Environmental Modelling & Software 63 (2015) 185–198

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Review

Web technologies for environmental Big Data

Claudia Vitolo [a, *], Yehia Elkhatib [b], Dominik Reusser [c], Christopher J.A. Macleod [d], Wouter Buytaert [a]

[a] Civil and Environmental Engineering Department, Imperial College London, SW7 2AZ London, United Kingdom
[b] School of Computing & Communications, Lancaster University, LA1 4WA Lancaster, United Kingdom
[c] Potsdam Institute for Climate Impact Research, Climate Impacts & Vulnerabilities, P.O. Box 60 12 03, 14412 Potsdam, Germany
[d] The James Hutton Institute, Information and Computational Sciences Group, Craigiebuckler, Aberdeen AB15 8QH, Scotland, United Kingdom

ARTICLE INFO

Article history:
Received 21 October 2013
Received in revised form
15 September 2014
Accepted 10 October 2014
Available online 31 October 2014

Keywords:
Web-based modelling
Big Data
Web services
OGC standards

ABSTRACT

Recent evolutions in computing science and web technology provide the environmental community with continuously expanding resources for data collection and analysis that pose unprecedented challenges to the design of analysis methods, workflows, and interaction with data sets. In the light of the recent UK Research Council funded Environmental Virtual Observatory pilot project, this paper gives an overview of currently available implementations related to web-based technologies for processing large and heterogeneous datasets and discuss their relevance within the context of environmental data processing, simulation and prediction. We found that, the processing of the simple datasets used in the pilot proved to be relatively straightforward using a combination of R, RPy2, PyWPS and PostgreSQL. However, the use of NoSQL databases and more versatile frameworks such as OGC standard based implementations may provide a wider and more flexible set of features that particularly facilitate working with larger volumes and more heterogeneous data sources.

Software and data availability

Software name	Environmental Virtual Observatory pilot (EVOp)
Developers	EVOp team
Contact information	pvo@ceh.ac.uk
Hardware required	Any web-enabled device with a modern web browser
Software required	Internet browser (Chrome, Firefox and Opera)
Program languages	Java, JavaScript, R, Python and SQL.
Availability	Users can access the official website http://evo-uk.org. Access to EVOp data and applications is restricted to EVOp project partners, however user accounts can be made available to researchers upon request.

1. Introduction

1.1. The internet for sharing and linking data and models

Environmental sciences are witnessing a rapid increase in the amount of relevant published information on the internet. A large share of these data is the result of environmental monitoring, either *in situ* or via remote sensing (Kogan et al., 2010; Tsou et al., 2003) that is being made available by government institutions, private companies and citizen scientists (Buytaert et al., 2012). However, many other data that are not collected for environmental purposes may be useful for environmental science. Examples include geo-tagged photographs that may contain information about land cover and hydro meteorological conditions, disturbance patterns in telecommunication systems that provide information about weather patterns, data feeds from internet-enabled objects (the *Internet of Things* Chaouchi (2013)), online social network interactions, and many others.

In web architecture, each piece of information is typically referred to as a "resource" and can be described, regardless of its

* Corresponding author.
E-mail address: c.vitolo@imperial.ac.uk (C. Vitolo).

http://dx.doi.org/10.1016/j.envsoft.2014.10.007

type and content, by defining its properties and relations with other resources. This is the basic concept behind the Semantic Web[1] (Berners-Lee et al., 2001) that aims to generate a web of interconnected data, also called Linked Data[2] (Bizer et al., 2009). The linkage is possible by associating a unique identifier (HTTP URI) and a standardized description to each resource (Manola et al., 2004).

Linking resources in a semantic manner enhances searching capabilities over the web but the environmental sciences, amongst other disciplines, are held back in this process by practical issues. Some of them are due to the lack of structured metadata and non-common use of controlled vocabularies, some others arise from data disclosure. While there are cases in which data are simply not appropriate to be publicly published (e.g. data related to health and properties) in many other cases the lack of funding and incentives for sharing data is an insurmountable obstacle. Providing open data can be a costly process, both in terms of time and resources. Additionally, other issues such as apathy, confusion and untrusted quality-control cause databases owned and/or managed by many institutions to be not publicly access ible. As a consequence, the re-use and re-purpose of these data is often limited by intellectual property rights, patents and other mechanisms of control. On the contrary, there is a trend of increasing transparency, in which information produced at public expense should be made open and freely available to improve public involvement in the process of decision and policy making (Roberts, 2012; Hand, 2012). Many governments are currently committed to publish open data. For instance, the United Kingdom has recently launched data.gov.uk, which serves publicly available data and is based on the Linked Data paradigm, providing what is called "Linked Open Data" (LOD).

However, "publishing linked data into the cloud does not necessarily meet the requirements of reuse", scientific information/results should be "associated with provenance to aid interpretation and trust, and description of methods to support reproducibility" (Bechhofer et al., 2010). Scientists are already testing novel ways of gathering and manipulating increasing volume of data, as in the *climateprediction.net* experiment (Thorpe, 2009). The hope is to achieve "extreme openness of a data web where all information of scientific value […] is placed on the internet in machine and human-readable formats" (Nielsen, 2011).

1.2. Big Data

Big Data is defined as any collection of data sets which volume and complexity make data management and processing difficult to perform using traditional tools (i.e. handling N-dimensional data sets using plain text files and/or SQL databases). Those problems invest Big Data monoliths as much as ecosystems of small data (Pollock, 2013a) causing major concern for most private and public data providers for which "small quantities do not equal simpler management" (Akers, 2013). Even though Big Data is usually associated with the LOD concept, it is generally comprised of linked and non-linked data, open and private data, and, as such, it is characterized as being composed of the "three Vs": significant growth in the *volume*, *velocity* and *variety* of data (Dumbill, 2012). In this review, we include in the above definition of Big Data also the collection of technologies that cope with the effects of this abundance and heterogeneity, proposing solutions to meet the needs of a modern scientific community (Evans and Foster, 2011; King, 2011; Overpeck et al., 2011; Reichman et al., 2011).

Using Big Data involves many challenges. First of all, the sheer quantity of data poses technical difficulties for obtaining and processing. The Coupled Model Inter-comparison Project, for example, is producing a state-of-the-art multi-model dataset for a better understanding of climate variability and climate change. In the fifth phase of the project, the volume of produced model output and the difficulty in distribution led to the migration from a central repository to the use of a distributed system (Taylor et al., 2012).

Data heterogeneity, although used in "environmental knowledge integration" as an added value for decision makers (Blythe and Dadi, 2012), poses a major challenge to research teams as well as data-driven businesses (McAfee and Brynjolfsson, 2012). The accuracy and precision of measurements, for instance, can be highly variable depending on the source and method. Many data, including satellite images, are indirect measurements or proxies that need to be carefully processed in order to identify and attribute trends (Beven et al., 2012). Signs of expedited climate changes, for example, can be derived from the frequency and severity of extreme weather events for which classification still poses many open questions. Similarly, spatial distribution of flooding can be derived by interpreting remote sensing data. However seasonal variation of the vegetation and the small-scale topography can make patterns in land cover extremely difficult to detect.

1.3. Resource and approaches for using Big Data

Although new desktop applications are being developed to provide a number of Big Data analysis tools (Birney, 2012; Sellars et al., 2013; Steed et al., 2013), the internet, as well as being a source of data, also provides powerful tools for data processing, visualisation, simulation, prediction and sharing. A variety of projects in different countries are analysing how this potential can be harnessed, such as the UK Natural Environment Research Council-funded Environmental Virtual Observatory pilot (EVOp) project[3], the Earth Cube initiative of the US National Science Foundation[4], and the Global Earth Observation System of Systems[5] (Lautenbacher, 2005).

Because of the reliance on standardized data exchange, the internet provides a powerful environment to orchestrate complex workflows that rely upon distributed and modular components, chained together by web service technologies, as suggested by Dietze et al. (Dietze et al., 2013) who proposed the paradigm of "models as scaffold" to integrate data sources and data sets on different spatial/temporal/organizational scales.

Such integrated systems can be used to support the next generation of environmental science. By providing access to data of different sources and scales, they support the creation and execution of different workflows to process the data in different ways and provide sophisticated web tools to enable shared virtual laboratories to carry out collaborative experiments. They also encourage online publishing and reuse whilst retaining citation and provenance. This may lead to a wide and varied dissemination of data and results across domain boundaries and beyond to the general public. In this paper, we refer to such shared virtual laboratories as "environmental virtual observatories", after the eponymous UK research programme (see also Beven et al. (2012)) and other initiatives with similar intent but concerning other scientific disciplines, such as the Biodiversity Virtual e-Laboratory[6], the U.S. Virtual Astronomical Observatory[7] and the Virtual Observatory and

[1] http://www.w3.org/standards/semanticweb/, accessed 15th September 2014.
[2] http://linkeddata.org, accessed 4th October 2013.
[3] http://www.evo-uk.org, accessed 29th April 2014.
[4] http://earthcube.ning.com, accessed 27th August 2013.
[5] http://www.earthobservations.org/geoss.php, accessed 3rd September 2014.
[6] http://www.biovel.eu, accessed 27th September 2013.
[7] http://www.virtualobservatory.org, accessed 27th September 2013.

Ecological Informatics System[8] among many others. Building such environmental virtual observatories requires the use of several tools for data acquisition, analysis and communication (Laniak et al., 2013; Gibert et al., 2012; McIntosh et al., 2011). Efficient approaches do not consider those tasks separately but as a suite of interconnectable building blocks that can be orchestrated according to necessity.

Data acquisition has usually been operated via data access points (web links such as http or ftp pages) but required periodical updates. Alternative solutions involve the use of metadata catalogues to ease harvesting as well as data discovery (Ames et al., 2012). Using a catalogue allows a screening of available data sources before their acquisition. The retrieval can then be followed by data quality assessment and the result fed into a processing tool. The processing itself can also be deconstructed to several subtasks. For instance, averaging or interpolating available data in space and time is often necessary before the data can be fed into models or other algorithms for scenario analysis, hypothesis testing and prediction. Finally, the interpretation and limitations of output results are often not immediately understandable, therefore reports and numerical synthesis (e.g. tables) are also supported by visual representations such as maps and plots.

A shared virtual research environment requires these tools to be easily accessible and interoperable. In particular, interoperability of building-blocks is a major source of concern which can be limited by defining standards and setting up workflows (Merrin and Cuddy, 2009; Cuddy and Fitch, 2010).

