International Environmental Modelling and Software Society (iEMSs)
2010 International Congress on Environmental Modelling and Software
Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada
David A. Swayne, Wanhong Yang, A. A. Voinov, A. Rizzoli, T. Filatova (Eds.)
http://www.iemss.org/iemss2010/index.php?n=Main.Proceedings

Opportunities and limitations of DelftFEWS as a scientific workflow tool for environmental modelling

Peter Gijsbers
Deltares USA Inc., Silver spring, Maryland, USA, Peter.Gijsbers@deltares-usa.us

Abstract: DelftFEWS is a software platform for real-time operational forecasting systems in the hydrological, hydraulic and water quality domain. Operational water management agencies around the world deploy over 40 simulation models in manual and automated time series processing workflows to produce real-time forecasts information with DelftFEWS. DelftFEWS has also been used outside the scope of real-time applications in studies requiring execution of workflows composed of long term simulations and statistical post-processing tasks. DelftFEWS has not specifically been designed for this purpose and does not intend to be a generic scientific workflow tool similar to e.g. Kepler. However, it offers features which make it, under some conditions, a useful production environment for scientific workflows where a chain of time series are being manipulated by generic (spatial) time series handling operations and dynamic simulation models. Data that cannot be represented as a (spatial) time series cannot be properly accommodated by the system. DelftFEWS is not a suitable environment for discovery types of workflows as its architecture, workflow composition process and its graphical user interface do not offer suitable facilities for the user.

***Keywords**:* scientific workflows; DelftFEWS; environmental modelling

1. INTRODUCTION

1.1 DelftFEWS

DelftFEWS is a state-of-the-art, real time software infrastructure for operational water management and forecasting [Werner et al. 2006]. The system is a sophisticated collection of modules, which can be chained in workflows to build an operational water management system customised to the specific requirements of an individual agency. The philosophy of the system is to provide an open shell system for managing the operational management process [Werner et al. 2004]. The shell system offers workflow based data handling capability, which can combine a comprehensive set of general data transformation functions with an open interface to external models. Over the past 7 years, more than 40 different simulation models of different suppliers have been made operational in the field of surface water and groundwater hydrology, water quality and inland and coastal hydraulics. The modular and highly configurable nature of the system allows it to be used effectively both in rudimentary systems and in highly complex systems utilising several simulation models. DelftFEWS can either be deployed in a manually driven stand-alone mode, as a fully automated client-server application, or as a web-service component.

1.2 Applications

DelftFEWS is mostly applied for real-time forecasting purposes. The overall workflow structure for such forecasting application is composed of a set of workflows, each composed of one or more sub-workflows or activities (see Figure 1). The workflows are

typically executed independently, with different time intervals. Fall back mechanisms can be applied to adjust the strategy in case one or more activities fails to provide data.

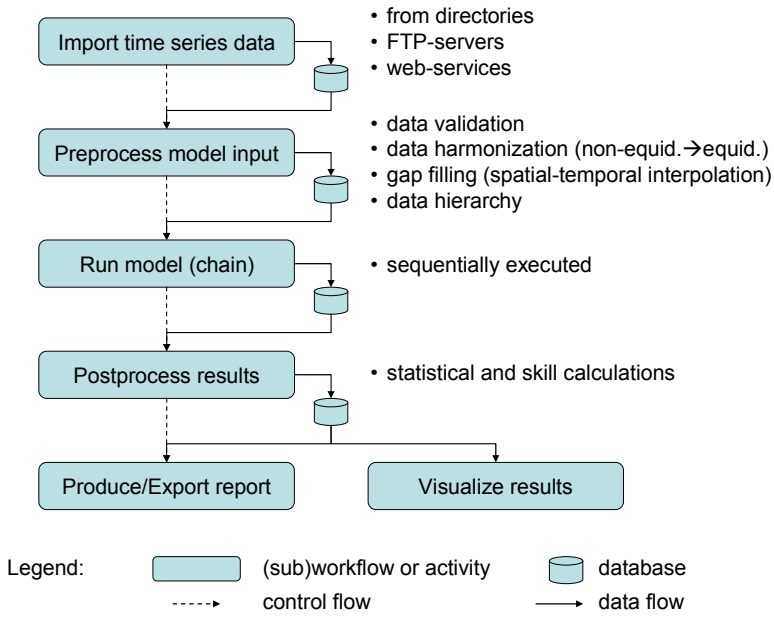

Figure 1. Typical DelftFEWS (forecasting) workflow

In addition to its operational use by forecasting agencies around the world, DelftFEWS is also being used for several research and management studies in the water domain.

