Prepared for:

RIKZ

Calibration tool for SWAN – Detailed Specification Calibration Instrument

Activity 8.3 of SBW project Waddenzee

Report

July 2007

H4918.65

WL | delft hydraulics

Prepared for:

RIKZ

Calibration tool for SWAN – Detailed Specification Calibration Instrument

Activity 8.3 of SBW project Waddenzee

Herman Gerritsen, Stef Hummel, Martin Verlaan

Report

July 2007

WL | delft hydraulics

Client:	RIKZ
Title:	Calibration tool for SWAN. Detailed specification calibration instrument. Activity 8.3 of SBW project Waddenzee

Abstract:

The spectral wave model SWAN plays a key role in the estimation of the Hydraulic Boundary Conditions (HBC) for the primary sea defences of the Netherlands. Since uncertainty remains with respect to the reliability of SWAN for application to the geographically complex area of the Wadden Sea, a number of activities have been initiated under the project SBW Waddenzee (project code H4803, continued under project code H4918), to devise a strategy for the improvement of the model. One aspect is providing a software environment for efficient and effective calibration of SWAN for Wadden Sea applications.
In 2006 an analysis was made of the programme of requirements for a calibration instrument and 2007 for the wave model SWAN (Activity 8.1 of SBW Waddenzee). In a subsequent study, Activity 8.2, a survey of available calibration techniques was made, leading to a selection of appropriate techniques for implementation in the Calibration Instrument. The present document presents the results of Activity 8.3: specification of the blueprint for the realisation of the Calibration Instrument for SWAN, following the conclusions and recommendations in (WL, 2006b; 2007a).

Following an introduction, the existing generically applicable software environments DATools, COSTA and UATools for data assimilation, calibration and uncertainty analysis are discussed. These are state-of-the-art developments, with many concepts and features that have already been tested and used in various applications. The present experience with three key calibration techniques, within and directly around the project team is presented and the selection of DATools / COSTA as the basis for the calibration instrument is further underpinned. Subsequently, the components and interfaces and further overall design aspects are presented. Furthermore, an overview of available, re-usable software elements is presented plus a prioritisation for realisation in the first implementation of the calibration instrument ("must have" functionalities). The relation between the calibration instrument and the validation instrument SHiVa, which is being developed in Activity 8.7, is discussed. Here, modules and interfaces used by both instruments are described, such as visualisation, statistical evaluations and the Goodness-of-Fit criterion. Finally, the approach to the testing of key elements with a simple prototype software set-up is summarised, to verify that the concepts are technically feasible and efficient.

References:			RIKZ contract RIKZ1797					
Ver	Author		Date	Remarks	Review		Approved by	
0.8	Herman Gerritsen		15 June 2007	Draft	J. Groeneweg		W. Tilmans	
0.9	Herman Gerritsen		29 June 2007	Revised Draft	J. Groeneweg		W. Tilmans	
1.0	Herman Gerritsen		26 July 2007	Final	G. Ruessink, G.Ph. van Vledder, J. Groeneweg		W. Tilmans	

Project number:	H4918.65
Keywords:	spectral wave models, model calibration, software design, DATools, SWAN
Number of pages:	57
Classification:	None
Status:	Final

Contents

List of acronyms and abbreviations

1 Introduction 1–1
1.1 The Hydraulic Boundary Conditions (HBC) 1–1
1.2 Strengths and Loads Water Defences Natural boundary conditions Wadden Sea (SBW RVW Wadden Sea) 1–1
1.3 Activity 8.3: Blueprint of the calibration tool for SWAN 1–2
1.3.1 Subdivision into activities A - D: 1–2
1.3.2 Products 1–2
1.3.3 Organisation of this report 1–3
2 Existing software elements, Calibration methods 2–1
2.1 The existing DATools, COSTA and UATools environments 2–1
2.2 Calibration methods; existing expertise 2–2
2.3 DATools-COSTA as basis for the Calibration Instrument 2–3
2.4 Parallel developments, synergy issues 2–4
3 Blueprint of the Calibration Instrument (C-I) 3–1
3.1 Key characteristics of the C-I 3–1
3.2 Components and interfaces in the C-I 3–1
3.3 List of software elements and classification for realisation (“must have” functionalities) 3–4
3.4 The Goodness-of-Fit optimisation criterion 3–10
3.5 Formulation of (soft) constraints on parameter ranges 3–10
4 Relations with the validation instrument (SHiVa) 4–1
4.1 Evaluation of simulation results against measurements 4–1
4.2 Post-processing and visualisation of simulation results against measurements 4–2
4.3 General statistical measures and Goodness-of-Fit 4–3
4.4 Use of the SWAN model wrapper 4–3
5 Testing of key aspects of the calibration 5–1
6 Summary and follow up 6–1
7 References 7–1

