file(s) there are many and various standards for file based data: XML and its various flavours; JSON and NetCDF. There are metadata standards for data – ISO19115, however there isn't currently a model metadata standard, although there is a groundswell of opinion on what is required (e.g. Harpham and Danovaro, 2015).

For models as components then there are a number of standards depending on the discipline area e.g. climate, hydrology, environmental regulation, insurance, human health. Climate models can be linked using a number of different frameworks, such as Bespoke Framework Generator or BFG (Barkwith et al., 2014) and with models from other disciplines (e.g. Goodall et al., 2013). Hydrology has well-developed model linking approaches such as OpenMI, and FluidEarth which provide a toolkit to implement this standard (Harpham et al., 2015). This OpenMI implementation (.NET/in-memory) is complimented by the HPC based CSDMS (Peckham et al, 2013). The US EPA has developed their system, FRAMES_3MRA, to enable rapid assessment of the environmental impact of potentially polluting activities (Whelan et al., 2014). The insurance industry, which relies on Catastrophe models (Royse et al., 2014) to assess risk has spawned its own approach to providing a framework to link models, that of OASIS-LMF (Barkwith et al., 2014). Even a discipline area seemingly far removed from environmental modelling such as modelling the human body to safely test new drugs has developed its integrated modelling approaches (e.g. Virtual Physiological Human; Hunter et al., 2010). Whereas in the military sphere, the US DoD has developed the High Level Architecture (HLA) (Hemingway et al., 2012) which is now in use by NATO for exchanging data during military exercises. The HLA can be used for model linking once the model is made available as a discrete and linkable entity.

Of increasing popularly is the idea of Models as a Service (MaaS) whereby the outputs from any model can be made available as an interrogatable service. This is normally as a Application Processing Interface (API) which can exist in the form of a Web Processing Services which is a protocol for passing data via internet (OGC standard). APIs are generic description of a WPS and can be served over different means, i.e. local network. However once made available via the internet then the APIs can be universally available. Finally APIs are useful where outputs are required from a model and two-way interaction is not required.

3.5 EXAMPLES DEVELOPED AS PART OF WP4 – USE OF APIS

To show the utility of APIs in providing suitable model output a simple example has been developed to draw on both the observation data and model predictions available through the Meterological Office (MO) and Shoothill (Environment Agency data) APIs. The outline workflow is shown in Figure 2. Observed data along with the results of weather forecasts from the two APIs are interpreted by simple rules which determine whether the river state both now and the immediate future is safe for rowing crews of different levels of experience to undertake their training session. The rules for novice and senior crews are provided in Appendix 1 and are based on the stage (height of water) in the river and both windspeed and wind gusts. Senior crews can go out on the river with higher river stage and more blustery wind conditions than the novice crews.

The code, DataFeedRowing is provided in Appendix 2 along with an example of the text output.

The code is written in C# and has two main classes: CurrentState and FutureState. The former returns the observed values of the River Ouse at Foss Barrier. These are the stage for every 15 minutes for the last 3 days. Additionally the observed meteorological values at Topcliffe are sought. The latter class, FutureState returns the flood forecasts for the next three days and the MO forecast for Topcliffe.

Aside from creating the access to the APIs, the string is read by a JSON "de-serialiser" from a standard library (Newtonsoft.Json) and then processed to provide the data required.

The output form the code is a traffic light assessment for the current and future (one hour ahead) state of the river for both novice and senior crews (see Figure 3).

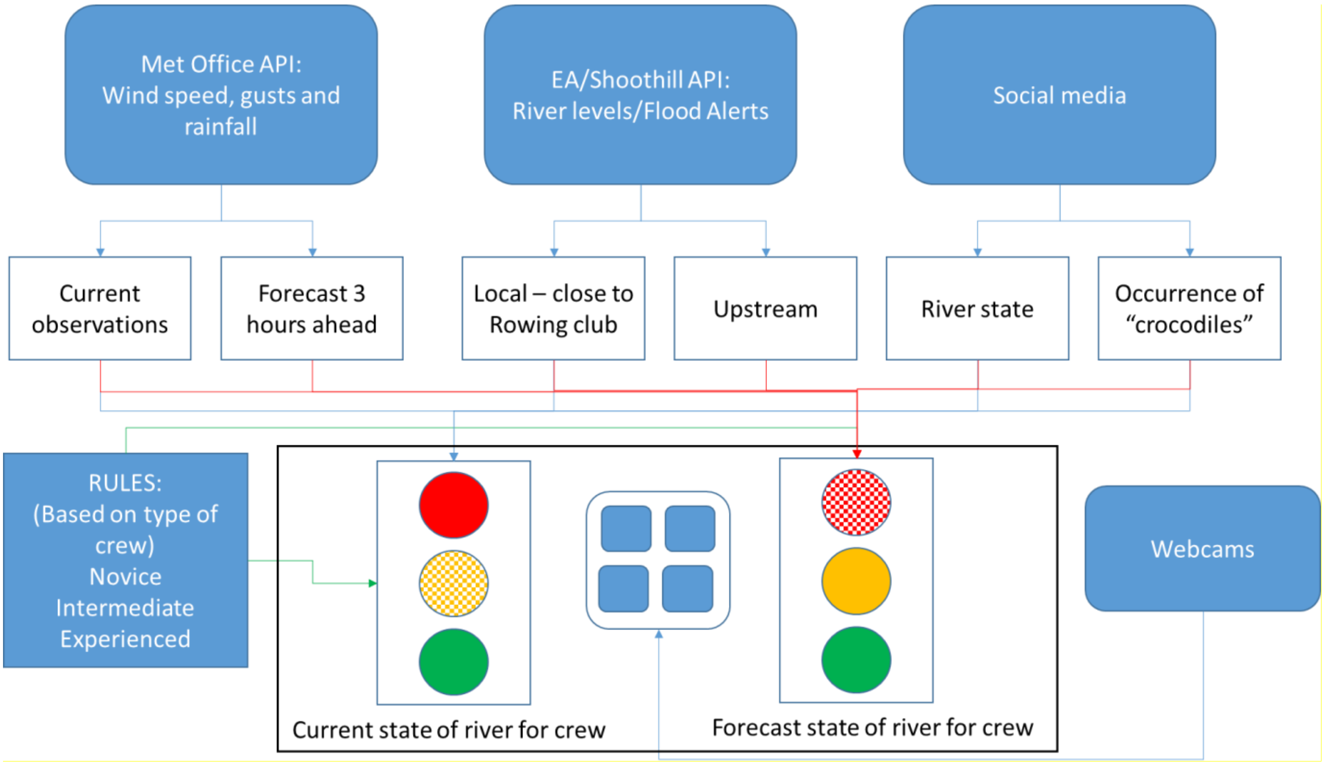


Figure 2. Proposed workflow for rowing use case example.

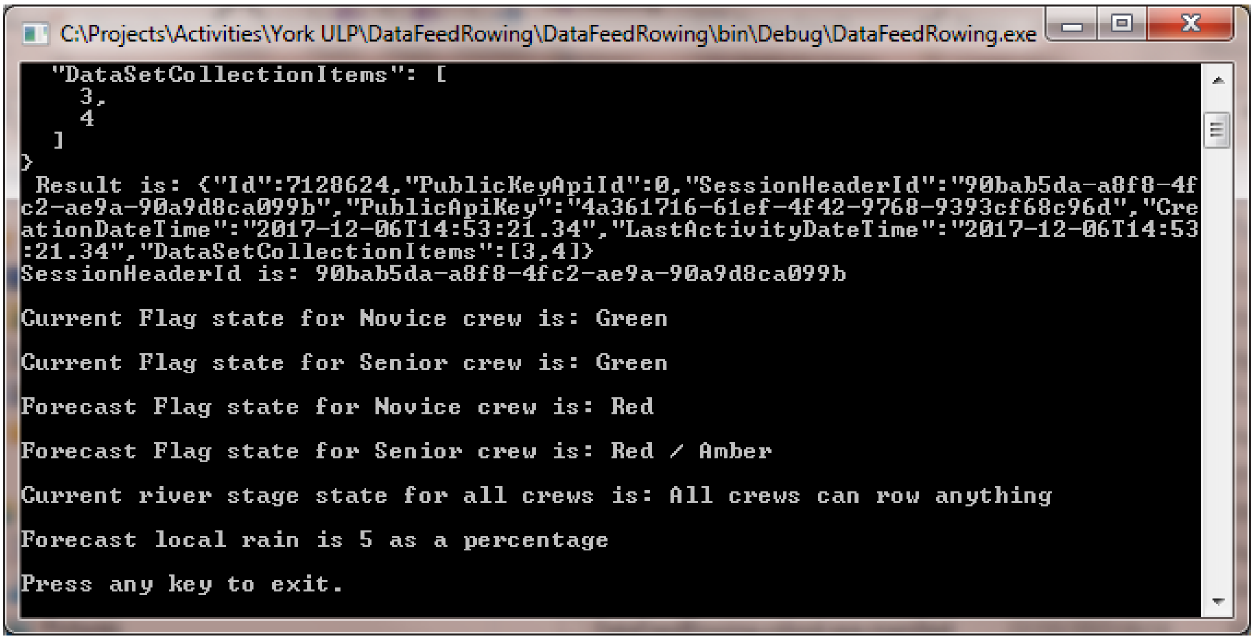


Figure 3. Screenshot showing example of output from the code.