One of the most interesting types of DelftFEWS applications from the scientific workflow context is the application in the GRADE project (Generator of Rainfall And Discharge Extremes (GRADE) for the Rhine and Meuse basins). This project [Wit et al. 2007] utilized DelftFEWS to conduct extreme long simulation runs of hydrological runoff and routing to assess flooding conditions using generated rainfall series of 10.000 years length. A workflow was established using the HBV model for runoff computation and hydrological routing [Lindström et al. 1997]. Whenever a peak event was detected in the computed time series, the workflow engine adapts its execution strategy and assigned the flood routing computation activity to a more precise - and computation intensive – hydraulic routing model. Simulations have been conducted with a stand-alone application.

Another interesting scientific workflow application using DelftFEWS is the National Groundwater Modelling System (NGMS), implemented for the Environment Agency of England and Wales. DelftFEWS provides the software platform to conduct impact assessment of groundwater abstractions for policy and planning purposes using numerical groundwater models, such as Modflow, and recharge models [Farrell et al. 2008]. Within this application four default water abstraction scenarios are identified (Historic, Naturalized, Recent Actual, Fully Licensed) to reflect the past and current water management situation. For each scenario, a ‘default’ workflow has been defined to conduct a groundwater simulation run, computed the differences with the other scenarios and calculate statistical aggregates such as mean monthly values. Three additional workflows have been formulated to accommodate what-if scenarios, either derived from the Historic, the Recent Actual or the Fully Licensed situation. For each what-if scenario, the difference with its ‘default’ scenario is calculated, including associated monthly statistics. Within this client-server system, workflow runtimes can vary between 20 minutes and 20 hours depending on the size of the associated groundwater model.

1.3 Scientific Workflows

Scientific workflows combine data and processes into a configurable, structured set of steps that can be implemented in semi-automated computational solutions to address scientific problems. Scientific workflows have become an increasingly important paradigm to accelerate scientific research, as it allows scientific experiments to be conducted through massive computation instead of labour intensive laboratory work. This experimental research - called "in silico" by Woollard et al. [2008] – can be characterized by three phases, a classification that also can be applied to scientific workflow environments:

- discovery: rapid investigation of a scientific principle in which hypotheses are formed, tested, and iterated on rapidly,
- production: the application of a newly formed scientific principle to large data sets for further validation
- distribution: sharing of data results for vetting by the larger scientific community.

While DelftFEWS has been designed for real-time application in the water domain, its functionality may be suitable for some environmental scientific workflows. Using the general (idealized) requirements of Ludäscher et al. [2006] and the categorization of Woollard et al. [2008] as a guidance for discussion, this paper will illustrate that the value of DelftFEWS as a scientific workflow system is primarily in the production phase for time series simulation based research.

2. CAPABILITIES TO MEET SCIENTIFIC WORKFLOW NEEDS

2.1 Access to Data, Resources and Services

Workflows need input data, components that can do the work and resources where the work can be executed. The data ingest into DelftFEWs is a pull-based activity, where an Import workflow reads available files at a directory or FTP-site, or queries a web-service. DelftFEWS supports various common standards and a large number of proprietary formats for the import of scalar and grid time series [Gijsbers et al. 2008]. The supported global standards include Grib, NetCDF (CF-convention) and the DelftFEWS Published Interface. Unfortunately, the hydrological world does not have a global standard for time series data yet. When such standard emerges, it is likely to be supported by DelftFEWS. Services and access to computational resources will be discussed in the next two sections.

