Modelling provenance in hydrologic science: a case study on streamflow forecasting

Yanfeng Shu, Kerry Taylor, Prasantha Hapuarachchi and Chris Peters

ABSTRACT

The web, and more recently the concept and technology of the Semantic Web, has created a wealth of new ideas and innovative tools for data management, integration and computation in an open framework and at a very large scale. One area of particular interest to the science of hydrology is the capture, representation, inference and presentation of provenance information: information that helps to explain how data were computed and how they should be interpreted. This paper is among the first to bring recent developments in the management of provenance developed for e-science and the Semantic Web to the problems of hydrology. Our main result is a formal ontological model for the representation of provenance information driven by a hydrologic case study. Along the way, we support usability, extensibility and reusability for provenance representation, relying on the concept of modelling both domain-independent and domain-specific aspects of provenance. We evaluate our model with respect to its ability to satisfy identified requirements arising from the case study on streamflow forecasting for the South Esk River catchment in Tasmania, Australia.

Key words | domain semantics, ontologies, provenance modelling, streamflow forecasting

Yanfeng Shu (corresponding author)
Chris Peters
CSIRO Tasmanian ICT Centre,
GPO Box 1538,
Hobart, TAS 7001,
Australia
E-mail: *yanfeng.shu@csiro.au*

Kerry Taylor
CSIRO ICT Centre,
GPO Box 664,
Canberra, ACT 2601,
Australia
and
College of Engineering and Computer Science,
Australian National University,
Canberra, ACT 0200,
Australia

Prasantha Hapuarachchi
CSIRO Land and Water,
Graham Road,
Highett, VIC 3190,
Australia

INTRODUCTION

Hydrology is the study of the occurrence, distribution and movement of water on, in and above the Earth (see http://www.economicexpert.com/a/Hydrology.htm). Simulating a hydrologic phenomenon can be complex. In addition to the challenges of representing physical dynamics, it may involve coupling multiple models and accessing multiple data sources. For example, water quality management routinely requires the coupling of multiple models to reflect the structure of the water flow through upper catchments, freshwater streams and tidal river estuaries (Taylor *et al*. 1999). In a recent major exercise to quantify water availability of the Murray Darling Basin in Australia, a large variety of models were used: climate models, catchment water yield models, operational river system models, groundwater models and water accounting models (CSIRO 2007).

It is increasingly important to capture the provenance of simulation results developed this way. It is necessary for the purpose of the interpretation of results, especially as quantifying uncertainty in results in such a setting is very difficult. The authority and method by which a result was computed is a proxy for formal quality evaluation and can engender the appropriate level of trust in data. In some cases in Australia, results such as these are used for major, controversial, economic and environmental policy decisions which must be supported, and be seen to be supported, by the best science available. In addition, provenance is important for the application of the basic principles of scientific transparency and scientific knowledge evolution: enabling the capture, reproduction and improvement of best practices.

A recent incubator group of the World Wide Web Consortium (W3C, see http://www.w3.org/2005/Incubator/prov/) defines provenance as follows:

> 'Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource. Provenance provides a critical foundation for assessing authenticity, enabling trust, and allowing reproducibility. Provenance

doi: 10.2166/hydro.2012.134

assertions are a form of contextual metadata and can themselves become important records with their own provenance.'

For hydrology and its related disciplines, provenance, as a kind of metadata, can describe the original sources of data, data manipulations such as filtering and interpolation, the sources of models, the flow of data between models, the intermediate data products, the calibration of model parameters, and the purpose, destination, authority and quality estimates for any final data products.

One fundamental issue for provenance is its representation. A representation needs to be rich enough to support both domain-independent and domain-specific retrieval of the captured provenance knowledge. For example, a domain-independent question might ask for all the steps taken in a workflow that produced a named data file. A domain-specific question might ask for all the values of calibration parameters that have been used by any rainfall–runoff model applied over the upper sub-catchments of the Molongolo River during the summer of 2001. Domain-independent questions correspond to a low-level view of provenance tasks that users have at hand; answering such questions requires causal relationships between data products to be captured as provenance. Domain-specific questions, on the other hand, represent a high-level view of provenance tasks; answering such questions requires provenance to be enriched with domain semantics.

In this paper, we contribute to the design of a representation aimed at answering both types of provenance questions. We develop our representation in the context of three guiding principles which are essential to any exercise in provenance capture and representation. These principles are as follows:

- A representation should be use case driven. Use cases play an important role in provenance representation. It is through use cases that we know what provenance requirements are; by examining the requirements, we are then able to identify what needs to be captured and represented, and at what level of granularity, in order to satisfy the requirements.
- A representation should be usable. One primary goal of a representation is to facilitate the use of provenance. This in turn requires the representation to be able to support querying and, we argue, support reasoning over provenance. The usability of a representation is also reflected in its interoperability with other provenance representations.
- A representation should be extensible. Use cases can never be exhaustive: there are always cases not examined previously. The important thing is to be able to discover generic provenance requirements from the use cases at hand, to meet these requirements, and to make a representation reusable for other possible use cases. A good representation will support extensibility to expand the scope of the domain and artifacts being represented, and also to expand the detail and level of granularity being represented, without forcing loss of access to prior provenance data.

These three principles serve as our guidelines for deciding what to represent and how to represent it. In following these principles, we start from a use case study. We investigate provenance requirements for streamflow forecasting, including what kind of provenance information to be provided, and at what level of granularity, in order to help users assess the quality of a streamflow forecasting result.

Having determined what to represent, we next decide how to represent it. We address this from the following aspects. First, we employ a modular approach to provenance modelling, which decouples domain-independent provenance from domain-specific provenance, and application-general provenance from use-case-specific provenance. As such, we support extensibility and reusability for provenance representation. Second, we model provenance by extending the widely used Open Provenance Model (OPM) (Moreau *et al*. 2010), which promotes interoperability with other provenance representations. Third, we encode provenance in the formal logic-based Web Ontology Language (OWL, http://www.w3.org/TR/owl-features/), which not only enables expressive provenance representation, but also facilitates querying and reasoning over provenance.

Finally we evaluate our provenance representation with respect to the requirements established by the use case.

In summary, the contributions of this paper are as follows:

1. An analysis of provenance requirements for streamflow forecasting.
2. A representation of provenance information in streamflow forecasting with support for usability, extensibility and reusability principles. This is the first time these principles have been comprehensively applied and tested in the context of hydrologic science.
3. An evaluation of the representation with respect to its ability to meet the requirements identified from the use case.

The remainder of this paper is organised as follows. The next section gives a review of related work. The third section introduces the use case. The fourth section discusses provenance requirements for streamflow forecasting. Then we present the provenance model, followed by an evaluation in the sixth section. Finally, we end the paper with our conclusions.

RELATED WORK

Provenance has been studied in a number of domains, e.g. physics (Foster *et al.* 2002), earth sciences (Frew & Bose 2001), bioinformatics (Greenwood *et al.* 2003) and computer science (Davidson *et al.* 2007; Tan 2007). A comprehensive survey of provenance for scientific data computing is given by Bose & Frew (2005), which covers lineage-related standards, lineage research and prototypes. Simmhan *et al.* (2005) describe a taxonomy to categorise provenance systems along provenance usage, subject, representation, storage and dissemination dimensions. The requirements of recording, querying and using provenance in e-science experiments is defined by Miles *et al.* (2006), based on an analysis of a range of use cases from several domains.