It is therefore timely to reflect upon how technological advances can leverage more flexible and integrated data analysis in environmental science. For example, web services make it possible to modularize and flexibly combine different simulation models and tools to construct tailor-made workflows, potentially underpinning much richer and more interactive decision support systems. The construction of workflows is a particularly inter/trans-disciplinary task. The interaction or coupling of web services is facilitated by the development of standards while tools for workflow orchestration allow for automatically combining and connecting different data sources, models and web services. In cloud based systems, orchestration is fundamentally important to improve scalability and allow workflows and processes to embrace different domains.

In order to tackle environmental issues, very different types of models need to be combined. For instance, climatic, hydrological and ecological models typically have to be combined to assess climate change impacts on ecosystem services. Consequently, common standards for data encoding and representation between those scientific disciplines are developed and implemented as software specifications. They are underpinned by ontologies, which are agreements about a shared conceptualization used to organize keywords and database concepts by capturing the semantic relationships among the keywords or among tables and fields in a database (Gruber, 1993). Semantic relationships give users an abstract view of an information space for their domain of interest (Huhns and Singh, 1997) and introduce knowledge-based computing for effective integration of quantitative models, as done by Villa et al. (2009) for the ARIES project[9].

1.4. Motivations and outline

The purpose of this paper is to introduce the range of available tools and technologies for web-based environmental modelling but also document investigations undertaken by the authors when prototyping the EVOp. EVOp's aim was to link data, models and expert knowledge to make environmental monitoring and decision making more efficient and transparent to the whole community. Therefore robust and reproducible methods to access and manipulate available data were needed along with effective communication tools tailored to be used by users with different levels of expertise.

This paper, therefore, reviews the current state of art of web-based environmental data processing tools in the Big Data era. We believe this is of great significance to a myriad of efforts within the different scientific communities that are aiming to capitalise on such tools to build research collaboration environments and virtual observatories. The paper particularly focuses on the technological advancements and standards that are relevant to the environmental science community. We shed light on the different options available for assembling web service architectures, detailing aspects pertaining to data management and manipulation. We include examples of efforts that relate to the subject, describing the technological contribution of each specific project.

The paper follows the schematic structure depicted in Fig. 1. Section 2 describes typical web service architectures, highlighting the complexity of the communication between client and server. In Section 3 the existing technologies related to data discovery, storage and exchange are presented. Section 4 focuses on the processing of data over the internet, while Sections 5–7 explore existing options for visualisation of data and model results using either web services or standard compatible desktop applications, mentioning also technologies to ease discovery and chaining of those applications. Section 8 presents example applications which already combine some of the most common standards. Section 9 provides a brief multi-criteria framework to compare the presented technologies and illustrate the prototype web stack developed within the EVOp project. The assessment framework is based on a summary table in which we present the criteria taken into consideration when designing the EVOp. Some of those criteria are common to all the categories, while others are technology-specific. Common criteria inform the reader on whether a technology is considered a standard and on the level of support and complexity. Technology-specific criteria are variable and concerned with limitations, requirements and scalability amongst others. Whenever possible, the options were sorted by relevance, in descending order. Therefore, the most relevant solution to the EVOp is always on the top of the list for each category. Lastly, Section 10 draws the most important conclusions.

2. Web services and system architectures

Web services are essential in the orchestration of internet-based workflows. In essence, a web service is an application that enables access to its functions using established internet standards. As such they provide seamless cross-platform interoperability between different loosely coupled systems. Currently, two main architectural styles are most commonly used: SOAP and REST.

SOAP services use remote procedure calls to invoke functions on remote systems. Means of invocation (i.e. functions, parameters, return values, etc.) are described using the Web Services Description Language (WSDL). Using SOAP, clients generate “stubs” to match the service's interface. Data sent over the network is serialised into a structured XML format (see also Section 3), which makes it machine-readable and implementation-independent. SOAP services can discover more services through a UDDI registry (similar to a directory service), while users can do so through data portals with search capability. This architecture relies on a host of further specifications to govern such aspects as security, privacy, and reliability of message exchange.

REST, or Representational State Transfer, is an alternative architectural model where each resource has a URI. In REST, interaction

[8] http://voeis.org, accessed 27th September 2013.
[9] http://www.ariesonline.org, accessed 7th October 2013.

with the Web service is based upon stateless transfer between different resource representations. REST, thus, advocates loose coupling of applications via a uniform interface and basic HTTP operations centred on resource states rather than transactions (as in the case of SOAP services), and is very similar to the World Wide Web model. Messages exchanged in a RESTful architecture are self-descriptive using metadata. RESTful services could be described using either WSDL 2.0 or Web Resource Description Language (WRDL).

SOAP and RESTful web services have very different philosophies. SOAP is a protocol for XML-based distributed computing, whereas REST is much closer to a bare web-based design. RESTful is conceptually less complicated than SOAP, the only web protocol needed is HTTP. This means that RESTful Web services go through firewalls without special configuration and are easier to develop.

From a client point of view, the use of REST implies that parameters are passed through the URI. The example below shows the HTTP GET request to a hypothetical WPS process. This is split over two lines to illustrate its components. The first line shows the service root URI followed by the resource path, the second line shows, instead, the lists the parameters.

http://www.server.com/pywps/pywps.cgi?
service=wps&request=execute&identifier=mymodel

In a SOAP web service, instead, parameters are passed through a POST payload, as in the example below.

```
POST www.server.com
Path: /pywps/pywps.cgi

<?xml version="1.0"?>
 <soap:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body>
      <m:RunService>
        <m:service>wps</m:service>
        <m:request>execute</m:request>
        <m:identifier>mymodel</m:identifier>
      </m:RunService>
    </soap:Body>

 </soap:Envelope>
```

3. Data standards

3.1. Data encoding

At the upstream end, an environmental data processing workflow typically starts with one or several datasets. In a web environment, relevant datasets are retrieved from data services available either locally or over the internet. Depending on the service and the type of information, data can be presented in different formats. Modelling platforms are, therefore, required to interact with a mixture of data formats, including plain text, markup languages and binary files.

To enable cross-client and cross-platform compatibility, some currently existing web-based data services adopt a plain text format. The Critical Zone Observatories[10] (Niu et al., 2011) and the Geoinformatics for Geochemistry System[11] (Lehnert et al., 2003) are examples of database web services adopting plain text format. Their integrated systems store data, whenever possible, as an ASCII text table. The attached metadata uses an expanded Observations Data Model (Horsburgh et al., 2008) vocabulary or a unique sample identification code to retrieve data and set standards for metadata and data reporting. A user can also retrieve data manually, as these services make available map interfaces and visualisation tools along with analysis tools (Lehnert et al., 2003).

A main advantage of using plain text is its accessibility without specific tools, which makes the method future-proof. However, from the viewpoint of workflow orchestration, 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 is not intended to be human-readable but machine parsable. The eXtensible Markup Language (XML) is a common standard for data interchange. Utilising XML has many advantages: it combines data and metadata in one single file, it uses a text format and complies with well documented standards. As a cross-platform format, it is not exclusive to any particular operating system or development platform and it is typically well supported by data management software such as databases and GIS platforms. Additionally, specific varieties of XML have been developed for handling environmental data.

WaterML, for instance, is the standard format for the transfer of hydrologic data between data servers and users. A first version was published in 2009 WaterML1.0 (Maidment et al., 2009; Valentine and Zaslavsky, 2009). Since then WaterML has evolved towards a standard approved by the Open Geospatial Consortium[12] (OGC WaterML2.0[13], Yu et al., 2011), to enable compatibility with the OGCs web services such as the Sensor Observation Service (SOS) and/or Web Feature Service (WFS). The development of this standard proceeds in two parts: part 1 is concerned with time series, part 2 with ratings, gaugings and sections. Because part 1 is not water-specific, the standard could be applied across different domains. For this reason, there has been a recent proposal to rebrand WaterML2 (part 1) as TimeSeriesML[14]. Similarly, the OGC defined a markup language to deal with geographical features called Geography Markup Language[15] (GML, Lake, 2005), which enables convenient descriptions of vector, coverages and sensor data. While WaterML and GML handle actual data, UncertML (Williams et al., 2008) provides a conceptual model to encode metadata related to

[10] http://czo.colorado.edu/html/research.shtml, accessed 4th October 2013.
[11] http://www.earthchem.org, accessed 27th August 2013.
[12] http://www.opengeospatial.org, accessed 7th October 2013.
[13] http://www.opengeospatial.org/standards/waterml, accessed 7th October 2013.
[14] https://portal.opengeospatial.org/files/?artifact_id=56304, accessed 3rd September 2013.
[15] http://www.opengeospatial.org/standards/gml, accessed 27th August 2013.