2.2 Service Composition and Reuse and Workflow Design

Scientific workflows can be seen as a chain of data handling services that are executed in a particular fashion, e.g. sequentially, parallel, event driven or time step based. In many scientific workflow systems, web-services and Grid computing are the resources that need to be combined and orchestrated. DelftFEWS is more traditional in its architecture, as the services used in its workflows are internal modules, typically executed sequentially inside the workflow engine. Modules themselves may be able to run multiple processing tasks in parallel. The most important DelftFEWS modules are the data transformation module and the general adapter module. Workflows call instances of these modules in a particular sequence, while data flows from one module instance to the next via the input and output time series defined. DelftFEWS workflows are typically configured by hand and/or by scripts which takes the knowledge from another data source. Because of this nature, DelftFEWS is not a suitable candidate workflow environment for discovery type of work.

Data transformations provide the 'plumbing' to transform data in the proper format for the models, called the actors by Ludäscher et al. [2006]. The spatial-temporal and data handling transformations offered by this DelftFEWS module are of a high granular level. The available functions are primarily aimed at preparation of a consistent and complete set of model inputs, e.g. time step harmonization, (dis)aggregation, spatial and temporal gap filling. To accommodate post-processing a large set of functions are available within DelftFEWS to conduct time series and ensemble statistics. Where required, these

transformations can be made conditional, either by time (e.g. do transformation *m* during the summer and transformation *n* during the winter season) or by time series value (e.g. if parameter value at location A > 3 do transformation *n* else do transformation *m*).

The DelftFEWS General Adapter is the module which facilitates the execution of a wrapped external model by way of communication via the DelftFEWS Published Interface (PI) data exchange format [Deltares 2010]. As described in Gijsbers et al. [2008], the General Adapter exports model data sets (schematizations) and model states in native format, while time series are exchanged in PI-XML or NetCDF (grids). A model specific adapter converts this data into the native format for the model. After model execution, the results are converted back by the model adapter into the PI-format for ingest by the General Adapter. Using this concept, all model specific knowledge is embedded in the model adapter, while DelftFEWS only manages time series and model states and data sets.

2.3 Scalability and Detached Execution

Dependent on the kind of workflow, large volumes of data may need to be processed in long running workflows, requiring high-end computational resources, running in the background (so-called detached execution), possibly in distributed fashion. DelftFEWS offers scalability in both stand-alone and client-server applications. Workflows can be configured to run in deterministic (i.e. single trace) or ensemble mode. While most ensemble workflows are based on multiple input traces, some forecasting applications use multi-model ensembles by composing different model chains that execute with the same input. DelftFEWS offers parallelization capabilities to conduct an ensemble workflow on a multi-processor machine by parallel execution of different ensemble member traces.

Detached execution capabilities, i.e. running jobs in the background, are available in the client-server setup of DelftFEWS. Client applications can send a workflow for execution to the Master Controller, which adds the workflow to a job queue and assigns the job to a logical instance of the DelftFEWS engine (a 'shell server'), when it becomes idle [Gijsbers et al. 2008]. Typically, such system layout involves multiple logical shell servers, allowing different workflows to be conducted in the background at the same time. Ensemble workflows may be deployed server side in parallel fashion using a Condor Grid.

2.4 Reliability and Fault-Tolerance

Reliability is very important for an operational task such as flood forecasting. DelftFEWS therefore offers mechanisms, both at the workflow level and module level, to ensure that results can be generated even when the preferred data sources or modelling services are unavailable. The strategy towards a reliable workflow execution starts with mechanisms to prevent model failure due to missing input data. The data transformation module offers the following gap filling functions for this purpose:

- Data hierarchy: a data merge function selects the preferred data source if available, while offering an alternative source if the preferred source is missing.
- Interpolation: a variety of spatial and temporal interpolation functions are available to fill data gaps.

The second defence mechanism against workflow execution failure is the ability to specify a fall back activity for each activity (nested workflow, internal module or external model) in the workflow. This fall back activity is executed in case the primary activity within the workflow fails. Finally, workflow results may trigger other workflows to be kicked off.

2.5 User Interaction

Different types of workflow applications require different types of user interaction. Discovery workflow systems require a user interface which accommodates composition of workflows, while production and distribution workflows systems require user interfaces that accommodate traceability and insight.

DelftFEWS is very much oriented towards production type of workflows. Initially its focus was on automated scheduling of forecast runs as this was the predominant paradigm of flood forecasting in Europe. In the USA, flood forecasters are working more interactively with the data, requiring new capabilities to accommodate user interaction during workflow execution. The so-called modifiers concept has been implemented and the granularity of the workflow steps has been adapted such that the user has the ability to interact and modify the data before each processing step. The graphical user interface has been adjusted to accommodate adjustment of time series and parameter values and switching between input series. After each modification, the user needs to re-run the sub-workflow in which the change has been conducted.