A key component of a provenance system is provenance representation. There are two major representation approaches (Simmhan *et al.* 2005): inversion and annotation. The inversion approach relies on inversion functions to find the derivation history of a data product (Woodruff & Stonebraker 1997; Cui & Widom 2000). The annotation approach, on the other hand, treats provenance as metadata and pre-computes it. Our work follows the annotation approach. Several provenance annotation models have been proposed in the literature. The Proof Markup language (PML) (da Silva *et al.* 2006) focuses on modelling provenance in reasoning systems and covers the concepts for describing conclusions and justifications. Provenir (http://wiki.knoesis.org/index.php/Provenir_Ontology) is built based on the OBO Relation Ontology (RO) (Smith *et al.* 2005) and has been extended for modelling provenance in oceanography (Sahoo *et al.* 2010) and biology experiments (Sahoo *et al.* 2009; Missier *et al.* 2010). OPM, which our work is based on, is currently the most widely used. It results from a community effort to achieve interoperability of workflow systems, and its recent applications include reproducing scientific results (Moreau 2011), representing scientists' intent (i.e. experimental constraints and goals) (Pignotti *et al.* 2011), tracking provenance in distributed systems (Groth & Moreau 2011) and modelling provenance on the web (Freitas *et al.* 2011). Common to PML, Provenir, OPM and some other models is the representation of process and data dependences. The W3C Provenance Working Group (http://www.w3.org/2011/prov/), which involves various communities with interests in the provenance space, is currently defining a language for exchanging provenance information among applications. The importance of semantic provenance for e-science is explicitly pointed out by Sahoo *et al.* (2008), who define semantic provenance as 'information created with reference to a formal knowledge model or an ontology that imposes a domain-specific provenance view on scientific data'. Along this line, Zhao *et al.* (2011) further classify provenance information into three types: simple provenance traces with no domain-specific semantics, semantic provenance and provenance traces that comply with the Linked Data standard (see http://www.w3.org/standards/semanticweb/data).

In hydrologic science, the only provenance work we are aware of is by Dozier & Frew (2009), where a snow mapping example is given. In the example, scientists first compute fractional snow cover maps from daily satellite observations and then filter, smooth and interpolate the maps to provide the best estimate of the daily snow cover. To capture and manage provenance in the mapping process, a software environment, the Earth System Science Server (ES^3) (Frew *et al.* 2008), is used, where provenance is modelled as a direct graph of processes and their input and output files. Our work complements Dozier & Frew's work in that we

model not only domain-independent aspects of provenance but also *domain-specific* aspects, which allows us to support a wider range of provenance inference tasks; also, by building our work on OPM, we facilitate our representation's interoperability with other provenance models.

CASE STUDY: STREAMFLOW FORECASTING

The South Esk River catchment (Figure 1) (DIPIW 2009) is located in the northeast and midlands of Tasmania, Australia, and extends over an area of approximately 3,350 km^2. The catchment rises in the Fingal Tier in the east and is bounded by the Ben Lomond Range and Mt Saddleback to the north. Its major tributaries are the Nile, St Pauls and Break O'Day rivers. Downstream of Longford, the South Esk River receives inflow from the Macquarie and Meander rivers and flows into the Tamar Estuary. The river system (above its confluence with the Macquarie River) is largely unregulated, and has the key characteristics of a natural flow regime, including seasonal distribution and variability in flows, and natural rates of rise and fall in river height. The primary land uses in the catchment are agriculture and forestry.

The catchment is covered by 19 rain gauges, 11 streamflow gauges and 10 automatic weather stations (AWS) to collect observations of rainfall, streamflow (or stage) and climate data such as air temperature and wind speed. The gauges and stations are operated by several organisations, including the Bureau of Meteorology (BoM), Hydro Tasmania, Forestry Tasmania, CSIRO and the Department of Primary Industries, Parks, Water and Environment (DPIPWE).

To provide near-real-time sensor observations, we have developed a hydrologic sensor web application (http://www.csiro.au/sensorweb/au.csiro.OgcThinClient/OgcThinClient.

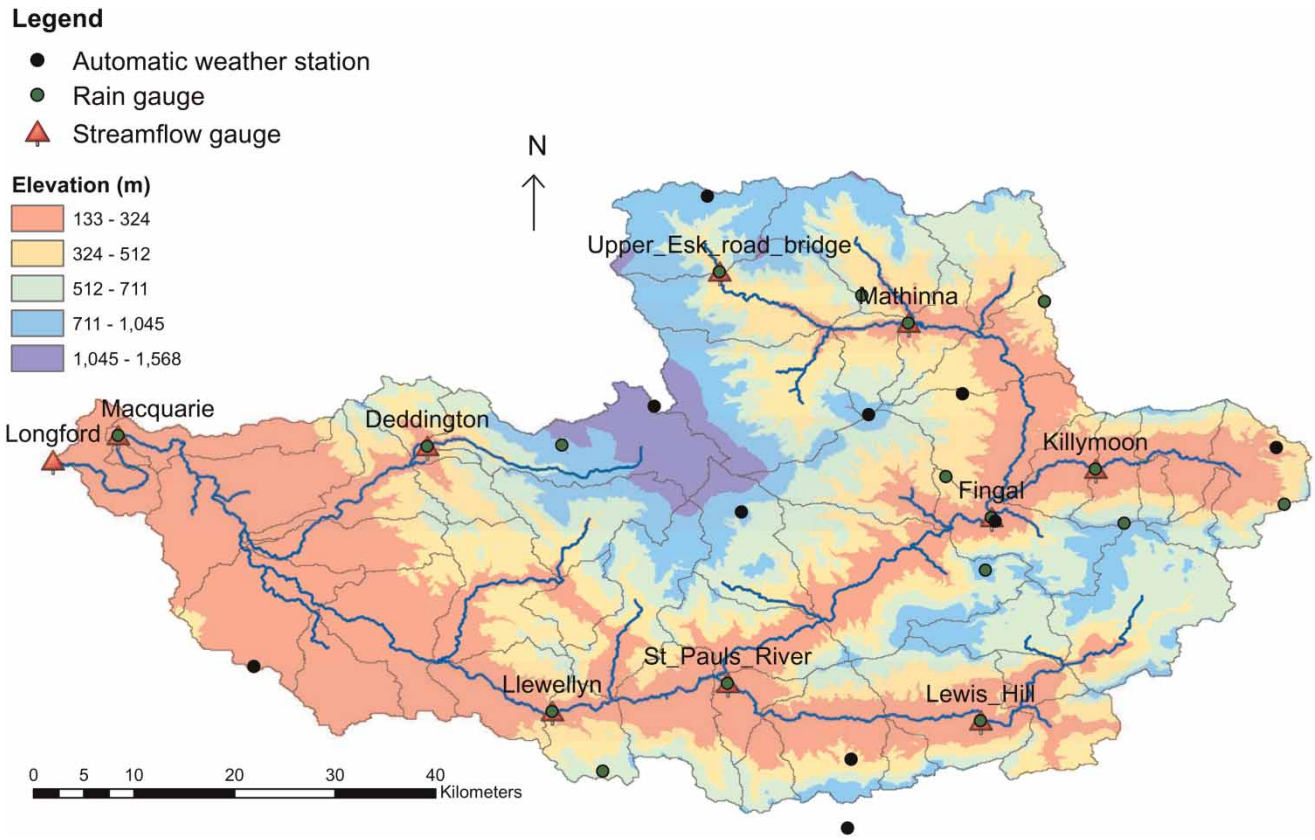

Figure 1 | The South Esk River catchment.