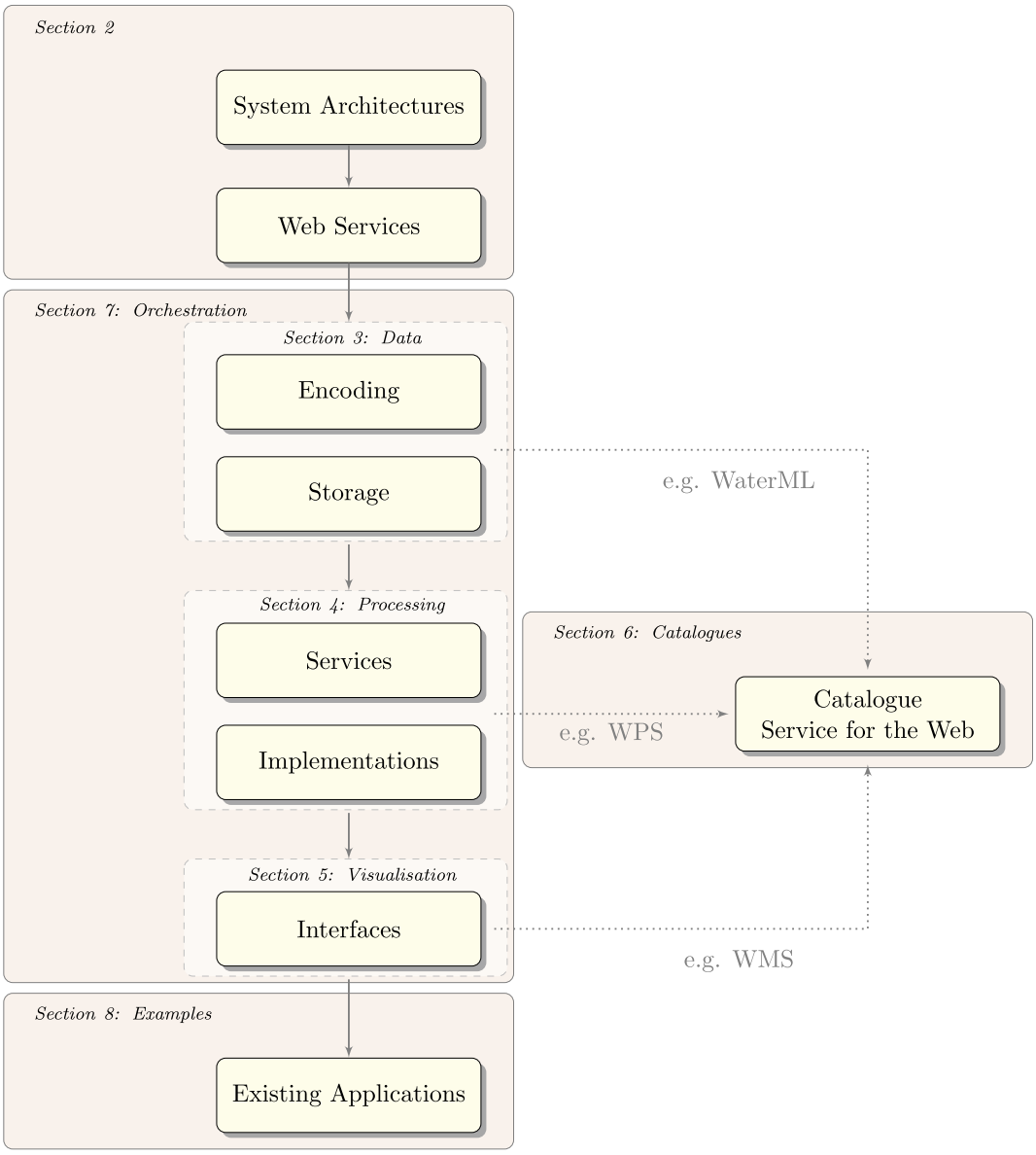

Fig. 1. Paper's schematic structure.

uncertainties, with accompanying markup language. As such, it also allows uncertainty to be propagated through data processing workflows. At the moment, UncertML is able to describe only probabilistic representations of uncertainty in random quantities. It does not deal with concepts such as fuzzy sets, random processes or belief functions (Williams et al., 2008).

By 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 plain text based markup languages often have problems of speed with processing huge N-dimensional datasets. In these cases, binary formats are better suited. The meteorological and climate communities faced this problem first and opted for binary formats such as the GRIdded Binary (GRIB), the Network Common Data Format (NetCDF) and the Hierarchical Data Format (HDF), which are all open standards.

GRIB originated from the Aeronautical Data Format and is used in meteorology to store forecast weather data. It can store a maximum of 4 dimensions, each of which has separable coordinate variables. NetCDF, developed by NASA-UCAR, is an open standard that is used for array-oriented datasets. Unlike GRIB, it supports the storage of N-dimensional data sets. Its version 4 is based on the HDF5, a hierarchical data format, which was originally developed at the National Center for Supercomputing Applications but is now supported by the non-profit HDF Group. UCAR funded the NetCDF Markup Language (NcML) project[16] (Nativi et al., 2005) to merge the advantages of the binary and XML formats. The new format is an XML representation of NetCDF metadata, containing an XML-based metadata section that describes what is in a binary data section. NcML is also useful for HDF4 and HDF5 files accessed via OPeNDAP.

3.2. Data provenance

In the context of semantic web services (SWS), data provenance is becoming increasingly important for inspecting and verifying

[16] http://www.unidata.ucar.edu/software/netcdf/ncml/, accessed 27th August 2013.

quality, usability and reliability of data in distributed computing environments (Xu and Wang, 2010; Bechhofer et al., 2010). Behind the concept of provenance is the dynamic nature of data. Data captured and/or archived for environmental purposes continues to evolve over time as it is transformed and analysed through different tools and by different organizations.

Creating different copies of the same dataset is not recommended as this introduces a data maintenance problem in the system. Instead, it is important to keep track of changes occurring and store a record of the process that led to the current state. Data provenance can, in this way, guarantee reliability of data and reproducibility of results, key issues in a scientific context (Tilmes et al., 2010).

Distributed version control systems (such as Git and Mercurial) have been designed to ease the traceability of changes, in documents, codes, plain text data sets and more recently geospatial contents[17] (Spinellis, 2012; O'Sullivan, 2009). Git and Mercurial based repositories are generally hosted online using services such as GitHub[18] (Gandrud, 2013) and BitBucket[19] to improve collaboration and efficiency especially for open source based projects. The efficiency of those tools is, however, very limited when the file size exceeds 100 MB (Pollock, 2013b).

3.3. Data storage

Relational databases were first introduced by Codd (1970) and are currently the predominant choice in storing and sharing environmental data. A common Relational Database Management System (RDBMS) assumes that data can be organised in tables (with a relatively simple structure) and that relations set among tables can be used to perform complex queries. Some popular RDBMS options are: PostgreSQL and MySQL. They both use standards such as SQL and XML and can therefore support data formats mentioned in the previous section. Technologies to handle explicitly spatial data are also well established, with specific data schemas and high-performance processing options for their large file sizes and specific structures (e.g. PostGIS[20]).

The Big Data era has, however, brought to attention many limitations associated with RDBMS, especially when handling complex data formats. With the growth of data availability and its increased heterogeneity, in fact, scalability and flexibility have become major concerns. Current scientific applications deal with large volumes of unstructured or semi-structured data such as multidimensional arrays, irregular meshes and graphs, which cannot be represented in terms of relations. Key requirements of the next generation of databases are, therefore, the capabilities to read, modify and update unstructured data sources without making copies but by versioning (spatial) data and keeping track of data provenance.

NoSQL databases have been increasingly used to overcome the inflexibility of relational databases with regard to highly heterogeneous data, and to provide improved support for distributed queries and integrated caching (Xiang and Hou, 2010). NoSQL databases do not have a predefined schema that dictates a uniform and fixed definition of the stored data in rows. In this way, database fields can be modified over time and can adapt to future requirements. NoSQL databases store data in more flexible internal structures, commonly a hierarchical structure of key–value pair arrays (e.g. DynamoDB[21], Sivasubramanian, 2012), multidimensional arrays (e.g. RASDAMAN[22] (Baumann et al., 1997, 1998) and SciDB[23] (Brown, 2010)) or objects (e.g. Versant API[24]).

Environmental and climatic scientific applications, using multidimensional raster data, tend to opt for array-based database systems. RASDAMAN, in particular uses an SQL-style language for querying and also provides service interfaces for the OGC-WCS[25]/WCPS[26]/WCST[27]/WPS[28] standards. SciDB is an alternative array database system, which features nested multidimensional array and binding with many languages such as R, Python and C++.

However, databases may contain different types of objects and the migration from SQL to NoSQL is not always a feasible option, for example, due to the high implementation costs. In these cases a NoSQL engine can be built on top of an existing relational database. This is possible with "triplestore" (Rusher, 2008), a database engine based on the Resource Description Framework (RDF) (Manola et al., 2004) and NoSQL query languages (such as SPARQL). RDF is usually used to describe information and resources on the web, where relationships between objects and their properties need to be machine-interpretable via a series of rules and reasoning ("semantic web").

Computer applications used to store and deliver database services (and optionally to perform data analysis) are called "Database Servers". Particularly relevant examples for environmental sciences are CUAHSI HydroServer[29] (Conner et al., 2013) and THREDDS Data Server, both open source solutions and OGC compliant. CUAHSI HydroServer provides data using the SOAP protocol over TCP/IP. THREDDS Data Server[30], instead, provides remote access to many types of real-time and archived scientific datasets using OPeNDAP (any CDM, e.g. NetCDF and GRIB) OGC WMS and WCS, HTTP, and other remote data access protocols. It allows the subsetting of datasets by latitude/longitude, bounding box, time range, vertical coordinates and lists of variables.

4. Modelling services and processing

Making data available through web-services, as described in the previous section, is an important part of Environmental Virtual Observatory type applications. In order to tackle environmental issues, data need to be processed with quantitative approaches. While environmental data processing algorithms and models have been written in very many environments, nowadays mathematically and statistically oriented scripting languages such as Matlab, R and Python are gaining popularity as a fast and reliable way to modular and flexible model development. Over many years the scientific community has developed numerous models to easily simulate a wide variety of environmental processes. Despite efforts to the contrary, the publishing and sharing of models has often lagged behind (Buytaert et al., 2008).

4.1. Existing standards and implementations

The OGC Web Processing Service (WPS) (Castronova et al., 2013a) has emerged as a popular standard for web-based geoprocessing, implemented in a wide range of GIS software libraries and clients (Brauner et al., 2009). WPS only defines standard means

[17] http://geogig.org/, accessed 3rd September 2014.
[18] https://github.com/, accessed 4th October 2013.
[19] https://bitbucket.org, accessed 4th October 2013.
[20] http://postgis.net, accessed 4th October 2013.
[21] http://aws.amazon.com/dynamodb, accessed 3rd September 2014.
[22] http://www.rasdaman.org/, accessed 3rd September 2014.
[23] http://scidb.org/, accessed 3rd September 2014.
[24] http://www.actian.com/products/operational-databases/, accessed 3rd September 2014.
[25] http://www.opengeospatial.org/standards/wcs, accessed 27th August 2013.
[26] http://www.opengeospatial.org/standards/wcps, accessed 7th October 2013.
[27] http://schemas.opengis.net/wcst/, accessed 27th August 2013.
[28] http://www.opengeospatial.org/standards/wps, accessed 27th August 2013.
[29] http://hydroserver.codeplex.com, accessed 27th October 2013.
[30] http://www.unidata.ucar.edu/software/tds, accessed 27th August 2013.

of communication between devices (server and clients) but to make use of existing modelling codes, additional software layers have been implemented to connect with various libraries (e.g. R packages) and geospatial tools (e.g. Grass GIS and ArcGIS).

A relatively early evaluation and implementation of WPS was presented by Michaelis and Ames (2009) who tested algorithms for watershed delineation and raster manipulation.

Popular frameworks are PyWPS[31], Zoo[32] (Fenoy et al., 2012) and 52NorthWPS[33]. PyWPS is a Python-based open source project whose main objective is the implementation of GRASS-GIS tools as web services but it also supports Python scripting, OpenLayers, Mapserver and SOAP/WSDL. One of its most convenient features is the integration into Mod_python, an Apache module, which embeds the Python interpreter within the server and guarantees 50 times faster request processing (Fenoy et al., 2012). PyWPS can connect to R through the existing connector RPy2. It is used by the Ground European Network for Earth Science Interoperations – Digital Repositories[34], INTAMAP[35] for its cross-validation service, Netmar[36] (Leadbetter et al., 2013) and the EVOp. The ZOO project is a recent open source (C-based implementation) WPS framework to create and chain WPS Web services. Contrary to other similar services, it supports several programming languages in order to provide an easy method to create new web services. ZOO allows processing of vector and raster data online in a standardized way. Zoo-Kernel can communicate to GRASS GIS through "GRASS XML to ZOO configuration file converter" and deliver a full-featured WPS. Finally, 52-North is an open source software initiative that provides implementations for many OGC standards. The entire framework is written in Java and the WPS component supports raw data, HTTP, SOAP and WSDL. It provides links to ArcGIS and GRASS-GIS functionalities while R scripts can be exposed as WPS processes through WPS4R. INTAMAP is one of the first experimental examples of web services based on the OGC WPS standard. It is built on 52-North WPS and provides functionality to exchange data, undertake statistical analysis, automatically visualise results and communicate uncertainty via UncertML (Cornford, 2009).

