8-1-2014

Model Configuration And Data Management In The Short-Term Water Information Forecasting Tools

Jean-Michel Perraud

James Bennett

David Robertson

Phil Ward

11th International Conference on Hydroinformatics
HIC 2014, New York City, USA

MODEL CONFIGURATION AND DATA MANAGEMENT IN THE SHORT-TERM WATER INFORMATION FORECASTING TOOLS

JEAN-MICHEL PERRAUD (1), JAMES BENNETT (2), DAVID ROBERTSON (2), PHIL WARD (3)
(1): CSIRO Land and Water, Black Mountain, Canberra, ACT, Australia
(2): CSIRO Land and Water, Highett, Melbourne, VIC, Australia
(3): CSIRO Land and Water, Clayton, Melbourne, VIC, Australia

The Short-term Water Information and Forecasting Tools (SWIFT) is a suite of tools for flood and short-term streamflow forecasting, consisting of a collection of hydrologic model components and utilities. Catchments are modeled using conceptual subareas and a node-link structure for channel routing. The tools comprise modules for calibration, model state updating, output error correction, ensemble runs and data assimilation. Given the combinatorial nature of the modelling experiments and the sub-daily time steps typically used for simulations, the volume of model configurations and time series data is substantial and its management is not trivial. SWIFT is currently used mostly for research purposes but has also been used operationally, with intersecting but significantly different requirements. Early versions of SWIFT used mostly ad-hoc text files handled via Fortran code, with limited use of netCDF for time series data. The configuration and data handling modules have since been redesigned. The model configuration now follows a design where the data model is decoupled from the on-disk persistence mechanism. For research purposes the preferred on-disk format is JSON, to leverage numerous software libraries in a variety of languages, while retaining the legacy option of custom tab-separated text formats when it is a preferred access arrangement for the researcher. By decoupling data model and data persistence, it is much easier to interchangeably use for instance relational databases to provide stricter provenance and audit trail capabilities in an operational flood forecasting context. For the time series data, given the volume and required throughput, text based formats are usually inadequate. A schema derived from CF conventions has been designed to efficiently handle time series for SWIFT.

BACKGROUND

The Short-term Water Information and Forecasting Tools (SWIFT) (Pagano *et al.* [4]) is a continuous streamflow modelling package designed for scientific research and operational short-term forecasting. In this context, continuous modelling involves the simulation of the effects of soil moisture variability on runoff production efficiency.

The Australian Bureau of Meteorology (the Bureau) is seeking to expand and improve its short-term streamflow forecasting services. The Bureau is engaged in the Water Information Research and Development Alliance (the Alliance) with CSIRO. SWIFT is developed by CSIRO to facilitate research, as well as provide a platform for the Bureau to evaluate and adopt

new forecasting technologies, including ensemble streamflow forecasts. The Bureau requires that the forecasts be skilful and their uncertainty reliably quantified. The methods to produce the forecasts should be practical, transparent, and make effective use of available information.

To align with the focus of HIC 2014, we will present the ongoing developments in SWIFT afferent to the management of data. For the purposes of this paper, data comprises numeric time series inputs and outputs, and the information defining the semi-distributed model structure and its behavior.

Overview of use cases

SWIFT is the hydrological modelling component of a wider conceptual workflow for streamflow forecasting (Figure 1)