html). As part of the application, we also provide streamflow forecasts for the catchment for a period of up to 3 days by using a rainfall–runoff model, GR4H (Mathevet 2005) (an hourly version of GR4J (Perrin *et al.* 2003)). The model is configured to run in semi-distributed mode with Muskingum channel routing. It uses hourly spatially interpolated rainfall, hourly observed streamflow and monthly potential evaporation (ET) data. Rainfall data come from two types of source – observations of rain gauges and weather stations, and 72-h forecasts of a numerical weather prediction (NWP) model, the Conformal-Cubic Atmospheric Model (CCAM) (McGregor & Dix 2008) – and are interpolated on a 1 km × 1 km grid using the simple kriging method. ET data come from the Australian Water Availability Project (AWAP: http://www.csiro.au/awap/). In the study, we use data from 1995–2010 for model calibration.

Figure 2 shows the forecasting process for the South Esk River catchment, which involves the following major steps:

- **Temporal normalisation:** aggregation or extrapolation of data at a specific temporal scale.
- **Spatial normalisation:** interpolation of data at a specific spatial scale.
- **Model calibration:** calibration of model parameters using the shuffled complex evolution optimisation method (SCE-UA) (Duan *et al.* 1992).
- **Model simulation:** simulation of streamflow by first running the calibrated model for a certain period (i.e. warm-up period), and then using the model states (e.g. soil moisture, groundwater and channel storages) at the end of the warm-up period as the model's initial condition with rainfall forecasts to simulate streamflow.

PROVENANCE REQUIREMENTS

Given a case such as the above, and a streamflow forecasting result, we are interested to know: what kind of information, if provided, can help users assess the quality of the result? To answer this question, we require a clear understanding of what quality is. There are a number of definitions of quality, and the most frequently adopted one is 'fitness for use' (Juran *et al.* 1974). Based on this definition, quality is usually described by a set of quality dimensions which represent desirable characteristics for an information resource. For example, Wang & Strong (1996) describe quality along 15 dimensions including accuracy, interpretability and reputation.

There is no 'one size fits all' set of quality dimensions. For assessing the quality of a forecasting result, accuracy is one major consideration. The accuracy of a forecasting result, i.e. the extent to which the forecasts are close to the observations, is affected by uncertainties in hydrologic modelling and forecasting. Common sources of uncertainty include model structure, parameters, and input and observed output data (Butts *et al.* 2004). Model structural uncertainty arises from the inability of a model to truly represent a hydrologic system; parameter uncertainty is caused by the so-called equifinality (Beven 1993), i.e. the

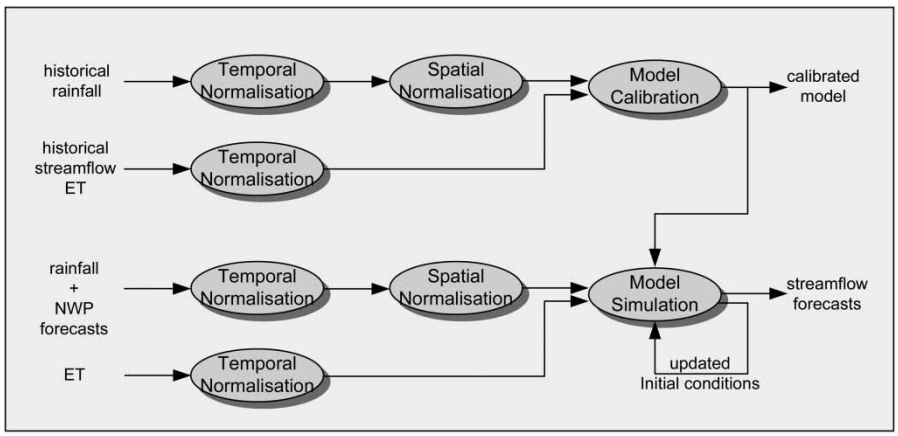

Figure 2 | Streamflow forecasting for the South Esk River catchment.

nonuniqueness or nonidentifiability of parameters in the feasible parameter space; and data uncertainty is due to measurement errors (including both systematic and random errors), limited spatial and temporal sampling, or errors introduced by data processing (e.g. spatial interpolation).

To facilitate accuracy assessment, information on these uncertainty sources needs to be provided. With such information, users can then determine, based on their knowledge, which uncertainties should be accounted for, and which can be neglected without affecting forecast accuracy. In general, the following information is of interest:

- **The hydrologic model used**. This includes its version, processes, inputs, outputs, states, (free) parameters, spatial treatment of inputs and parameters (i.e. lumped, distributed or semi-distributed) and routing schemes.
- **The hydrologic data used and generated.** Examples of data are forcing data (e.g. precipitation – generally regarded as a significant source of uncertainty in hydrologic models (Milly *et al*. 2005), potential evaporation), catchment physical characteristics (e.g. area, elevation and channel lengths, slope), observations of model states or fluxes, and model output. For easy interpretation of data, contextual information of data, e.g. forecast lead-time and temporal resolution, may be provided as well.
- **The sources and providers of hydrologic data.** Examples of data sources are rain gauges, weather stations, satellites and NWP models. The quality of data is greatly influenced by their origin. For example, errors in rainfall observations are always associated with gauge type, accuracy and calibration. Data may be quality-checked by providers. Most quality-checked datasets include flags to indicate the accuracy of a particular observation. Flags are used to indicate missing values, automated filtering errors, uncommon situations like ice or hail, and whether data are observed, interpolated or derived using a rating table.
- **The steps involved, including the methods and configurations used, and the agents who performed the steps.** Hydrological modelling and forecasting can be considered to comprise four high-level steps: a *pre-processing* step to prepare data before use in the model, a *model calibration* step to estimate model parameter values, a *model simulation* step to run the model and generate forecasts, and a *post-processing* step to process forecasts for end use (in this paper, we focus on batch model calibration, i.e. using batch of data for calibration (Moradkhani & Sorooshian 2009)). Depending on use case, each of these steps may consist of a set of sub-steps. For example, a pre-processing step may have sub-steps such as bias correction, temporal aggregation and spatial interpolation. A step may use a different method, have a different configuration or be performed by a different agent. For example, the method used for model calibration can be manual or automatic; the calibration period can range from a few years to more than ten years; and the calibration can be performed by people with low or high levels of expertise.