DelftFEWS offers no graphical user interface capabilities to discover new models or compose new workflows. Insight in workflows is provided via the so-called Workflow Navigator (Figure 2). The tool displays the workflow structure in a tree view, allowing the user to drill down to the (computed) time series, model parameters and configuration of each activity. Configuration mistakes are highlighted by a red cross icon, supported with a message holding more information on the error.

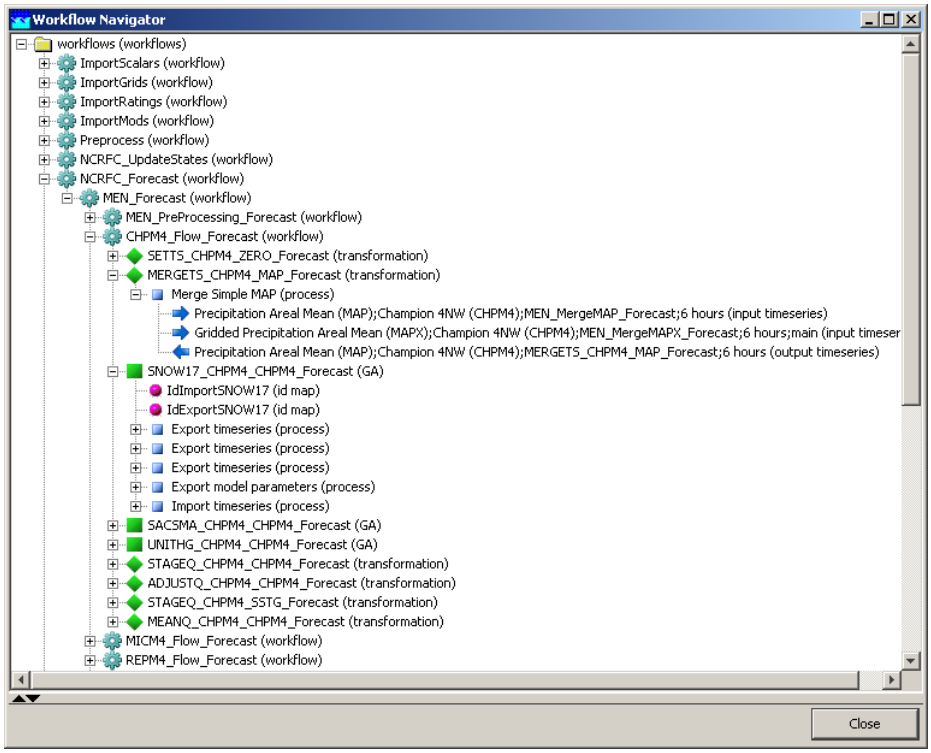

Figure 2. The DelftFEWS workflow navigator

2.6 Smart Reruns

Modification of intermediate data will require rerun of a portion of the workflow. Which portion of a workflow needs to be re-run depends on the topology of the sub-workflows. Only sub-workflows downstream of the modified data will be re-run. An example is given using the workflows of Figure 2, assuming that CHPM4, MICM4, and REPM4 are nodes in the downstream order of the topology. An input time series change to a model in the MICM4_Flow_Forecast workflow will require a rerun of MICM4_Flow_Forecast and REPM4_Flow_Forecast. Workflow CHPM4_Flow_Forecast will not need to be rerun.

In addition, DelftFEWS offers a decision structure in which the workflow can decide to conduct a portion of the run with a more detailed and more computation intensive module.

This facility has been utilized in the GRADE application to run a computation intensive hydraulic model when a pre-defined threshold (a flood stage) was exceeded in the more simple model. The workflow can also decide to conduct a re-run using a another module for a specific time window around the maximum value of a given period. For example, within the GRADE project, the workflow scanned for each simulation year the data computed by the simple routing model, detected the annual flood peak and deployed the hydraulic model for a period of a month around the occurrence of the peak. Such smart decision mechanism prevents the need for computation intensive runs if less computer intensive jobs can indicate that no thresholds are passed or peak events occur.

