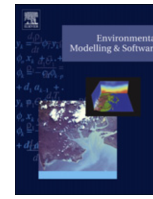
Environmental Modelling & Software 40 (2013) 65–77

Contents lists available at SciVerse ScienceDirect

Environmental Modelling & Software

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



The Delft-FEWS flow forecasting system

M. Werner[a,b,*], J. Schellekens[a], P. Gijsbers[a], M. van Dijk[a], O. van den Akker[a], K. Heynert[a]

[a] Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands
[b] UNESCO-IHE Institute for Water Education, Westvest 7, P.O. Box 3015, 2601 DA Delft, The Netherlands

A R T I C L E I N F O

Article history:
Received 30 December 2011
Received in revised form
16 July 2012
Accepted 18 July 2012
Available online 29 August 2012

Keywords:
Flow forecasting
Model integration
Hydrological modelling
Data standards
Operational hydrology

A B S T R A C T

Since its introduction in 2002/2003, the current generation of the Delft-FEWS operational forecasting platform has found application in over forty operational centres. In these it is used to link data and models in real time, producing forecasts on a daily basis. In some cases it forms a building block of a country-wide national forecasting system using distributed client-server technology. In other cases it is applied at a much smaller scale on a simple desktop workstation, providing forecasts for a single basin. The flexibility of the software in open integration of models and data has additionally appealed to the research community.

This paper discusses the principles on which the Delft-FEWS system has been developed, as well as a brief background of the architecture of the system and concepts used for storing and handling data. One of the key features of the system is its flexibility in integrating (third-party) models and data, and the available approaches to linking models and accessing data are highlighted. A brief overview of different applications of the system is given to illustrate how the software is used to support differing objectives in the domain of real time environmental modelling.

Software availability

Name of software: Delft-FEWS
Contact: fews.info@deltares.nl
Platform: MS Windows and Linux
Coding Language: Java
Availability: www.delft-fews.eu
Documentation: http://public.deltares.nl/display/FEWSDOC/Home
Cost: Free licence for end-users

1. Introduction

Operational forecasting of river flow is becoming increasingly widespread, answering to several objectives such as the provision of early warning of floods to initiate a timely response (Krzysztofowicz et al., 1992; Haggett, 1998; Penning-Rowsell et al., 2000; Parker and Fordham, 1996; De Roo et al., 2003), prediction of low flows for navigation (Renner et al., 2009), or water resource predictions to support reservoir operation (Faber and Stedinger, 2001). Typically delivery of operational flow forecasting is the mandate of operational agencies at the national (Werner et al., 2009), or at the (trans-boundary) basin level (Plate, 2007). Real time observations, and in most cases model predictions, are used as guidance to decision makers on actions to be taken in response to an observed or forecast state of the water system.

To organise the complex process of using data and models in real time, and to combine these in products that can be used in guidance to the decision making process, most operational centres employ flood/flow forecasting systems. Such systems form a special class of environmental decision support systems as they operate in real time, rather than as a tool in support of strategic planning (Matthies et al., 2007). Early examples of such real time decision support systems, typically referred to as flood forecasting systems, include the National Weather Service River Forecasting System (NWSRFS) used for river flow forecasting in the 13 river forecasting centres across the United States (Burnash, 1995), the River Flow Forecasting System (RFFS) applied in the Northeast forecasting centre in England as well as the White Cart Catchment in Scotland (Moore et al., 1990), the Midlands Region Forecasting System used in the Midlands forecasting centre in England (Dobson et al., 1990), and the flood warning system used for the Blue Nile in the Sudan (Grijssen et al., 1992). Conceptually these four examples can be divided into two categories. In the case of the latter two, the forecasting system was essentially built as a shell around the hydrological and hydraulic models used. Werner and Whitfield (2007)

* Corresponding author. Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands. Tel.: +31 88 335 8207; fax: +31 88 335 8582.
E-mail addresses: micha.werner@deltares.nl, m.werner@unesco-ihe.org (M. Werner).

1364-8152
http://dx.doi.org/10.1016/j.envsoft.2012.07.010

refer to this as a model-centric approach. Any change in the model or in the data used to drive the models in real time may require a redesign and redevelopment of the system. In the case of the former two systems a much more modular approach was chosen. Forecasting processes are constructed as a combination of modelling steps and data transformation algorithms. These are then combined to provide required forecast capabilities. Flexibility is achieved through integrating new models and algorithms into the code base.

There are many hydrological and hydraulic models that can potentially be used in operational forecasting, and developments in these result in changing requirements on operational forecasting systems. Additionally the requirements to the use of these models change rapidly due to the increasing availability of real time data from terrestrial networks, from radar and satellite based systems, as well as due to advances in meteorological forecasting. This calls for a flexible approach in establishing sustainable real time decision support systems that can adapt to these changing needs. Rizzoli et al. (2008) advocate abandoning the concept of building monolithic modelling applications in favour of adopting component based modelling frameworks that are constructed from well-defined and documented building blocks. Such an approach was embraced in the development of the Delft-FEWS framework (Werner et al., 2004; Werner and Heynert, 2006). The main purpose of this framework is to provide a platform through which operational forecasting systems can be constructed, and that allows flexibility in the integration of models and data. In contrast to the NWSRFS and the RFFS systems that also follow a modular approach, the Delft-FEWS system contains no inherent hydrological modelling capabilities within its code base. Instead it relies entirely on the integration of (third party) modelling components. Since its introduction in its current form in 2002/2003, this system has been applied in 40+operational flow/flood forecasting centres. Key to its rapid adoption has been the collaborative development process, as well as its ability to build on existing knowledge through integration of existing models and methods where these are available. Both are key factors for the adoption of decision support frameworks within an organisation (Argent et al., 1999).

This paper first provides a short review of the operational forecasting process and the role of Delft-FEWS within that process. Section 3 provides an overview of the philosophy and the most important components and features of Delft-FEWS, while Section 4 discusses some example applications of the Delft-FEWS system in research and operations. A discussion of the systems strengths and limitations is provided in Section 5. Section 6 finally provides a summary of the paper, as well as an outlook on the future development of Delft-FEWS.

2. Role within the forecasting and warning process

Operational forecasting and warning capabilities have been developed in many river basins across the world. Although a wide variety of approaches can be found, the key elements of the flood forecasting and warning process are summarised by Haggett (1998) as four main steps; (i) Detection, (ii) Forecasting, (iii) Dissemination and Warning, and (iv) Response. Within these four steps, Delft-FEWS focuses on the second, or the forecasting step. The primary objective of this step is to provide additional lead time through predictions of short term future hydro-meteorological conditions (Werner et al., 2005). These predictions are used as guidance in making the decision to take an action such as the issuing of a warning. This may then lead to an appropriate response being initiated. As the name suggests, forecasting systems provide support primarily for the second step. This requires the ability to integrate real-time data from hydrological and meteorological observation networks, and the dissemination of prediction results through appropriate products to the warning process. Within the forecasting step, hydrological and hydraulic models may be used to develop a prediction, and the forecasting system needs to support the operation of these models in real-time. These models use real-time input data that has been processed to an appropriate spatial and temporal scale. Additional to the use of these data in running models, the forecasting system needs to support data assimilation and updating, whereby simulated model results are updated to reduce predictive uncertainty (Madsen et al., 2000). When using real-time data, these should be subjected to a rudimentary quality control. Within Delft-FEWS this includes simple range and rate-of-change checks. It is clear that a full quality control cannot be applied during the forecasts process given the time available. This will often occur at a later stage, with the quality controlled data then being stored in the hydrological archive.

To increase lead time, meteorological forecast data from for example Numerical Weather Prediction models (Bartholmes and Todini, 2005) is increasingly being used. This requires the forecasting system to import and process the data from these to serve as future precipitation inputs for the hydrological and hydraulic model chain. Fig. 1 provides a schematic view of the connection between the forecasting system to real time data acquisition systems and dissemination systems. The figure also shows the link to climatological and reference information, as well as archived data. These provide important auxiliary information to the forecaster and can be of use in the (prognostic) verification of forecasts (Demargne et al., 2009).

Operational systems, such as flood forecasting systems will usually be used for quite a period of time, in some cases this may be 20 years or longer (Burnash, 1995; Grijssen et al., 1992). Clearly during this period neither real-time data availability, information requirements in the warning process, nor capabilities of meteorological, hydrological, and hydraulic models will remain the same. There will be a strong desire to incorporate advances in these into the operational domain, thus requiring the forecasting systems to be adaptable to these changing needs.