The granularity of information to be provided, in particular the granularity of process and data information, varies from use case to use case. In the case of South Esk streamflow forecasting, information on each individual step is needed. This includes information on both temporal and spatial normalisation steps. Although these two steps could be considered at a coarse level as a single data pre-processing step, information of the method used in each step is important for the assessment of the quality of the data (to be used as model input). Besides the methods, the inputs and outputs of each step need to be considered as well, which can be as simple as a single value, e.g. the temporal scale used by temporal normalisation, or as complex as a large set of values in the form of files, e.g. rainfall data. For the latter case, we treat the dataset as a whole (instead of individual data values) and focus more on common features of data which are typically looked at, e.g. the time period and timestep of a time series, or the location of a data file (this also avoids representing provenance of individual data values, which possibly requires huge provenance storage).

The requirements for the information to be provided are also reflected in the provenance questions that can be asked. There are two types of provenance question: domain-specific and domain-independent, and the difference between them is whether or not domain semantics are needed to answer questions. A domain-specific question could be: 'Find all *data preprocessing* steps involved that contribute to the computation of a *streamflow forecasting*

result' and its domain-independent version may be: 'Find all steps involved that contribute to the computation of a result.' Provenance information enriched with domain semantics can be used to answer both types of question. For the South Esk use case, example provenance questions can be, given a forecasting result:

- **Q1:** what's the lead time of the result and at what temporal resolution?
- **Q2:** which hydrologic model was used?
- **Q3:** who calibrated the model and which calibration method was used?
- **Q4:** what data were used as model input in the simulation?
- **Q5:** which preprocessing steps and methods were performed for a certain model input?
- **Q6:** for a gauge involved, when was it last calibrated?

THE PROVENANCE MODEL

From the requirements, we derive a set of key concepts that should be covered by a provenance model. We construct the model with a modular approach by decoupling domain-independent provenance from domain-specific provenance, and application-general provenance from use-case-specific provenance. Such an approach enables part of the model to be able to be reused. Three modules are generated: a basic module for describing the concepts independent of any particular domain, an application module for the concepts common to streamflow forecasting applications and a use case module for the concepts specific to a forecasting case (e.g. the South Esk case), among which the application and use case modules together describe domain-specific provenance.

We represent the model using the Web Ontology Language (OWL). OWL is expressive enough to provide precise description of the provenance information that needs to be represented. Also, it supports querying and reasoning over provenance. The reasoning enables a relatively compact representation (because we can rely on inference to infer data that are not represented), and is also critical for managing the relationships between domain-specific and domain-independent parts of our representation.

Representing domain-independent provenance

Regardless of domain, some concepts are fundamental, underlying provenance requirements of any application. For example, those describing:

- the steps involved and associated methods;
- the artifacts used or generated by a step;
- the agents which interact with a step or an artifact;
- relationships between steps, artifacts and agents, e.g. an artifact was generated by a step.

To represent these concepts, we leverage existing work on provenance modelling. In this paper, we use the Open Provenance Model (OPM) (Moreau *et al.* 2010).

Throughout this section, we use the terms in italic type to denote the concepts defined in OPM and the terms in teletype to denote OWL classes (with the first letter in uppercase) or properties (with the first letter in lowercase). OPM is an abstract model which provides a specification to express provenance information. It defines provenance as a directed graph, whose nodes are:

- *artifact*: immutable piece of state, which may have a physical embodiment in a physical object, or a digital representation in a computer system;
- *process*: action or series of actions performed on or caused by artifacts, and resulting in new artifacts;
- *agent*: contextual entity as a catalyst of a process, enabling, facilitating, controlling or affecting its execution;

and whose edges are:

used (from a process to an artifact): a causal relationship intended to indicate that the process required the availability of the artifact to be able to complete its execution;

wasGeneratedBy (from an artifact to a process): a causal relationship intended to mean that the process was required to initiate its execution for the artifact to have been generated;

wasTriggeredBy (from process P_2 to process P_1): a causal dependence that indicates that the start of P_1 was required for P_2 to be able to complete;

wasDerivedFrom (from artifact A_2 to artifact A_1): a causal relationship that indicates that A_1 needs to have been generated for A_2 to be generated;

wasControlledBy (from a process to an agent): a causal dependence that indicates that the start and end of the process was controlled by the agent.

Besides nodes and edges, there are some other concepts defined in OPM, such as *roles* and *accounts*. Roles are used to 'designate an artifact's or agent's function in a process', and accounts to 'represent a description at some level of detail as provided by one or more observers' (Moreau *et al.* 2010). Nodes and edges can belong to accounts; together they represent, from an observer's point of view, 'how 'things', whether digital data, physical objects or immaterial entities, came to be in a given state, with a given set of characteristics, at a given moment' (Moreau *et al.* 2010). Suppose a workflow consists of two steps *P1* and *P2*; an execution of the workflow by *A* transformed data *a1* into *a2*. Then, an *account* of the execution can include: *artifacts a1* and *a2*, *processes P1* and *P2* (controlled by `agent` *A*), *P1 used a1*, *a2 wasGeneratedBy P2*, *P2 wasTriggeredBy P1* and *a2 wasDerivedFrom a1*.

OPM also defines a set of one-step (or completion) and multi-step inference rules that can be applied to a provenance graph. An example of one-step inferences is that a *wasTriggeredBy* relationship can be inferred from the existence of *used* and *wasGeneratedBy* relationships. Multi-step inferences are associated with multi-step versions of existing relationships, i.e. *used**, *wasGeneratedBy**, *wasTriggeredBy** and *wasDerivedFrom**. They are used to find the causes of an artifact or a process which possibly involve multiple transitions. For example, process *P used** artifact *a* if *P* used an artifact that was *a* or was derived from *a* possibly using multiple steps.

OPMO (http://openprovenance.org/model/opmo) is an OWL-based encoding of the latest specification of OPM (i.e. v1.1). It maps OPM nodes and edges to OWL classes. The mapping of edges to classes allows additional edge properties to be expressed, such as time and role information. For example, used relationships are represented in OPMO by class `Used`, whose instances have and only have `Processes` as `effect` and `Artifacts` as `cause`, and have at least one `Role`. OPM inference rules are partly expressed in OPMO: `wasDerivedFromStar` (corresponding to *wasDerivedFrom** in OPM) is defined as a transitive property in OWL, with `wasDerivedFrom` as its subproperty (note that OPMO defines both WasDerived-From (WDF) and wasDerivedFrom, with the former as an OWL class and the latter as an OWL property); other multi-step relationships are defined based on `wasDerivedFrom*`, and can be inferred by OWL by means of property chains (as such, we can choose not to instantiate these relationships). For example, `usedStar` can be inferred from `used` and `wasDerivedFromStar`, expressed in Description Logics (DL) (Baader *et al.* 2003) as follows (Table 1 shows the DL notations used in ths paper. Please refer to Baader *et al.* (2003) for their formal semantic interpretations):

$$\texttt{used} \circ \texttt{wasDerivedFromStar} \sqsubseteq \texttt{usedStar}$$

We further extend OPMO to cover the following generic and domain-independent concepts and properties (the resulting ontology is OPMO+, shown in Figure 3):

- `Method` and `useMethod`: a method describes how a process was performed, as defined by $Process \sqsubseteq = 1\ useMethod.Method$.
- `partOf`: an artifact (resp. a process or an agent) can be part of another artifact (resp. process or agent). We define `partOf` as transitive and specify both its domain and range to be `Artifact`, or `Process`, or `Agent`.
- `providedBy`: an artifact may be provided by an agent, defined by

$$Artifact \sqsubseteq \forall\ providedBy.Agent$$

- `Source` and `hasSource`: an artifact may come with source information (e.g. sensor), defined by $Artifact \sqsubseteq \forall\ hasSource.Source$.