2.7 Smart Semantic Links

Ludäscher et al. [2006] indicates that a scientific workflow system should assist workflow design and data binding phases by suggesting which actor components (i.e. external models) might possibly fit together. DelftFEWS is not helpful for this discovery type of application. It offers no semantic facilities, nor any workflow configuration editing capabilities. All workflow configuration needs to be conducted with external tools (e.g. XML-editors and scripts). The only assistance offered is primary (xsd) and secondary (overall content consistency) validation and error visualization in the workflow navigator.

2.8 Data Provenance (Reproducibility)

“In silico” research experiments with scientific workflow system should be reproducible. Ludäscher et al. [2006] indicate that a scientific workflow system should be able to automatically log the sequence of applied steps, parameter settings and (persistent identifiers of) intermediate data products. With legal liability issues looming at each flood event, reproducibility of forecast runs is of major importance. The DelftFEWS database keeps track of each piece of data. It registers the producing data source (module instance), the producing task run (workflow), and the date-time when the data is written to the database. All information is included when archiving the run. For complete reproduction, the associated configuration and the binaries used can be archived as well.

2.9 Distribution

While data distribution is not a primary purpose of DelftFEWS, its data synchronization capabilities between the Master Controller and client applications allows users to get easy, automated or on-demand, access to workflow results. Better access to models and model results was the main driver behind the National Groundwater Modelling System.

3. LIMITATIONS TO MEET SCIENTIFIC WORKFLOW NEEDS

3.1 Data Types

DelftFEWS is tailored to deal with transient data of coarse granularity, i.e. regular and irregular time series in scalar, longitudinal or grid structure. All time series are location bound, and are either geo-referenced or bound to a dummy location placeholder. Static data can only be handled as part of the configuration. While powerful if time series are the key data concept in the scientific workflow, DelftFEWS should not be used as a scientific workflow application if data connectivity is not achieved through passing time series.

However, even if time series are the major carrier of information exchanged, attention needs to be paid to the semantics and multiple characteristics of time in the forecasting domain. The forecasting domain is characterized by the fact that the clock moves forward in time. Typically each forecast can be assigned to a specific start time of the run and an offset to the actual forecast time (e.g. the forecast of 12Z UTC, may have models starting 6

hours earlier). This concept underlies the different types of time series distinguished within DelftFEWS. The first categorization is by data source:

- External: ingested from an external source
- Simulated: internally generated (computed).

The second categorization is by their relation to time:

- historical time series: continuous in time, i.e. one parameter value per timestamp
- forecasting time series: characterised by the base time (start time) of the forecast, i.e. allowing multiple runs and multiple parameter values per timestamp.

3.2 Run Management and Run Comparison

When applying DelftFEWS outside the forecasting domain, careful attention needs to be given to proper semantic understanding of the time series types and valid ways of application by the modules in a specific workflow. The following section explains which time series types can be used under which circumstance to enable a certain type of data comparison.

Data generated for the same time stamp can be stored in historical or forecasting time series types. The historical time series types can be used to create a baseline run, i.e. a run which provides the basis for comparison with other runs. The forecasting time series types can be used in workflows that accommodate what-if scenarios. The time series generated can be compared against historical times series by using the module reference that generated the historical time series.

Forecasting time series from module A can be compared against forecasting time series from module B within the same run. Comparison against a series generated by module B in a previous run is only feasible if the time series from module B is generated in another workflow. A proper semantic understanding of the 'current' forecast is important to conduct such comparison.

Within the forecasting domain, each forecast is basically identified by its base time (T0), i.e. the forecast of 12 o'clock. A forecast run needs to be approved to make its results 'current' hence making its results instantly available to system components. Data is by default only retrieved for the 'current' run, i.e. the last approved run of a specific workflow or the workflow run currently being calculated.

Workflows can be auto-approved at configuration time. Alternatively, a data management dialog is available to make another run 'current' or to add the results of other runs to the displays, already showing the current run. What-if scenarios specified for a workflow can be used as a key to identify a run in the data management dialog.

3.3 How to Make it Work in Practice ?

The National Groundwater Modelling System illustrates how a set of scientific workflows can be composed to accommodate run comparisons against a baseline. Table 1 illustrates the time series types used for the different types of workflows.