Change is, however, not easy to achieve given the operational setting of the forecast process. Besides linking data and models, the forecasting system also provides the interface to the forecasting

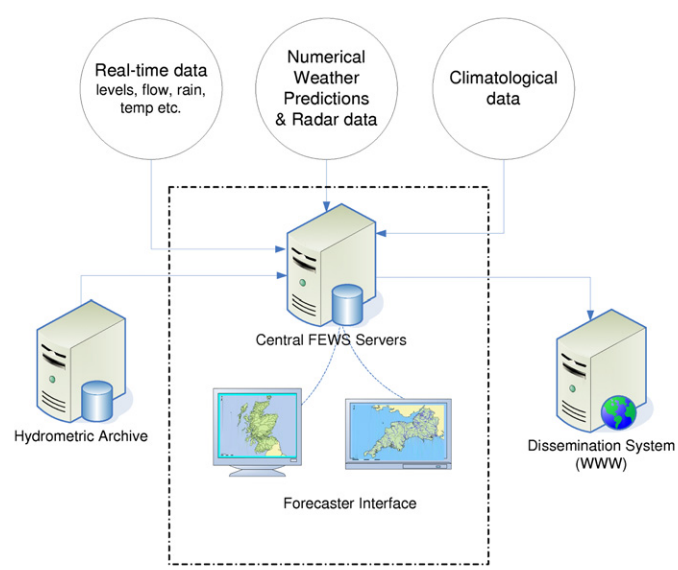


Fig. 1. Schematic structure of a flood forecasting system, showing the position of Delft-FEWS and links to other primary systems within the operational environment.

team, which in most operational centres will constitute several persons. They want to process forecasts as efficiently as possible, while gaining clear information on data and forecast results. As the forecasting system forms an important tool in the day-to-day work of the forecasting centre, it is often found that changing the way it is used is not easy. This may not be due to technical constraints alone. A change in the way forecasters work may be required, sometimes including retraining. Forecasting organisations will often be cautious in introducing changes as a consequence (Werner and Whitfield, 2007). Ideally the forecasting system should therefore be flexible to allow change to models and data, while keeping the way forecasters work with it as constant as possible.

3. Structure of Delft-FEWS

3.1. Philosophy and history

Argent et al. (2009) discuss the disadvantage of decision support systems that have a fixed structure and rigid definition of input/output processes. Whilst such systems often suit the original requirements for which they were designed and built, adapting to changing needs will be difficult, and if attempted may even compromise the original design (Argent et al., 2009). With the changing needs posed on operational forecasting systems, the design philosophy of Delft-FEWS follows the concept described by Argent et al. (2009) in that it provides a shell through which an operational forecasting application can be developed specific to the requirements of an operational forecasting centre. In contrast to most other operational forecasting environments, whether built around a fixed set of models (Dobson et al., 1990, e.g.) or set of model components (Moore et al., 1990; Burnash, 1995), no modelling capabilities are part of the system. Harvey et al. (2002) note that when accommodating a wide range of modelling concepts, the inclusion of model specific knowledge in the central data model would significantly increase complexity. Rather than evolve around a (set of) models and modelling concepts in a model-centric approach, the foundation of Delft-FEWS is data-centric, with a common data-model through which all components interact. All time series data (both scalar and gridded) are stored in this common data-model in a database. Modelling capabilities are then linked to the system through one of the interfaces provided to the data-model.

The current generation of Delft-FEWS was developed from several predecessor operational forecasting systems. It's first implementation was an operational system developed for the Nile basin, used by the Ministry of Irrigation and Water Resources in Sudan (Grijssen et al., 1992). This system was adapted for the Punjab in Pakistan (Werner and Dijk, 2005), introducing flexibility in integrating models through a previous version of the model interface used in the current version. Many of the current concepts of data storage and data integration were implemented in the first generation of Delft-FEWS, which was developed within the scope of a European Research project (De Roo et al., 2003). The concepts of these predecessor systems were refined and combined in the current second generation of Delft-FEWS. The development of the second generation of the system was initiated with the establishing of the National Flood Forecasting System in England & Wales (Werner et al., 2009). This second generation of Delft-FEWS fully established the open approach to integration of models and data. Development of the system has additionally been very dynamic as it found application in an increasing number of operational forecasting systems. New developments are generally initiated by the need of additional functionality in these new applications. Here the concept of develop and share has been adopted, whereby these new capabilities are made available to the entire community. Although Delft-FEWS is targeted for use by operational forecasting agencies, its strengths in integrating data and models have also made it attractive to researchers in the field of forecasting. For non-commercial users, the licence agreement to use the system does not carry a cost. However, investments in new developments in the system are normally paid for by its users. Additionally, forecasting agencies that use the system in their operational forecast process are normally expected to enter into a support and maintenance agreement with the software developer or other licenced supplier. These investments in the software and the support and maintenance agreements form the commercial exploitation model of the software. The code base of Delft-FEWS is currently not fully open source. Although several modules interfacing to the core of the system are already open source, the source code of the core of the system is as yet not open. It is expected that this will also shortly be available once a development and management process is implemented that guarantees the requirement of the operational forecasting agencies using the software that it is stable under all conditions.

3.2. Data import and storage

All operational forecasting systems require (real-time) data from hydrological and meteorological observation networks to be imported. This data is used for analysing the hydrological situation, and as the input to hydrological and hydraulic models that provide prediction of future hydrological variables. In most operational systems, data from several sources is considered, with different data networks typically using different formats for storing and publishing data (Horsburgh et al., 2009). Efficient import of data from these different sources poses a significant challenge, not only due to the variety of formats being used, but in many cases also due to differences in the meta-data provided. Delft-FEWS provides a data import module that has been designed to handle a wide range of data formats. Despite this, it is often found in new applications that data formats need to be parsed that are not yet included. In the original development of Delft-FEWS (De Roo et al., 2003), a data import module was available that then could be configured through various settings to support a new format. Although this worked to some extent, it was found that the range of formats was so diverse that the complexity of a configurable import module quickly became unmanageable. In the current generation of FEWS an alternative approach is now being used, where a dedicated Java class is developed for each (new) data format. This data source specific Java class is only required to parse the particular format, and then submit the parsed data to a generic data handling framework that forms part of the import module. This includes methods for mapping location and data type identifiers from the external data format definitions to the Delft-FEWS internal definitions, as well as other generic handling methods.

Adoption of standards in data exchange formats can greatly enable the ease with which new data is integrated, avoiding the need of developing additional Java classes for each new data source. This has been shown in England & Wales, where the adoption of a singular XML (eXtensible Markup Language) format across all agencies for data exchange has greatly simplified integration of different systems (Werner et al., 2009). Additional efforts by the hydrological community are resulting in emerging standards for exchange of hydrological data, such as the WaterML standard that is being developed by the WMO/OGC Domain Working Group Hydrology (Taylor, 2010). The Delft-FEWS import module already includes classes that can deal with most current and emerging standards. This allows data from systems that provide data in such standard formats to be readily imported. Within the meteorological community standards for data exchange are well established, such as GRIB, GRIB2, and the BUFR format, as well as more recently NetCDF with CF (Climate Forecast) conventions. Such standards have been widely adopted in the case of geo-spatial data, and data from such sources can be readily imported (Weerts et al., 2010).

Once parsed, all data is stored within the Delft-FEWS database. This contains both configuration data, including location specific information (e.g. coordinates and properties of forecasting points, map layers etc.), and dynamic time series data either imported from external sources. Dynamic data can additionally be produced by internal data manipulation methods or models run from within the system. In an operational environment large volumes of dynamic time series data are employed. This volume is rapidly increasing, particularly with the emergence of high resolution numerical weather predictions (De Roo et al., 2003), remotely sensed data products (Weerts et al., 2010) and distributed models used in operational forecasting (Jasper et al., 2002). Within the Delft-FEWS data model, time series are uniquely identified by their location and data type, as well as an *id* related to the source of the data (e.g. the external source or hydrological model of which the time series are a result). This source *id* allows multiple traces of the same data type to co-exist at the same location. This allows for easy support of multi-model ensembles. For a forecasting system the ability to identify data based on the time of forecast (the forecast origin) is important, and through this unique key Delft-FEWS can store and retrieve multiple forecasts that overlap in time. Additionally, all time series data are considered an ensemble (with the deterministic case simply being an ensemble with only one member), making the data model well suited to support the emerging concepts of ensemble forecasting (Cloke and Pappenberger, 2009; Schellekens et al., 2011). Time series data is additionally either scalar, vector, or gridded data, though all different types are uniformly stored as binary objects in a time series table. None of the functional components (including the linked models) have direct access to the times series table, which is accessible only through the data access module.

3.3. Data processing and manipulation

Most of the data that is imported from external sources is not at the appropriate temporal and spatial scale to be applied as an input to a forecasting model, or to be used directly in product generation. As a consequence, generic data processing steps form the predominant effort in most applications of models in the forecast environment. Some examples include data validation, serial and spatial interpolation, aggregation and disaggregation, and merging data. To support this, a core capability of Delft-FEWS is an extensive library of data processing functions (an overview of the available functions can be found in the web based documentation of Delft-FEWS[1]). This includes specific hydrological functions, such as transforming stage data to discharge, applying temperature lapse rates, and applying bias correction using an ARMA model. Examples of data processing steps found in applications of the system include:

- Quality control of rain gauge data imported from a real-time hydrological database, aggregation of 15 min totals to hourly totals, interpolation of the gauged data to a rainfall field using Thiessen polygons, sampling of the rainfall field with the catchment delineation for a lumped hydrological model. This provides inputs for the hydrological model based on observed rainfall.
- Conversion of an image of radar reflectivity using a z-R relationship to provide radar rainfall rate, sampling the radar rainfall field using the same catchment delineation as above (Werner and Cranston, 2009), sampling a gridded numerical weather product using the same delineation and merging these two products. This provides rainfall inputs for the nowcast/forecast period.
- Bias correction of a simulated hydrograph at the catchment outlet of a hydrological model using an ARMA error correction algorithm (Broersen and Weerts, 2005), scaling of the hydrograph to account for small tributaries not covered by the model, constraining boundary flows to defined minimum values. This provides the upstream boundary condition for a hydrodynamic model.

Most of the library of functions provided work equally on scalar and gridded time series. For complex spatial operations, Delft-FEWS itself does not provide support within its own code-base, but utilises an embedded integration of the PCRaster spatial processing engine (Karssenberg et al., 2010). This library of functions, including access to the spatial processing engine, is provided primarily through a module referred to as the transformation module. As with all other modules this communicates with the database solely through the data access layer (Fig. 2). Although the library of functions provided is extensive, in selected applications it is found that additional algorithms are required. For this a generic equation editor is provided. In case this is limiting, more complex algorithms can be developed as a new Java class coded to communicate with the application programming interface (API) provided.