4 Current approaches

Given the complexity of the problem and its likely solution then it is necessary to examine what has been undertaken and to draw on best practice. To that end exemplars from the UK and around the world have been sought. There are, however, a number of different strands which are related, but need to be described.

Smart City "landscape": There are a number of cities which have been at the forefront of the smart cities movement. These include, but are not limited to: Chicago, Boston, Barcelona, Helsinki, Amsterdam, Songdo (Korea) and Stockholm. In the UK, funding has been provided to develop Smart Cities, for example Glasgow recently won £24m from the Future Cities Demonstrator Programme (Innovate UK) competition to develop a Smart City infrastructure.

Smart cities are built on a number of features such as:

- Open Data – availability of data in an open form, i.e. can be readily licenced and used to quantify urban processes
- Data analytics – the analysis of these data to provide an understanding how cities operate
- Predictive analytics – using existing interpretation of data to make predictions of what may happen in the future

As well as this, integrated modelling in the urban environment is regularly practised and many examples exist in the literature. However, the term integrated refers to one or more different disciplines included in the model (i.e. traffic movement and air pollution and its movement) and not generic integration approaches to address questions which require solutions form different disciplines.

There are a number of UK-based and international exemplars which promise the future of integrated modelling or at least provide a pathway to its operation. These are described in more detail below.

4.1 UK

There are a range of different initiatives, the majority of which have not actually produced a working model, even at a pilot phase. For the water sector, the recent Urban Simulators for Water (STFC and The UK Water Partnership) initiative has undertaken a workshop to determine what the challenges are and how a system that allows the user to develop a solution can be produced. They report that the appetite exists for such an initiative, but that significant resources are required to develop an integrated approach for urban simulators for water (STFC / The UK Water Partnership, 2016).

The FuturICT project is an EU funded project to develop a living lab type approach for smart cities ICT. They are discipline agonistic, but express an ambition to develop a system whereby users can integrate their own models. Examples are given of integrated approaches, but these are bespoke solutions for which the question is known beforehand. The project was reported on in 2012, but no internet presence of a resulting integration system can be found.

One of the partners in the FuturICT is CASA – UCL based organisation which specialise in Agent Based Modelling in urban environments. They, together with colleagues within FuturICT, have developed open-source Agent Based Model MATSim. The latter can be used in urban environments to simulate vehicle movements.

Imperial College has developed a model exchange platform called Digital City Exchange. Data and models are treated equally and can be linked. Examples are given for impact of using electric vehicles on the grid. The project was reported on in 2012, but no internet presence of a result can be found.

4.1.1 CitySim – University of Nottingham

This is a project led by University of Nottingham and funded by the Leverhulme trust. Its aim is to model energy use in cities holistically using the CitySim energy demand simulator for buildings. However building energy use is only part of the energy usage within a city and other forms such as the energy for transportation systems are required to be included in any simulation. In order to examine the interactions and feedbacks between different energy uses then linking with other models, e.g. transport is necessary.

Whilst the project is in its early stages the model linkage approach chosen is HLA – DoD model linking approach. This requires that models are converted to "Simulator Entities" which can provide data to the run-time infrastructure or RTI (see Figure 4). The RTI controls the running of each entity and the exchange of data between them at runtime.

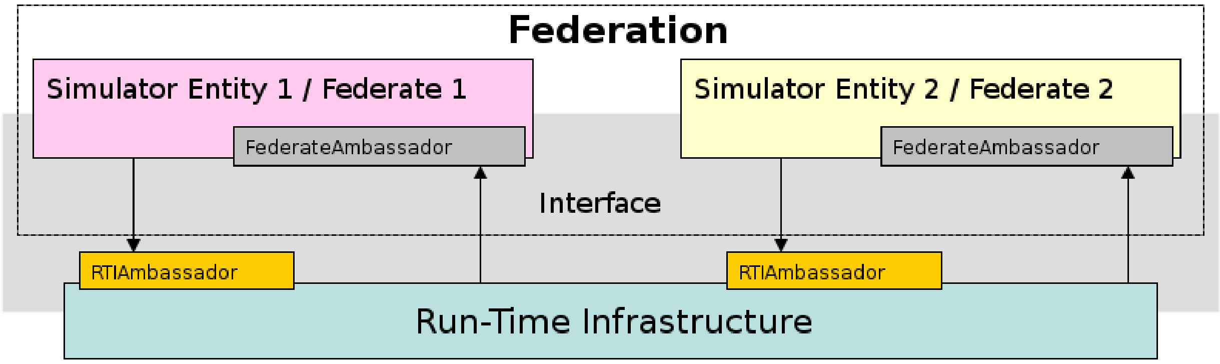


Figure 4. Example of HLA architecture (provided from www.msco.mil used under CC BY SA (creativecommons.org/licenses/by-sa/3.0/))

4.1.2 Tombolo

Tombolo is an Innovate UK funded project of three years duration which started in summer 2015. It is being undertaken by a partnership of the Future Cities Catapult and Space Syntax (planning, transport modellers; www.spacesyntax.com). Its aim is to provide an integrated approach to undertake health based studies in an urban setting and there are three case studies:

1. Leeds – Community Resilience and in particular examining social isolation
2. Milton Keynes – Obesity and how it can be tackled by active transport, e.g. walking
3. Greenwich – Access to services, e.g. GPs, schools, workplace, etc

The project is currently developing an architecture around addressing these use cases by using a workflow-based approach that will enable the user to mix and match their own data with open data sources. Based on JSON, then there are two basic elements:

- Ingredient list
- Recipe

The ingredients define the data sources and the recipe determines how the data will be processed and exported. Figure 5 shows the planned architecture of the system. Users will be able to combine their own data which sits locally with open datasets which can be accessed using pre-written importers. These can be drawn into a local data store and then processed using a recipe and either exported or combined with pre-built models to produce data required.

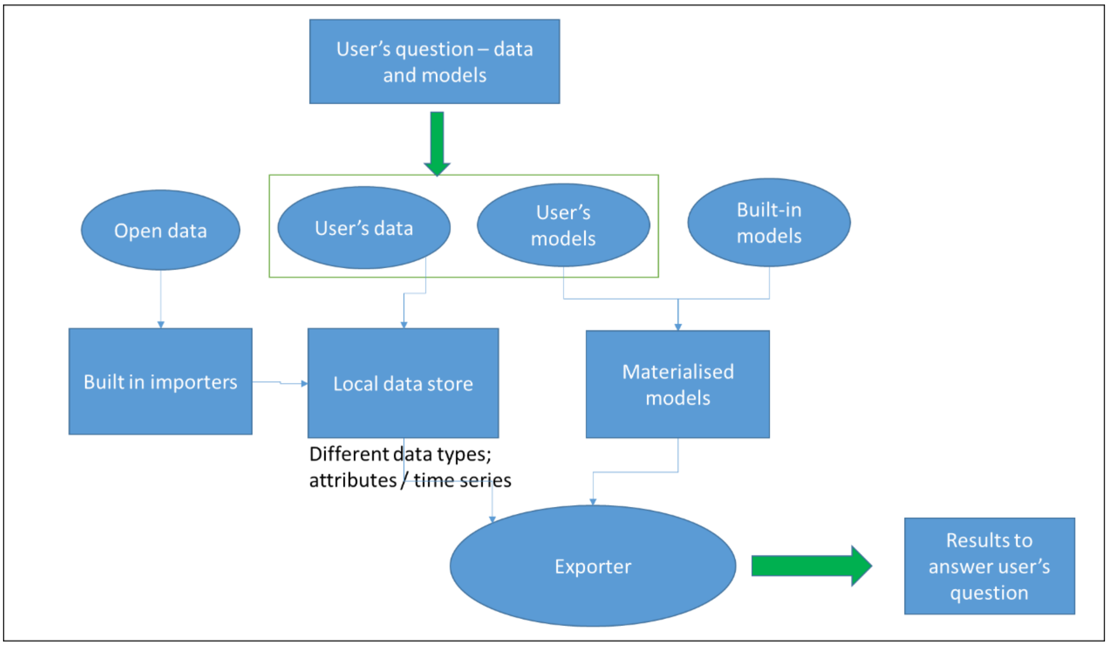


Figure 5. Planned system architecture for Tombolo

Whilst two-way model integration is not currently planned to be incorporated into the planned version of Tombolo, there is scope for the inclusion of APIs using a modified built-in importer. The outputs from the system could be served out using an OpenMI compliant web processing service component (similar to that reported in Castronova et al., 2013). This will enable the Tombolo workflow to be incorporated into an existing composition.

4.2 INTERNATIONAL

Smart Cities are proliferating around the globe. Many cities are genuinely developing their smart credentials, and a significant number are jumping on the bandwagon. The initial thrust of Smart Cities was ICT based. Hardware and software were combined to produce improved ICT systems for managing cities. A realisation has arisen that this is not the only possible way and that the user needs to be taken into account, so-called user-centred design (Saunders & Baeck, 2015). The important question for the purposes of this report is what do they offer as ways of integrating models?