Table 1. Overview of workflows within NGMS

Workflow	Workflow Type	Time series type	Compared against
Default Historical	Baseline	simulated historical	Default Naturalized
Default Naturalized	Baseline	simulated historical	
Default Recent Actual	Baseline	simulated historical	Default Naturalized
Default Fully Licensed	Baseline	simulated historical	Default Naturalized
Modified Historical	What if	simulated forecasting	Default Historical
Modified Recent Actual	What if	simulated forecasting	Default Recent Actual
Modified Fully Licensed	What if	simulated forecasting	Default Fully Licensed

As shown, a What-if scenario from a specific workflow cannot be compared against another what-if scenario conducted with the same workflow. This is caused by the fact that a forecasting workflow can only have one current forecast run. This may be an approved run, but as soon as a new calculation starts for this workflow, the run underway becomes the current one.

4. OPPORTUNITIES FOR SCIENTIFIC WORKFLOW APPLICATION

Keeping the semantic limitations in mind, DelftFEWS can successfully be applied as a scientific workflow environment for production workflows in the field of time series based simulation modelling and forecasting. Within this context, it can also distribute the results to a users who is connected to the client-server system. The workflow structure, modules and data type semantics offer a high level of granularity, which opens the door for generation of DelftFEWS production workflows from discovery workflow applications.

5. CONCLUSIONS

Although DelftFEWS is not designed as a generic scientific workflow system, it can under certain conditions be used for workflows which Woollard et al. [2008] categorize as production workflows. Once the workflows are designed, DelftFEWS can provide the scientific production environment for time series data processing and simulation modelling. However, its primary application purpose, forecasting, has resulted in a data model and database design that requires extra attention when designing the workflow and time series handling concept for application in DelftFEWS. DelftFEWS may offer, in its client-server setting, a suitable distribution environment for workflow results. However, DelftFEWS is not a suitable environment for discovery purposes as its architecture, workflow composition process and its graphical user interface do not offer appropriate user facilities.

REFERENCES

Deltares, DelftFEWS documentation. http://fewswiki.deltares.nl, 2010.

Farrell R., M. Whiteman, P. Gijsbers, The National Groundwater Modelling System for England and Wales *In: Proc. ModelCARE 2007 Conf., Calibration and Reliability in Groundwater Modelling: Credibility of Modelling,* IAHS Publ. 320, 95–100, 2008.

Gijsbers, P.J.A., M.G.F. Werner, and J. Schellekens, Delft FEWS: A proven infrastructure to bring data, sensors and models together. *in Proc. of the iEMSs 2008 Fourth Biennial Meeting: Int. Congress on Env. Modelling & Software* Vol. 1, 28–36, 2008.

Lindström, G., B. Johansson, M. Persson, M. Gardelin, and S. Bergström, Development and test of the distributed HBV-96 hydrological model. *J. of Hydrology*, 201: 272–288, 1997.

Ludäscher B., I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, and Y. Zhao, Scientific Workflow Management and the Kepler System. *Special Issue: Workflow in Grid Systems. Concurrency and Computation: Practice & Experience* 18(10): 1039–1065, 2006.

Werner, M., M. van Dijk, and J.Schellekens, Delft-FEWS: an open shell flood forecasting system *Proc. of the 5th Int. Conf. on Hydroinformatics.* Vol.2, 1205–1212, 2004.

Werner, M.G.F. and K. Heynert, Open Model integration – a review of practical examples in operational flood forecasting. *Proc. of the 7th Int. Conf. on Hydroinformatics. Gourbesville, Cunge, Guinot and Liong (eds), Research Publishing,* Vol.1, 155–162, 2006.

Wit, M. de, and A. Buishand, Generator of Rainfall And Discharge Extremes (GRADE) for the Rhine and Meuse basins. *RWS RIZA report 2007.027.* KNMI-publication 218. Rijkswaterstaat, Lelystad, the Netherlands, 2007.

Woollard, D., N. Medvidovic, Y. Gil, and C.A. Mattmann, Scientific Software as Workflows: From Discovery to Distribution, *Software,* IEEE 25.4, 37–43, 2008.