3.4. Linking external models

The approach to the integration of models to be run as a part of the forecast process in Delft-FEWS has been chosen to be simple yet effective. Typically a forecasting process may use a cascade of models such as a snowmelt model, a rainfall-runoff model and a routing model. These models are often independent, with the forcing of each downstream model being the result of the model upstream of it. This means the models can be run sequentially, and independently, with data being passed to and from the database at each step in the model cascade.

In contrast to integration at the algorithm level as proposed in for example the OpenMI interface standard (Gregersen et al., 2007), the approach taken to integrating models is to run these as an external process. Delft-FEWS provides the required input data and parameters, executes the model, and reads the results. Over 50 model codes from a broad range of model developers and suppliers have been integrated to run from Delft-FEWS (an overview of the models integrated can be found on the web based documentation of Delft-FEWS[2]). The data formats of these models vary widely. In the original version of Delft-FEWS the approach for integrating external models was through a model wrapper that communicated directly with the Delft-FEWS database. However, as the number of models increased it became apparent that this approach was becoming increasingly complex. This complexity is found not only in the technical challenges of model integration, but as noted by Parker et al. (2002), communication in integrated modelling is an important issue, with the risk of misinterpretation of data and parameters. As the number of models increased it was found that such misinterpretation occurred more frequently. To reduce this

[1] http://publicwiki.deltares.nl/display/FEWSDOC/05+Configuring+the+available+DELFT-FEWS+modules.

[2] http://public.deltares.nl/display/FEWSDOC/Models+linked+to+Delft-Fews.

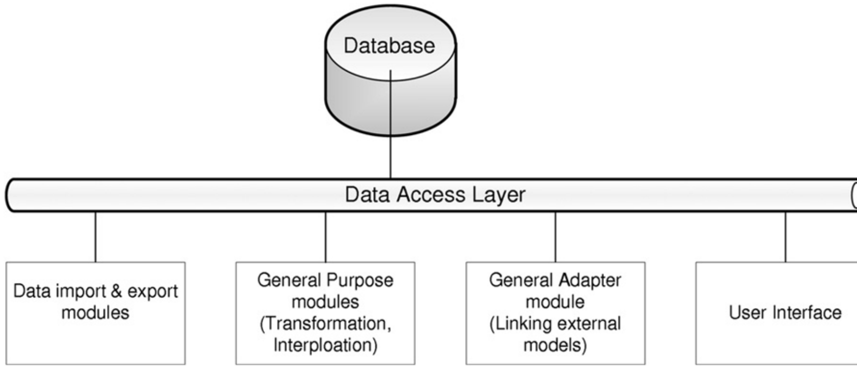


Fig. 2. Architecture of Delft-FEWS showing the data base, the data access layers and examples of functional modules that communicate through the data access layer.

complexity, Delft-FEWS now takes the approach of a well defined interface layer through which all communication with models passes. This interface is defined using the eXtensible Markup Language (XML). The advantage of this XML based interface is that all data (formats) exchanged can be independently verified using industry standard tools. The definitions of the information that can be exchanged with models is published in the open domain, and includes time series data (scalar, vector and gridded), model parameters, states, as well as meta-information and run diagnostics in the form of log messages. This approaches the concept of the open modelling framework proposed by Kokkonen et al. (2003), who also advocate the use of an independent XML interface to external models. All of the models that have been integrated with Delft-FEWS and are currently running in operational systems follow this approach. Delft-FEWS generates the input data as a set of XML files to a defined location; an adapter developed specifically for the model in question transforms this to the required native format in a pre-processing step; Delft-FEWS executes the model; and the adapter to that model then converts the native formatted results into XML formatted files in a post processing step. Delft-FEWS subsequently imports the results into the database from the XML files (see Fig. 3). Although there are variations on whether the model execution is done by Delft-FEWS or the model adapter, the principle is the same for all models. Exchanging data with the model is primarily through XML files. In some cases these XML files may become very large, which may lead to I/O bottlenecks and subsequent performance issues. Similar I/O bottlenecks may occur where several hundreds of small but independent models are to be run in sequence in a single forecast run. Options to improve the performance of the file-based exchange have, however, been introduced. This includes the use of binary-XML files, streaming files through memory, and the use of NetCDF-CFfiles.

In the development of the adapter, it is the preferred approach that the developer of the original model is the custodian of the code to the adapter. This ensures that the model–adapter combination will continue to work should internal formats of the model change. Additionally, when setting up a model to run from Delft-FEWS through the interface, a good understanding of the model itself is required. In most cases model specific configuration files will need to be configured for the model to run properly through the adapter.

The effort of developing an adapter for a model code not previously integrated with Delft-FEWS will vary depending on the complexity of the model I/O formats. Under the condition that a model code is suitable for running as a part of a real time forecast process, software development efforts have been found to be in the order of two weeks to a few months. The level of effort may be a little larger for distributed models, where the exchange of data typically includes both scalar and gridded time series, but this has not been found to be significant for the models currently integrated. The level of effort in configuring a distributed model to run as a part of the forecast process once the adapter has been developed is equally comparable to the level of effort in configuring a lumped model.

3.5. Data export and product generation

The final step of the forecast process is in most cases the generation of products that can then be disseminated to the warning process. Perhaps even more than the inputs from different data sources, products generated for further dissemination vary widely across different forecasting and warning organisations. Three main forms of product generation are supported. The first two follow a mechanism where products are generated by Delft-FEWS and exported. Delft-FEWS can generate web reports with graphs, tables as well as summary reports. These are generated based on HTML templates. Alternatively, Delft-FEWS can export time series in a variety of formats. This includes some of the existing standard formats such as XML (Werner et al., 2009; Taylor, 2010) and NetCDF-CF. As with the data imports, the export module provides a framework that can be extended to support additional export formats through a dedicated Java class developed to the software interface provided. In the third method external applications actively retrieve data from Delft-FEWS. The database allows (limited) access through a JDBC mechanism, as well as (more extensive) access through a web-services interface. These web-services not only offer access such as the reading of data from the database, but additionally can be used to post data to the database, as well as to request the system to run specified (forecast) tasks.

Products generated by Delft-FEWS may contain both deterministic forecasts, as well as probabilistic forecasts. Communicating probabilistic forecasts is becoming increasingly important in dissemination (Bruen et al., 2010). In graphs and tables generated within Delft-FEWS these can be shown as ensemble traces, statistical summaries, as well as summary information on the probability of exceeding selected thresholds. Alternatively, the ensemble traces can be exported as time series. This latter approach is taken in the Delft-FEWS application in Switzerland discussed in the example applications below, where forecasts are exported to a national common platform on natural hazards (Heil et al., 2010). This common platform will be one of the cornerstones for natural hazard mitigation in Switzerland in the years to come.

3.6. Running sequential functional steps in the workflow process

As outlined in the previous sections, the forecasting process is often a sequence of steps, starting with the import of data,

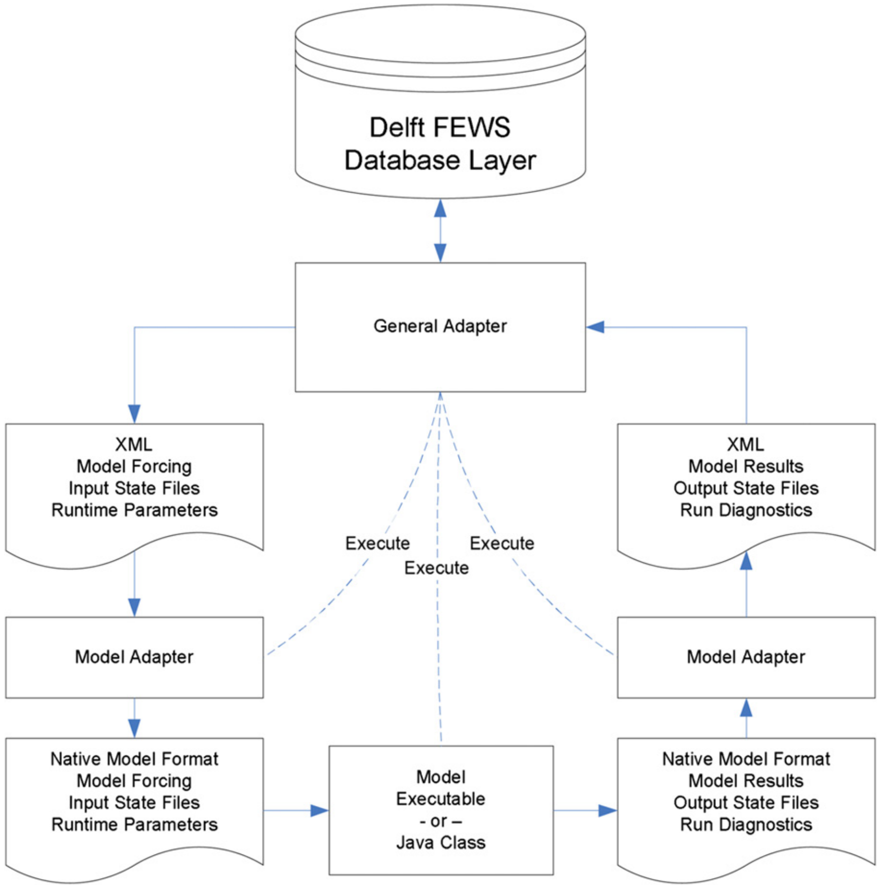


Fig. 3. Linking Delft-FEWS with external models. The figure shows the flow of data through XML and native model formats using solid lines, while executable commands are shown by dashed lines.

a number of data processing and modelling steps, and culminating in the generation of products to be disseminated to the warning process. In Delft-FEWS none of the functional modules have direct access to data except through the data access layer. There is also no direct communication of data between modules. Each of the steps follows the same pattern of retrieving required inputs from the database, applying a functional step, and returning data to the database for use in a subsequent step. The functional steps may employ an internal algorithm, or run an external process model. These steps are defined using a prescribed XML formatted configuration file, defining the input time series to be retrieved from the dynamic database, the parameters of the functional step, and the expected outputs (in the form of identifiers under which the output time series are to be saved in the database for later retrieval).