The above-mentioned implementations are generally applicable to a variety of contexts. However domain-specific projects may require tailored solutions, as highlighted by Goodall et al. (Goodall et al., 2011) who faced the problem of implementing web modelling services for water resources. They suggested to overcome the lack of specificity of the OGC-WPS standard by combining it with OpenMI to provide an interface specification specifically designed for water resource simulation models.

4.2. Combining multiple models

When data processing involves the use of multiple models, coupling them makes processing more efficient. There are many frameworks already developed to build modelling applications based on components, such as the well established Java-based Object Modelling System, currently at version 3.0 and supported by the U.S. Department of Agriculture and various other agencies and organizations. More recent European activities have led to the implementation of OpenMI[37] (Gregersen et al., 2007), developed in C# and Java programming languages, which has already become a standard for communication between large (commercial) models. An open source graphical user interface to couple models with OpenMI, called Pipistrelle[38], was developed by HR Wallingford within the FluidEarth project and supports MapWindow (Ames et al., 2008) for linking models and displaying GIS layers. Although OpenMI is designed for model components residing on the same computer, it has also been implemented as a web service to demonstrate its applicability in a service-oriented architecture (Goodall et al., 2007; Gijsbers et al., 2010). More recently, Castronova et al. (2013b) illustrated that OpenMI components can be used to model evapotranspiration consuming CUAHSI HIS time series data in input.

4.3. Distributed processing

Standardized web services hold promise for the sharing of model components and leveraging the reuse of existing codes. This allows for abstracting the actual implementation environment of the model behind a platform and programming language agnostic interface. In environmental science, the combined use of Big Data with complex processes is approached by using High Performance Computers (Cabellos et al., 2011) to optimize the trade-off between computational effort and the processing time of highly demanding tasks. With the advent of cloud computing, the power of distributed processing is taken to a further level using virtualization to encapsulate an operating system instance.

Public clouds are usually services offered over the internet, mainly oriented to collaborative projects. On one hand they guarantee the maximum flexibility in terms of scalability, as they have access to a large number of computer resources and can adapt the working units to the workload on demand. On the other hand, they are more exposed and therefore vulnerable if compared with the private counterpart. Security and reliability of private clouds make them the preferred choice of institutions and businesses concerned with sensible data and privacy issues, such as those in the health domain. Hybrid options, however, are relatively less adopted because of the high level of planning, management and maintenance required to provide both private and public services. For example, Amazon, which provides public elastic cloud, virtual private cloud and data storage along with features for improving scalability and load balancing, has inspired many research teams who developed cloud computing toolkits such as the multi-disciplinary Star Cluster[39]. Star Cluster is an MIT open source project intended to simplify the deployment of distributed and parallel computing applications for hydrological analysis, genetics and bio-chemistry.

Currently, the dominant implementation of cloud computing level parallel processing is MapReduce (Dean and Ghemawat, 2008). It was developed and patented by Google to process extremely large datasets over a commodity computing cluster. It abstracts the difficulties of developing scalable distributed applications, such as fault tolerance and locality-aware data distribution. Such features, along with its simplistic programming approach, allow it to be used by any programmer. MapReduce works on a set of key/value pairs as an input. The programming task is simplified into two processing stages. The Map stage processes the input set and produces an intermediate set of key/value pairs. The key/value pairs are then grouped to be processed by the Reduce stage, which generates another set of pairs. The Map and Reduce stages can be as simple or complex as required, also composing chains of

[31] http://pywps.wald.intevation.org, accessed 4th October 2013.
[32] http://www.zoo-project.org, accessed 4th October 2013.
[33] http://52north.org, accessed 4th October 2013.
[34] http://www.genesi-dr.eu, accessed 27th August 2013.
[35] http://intamap.org, accessed 4th October 2013.
[36] http://netmar.nersc.no, accessed 4th October 2013.
[37] http://www.openmi.org, accessed 22nd August 2013.
[38] http://sourceforge.net/projects/fluidearth, accessed 4th October 2013.
[39] http://star.mit.edu/cluster, accessed 4th February 2014.

computations. MapReduce can be applied to a wide range of applications, Google Web Search being a notable example.

Apache has developed an open source implementation of MapReduce called Hadoop[40] (White, 2010), successfully applied to a variety of computational problems. Examples include commercial uses such as Facebook and eBay, and scientific research such as Geographical Information Systems (Chen et al., 2008), cell structure analysis (Zhang et al., 2010) and image coaddition (Wiley et al., 2010). An R and Hadoop Integrated Processing Environment (RHIPE) also exists. A commonly used alternative is the Message Passing Interface (MPI) (The MPI Forum, 1993) or its well known free implementation OpenMPI[41]. MPI and OpenMPI are libraries of functions/subroutines which can run on either shared or distributed memory architectures. Those functions are, however, implemented at a level more strongly tied into a particular platform which makes more difficult to scale them easily to cloud computing applications. In addition their performance is limited by the communication network between the nodes.

5. Data visualisation and interaction

Effective visualisation is a key element in applications for decision support, whether to show available data or output data processing and simulation results. Web services are particularly suitable for this scope. Current technologies provide tools, which are as rich and interactive as common desktop applications. However, web-based applications are more accessible and can be generated based on an adaptive design. Much of the information technology research is, in fact, investing in exploring smarter ways of dynamically adapting the content of websites and services to better address user needs (Yao and Ohsuga, 2000; Brusilovsky et al., 2007).

Web charts and maps already allow for user interaction. Users can, for example, read values for data points directly hovering over a graph, zoom in/out on a particular portion of a map/graph and overlap different information and scenarios on demand. Many open source Javascript plotting libraries provide excellent plotting tools. Some examples are the jqPlot[42] and Flot[43] for jQuery[44], but also Protovis[45], Processing[46], Raphael[47], D3[48], Google charts[49] among many others. Wikipedia provides a comprehensive evaluation framework for comparing many charting options[50].

Deploying georeferenced map images over the Internet, instead, is commonly done by using the OGC dedicated standard called Web Mapping Service[51] (WMS). A WMS consists of a mapping server using data from a GIS database. Major GIS and mapping software support WMS, e.g. MATLAB, ESRI's products, Google Earth, QGIS, and GRASS GIS. The most widely used platforms for publishing spatial data and interactive mapping applications on the web are MapServer and GeoServer, both open source software. MapServer[52] is a geographic data rendering engine written in C which also supports PHP, Python, Perl, Ruby, Java and .NET. As client or server, it supports several OGC standards and a multitude of raster data formats (via GDAL library), vector data formats (via OGR library) and projections (via Proj.4 library). A MapServer application can be easily set up using frameworks such as p.mapper (PHP/MapScript) and GeoExt (Javascript) which integrate functions like: zoom/pan interface, query, multilingual user interface, and a plugin API to add custom functionalities. GeoServer[53] is an easy to use software server written in Java that supports several vector and raster data formats as well as embedding the EPSG database for map projections.

Although based on a different approach, Google Maps has been for years one of the most popular applications for web-mapping. Google has developed numerous internet-based mapping applications, some of them also available as desktop applications (e.g. Google Earth). Google Earth can establish WMS connections and save/share the content as a KML file. Both Google Earth and Google Maps can access Google Earth Engine, a platform providing an extremely large repository of georeferenced satellite imagery, terrain datasets, and vector data (such as roads, borders, population centres, soil information and climate information). Google Earth Engine also allows researchers and scientists to analyse the imagery through Google's own computing infrastructure. This is particularly relevant when time for processing is restrictive or the amount of data to analyse is prohibitive with normal infrastructures. Comparing the existing mapping applications is not an easy task. Performances are extremely variable with the nature of the task to perform, the type of data to use and the server machine utilized. For this reason, the Open Source Geospatial Foundation[54] sponsors every year a benchmarking session for desktop and web based mapping applications at the FOSS4G conference, which results are available online[55].

5.1. Interfaces

The web service components described so far are meant to be used by software packages and not by users. The end user interacts with applications, typically referred to as “clients”. Some examples of interfaces include CUAHSI-HydroShare[56], QGIS[57], and uDig[58], among many others. The CUAHSI-coordinated HydroShare project aims to compliment the desktop-based client HydroDesktop. QGIS is a cross-platform open source GIS application that can be extended easily with modules written in Python or C++. As an OSGeo Foundation's project, it has evolved incredibly fast during recent years. It incorporates WS-features like: importing GPS data into PostGIS, support for OpenLayers, WMS, WFS and WPS. The most recent version of QGIS is also capable of serving maps similar to Mapserver and Geoserver. Lastly, uDig is an open source application framework based on Java, which aims at providing a solution for desktop GIS, data access, editing and viewing.

6. Web catalogues

Once data and services are developed, tested and available in the public domain, they can theoretically be accessed from anywhere. However, discovering available services is difficult without proper

[40] http://hadoop.apache.org, accessed 4th October 2013.
[41] http://www.open-mpi.org, accessed 27th August 2013.
[42] http://www.jqplot.com, accessed 4th October 2013.
[43] http://www.flotcharts.org, accessed 4th October 2013.
[44] http://jquery.com, accessed 4th October 2013.
[45] http://mbostock.github.io/protovis, accessed 4th October 2013.
[46] http://processing.org, accessed 4th October 2013.
[47] http://raphaeljs.com, accessed 4th October 2013.
[48] http://d3js.org, accessed 4th October 2013.
[49] https://developers.google.com/chart, accessed 4th October 2013.
[50] http://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks, accessed 4th October 2013.
[51] http://www.opengeospatial.org/standards/wms, accessed 27th August 2013.
[52] http://mapserver.org, accessed 4th October 2013.
[53] http://geoserver.org, accessed 4th October 2013.
[54] http://www.osgeo.org/, accessed 4th August 2014.
[55] http://wiki.osgeo.org/wiki/Benchmarking_2013, accessed 4th February 2014.
[56] http://www.cuahsi.org/HydroShare.aspx, accessed 17th September 2013.
[57] http://www.qgis.org, accessed 19th February 2014.
[58] http://udig.refractions.net, accessed 19th February 2014.

description of their functionality and other metadata. Such service is typically offered through a catalogue. An example of an internet-based system providing unified access to data, tools and models is the CUAHSI Hydrologic Information System[59] (Horsburgh et al., 2009), which allows users to discover, use and manage time series published by agencies and universities using the standard WaterOneFlow and WaterML1.0 as output format.