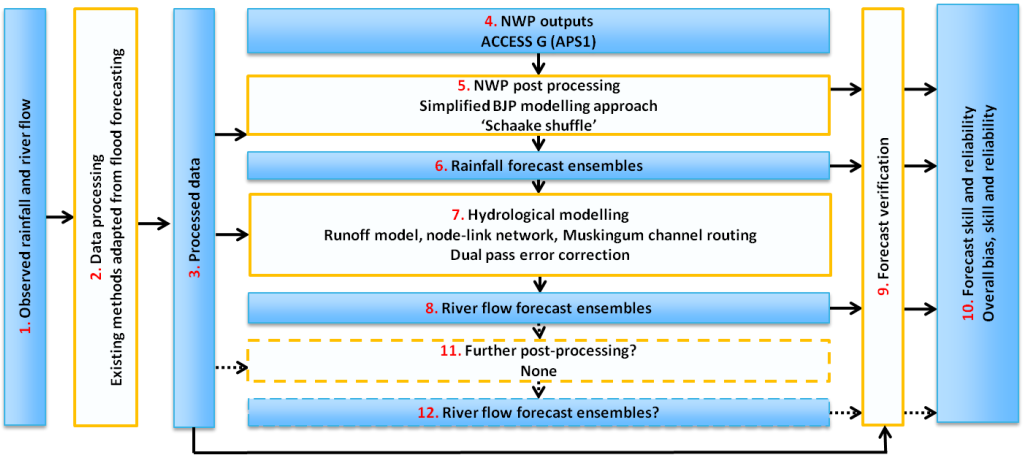

Figure 1. SWIFT (subsystem 7) within the System for Continuous Hydrological Ensemble Forecasting

SWIFT has to address several high level use cases. The simplest run mode is obviously running the model over a time span to output time series. A catchment model usually needs to be calibrated at regular intervals with the addition of new climate and streamflow observation over time, and calibration methodologies explored and assessed in a research context. These use cases are rather common in most hydrologic modelling exercises and will probably be familiar to the reader.

The forecasting aspect prominent in SWIFT introduces additional requirements. In real-time forecasting, the model needs to be hot-started, and allow for adjusting initial states to assimilate information external to the model. Some form of data assimilation is present with the dual-pass error correction technique described in Pagano *et al.* [3]. Other techniques in the category of data assimilation are envisaged but not elaborated on in this paper.

Forecasting is by its very nature dealing with the range of possible evolutions of the system in the future, with an inherent probabilistic approach. Both research and operational uses require ensemble model runs that can be of a large size by hydrologic modelling standards. We note that for streamflow forecasting, it is critically important that all ensemble members are recorded (rather than just some summary statistic, e.g. the mean of the ensemble). This is because streamflow series are usually highly autocorrelated, and taking summary statistics of an ensemble – e.g. the mean, or similar – may not replicate crucial temporal elements of the individual ensemble members (e.g., the rate of rise, the timing of peaks, the rate of fall).

SWIFT catchment models can run at a variety of time steps, in practice it is currently used mostly at the hourly time step. Calibration over up to a decade and ensemble forecasting generate a substantial computational load. Parallel computing within multi-core machines and/or on a compute cluster have to be considered. One ramification of this is the possible need for concurrent access to data, which is usually easy in reading mode, but a complicated matter in write mode such as writing forecast outputs.

A particularly challenging demand on SWIFT is its aim to support both research and operational use. For both research and decision support, but in particular for the latter, reproducibility and transparency is essential. Regarding its data subsystems, operational use must rely on data with a strong audit trail capability. SWIFT can be operated from the Flood Early Warning System (FEWS), Werner *et al.* [6], and input/output adapters exist to exchange data between the two software systems.

Table 1 gives a summary of the mostly independent aspects of the use cases that contribute to make the management of data for SWIFT a challenging problem. Being largely orthogonal aspects, the data dealt with, at least in terms of the model outputs, tend to be an exponential function of the number of these aspects present in the use case at hand.

Table 1. Dimensions of the problem correlated to data and computational sizes

Dimension	Description
Time	Time span of the simulations and input/output time series
Time step	Hourly to daily, with possible sub-hourly uses.
Forecast lead time	Forecast made ahead of the current simulation time, e.g. 7-day hourly forecast.
Ensemble forecast	Forecast made ahead of a point in the simulation time based on ensemble weather predictions
Ensemble simulation	Long term simulation made on alternate climate inputs series or model parameters
Retrospective forecast	For every time step in a retrospective simulation, perform a forecast. Used to assess the performance of forecasting algorithms on past events.
Calibration objectives	Alternate calibration methods for assessment of their performances.
Model configuration	Variation in the structure of the model, for instance alternate rainfall-runoff model for inter-comparison. This includes supporting hot-start a model for real-time forecasting.
Data quality code	Quality code in time series can be used to weight the information content