Logical steps in each forecast process are grouped in what is referred to as a *workflow*, which simply lists the sequence of functional steps in the correct order in which these are to be run. The granularity of these workflows can be as small as the individual functional step, or include sequences of tens or even hundreds of steps. A typical example of a sequence of steps in a workflow is shown in Fig. 4. Workflows themselves can also be nested. This can be done to structure the process, but can also allow a forecast process for a selected basin to be run individually, or as part of an over-arching workflow that runs multiple basins. This allows for defining a flexible modelling framework, as proposed also by Andrews et al. (2011). However, in the case of Delft-FEWS the third-party models may in some case be proprietary models and not open access as prescribed by Andrews et al. (2011).

When running sequential steps within a workflow in Delft-FEWS, each step is run for the full time window required (e.g. the

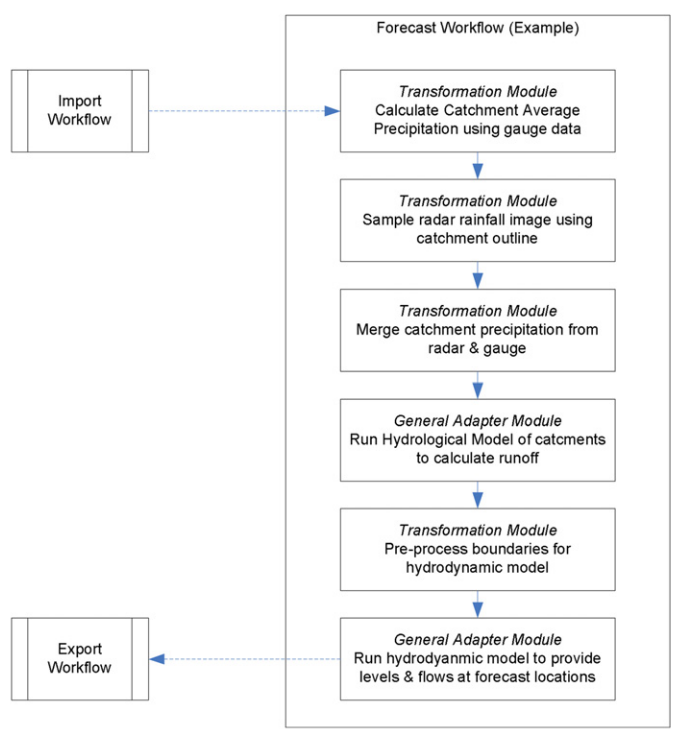


Fig. 4. Typical example of a workflow, showing the forecast process for a simple basin.

lead time of the forecast) before moving on to the next step. This precludes tightly coupling of models, even explicitly. The ability to run coupled models as a composition of models interacting via the OpenMI interfaces (Gregersen et al., 2007), however, permits more tightly coupled models to be executed by Delft-FEWS. To Delft-FEWS, this OpenMI composition is then seen as a single step in the workflow.

3.7. User interaction

In the day-to-day operation of an operational forecasting centre, duty forecasters interact with Delft-FEWS primarily through its user interfaces. With the interaction between the forecaster and the system in mind, the design of the user interface is focused on efficiently providing access to the large amounts of data that typically need to be consulted to guide the forecast process. This includes map oriented overviews of gauge and forecast stations with indications of alarm status through icons, thumbnail and full-size graphics, as well as dynamic displays of spatial fields such as outputs from numerical weather prediction models or dynamic inundation maps. Fig. 5 provides an example of the main display of the system showing locations displayed on the map and thumbnail graphics.

How the forecasters interact with the system through its displays has been found to vary quite significantly, depending very much on the forecasting procedures and the set-up of the Delft-FEWS system within the distributed environment. Werner and Janssen (2009) identified two main paradigms to forecaster interaction. The first of these, for which Delft-FEWS was originally developed, follows a relatively passive approach to forecasting in the operational setting. In this paradigm the model structure and parameters are established when setting up of the forecasting system. In real-time operation the models are then run with little or no interaction with the forecaster. Data assimilation techniques may be used to reduce forecast bias, but again there is no interaction from the user in operational use. The forecaster in this role typically monitors forecast runs that are scheduled at regular intervals, or may initiate additional runs as the need arises. The loosely coupled architecture of Delft-FEWS fits this paradigm well, and most operational environments that follow it are set up in a client-server mode, where the forecaster views data and forecast results on a client. All forecast tasks are carried out on a central server, with the results then synchronised to the (distributed) clients for viewing. In several cases the central servers may be located off-site, or even at centralised IT hosting services. This also allows for a dual or multiple server system to be established in duty-standby mode.

The second paradigm is different. Here forecasters will actively interact with models during the operational forecasting process. Parameters of the models, as well as data and model states may be

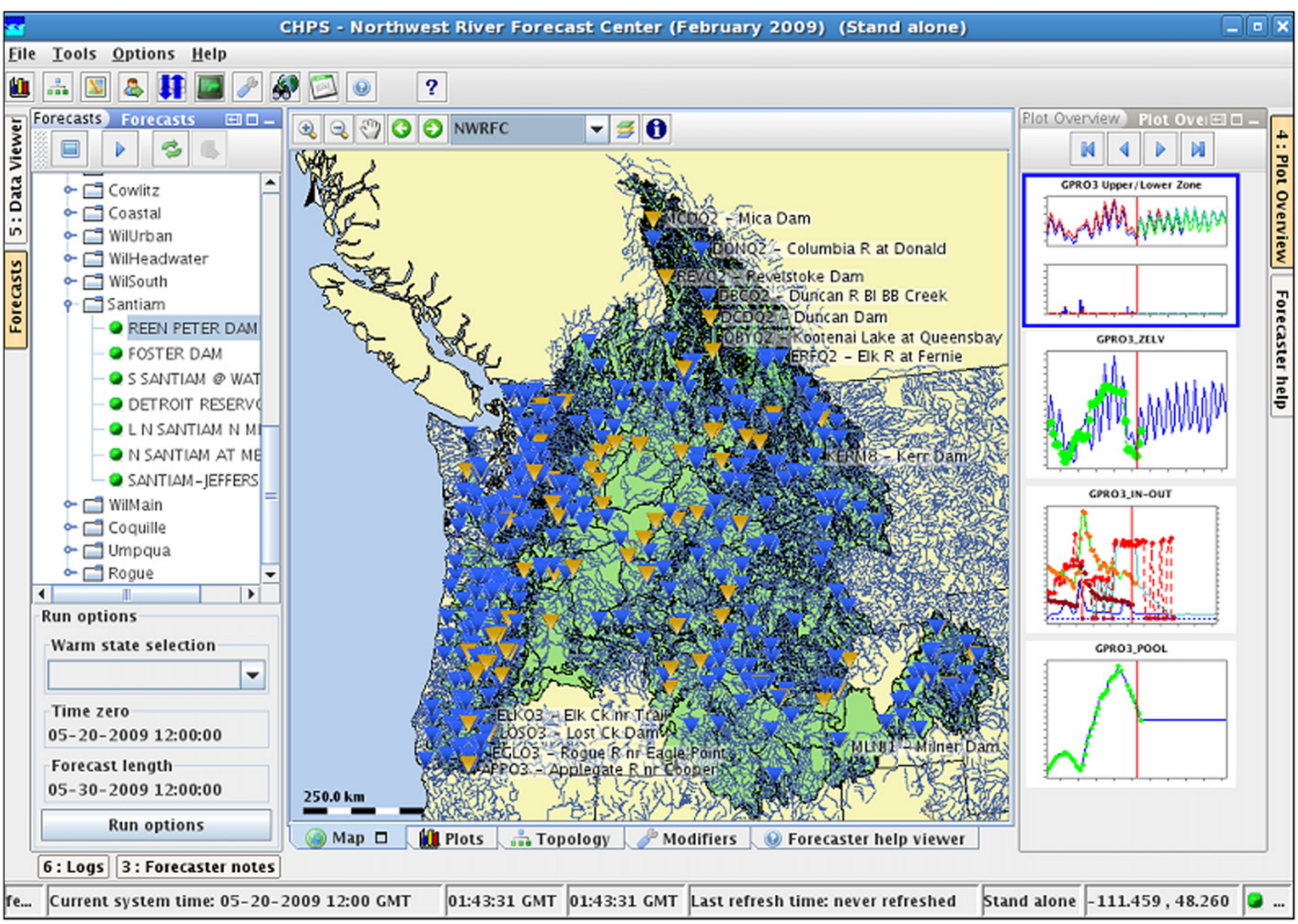


Fig. 5. Example of a user interface configuration of the FEWS system, showing the main display of the operational system of the Northwest Forecasting Center of the National Weather Service, USA.

amended in an interactive forecast process to reduce the bias in the forecast. Adjustments are made based largely on judging model outputs against observed data just prior to the forecast start time. This constitutes a form of data assimilation, but driven by the user rather than pre-defined algorithms. From the point of view of the architecture of Delft-FEWS this paradigm poses a larger challenge. When run in isolation in the single-user mode on the forecasters workstation (which is referred to as stand alone mode) this interaction is straightforward. However, in the client-server set-up with multiple users at the same time, issues of concurrency and upstream-downstream dependencies arise. Additionally, interacting with runs executed on a (remotely located) central server will make interactive usage difficult due to the longer response times. To reduce the overhead of communication, the approach taken is then to run models on the local user's workstation when in interactive mode. The final configuration including all forecaster specified amendments is subsequently run on the central server.

