Minimal requirement provenance capturing workflow systems for hydrological modelling

T. C. Smith and A.J. Davidson

NSW Department of Planning, Industry and Environment, NSW
Email: timothy.smith@dpi.nsw.gov.au

Abstract: The NSW Department of Planning, Industry and Environment (DPIE) creates hydrological models to inform policy and decision making in the water sector. High-quality models can be used for decades, and can sometimes be used for purposes not originally envisioned at their creation, delivering additional benefits and insights. In order to maintain the models high quality, new information (such as better data sets) must be incorporated into the model in a timely and accurate fashion. Potential inputs must be assessed and processed consistently and have sufficient metadata for auditing and quality assurance. Historically the manual processes involved with these requirements have contributed to models being expensive to create and maintain.

Well-designed information systems can reduce the maintenance costs of models through simplified data acquisition, repeatable quality assurance and transformation processes, and tracking user activity through time. However, such systems can pose their own management challenges. They require careful consideration of business requirements and capability so that the benefits of adoption remain worth the implementation and maintenance costs.

The Time-series Input Management System (TIMS) is a software system designed to address the challenges of long-term, high volume data management by guiding users through provenance capturing workflows for acquiring and quality assuring time-series of hydrological models. This system is designed to provide:

- An easy to use provenance capturing workflow system that functions without complex software and hardware dependencies.
- A guided workflow-based system that captures sufficient metadata automatically to assemble a provenance trail, and provides a flexible user experience that is of sufficient quality to promote user uptake.

TIMS requirements are simple and the system can be run from a USB drive. User adoption has been promising, and system functionality has been progressively rolled out through DPIE's surface water modelling group. TIMS allows provenance to be both captured and extracted. Introspective workflows and operations that consume provenance information to assist in modelling work have been successfully created, with extensive consultation with technical reference groups to ensure processes embody best modelling practice. Ongoing development on TIMS continues to expand functionality to meet user needs.

Keywords: *Provenance, workflows, hydrology*

1. INTRODUCTION

The NSW Department of Planning, Industry and Environment (DPIE) creates hydrological models to inform policy and decision making in the water sector. High-quality models can be used for decades, and can sometimes be used for purposes not originally envisioned at their creation, delivering additional benefits and insights. In order to maintain the models high quality, new information (such as better data sets) must be incorporated into the model in a timely and accurate fashion. Potential inputs must be assessed and processed consistently and have sufficient metadata for auditing and quality assurance. Historically the manual processes involved with these requirements have contributed to models being expensive to create and maintain.

Well-designed information management systems can reduce the maintenance costs of models through simplified data acquisition, repeatable quality assurance and transformation processes, and tracking user activity through time. They require careful consideration of business requirements and capability so that the benefits of adoption remain worth implementation effort. Designing effective, automated, provenance capturing systems is more complex where workflows are complex, partially manual, not standardised or run on distributed, heterogeneous systems (Hartcher 2013). Software systems can reduce the cost of use to individual users, but without good upfront design, such systems can become expensive, complex, bespoke, and highly dependent on other systems or technology. Good system design should optimise the potential for re-use elsewhere in organisations, reduce future maintenance costs, and consider interoperability with other systems.

Time-series management practices (extraction, transformation and coordination) are well known, standardized, and relatively straightforward, require little manual intervention, and are also an area whose outputs are likely to be referenced later by larger provenance traces. This makes them an excellent candidate domain for automated provenance capture with the exception that they often involve interactions between distributed heterogeneous systems.

We created a system with two initial design goals;

- An easy to use provenance capturing workflow system that functions without complex software and hardware dependencies.
- A guided workflow-based system that captures sufficient metadata automatically to assemble a provenance trail (Barga and Digiampietri 2006), and provides a flexible user experience that is of sufficient quality to promote user uptake.

2. TIME-SERIES INPUT MANAGEMENT SYSTEM

The Time-series Input Management System (TIMS) is an intentionally simple system that uses workflows to reduce the costs around maintaining time-series inputs to models by;

- automating many of the steps involved in locating, cleaning and extending time-series,
- automating the capture of metadata that describes the time-series,
- providing a standardised mechanism to display metadata to users, and
- providing mechanisms to communicate provenance information about time-series directly to other systems that maintain models.

2.1. Installation system requirements

TIMS was created to minimise barriers that reduce adoption, specifically cost and complexity. It uses a small number of software components that are readily available, free, easily installed, and do not require specialized computer systems. The systems and standards these components utilise are widespread and well understood
The components are presented below;

- TortoiseSvn and Subversion: a file-based version control system used as a data store
- Bespoke command-line utilities: Activities within a workflow are a series of bespoke command-line utilities with agreed command-line interface (Smith, Car and Smith 2013) written in C# and executable from command line without being installed.