These extensions make it possible to represent such information as the method used in a process, the source of an artifact, whether an artifact is part of another artifact,

Table 1 | The DL notations used in the paper (*C* and *D* are concept descriptions, *R* and *S* are roles or properties, and *n* is a number)

Symbol syntax	Description
$C \sqsubseteq D$	Inclusion
$C \sqcap D$	Intersection
$\forall R.\ C$	Universal value restriction
$=nR.\ C$	Quantified (exact) number restriction
$R \circ S$	Composition

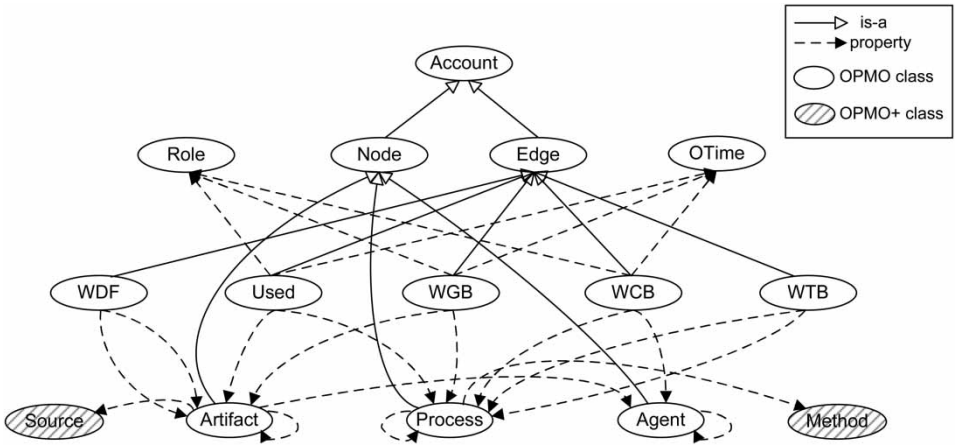

Figure 3 | OPMO + : the domain-independent module.

and who provided the artifact, and thus to answer related provenance questions.

Representing domain-specific provenance

To represent domain-specific provenance, we extend OPMO+. We first create an ontology for describing provenance information in streamflow forecasting in general, which results in FFPO (i.e. the flow forecasting provenance ontology, see Figure 4). FFPO is specific to streamflow forecasting applications in the hydrologic domain. It is designed as an upper provenance ontology for streamflow forecasting, that is, it covers general concepts of streamflow forecasting (e.g. `Calibration`, `HydroModel` and `GlobalOptimisation`) and can be extended to express provenance specific to a use case. In the following, we first describe how we construct FFPO, then we use the South Esk use case as an example to illustrate how we extend FFPO to generate SEPO (i.e. the South Esk provenance ontology, see Figure 5).

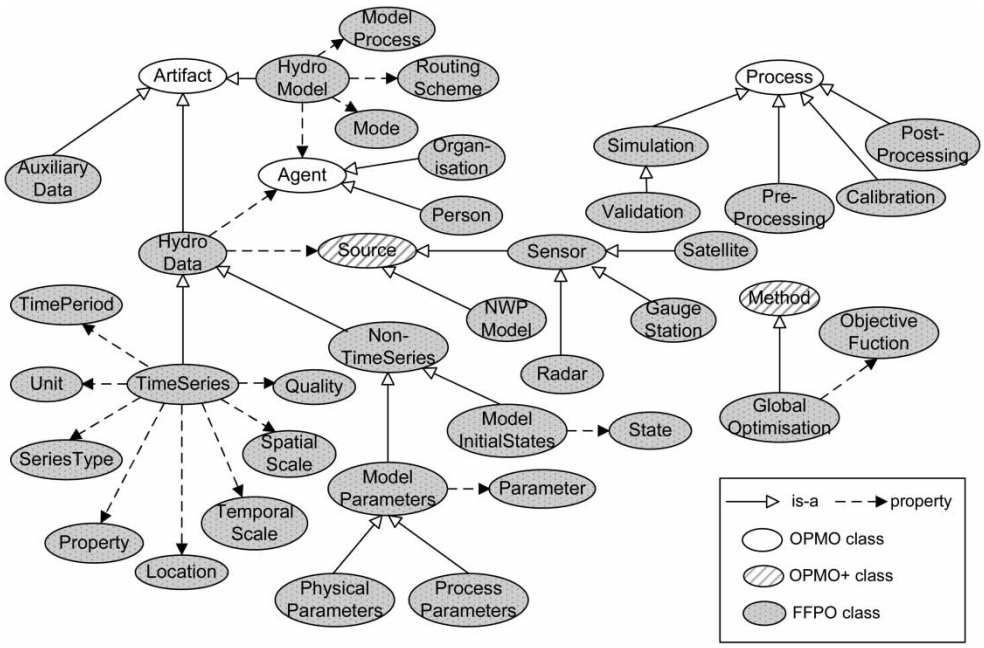

Figure 4 | FFPO: the flow forecasting module.

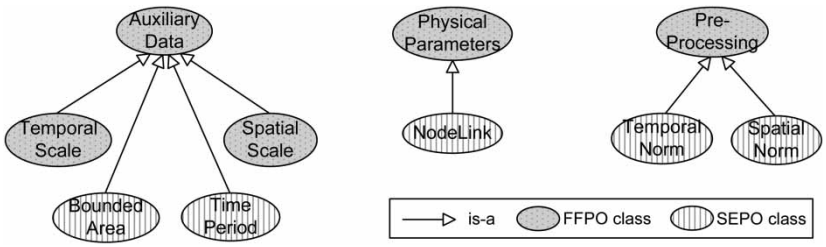

Figure 5 | SEPO: the South Esk use case module.

For FFPO, we first decide the concepts for describing the steps involved in streamflow forecasting. There are four high-level steps: data pre-processing, model calibration, model simulation and forecast post-processing. The calibration and simulation steps are always central to hydrologic modelling. However, before a model run available data typically needs to be transformed for input (i.e. pre-processing) and then, after a model run, outputs may need to be further processed for end use (i.e. post-processing). Details of data pre-processing and forecast post-processing are very use-case-specific. To ensure FFPO can be extended for different use cases, we only include general concepts of streamflow forecasting in FFPO. Each of these steps is a type of `opmo:Process`, and four classes are created, i.e. `PreProcessing`, `Calibration`, `Simulation` and `PostProcessing`. Model validation is regarded as a type of model simulation and we model it as a subclass of `Simulation`. These classes inherit the characteristics of `opmo:Process` and can therefore use and generate an `opmo:Artifact`, be controlled by an `opmo:Agent` and triggered by another `opmo:Process`. Also, an `opmo+:Method` can be specified for an `opmo:Process`. To describe global optimisation methods and objective functions used in model calibration, we introduce `GlobalOptimisation` as a subclass of `opmo+:Method` and associate it with an `ObjectiveFunction`.