4.2.1 Asia

Asia does have a very strong smart cities movement. There are many and various: South Korea (Songdo), Singapore, Masdar City (UAE) to name but a few. Songdo is an example of a bespoke smart city built from the ground up on reclaimed land 65 km west of Seoul, and Singapore is classed as a smart nation given its status of an island state.

These examples are mainly ICT driven and do not necessarily include modelling of any description. However there is a growing movement for "predictive analytics" which build on data availability, i.e. Open Data movement and availability of code and tools to produce useful predictions. Examples are given below for Chicago and Nairobi.

A couple of interesting views on Smart Cities in Asia, firstly regarding sensors:

"Sensors with sophisticated control systems can work in cities such as Melbourne or London or Sydney, but 90 percent of people in cities don't live in cool, temperate climates -- they live near the equator and there, most of this smart tech does not apply," says Gerhard Schmitt of ETH Zurich. "These cities can be smart, but the innovations need to be affordable and useable." http://www.wired.co.uk/article/smart-city-planning-permission

4.2.2 Chicago city

Chicago City Mayor Emanuel, elected in 2011, made it one of his priorities to make Chicago City's data open and provide it via an open data portal. This proved successful and a number of initiatives have been undertaken to utilise this infrastructure. One example was the development of a script to predict failures in restaurant inspection using predictive analytics– i.e. prediction: ttps://chicago.github.io/food-inspections-evaluation/. The aim was to prioritise visits by health inspectors by ranking fast food outlets in order of likelihood of non-compliance with food hygiene regulations.

The decision was taken to make both the data and the tools developed to undertake the predictions available as Open-source. The choice of the code was R and the finished project (code, data and documents) has been published on GitHub. The end result was a straight R script which produced graphical and tabular output, but no interface was produced. The aim was to make the process as open and transparent as possible as well as encouraging users to build on the approach.

4.2.3 Nairobi city

One of the common methods of transport in Nairobi is a Matatu or minibus taxi which follow fixed routes. The tracking of these vehicles using student volunteers with GPS enabled a route map to be generated (Figure 6). Open source tools was used to serve out the data and a range of apps have been built on the data – including the open source traffic data platform: www.ma3route.com. Whilst modelling as such has not been undertaken on the data it has been used for further analysis. Traffic calming measures using oil drums to close off certain junctions have been assessed using data gleaned by this open approach. The Nairobi example shows how limited resources can be used to build useful outputs.

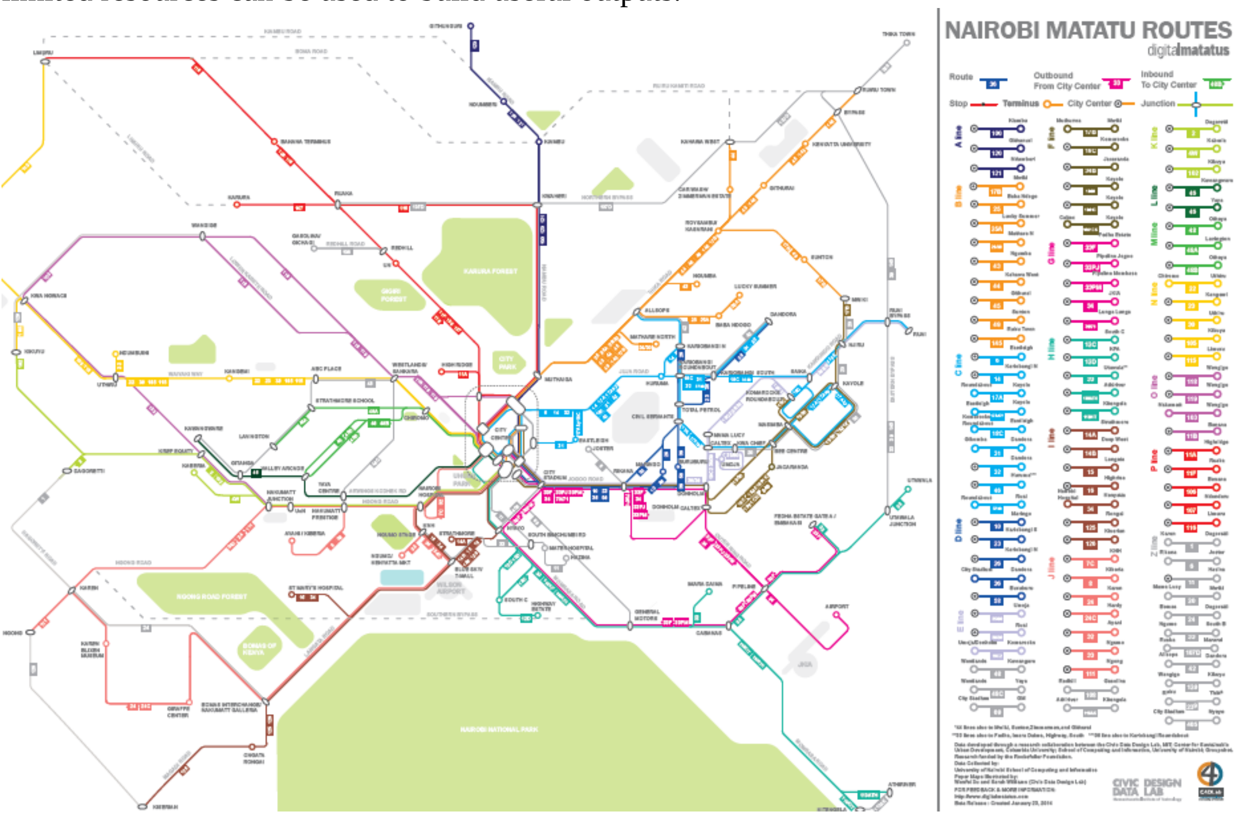


Figure 6. London tube map style Matatu routes, Nairobi (from www.digitalmatatus.com/map.html).

5 Suggested way forward

Model integration is not straight-forward. It is well recognised as a significant problem for trained scientists / modellers to undertake. There is now a global community dedicated to solving different aspects of this subject. The challenge of model integration is exacerbated by the limited availability of models that are properly described, available and linkable. Further once the aim of model integration is extended to “plug and play” systems that can be used by non-technical specialists then the challenge increases markedly.

But there is an appetite and need for problems to be solved in an integrated manner, e.g. recognised in the number of projects that attempt it and workshop reports, e.g. Water in the City / Urban Simulators. These projects are defining the requirements for model integration to solve environmental problems. Collaboration becomes increasingly important as it is a large and significant challenge: of the order of “painting the Forth Bridge”. The solution requires international effort from multiple organisations each building on the results of the others.

Exemplars exist that show how data can be used, one good example is Chicago and the use of predictive analytics to help food inspections undertake their work. This relied on a pre-existing open data platform and then the code is produced in an open and transparent way. Whilst the outcome is fixed, the code is made available and its underlying assumptions are documented.

Another promising approach is semantic reasoning whereby the questions is re-cast into a pre-defined structure. So data and / or models are used to provide outputs which answer these questions and then are “tagged” with the following semantic description: Flood events in York which affect domestic properties in postcode YO23. By re-framing the outputs in this way then the linkages between location, data type, etc can much more easily be searched and connections made.

APIs offer an excellent way of making model results available. They can be exploited at very low computational cost as the model is run somewhere else and the results stored. However, there is limited availability for feedback and interaction, i.e. modifying model results based on the output from another model.

Integrated models that exist as bespoke codes or as compositions (series of linked codes) should be able to be incorporated into any systems. These could possibly answer the questions posed and mean that no further development is required. There should be a library of pre-formed solutions which are made available in a user-friendly way.

To produce a suitable working system for the YCEO there should be three elements to the system:

- Tools and techniques to combine models / APIs / Data to produce bespoke solutions.
- Availability of existing solutions either built using underlying tools or provided by external organisations.
- The solutions wrapped in semantic reasoning.

Next steps:

- Engage with Tombolo project to develop an API based time variant use case, i.e. rowing example.
- Produce a trial version based on the use cases developed within the project.

References

Barkwith, A.K.A.P.; Pachocka, M.; Watson, C.; Hughes, A.G.. 2014 Couplers for linking environmental models: scoping study and potential next steps. Nottingham, UK, British Geological Survey. (OR/14/022) (Unpublished)

Bastin, L., Cornford, D., Jones, R., Heuvelink, G.B., Pebesma, E., Stasch, C., Nativi, S., Mazzetti, P. and Williams, M., 2013. Managing uncertainty in integrated environmental modelling: The UncertWeb framework. *Environmental Modelling & Software*, *39*, pp.116-134.

Castronova, A. M., Goodall, J. L., & Elag, M. M., 2013. Models as web services using the open geospatial consortium (ogc) web processing service (wps) standard. Environmental Modelling & Software, 41, 72-83.

Goodall, J.L., Saint, K.D., Ercan, M.B., Briley, L.J., Murphy, S., You, H., DeLuca, C. and Rood, R.B., 2013. Coupling climate and hydrological models: Interoperability through Web Services. *Environmental modelling & software*, *46*, pp.250-259.

Harpham, Q. and Danovaro, E., 2015. Towards standard metadata to support models and interfaces in a hydro-meteorological model chain. *Journal of Hydroinformatics*, *17*(2), pp.260-274.