Appendices

A Note on ARCADIA and DATools A–1
B Implementation environment..... B–1
C Stochastic model for SWAN C–1
D Functional Requirements for the C-I..... D–1
E Overview of model parameters and input parameters..... E–1
F Prototype test environment (java code) F–1

List of acronyms and abbreviations

C-I	Calibration instrument
COSTA	COmmon Set of Tools for Assimilation of data; name of software environment
DATools	Data assimilation toolbox; name of software environment
DCSM	Name of hydrodynamic model application: Dutch Continental Shelf Model
DMI	Data Model Integration
DUD	Doesn't Use Derivatives (model calibration technique)
FEWS-NL	Flood Early Warning System - Netherlands
GoF	Goodness of Fit criterion
HBC	Hydraulic Boundary Conditions
HBV	Name of software code for rainfall-runoff modelling
HRT	Hydraulic Review Team for the SBW Wadden Sea study
HYMOD	Name of software code for rainfall-runoff modelling
HYMONA	Model development project at Rijkswaterstaat
MLM	Maximum Likelihood Method
OpenDA	Interfacing standards of DATools / COSTA with process models
PDF	Probability Density Function
REW	Representative Elementary Watershed (rainfall-runoff model)
RRSQRT	Reduced Rank Square Root Filter
SBW	Sterkte en Belasting Waterkeringen (Strengths and Loads Water Defences)
SC	Soft Constraint
SHiVa	SWAN Hindcast and Validation instrument
SOBEK-RE	Name of software code for river flows (version: River Engineering)
SOBEK-CF	Name of software code for river flows (version: Channel Flow)
SWAN	Name of software code: "Simulating WAves Nearshore"
TRIWAQ	Name of software code for three-dimensional hydrostatic flows
UATools	Name of software code: "Uncertainty analysis toolbox"
UOBYQA	Unconstrained Optimisation BY Quadratic Approximation (model calibration technique)
VTV	Voorschrift Toetsen op Veiligheid (Regulation on Assessment of Safety)
WAQUA	Name of software code for depth-integrated hydrostatic flows
WL	Delft Hydraulics

1 Introduction

1.1 The Hydraulic Boundary Conditions (HBC)

According to the Dutch "Law on the Water Defences" (Wet op de Waterkering, 1996), at five year intervals (2001, 2006, 2011, etc.), the primary water defences of the Netherlands need to be assessed against the required protection level as defined by the Hydraulic Boundary Conditions (HBC) and in the "Regulation on Assessment of Safety" (Voorschrift op Toetsen op Veiligheid, VTV, 2004). These HBC need to be reconfirmed or, if needed, redefined every five years by the Minister of Transport, Public Works and Water Management.

Especially for the Wadden Sea, uncertainty exists about the quality of the present Boundary Conditions. The origin of this lies in the fact that they have been obtained from an inconsistent set of measurements and design values (WL, 2002). For the remaining sections of the Dutch coast (Holland Coast and the South-Holland /Zeeland Delta) the wave transformation model SWAN is being used. The acronym SWAN denotes: Simulating WAves Nearshore (Booij et al., 1999; SWAN website: http://130.161.13.149/swan/).

Presently, insufficient measurement data exist to validate the wave model SWAN for the Wadden Sea. For this reason it is not yet possible to derive reliable and validated boundary conditions using this model for an area such as the Wadden Sea.

It is therefore also still unclear to which level processes such as the penetration of long waves plays or may play a role. For example, some storm measurements near the Emmapolder in Groningen have shown that swell can contribute substantially to the wave height (some 30%; personal communication F. den Heijer, RIKZ).

1.2 Strengths and Loads Water Defences Natural boundary conditions Wadden Sea (SBW RVW Wadden Sea)

The problem outlined above was the immediate trigger for the sub-project "Strengths and Loads Water Defences Natural boundary conditions Wadden Sea" (SBW RVW Wadden Sea) to request WL|Delft Hydraulics to formulate a Plan of Action with a strategy for answering the primary objective: "How to derive reliable Hydraulic Boundary Conditions for the year 2011 for the Wadden Sea area?" (WL, 2006a).

This Plan of Action is part of the project Strengths and Loads Water Defences (SBW, 2005) as this project has the task to improve the quality of the models and methods with which the HBC are derived such that authorities and other experts have sufficient confidence to use these instruments for the five year assessment. This is realised amongst others by conducting measurements in the project SBW - Field measurements, by developing methods and models for translation of the so-called deep-water statistics to statistics valid near the water defences and by considering the (increased) loads on the defences and the defences' fail mechanisms.