ANALYSIS

Model configurations as building blocks

While a detailed description and analysis of the use cases cannot fit in this paper, the identification of the following key types of high level information defining simulation tasks will not come as a surprise to the reader. Yet, there is surprisingly little literature with a formal description of the building blocks to manage model configuration. The use cases listed entail a combination of the following elements:

- The structure of the catchment model, e.g. lumped or semi-distributed; what model structure represents the water fluxes (Sacramento, GR5H, etc.)
- The mapping of input climate time series to the input variables of the specific structure of the catchment model. The source of the data may consist of a netCDF file with a schema we will describe later, or a series of text files with file name conventions. The point is that a layer of abstraction is needed in the system to shield the environmental modelling logic from the on-disk representation of input time series.
- The parameterization to apply to the model structure. A set of model parameters may be applied identically to all sub-areas and channel routing algorithms, or grouped by subsets thereof, for instance when transferring parameter sets from calibrated catchments to ungauged ones.
- The initialization of the model state variables prior to the first time step of the simulation, typically setting the level of the 'water buckets' typically found in lumped conceptual rainfall-runoff models.
- The specification of the state variables of the model that are recorded as output time series, e.g. "record runoff depth from each sub-area, and the streamflow discharge at predicted gauge points"
- The specification of the statistics applied to the output time series, e.g. "get the maximum discharge at gauge X,Y,Z, and the Nash-Sutcliffe efficiency of the daily streamflow for the whole catchment"

While this list may look self-evident, in our experience its translation in a software system rarely reflects the conceptual separation in these modelling configuration elements. Information elements that should be separate are stored within a common text file, or conversely a single element is spread across files.

Prevent redundancy of model configuration information

The legacy SWIFT model configuration access arrangements need to duplicate information by copying, pasting and modifying text files to capture a new modelling task. While this may have some benefits argued for in terms of self-contained information for a given modelling task in isolation, there are several drawbacks, with longer term difficulties. The utility programs needed to manage this manipulation of configuration files are effectively trying to recreate capabilities already found in versioning systems and database management systems. The main drawback is that this is not an approach that facilitates the reliable capture of the provenance of model configurations and the resulting predictions.

The system reengineering currently underway promotes model configuration as building blocks that can be composed to define variation from a base case. To provide a provenance trail of model prediction, and it is easier to trace and assess a posteriori the impact of erroneous input data on several modelling workflows, correct and rerun.

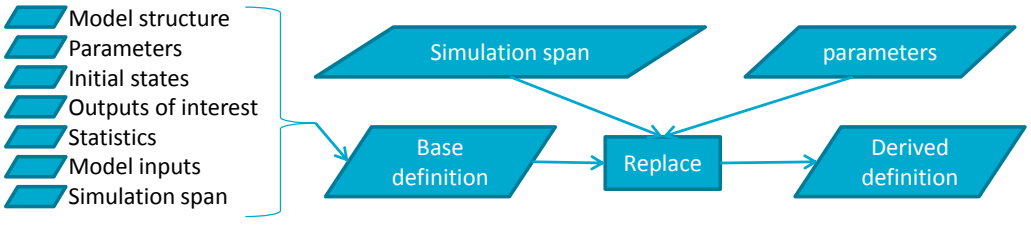

Figure 2. Elements of simulation configurations as building blocks

Time series data storage and netCDF

Plain text files are not well suited to storing the large volumes of data generated for and by ensemble streamflow forecasts with numerical weather prediction models. netCDF (Rew et al [5]) is a binary file format developed primarily for climate, ocean and meteorological data. Detailed, formalised descriptions of the data (metadata) can be included inside the netCDF file, and netCDF can store highly compressed data, making the format suitable for SWIFT. netCDF has traditionally been used to store time slices of gridded data, rather than complete time series of point data, however the format is easily adapted to storing time series data at point locations of the kind commonly used in hydrological modelling (e.g. streamflow at gauge sites; rainfall in catchment subareas or rainfall at weather stations).