- Batch files: Workflows are rendered as windows batch files (with constrained syntax) that marshal and execute the Activities by simply executing them and passing parameters

While there are also libraries and utilities for advanced users, these are written in C#, and optional.

These components and systems were chosen for the following reasons:

- Low running costs, as all the requirements can be acquired for free.
- The systems have good projected forecasts for ongoing support. This is based on the theory that, lacking additional information, one should assume any given technology is in the middle of its lifespan. Older systems are thus more likely to be available for longer. Systems that already have a high adoption are more likely to remain supported into the future. E.g. it is unlikely that Microsoft will drop support for its Command Line, which has been available since MS-DOS.
- The systems and standards involved are easily available, well documented, and well understood. This increases the likelihood of re-use of the system by others and improves the potential for active participation in development by a wider range of users.

2.2. Data conceptualisation

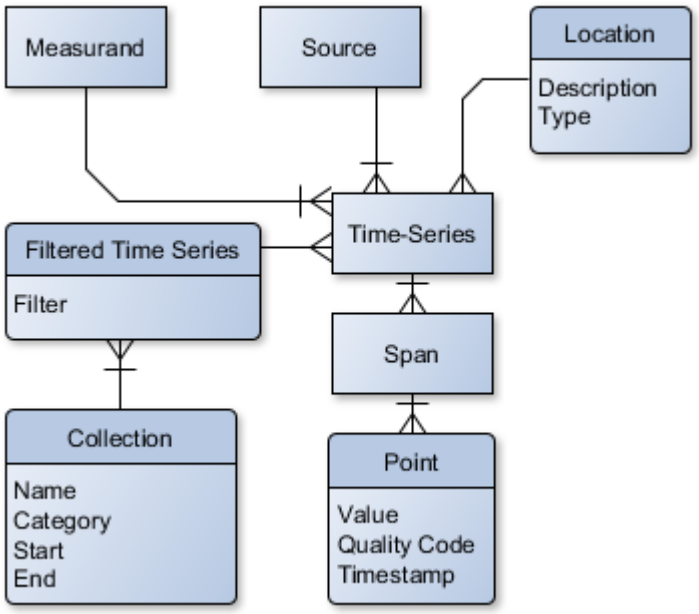

Figure 1. Data Structure

Figure 1 gives the entity breakdown of the way data is structured in TIMS – though metadata is excluded for simplicity. All entities can and should have metadata, and should contain sufficient to identify how they were created.

A time-series has a **location** identifier (such as a gauge number), a **measurand** (such as flow rate or rainfall), a data **source**, and a number of **spans**. Spans contain a series of data **points** related in some way (usually because they were acquired by the same workflow). Each point must have a timestamp which identifies their temporal identity, and usually have a **Value and** a **Quality Code**.

Collections contain views of time-series, which are created using a user specified filter and artificially constraining the start and end point of the time series.

Not shown in the diagram is that as the datastore is version controlled, the given state of any part of the data store at any time can be retrieved. The state is changed via executing workflows and all executed workflows are identified, linked to the state changes in the data store, and themselves stored in perpetuity.

2.3. Operation

TIMS can be operated by a user through one of four methods.

a. choose a workflow template from the user interface, inputs required configuration data and runs the configured workflow.

b. build a bespoke workflow, either by hand or (more likely) using one of the associated libraries such as BatchUtilities (a TIMS C# library that provides mechanisms to create, parse and process workflows).

c. create an operation, add it to TIMS, create a bespoke workflow and execute it.

d. change the state of the data store and enter documentation by hand.

This last option is required because while there is a large advantage to automated collection (Barga and Digiampietri 2006), the system must be flexible enough to deal with the occasional un-automatable request (Car, Hartcher and Stenson 2013). We assumed the majority of users would access the system using the first two methods, so have focused on usability for these. The last two methods have only been used by workflow developers to date.

The basic process to use the User Interfaced is as follows:

Step 1: User selects a Workflow Template

Step 2: User configures the Workflow Template to create a specific workflow.

Step 3: System executes the workflow, which alters the data store.

Step 4: If the execution is successful, the System commits the changes.

Step 5: If changes are committed, the System commits the workflow.

Step 1: User Selects a Workflow Template. When using the UI, TIMS presents the user with a simple graphical user interface that lists a number of available workflows (Figure 2).