Thorough validation of the model SWAN for application to the Wadden Sea is another objective of SBW Wadden Sea. To enable efficient and repeated validation cycles for SWAN with different data, different conditions and also different versions of SWAN, it was decided to develop in four stages a flexible software instrument for model calibration. In 2006 an analysis was made of the programme of requirements for a calibration instrument for the wave model SWAN, to be used within the framework of SBW Waddenzee (WL, 2006a). In a subsequent study, a survey of available calibration techniques was made, leading to a selection of appropriate techniques for implementation in the Calibration Instrument (WL, 2007a). Activities 8.3 and 8.4 will provide a detailed specification of the Calibration Instrument and its software realisation including user documentation, respectively.

Based on the results of the Plan of Action, the SBW Waddenzee project is continued in 2007 according on the approved work plan and proposal (WL, 2007b), contract number RKZ 1797. The Delft Hydraulics project code for these activities is H4918.

1.3 Activity 8.3: Blueprint of the calibration tool for SWAN

The present document presents the results of Activity 8.3 (Sub-project H4918.65): specification of the blueprint for the realisation of the Calibration Instrument for SWAN, following the conclusions and recommendations in (WL, 2006b; 2007a), including a simple prototype to test key software elements.
The following items summarise the activities and products of Activity 8.3:

1.3.1 Subdivision into activities A - D:

Activity 8.3 comprises the following partial activities A - D:

A: Specification of the functionality of the calibration instrument in a detailed blueprint, making the distinction in “must have”, “should have” and “nice to have (but not in this release)” kind of functionalities;
B: Inventory of already available software elements for realisation of the calibration instrument;
C: Description of the exchanges / interfaces between the calibration instrument and SHiVa (SWAN Hindcast and Validation instrument;
D: Set up of a simple prototype to test the key elements of the calibration cycle.

1.3.2 Products

The products of Activity 8.3 are:
A report describing the results of the activities A – D;
A prototype test environment with limited functionality.

The “must have” functionalities form the basis for the first release of the calibration instrument, to be realized in Activity 8.4 (WL, 2007b). Examples of “should have” and “nice to have but not in this release” functionalities are: additional calibration techniques, extensions of the GoF functionality, and treatment of constraints.

1.3.3 Organisation of this report

In Chapter 2 the existing generically applicable software environments DATools, COSTA and UATools for data assimilation, calibration and uncertainty analysis are discussed. These are state-of-the-art developments, with many concepts and features that have already been tested and used in various applications. The present experience with three key calibration techniques, within and directly around the project team is discussed. The selection of DATools / COSTA as the basis for the calibration instrument is further underpinned *(result of activity 1.3.1-B)*. In Chapter 3 the components and interfaces and further overall design aspects are discussed. Furthermore, Chapter 3 presents an overview of available, re-usable software elements and a prioritisation for realisation in the first implementation of the calibration instrument *(result of activity 1.3.1-A)*. Chapter 3 also defines the specific form of the Goodness-of-Fit and Soft Constraints which will be implemented. The relation between the calibration instrument and the validation instrument SHiVa, which is being developed in Activity 8.7, is discussed in Chapter 4. Here, modules and interfaces used by both instruments are described, such as measurement storage and retrieval, visualisation, statistical evaluations and the SWAN model wrapper *(result of activity 1.3.1-C)*. In Chapter 5, the testing of key elements with a simple prototype software set up is described, to verify that the concepts are technically feasible and efficient *(result of activity 1.3.1-D)*.

A complete draft version of the report was compiled by dr. Herman Gerritsen, with extensive technical input by ir. Stef Hummel and dr. Martin Verlaan. The internal review was conducted by dr. Jacco Groeneweg. Two independent reviews of the revised draft version were conducted by dr. Gerben Ruessink (Utrecht University, Faculty of Geosciences) and dr. Gerbrant van Vledder (Alkyon), respectively. After having taking into account their comments and suggestions in a draft final version, the latter was reviewed by dr. Ap van Dongeren, upon which the present final version was prepared.

