HYDROLOGICAL MODELS AS WEB SERVICES: AN IMPLEMENTATION USING OGC STANDARDS

CLAUDIA VITOLO (1), WOUTER BUYTAERT (1), DOMINIK E. REUSSER (2)
(1): Civil and Environmental Engineering Department, Imperial College London, South Kensington Campus, London, SW7 2AZ, UK
(2): Potsdam Institute for Climate Impact Research, Climate Impacts & Vulnerabilities P.O.Box 60 12 03, 14412 Potsdam, Germany

Numerous examples of web services exist, spanning a wide range of different fields and disciplines. However, environmental modelling tends to lag behind other disciplines with very few currently available implementations. This study explores the feasibility of implementing hydrological models as web services using existing and widely accepted standards and protocols.

For data storage and management, markup languages are the most common choices. Whilst encoded formats and hybrid solutions have demonstrated to be more efficient with large multi-dimensional datasets, the Open Geospatial Consortium (OGC) provides xml standards for sharing hydrological data, particularly Observations & Measurements and WaterML. Adopting such standards with a well defined set of ontologies will allow for high level of description, easier data discovery and provenance. However, in practice it may also result in slowing down parsing and processing. The right balance between standardization and speed may lay in simpler descriptions.

Scripting languages are commonly used to implement routines. The scripts can then be served and integrated with GIS tools using processing services. Interactive maps and plots are also easy to implement on-line using, for instance, Javascripts and Web Mapping Services (WMS).

As a proof of concept, a set of hydrological modelling tools has been implemented as web service in order to carry out hydrological analysis on two catchments in the UK. For each catchment, data have been stored and shared comparing different markup formats connected to a Relational Database Management System. Hydrological models (implemented in R) have been deployed using the PyWPS implementation based on the OGC-WPS standard. Geographical information is shown using Javascript and processed using the GRASS-GIS bridge available within PyWPS. Final results are displayed utilizing Google Charts.

The implemented system proves that a complete hydrological analysis can be efficiently undertaken utilizing on-line tools. However, further investigations should be carried out to harmonize the way models are coupled together.

Keywords: Hydrological Models, Web Services, OGC Standards, Virtual Observatory.
INTRODUCTION

Water resources management is facing enormous challenges, at the global and local scales. Issues such as balancing demand and supply, ensuring appropriate quality and security of water supply, and reducing negative environmental impacts, are outlined in the vision for the European water sector formulated by the European Water Supply and Sanitation Technology Platform (www.wsstp.eu). The EU Water Framework Directive has also expressed the need for accurate river flow and water quality predictions.

All these activities involve the use of water resource planning at the level of river basins, for which models and data are required. Modelling is by no means sufficient to address the above mentioned issues but it is an essential tool for data analysis, simulation and prediction, all of which support management and decision making.

An efficient application of models in management and decision making requires models that are easily accessible, portable to different environments and flexible (Buytaert et al [4]). These three demands are met by shared research environments. Astronomical scientists faced first these problems and coined the term Virtual Observatory (VO).

A Virtual Observatory is defined as a web-based collection of databases, model routines and visualization tools bound together to form a scientific research environment in which several types of analysis can be conducted.

Most of the existing environmental related web services are focused on the management of geo-databases (e.g. Google Earth Engine) and web mapping (e.g. WorldMap), whilst the processing of geo-datasets through widely accepted and validated models has been relatively unexplored.

This paper presents and analyses a web service implementation for Environmental purposes. The system is particularly tailored for modelling hydrological processes, aiming to implement the concept of 'Environmental Models of Everywhere' as envisaged by Beven [3].

A FLEXIBLE APPROACH TO HYDROLOGICAL MODELLING

A classical approach to hydrological modelling consists of a 'local' analysis of data using a specific model structure.

Modellers are incredibly skilled in the interpretation of observations through the simplification of a model. They can shape the results using model behavior. This entails the risk of seeing models as a static, optimal representation of reality, rather than falsifiable hypotheses.

Any model, instead, can be safely 'broken' into a set of concepts or hypotheses, which are interchangeable and should be able to interact with other concepts. This would lead to a process in which we can collect all these 'acceptable concepts' and, for an arbitrary catchment, instigate a process of falsification which should ideally result in an optimal combination of models for a given purpose. For this, we need a 'flexible modelling framework' which is the ground of this experimental attempt.
However, there is no infallible model, because there is no infallible representation of reality. Therefore, uncertainty must be quantified and communicated properly in order to clarify the degree of reliability of the predictions made, particularly if the results are extrapolated to data-poor environments.