Perhaps the most attractive aspect of netCDF is that it is already designed for multi-dimensional datasets. As with all hydrological modelling software, SWIFT must be able to handle time series data at various locations, and the netCDF files are structured with the number of stations as one dimension and time as another dimension (time is treated as the unlimited – i.e. expandable – dimension in the netCDF files). In addition, SWIFT must also be able to handle (large) ensembles and forecast lead-times. Both ensemble size and forecast lead time are assigned dimensions in the netCDF files. The ensemble dimension is conceptually straightforward: ensemble members are simply replicates of a given variable (e.g. streamflow forecasts).

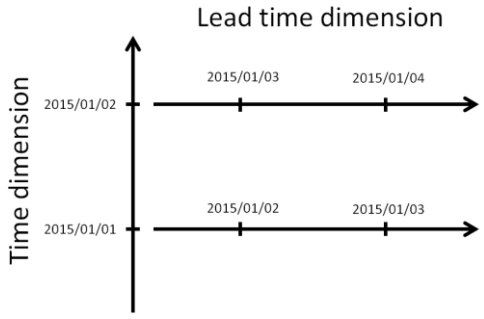

Figure 3. simulation time and lead time dimensions in the netCDF schema

The lead time dimension is defined in relation to the time dimension, and is somewhat more conceptually difficult. Each point on the time dimension may be an instance where a forecast is issued (displayed graphically in Figure 3). For example, forecasts may be issued on consecutive days at the same time, e.g., on January 1, 2015 at 9:00 am and the again at January 2, 2015 at 9:00 am. These forecasts may have long lead times, for example 10 days. Therefore these forecasts overlap considerably (for the 9 days from January 2 onward). Usually, netCDF files rely on only a single time dimension, however the addition of the lead-time dimension allows large archives of forecasts to be stored in a single file, allowing SWIFT to efficiently generate and store many hindcasts for long periods.

Target architecture

From the standpoint of the data subsystems of SWIFT, a key architectural aspect is the separation of generic data handling layer from the persistence layer (Figure 4), so that the details of the storage format do not creep into the core of the system. Judging from the current

technical literature, for instance in Miller [2], the approach is considered state of the art in data-centric business projects, but the hydrology domain has yet to fully catch up to this practice.

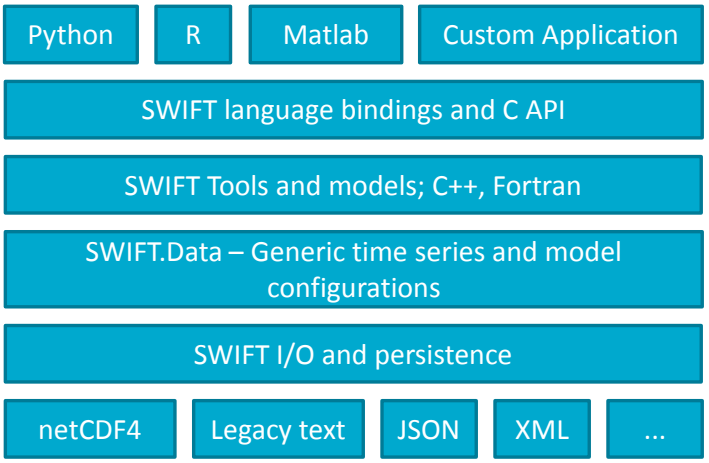

Figure 4. high-level target architecture for SWIFT

Higher up in the software stack, SWIFT has a layer dedicated to exposing an Application Programming Interface (API) that can be accessed by software products and applications. Model configuration and simulation definitions are manipulated at a high level (i.e. with a minimised amount of code and tedium) by the user, using his/her preferred access arrangements (programming languages, workflow systems, etc.). We believe this feature will make a big difference in the user experience, and down the track to the management of the system for operational forecast.

IMPLEMENTATION

SWIFT has been implemented until 2014 in Fortran90. As the needs demanded of the software for research and operational purposes expand, access to reusable libraries written in other languages is warranted to efficiently grow some of the capabilities. A layer written in C++ is currently added, as a common denominator to more easily reuse third party libraries. So far prominent examples are libraries to use the JavaScript Object Notation (JSON), Boost libraries (www.boost.org), and xUnit++ unit testing framework. C/C++ is also easier to bind to from a variety of other programming languages than Fortran.