2 Existing software elements, Calibration methods

2.1 The existing DATools, COSTA and UATools environments

As basis for the calibration tool we will take the existing DATools software environment. DATools is a generic toolbox of data assimilation tools which can be generically and flexibly coupled to any process model for the purpose of enhancing model forecast (or hindcast) results or for model calibration. The concepts of such an approach have been generically tested, e.g. for reservoir modelling and atmospheric chemistry at TU Delft (M. Verlaan, personal communication, 2007). A first release of the DATools software in July 2006 is presently being used in applications for various types of geophysical processes and practical user experience is increasing. Presently, couplings of DATools have been realised with the river flow model SOBEK-RE, with SOBEK-CF, the hydrological models HBV and HYMOD, and with the hydrological model REW. So far, DATools applications with these models have focussed on the filtering of model results to improve operational forecasts. An example is the HBV-SOBEK rainfall-runoff modelling for the Rhine (Weerts and El Serafy, 2006). DATools is also used within FEWS as the module for operational forecast enhancement. An example of this is the FEWS-NL system, which is used at RIZA Arnhem for forecasting Rhine river flows, see (Warmink, 2007). More detailed descriptions of the DATools aspect in this system and application results are presently being reported (Weerts et al., 2007; WL, 2007c). A further description of DATools with an application of the Ensemble Kalman Filter applied to Delft3D-FLOW forecasts of a 3D stratified flows is described in (El Serafy et al., 2007). For real life calibration applications DATools has not yet been tested.

In parallel to DATools the COSTA environment for application of data assimilation tools is being developed at TU Delft and Vortech. As both model environments have similar aims, in 2005 the two groups decided to define a common architectural concept. Early 2007, activities were started to migrate to a common interface protocol (OpenDA) with process models. The COSTA group will now mainly focus on the development and testing of new data assimilation (and calibration) techniques, while the DATools group will focus on realising user friendly couplings and user tools for end users. Both will strictly adhere to the jointly defined OpenDA interface formats, to make the tools interchangeable. For the design of COSTA and successful application of its RRSQRT Kalman filter to a WAQUA/TRIWAQ based DCSM application, see (Van Velzen, 2006) and (Van Velzen and Verlaan, 2007).

One of the advantages or benefits of using DATools / COSTA as the basis for the Calibration Instrument over other existing environments (e.g. the TU Delft development ARCADIA), is the state-of-the-art of its software design, its methods and techniques, and the flexible and generic applicability to other models than SWAN. Examples are WAQUA, TRIWAQ, DELFT3D-FLOW, SOBEK. This is expected to lead to a larger user group, efficient learning curves, more efficient and more robust codes, and sharing of cost of further developments such as enhancements and migration. For a concise assessment of ARCADIA and DATools, see the note to the Hydraulic Review Team (HRT), Appendix A.

For uncertainty analysis of models, the UATools module/model has been developed. It has been designed using the same concepts as DATools. UATools and DATools share many components / routines, leading to efficiency and economy in development, maintenance and use. UATools is the basis for the uncertainty analysis in Activity 2.3 of the SBW Waddenzee project. For this reason, we will often denote the environment as DATools / UATools.

2.2 Calibration methods; existing expertise

Taking into account the functional requirements formulated in (WL, 2006b), in (WL, 2007a) three calibration techniques were selected that satisfy the requirements. In order of priority of implementation in the Calibration Instrument these are:

- DUD (Doesn't Use Derivatives) (Ralston and Jennrich, 1978)
- Downhill Simplex method (Nelder and Mead, 1965; Press et al., 1986)
- Powell's quadratically convergent method; UOBYQA (Powell 1964, 2002)

There is significant expertise within the SBW project team with the application of the first of these calibration methods. For over 15 years, the DUD method has been used within Delft Hydraulics for a range of process models. Examples are the parameter estimation of air-sea momentum exchange in storm surge models, estimation of bed friction in river models, and optimization of parameters for component in models of flow in closed circuits. An implementation of the DUD method is described in (Van den Boogaard, 2003). For a description and test results of use of DUD for model calibration within the FEWS environment, see (Weerts, 2004). An example of the application of the Nelder-Mead Downhill Simplex method is the calibration of SWAN 40.20 for the field cases Petten, Slotermeer and Western Scheldt (Alkyon, 2003). A Fortran implementation of Powell's UOBYQA method is available, but so far the project team has no pronounced user experience with this method. Colleague research groups at TU Delft have experience with Simplex and e.g. conjugated gradient based calibration techniques in a generic framework (Martin Verlaan, personal communication, 2007), so on all aspects relevant experience is easily accessible.

All three methods can accommodate hard and soft constraints. Soft constraints are expeditious in terms of implementation and the determination of uncertainties. Using a least squares criterion, a Maximum Likelihood interpretation allows the uncertainties in the estimated parameters to be efficiently quantified in the form of confidence bands, using the Jacobian and / or Hessian of the cost function (i.e. the calibration criterion).

In local search methods such as the ones mentioned above, it is important to start from a reasonable initial guess. This is understood to be an initial guess in the region of the solution space where one should be able to converge to the desired minimum and not get stuck in an unphysical local minimum, while the parameter value does not gradually move out of the physically realistic range during the calibration process. For the above reason it is recommended to perform a "pre-calibration". This may require a parameter initialization module, to determine such an initial parameter setting, with which the automated calibration cycle is then (re-)started.