Harpham, Q., Lhomme, J., Parodi, A., Fiori, E., Jagers, B., & Galizia, A., 2016. Using OpenMI and a Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy. JAWRA Journal of the American Water Resources Association, 52(4), 933-949.

Hemingway, G., Neema, H., Nine, H., Sztipanovits, J., & Karsai, G., 2012. Rapid synthesis of high-level architecture-based heterogeneous simulation: a model-based integration approach. Simulation, 88(2), 217-232.

Hunter, P., Coveney, P.V., de Bono, B., Diaz, V., Fenner, J., Frangi, A.F., Harris, P., Hose, R., Kohl, P., Lawford, P. and McCormack, K., 2010. A vision and strategy for the virtual physiological human in 2010 and beyond. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 368(1920), pp.2595-2614.

Kelly, R.A., Jakeman, A.J., Barreteau, O., Borsuk, M.E., ElSawah, S., Hamilton, S.H., Henriksen, H.J., Kuikka, S., Maier, H.R., Rizzoli, A.E. and van Delden, H., 2013. Selecting among five common modelling approaches for integrated environmental assessment and management. Environmental modelling & software, 47, pp.159-181.

Knapen, R., Janssen, S., Roosenschoon, O., Verweij, P., De Winter, W., Uiterwijk, M. and Wien, J.E., 2013. Evaluating OpenMI as a model integration platform across disciplines. *Environmental modelling & software*, *39*, pp.274-282

Laniak, G.F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M. and Peckham, S., 2013. Integrated environmental modeling: a vision and roadmap for the future. *Environmental Modelling & Software*, *39*, pp.3-23.

Nativi, S., Mazzetti, P., & Geller, G. N., 2013. Environmental model access and interoperability: The GEO Model Web initiative. Environmental Modelling & Software, 39, 214-228.

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

Saunders T & Baeck p, 2015. Rethinking Smart Cities from the ground up. NESTA.

Shrestha, N.K., Leta, O.T., De Fraine, B., Van Griensven, A. and Bauwens, W., 2013. OpenMI-based integrated sediment transport modelling of the river Zenne, Belgium. *Environmental modelling & software*, *47*, pp.193-206.

Sophocleous, M. A., Koelliker, J. K., Govindaraju, R. S., Birdie, T., Ramireddygari, S. R., & Perkins, S. P., 1999. Integrated numerical modeling for basin-wide water management: The case of the Rattlesnake Creek basin in south-central Kansas. Journal of Hydrology, 214(1), 179-196.

STFC / The UK Water Partnership, 2016. “Urban Simulators for Water Workshop Report – March 2016”.

Van Assel, J., Waterschoot, G., Devroede, N., Ronse, Y., Anderson, S. and Millington, R., 2010. Modelling bidirectional interactions between sewer and river systems using OpenMI-a case study in the Scheldt River Basin (Belgium). *NOVATECH 2010*.

Van Ittersum, M.K., Ewert, F., Heckelei, T., Wery, J., Olsson, J.A., Andersen, E., Bezlepkina, I., Brouwer, F., Donatelli, M., Flichman, G. and Olsson, L., 2008. Integrated assessment of agricultural systems–A component-based framework for the European Union (SEAMLESS). *Agricultural systems*, *96*(1), pp.150-165

Whelan, G., Kim, K., Pelton, M.A., Castleton, K.J., Laniak, G.F., Wolfe, K., Parmar, R., Babendreier, J. and Galvin, M., 2014. Design of a component-based integrated environmental modeling framework. *Environmental Modelling & Software*, *55*, pp.1-24.

Appendix 1 Specification for rowing use case

WORKING OUT WHEN TO ROW:

Weather data, rower experience and competency, river data such as busyness, risks such as debris, other crews...

Websites:

River levels and flooding:

Gov.uk flood for Viking Recorder (the old EA website was much better-updated every 3 hours whereas this one only updates every 12 hours if lucky, usually every 24 hours is published, but more regularly in flood times): https://flood-warning-information.service.gov.uk/station/8208

River Ouse Twitter to monitor it when it is rising fast-twitter repeats the Viking Recorder data but can be easier to access

Gauge Map: River Ouse, Viking Recorder and the Dales so you can see which feed into the Ouse: http://www.gaugemap.co.uk/#!Detail/2060

Wind speed and weather:

Uni electronics website, BBC weather (wind, lightning)

Anemometer at the boat house (BH)

Risk assessments usually done by eye at the BH:

-wind direction strongly pushing the boat into the steps when attempting to boat would mean they need assistance, or to not go out.

-debris in the river-flood water full of logs or branches, or particularly large logs etc.

-no rowing in lightning storms- weather websites are checked, but if you see lightning you must wait half an hour for it to pass before boating, or is already on the river, you must land at the nearest available point and wait half an hour.

-events on the water, crews running side by side racing etc, general busyness of the river-eg bank holiday Naburn cruiser procession was a nightmare for any smaller boats, the novice women's 4 in a boat with low sides struggled and had to stop rowing

Rower competency is usually classed in terms of:

-experience (number of outings-the university classes a novice as someone with less than a year of rowing 3-4 sessions a week at least, and/or people without significant race experience, a senior has at least a year of experience, but also within that there are highly experienced seniors/veterans -on some rivers only a senior cox ie a cox who has competed to a high level or coxed safely for years will be allowed to take a crew out in red flag conditions),

-safety talk attendance and safety capsize drill in the swimming pool (capsize drill teaches procedure and how to get back in the boat, or buddy rescue, rowers not allowed on the water past term 1 without these as some sessions may be unaccompanied by coach),

-coxed or uncoxed boat (cox will usually have better chance of seeing debris, dealing with bad conditions as not trying to row at the same time),

-gender (mostly relating to strength, although experience is also a factor-eg a senior men's 8 will be stronger and better able to cope with bad conditions relating to speed of current than a senior women's 8, however a senior women's 8 will boat in worse conditions than a novice men's 8),

-captain/coach's assessment of capability-novice rowers initially mixed in with senior rowers and coxed by senior coxes for first sessions, and accompanied by launch-only when they reach the coach's idea of basic competency can they row in totally novice crews, likewise novice rowers or coxes steering for their first sessions are accompanied by launch until captain/coach declares their capability-assessments like steering through bridges, avoiding other crews, commanding their own crew safely-eg recently a novice men's 2x rowed into the bank-it was later discovered that the bowman has a slight visual impairment-thus should have not been permitted to steer.

-assessment of the route-eg rower scared to go in a single scull would start on a straight course, close to the bank (and rescue).

Equipment

-boat category-smaller boats are less stable, and slower (single, double scull, coxed 4 compared to coxed 8), however a quadruple scull and an 8 are nearly matched in stability and speed (a novice men's 8 likely to cope better in windy conditions than a novice men's double scull).

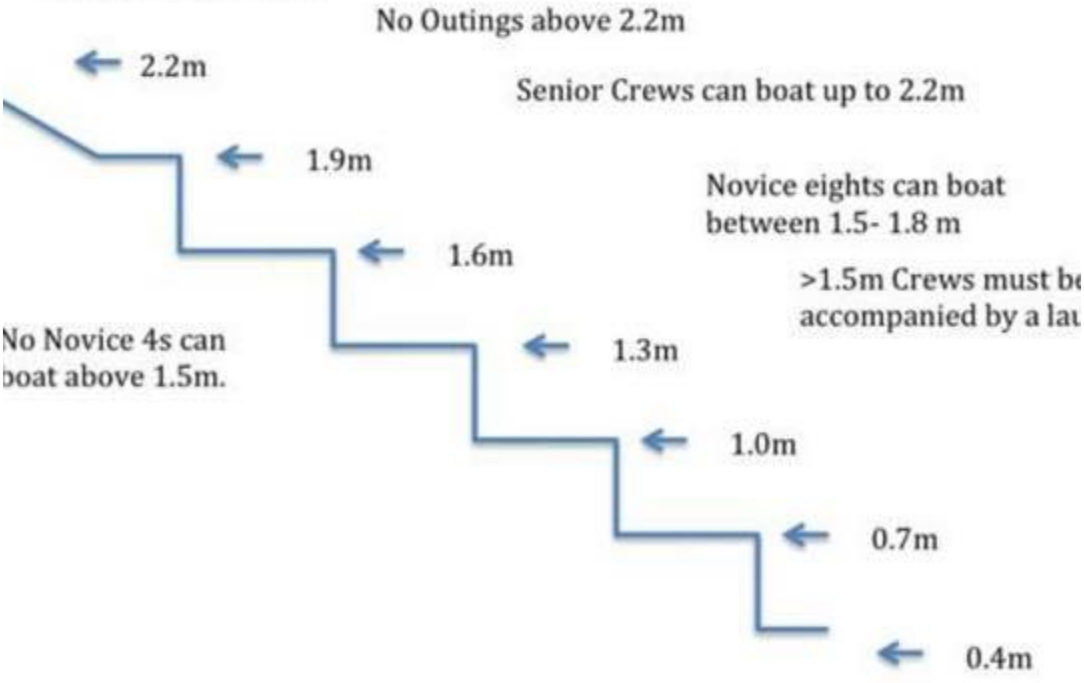
-age, maintenance and buoyancy of equipment. Since the high profile death of an OUBC rower boat manufacturers now ensure that the boat has enough buoyancy underneath the rowers to remain afloat, however, boats made before this time do not have internal buoyancy and are subject to swamping in bad conditions (or busy river times with large cruisers, jet skis etc), thus this is part of the risk assessment process-number of buoyancy bags, river or weather conditions relating to specific equipment,

-height of the sides of the boat, whether crew is sitting high or low in the water-eg Jany 4+ has low sides, needs a lightweight women's crew, but even then in windy conditions the boat is more likely to get water in it; taping the riggers can alleviate some of the splash coming up into the boat-usually a method used when racing on the tideway.

