# 11. FUNCTIONALITY OF SOFTWARE PACKAGES IN RELATION TO QUALITY ASSURANCE IN MODELLING

Peter J.G.A. Gijsbers
*WL | Delft Hydraulics, The Netherlands (WL | DH)*

## 11.1 QA topics to be supported by software packages

Nowadays, modelling is closely related to the application of software. Software codes can be applied in various contexts, e.g. as a research tool or as a production tool for consultancy studies. These differences also are reflected in the software product itself. Typically three types of software codes can be identified:

- Research code
- Public domain software
- Commercial software

Researchers often have access to the code and are able to modify it when needed. Public domain software is available as binary executables, so the code can not be modified by the user. Most codes developed at universities and by the US Government are put in the public domain. Many companies have made commercial products (user interfaces, databases etc.) around the public domain model code. Commercial software can be based on public domain model code embedded in a GIS and database environment. This typically is a commercial activity taking place in the US. In Europe, most commercial software is composed of privately owned model code embedded in a GUI and database environment. The latter situation often resulted in more advanced developments of model algorithms. Model development based on public domain software or commercial software often comes down to populating the database with appropriate data representing a model of the study at stake. In general, commercial software incorporates (vendor) support facilities, such as free demo versions, a helpdesk, etc. With respect to the assessment of software for QA purposes, the difference between research code, public domain software and commercial software is quite important.

### *11.1.1 Code verification*

As mentioned in chapter 3, code verification, is required to assure the quality of transforming a conceptual model into model code. As quality control in software engineering is well developed, various procedures are by e.g. IEEE. However, as these 'official' procedures require quite some effort, many model code developers are not willing to take all these efforts. As a minimum assurance, one should at least pay attention to:

- proper design, both functional and technical, preferably based on state-of-the-art software engineering paradigms such as Object Orientation, Component Based Development, using standard notation methods such as UML (Unified Modelling Language);
- proper implementation of this design into well structured model code, paying attention to error handling and prevention of code redundancy;

- code review by a software engineer or another IT-specialist which is not involved in the code development;
- extensive testing of the model behaviour under the various ranges of applicability, i.e. does the model behave as expected.

### *11.1.2 Reproducibility, data consistency and tracking, archiving*

Quality Assurance in modelling studies is highly related to the ability of a modeller to reproduce previous steps of a modelling process as well as to restore model runs being executed and archived. Preferably each set of model data at any step of the modelling process (input/output, intermediate or final, partially complete or not), can be traced and reproduced, using a reference which can be associated with the model journal.

To attain this objective, the software system should incorporate a mechanism for consistent management of data sets related to a specific step in the modelling study. A common name for this type of management is 'case management' or 'scenario management'. The tasks of this management system are:

- to preserve consistent sets of model input with associated model output;
- to enable identification and tracing of those data sets;
- to enable identification underlying model codes;
- to enable data storage and archiving.

As an organisation might run several projects or model studies, a hierarchical management structure is required in which the highest level is 'study/project' management. Each study has its own model journal and its own case management, i.e. case management is at the second level.

The challenge of product developers creating such data management systems is to create a flexible system which allows modifications in input data and naming of models and files, without loosing the consistency in the entire chain of data sets associated to a model run.

### *11.1.3 Quality control on model input*

Data collection for a modelling study always is a difficult issue. Often, the modeller is happy with any information available. Unfortunately however, model studies are sometimes taking place without thorough checking of the data. This check is of special importance to the data sets used for calibration and validation. Inconsistency in multiple-year time series (e.g. due to the move of a measurement station downstream of a confluence or intake) might not be noticed, while it prevents a model from being well calibrated. A modeller might come to the conclusion that the fit after calibration is bad, blaming the model, while in reality this bad fit might be caused by inconsistencies in the data set.

A modeller should at least perform some basic validation checks on temporal and spatial consistency of time series applied.