All changes made to models and parameters are stored together with each forecast run on the central server. This allows for an audit trail of the settings used for each forecast. Additionally, the forecaster may enter notes with each forecast to provide additional information should this be required. During an interactive session, configurations for each iterative run are available to that user, but will be deleted once the interactive session has been terminated.

There are several different intermediate forms of interaction. In some cases forecasters may only interact with specified types of data (e.g. projected releases received from reservoir operators), or more extensively with model parameters. In the different forecasting systems in which Delft-FEWS has been applied, such choices have been found to depend primarily on the forecasting procedures used by the particular operational centre, and less by technical constraints.

4. Applications

To illustrate the use of Delft-FEWS, three example applications are briefly discussed. These include a full-scale client-server set-up providing a forecast service at the national level, a smaller scale set-up in a basin, and selected applications of Delft-FEWS in research.

4.1. Flood forecasting in the UK: England, Wales and Scotland

Werner et al. (2009) describe recent developments in operational flood forecasting capabilities in England & Wales, where forecasting is the responsibility of the Environment Agency (EA). Developments in Scotland are also described. In Scotland forecasting is the responsibility of the Scottish Environment Protection Agency (SEPA). Although these two organisations are independent, there is similarity in the approach taken in operational flood forecasting. Seven regional forecasting agencies across England, together with EA Wales, are responsible for providing forecasts at fluvial, tidal and coastal forecasting locations. These forecasts are mainly provided to one of the 20 warning areas within each region (Werner et al., 2009). The warning is disseminated to the public and professional parties through offices in each of the warning areas. Historically, forecast capabilities were developed independently in many of the seven regions in England as well as in EA Wales. This led to a diverse set of capabilities, and many different models used operationally in delivering forecasts. These models range from hydrological and hydraulic models, through event-methods to simple regressions and spreadsheet based lookup tables. Major floods in 1998 triggered the development of a national approach to the delivery of forecasts. Adopting Delft-FEWS as its backbone, the National Flood Forecasting System (NFFS) (Whitfield, 2005) was established for the delivery of operational forecasts in a nationally consistent way. This became operational in three regions in 2005, with the remaining four and EA Wales becoming operational in 2006 and 2007. Although there is an objective to ultimately harmonise the approach to forecasting, the NFFS was designed to first integrate the diversity of the different models used operationally. This was done to ensure continuity of service, despite the change in the system through which forecasts were delivered. All models used historically were integrated using the open model integration approach provided by Delft-FEWS to run from the common platform. Since the introduction of NFFS the variety of models used has converged somewhat. More notable has been the increase in the number of catchments and river reaches covered by operational models (Whitfield, 2005), with a particular increase in the application of hydrodynamic models (Werner et al., 2009). Additionally, the use of Delft-FEWS as integrating platform has allowed data used across the regions to become increasingly uniform, while easing the introduction of new products such as (ensemble) Numerical Weather Predictions.

Although several of the EA regions had extensive forecasting capabilities prior to the introduction of the NFFS, forecasting in Scotland was very much in its infancy prior to the introduction of Delft-FEWS in 2006. Largely utilising the same models and data used also by the EA, SEPA has rapidly developed these capabilities – initially for selected catchments in the South-West of Scotland, but gradually expanding to cover all larger catchments across Scotland (Werner et al., 2009).

Both the EA and SEPA use the full client-server capabilities of Delft-FEWS, where the databases, modelling servers and central servers are hosted at central computing facilities. Users can log in from regional offices through the clients. Fig. 6 shows the distributed set-up of the system across the UK. This shows the centralised services which are hosted in Leeds and Peterborough for the EA in a duty-standby mode. In Scotland a single server is currently hosted at the central SEPA IT services in Stirling. Within the set-up of the EA, users can connect to both the Leeds and Peterborough servers, though only connections to Leeds are indicated in the figure for simplicity. Connections are also only shown from the locations of the regional forecasting centres, though users can equally connect from other (area) offices, as well as from any location with secure network access, including forecasters' homes.

Following extensive flooding in 2007, and subsequent recommendations from the Pitt review (Pitt, 2008), the EA has since joined forces with the UK Met. Office and established a National Forecasting Centre that complements the regional centres to provide advance warning across the England & Wales (Price et al., 2012). The centre employs the Grid-2-Grid model (G2G, Price et al., 2012) using a meteorological ensemble model to derive probabilistic forecasts in both gauged and ungauged catchments. A country-wide setup using the same G2G model approach is similarly being established in Scotland by SEPA.

4.2. Flood forecasting in Switzerland

Operational forecasting for the upper Rhine basin in Switzerland is provided by the Swiss Federal Office for the Environment (FOEN), from the forecasting centre in the capital, Bern. The upper Rhine basin drains most of the Northern Alps and poses significant challenges in operational forecasting due to the complexity of the hydrological and meteorological processes in mountainous terrain (Bürgi, 2002). The management of the multiple reservoirs in the basin, as well as discharge at the outlets of the larger Alpine lakes contribute further to the complexity.

In the current operational forecasting system used by FOEN, the Delft-FEWS system is used to integrate operational hydrological and meteorological data, meteorological forecasts, and the HBV

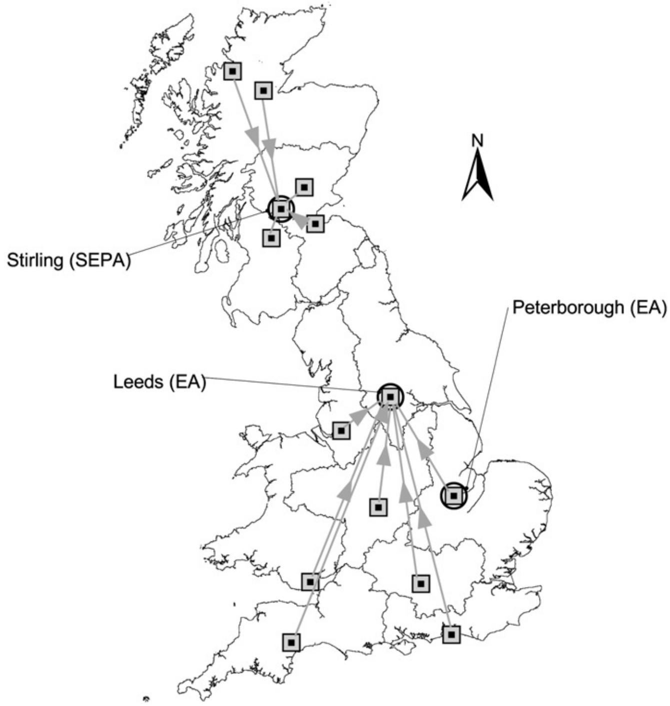


Fig. 6. Set-up of the National Flood Forecasting System in England & Wales (EA), and FEWS Scotland (SEPA). Squares show the location of regional offices from which clients may log on, while those enclosed with a circle indicate the location of centralised IT services.

conceptual hydrological forecasting model (Lindström et al., 1997). This is the primary hydrological model in current use. The Swiss Rhine basin is divided into 62 sub-basins, each calibrated to a gauge at the outlet, and routed through the river network using a simplified Muskingum scheme internal to the HBV model. Originally this model was established as a single unit (referred to in HBV as a district), running all 62 basins sequentially within HBV. Recently it was recognised that this did not fully utilise observed data in intermediate gauges, and the model was broken into multiple districts, utilising the ability of Delft-FEWS to run multiple model components in a forecast workflow. The hydrograph at a HBV-headwater district is now corrected using an auto-regressive error correction model (Broersen and Weerts, 2005) at the gauged outlet. The corrected flow is then passed back in to the HBV district downstream for routing to the next gauge.

With its location in the Alpine headwaters of the Rhine, the forcing derived from meteorological forecasts is an important source of uncertainty. This is recognised through running the same modelling chain using multiple meteorological forecasts. In day-to-day operation, deterministic forecasts are run using the COSMOCH2 model (0.02° resolution, 24 h lead time) (Zappa et al., 2008), the COSMOCH7 model (0.06° resolution, 72 h lead time) (Zappa et al., 2008), and the ECMWF Deterministic forecast (0.125° resolution, 240 h lead time). COSMOCH2 forecasts are updated every 3 h, effectively providing a time-lagged ensemble (Zappa et al., 2008). Additional to the deterministic forcing models, the 16 member COSMO Limited area Ensemble Prediction System (COSMO-LEPS, 0.0625° resolution, 174 h lead time) (Marsigli et al., 2005) is used. In the scope of the Mesoscale Alpine Programme Demonstration of Probabilistic Hydrological and Atmospheric Simulation of Flood Events (MAP D-PHASE) (Zappa et al., 2008), a 21 member multi-model ensemble assembled from deterministic forecasts made by contributing meteorological agencies across Europe (SRNWP-PEPS, Quiby and Denhard, 2003) was piloted. Its use has not been continued beyond the pilot for operational reasons. The use of these multiple forcing models in driving the same model chain does show the versatility of Delft-FEWS in dealing with multiple inputs and allowing results of each to be viewed independently. This is particularly so for the SRNWP-PEPS ensemble, where each of the members has a different spatial resolution, domain, lead time, and time step. Additionally, the size of the ensemble could be up to 21 members, but often varies depending on whether contributing meteorological agencies provide their input forecast(s) on time. Delft-FEWS utilises the self-describing properties of the GRIB formatted forecasts to identify the varying spatial resolution.