Steps

club in line with other river users and enable crews to take full advantage of wate
time.
The heights used in this draft have been calculated from known heights however they are not accurately measured and cannot be treated as such.
This draft aims to be implemented as policy on a trial basis for the remaining Spri term 2014 in order to gauge (more) accurate river heights and appropriate advice regarding crew competency.

All River levels to be taken from the Viking recorder and crews must consul the UYBC Safety Flag before boating. Although crews may be able to boat a these heights they must take the stream and other factors into consideratio before boating. Each crew must therefore consider individual risk assessments so that visibility, general competency, stream and debris are taken into account.



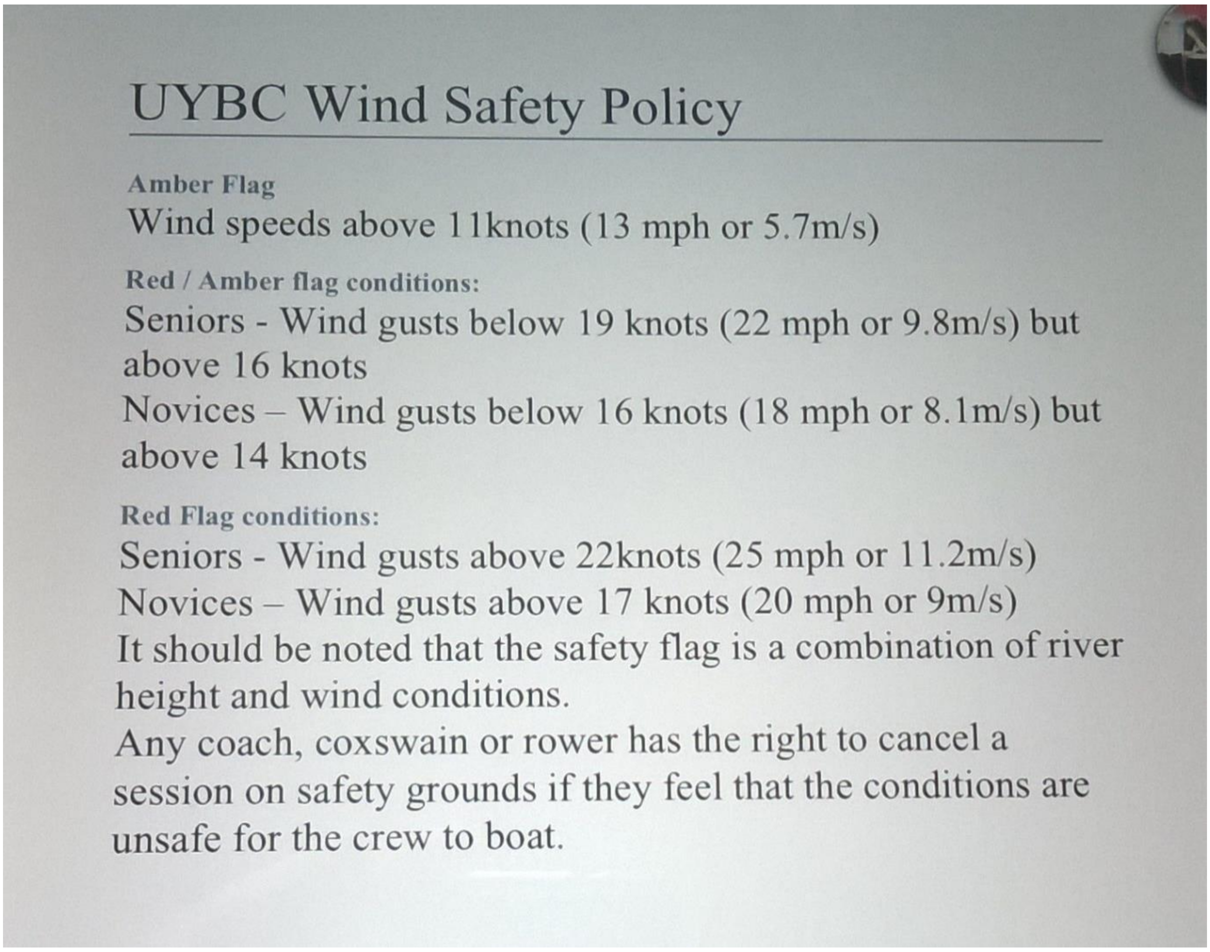
Novices are defined as rowers that have less than one university year's experien

Eights can be launched up to around 1.7/8m (an exact measurement needs to be found).

No novice crews can go out above the 1.9m mark (top step).
Senior crews with senior coxes can go out accompanied up to 2.2m mark.

Above 1.5 all crews must be accompanied by a safety launch. As crews have to b accompanied over 1.5m there will always be a coach with the crew who can mak more impartial call about conditions. A launch in flood must have two people operating it, and one launch may be responsible for multiple crews providing they stay together.

Wind


UYBC Wind Safety Policy
Amber Flag
Wind speeds above 11knots (13 mph or 5.7m/s)
Red / Amber flag conditions:
Seniors - Wind gusts below 19 knots (22 mph or 9.8m/s) but above 16 knots
Novices – Wind gusts below 16 knots (18 mph or 8.1m/s) but above 14 knots
Red Flag conditions:
Seniors - Wind gusts above 22knots (25 mph or 11.2m/s)
Novices – Wind gusts above 17 knots (20 mph or 9m/s)
It should be noted that the safety flag is a combination of river height and wind conditions.
Any coach, coxswain or rower has the right to cancel a session on safety grounds if they feel that the conditions are unsafe for the crew to boat.

Appendix 2 C# code for simple API call example

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace DataFeedRowing
{
    public class RiverState {

        public float currentWindspeed;
        public float currentWindgust;
        public float currentRiverstage;
        public float forecastWindspeed;
        public float forecastWindgust;
        public float forecastRainfallLocal;
        public float forecastRainfallUpstream;


        public RiverState()
        {
            currentWindspeed = 0;
            currentWindgust = 0;
            currentRiverstage = 0;
            forecastWindspeed = 0;
            forecastWindgust = 0;
            forecastRainfallLocal = 0;
            forecastRainfallUpstream = 0;
        }
        public string InitializeShoothillAPI()
        {
```

```
            using (WebClient wc = new WebClient())
            {
                wc.Headers[HttpRequestHeader.ContentType] = "application/json";
                string uri = "http://riverlevelsapi.shoothill.com/ApiAccount/ApiLogin";
                string data = "{'PublicApiKey':SHOOTHILL API KEY, 'ApiVersion':'2'}";
                string result = wc.UploadString(uri, data);
                dynamic authorizedSession = JsonConvert.DeserializeObject<dynamic>(result);
                Console.WriteLine(authorizedSession);
                Console.WriteLine(" Result is: " + result);
                return authorizedSession.SessionHeaderId;
            }

        }
        public string RowingFlagStateCurrent(String crewType)
        {
            string flagState = "Green";
            if(currentWindspeed > 13) {
                flagState = "Amber";
                }
            if (crewType == "Senior" && currentWindgust > 18 && currentWindgust < 22)
            {
                flagState = "Red / Amber";
            }
            if (crewType == "Novice" && currentWindgust > 15 && currentWindgust < 18)
            {
                flagState = "Red / Amber";
            }
            if (crewType == "Senior" && currentWindgust >= 22 )
            {
                flagState = "Red";
            }
            if (crewType == "Novice" && currentWindgust >= 18 )
            {
                flagState = "Red";
            }
            return flagState;
        }
        public string RowingFlagStateFuture(String crewType)
        {
            string flagState = "Green";
            if (currentWindspeed > 13)
```