### *11.1.4 Quality control on model output*

Quality control on model output is a widespread issue, having relations to code verification (i.e. the model behaviour test), sensitivity analysis of the conceptual model and its implemented model code, model calibration, justification of site-specific model behaviour (i.e. does it represent the real world system behaviour) and uncertainty analysis. From a software perspective, calibration, sensitivity analysis and uncertainty analysis are closely related as they all deal with modification of the model input while assessing the resulting output with some kind of performance indicator. These issues will be dealt with in a separate section.

The standard means of quality control on model output is visual inspection, requiring generation of appropriate model output at various levels of detail and/or aggregation. In addition, modelling software might provide specific methods to perform standard tests such as balance checks, robustness tests, unit dimension tests and split sample tests.

An example for extensive balance checking is the so-called ‘fraction’ method. In this method, all ‘sources’ (inflow series) are ‘labelled’ (e.g. given a concentration of value ‘1’), and the matter is routed (as a conservative matter) throughout the system. At each location in the system, the fractions from the various ‘sources’ all should add up till value ‘1’. If not, some matter is lost by the model. This fraction-method is very appropriate in checking the validity of model schematisations as well as the computational algorithms.

### *11.1.5 Facilities for model calibration and uncertainty analysis*

Models need to be calibrated. Manual calibration as well as uncertainty analysis, often based on some ‘trial and error’ approach, is a tedious task, but in some cases the only way forward. Manual calibration is time consuming, not only because of its non-optimised search technique, but also due to the time span needed for one evaluation (from data modification to evaluation of outcome). Sometimes software developers improved this situation a bit by developing a GUI dedicated to calibration (i.e. providing quick access to relevant input parameters and model run activation, and presenting output in a format dedicated to calibration purposes).

In various modelling domains, automatic calibration tools, based on ‘classical’ parameter optimisation, are quite popular. Although model calibration only requires adjustment of input data sets, some calibration tools require modification of the modelling software source code. In case of R&D-oriented modelling software this might not be a big issue, as the model user often is the owner of the source code. However, the majority of the consultancy work is undertaken with modelling software, which is only available as a compiled binary component. In these situations, this modification is not possible, as the model user typically does not own the source code. In such situations, calibration software would be more appropriate if they could interact with modelling software at the data level (input/output).

Sometimes a ‘classical’ parameter optimisation approach is too limited in its capability to developed well-calibrated models addressing complex 3D-problems. In these circumstances, other techniques, such as inverse modelling, can be more appropriate [see e.g. Hill, 1998]. However, the mathematical problem at stake is such specific, that development of specific software tools is required. These tools can even be dedicated for a specific model software application at a certain study area.

### *11.1.6 Linking modelling software with HarmoniQuA-tools*

Four tools will be developed within the HarmoniQuA project, all to be available as stand-alone or plug-in version [see the Requirements Analysis report of HarmoniQuA (work package 2)]:

- A guideline tool;
- A monitoring tool;
- A report tool;
- An advisor tool.

Modelling software will have a close connection with the monitoring tool, i.e. the tool that records and stores all Modelling & Simulation activities. All information of a specific M&S job will be stored on a model record. Model records will be combined and stored in a model archive. Each model record contains in this way every activity of a model job and other tools can use this information.

It is foreseen that the link between the modelling software and the monitoring tool should at least accommodate exchange of meta-information, including references to model runs or model data sets in preparation. This link is associated with the ability to meet the requirements described in Section 11.1.2 on reproducibility, data consistency, tracking, and archiving.

## 11.2 **Review of functionalities of model codes within the respective domains**

### *11.2.1 Reproducibility, data consistency and tracking, archiving*

Commercial water related modelling software, and sometimes public domain software, generally accommodate some way of data management such as naming of models and model runs. Associated data files often are grouped in one (or a few) folders. However, control is often a weak point in case of input data modifications outside the main Graphical User Interface. This weakness partly occurs due to the dilemma to allow flexibility allowing modifications (e.g. during model preparation, or during trail-and-error analysis) while keeping a consistent and well traceable data set.

For instance, SOBEK, a software suite of WL | Delft Hydraulics for urban, rural and river systems (hydrology, hydrodynamics and water quality), contains a case management system which by some users is considered very strict (and therefore not considered user friendly). This product compares an internally registered time of modification with the time registered in the hard disk repository. Modifications outside the GUI will result in differences, that are flagged by the case management system.