Recently FOEN has started with the integration of distributed hydrological models within the forecast modelling chain. In the Sihl and Linth catchments (see Fig. 7) the PREVAH model is used (Viviroli et al., 2009). These were previously running as separate local forecasting system for these catchments (Zappa et al., 2008; Addor et al., 2011). In the Emme catchment, the WaSIM-ETH distributed model (Schulla, 1997; Jasper et al., 2002) has been integrated. Although each follows different model concepts, both distributed models are built on a 500 m resolution digital elevation model. This resolution is quite a bit higher than the HBV models applied originally in these catchments. In the Sihl catchment the new setup also allows the division of the basin into headwaters flowing into a hydropower dam, with turbine flow being diverged after to an external basin (Lake Zurich). Spillage flows to the flash-flood prone areas downstream of the dam (Addor et al., 2011), and is then routed to the catchment outlet.

Although this marks a change to the modelling approach in these basins, the impact on the operational forecast process and on the delivery of products to the end users has been minimal. These new models simply replace the respective HBV districts in the forecast workflow(s). A drawback of using distributed models when compared to the conceptual HBV model is the increase of run-time. In the case of HBV the typical run-time for a forecast is in the order of seconds for a small basin such as the Emme (which in HBV is divided in three sub-basins). The run-time of the WaSIM-ETH model is in the order of 2–3 min. This will be the main difference to the forecaster's experience. In the case of the WaSIM-ETH model this run-time is influenced by algorithm as well as the amount of (gridded) output data that the model can provide. Limiting the outputs to include only those used in the forecast process can reduce the run-time. There are, however, no options in Delft-FEWS itself to influence the model algorithm. However, in the case of ensemble runs, Delft-FEWS does allow model runs to be executed across multiple nodes in parallel, which can reduce overall run-times.

4.3. Research applications

The two previous applications discussed the use of Delft-FEWS as a tool to support the real-time operational forecasting process. Although this has been the main design objective, the system has additionally found wide application in the research community. Renner et al. (2009) use Delft-FEWS to run an extensive set of daily hindcasts in the Rhine basin using two input ensembles. The objective of these hindcasts was to assess the quality of the ensemble forecasts, and how this varies with lead-time and basin size. Such extensive hindcasts can easily be carried out as the data model allows loading the database with several years of (ensemble) meteorological forecasts, and subsequently running the set of hindcasts as a batch process. Each hindcast run then selects the meteorological forecast at or before its own forecast start time, essentially ignoring 'future' forecasts already in the database. Similar hindcast exercises have been carried out in the White Cart

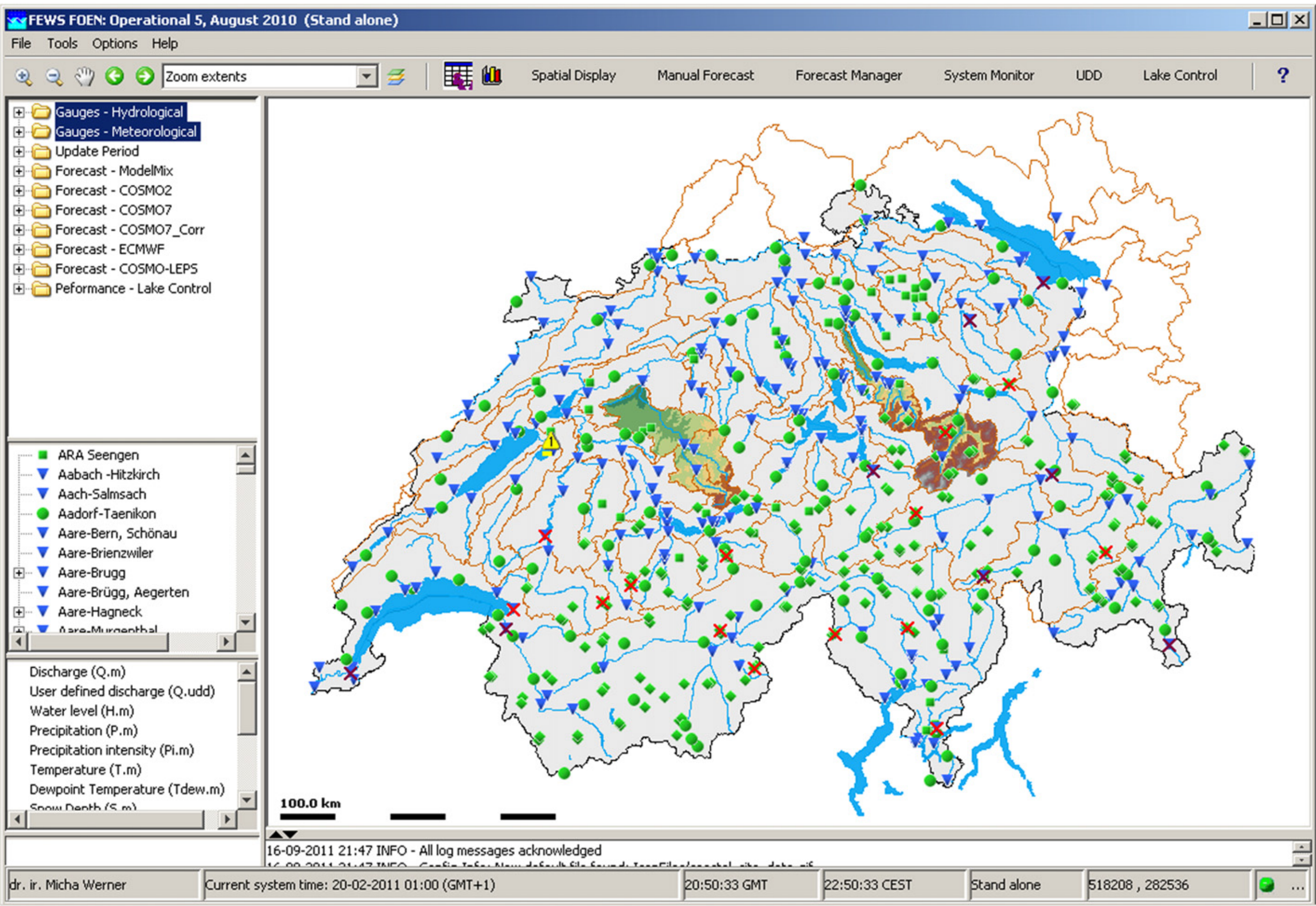


Fig. 7. Main map display of the forecasting system for the Rhine Basin in Switzerland. Inverted (blue) triangles represent the network of hydrological gauges, while (green) circles, diamonds and squares represent meteorological gauges from different agencies. The triangle with an exclamation mark indicates a threshold crossing, crosses indicate missing data at that station. Also shown are the digital elevation models for WaSIM model in the Emme Catchment (towards the West), and the PREVAH model in the Sihl and Linth catchments (towards the East). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Catchment in Scotland over a 15 year period (Werner and Cranston, 2009; Verkade and Werner, 2011).

Weerts et al. (2010) demonstrate how Delft-FEWS can be used in flexibly importing and processing different remotely sensed data products. These were then applied in (research) applications such as fire prediction and the assessment of climate change impacts. Delft-FEWS was also used in this context by te Linde et al. (2010), who ran continuous simulations for over 10,000 years with synthetic input time series re-sampled from station data over the Rhine basin (Beersma and Buishand, 2003). Sperna Weiland et al. (2012) have shown the use of Delft-FEWS in assessing the ability of climate models in simulating processes of the global hydrological cycle. Whilst many of these researchers have had some connection to the development team, others have used the system quite independently. Kuntiyawichai et al. (2010) applied Delft-FEWS in integrating the SWAT hydrological model with a hybrid 1D–2D inundation model code for research on inundation patterns in the Mun–Chi basin in Thailand. Corzo et al. (2009) applied Delft-FEWS in research on hybrid modelling approaches, combining data driven models and process based models.

5. Strengths and limitations

The two operational forecasting systems described briefly above are a very small sample of the range of operational forecasting systems using Delft-FEWS. The list of research applications is also by no means exhaustive. However, these examples do illustrate the diversity of forecasting processes and data-model integration issues supported. Such support has been one of the design philosophies from the outset. In many cases the choices made in establishing interfaces have been quite pragmatic and purposely kept simple, which has led to a very rapid development in the integration of modelling concepts. This is helped by the clear definition of interfaces using independently verifiable XML exchange files (see also discussion on the use of XML by Kokkonen et al., 2003). The incentive to develop adapters to make models compliant with Delft-FEWS has mostly been the need to integrate an existing model into the operational domain, thus avoiding expensive and time consuming model replacement. Some of the models that have been integrated are used in only one or two forecasting centres which could be seen as propriety development. However, in other cases the need for integration of a model is of benefit to many. The best examples of these are the HEC-HMS (Scharffenberg, 2003) and HEC-RAS (Brunner, 2002) model codes, which are used in several operational applications of Delft-FEWS. These two models are widely used, in particular in developing countries where there are insufficient means for acquiring more commercial model codes.