```
            {
                flagState = "Amber";
            }
            if (crewType == "Senior" && forecastWindgust > 18 && forecastWindgust < 22)
            {
                flagState = "Red / Amber";
            }
            if (crewType == "Novice" && forecastWindgust > 15 && forecastWindgust < 18)
            {
                flagState = "Red / Amber";
            }
            if (crewType == "Senior" && forecastWindgust >= 22)
            {
                flagState = "Red";
            }
            if (crewType == "Novice" && forecastWindgust >= 18)
            {
                flagState = "Red";
            }
            return flagState;
        }
        public string RowingDecisionCurrent()
        {
            string outingState = "No outings as default";
            if (currentRiverstage < 1.5){
                outingState = "All crews can row anything";
            }else if (currentRiverstage >= 1.5 && currentRiverstage < 1.8){
                outingState = "no novice 4s - all with safety launch";
            }else if(currentRiverstage >= 1.8 && currentRiverstage < 2.2){
                outingState = "no novices at all - seniors with safety launch";
            }else if(currentRiverstage >= 2.2){
                outingState = "No outings for all crews";
            }


            return outingState;
        }

    }

class Program
```

```
    {

        static void Main(string[] args)
        {
            // Instantiate class
            RiverState test = new RiverState();
            String directoryOutput = "C:/Projects/Activities/York ULP/DataFeedRowing/test.txt";
            // String directoryOutput = "test.txt";
            // Setup Shoothill API
            Console.WriteLine("Getting data from Shoothill API"+"\n");
            string SessionHeaderId = string.Empty;
            SessionHeaderId = test.InitializeShoothillAPI();
            Console.WriteLine("SessionHeaderId is: " + SessionHeaderId + "\n");
            CurrentState RiverCurrent = new CurrentState(SessionHeaderId, directoryOutput);
            System.IO.File.AppendAllText(directoryOutput, "CS - SessionHeaderId is: " + RiverCurrent.SessionHeaderId +
"\n");
            RiverCurrent.ReturnData();

            test.currentWindspeed = RiverCurrent.ReturnWindspeed();
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Current Windspeed (In main) is: " + test.currentWindspeed
+ "\n");
            test.currentWindgust = RiverCurrent.ReturnWindgust();
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Current Windgust (In main) is: " + test.currentWindgust +
"\n");
            test.currentRiverstage = RiverCurrent.ReturnRiverStage();
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Current Riverstage (In main) is: " +
test.currentRiverstage + "\n");

            FutureState WeatherFuture = new FutureState(SessionHeaderId, directoryOutput);
            System.IO.File.AppendAllText(directoryOutput, "\n" + "FS - SessionHeaderId is: " +
WeatherFuture.SessionHeaderId + "\n");
            WeatherFuture.ReturnData();
            test.forecastWindspeed = WeatherFuture.ReturnWindspeed();
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Forecast Windspeed (In main) is: " +
test.forecastWindspeed + "\n");
            test.forecastWindgust = WeatherFuture.ReturnWindgust();
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Forecast Windgust(In main) is: " + test.forecastWindgust
+ "\n");
            test.forecastRainfallLocal = WeatherFuture.ReturnPrecipitationLocal();
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Forecast Chance of ppt Topcliffe (In main) is:" +
test.forecastRainfallLocal + "\n");
```

```
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Current Flag state for Novice crew is: " +
test.RowingFlagStateCurrent("Novice") + "\n");
            Console.WriteLine("Current Flag state for Novice crew is: " + test.RowingFlagStateCurrent("Novice") + "\n");
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Current Flag state for Senior crew is: " +
test.RowingFlagStateCurrent("Senior") + "\n");
            Console.WriteLine("Current Flag state for Senior crew is: " + test.RowingFlagStateCurrent("Senior") + "\n");
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Forecast Flag state for Novice crew is: " +
test.RowingFlagStateFuture("Novice") + "\n");
            Console.WriteLine("Forecast Flag state for Novice crew is: " + test.RowingFlagStateFuture("Novice") + "\n");
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Forecast Flag state for Senior crew is: " +
test.RowingFlagStateFuture("Senior") + "\n");
            Console.WriteLine("Forecast Flag state for Senior crew is: " + test.RowingFlagStateFuture("Senior") + "\n");
            System.IO.File.AppendAllText(directoryOutput, "\n" + "Current river stage state for all crews is: " +
test.RowingDecisionCurrent() + "\n");
            Console.WriteLine("Current river stage state for all crews is: " + test.RowingDecisionCurrent() + "\n");
            Console.WriteLine("Forecast local rain is " + test.forecastRainfallLocal + " as a percentage" + "\n");

            // Keep console window open in debug mode
            Console.WriteLine("Press any key to exit.");
            Console.ReadKey();

        }


    }
}
```

```
CurrentState.cs
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace DataFeedRowing
```

```
{
    public class CurrentState
    {
        public string SessionHeaderId = string.Empty;
        private string directoryOutput = string.Empty;
        private float Windspeed = 0;
        private float Windgust = 0;
        private float Riverstage = 0;
        public CurrentState(string SessionHeaderIdOverall, string directroyOutputOverall)
        {
            SessionHeaderId = SessionHeaderIdOverall;
            directoryOutput = directroyOutputOverall;
            using (WebClient wcSH = new WebClient())
            {
                wcSH.Headers["SessionHeaderId"] = SessionHeaderId;
                string result =
                wcSH.DownloadString("http://riverlevelsapi.shoothill.com/TimeSeries/GetTimeSeriesStations?dataTypes=3");
                //Each TimeSeriesStation object in the result contains a list of its gauges.
                dynamic RiverGaugeIds = JsonConvert.DeserializeObject<dynamic>(result);

                System.IO.File.WriteAllText(directoryOutput, " Number of River Gauges is: " +
Convert.ToString(RiverGaugeIds.Count) + "\n");

            }
        }

        public void ReturnData()
        {
            using (WebClient wcSH = new WebClient())
            {
                wcSH.Headers["SessionHeaderId"] = SessionHeaderId;

                string GaugeData =

wcSH.DownloadString("http://riverlevelsapi.shoothill.com/TimeSeries/GetTimeSeriesRecentDatapoints/?stationId=2059&datatyp
e=3&numberDays=5");
                dynamic RiverLevelsFossBarrier = JsonConvert.DeserializeObject<dynamic>(GaugeData);
                System.IO.File.AppendAllText(directoryOutput, Convert.ToString(RiverLevelsFossBarrier) + "\n");
                System.IO.File.AppendAllText(directoryOutput, " River gauge id:" +
Convert.ToString(RiverLevelsFossBarrier.gauge.id) + "\n");
```

```
                System.IO.File.AppendAllText(directoryOutput, " Q95:" +
Convert.ToString(RiverLevelsFossBarrier.gauge.additionalDataObject.percentile95) + "\n");
                System.IO.File.AppendAllText(directoryOutput, " Q5:" +
Convert.ToString(RiverLevelsFossBarrier.gauge.additionalDataObject.percentile5) + "\n");
                System.IO.File.AppendAllText(directoryOutput, " No. values " +
Convert.ToString(RiverLevelsFossBarrier.values.Count) + "\n");

                for (int i = 0; i < RiverLevelsFossBarrier.values.Count; i++)
                {
                    System.IO.File.AppendAllText(directoryOutput, Convert.ToString(RiverLevelsFossBarrier.values[i].time)
+ "," + Convert.ToString(RiverLevelsFossBarrier.values[i].value) + "\n");
                }

                GaugeData =

wcSH.DownloadString("http://riverlevelsapi.shoothill.com/TimeSeries/GetTimeSeriesRecentDatapoints/?stationId=2060&datatyp
e=3&numberDays=5");
                dynamic RiverLevelsViking = JsonConvert.DeserializeObject<dynamic>(GaugeData);
                System.IO.File.AppendAllText(directoryOutput, Convert.ToString(RiverLevelsViking) + "\n");
                // Sample MO web call
                using (WebClient wcMO = new WebClient())
                {
                    wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                    string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxobs/all/json/3265?res=hourly&MO API
KEYMO API KEY;
                    string fileName = "3265.json";
                    wcMO.DownloadFile(uri, fileName);
                    String resultMOJson = wcMO.DownloadString(uri);
                    String uri2 = "http://datapoint.metoffice.gov.uk/public/data/val/wxobs/all/xml/3265?res=hourly&MO API
KEY;
                    String resultMOxml = wcMO.DownloadString(uri2);
                    System.IO.File.AppendAllText(directoryOutput, "Observed at Topcliffe:" + resultMOxml + "\n");
                    dynamic TopCliffeObservations = JsonConvert.DeserializeObject<dynamic>(resultMOJson);
                    System.IO.File.AppendAllText(directoryOutput, "Observed at Topcliffe JSON:" +
Convert.ToString(TopCliffeObservations.SiteRep.DV.Location.Period) + "\n");
                    System.IO.File.AppendAllText(directoryOutput, "Last observed at Topcliffe JSON: Period: " +
Convert.ToString(TopCliffeObservations.SiteRep.DV.Location.Period.Count) + " Day: " +
Convert.ToString(TopCliffeObservations.SiteRep.DV.Location.Period[1].Rep.Count) + "\n");

                }
```

```
            }
        }

        public float ReturnWindspeed()
        {
            using (WebClient wcMO = new WebClient())
            {
                wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxobs/all/json/3265?res=hourly&MO API 
KEY;
                String resultJSON = wcMO.DownloadString(uri);
                dynamic TopCliffeObserved = JsonConvert.DeserializeObject<dynamic>(resultJSON);
                int valueDay = TopCliffeObserved.SiteRep.DV.Location.Period.Count;
                int valueHour = TopCliffeObserved.SiteRep.DV.Location.Period[valueDay-1].Rep.Count;
                Windspeed = TopCliffeObserved.SiteRep.DV.Location.Period[valueDay-1].Rep[valueHour-1].S;
                System.IO.File.AppendAllText(directoryOutput, "\n" + " Observation at Topcliffe JSON:-  Windspeed is: " + 
Windspeed +"\n");
                return Windspeed;
            }
        }
        public float ReturnWindgust()
        {
            using (WebClient wcMO = new WebClient())
            {
                wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxobs/all/json/3265?res=hourly&MO API 
KEY;
                String resultJSON = wcMO.DownloadString(uri);
                dynamic TopCliffeObserved = JsonConvert.DeserializeObject<dynamic>(resultJSON);
                int valueDay = TopCliffeObserved.SiteRep.DV.Location.Period.Count;
                int valueHour = TopCliffeObserved.SiteRep.DV.Location.Period[valueDay-1].Rep.Count;
                if(TopCliffeObserved.SiteRep.DV.Location.Period[0].Rep[0].G != null)
                {
                    Windgust = TopCliffeObserved.SiteRep.DV.Location.Period[0].Rep[0].G;
                }