DISCUSSION AND CONCLUSIONS

The most immediate needs for SWIFT are to support the research endeavours in short term streamflow forecasting. However, its use in an operational context at the Australian Bureau of Meteorology is expected to ramp up over the coming two years. We will not enter in this paper in a discussion on the governance of SWIFT thus required, but identify some research and development areas that may need particular consideration.

We believe that the design and implementation of the SWIFT data subsystems presented in this paper is a solid foundation for dual use in research and operations. To fully realize the value, this needs to fit in broader information management system, beyond the scope of the project owning SWIFT. The Research Alliance comprises research topics on information

modelling and management. Existing research contributions include the Water Data Transfer Format (WDTF) and validation services described in Yu *et al.* [7], and the Provenance Management System (PROMS) described in Car [1]. As we understand, WDTF has been designed and used to format, manage and transfer observation data. It has a direct relevance to observed climatic and streamflow time series used as inputs to SWIFT, and can be one of the preferred data format. One possible topic to explore is its applicability to time series that are not observations but closely related, such as ensemble weather prediction or streamflow series. The relevance may not be in a formatting sense, but conceptual and for metadata management.

netCDF is the preferred storage format for SWIFT data, especially if this data is multi-dimensional. The SWIFT WDTF data schema naturally builds on the Climate and Forecast conventions, as many of these conventions are relevant to the generic design of data schema and metadata. To our knowledge there is no publication dealing specifically with short term ensemble forecasting. One of the outcomes of SWIFT could thus be to propose conventions for such netCDF data schemas.

Finally, a provenance management subsystem is needed to maintain a reliable trail of the data products output by SWIFT, in particular in an operational context. A pragmatic approach to explore the systems design ramifications for SWIFT would be a limited use case aiming to transpose the work done in Car [1] for the provenance of spatial data products to streamflow forecasting.

This paper is presenting the current design and implementation of the model configuration and time series management subsystems of SWIFT, and its connections to some component of a broader information system for the hydrology domain. Other aspects of SWIFT will be presented in subsequent publications, notably the computational aspect.

ACKNOWLEDGEMENTS

This work is carried out in the CSIRO Water for Healthy Country National Research Flagship and is supported by the Water Information Research and Development Alliance between CSIRO and the Australian Bureau of Meteorology.

REFERENCES

[1] Car, N.J., "A method and example system for managing provenance information in a heterogeneous process environment – a provenance architecture containing the Provenance Management System (PROMS)", *MODSIM2013, 20th International Congress on Modelling and Simulation*, Adelaide, Australia, (2013), pp 824–830

[2] Miller, J., "Design Patterns for Data Persistence", *MSDN Magazine*, April 2009, http://msdn.microsoft.com/en-us/magazine/dd569757.aspx#id0400058

[3] Pagano, T.C., Wang, Q.J., Hapuarachchi, H.A.P., Robertson, D., "A dual-pass error-correction technique for forecasting streamflow", *Journal of Hydrology*, Vol. 405, Issues 3–4, (2011), pp 367-381

[4] Pagano T.C., Hapuarachchi H.A.P., Shrestha D.L., Ward P., Anticev J. and Wang Q.J., Hydrologic modelling with Short‐term Water Information Forecasting Tools (SWIFT) to support real‐time short‐term streamflow forecasting, *WIRADA Water Information Research and Development Alliance: Science Symposium Proceedings*, Melbourne, Australia, 1–5 August 2011. CSIRO: Water for a Healthy Country National Research Flagship, (2012), pp 99-105.

[5] Rew, R. K., Hartnett E. J. and Caron J., "NetCDF-4: Software Implementing an Enhanced Data Model for the Geosciences", *22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, AMS (2006).

[6] Werner, M.G.F., Van Dijk, M., Schellekens, J. (2004) DELFT‐FEWS: an open shell flood forecasting system. In Liong SY, Phoon K, Babovic V (eds.) 6th International Conference on Hydroinformatics. World Scientific Publishing Company, Singapore, pp 1205–1212.

[7] Yu, J., Cox, S., Walker, G., Box, P.J. and Sheahan, P., "Use of standard vocabulary services in validation of water resources data described in XML", *Earth Science Informatics*, Vol. 4, No. 3 (2011), pp 125-137.