2.3 DATools-COSTA as basis for the Calibration Instrument

As argued above, it will be efficient and effective to base the calibration tool on the data-assimilation environments that since 2003-2004 are being developed by Delft Hydraulics, Rijkswaterstaat and co-operating third parties. These data-assimilation environments are:
DATools/UATools - Data Assimilation Tools / Uncertainty Analysis Tools, Delft Hydraulics (further referred as DATools);
COSTA - COmmon Set of Tools for Assimilation of data, RWS and third parties.

As described in Section 2.1, the developments of both environments started independently from one other, but have recently adopted a common interface protocol (the so-called OpenDA standards) for coupling with process models. Besides, the COSTA and DATools development groups agreed on a largely complimentary focus in their intended user groups and activities, to enhance efficiency and effectivity, and avoid duplication of developments:

- The intended user group of the COSTA environment and utilities is the 'programming end user', in other words, the *developer* of filtering and calibration algorithms; in practice this will often mean users and developers in universities.
- The DATOOLS/UATOOLS environment targets the group of end users who want to *apply* existing filtering, calibration and uncertainty analysis methods on models that have been coupled or 'wrapped' to enable the essential data exchange with the DATOOLS/UATOOLS environment.
- The COSTA development group will focus its activities mainly on the data assimilation content and less on further developing the basic software environment.
- The DATools development group will focus its activities on providing a flexible end-user environment and typical user-friendly end-user functionalities, such as modules for prescription of uncertainties, visualisation, etc.
- The COSTA and DATools group will together ensure adherence to the joint OpenDA standards for interfacing with process models

Although DATools and COSTA are generic in the sense that the can be applied with any process model, each process model requires the one-time development of an interface to exchange data with DATools / COSTA.

To realise the above, a so-called SWAN "wrapper" needs to be realised once, which exchanges the data between SWAN and the DATools-Costa based calibration instrument. This wrapping is generic for the SWAN software in the sense that it is independent of the specific SWAN application and the SWAN version, provided the input and output routines and the number of parameters remain unchanged. The exchanged data consists of the model input and model results, which are communicated between DATools-Costa and the SWAN-wrapper by using XML-files. At the SWAN-side, a small piece of specific code has to be written to translate the contents of an XML-file into SWAN-input, and to translate the SWAN-output to an XML-file. At the DATools-COSTA side no action has to be taken, because the XML-files will be handled by the already available generic 'black box model wrapper' (see also Appendix C).

2.4 Parallel developments, synergy issues

In parallel to elaborating the blueprint for the calibration tool, the DATOOLS/COSTA compatibility has been studied in detail, and it has been sharpened where needed for efficiency and genericity (WL-DMI project DATools; RIKZ project HYMONA). Choices have been made regarding the various implementation languages (see Appendix B: Implementation environment), and some of the interfaces have been re-examined and upgraded. Especially the existing interfaces provided by the stochastic model and the (deterministic) model have been adjusted, albeit slightly. The reasons for this were the fact that DATOOLS up till now mainly has been used for filtering and uncertainty-analysis, while COSTA did not distinguish these two aspects of a model clearly enough.
The adjusted model interface is presently being implemented for the various DATOOLS and COSTA modules. This has no repercussions on the blueprint, and will only affect the prototype in a minimal way (the models will be completely dummy, instead of running a simple real model). The cost of these adjustments will be funded from a DELTARES DMI Synergy Project (X0370), which has recently started. These adjustments are among its very first project activities.

The advantage of this compatibility is standardisation and enhanced genericity of the system, which will result in increased use for a range of calibration and (operational) forecasting studies by a large user group. Besides WL and Rijkswaterstaat, this will comprise consultants developing and applying models for Rijkswaterstaat, TUDelft, and others. Spin-offs will be sharing of expertise, increase of functionality and documentation from other sources than SBW Wadden Sea, easier and cheaper maintenance, etc.

3 Blueprint of the Calibration Instrument (C-I)

3.1 Key characteristics of the C-I

Given the available environments and the recent DATOOLS/COSTA developments discussed in Chapter 2, the C-I will be characterised by:

- Using DATOOLS as configurable run time environment;
- Using the wrapped SWAN version for the interfacing DATools – SWAN; this wrapping is presently being realised and tested by HKV in commission of RIKZ, with overall project completion date 31 July 2007, and does not involve any cost to SBW Waddenzee;
- Using the DATOOLS/COSTA component definition for the interaction between components; this will be realised in the DELTARES DMI Synergy project by September 2007, at no cost to SBW Waddenzee project;
- Using COSTA and DATOOLS and further existing components where available;
- Using a least squares criterion for optimisation and implementation of soft constraints in view of easy and theoretically sound determination of estimates for the uncertainty around the optimised parameter values ("must have");
- Inclusion of a restart option, storage of selected intermediate and end results after convergence in order to allow a restart from selected intermediate or converged results with an adjusted set of run control settings ("must have").