                System.IO.File.AppendAllText(directoryOutput, "\n" + " Observation at Topcliffe JSON:-  Windgust is: " + 
Windgust + "\n");
                return Windgust;
            }
        }
```

```
        public float ReturnRiverStage()
        {
            using (WebClient wcSH = new WebClient())
            {
                wcSH.Headers["SessionHeaderId"] = SessionHeaderId;

                string GaugeData =

wcSH.DownloadString("http://riverlevelsapi.shoothill.com/TimeSeries/GetTimeSeriesRecentDatapoints/?stationId=2060&datatyp
e=3&numberDays=5");
                dynamic RiverLevelsViking = JsonConvert.DeserializeObject<dynamic>(GaugeData);
                Riverstage = RiverLevelsViking.values[RiverLevelsViking.values.Count-1].value;
                System.IO.File.AppendAllText(directoryOutput, "\n" + " Observation at Viking Recorder JSON:-  Riverstage
is: " + Riverstage + "\n");
                return Riverstage;
            }
        }

            ~ CurrentState()
        {
            Console.WriteLine(" Removing object from memory");
        }
    }

}
```

FutureState.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace DataFeedRowing
```

```
{
    class FutureState
    {
        public string SessionHeaderId = string.Empty;
        private string directoryOutput = string.Empty;
        private float Windspeed = 0;
        private float Windgust = 0;
        private float ChanceOfPpt = 0;

        public FutureState(string SessionHeaderIdOverall, string directoryOutputOverall)
        {

            SessionHeaderId = SessionHeaderIdOverall;
            directoryOutput = directoryOutputOverall;
        }

        public void ReturnData()
        {
            using (WebClient wcSH = new WebClient())
            {
                wcSH.Headers["SessionHeaderId"] = SessionHeaderId;
                string result =

wcSH.DownloadString("http://riverlevelsapi.shoothill.com/ThreeDayFloodForecast/GetCurrentForecastByRegions?regions=Midlan
ds&region=Northwest");
                System.IO.File.AppendAllText(directoryOutput, " Three Day Forecast :" + result + "\n");
                dynamic OBHIds = JsonConvert.DeserializeObject<dynamic>(result);

                // Sample MO web call
                using (WebClient wcMO = new WebClient())
                {
                    wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                    string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/json/3265?res=3hourly&MO
API KEY;
                    string fileName = "3265.json";
                    wcMO.DownloadFile(uri, fileName);
                    String resultJSON = wcMO.DownloadString(uri);
                    String uri2 = "http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/xml/3265?res=3hourly&MO
API KEY;
                    String resultMO = wcMO.DownloadString(uri2);
                    System.IO.File.AppendAllText(directoryOutput, " Forecast at Topcliffe XML: " + resultMO + "\n");
```

```
                    dynamic TopCliffeForecast = JsonConvert.DeserializeObject<dynamic>(resultJSON);
                    System.IO.File.AppendAllText(directoryOutput, "\n" + " Forecast at Topcliffe JSON: " + "\n" +  "Full
forecast: " + TopCliffeForecast.SiteRep.DV.Location + "\n");
                    System.IO.File.AppendAllText(directoryOutput, "\n" + " Forecast at Topcliffe JSON: " + "\n" + "
Windspeed: " + TopCliffeForecast.SiteRep.DV.Location.Period[0].Rep[0].S + "\n" + "Wind direction: " +
TopCliffeForecast.SiteRep.DV.Location.Period[0].Rep[0].D);

                }

            }
        }
        public float ReturnWindspeed()
        {
            using (WebClient wcMO = new WebClient())
            {
                wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/json/3265?res=3hourly&MO API
KEY;
                String resultJSON = wcMO.DownloadString(uri);
                dynamic TopCliffeForecast = JsonConvert.DeserializeObject<dynamic>(resultJSON);
                Windspeed = TopCliffeForecast.SiteRep.DV.Location.Period[0].Rep[0].S;
                System.IO.File.AppendAllText(directoryOutput, "\n" + " Forecast at Topcliffe JSON:-  Windspeed is: " +
Windspeed);
                return Windspeed;
            }
        }
        public float ReturnWindgust()
        {
            using (WebClient wcMO = new WebClient())
            {
                wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/json/3265?res=3hourly&MO API
KEY;
                String resultJSON = wcMO.DownloadString(uri);
                dynamic TopCliffeForecast = JsonConvert.DeserializeObject<dynamic>(resultJSON);
                Windgust = TopCliffeForecast.SiteRep.DV.Location.Period[0].Rep[0].G;
                System.IO.File.AppendAllText(directoryOutput, "\n" + " Forecast at Topcliffe JSON:-  Windgust is: " +
Windgust);
                return Windgust;
            }
        }
```

```
        public float ReturnPrecipitationLocal()
        {
            using (WebClient wcMO = new WebClient())
            {
                wcMO.Headers[HttpRequestHeader.ContentType] = "application/json";
                string uri = "http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/json/3265?res=3hourly&MO API
KEY;
                String resultJSON = wcMO.DownloadString(uri);
                dynamic TopCliffeForecast = JsonConvert.DeserializeObject<dynamic>(resultJSON);
                ChanceOfPpt = TopCliffeForecast.SiteRep.DV.Location.Period[0].Rep[0].Pp;
                System.IO.File.AppendAllText(directoryOutput, "\n" + " Forecast at Topcliffe JSON:-  Chance of rain is: "
+ ChanceOfPpt);
                return ChanceOfPpt;
            }
        }


    }
}
```

Appendix 3 Example output for simple API call example


```
Observed at Topcliffe:<?xml version="1.0" encoding="UTF-8"?><SiteRep><Wx><Param name="G" units="mph">Wind
Gust</Param><Param name="T" units="C">Temperature</Param><Param name="V" units="m">Visibility</Param><Param name="D"
units="compass">Wind Direction</Param><Param name="S" units="mph">Wind Speed</Param><Param name="W" units="">Weather
Type</Param><Param name="P" units="hpa">Pressure</Param><Param name="Pt" units="Pa/s">Pressure Tendency</Param><Param
name="Dp" units="C">Dew Point</Param><Param name="H" units="%">Screen Relative Humidity</Param></Wx><DV dataDate="2017-12-
06T14:00:00Z" type="Obs"><Location i="3265" lat="54.204" lon="-1.39" name="TOPCLIFFE" country="ENGLAND" continent="EUROPE"
elevation="25.0"><Period type="Day" value="2017-12-05Z"><Rep D="WSW" H="76.1" P="1030" S="7" T="8.3" V="50000" W="8" Pt="F"
Dp="4.4">840</Rep><Rep D="W" H="73.4" P="1030" S="9" T="8.3" V="50000" W="7" Pt="F" Dp="3.9">900</Rep><Rep D="W" H="77.0"
P="1030" S="9" T="7.9" V="50000" W="7" Pt="F" Dp="4.2">960</Rep><Rep D="WSW" H="78.7" P="1029" S="16" T="7.7" V="50000"
W="7" Pt="F" Dp="4.3">1020</Rep><Rep D="WSW" H="75.3" P="1029" S="7" T="7.5" V="50000" W="8" Pt="F" Dp="3.5">1080</Rep><Rep
D="WSW" H="77.5" P="1029" S="6" T="7.1" V="50000" W="7" Pt="F" Dp="3.5">1140</Rep><Rep D="S" H="77.5" P="1029" S="5" T="6.9"
V="45000" W="7" Pt="F" Dp="3.3">1200</Rep><Rep D="SW" H="73.8" P="1028" S="6" T="7.7" V="50000" W="7" Pt="F"
Dp="3.4">1260</Rep><Rep D="SSE" H="75.5" P="1028" S="6" T="7.7" V="50000" W="7" Pt="F" Dp="3.7">1320</Rep><Rep D="SSW"
H="76.5" P="1027" S="10" T="7.6" V="50000" W="8" Pt="F" Dp="3.8">1380</Rep></Period><Period type="Day" value="2017-12-
06Z"><Rep D="WSW" H="72.4" P="1027" S="14" T="8.6" V="50000" W="8" Pt="F" Dp="4.0">0</Rep><Rep D="SW" H="75.0" P="1026"
S="9" T="8.5" V="50000" W="8" Pt="F" Dp="4.4">60</Rep><Rep D="W" G="29" H="71.1" P="1026" S="14" T="9.7" V="50000" W="8"
Pt="F" Dp="4.8">120</Rep><Rep D="W" H="74.2" P="1025" S="18" T="9.5" V="50000" W="8" Pt="F" Dp="5.2">180</Rep><Rep D="W"
G="29" H="74.2" P="1024" S="10" T="9.5" V="50000" W="7" Pt="F" Dp="5.2">240</Rep><Rep D="W" H="68.7" P="1023" S="16"
T="10.1" V="50000" W="2" Pt="F" Dp="4.7">300</Rep><Rep D="WSW" G="32" H="78.4" P="1023" S="14" T="9.2" V="50000" W="7"
Pt="F" Dp="5.7">360</Rep><Rep D="SW" H="79.0" P="1023" S="6" T="8.9" V="50000" W="7" Pt="F" Dp="5.5">420</Rep><Rep D="SW"
G="25" H="77.9" P="1022" S="13" T="9.5" V="50000" W="7" Pt="F" Dp="5.9">480</Rep><Rep D="SW" H="76.9" P="1022" S="13"
T="9.7" V="50000" W="8" Pt="F" Dp="5.9">540</Rep><Rep D="S" G="29" H="81.8" P="1022" S="10" T="9.2" V="50000" W="8" Pt="F"
Dp="6.3">600</Rep><Rep D="S" H="79.1" P="1021" S="18" T="9.8" V="50000" W="8" Pt="F" Dp="6.4">660</Rep><Rep D="SSW" G="29"
H="73.9" P="1020" S="15" T="10.4" V="50000" W="7" Pt="F" Dp="6.0">720</Rep><Rep D="S" H="77.1" P="1019" S="9" T="10.3"
V="45000" W="8" Pt="F" Dp="6.5">780</Rep><Rep D="SSE" H="79.2" P="1018" S="10" T="10.0" V="50000" W="8" Pt="F"
Dp="6.6">840</Rep></Period></Location></DV></SiteRep>
```


```
Last observed at Topcliffe JSON: Period: 2 Day: 15


 Observation at Topcliffe JSON:-  Windspeed is: 10


Current Windspeed (In main) is: 10
```