The inputs and outputs of the steps involved in streamflow forecasting can be grouped into three categories: hydrologic data, hydrologic models and all others (which we call auxiliary data). Each category is modelled as a subclass of `opmo:Artifact`, and three classes are thus created, i.e. `HydroData`, `HydroModel` and `AuxiliaryData`. `HydroData` include hydrologic observations, those derived from observations and those used or generated by a hydrologic model as model input or output, model parameters or states. They can be `TimeSeries` or `NonTimeSeries`. Common features of `TimeSeries` include the hydrologic quantity or property observed or derived, the unit used, the time period covered and the time-series type which characterises the relationship between a time instant and a value (e.g. continuous or accumulative), as defined below:

$$\texttt{TimeSeries} \sqsubseteq (=1\ \texttt{hasProperty.Property}) \sqcap (=1\ \texttt{hasUnit.Unit}) \sqcap (=1\ \texttt{hasTimePeriod.TimePeriod}) \sqcap (=1\ \texttt{hasSeriesType.SeriesType})$$

which specifies a time series has exactly one property, one unit, one time period and one series type. In addition, a `TimeSeries` can have a regular `TemporalScale` or `SpatialScale`}, and a `Location` (in the case of forecasts), be provided with an overall `Quality`, or have a `Source`, e.g. `NWPModel` or `Sensor`. We define a `GaugeStation` as a type of `Sensor` and describe it by type, model, location, accuracy and calibration history. Examples of `NonTimeSeries` include `ModelInitialStates` and `ModelParameters`. There are two types of `ModelParameters`: `PhysicalParameters` which can be directly measured, and `ProcessParameters` which need to be calibrated. A `HydroModel` (physically based) is characterised by its simulation mode (i.e. lumped, distributed or semi-distributed), processes (e.g. water or energy balance) and routing schemes. We describe a `HydroModel` mainly from its static aspects and capture its dynamic aspects through model calibration and simulation steps. `AuxiliaryData` are typically used to configure a step, e.g. the calibration period used by model calibration; they can also be auxiliary output of a step, e.g. skill score reports generated by model simulation.

Besides the steps, inputs and outputs, we also model the individuals and organisations involved in streamflow forecasting. We create two classes, `Person` and

Organisation, as subclasses of opmo:Agent for representing the person or organisation, for example, providing HydroData, creating a HydroModel or performing an opmo:Process (e.g. Calibration).

Based on FFPO, we can create the provenance ontology for the South Esk use case. The steps to be modelled in the use case are temporal and spatial normalisation steps. We create two classes for this, i.e. TemporalNorm and SpatialNorm, as subclasses of PreProcessing. A TemporalNorm step requires TimeSeries (e.g. rainfall or streamflow data) as input, and also a TemporalScale which specifies the time-step for the output time series. We define TemporalScale as a subclass of AuxiliaryData. The same applies for SpatialScale and BoundedArea used by a SpatialNorm step, and TimePeriod by Calibration and Simulation. The only physical parameter used by model calibration and simulation is NodeLink, which describes the physical characteristics of the catchment and its channel network. For each step, a Method can be specified. For example, we can specify the method used by model calibration as follows:

$$\texttt{Calibration} \sqsubseteq \forall\ \texttt{useMethod.SCE-UA}$$

Also, to specify the inputs or outputs directly used or generated by a hydrologic model, and to differentiate the warm-up period from the simulation period (both as instances of TimePeriod) used by model simulation, we create six roles (as instances of Role): model_input, model_output, general_input, general_output, warmup_period and simulation_period.

REQUIREMENTS REVISITED

With the provenance model described above, we satisfy the requirements (in the fourth section) by representing generic provenance concepts and then enhancing them with domain-specific semantics. We now illustrate how the model can be used to address the requirements as exemplified by Q1–Q6 .

The model is instantiated using the data from the South Esk use case. Figure 6 shows part of provenance instance data in N3 format (http://www.w3.org/DesignIssues/Notation3):

(a) a forecasting result, *streamflow_forecasts_1*;
(b) the simulation step which generated the forecasting result, *simulation_1* (*time_period_1* and *time_period_2* represent the warm-up period and the simulation period, respectively);
(c) the gridded rainfall data used by the simulation step, *rainfall_gridded_1*;
(d) the spatial normalisation step which generated the gridded rainfall data, *spatial_norm_1*;
(e) the temporally normalised rainfall data used by the spatial normalisation step, *rainfall_tn_1*;
(f) the temporal normalisation step which generated the temporally normalised data, *temp_norm_1*;
(g) the rainfall data provided by BoM and used by the temporal normalisation step, *rainfall_original_1*;
(h) the sensor from which the rainfall data were observed, *sensor_1*.

Based on the instances serialised as RDF triples (http://www.w3.org/TR/rdf-primer/), we then use SPARQL (an RDF query language (http://www.w3.org/TR/rdf-sparql-query/). Queries in SPARQL are expressed in an SQL-like syntax, and answered via triple pattern matching) to query provenance. SPARQL is not aware of the OWL semantics (e.g. subclass relationships) that are using in our provenance ontology. To handle this, we can either pre-compute inference results with the help of an OWL reasoner (Ma *et al.* 2009) or perform reasoning at query time by extending the SPARQL query engine (Ma *et al.* 2008). In the former case, more storage space is needed and the reasoning engine is independent of query processing; while in the latter case longer response times are expected and the reasoning engine needs to be bundled with query processing services. Details of the reasoning algorithms and tradeoffs are outside the scope of this paper.

Below we focus on how Q1–Q6 are addressed using the concepts covered by the model. Suppose provenance is stored in *trace.owl* and *streamflow_forecasts_1* is the forecasting result of interest.

Q1: What's the lead time and at what temporal resolution?

As contextual information of time-series data such as lead time and temporal scale has been represented by the model (in FFPO), we can easily answer this query using the following SPARQL query:

```
PREFIX ffpo: <http://www.csiro.au/provenance/ffpo.owl#>
PREFIX : <http://www.csiro.au/provenance/trace.owl#>
```

Figure 6 | Excerpt of provenance for the South Esk case.

```
SELECT ?time_period ?time_scale
FROM   <http://www.csiro.au/provenance/trace.owl#>
WHERE {
:streamflow_forecasts_1     ffpo:hasTimePeriod ?time_period;
ffpo:hasTimeScale ?time_scale.
}
```

Q2: Which hydrologic model was used?

Answering this query requires finding the `Simulation` step used to generate the forecasting result possibly through

multiple steps. We achieve this through wasGeneratedByStar defined in OPMO.

```
PREFIX opmo: <http://openprovenance.org/model/opmo>#>
PREFIX ffpo: <http://www.csiro.au/provenance/ffpo.owl#>
PREFIX : <http://www.csiro.au/provenance/trace.owl#>
SELECT ?model
FROM <http://www.csiro.au/provenance/trace.owl#>
WHERE {
:streamflow_forecasts_1 opmo:wasGeneratedByStar ?process.
?process a ffpo:Simulation;
        opmo:used ?x.
?x a ffpo:HydroModel.
}
```

Q3: Who calibrated the model and which calibration method was used?