3.2 Components and interfaces in the C-I

Figure 1 shows the main components that are present in the C-I. The yellow box represents the set of configuration files that will be read when starting the instrument. The (darker colour) top layer of the other boxes represents the DATOOLS/COSTA interface of the involved component, while the (lighter colour) lower layer represents the instantiated component that implements the interface.

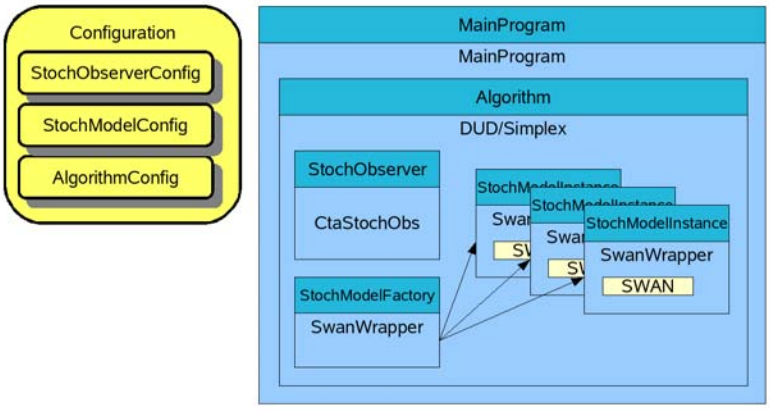


Figure 1: Main components in the C-I.

The functionality of the various components is as follows:

MainProgram

The main program reads the configuration file. The configuration file describes which observations will be used, which model (here read: SWAN) will be calibrated, which model application or applications (e.g. Amelander Zeegat) will be used, and which calibration algorithm will be initialised and used.

The main program instantiates a StochObserver component for reading the observations, and instantiates a so-called 'factory' for creating (stochastic) model instances (for SWAN read: SWAN model run files, with suitably chosen settings for the parameters to be calibrated) whenever a new run has to be performed. We note that for calibration, the model instances are not stochastic in the strict sense of the word, though. A calibration technique such as DUD does require a series of model instances, though, which will be evaluated systematically, to provide suggested parameter settings for a new model instance.
After that, the main program instantiates the component that performs the algorithm , i.e. the actual calibration, and provides it with handles to the observer and the stochastic model factory.

Algorithm

The (calibration) algorithm performs the actual identification of the optimal parameter settings for the parameters that have been selected for calibration. It lets the model factory create model instances (SWAN model run files) whenever it needs, after which it modifies the model instance's parameters based on the intrinsic algorithm's approach, starts the simulations and asks for the results. The algorithm uses the StochObserver component to retrieve measurements. See Figure 2 for more details on the calibration algorithm.

StochModelFactory / StochModelInstances

The factory creates the required stochastic model instances. In general, a stochastic model instance exists of a deterministic model (here: SWAN) and a stochastic model defining the prescribed uncertainties on the parameters (see Appendix C for details). However, the algorithm in the calibration instrument accesses each of these two subcomponents directly, as shown in Figure 2.

StochObserver

The Stochastic observer delivers the measurements. For calibration applications the measurements are often treated as non-stochastic, that is, as being exact and having no uncertainty. The DATOOLS/COSTA StochObserver, and each of its available implementations, assumes the measured values to be stochastic. The C-I includes the measurement uncertainty by prescribing the known or assumed mean plus standard deviation of the observations. This uncertainty enters in the least squares criterion in the calibration or optimisation procedure, see the next section.