However, discovering a service is not trivial due to semantic heterogeneity. There is a continuing need for further standardization of definitions to ensure consistency among concepts belonging to the same domain and across different domains. Chilingarian et al. (2007) also demonstrate how capturing the semantics of distributed archived information and tools will lead to more effective discovery and interoperability. For this reason, the OGC uses controlled vocabularies such as the Web Ontology Language (McGuinness and Van Harmelen, 2004), to develop data models (e.g. OGC-ODM) and define a standard Catalogue Service for the Web[60]. The latter is comprised of an application schema for metadata used for both the registration and discovery of services (Gwenzi, 2010). The python-based PYCSW[61] is the most popular implementation of the CSW standard.

7. Workflow orchestration

As individual model components can be coupled to work as a unique modelling platform, so web services can be chained together to discover sources of information, process them and communicate the results on-the-fly. This process is typically referred to as workflow orchestration. The complexity of web-based component chaining can be significantly reduced by the use of dedicated orchestration software (Weiser and Zipf, 2007). Additional advantages of formal workflow orchestration are a more controlled and auditable execution and re-execution of the entire procedure.

By definition, a workflow is an execution pipeline. It is composed of basic execution units, such as executable binaries, scripts and web services. They provide advanced users (i.e. domain specialists from the scientific or governmental communities) with the capability to create complex self-contained experiments that can later be easily tweaked and replayed. This offers great added value in terms of reproducibility and traceability. If described in a standard way, a workflow can be shared and reused by others in order to build upon it, reproduce results, or compare techniques. Indeed, sharing workflows has proven to be quite useful in other fields of science to support collaborative research communities (e.g. bioinformatics).

However, engineering workflows is a major challenge for scientists. Workflows have the propensity to become increasingly complex, with an increasing number of potentially heterogeneous data sources to be combined and connected. During the last decade, a variety of workflow orchestration tools has emerged and been adopted by various scientific disciplines. Some examples are Taverna, Kepler, jABC and BPEL (De Jesus et al., 2012b,a; Yu et al., 2012; Da Silva et al., 2012; Steffen et al., 2007; Lamprecht, 2013). Taverna's website provides extensive documentation and a section dedicated to clarify the differences with Kepler[62]. Those are mainly related to the models of computation utilized and the user communities served. Kepler is computationally more flexible, however Taverna seems to have a wider user community. The jABC is a multi-purpose modelling framework with workflow applications in scientific as well as in technical and business-oriented domains. In contrast to Taverna, Kepler and the majority of other scientific workflows systems that follow a data-flow modelling approach, workflow models in the jABC represent the flow of control and can thus also express more complex program structures (e.g. conditional branches and loops). Another distinguishing feature is that it has been developed with a particular focus on the incorporation of formal methods in the workflow development process. BPEL, instead, was initially designed to be used with business workflows however is being increasingly used by many scientific communities (Tan et al., 2010).

8. Existing applications of integrated systems

Achieving full integration between data and models in a user-friendly web tool is an ambitious target, but there are many attempts in fields related to different environmental aspects. While a full review of these tools is beyond the scope of this paper, some notable examples include OpenEarth, ROADNet, REAP, GEO-ELCA, and DataONE.

In order to share knowledge and lessons learnt from different projects and to avoid replicating previous efforts, OpenEarth[63] launched its own integrated approach for managing data, models and tools (van Koningsveld et al., 2010). OpenEarth is a free and open source initiative that hosts raw data, scripts, model schematization and model results (NetCDF collection on an OPeNDAP server) through a set of web services. It also provides open source software for visualisation (based on KML and Google Earth).

The ROADNet project[64], aims to develop an integrated, seamless, and transparent environmental information network that will deliver geophysical, oceanographic, hydrological, ecological and physical data to a variety of end users in real-time. ROADNet's architecture provides a suite of functionalities seamlessly assembled to form a grid which will address system and data interoperability issues, but it does not yet address semantic interoperability and information integration issues (i.e. techniques that move beyond simple distributed access to data files). ROADNet does not include features for persistent archives and for user tools and interfaces.

Realtime Environment for Analytical Processing[65] (REAP) is a cyber infrastructure development project, focused on creating technology in which scientific workflow tools can be used to access, monitor, analyse and present information from field-deployed sensor networks, for both the oceanic and terrestrial environments and across multiple spatio-temporal scales. This environment for near real-time analytical processing provides an open source, extensible and customizable framework for designing and executing scientific models that consume data streams from sensor networks, and for combining data grids constructed through other projects (ROADNet, CENS ESS, OPeNDAP, EarthGrid) with the scientific workflow management system Kepler (Ludäscher et al., 2006).

There is also GEO-ELCA, a prototype implementation of an environmental decision support system 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, 2008).

[59] http://his.cuahsi.org, accessed 27th August 2013.
[60] http://www.opengeospatial.org/standards/cat, accessed 27th October 2013.
[61] http://pycsw.org, accessed 27th October 2013.
[62] http://www.taverna.org.uk/documentation/faq/general/kepler-taverna-difference/, accessed 4th February 2014.
[63] http://www.openearth.nl, accessed 4th October 2013.
[64] http://roadnet.ucsd.edu, accessed 4th October 2013.
[65] http://reap.ecoinformatics.org, accessed 28th August 2013.

Table 1
Summary table of web technologies taken into consideration for the EVOp. Columns 2 to 4 refer to common criteria, while the last three columns are technology specific. Common criteria attempt to clarify: 1) if the technology is considered a standard, 2) the level of support (high/medium/low), 3) the level complexity in terms of usability (high/medium/low). Conventionally, a small file (SF) is ≤100 MB and a large file (LF) >100 MB.

	Standard	Support	Complexity			
WS type				**Requirements**	**App. state**	**Scalability**
RESTful	no	medium	low	use of HTTP	stateless	easy
SOAP	yes	high	high	none	server-side	difficult
Data type				**Requirements**	**Share**	**Limitations**
Plain text file	no	low	low	none	easy	non-easily parsable, SF
XML-based	yes	high	medium	libraries	easy	non-human readable, SF
Binary	yes	high	medium	libraries	difficult	non-human readable
XML-Binary	no	low	high	libraries	difficult	non-human readable
Database type				**Query lang.**	**Size**	**Limitations**
SQL	yes	high	medium	standard	small to medium	no complex data
NoSQL	no	medium	medium	non-standard	medium to large	none
WPS implem.				**Language**	**Other languages support**	**Other OGC WxS supported**
PyWPS	yes	low	medium	Python	R via RPy2	WMS/WFS/WCS
52NorthWPS	yes	high	medium	Java	R via WPS4R	many
ZOO	yes	medium	medium	C++	many languages	WMS
Workflow app.				**Model of computation**	**Communities**	**On-line editor**
Taverna	no	medium	medium	many	Environmental, Bioinformatics, Physics	yes (experimental)
Kepler	no	medium	medium	lambda calculus	Engineering, Life and Computer Sciences	no
BPEL	no	medium	medium	many	mainly Businness	yes (experimental)
jABC	no	medium	low/medium	control-flow semantics	Scientific and Business	under development
Javascript visualisation library				**Pre-processing effort**	**Supported plot types**	**Customisability**
D3	no	high	high	low	many	high
Flot	no	high	medium	low	many	high
Protovis	no	medium	high	low	many	high
Cloud deployment model				**Upfront cost**	**Time to build**	**Data protection**
Public	no	high	low	low	low	low-medium
Private outsourced	no	low	medium	high	medium	high
Private self-managed	no	medium	high	high	high	medium

Finally, DataOne[66] is another cyber-infrastructure, supported by the U.S. National Science Foundation, to access Earth observational data (Michener et al., 2011). Its first prototype implements client libraries in Java, Python and R and aims to enable users to mount the entire DataONE cloud infrastructure as a file system.

9. EVOp implementation

The EVOp features a set of web applications to allow access, visualisation and use of various environmental information via models and local community tools. A multi-criteria framework was set up to select, amongst the web technologies illustrated in the previous sections, those suitable to build a web stack prototype for the EVOp web applications. The criteria used in the assessment are summarised in Table 1 and divided into seven sections: web service type, data type, database type, WPS implementation, workflow application, Javascript visualisation library.

The first section shows that both SOAP and RESTful styles have, initially, been considered. However, due to ease of development, use of the existing web infrastructure, the steep learning curve amongst other reasons, REST architecture was the most appropriate choice for the majority of services provided by the EVOp (Elkhatib et al., 2013). From a modelling viewpoint, the goal was to identify the most suitable technologies and existing implementations that would support the heterogeneity of environmental data and modelling tools. The main requirement, was to be able to switch between a widely used model such as TOPMODEL (Beven and Kirkby, 1979) and a novel modelling framework called FUSE (Clark et al., 2008). The source code for both models was already available. TOPMODEL was implemented as an R package while FUSE was available as Fortran code. At the time the R community had already shown wide interest in web technologies producing numerous packages to interact with web data sources, R connectors to other web-oriented languages and implementations of the OGS WPS standard. R became the natural choice for the modelling backend, as any modelling task can be web-enabled with minimum effort.

On the data management side, instead, the requirement was to span various online data sources, allowing to demonstrate the possibility to extract information from html pages (web-scraping), to share media content (e.g. images and videos) and to link with real-time data services (e.g. last few hours of rainfall recordings). However, this set of information was not sufficient to feed the selected modelling tools. The team obtained the necessary data from partner institutions, accepting the terms and conditions of a restrictive license which did not allow to deploy the demo applications in the public domain. Data consisted of numerous small datasets in plain text format. This did not pose particular challenges in terms of sharing and transferability but mainly in parsing the information as datasets were not formatted in a standard way. Information was therefore collated, standardized and transferred into a PostgreSQL database.

Models were deployed using the python based implementation of the OGC WPS standard (PyWPS). The choice was mainly driven by the fact that the application required the use of routines previously developed in the R language. At the time, the only working option that allowed to reliably call R libraries was PyWPS through the RPy2 connector. If the same choice was made today, the prototype would have probably made use of the 52North implementation, which currently provides the most comprehensive and well supported framework. The orchestration of the processes needed to perform uncertainty analysis based on the GLUE methodology (Beven and Binley, 1992) was briefly explored using Taverna, which was considered the most appropriate choice due to the extensive support to the environmental community and the

[66] http://www.dataone.org, accessed 28th August 2013.