To answer this query, we can use a similar approach to the one we use in Q2: first, we find the Calibration step; then we use wasControlledBy and useMethod (defined in OPMO and OPMO+, respectively) to find the person who calibrated the model and the calibration method.

```
PREFIX opmo: <http://openprovenance.org/model/opmo>#>
PREFIX opmo+: <http://www.csiro.au/provenance/opmo+.owl#>
PREFIX ffpo: <http://www.csiro.au/provenance/ffpo.owl#>
PREFIX : <http://www.csiro.au/provenance/trace.owl#>
SELECT ?person ?method
FROM <http://www.csiro.au/provenance/trace.owl#>
WHERE {
:streamflow_forecasts_1 opmo:wasGeneratedByStar ?process.
?process a ffpo:Calibration;
        opmo:wasControlledBy ?person;
        opmo+:useMethod ?method.
}
```

Q4: What data were used as model input in the simulation?

Answering this query requires differentiating the data used by the Simulation step, as not all of them were used as model input. We achieve this by specifying the roles of data (Role instances are defined in SEPO).

```
PREFIX opmo: <http://openprovenance.org/model/opmo>#>
PREFIX ffpo: <http://www.csiro.au/provenance/ffpo.owl#>
PREFIX sepo: <http://www.csiro.au/provenance/sepo.owl#>
PREFIX : <http://www.csiro.au/provenance/trace.owl#>
SELECT ?property
FROM <http://www.csiro.au/provenance/trace.owl#>
WHERE {
 :streamflow_forecasts_1 opmo:wasGeneratedByStar ?process.
   ?process a ffpo:Simulation.
   ?used a opmo:Used;
         opmo:effect ?process;
         opmo:cause ?artifact;
         opmo:role sepo:model_input.
   ?artifact a ffpo:TimeSeries;
             ffpo:hasProperty ?property.
}
```

Q5: Which preprocessing steps and methods were performed for a certain model input (e.g. *rainfall_gridded_1*)?

To answer this query, we need to find all the steps that are subtypes of Preprocessing (defined in FFPO), and the methods used in these steps. Again, wasGeneratedByStar is used.

```
PREFIX opmo: <http://openprovenance.org/model/opmo>#>
PREFIX opmo+: <http://www.csiro.au/provenance/opmo+.owl#>
PREFIX ffpo: <http://www.csiro.au/provenance/ffpo.owl#>
PREFIX : <http://www.csiro.au/provenance/trace.owl#>
SELECT ?process ?method
FROM <http://www.csiro.au/provenance/trace.owl#>
```

```
WHERE {
:streamflow_forecasts_1 opmo:wasGeneratedByStar
    ?process.
    ?process a ffpo:PreProcessing;
             opmo + :useMethod ?method.
}
```

Q6: For a gauge involved, when was it last calibrated?

To answer this query, we need to find all the time series that have `GaugeStation` as `Source`. We use `wasDerivedFromStar` to achieve this.

```
PREFIX opmo: <http://openprovenance.org/model/opmo>#>
PREFIX ffpo: <http://www.csiro.au/provenance/ffpo.owl#>
PREFIX : <http://www.csiro.au/provenance/trace.owl#>
SELECT ?sensor ?last_calibration_time
FROM <http://www.csiro.au/provenance/trace.owl#>
WHERE {
:streamflow_forecasts_1 opmo:wasDerivedFromStar
    ?x.
    ?x a ffpo:TimeSeries;
      :hasSource ?s.
    ?s a ffpo:GaugeStation;
      ffpo:lastCalibratedAt?last_calibration_
    time.
}
```

DISCUSSION AND CONCLUSIONS

Provenance has been studied in a number of domains, such as earth sciences and bioinformatics. One fundamental issue for provenance is its representation. This paper presents a provenance model for representation of provenance information in streamflow forecasting. Driven by a case study, the model has been designed to support usability, extensibility and reusability, by extending the widely used Open Provenance Model (OPM), using the formal logic-based Web Ontology Language (OWL) for encoding, and decoupling domain-independent provenance from domain-specific provenance, and application-general provenance from use-case-specific provenance. Further, the model has been evaluated with respect to its ability to satisfy the requirements as exemplified by a set of provenance questions.

Being metadata describing environmental data, the model bears some similarities to existing standards such as OGC Observations and Measurements (O&M) (Cox 2007a, 2007b), Ecological Metadata Language (EML) (EML Project Members 2008) and Water Markup Language (WaterML) (Zaslavsky *et al*. 2007). However, they are designed for different purposes, thus having different scopes and foci. While the model presented is used to represent the derivation history of a forecasting result, including the sources, intermediate data products and the process that led to the result, O&M and similar standards are used for exchanging information describing the collection, analysis and reporting of environmental observations. They may contain similar concepts; however, these concepts typically have different intensions or meanings. For example, the `Process` concept of the model (as inherited from OPMO) denotes 'action or series of actions performed on or caused by artifacts, and resulting in new artifacts' (Moreau *et al*. 2010), while in O&M, the concept denotes the procedure to generate an observation result, which can be a sensor, a human observer or an algorithm applied (Cox 2007a). Their meanings overlap only partially (when the O&M process refers to an action, e.g. an algorithm applied).

The model may change over time, due to changes to OPM, or changes to the conceptualisation of provenance requirements in streamflow forecasting (e.g. a better understanding of the requirements). In such a case, we need to manage multiple versions of the model and ensure that provenance instance data that conform to the older versions can still be interpreted correctly. There has been some research work on ontology versioning and evolution (e.g. Noy & Klein 2004; Noy & Musen 2004). For our work, we need to find out what are possible changes to the model and their effects on instance data. We leave this to future work.

ACKNOWLEDGEMENTS

This work is supported by CSIRO's Water for a Healthy Country Flagship and the Tasmanian ICT Centre. The Tasmanian ICT Centre is jointly funded by the Australian

Government through the Intelligent Island Program and CSIRO. The Intelligent Island Program is administered by the Tasmanian Department of Economic Development, Tourism and the Arts. The authors would like to thank the Real-Time Water Information Systems project team of the Centre for providing the South Esk use case, and Thomas Pagano for useful discussions.

REFERENCES

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D. & Patel-Schneider, P. F. (eds) 2003 *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge.

Beven, K. J. 1993 Prophecy, reality and uncertainty in distributed hydrological modeling. *Adv. Wat. Res.* **16** (1), 41–51.

Bose, R. & Frew, J. 2005 Lineage retrieval for scientific data processing. *ACM Comput. Surveys* **37** (1), 1–28.

Butts, M. B., Payne, J. T., Kristensen, M. & Madsen, H. 2004 An evaluation of the impact of model structure on hydrological modelling uncertainty for streamflow simulation. *J. Hydrol.* **298**, 242–266.

Cox, S. 2007a Observations and Measurements – Part 1 – Observation Schema Version 1.0 OGC. OGC document 07-022r1.

Cox, S. 2007b Observations and Measurements – Part 2 – Sampling Features Version 1.0 OGC. OGC document 06-188r1.

CSIRO 2007 A Report to the Australian Government from CSIRO Murray-Daring Basin Sustainable Yields Project. Available from: http://www.csiro.au/files/files/pgbx.pdf.

Cui, Y. & Widom, J. 2000 Practical lneage tracing in data warehouses. In: *Proc. ICDE'00*. IEEE, Piscataway, NJ, pp. 367–378.

da Silva, P. P., McGuinness, D. L. & Fikes, R. 2006 A proof markup language for semantic web services. *Inf. Syst.* **31** (4).