Infoworks, a software suite of Wallingford Software Ltd. (rivers, water supply & distribution, waste water), provides management of model data and accommodates multiple users. Furthermore, it provides model version control and audit trail through a central 'master' database.

The MIKE suite from DHI Water & Environment (urban drainage, water supply & distribution) also contains features of scenario management.

WaterGEMS (water distribution modelling) by Haested Methods (US) contains a data and scenario management system.

Other widely used software packages are less clear in their scenario management capabilities, although the large ones do in some way account for it (e.g. US-based products such as SMS, GMS, WMS for respectively surface water, groundwater and watersheds, by Environmental Monitoring Systems, Inc.).

### *11.2.2 Quality control on model input*

Model calibration plays an important role in quality control on model input. However, model calibration is not the only means to control the quality of a model. Many software packages allow visual inspection and modification of model input. Automated validation checks are often included to take care that the model data fulfils the requirements of the computational core (e.g. is all required data specified?). Seldom however, automated validation checks are incorporated which can warn the modeller for 'garbage input'. The underlying reason is the difficulty to identify 'garbage-data' from 'no-garbage data', other than by basic checks on allowable ranges.

Some dedicated software packages (i.e. non-model packages) are available to check spatial and temporal consistency of monitored water related data which might be used as boundary conditions or for calibration purposes. Commercial software packages dedicated to this task in hydrological data validation and management are HYMOS (WL | Delft Hydraulics, NL), GANDALF (DHI Water & Environment), HYDATA (CEH Wallingford, UK), WISKI (Kisters AG, D) and HYDRON/TimeStudio (Hydron, AU). The latter two include fully equipped telemetry systems).

### *11.2.3 Quality control on model output*

Quality control on model output is typically done by visual inspection of simulation results. Many packages can provide both aggregated overviews as well as details on a time-step basis. Seldom, specific tools or methods are available to perform standard tests such as balance checks or robustness tests. The fraction-method that has been mentioned earlier in this chapter, which is an example of a balance check, is applied in software of WL | Delft Hydraulics, but it does not seem to be widespread in the world.

### *11.2.4 Facilities for model calibration and uncertainty analysis*

As model calibration is a time-consuming effort, many codes have been developed for this purpose. Again, a distinction can be made between research codes, public domain codes, commercial codes, combinations of these (such as PEST and MODFLOW) and dedicated calibration modules incorporated in commercial simulation packages (e.g. Infoworks, WaterGEMS, MIKE SHE, GMS). Typically these codes deliver one 'optimal' parameter set, without any notification that a variety of parameter sets might result in similar output.

## 11.3 **Analysis, conclusions and recommendations**

Many commercial software products provide some basic facilities relevant to QA. Each on its own way and each paying more attention to one than the other. Only few software packages accommodate a clear referencing system which can be used as the HarmoniQuA monitoring tool. Quality control tools for model output, other than visual inspection, seem to be the least well developed. Auto-calibration tools become more widespread, but have the disadvantage to provide an ‘optimal’ set of parameters, without indicating what other sets of parameters might also be suitable.

Research model codes seldom account for the type of code version control and data management needed to meet QA objectives such as reproduction of model results. Public domain software is somewhere in the middle. E.g. the computational algorithm MODFLOW is available in many software products where each of them has specific pro's and cons in relation QA aspects.

## 11.4 **Acknowledgements**

The present work was carried out within the Project "Harmonising Quality Assurance in model based catchments and river basin management (HarmoniQuA)", which is partly funded by the EC Energy, Environment and Sustainable Development programme (Contract EVK2-CT2001-00097).

## 11.5 References

Hill, M.C. (1998) *Methods and Guidelines for effective model calibration*. USGS Water Resources Investigations Report 98-4005, USGS, Denver, Colorado.

Web sites:

- www.wldelft.nl
- www.wallingfordsoftware.co.uk
- www.dhisoftware.com
- www.heasted.com
- www.scisoftware.com/products/pest_overview/pest_overview.html
- www.ems-i.com
- www.kisters.de
- www.hydsys.com