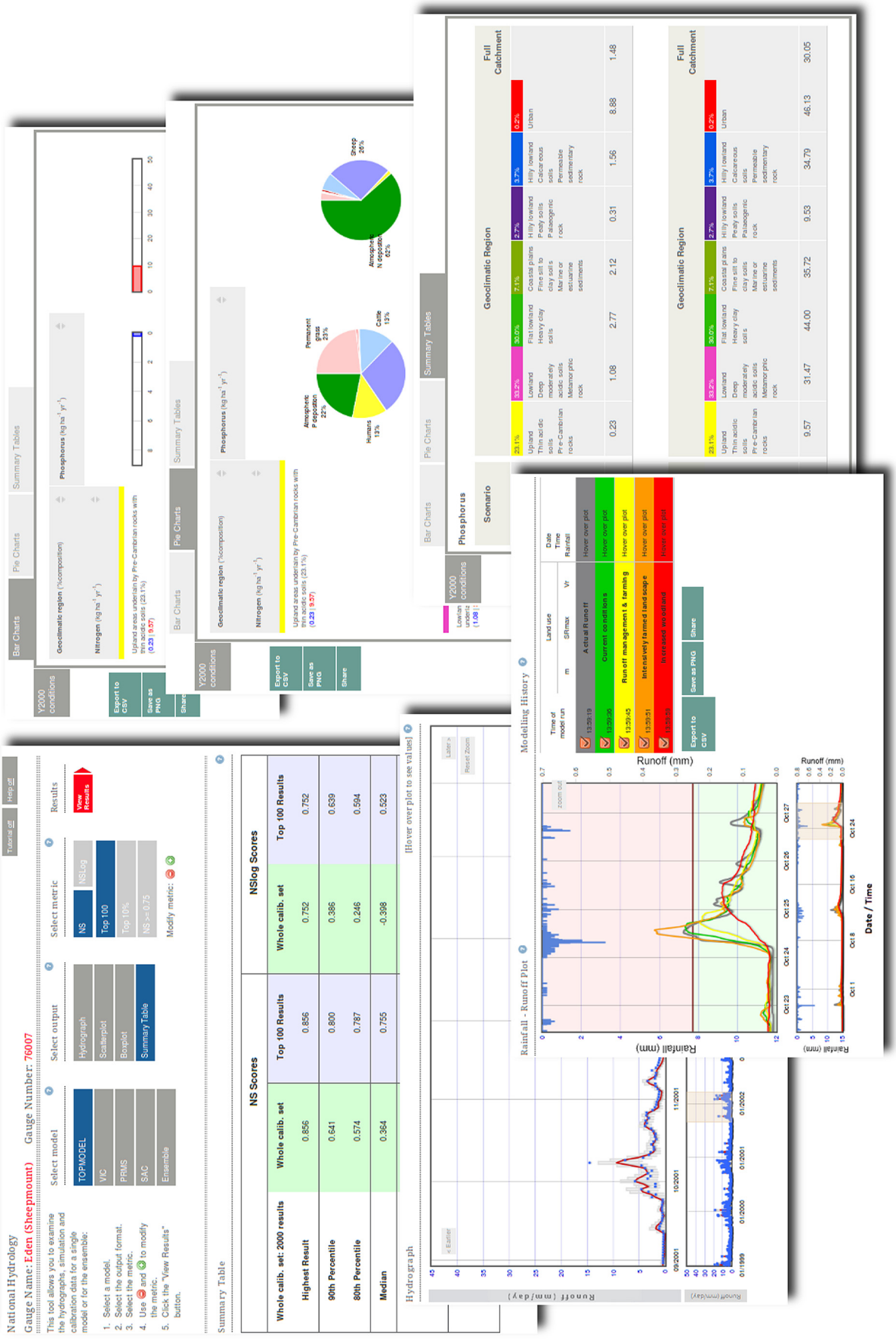

Fig. 2. Examples of interactive charts implemented as part of EVOp.

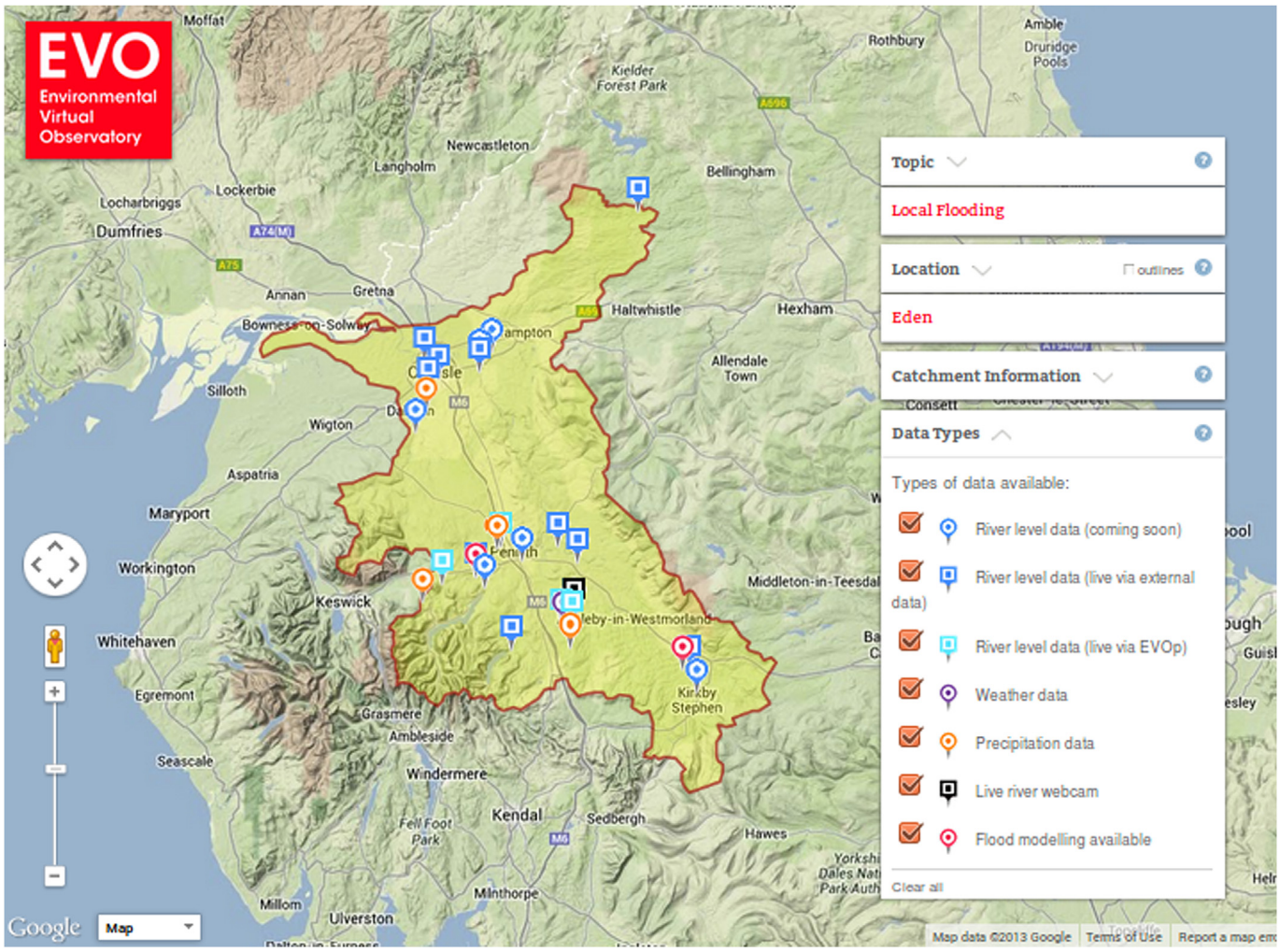

Fig. 3. EVOp's map explorer page.

development of an experimental online workflow editor[67]. Model results were generated in XML format to ease transferability and interoperability among different components. For larger size and variety of data, instead, binary and hybrid XML-binary solutions would have been more appropriate.

Communication between client and server was based on HTTP GET requests and XML responses, as suggested by the OGC WPS standard. Those non-human readable formats were carefully hidden behind a user friendly graphical interface. Due to the team's previous experience and familiarity with Flot and Google maps, those tools were used to generate the EVOp interactive charts shown in Fig. 2 and the map explorer in Fig. 3.

10. Conclusions

This paper presents a review of the most relevant web technologies dealing with "Big Environmental Data". A common thread and the main motivation of this work is to document investigations carried out when prototyping the UK Environmental Virtual Observatory pilot.

Evidence has shown that technologies can be effectively combined in many different ways depending on the specific modelling needs. However domain-specific projects require often tailored solutions. Numerous options for data formats, storage, processing, visualisation and chaining of service components are taken into consideration.

We found that, for example, despite the common practice of using plain text, self-describing data formats would be a better solution to store and transfer environmental data as they could integrate metadata information and standardised definitions of domain-specific variables and uncertainties. Also, as larger volumes of data become available, data becomes less structured and therefore more complex. NoSQL databases have been found to deal better with complex and non structured information than traditional relational databases. Even though web-based processing can be approached in many different ways, at the moment the 52North framework seems to provide the most comprehensive and well supported platform currently available. Javascript libraries (e.g. Flot and D3) provide great potential to enable highly customised and interactive web-based visualisation. A clear separation line cannot be drawn, instead, for the most popular workflow orchestration tools, which functionalities are very similar.

Acknowledgements

This work was supported by the Natural Environment Research Council pilot projects on Environmental Virtual Observatory technologies NE/I002200/1 and NE/I004017/1.

[67] http://onlinehpc.com, accessed 3rd September 2014.

The authors would like to thank the EVOp full project team for their collaboration: Lucy Ball (CEH), Gordon Blair (Lancaster University), John Bloomfield (BGS), P. Brewer (Aberystwyth University), Lucy Cullen (CEH), Julie Delve (CEH), Bridget Emmett (CEH), Jim Freer (Bristol University), Sheila Greene (Reading University), Robert Gurney (Reading University), P.M. Haygarth (Lancaster University), Penny Johnes (Reading University), Jane Lewis (Reading), E. Mackay (Lancaster University), M. Macklin (Aberystwyth University), K. Marshall (The James Hutton Institute, Aberdeen), Adrian McDonald (Leeds University), Nick Odoni (Bristol University), Barbara Percy (Reading University), P.F. Quinn (Newcastle University), Gwyn Rees (CEH), M. Stutter (The James Hutton Institute, Aberdeen), Bholanath Surajbali (Lancaster University), Doerthe Tetzlaff (Aberdeen University), N. Thomas (Aberystwyth University), John Watkins (CEH), M.E. Wilkinson (Newcastle University, now at The James Hutton Institute, Aberdeen), Bronwen Williams (CEH).