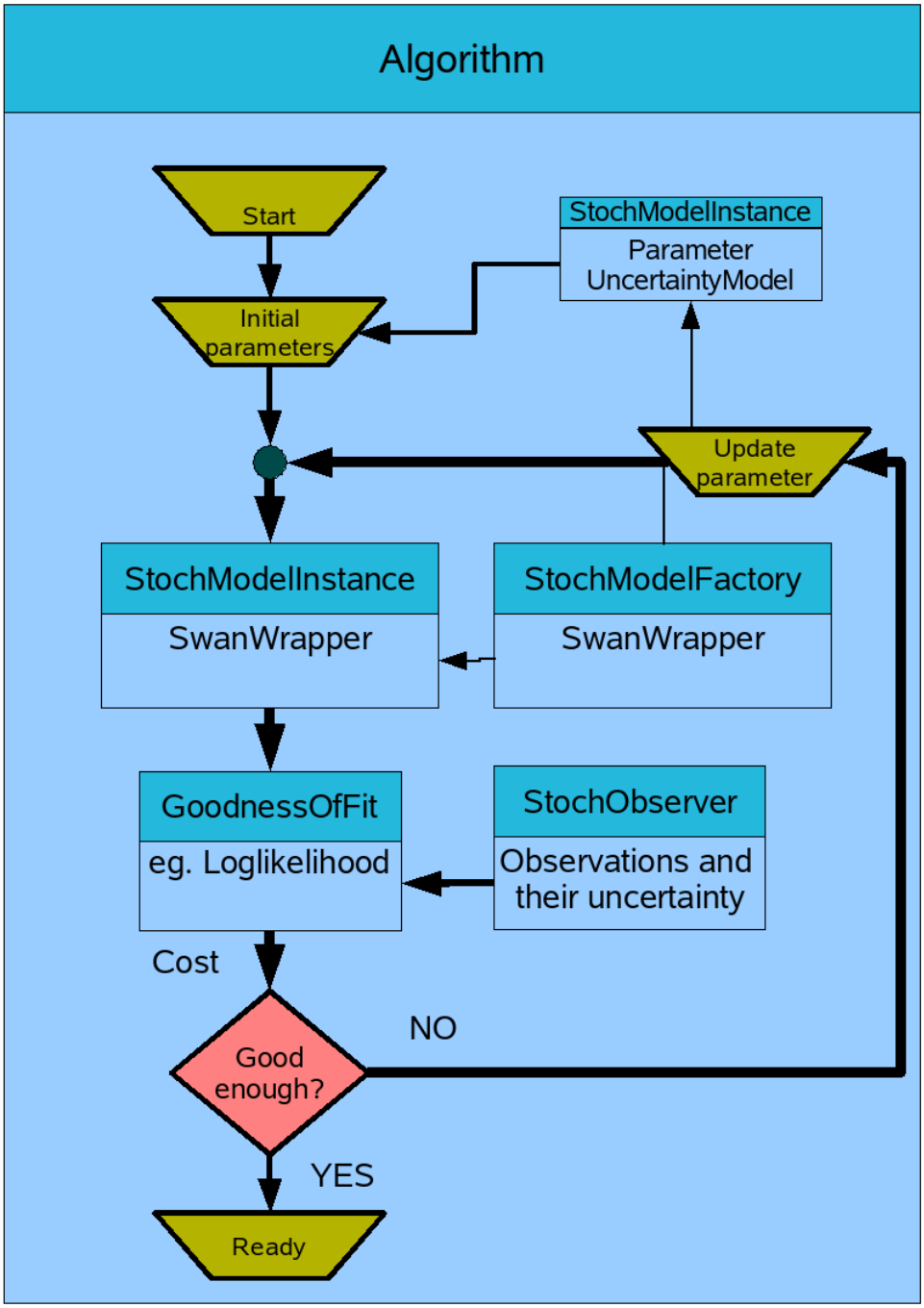


Figure 2: The main elements and dependencies in the calibration algorithm.

Inside the algorithm, the following elements can be distinguished, see Figure 2:

ParameterUncertaintyModel

This component specifies the probability density function (PDF) of the parameters that are being calibrated (if not known: assume Gaussian), the minimum and maximum range between which parameters can vary, etc.

The parameter uncertainty model will be extended with a *Calibration Initialisation Support Tool*, that generates a useful initial set of parameter realisations, by initially performing a number of model runs. See also (WL, 2007a). This *Calibration Initialisation Support Tool* is defined as a "must have" feature, see the Table below.

GoodnessOfFit

The GoodnessOfFit (GoF) module is a "must have". It defines and evaluates the cost function(s) that have been selected for the calibration, see Sections 3.4 and 3.5 below.
The definition of measurement stations, weights, and parameters that will be included in the GoodnessOfFit module will be specified in a configuration file, to allow for a flexible user configuration.

CalibrationCriteria (Good enough?)

The CalibrationCriteria module is a "must have", which defines and evaluates the termination criteria, like the maximum number of function evaluations (= SWAN simulations), the maximum change in the cost function evaluation and/or the maximum change in the parameters.
The CalibrationCriteria module will be made configurable for the user.

3.3 List of software elements and classification for realisation ("must have" functionalities)

For implementation of the first version of the calibration tool, the status of each involved component needs to be defined: whether it is already available and fit for use, and if not, from which existing component it can be derived, including specification of the required adjustments. Requirements and adjustments need to be categorised in steps that are labelled 'must have', 'should have' or 'nice to have but not in this release', where 'nice to have but not in this release' may involve the end user's usability point of view as well as the software maintainer's software design requirements point of view.