Although the flexibility of Delft-FEWS in integrating external data, processing and displaying data, and running models in the operational domain is a major strength, it is equally a weakness. The flexibility that is available in setting up the system, results in a broad range of choices that can be made. There is unfortunately not a single best way to configure a forecast process. To those users that have a simple forecast requirement, and perhaps limited experience in using models and data in real time, the broad palette of choices to be made may then be complex and quite daunting. In many cases users have also been found to have difficulty in identifying where the boundaries lie between Delft-FEWS and a coupled, yet independent model. Additionally, those setting up the system will need to deal not only with the configuration of Delft-FEWS and the link to external models, but also with the complexity of the external models themselves. Most models use a set of conventions quite different to those used in Delft-FEWS. Such diversity is difficult to resolve when allowing the open integration of different model concepts. Experience shows that some time is required to fully appreciate this complexity. The flexibility of Delft-FEWS can be benefited once that understanding has been established.

From the IT perspective, the development in Java has proven to offer several benefits. One such benefit is that the system is platform independent. Currently most operational centres operating the system use either a complete Windows operating system (both client and server) or a mixed system, with the main central server on a Linux operating system and clients and back-end model calculation servers using the Windows platform. There are also systems in use that use Linux both for servers and clients. Additionally, components such as the central database, as well as the application server used in the client server environment use software from industry standard vendors. This includes Oracle, Microsoft SQL Server and PostgreSQL for the database, and JBoss and Oracle Weblogic for the application server. This flexibility allows the system to integrate well in existing IT infrastructure environments.

6. Summary and outlook

In this paper the Delft-FEWS operational forecasting platform is presented. The objective of the system is not to provide forecasting capabilities in the form of hydrological modelling algorithms, but rather to provide the platform through which model codes can be brought to the operational domain. These models can then be linked with data from operational networks, as well as with the advances in related domains such as (probabilistic) meteorological forecasting. The structure of Delft-FEWS includes a data storage layer, a data access layer, as well as several components for importing, manipulating, viewing and exporting data. Although these components provide a range of tools required in using data and models within the operational domain, key to the open concept of the system are the open interfaces that allow integration of external models and algorithms. One of the most important of the open concept is the XML interface layer through which external models can be linked for use in the operational domain. This interface is relatively simple and has been applied in linking over 50 models. Most of these are used in operational forecasting centres, with some applied as yet only in the research domain.

The strong focus of using models to provide guidance in the operational forecasting process has been one of leading principles in linking external models and data through simple yet robust and easy to test XML interfaces. Through separating the models that provide the hydrological functionality from the process with which forecasts are made and disseminated, the forecasting methods used at operational forecasting centres can be more flexible. The separation reduces the impact when adapting to changing needs as well as to changing capabilities in models and data, as the operational forecast process will not need to be changed if there are changes to the underlying models. An additional benefit of the open approach is that existing forecasting procedures and models can often be integrated into the operational forecasting domain.

This is a clear advantage of the open approach to model integration. Not only is adapting to changing needs easier from the technical perspective, it is also easier from the organisational perspective. A gradual change process reduces the threat to continuity of service, obviously an important consideration in the domain of operational forecasting. This has been shown in the development of the National Flood Forecasting system across England & Wales, where existing models and procedures were first integrated into the new operational domain, and model replacement undertaken only after that. In the example of the use of Delft-FEWS in the Rhine basin in Switzerland, the gradual replacement of models used is shown to have little effect to forecasting procedures and the dissemination of forecast results. The flexibility in integrating models and data is additionally an asset to the research community. This is shown in selected research projects that have employed the strengths of the system in linking models and data, as well as in running extensive hindcast runs for evaluating forecasting quality.

Although flexibility has distinct advantages, it is found that this does result in an increase in complexity. There are often several options to reach a given goal, and the range of options can be difficult to oversee. To date development has primarily been driven by new applications of the system or to answer specific forecasting requirements, and the number of applications is increasing. The resulting increase in complexity has been recognised, and several efforts to reduce and manage complexity have been introduced. These are aimed primarily at developing interfaces through which external application can interact with the system, in particular web services interfaces. This fosters the modular approach, where from a technical perspective external models and applications can be integrated with the system, without influencing or compromising functions of the core.

In its brief history, the Delft-FEWS has been applied in over 40 operational forecasting centres, as well as in support of several research efforts in hydro-meteorological forecasting. This rapid growth can be attributed to the open approach allowing easy integration of models and data in the operational domain, as well as to the concept of sharing of new developments to the benefit of the user community. Such rapid development and widespread application does pose the risk of continuously adding complexity for growing and diverging needs. This may in time lead to a level of complexity that is unsustainable. However, the modular approach and the support of open data and model protocols that have been adopted by Delft-FEWS can help manage this complexity. It is argued that the modular and open approach adopted by Delft-FEWS are prerequisite in the sustainable development of environmental decision support frameworks used by operational organisations.

Acknowledgement

The authors would like to acknowledge the research and development team who have been instrumental in the development of Delft-FEWS. Additionally the development of the system is indebted to the support and collaboration of the many operational forecasters and researchers in the operational forecasting centres and research organisations that have supported and contributed ideas. These are too numerous to name individually. Dr. Massimiliano Zappa and two anonymous reviewers are thanked for their critical review and constructive comments. They have helped improve the manuscript considerably. Alex Minett is thanked for reviewing the use of English in the manuscript.

References

Addor, N., Jaun, S., Fundel, F., Zappa, M., 2011. An operational hydrological ensemble prediction system for the city of Zurich (Switzerland): skill, case studies and scenarios. Hydrology and Earth System Sciences 15, 2327–2347.

Andrews, F., Croke, B., Jakeman, A., 2011. An open software environment for hydrological model assessment and development. Environmental Modelling & Software 26, 1171–1185.

Argent, R., Grayson, R., Ewing, S., 1999. Integrated models for environmental management: issues of process and design. Environmental International 25, 693–699.

Argent, R., Perraud, J., Rahman, J., Grayson, R., Podger, P., 2009. A new approach to water quality modelling and environmental decision support systems. Environmental Modelling & Software 24, 809–818.

Bartholmes, Todini, 2005. Coupling meteorological and hydrological models for flood forecasting. Hydrology and Earth System Sciences 9, 333–346.

Beersma, J., Buishand, T., 2003. Multi-site simulation of daily precipitation and temperature conditional on the atmospheric circulation. Climate Research 25, 121–133.

Broersen, P., Weerts, A., 2005. Automatic Error Correction of Rainfall-Runoff Models in Flood Forecasting Systems. IEEE, pp. 963–968.

Bruen, M., Krahe, P., Zappa, M., Olsson, J., Vehvilainen, B., Kok, K., Daamen, K., 2010. Visualizing flood forecasting uncertainty: some current european EPS platformsCOST731 working group 3. Atmospheric Science Letters 11, 92–99.

Brunner, G., 2002. HEC-RAS, River Analysis System Hydraulic Reference Manual. Technical Report. Hydrological Engineering Center, US Army Corps of Engineers, Davis, CA, USA.

Bürgi, T., 2002. Operational flood forecasting in mountainous areas - an interdisciplinary challenge. In: Spreafico, M., Weingartner, R. (Eds.), International Conference in Flood Estimation. CHR, pp. 397–406.

Burnash, R., 1995. The NWS river forecasting system catchment modelling. In: Singh, V. (Ed.), Computer Models of Watershed Hydrology. Water Resources Publications, New York, USA, pp. 311–366.

Cloke, H., Pappenberger, F., 2009. Ensemble flood forecasting: a review. Journal of Hydrology 375, 613–626.

Corzo, G.A., Solomatine, D.P., Hidayat, de Wit, M., Werner, M., Uhlenbrook, S., Price, R.K., 2009. Combining semi-distributed process-based and data-driven models in flow simulation: a case study of the Meuse river basin. Hydrology and Earth System Sciences 13, 1619–1634.

De Roo, A., Gouweleeuw, B., Thielen, J., Bartholmes, J., Bongioannini-Cerlini, P., Todini, E., Bates, P., Horritt, M., Hunter, N., Beven, K., Pappenberger, F., Heise, E., Rivin, G., Hills, M., Hollingsworth, A., Holst, B., Kwadijk, J., Reggiani, P., van Dijk, M., Sattler, K., Sprokkereef, S., 2003. Development of a European flood forecasting system. International Journal of River Basin Management 1, 49–59.

Demargne, J., Mullusky, M., Werner, K., Adams, T., Lindsey, S., Schwein, N., Marosi, W., Welles, E., 2009. Application of forecast verification science to operational river forecasting in the U.S. national weather service. Bulletin of the American Meteorological Society 90, 779–784.

Dobson, C., Davies, G., White, W., 1990. Integrated real time data retrieval and flood forecasting using conceptual models. In: International Conference on River Flood Hydraulics. John Wiley & Sons, Oxford, UK, pp. 21–30.

Faber, B., Stedinger, J., 2001. Reservoir optimization using sampling SDP with ensemble streamflow prediction (ESP) forecasts. Journal of Hydrology 249, 113–133.

Gregersen, J.B., Gijsbers, P.J.A., Westen, S.J.P., 2007. OpenMI: open modelling interface. Journal of Hydroinformatics 9, 175.

Grijssen, J., Snoeker, X., Vermeulen, C., El Amin, M., Nur, M., Mohamed, Y., 1992. An information system for flood early warning. In: Saul, A. (Ed.), Floods and Flood Management. Kluwer Academic Publishing, pp. 263–289.