EXISTING STANDARDS

When implementing hydrological models as web services it is important to integrate available information on the client’s side with libraries and database web services already available over the web. Each web service can set its own formats but to exchange data efficiently it must comply with existing and already widely accepted standards.

The Open Geospatial Consortium operates worldwide since 1994 to set interoperability standards and specifications for geospatial content and services, GIS data processing and data sharing.

There are several standard compliant tools available over the Internet to store, spread, process and interact with data. The following sections will briefly introduce existing standards with increasing level of abstraction from data formats, over modelling services to user interface.

Data formats

Data can be presented in different standard formats such as plain text, markup languages and encoded files.

To guarantee cross-client and cross-platform compatibility, some database web services adopt a plain text format. The main advantage of using plain text is that the information does not expire when the program is no more in use.

However extracting information is much easier if the format of the plain text file is self-describing. This can be achieved using a Markup Language which have many advantages:

- it combines data and metadata in one file,
- it is still a plain text and complies with a globally accepted standard,
- as cross-platform format it is not exclusive to any particular operating system or development platform,
- all databases and GIS products include ML support as a standard feature.

For environmental related purposes specific variety of ML have been developed based on the common standard for data interchange eXtensible Markup Language (XML): WaterML or hydrological data, UncertML for uncertainty, GML/KML for geographical information. Using Markup Languages the semantic meaning of the data can be extracted from the file itself making it suitable to optimally represent the metadata.

On the other hand, practice has shown that huge N-dimensional datasets are processed too slowly and they are best stored and handled in a binary format. Meteorological and climate community has first faced this problem and opted for binary formats such as GRIB and NetCDF.
UCAR recently funded the ongoing 'NcML project' to merge the advantages of the binary and XML formats. The new format (NcML) is an XML representation of NetCDF metadata. In other words it contains an XML-based metadata section that describes what is in a binary data section.

Relational databases are currently the predominant choice in storing and sharing environmental data. A common Relational Database Management System (RDBMS) can use standards such as SQL and XML and also incorporate spatial data. Geographical information can be easily imported as PostGIS (geo-extension of PostgreSQL) allowing better performances when using large size files.

Modelling Services and Processing
Tackling environmental problems requires quantitative approaches. Scripting languages such as Matlab, R and Python are currently used to develop model components and describe natural phenomena and processes. Over many years the scientific community has developed numerous models to easily simulate a wide variety of environmental processes. Despite efforts, in many cases there is no accepted method to publish and share models, leading to many re-inventions.

Web services can be used to share not only data but also model components and leverages the re-use of existing codes. Model components are implemented as functions in the above mentioned languages using wrappers. This is a straightforward way that is potentially expendable over a network.

Network capabilities and processing power of modern computers enable distributed geoprocessing on the web. Web-based geoprocessing is the evolution of geo-processing tools since geodata has become largely available through Spatial Data Infrastructures (SDI).

The OGC standard for web-based processing is the Web Processing Service (WPS). This is a general software on top of which implementations have been developed to adapt it to specific processing needs. For environmental/geographical purposes, for instance, it is very useful to access R-packages and Grass GIS capabilities (for geo-spatial analysis).

Some commonly used implementations are:

- 52North-WPS: open source implementation written in Java which supports raw data, HTTP, SOAP and WSDL. It is already linked to ArcGIS and GRASS while an R connector is under development.
- PyWPS: python-based project under a GNU General Public License. It’s main objective is the implementation of GRASS-GIS tools as web services but it also supports python scripting, OpenLayers, Mapserver and SOAP/WSDL.
- ZOO-project: C-based open source implementation to create and chain WPS Web services. In spite of other similar services, it supports several programming languages in order to provide an easy method to create new web services.

Coupling of distributed programming models can also be achieved by using many commercial and open source applications, such as OpenMI.
User interface

Web service components described so far, are meant to be used by software packages and not directly by humans. The end user interacts only with web client applications which can either be stand-alone applications such as CUAHSI-Hydro-desktop, Quantum GIS, uDIG or simple web pages.

EXISTING APPLICATIONS

There have already been attempts in many fields related to different environmental aspects:

- INTAMAP, an open source project funded by the European Commission used to exchange data, undertake statistical analysis (mainly interpolations), visualize results and communicate uncertainty through the UncertML.

- CUAHSI (a consortium of 126 universities, colleges, and research institutions from the U.S. and around the world) set up an internet-based system for sharing hydrological data (CUAHSI Hydrologic Information System) simulations, analysis and predictions (Community Hydrologic Modelling Platform) (Murdoch et al. [9]).