References

Akers, K.G., Feb. 2013. Looking out for the little guy: small data curation. Bull. Am. Soc. Inf. Sci. Technol. 39 (3), 58–59.

Ames, D.P., Michaelis, C., Anselmo, A., Chen, L., Dunsford, H., 2008. MapWindow GIS. In: Shekhar, S., Xiong, H. (Eds.), Encyclopedia of GIS. Springer US, Boston, MA, pp. 633–634.

Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., Nov. 2012. HydroDesktop: web services-based software for hydrologic data discovery, download, visualization, and analysis. Environ. Model. Softw. 37, 146–156.

Baumann, P., Furtado, P., Ritsch, R., Widmann, N., 1997. Geo/Environmental and medical data management in the RasDaMan system. In: In Proceedings of the 23th VLDB Conference, pp. 548–552.

Baumann, P., Dehmel, A., Furtado, P., Ritsch, R., Widmann, N., Jun. 1998. The multidimensional database system RasDaMan. ACM SIGMOD Rec. 27 (2), 575–577.

Bechhofer, S., Ainsworth, J., Bhagat, J., Buchan, I., Couch, P., Cruickshank, D., DeRoure, D., Delderfield, M., Dunlop, I., Gamble, M., Goble, C., Michaelides, D., Missier, P., Owen, S., Newman, D., Sufi, S., Dec. 2010. Why linked data is not enough for scientists. In: 2010 IEEE Sixth International Conference on e-Science. IEEE, pp. 300–307.

Berners-Lee, T., Hendler, J., Lassila, O., May 2001. The semantic web. Sci. Am. 284 (5), 34–43.

Beven, K.J., Binley, A.M., 1992. The future of distributed models: model calibration and uncertainty prediction. Hydrol. Process. 6, 279–298.

Beven, K.J., Kirkby, M.J., 1979. A physically based variable contributing area model of basin hydrology. Hydrol. Sci. Bull. 24, 43–69.

Beven, K., Buytaert, W., Smith, L.A., Jun. 2012. On virtual observatories and modelled realities (or why discharge must be treated as a virtual variable). Hydrol. Process. 26 (12), 1905–1908.

Birney, E., Sep. 2012. The making of ENCODE: lessons for big-data projects. Nature 489 (7414), 49–51.

Bizer, C., Heath, T., Berners-Lee, T., Jan. 2009. Linked data – the Story so far. Int. J. Semant. Web Inf. Syst. 5 (3), 1–22.

Blythe, J.N., Dadi, U., 2012. Knowledge integration as a method to develop capacity for evaluating technical information on biodiversity and ocean currents for integrated coastal management. Environ. Sci. Policy 19, 49–58.

Brauner, J., Foerster, T., Schaeffer, B., Baranski, B., 2009. Towards a research agenda for geoprocessing services. In: 12th AGILE International Conference on Geographic Information Science 2009 Leibniz Universität. Vol. 1 of AGILE 2009. Hannover, Germany, pp. 1–12.

Brown, P.G., 2010. Overview of sciDB. In: Proceedings of the 2010 International Conference on Management of Data – SIGMOD'10. ACM Press, New York, New York, USA, p. 963.

Brusilovsky, P., Kobsa, A., Nejdl, W., 2007. The Adaptive Web. In: Lecture Notes in Computer Science, vol. 4321. Springer Berlin Heidelberg, Berlin, Heidelberg.

Buytaert, W., Reusser, D., Krause, S., Renaud, J., 2008. Why can't we do better than Topmodel? Hydrol. Process. 22 (August), 4175–4179.

Buytaert, W., Baez, S., Bustamante, M., Dewulf, A., 2012. Web-based environmental simulation: bridging the gap between scientific modeling and decision-making. Environ. Sci. Technol. 46 (4), 1971–1976.

Cabellos, L., Campos, I., Fernández-del Castillo, E., Owsiak, M., Palak, B., Póciennik, M., Apr. 2011. Scientific workflow orchestration interoperating HTC and HPC resources. Comput. Phys. Commun. 182 (4), 890–897.

Castronova, A.M., Goodall, J.L., Elag, M.M., 2013a. Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. Environ. Model. Softw. 41, 72–83.

Castronova, A.M., Goodall, J.L., Ercan, M.B., Jan. 2013b. Integrated modeling within a hydrologic information system: an OpenMI based approach. Environ. Model. Softw. 39, 263–273.

Chaouchi, H. (Ed.), Feb. 2013. The Internet of Things. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Chen, Q., Wang, L., Shang, Z., Dec. 2008. MRGIS: a MapReduce-enabled high performance workflow system for GIS. In: IEEE Fourth International Conference on EScience (eScience'08), pp. 646–651.

Chilingarian, I., Bonnarel, F., Louys, M., McDowell, J., 2007. Handling IFU datasets in the virtual observatory. In: Kissler-Patig, M., Walsh, J., Roth, M. (Eds.), Science Perspectives for 3D Spectroscopy, ESO Astrophysics Symposia, vol. 39. Springer Berlin/Heidelberg, pp. 29–31.

Clark, M.P., Slater, A.G., Rupp, D.E., Woods, R.A., Vrugt, J.A., Gupta, H.V., Wagener, T., Hay, L.E., 2008. Framework for Understanding Structural Errors (FUSE): a modular framework to diagnose differences between hydrological models. Water Resour. Res. 44, 91–94.

Codd, E.F., Jun. 1970. A relational model of data for large shared data banks. Commun. ACM 13 (6), 377–387.

Conner, L.G., Ames, D.P., Gill, R.A., Nov. 2013. HydroServer Lite as an open source solution for archiving and sharing environmental data for independent university labs. Ecol. Inf. 18, 171–177.

Cornford, D., 2009. INTAMAP: Implementation of a Web Client to Access the Web Service in a Simple to Use Manner. Tech. Rep. http://www.intamap.org/documents/D1_6_Implementation_of_a_web_client.pdf.

Cuddy, S.M., Fitch, P., 2010. Hydrologists Workbench – a hydrological domain workflow toolkit. In: Swayne, David A., Yang, Wanhong, Voinov, A.A., Rizzoli, A., Filatova, T. (Eds.), International Environmental Modelling and Software Society (iEMSs) 2010 International Congress on Environmental Modelling and Software Modelling for Environments Sake, Fifth Biennial Meeting. Ottawa, Canada, pp. 1–9. In: http://www.iemss.org/iemss2010/index.php?n=Main.Proceedings.

Da Silva, L.M., Braga, R., Campos, F., Mar. 2012. Composer-Science: a semantic service based framework for workflow composition in e-Science projects. Inf. Sci. 186 (1), 186–208.

De Jesus, J., Walker, P., Grant, M., Apr. 2012a. Creating OGC web processing service workflows using a web-based editor. EGU General Assem. Conf. Abstr. 14, 5734. http://adsabs.harvard.edu/abs/2012EGUGA..14.5734D.

De Jesus, J., Walker, P., Grant, M., Groom, S., Oct. 2012b. WPS orchestration using the Taverna workbench: the eScience approach. Comput. Geosci. 47, 75–86.

Dean, J., Ghemawat, S., Jan. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 107–113.

Dietze, M.C., Lebauer, D.S., Kooper, R., Sep. 2013. On improving the communication between models and data. Plant, Cell Environ. 36 (9), 1575–1585.

Dumbill, E., 2012. What is big data? An introduction to the big data landscape. http://strata.oreilly.com/2012/01/what-is-big-data.html.

Elkhatib, Y., Blair, G.S., Surajbali, B., Apr. 2013. Experiences of using a hybrid cloud to construct an environmental virtual observatory. In: Proceedings of the 3rd International Workshop on Cloud Data and Platforms – CloudDP'13. ACM Press, New York, New York, USA, pp. 13–18.

Evans, J.A., Foster, J.G., Feb. 2011. Metaknowledge. Science (New York, N.Y.) 331 (6018), 721–725.

Fenoy, G., Bozon, N., Raghavan, V., Jan. 2012. ZOO-Project: the open WPS platform. Appl. Geomat. 5 (1), 19–24.

Gandrud, C., 2013. GitHub: a tool for social data development and verification in the cloud. Political Methodol. 20 (2), 7–16.

Gibert, K., Sànchez-Marrè, M., Sevilla, B., 2012. Tools for environmental data mining and intelligent decision support. In: Seppelt, R., Voinov, A.A., Lange, S., Bankamp, D. (Eds.), International Environmental Modelling and Software Society (iEMSs) 2012 International Congress on Environmental Modelling and Software Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany, pp. 1726–1734. In: http://www.iemss.org/society/index.php/iemss-2012-proceedings.

Gijsbers, P., Hummel, S., Vaneçek, S., Groos, J., Harper, A., Knapen, R., Gregersen, J., Schade, P., Antonello, A., Donchyts, G., 2010. From OpenMI 1.4 to 2.0. In: International Congress on Environmental Modelling and Software Modelling for Environments Sake, Fifth Biennial Meeting, Ottawa, Canada.

Goodall, J.L., Robinson, B.F., Shatnawi, F.M., Castronova, A.M., 2007. Linking hydrologic models and data: the OpenMI approach. Eos Trans. AGU – Fall Meet. Abstr. Suppl. 1 (88), 52.

Goodall, J.L., Robinson, B.F., Castronova, A.M., 2011. Modeling water resource systems using a service-oriented computing paradigm. Environ. Model. Softw. 26 (5), 573–582.

Gregersen, J.B., Gijsbers, P.J.A., Westen, S.J.P., Jul. 2007. OpenMI: open modelling interface. J. Hydroinf. 9 (3), 175.

Gruber, T.R., Jun. 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5 (2), 199–220.

Gwenzi, J., 2010. Web Technology and Metadata Visualisation (Master thesis). International Institute for Geo-information Science and Earth Observation, Enschede, The Netherlands. http://geonetwork.tv/owl/JulietMSCthesislatest.pdf.

Hand, D., Jul. 2012. Open Data is a Force for Good, but Not without Risks. http://www.theguardian.com/society/2012/jul/10/open-data-force-for-good-risks.

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., May 2008. A relational model for environmental and water resources data. Water Resour. Res. 44 (5).

Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. An integrated system for publishing environmental observations data. Environ. Model. Softw. 24 (8), 879–888.