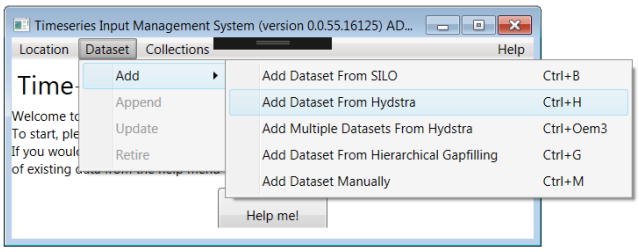

Figure 2. Selecting a workflow from the main screen

Step 2: User Configures the Workflow Template to create a Workflow. When a user chooses a workflow, they are presented with a series of inputs that must be provided to the workflow. Often these inputs will be augmented or simplified by the UI. For example: when a user enters a site in the screen below, TIMS will interrogate a specific external Hydstra system for up to date information on available measurands, then provide these as a selectable list. (Figure 3)

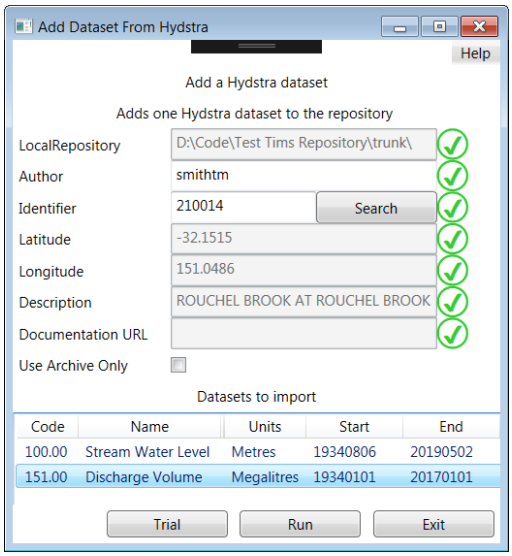

Figure 3. Adding a dataset from a data source

Step 3: System executes the workflow, which alters the data store. Upon clicking "Run" the UI creates a unique workflow identifier, creates an unversioned location within the repository for that identifier, creates a workflow batch file (Figure 4) and stores it in that location and then executes that workflow. The execution of the workflow is (optionally) surfaced to the user.

```
cd ..\..\Programs
if %errorlevel% neq 0 exit /b %errorlevel%
Repository refresh "..\TimeSeries"
if %errorlevel% neq 0 exit /b %errorlevel%
Hydstra fetch_by_datasource "NSW" "..\TimeSeries" "210014" "18890101" "20190730" "smithtm" "CP" "59923eae-4465-432f-8f5b-40540c96e337" 151.00
if %errorlevel% neq 0 exit /b %errorlevel%
ForsView build --provenance "..\TimeSeries\210014\Hydstra\151 "
if %errorlevel% neq 0 exit /b %errorlevel%
Repository commit "..\TimeSeries" "smithtm" "Add Hydstra Dataset 210014. This is part of process 59923eae-4465-432f-8f5b-40540c96e337"
if %errorlevel% neq 0 exit /b %errorlevel%
```

Figure 4. Example workflow batch file

Step 4: If the execution is successful, the System commits the changes. Critically the first step in every workflow should be to remove uncommitted changes from the datastore, and the last step in every workflow should be to commit changes made to the datastore by the workflow (but not the workflow itself) to version control with a particular message. The syntax of this message is controlled to include the activity, critical metadata, and the workflow identifier.

Step 5: If changes are committed, the System commits the workflow. The log of the execution is stored in the workflow's location in the repository as additional metadata. If the workflow successfully committed changes the workflow location (including the workflow and the log) are also committed to version control with a particular message. (Figure 5)

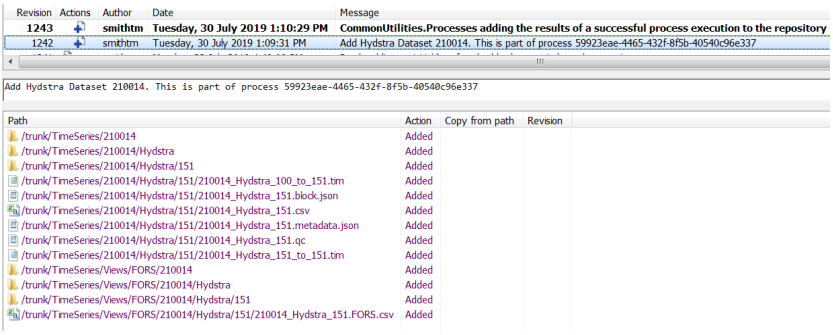

Figure 5. Log entires for workflow

The GUI has a trial mode which allows a workflow to be tested, and the results ignored. This executes a workflow as normal but prevents any changes from being committed to the version control system. This makes it possible for users to try the new functionality, or become comfortable with the system, without worrying about making irrevocable changes. Users can also set up an entirely separate test instances of TIMS

locally, allowing the user to confidently try new workflows on their local machine without any possibility of committing to the central data store.