Observation at Topcliffe JSON:- Windgust is: 0

Current Windgust (In main) is: 0

Observation at Viking Recorder JSON:- Riverstage is: 0.477

Current Riverstage (In main) is: 0.477

FS - SessionHeaderId is: 3d87a4b5-a511-479c-8844-7e81b98f5cd7

Three Day Forecast :[]

Forecast at Topcliffe XML: <?xml version="1.0" encoding="UTF-8"?><SiteRep><Wx><Param name="F" units="C">Feels Like
Temperature</Param><Param name="G" units="mph">Wind Gust</Param><Param name="H" units="%">Screen Relative
Humidity</Param><Param name="T" units="C">Temperature</Param><Param name="V" units="">Visibility</Param><Param name="D"
units="compass">Wind Direction</Param><Param name="S" units="mph">Wind Speed</Param><Param name="U" units="">Max UV
Index</Param><Param name="W" units="">Weather Type</Param><Param name="Pp" units="%">Precipitation
Probability</Param></Wx><DV dataDate="2017-12-06T13:00:00Z" type="Forecast"><Location i="3265" lat="54.204" lon="-1.39"
name="TOPCLIFFE" country="ENGLAND" continent="EUROPE" elevation="25.0"><Period type="Day" value="2017-12-06Z"><Rep D="SW"
F="6" G="20" H="77" Pp="5" S="13" T="10" V="VG" W="7" U="1">540</Rep><Rep D="SSW" F="8" G="29" H="74" Pp="1" S="16" T="10"
V="VG" W="3" U="1">720</Rep><Rep D="SSW" F="8" G="27" H="75" Pp="4" S="11" T="11" V="VG" W="7" U="1">900</Rep><Rep D="SSW"
F="7" G="29" H="81" Pp="7" S="16" T="10" V="VG" W="7" U="0">1080</Rep><Rep D="SSW" F="7" G="29" H="84" Pp="9" S="18" T="10"
V="VG" W="8" U="0">1260</Rep></Period><Period type="Day" value="2017-12-07Z"><Rep D="SSW" F="7" G="36" H="85" Pp="14"
S="20" T="11" V="VG" W="8" U="0">0</Rep><Rep D="SSW" F="8" G="34" H="86" Pp="48" S="18" T="12" V="VG" W="7"
U="0">180</Rep><Rep D="SSW" F="10" G="36" H="89" Pp="89" S="18" T="13" V="MO" W="15" U="0">360</Rep><Rep D="W" F="2" G="34"
H="75" Pp="29" S="18" T="6" V="EX" W="10" U="1">540</Rep><Rep D="W" F="4" G="34" H="69" Pp="13" S="18" T="8" V="EX" W="7"
U="1">720</Rep><Rep D="W" F="2" G="36" H="69" Pp="5" S="20" T="7" V="EX" W="1" U="1">900</Rep><Rep D="W" F="0" G="27"
H="78" Pp="32" S="13" T="4" V="EX" W="7" U="0">1080</Rep><Rep D="W" F="-1" G="20" H="85" Pp="19" S="11" T="3" V="VG" W="7"
U="0">1260</Rep></Period><Period type="Day" value="2017-12-08Z"><Rep D="W" F="-1" G="18" H="77" Pp="7" S="9" T="2" V="EX"
W="7" U="0">0</Rep><Rep D="WNW" F="-4" G="20" H="74" Pp="2" S="11" T="1" V="EX" W="2" U="0">180</Rep><Rep D="WNW" F="-6"
G="25" H="71" Pp="1" S="13" T="0" V="EX" W="0" U="0">360</Rep><Rep D="WNW" F="-6" G="27" H="68" Pp="0" S="13" T="0" V="EX"
W="1" U="1">540</Rep><Rep D="WNW" F="-3" G="34" H="62" Pp="1" S="18" T="3" V="EX" W="1" U="1">720</Rep><Rep D="WNW" F="-3"
G="34" H="63" Pp="1" S="18" T="3" V="EX" W="1" U="1">900</Rep><Rep D="WNW" F="-4" G="29" H="66" Pp="1" S="16" T="2" V="EX"
W="2" U="0">1080</Rep><Rep D="WNW" F="-4" G="25" H="67" Pp="1" S="13" T="1" V="EX" W="0" U="0">1260</Rep></Period><Period
type="Day" value="2017-12-09Z"><Rep D="WNW" F="-5" G="25" H="68" Pp="1" S="13" T="0" V="EX" W="0" U="0">0</Rep><Rep D="WNW"
F="-6" G="22" H="69" Pp="0" S="11" T="0" V="EX" W="0" U="0">180</Rep><Rep D="WNW" F="-6" G="25" H="69" Pp="0" S="11" T="-
1" V="EX" W="0" U="0">360</Rep><Rep D="WNW" F="-6" G="25" H="69" Pp="0" S="13" T="-1" V="VG" W="1" U="1">540</Rep><Rep

D="WNW" F="-2" G="27" H="63" Pp="0" S="16" T="3" V="EX" W="1" U="1">720</Rep><Rep D="WNW" F="-2" G="25" H="66" Pp="0" S="13" T="3" V="VG" W="1" U="1">900</Rep><Rep D="WNW" F="-4" G="20" H="73" Pp="0" S="11" T="1" V="VG" W="0" U="0">1080</Rep><Rep D="WNW" F="-5" G="18" H="76" Pp="0" S="9" T="0" V="VG" W="0" U="0">1260</Rep></Period><Period type="Day" value="2017-12-10Z"><Rep D="WNW" F="-6" G="16" H="80" Pp="0" S="9" T="-2" V="VG" W="0" U="0">0</Rep><Rep D="W" F="-6" G="13" H="82" Pp="0" S="7" T="-2" V="VG" W="0" U="0">180</Rep><Rep D="WSW" F="-6" G="9" H="85" Pp="2" S="7" T="-2" V="VG" W="2" U="0">360</Rep><Rep D="SSW" F="-4" G="11" H="89" Pp="7" S="7" T="-1" V="VG" W="7" U="1">540</Rep><Rep D="S" F="-1" G="16" H="86" Pp="15" S="9" T="2" V="GO" W="7" U="1">720</Rep><Rep D="SSE" F="-2" G="16" H="93" Pp="54" S="9" T="2" V="GO" W="24" U="1">900</Rep><Rep D="SSE" F="-3" G="13" H="95" Pp="72" S="7" T="1" V="GO" W="27" U="0">1080</Rep><Rep D="NNW" F="-3" G="13" H="94" Pp="58" S="7" T="0" V="GO" W="24" U="0">1260</Rep></Period></Location></DV></SiteRep>

Forecast at Topcliffe JSON:

Windspeed: 13

Wind direction: SW

Forecast at Topcliffe JSON:- Windspeed is: 13

Forecast Windspeed (In main) is: 13

Forecast at Topcliffe JSON:- Windgust is: 20

Forecast Windgust(In main) is: 20

Forecast at Topcliffe JSON:- Chance of rain is: 5

Forecast Chance of ppt Topcliffe (In main) is:5

Current Flag state for Novice crew is: Green

Current Flag state for Senior crew is: Green

Forecast Flag state for Novice crew is: Red

Forecast Flag state for Senior crew is: Red / Amber Current river stage state for all crews is: All crews can row anything