Huhns, M.N., Singh, M.P., 1997. Ontologies for agents. Internet Comput. IEEE 6, 81–83.
King, G., Feb. 2011. Ensuring the data-rich future of the social sciences. Science (New York, N.Y.) 331 (6018), 719–721.
Kogan, F., Powell, A., Fedorov, O., 2010. Use of Satellite and In-Situ Data to Improve Sustainability. Springer.
Lake, R., Nov. 2005. The application of geography markup language (GML) to the geological sciences. Comput. Geosci. 31 (9), 1081–1094.
Lamprecht, A.-L., 2013. User-Level Workflow Design. In: Lecture Notes in Computer Science, 8311. Springer Berlin Heidelberg, Berlin, Heidelberg. http://link.springer.com/10.1007/978-3-642-45389-2.
Laniak, G.F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., Hughes, A., Jan. 2013. Integrated environmental modeling: a vision and roadmap for the future. Environ. Model. Softw. 39, 3–23.
Lautenbacher, C.C.J., 2005. The global earth observation system of systems (GEOSS). In: IEEE International Symposium on Mass Storage Systems and Technology.
Leadbetter, A.M., Lowry, R.K., Clements, D.O., 2013. Putting meaning into NETMAR the open service network for marine environmental data. Int. J. Digital Earth 1–18.
Lehnert, K.A., Carlson, R., Hofmann, A., Langmuir, C.H., Lenhardt, W.C., Sarbas, B., Walker, D., Glazner, A., Farmer, L., Sep. 2003. Earthchem.org: integrating data management for igneous geochemistry. Geol. Soc. Am. Abstr. Programs 35 (6), 366.
Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y., 2006. Scientific workflow management and the Kepler system. Concurrency Comput. Pract. Exp. 18 (10), 1039–1065.
Maidment, D.R., Hooper, R.P., Tarboton, D.G., Zaslavsky, I., 2009. Accessing and sharing data using CUAHSI Water Data Services. In: Symposium JS.4 at the Joint Convention of the International Association of Hydrological Sciences, IAHS and the International Association of Hydrogeologists, IAH, vol. 331. IAHS-AISH Publication, pp. 213–223.
Manola, F., Miller, E., McBride, B., 2004. RDF primer. W3C Recomm. 10, 1–107.
McAfee, A., Brynjolfsson, E., 2012. Big data: the management revolution. Harv. Bus. Rev. 90 (10), 60–6, 68, 128.
McGuinness, D.L., Van Harmelen, F., 2004. OWL web ontology language overview. W3C Recomm. 10, 1–22.
McIntosh, B.S., Ascough, J.C., Twery, M., Chew, J., Elmahdi, A., Haase, D., Harou, J.J., Hepting, D., Cuddy, S.M., Jakeman, A.J., Chen, S., Kassahun, A., Lautenbach, S., Matthews, K., Merritt, W., Quinn, N.W.T., Rodriguez-Roda, I., Sieber, S., Stavenga, M., Sulis, A., Ticehurst, J., Volk, M., Wrobel, M., van Delden, H., El-Sawah, S., Rizzoli, A., Voinov, A., Dec. 2011. Environmental decision support systems (EDSS) development challenges and best practices. Environ. Model. Softw. 26 (12), 1389–1402.
Merrin, L.E., Cuddy, S.M., 2009. Implementation of a reporting workflow to maintain data lineage for major water resource modelling projects. In: 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, Cairns, Australia, pp. 4367–4373. http://mssanz.org.au/modsim09/J4/merrin.pdf.
Michaelis, C.D., Ames, D.P., Apr. 2009. Evaluation and implementation of the OGC web processing service for use in client-side GIS. GeoInformatica 13 (1), 109–120.
Michener, W., Vieglais, D., Vision, T., Kunze, J., Cruse, P., Janée, G., Jan. 2011. DataONE: data observation network for earth – preserving data and enabling innovation in the biological and environmental sciences. D-Lib Mag. 17 (1/2).
Nativi, S., Caron, J., Davis, E., Domenico, B., Nov. 2005. Design and implementation of netCDF markup language (NcML) and its GML-based extension (NcML-GML). Comput. Geosci. 31 (9), 1104–1118.
Nielsen, M., 2011. Reinventing Discovery: the New Era of Networked Science. Princeton University Press.
Niu, X., Lehnert, K.A., Williams, J., Brantley, S.L., Jun. 2011. CZChemDB and EarthChem: advancing management and access of critical zone geochemical data. Appl. Geochem. 26, S108–S111.
Overpeck, J.T., Meehl, G.A., Bony, S., Easterling, D.R., Feb. 2011. Climate data challenges in the 21st century. Science (New York, N.Y.) 331 (6018), 700–702.
O'Sullivan, B., 2009. Mercurial: the Definitive Guide. In: Definitive Guide Series, vol. 7. O'Reilly Media, Inc.
Pollock, R., 2013a. Forget Big Data, Small Data is the Real Revolution — Open Knowledge Foundation Blog. http://blog.okfn.org/2013/04/22/forget-big-data-small-data-is-the-real-revolution/.
Pollock, R., 2013b. Git (and Github) for Data — Open Knowledge Foundation Blog. http://blog.okfn.org/2013/07/02/git-and-github-for-data/.
Reichman, O.J., Jones, M.B., Schildhauer, M.P., Feb. 2011. Challenges and opportunities of open data in ecology. Science (New York, N.Y.) 331 (6018), 703–705.
Roberts, T., Feb. 2012. The Problem with Open Data. Computer Weekly. http://www.computerweekly.com/opinion/The-problem-with-Open-Data.
Rusher, J., 2008. TripleStore. Semantic Web Advanced Development for Europe (SWAD-Europe). http://www.w3.org/2001/sw/Europe/.
Sellars, S., Nguyen, P., Chu, W., Gao, X., Hsu, K., Sorooshian, S., Aug. 2013. Computational earth science: big data transformed into insight. Eos Trans. Am. Geophys. Union 94 (32), 277–278.
Sikder, I.U., Dec. 2008. Geospatial Web Services in environmental planning. In: 2008 11th International Conference on Computer and Information Technology. IEEE, pp. 424–429.
Sivasubramanian, S., 2012. Amazon dynamoDB: a seamlessly scalable non-relational database service. In: Proceedings of the 2012 International Conference on Management of Data. SIGMOD'12. ACM, pp. 729–730.
Spinellis, D., May 2012. Git. IEEE Softw. 29 (3), 100–101.
Steed, C.A., Ricciuto, D.M., Shipman, G., Smith, B., Thornton, P.E., Wang, D., Shi, X., Williams, D.N., 2013. Big data visual analytics for exploratory earth system simulation analysis. Comput. Geosci. 61, 71–82.
Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C., 2007. Model-Driven development with the jABC. In: Bin, E., Ziv, A., Ur, S. (Eds.), Hardware and Software, Verification and Testing, Lecture Notes in Computer Science, vol. 4383. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 92–108.
Tan, W., Missier, P., Foster, I., Madduri, R., Goble, C., Jun. 2010. A Comparison of using Taverna and BPEL in building scientific workflows: the case of caGrid. Concurrency Comput. Pract. Exp. 22 (9), 1098–1117.
Taylor, K.E., Stouffer, R.J., Meehl, G.A., Apr. 2012. An overview of CMIP5 and the experiment design. Bull. Am. Meteorol. Soc. 93 (4), 485–498.
The MPI Forum, 1993. MPI: a message passing interface. In: Proceedings of the Conference on High Performance Networking and Computing, pp. 878–883.
Thorpe, A., Mar. 2009. Environmental eScience. Philos. Trans. Ser. A Math. Phys. Eng. Sci. 367 (1890), 801–802.
Tilmes, C., Yesha, Y., Halem, M., Apr. 2010. Tracking provenance of earth science data. Earth Sci. Inf. 3 (1–2), 59–65.
Tsou, M., Guo, L., Stow, D., 2003. Web-based remote sensing applications and java tools for environmental monitoring. Online J. Space Commun. 3. http://spacejournal.ohio.edu/issue3/abst_tsou.html.
Valentine, D., Zaslavsky, I., 2009. CUAHSI WATERML 1.1-Specification. CUAHSI Overview document. Tech. rep., CUAHSI. http://his.cuahsi.org/wofws.html\#waterml.
van Koningsveld, M., de Boer, G.J., Baart, F., Damsma, T., den Heijer, C., van Geer, P., De Sonnevile, B., de Sonneville, B., 2010. OPENEARTH – inter-company management of: data, models, tools and knowledge. In: WODCON XIX: Dredging Makes the World a Better Place. World Organization of Dredging Associations, 14 pp. http://repository.tudelft.nl/view/ir/uuid:87e0b19a-b6c9-4761-a17c-91f525577499/.
Villa, F., Athanasiadis, I.N., Rizzoli, A.E., May 2009. Modelling with knowledge: a review of emerging semantic approaches to environmental modelling. Environ. Model. Softw. 24 (5), 577–587.
Weiser, A., Zipf, A., 2007. Web service orchestration of OGC web services for disaster management. In: Li, J., Zlatanova, S., Fabbri, A.G. (Eds.), Geomatics Solutions for Disaster Management. Springer Berlin Heidelberg, pp. 239–254. Ch. Lecture No.
White, T., 2010. Hadoop: the Definitive Guide. In: Definitive Guide Series, vol. 54. Yahoo Press.
Wiley, K., Connolly, A., Gardner, J.P., Krughof, S., Balazinska, M., Howe, B., Kwon, Y.C., Bu, Y., Oct. 2010. Astronomy in the Cloud: Using MapReduce for Image Coaddition. CoRR abs/1010.1.
Williams, M., Cornford, D., Bastin, L., Pebesma, E., 2008. Uncertainty Markup Language (UncertML). Tech. rep., Open Geospatial Consortium. http://www.opengeospatial.org/node/1002.
Xiang, P., Hou, R., Jul. 2010. Cache and consistency in NOSQL. In: 2010 3rd International Conference on Computer Science and Information Technology. IEEE, pp. 117–120.
Xu, G., Wang, Z., Jun. 2010. Data Provenance Architecture Based on Semantic Web Services. Tech. rep.
Yao, Y.Y., Ohsuga, S., 2000. Web Intelligence (WI). In: Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000. IEEE Comput. Soc, pp. 469–470.
Yu, J., Taylor, P., Cox, S., Walker, G., 2011. Validating water resources described in WaterML 2. 0. Geophys. Res. Abstr. 13.
Yu, G., Zhao, P., Di, L., Chen, A., Deng, M., Bai, Y., Oct. 2012. BPELPower – a BPEL execution engine for geospatial web services. Comput. Geosci. 47, 87–101.
Zhang, C., De Sterck, H., Aboulnaga, A., Djambazian, H., Sladek, R., 2010. Case study of scientific data processing on a cloud using hadoop. In: High Performance Computing Systems and Applications: Revised Selected Papers from the 23rd International High Performance Computing Symposium (HPCS'09), Kingston, Ontario, Canada, June 14–17, 2009 5976, pp. 400–416.