- OpenEarth is a free and open source initiative which hosts raw data, scripts, model schematizations and model results (netCDF collection on an OPeNDAP server) through a set of web services. It also provides an open source software for visualization (based on KML and Google Earth).

- GEO-ELCA is an interesting prototype implementation of environmental decision support systems related to the Exploratory Land Use Change Analysis. It is a demonstration of how geo-processing services can be integrated with environmental simulation models using OGC compliant connectors that support WMS and WPS (Sikder [10]).

CLOUD COMPUTING

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data-centers providing those services. The services themselves have long been referred to as Software as a Service (SaaS), while data-center hardware and software is what we call a Cloud (Armbrust et al [1]).

Cloud computing is pervasive nowadays with, internet search engines and web-based email being only few examples of the huge computational power that can be obtained running processes in parallel on thousands of CPUs.

Several studies have proven the great potential of cloud computing (Kondo et al [8]; Calheiros et al. [5]; Feng et al. [7]).
THE EXPERIMENTAL WEB SERVICE MODELLING SYSTEM

A prototype of web service modelling system was implemented at Imperial College London to test applicability and reliability of a novel approach to hydrological modelling based on FUSE modelling framework (Clark et al. [6]).

Data is stored in a PostgreSQL database and processes deployed using the python based implementation of the OGC standard WPS (PyWPS).

Communication between client and server is based on HTTP GET requests and XML responses which are seamlessly hidden behind a user friendly graphical interface.

Hydrological model, architecture and user scenario are described in the following paragraphs.

Flexible hydrological modelling

FUSE is an ensemble of four parent lumped models: PRMS, SACRAMENTO, TOPMODEL and ARNO/VIC. Each model is characterized by a different architecture of the upper and lower soil layers and for the parametrization of processes such as: evaporation, vertical percolation between soil layers, interflow, base flow and surface runoff.

FUSE can combine each element of one model with elements from other models to obtain hundreds of different model structures. Because of its flexible nature, FUSE identifies a wide variety of system behaviours, therefore selected as the most appropriate hydrological modelling framework available.

FUSE has been implemented as software package in R scripting language (based on the existing Fortran code developed by Martyn Clark) and exposed as web service, along with the already available TOPMODEL (Beven and Kirkby [2]) R package.

Web Service Architecture

The system was implemented on server machines with the following specifications and softwares installed:

- N.2 Processors: Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00 GHz
- Memory: 8 GB
- Operative System: Gnu-Linux Ubuntu 10.04
- Web Server: Apache HTTP Server
- GRASS GIS 7.0
- R version 2.13  (plus additional packages)
- Python 2.6  (plus development modules)
- PyWPS
- Rpy2
- PostgreSQL (plus PostGIS extension and administration tools)
Communication between models and data management system is based on the package RPostgreSQL, while communication between model and WPS implementation (python based) utilizes the RPy2 connector.

The service is available at http://www.envisim.org and allows to visualize public domain data and information about the available services. Sensitive data and simulation page are accessible through authentication.

**Web Service User scenario**

Once logged in, the user is in a personal workspace in which he/she can browse data from a cluster of PostgreSQL databases. The databases contain time series and geographical objects (thanks to the PostGIS extension) along with metadata.

The web client allows to query data based on location (if geographical) or period (if time-series). Information are stored in table format and can be mapped in any application schema utilizing a “translator” software. At the moment an ad-hoc schema is used but future developments will allow to create complaint OGC data specifically tailored for hydrological purposes (WaterML, O&M, etc.).

Exploratory Data Analysis is enhanced by Google Maps and Google Charts embedded in the web client. Data is processed through a list of available web services based on the PyWPS implementation of the OGC Web Processing Service.

The user can either select a catchment from the map or provide outlet coordinates. The catchment boundary is drawn on the fly utilizing public domain Digital Elevation Models (e.g. O.S. Landform Panorama) and GRASS GIS functionalities.

Hydrological models such as TOPMODEL and FUSE can then be selected, parameters modified and simulations run for a specific period of time. Results are plotted to compare observations against simulations, also highlighting the related uncertainties.

**CONCLUSIONS**

Implementation with existing software makes expression of environmental model as web service relatively simple. Making a diversity of models available through online web services would be a first step towards the realisation of the concept of modelling of everywhere.

The variety of existing solutions and the degree of stability requires some experimentation – we found the combination of R, RPy2, PyWPS, PostgreSQL and GRASS GIS to provide the best balance of features and complexity in application.

Making environmental modelling system available as web services thus provides an important bases for more collaborative research activities. Future research will highlight how efficient such a tool is compared to traditional approaches, especially in the light of the potentials of cloud computing technologies.
REFERENCES