2.4. Provenance storage and introspective workflows

Workflows, and operations within workflows, have multiple possible sources of inputs; user input, stored data, external information sources, and the provenance information stored within TIMS. Workflows and operations that use provenance information to control their behaviour are defined as "introspective" – that is, they cause TIMS to analyse its own previous processes to inform the behaviour of new processes. At this point only the "Append Collection" workflow, which analyses how time-series were first acquired to reacquire time-series and merge them together, is introspective.

The TIMS provenance model is based on the PROV standard, and while there is an intention to move to a stricter implementation of the standard, this has not yet happened. The information required to create a PROV representation is gathered now, but the information is still stored and surfaced in an idiosyncratic way. It is anticipated the use case that will provide the driver to move to the stricter standard is the need for provenance information to be surfaced to a system outside TIMS.

Currently provenance information is obtained by parsing the metadata in the data store (.json files), the metadata for the changes made to the data store (the version control system log, with syntax constrained messages), the workflow that made those changes (the workflow file), and the diagnostic information output from running the workflow (the workflow log).

3. CONCLUSIONS AND REFLECTIONS

TIMS has been successfully implemented, and has minimal software and hardware requirements – it is possible to run the entire system from a USB stick[1]. All required software is freely available. Hardware and memory requirements are minimal. Storage space could be problematic for particularly large repositories of time-series, but given the explosive growth in data storage capability the financial cost of the storage space is still minimal, and likely to decrease over time.

TIMS has been well adopted. Adoption rates surged when peer-to-peer support became available at most sites, suggesting that training was a barrier to adoption. As of this year TIMS is now mandated for acquisition of time-series for all new surface water models at DPIE.

A major barrier to adoption is performance, as this has a major effect on usability. As the system can exchange a lot of data with various other systems in the process of performing a workflow, users can become frustrated with the system. We are working to reduce the amount of information exchanged, have implemented a 'live view' of what is going on to reassure users the system is working, have worked to try to improve the connection to other systems, and are implementing much more complex workflow templates that can be left to do larger tasks without requiring supervision. We anticipate improved performance would significantly increase adoption, and suggest anyone implementing a similar system consider this a major requirement.

Inappropriate application of TIMS became an issue during adoption. Some users needed to create a 'snapshot' to review the state of data before starting work, and as this data should not be altered, TIMS was not the correct tool. We created a modified version of TIMS named the Data Review Tool (DRT) for this purpose. The DRT has much more basic provenance capture, improved performance characteristics, and precludes altering data. This is because there is only a single operation (acquire data) and it is only performed once – essentially running one workflow in one predefined way. The underlying libraries supporting TIMS have been shared more widely where there was a need for a single operation to be rolled out.

[1] Use of access permissions requires a database based subversion system

Improving provenance extraction has been identified as a future requirement, and a live provenance checking system identified as critical for quality assurance. While provenance extraction is currently time-consuming and manual, the information is present. It is intended that eventually, the provenance data will be cached and surfaced in a system such as PROMS (Car, Stenson and Hartcher 2014).

TIMS has successfully demonstrated the value of a simple user-mediating provenance capturing workflow system, so much so that systems are in development at DPIE for other related domains, specifically a Time-series Output Management System and series of report generation workflows. These systems build on the provenance information created by TIMS (or the DRT), so provenance on a later product, such as a model or report, can be traced back to original source data.

REFERENCES

Barga, R. Digiampietri, L. (2006). Automatic Generation of Workflow Provenance. In Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 1–9. Springer, Heidelberg

Car, N. Hartcher, M. and Stenson, M. (2013). Driving Data Management cultural change via automated provenance management systems. In Piantadosi, J., Anderssen, R.S. and Boland J. (eds) MODSIM2013, 20th International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2013, pp. 2506–2512. ISBN: 978-0-9872143-3-1.
https://www.mssanz.org.au/modsim2013/K5/car.pdf

Car, N. Stenson, M. and Hartcher M (2014). A Provenance Methodology And Architecture For Scientific Projects Containing Automated And Manual Processes. CUNY Academic Works.
http://academicworks.cuny.edu/cc_conf_hic/57

Hartcher, M. (2013). A governance framework for Data Audit Trail creation in large multi-disciplinary projects. In Piantadosi, J., Anderssen, R.S. and Boland J. (eds) MODSIM2013, 20th International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2013, pp. 2506–2512. ISBN: 978-0-9872143-3-1.
https://www.mssanz.org.au/modsim2013/K5/hartcher.pdf

Smith, T., Car N. and Smith, D. (2013). Creating workflows that execute external code bases that are under development. In Piantadosi, J., Anderssen, R.S. and Boland J. (eds) MODSIM2013, 20th International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2013, pp. 2506–2512. ISBN: 978-0-9872143-3-1.
https://www.mssanz.org.au/modsim2013/C8/smith.pdf