Haggett, C., 1998. An integrated approach to flood forecasting and warning in England and Wales. Journal of the Chartered Institution of Water and Environmental Management 12, 425–432.

Harvey, D., Han, D., Cluckie, I., 2002. A blueprint for next-generation modelling software. In: Hydroinformatics 2002, Cardiff, UK, pp. 1276–1281.

Heil, B., Petzold, I., Romang, H., Hess, J., 2010. The common information platform for natural hazards in switzerland. Natural Hazards.

Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. An integrated system for publishing environmental observations data. Environmental Modelling & Software 24, 879–888.

Jasper, K., Gurtz, J., Lang, H., 2002. Advanced flood forecasting in alpine watersheds by coupling meteorological observations and forecasts with a distributed hydrological model. Journal of Hydrology 267, 40–52.

Karssenberg, D., Schmitz, O., Salamon, P., de Jong, K., Bierkens, M.F., 2010. A software framework for construction of process-based stochastic spatio-temporal models and data assimilation. Environmental Modelling & Software 25, 489–502.

Kokkonen, T., Jolma, A., Koivusalo, H., 2003. Interfacing environmental simulation models and databases using XML. Environmental Modelling & Software 18, 463–471.

Krzysztofowicz, R., Kelly, K., Long, D., 1992. Reliability of flood warning systems. Journal of Water Resources Planning and Management 120, 906–926.

Kuntiyawichai, K., Schultz, B., Uhlenbrook, S., Suryadi, F., Werner, M., 2010. Appropriate Flood Mitigation Framework through Structural and Non-structural Measures for the Chi River Basin, Thailand. Mekong River Commission, Ventiane, Lao PDR, pp. 301–313.

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

Madsen, H., Butts, M., Khu, S., Liong, S., 2000. Data assimilation in rainfall runoff forecasting. In: 4th Hydroinformatics Conference, Cedar Rapids, Iowa.

Marsigli, C., Boccanera, F., Montani, A., Paccagnella, T., 2005. The COSMO-LEPS mesoscale ensemble system: validation of the methodology and verification. Nonlinear Processes in Geophysics 12, 527–536.

Matthies, M., Giupponi, C., Ostendorf, B., 2007. Environmental decision support systems: current issues, methods and tools (Preface). Environmental Modelling & Software 22, 123–127.

Moore, R., Jones, D., Bird, P., Cottingham, M., 1990. A basin-wide flow forecasting system for real time flood warning, river control and water management. In: White, W. (Ed.), International Conference on River Flood Hydraulics. John Wiley & Sons, Oxford, UK, pp. 21–30.

Parker, D., Fordham, M., 1996. Evaluation of flood forecasting, warning and response systems in the European union. Water Resources Management 10, 279–302.

Parker, P., Letcher, R., Jakeman, A., Beck, M.B., Harris, G., Argent, R.M., Hare, M., Pahl-Wostl, C., Voinov, A., Janssen, M., Sullivan, P., Scoccimarro, M., Friend, A., Sonnenshein, M., Barker, D., Matejicek, L., Odulaja, D., Deadman, P., Lim, K., Larocque, G., Tarikhi, P., Fletcher, C., Put, A., Maxwell, T., Charles, A., Breeze, H., Nakatani, N., Mudgal, S., Naito, W., Osidele, O., Eriksson, I., Kautsky, U., Kautsky, E., Naeslund, B., Kumblad, L., Park, R., Maltagliati, S., Girardin, P., Rizzoli, A., Mauriello, D., Hoch, R., Pelletier, D., Reilly, J., Olafsdottir, R., Bin, S., 2002. Progress in integrated assessment and modelling. Environmental Modelling & Software 17, 209–217.

Penning-Rowsell, E., Tunstall, S., Tapsell, S., Parker, D., 2000. The benefits of flood warnings: real but elusive, and politically significant. Journal of the Chartered Institution of Water and Environmental Management 14, 7–14.

Pitt, M., 2008. The Pitt Review: Learning Lessons from the 2007 Floods. Technical Report. Cabinet Office, London, UK.

Plate, E., 2007. Early warning and flood forecasting for large rivers with the lower Mekong as example. Journal of Hydro-environment Research 1, 80–94.

Price, D., Hudson, K., Schellekens, J., Moore, R., Clark, P., Harrison, T., Connolly, E., Pilling, C., 2012. Operational use of a grid-based model for flood forecasting. Water Management 165, 65–77.

Quiby, J., Denhard, M., 2003. SRNWP-DWD poor-man ensemble prediction system: the PEPS project. EUMETNET Newsletter 8, 9–12.

Renner, M., Werner, M., Rademacher, S., Sprokkereef, S., 2009. Verification of ensemble flow forecasts for the river Rhine. Journal of Hydrology 376, 463–475.

Rizzoli, A., Leavesley, G., Ascough II, J., Argent, R., Athanasiadis, I., Brilhante, B., Claeys, F., David, O., Donatelli, M., Gijsbers, P., Havlik, D., Kassahun, A., Krause, P., Quinnn, N., Scholten, H., Sojda, R., Villa, D., 2008. Integrated modelling frameworks for environmental assessment and decision support. In: Jakeman, A., Voinov, A., Rizzoli, A., Chen, H. (Eds.), Environmental Modelling, Software and Decision Support: State of the Art and New Perspectives. Elsevier, Amsterdam, pp. 101–118.

Scharffenberg, W.A., 2003. The Hydrologic Modeling System (HEC-HMS): Toward a Complete Framework for Hydrologic Engineering. ASCE, 172 pp.

Schellekens, J., Weerts, A.H., Moore, R.J., Pierce, C.E., Hildon, S., 2011. The use of MOGREPS ensemble rainfall forecasts in operational flood forecasting systems across England and Wales. Advances in Geosciences 29, 7784.

Schulla, J., 1997. Hydrologische Modellierung von Flussgebieten zur Absctzung der Folgen von Klimanderungen. Zürcher Geographische Schriften. ETH Zürich 69.

Sperna Weiland, F., van Beek, L., Weerts, A., Bierkens, M., 2012. Extracting information from an ensemble of GCMs to reliably assess future global runoff change. Journal of Hydrology 412-413, 66–75.

Taylor, P., 2010. OGC Implementation Specification 10–126: WaterML2.0: an O&M Profile for Water Observations Data. Technical Report OGC 10-126. Open Geospatial Consortium.

te Linde, A.H., Aerts, J.C.J.H., Bakker, A.M.R., Kwadijk, J.C.J., 2010. Simulating low-probability peak discharges for the Rhine basin using resampled climate modeling data. Water Resources Research 46.

Verkade, J.S., Werner, M.G.F., 2011. Estimating the benefits of single value and probability forecasting for flood warning. Hydrology and Earth System Sciences 15, 3751–3765.

Viviroli, D., Zappa, M., Gurtz, J., Weingartner, R., 2009. An introduction to the hydrological modelling system PREVAH and its pre- and post-processing-tools. Environmental Modelling & Software 24, 1209–1222.

Weerts, A., Schellekens, J., Sperna Weiland, F., 2010. Real-Time geospatial data handling and forecasting: examples from Delft-FEWS forecasting Platform/System. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 3.

Werner, M., Cranston, M., 2009. Understanding the value of radar rainfall nowcasts in flood forecasting and warning in flashy catchments. Meteorological Applications 16, 41–55.

Werner, M., Dijk, M., 2005. Developing flood forecasting systems: examples from the UK, Europe, and Pakistan. In: International Conference on Innovation, Advances and Implementation of Flood Forecasting Technology, Tromsø, Norway.

Werner, M., Heynert, K., 2006. Open model integration – a review of practical examples in operational flood forecasting. In: Gourbesville, P., Cunge, J., Guinot, V., Liong, S. (Eds.), 7th International Conference on Hydroinformatics, Nice, France, pp. 155–162.

Werner, M., Janssen, A., 2009. A discussion of paradigms in operational flood forecasting and their influence on forecasting system design and forecasting organisation. In: Proceedings of the 8th International Conference on Hydroinformatics, Concepcion, Chile.

Werner, M., Whitfield, D., 2007. On model integration in operational flood forecasting. Hydrological Processes 21, 1519–1521.

Werner, M., van Dijk, M., Schellekens, J., 2004. DELFT-FEWS: an open shell flood forecasting system. In: Liong, S., Phoon, K., Babovic, V. (Eds.), Proceedings of the 6th International Conference on Hydroinformatics. World Scientific Publishing Company, Singapore, p. 12051212.

Werner, M., Kwadijk, J., Schellekens, J., 2005. Flood early warning systems for hydrological (sub) catchments. In: Encyclopedia of Hydrological Sciences, vol. 1. John Wiley & Sons, pp. 349–364.

Werner, M., Cranston, M., Harrison, T., Whitfield, D., Schellekens, J., 2009. Recent developments in operational flood forecasting in England, Wales and Scotland. Meteorological Applications 16, 13–22.

Whitfield, D., 2005. The national flood forecasting system (NFFS) of the UK environment agency. In: International Conference on Innovation, Advances and Implementation of Flood Forecasting Technology, Tromsø, Norway.

Zappa, M., Rotach, M.W., Arpagaus, M., Dorninger, M., Hegg, C., Montani, A., Ranzi, R., Ament, F., Germann, U., Grossi, G., Jaun, S., Rossa, A., Vogt, S., Walser, A., Wehrhan, J., Wunram, C., 2008. MAP D-PHASE: real-time demonstration of hydrological ensemble prediction systems. Atmospheric Science Letters 9, 80–87.