Davidson, S., Boulakia, S. C., Eyal, A., Ludascher, B., McPhillips, T., Bowers, S., Anand, M. K. & Freire, J. 2007 Provenance in scientific workflow systems. *IEEE Data Engng. Bull.* **30**, 1–7.

DIPIW 2009 Draft South Esk River Catchment Water Management Plan. Available from: http://www.dpiw.tas.gov.au/internnsf/WebPages/JMUY-7J69CJ?open.

Dozier, J. & Frew, J. 2009 Computational provenance in hydrologic science: a snow mapping example. *Phil. Trans. R. Soc.* **367** (1890), 1021–1033.

Duan, Q., Sorooshian, S. & Gupta, V. K. 1992 Effective and efficient global optimization for conceptual rainfall-runoff models. *Wat. Res. Res.* **28** (4), 1015–1031.

EML Project Members 2008 Ecological Metadata Language (EML). Available from: http://knb.ecoinformatics.org/softwar/eml.

Foster, I. T., Vockler, J. -S., Wilde, M. & Zhao, Y. 2002 Chimera: a virtual data system for representing, querying, and automating data derivation. In: *Proc. SSDBM'02*, IEEE, Piscataway, NJ, pp. 37–46.

Freitas, A., Knap, T., O'Riain, S. & Curry, E. 2011 W3P: building an OPM based provenance model for the Web. *Future Gen. Comput. Syst.* **27**, 766–774.

Frew, J. & Bose, R. 2001 Earth system science workbench: a data management infrastructure for earth science products. In: *Proc. SSDBM'01*. IEEE, Piscataway, NJ, pp. 180–189.

Frew, J., Metzger, D. & Slaughter, P. 2008 Automatic capture and reconstruction of computational provenance. *Concurr. Comput. Practice Experience* **20** (5), 485–496.

Greenwood, M., Goble, C., Stevens, R., Zhao, J., Addis, M., Marvin, D., Moreau, L. & Oinn, T. 2003 Provenance of e-science experiments – experience from bioinformatics. In: *Proc. UK e-Science All Hands Meeting 2*. EPSRC, London, 223.

Groth, P. & Moreau, L. 2011 Representing distributed systems using the open provenance model. *Future Gen. Comput. Syst.* **27**, 757–765.

Juran, J. M., Gryna, F. M. & Bingham, R. S. 1974 *Quality Control Handbook*. 3rd edition. McGraw-Hill, New York.

Ma, L., Sun, X., Cao, F., Wang, C., Wang, X., Kanellos, N., Wolfson, D. & Pan, Y. 2009 Semantic enhancement for enterprise data management. In: *Proc. ISWC'09*. Springer, Berlin.

Ma, L., Wang, C., Lu, J., Cao, F., Pan, Y. & Yu, Y. 2008 Effective and efficient semantic web data management over DB2. In: *Proc. SIGMOD'08*. ACM, New York.

Mathevet, T. 2005 *Which Rainfall-runoff Model at the Hourly Time-step? Empirical Development and Intercomparison of Rainfall-runoff Models on a Large Sample of Watersheds*. ENGREF University, Paris, France.

McGregor, J. L. & Dix, M. R. 2008 An updated description of the conformal-cubic atmospheric model. In: *High Resolution Simulation of the Atmosphere and Ocean*. Springer, Berlin.

Miles, S., Groth, P., Branco, M. & Moreau, L. 2006 The requirements of recording and using provenance in e-science experiments. *J. Grid Comput.* **5** (1), 1–25.

Milly, P. C. D., Dunne, K. A. & Vecchia, A. V. 2005 Global patterns of trends in strem flow and water availability in a changing climate. *Nature* **438**, 347–350.

Missier, P., Sahoo, S. S., Zhao, J., Goble, C. & Sheth, A. 2010 Janus: from workflow to semantic provenance and linked open data. In: *Proc. IPAW'10*. Springer, Berlin.

Moradkhani, H. & Sorooshian, S. 2009 General review of rainfall-runoff modeling: model calibration, data assimilation, and uncertainty analysis. In: *Hydrological Modelling and the Water Cycle*. Springer, Berlin, pp. 1–24.

Moreau, L. 2011 Provenance-based reproducibility in the Semantic Web. *J. Web Semantics* **9**, 202–221.

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E. & Den Bussche, J. V. 2010 The open provenance model core specification (v1.1). *Future Gen. Comput. Syst.* **27** (6), 743–756.

Noy, N. F. & Klein, M. 2004 Ontology evolution: not the same as schema evolution. *Knowledge Inf. Syst.* **6** (4), 428–440.

Noy, N. F. & Musen, M. A. 2004 Ontology versioning in an ontology management framework. *IEEE Intell. Syst.* **19** (4), 6–13.

Perrin, C., Michel, C. & Andreassian, V. 2003 Improvement of a parsimonious model for streamflow simulation. *J. Hydrol.* **279**, 275–289.

Pignotti, E., Edwards, P., Gotts, N. & Polhill, G. 2011 Enhancing workflow with a semantic description of scientific intent. *J. Web Semantics* **9** (2), 222–244.

Sahoo, S. S., Barga, R., Sheth, A., Thirunarayan, K. & Hitzler, P. 2010 PrOM: a semantic web framework for provenance management in science. In: *Proc. WWW'10*. ACM, New York.

Sahoo, S. S., Sheth, A. & Henson, C. 2008 Semantic provenance for escience: managing the deluge of scientific data. *IEEE Internet Comput.* **12** (4), 46–54.

Sahoo, S. S., Weatherly, D. B., Mutharaju, R., Anantharam, P., Sheth, A. & Tarleton, R. L. 2009 Ontology-driven provenance management in escience: an application in parasite research. In: *Proc ODBASE'09*. Springer, Berlin.

Simmhan, Y. L., Plale, B. & Gannon, D. 2005 A survey of data provenance in e-science. *SIGMOD Record* **34** (3), 31–36.

Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A. L. & Rosse, C. 2005 Relations in biomedical ontologies. *Genome Biol.* **6**, R46.

Tan, W. C. 2007 Provenance in databases: past, current, and future. *IEEE Data Engng. Bull.* **30**, 3–12.

Taylor, K., Walker, G. & Abel, D. 1999 A framework for model integration for spatial decision support systems. *Int. J. Geogr. Inf. Sci.* **13** (6), 533–555.

Wang, R. Y. & Strong, D. M. 1996 Beyond accuracy: what data quality means to data consumers. *J. Mngmnt. Inf. Syst.* **12** (4), 5–34.

Woodruff, A. & Stonebraker, M. 1997 Supporting fine-grained data lineage in a database visualization environment. In: *Proc. ICDE'97*. IEEE, Piscataway, NJ.

Zaslavsky, I., Valentine, D. & Whiteaker, T. 2007 CUAHSI WaterML. OGC Dicussion paper OGC 07-041r1.

Zhao, J., Sahoo, S. S., Missier, P., Sheth, A. & Goble, C. 2011 Extending semantic provenance into the web of data. *IEEE Internet Comput.* **15** (1), 40–48.

First received 13 October 2011; accepted in revised form 26 January 2012. Available online 13 June 2012