The table on the following pages provides this information, by presenting for each component:

- the component or component characterisation.
- the label of the functionality ("must have", etc.).
- the available implementation(s), and/or available software that can be used as start for the implementation.
- the work needed to create the required implementation of the component.
- the project which bears the cost of the implementation of adjustments.
- relevant remarks if any.

Component	Label	Available implementation(s)	Additional work / remarks	Project bearing the cost	Remarks
Main program	"must have"	DATOOLS Executive	Minor changes (extend with calling the calibration algorithm instead of the filter library. Disable the factory function if this is provided by Swan model wrapper)	This project	Limited adjustments.
Main Configuration	"must have"	DATOOLS ExecutiveConfig	Minor changes (extend configuration with specification of calibration algorithm to be used)	This project	Maximally configurable for purpose of SWAN calibration.
Algorithm, *DUD*	"must have"	DUD method from COSTA.	Transcribe to java.[1]	This project	Limited adjustments.
	alternative (not selected)	DUD from FEWS calibration module.	Isolate from FEWS (the data used by the calibration is exchanged with the FEWS database. This has to be replaced by accessing the *Model* and the *StochasticObserver)*	Not applicable	
Algorithm, Powell	"must have"	Public fortran implementation as reference	Implement	This project	Full implementation in java, including testing and user documentation.
Algorithm, Simplex	"must have"	Simplex method in COSTA.	Transcribe to java.[1)]	HYMONA	Realisation in September 2007
Confidence bands	"must have"	--	Determination of Jacobian / Hessian; extraction of diagonal elements; presentation	This project	Full implementation.

1. In the near future, COSTA will migrate its algorithms to java.

Component	Label	Available implementation(s)	Additional work / remarks	Project bearing the cost	Remarks
Parameter ranges	"must have"	--	Setting of allowable parameter ranges, formulation of associated constraints in the GoF calibration criterion	This project	Full implementation.
StochasticModel	"must have"	DATOOLS StochModel, configured for SWAN	Check interface (take care that implementation adheres to recently slightly adjusted DATOOLS/COSTA interface)	DATools project	Limited adjustments only.
	alternative option (not selected)	COSTA ModelCombiner, configured for SWAN	Check interface (take care that implementation adheres to recently slightly adjusted DATOOLS/COSTA interface)	Not applicable	
Model, *SWAN-Wrapper*	"must have"	COSTA BlackBoxWrapper, configured for SWAN	Check interface (take care that implementation adheres to recently slightly adjusted DATOOLS/COSTA interface)	This project	Limited adjustments, if any.
Uncertainty Definition	"must have"	UATOOLS uncertainty module	Check interface (currently used for ensemble filtering and uncertainty analysis; check against the requirements for calibration)	H4918.32 (SBW Wadden Sea sub-project Uncertainty analysis)	Adjustments and extensions.
Uncertainty Definition, DUE	"should have"	DUE[2]	GUI improvements required (and planned in DATools/UATools developments).	project Q3555, AQUASTRESS	Not in present version (not required for calibration).

1. 2. DUE stands for Data Uncertainty Engine. Is has been developed by James Brown (VU Amsterdam) and Gerard Heuvelink (Alterra), and has been used in the DATOOLS and UATools environment for defining uncertainty on measurements, model forces and model state. Some additional actions have to be taken to improve the usability. However, this does not affect the calibration instrument.

Component	Label	Available implementation(s)	Additional work / remarks	Project bearing the cost	Remarks
Stochastic-Observer	"must have"	Measurement data file definition defined in SBW sub-project SHIVA	DATools/COSTA StochObserver interface on top of SHiVa - measurements (see section 4.1). Flag Stochastic functions as 'not implemented', reducing the amount of work to a minimum.	This project	Implement StochObserver interface on top of SHiVa - measurements.
	alternative option (not selected)	DATOOLS stochastic observer	Convert VI-measurements to FEWS PI-timeseries. Not required now, because only stationary parameters/models will be calibrated.	Not applicable	
	alternative option (not selected)	COSTA stochastic observer	Convert VI-measurements to COSTA SQLite database. Not required now, because only stationary parameters/models will be calibrated.	Not applicable	
TreeVector	"must have"	DATOOLS ModelState, COSTA StateVector	The current data structures in DATools and in COSTA have to be adjusted in such a way that they implement the TreeVector interface.	This project	Only additional cost to SBW, basic functionality costed to HYMONA.
Calibration Initialisation Support Tool	"must have"	-	Develop Initialisation Support component (see (WL, 2007a) for required functionality)	This project	Required functionality is largely covered by the functionality of the DUD module; limited adjustments
Goodness Of Fit	"must have"	FEWS calibration module[3].	Isolate GoF configuration and evaluation-implementation from FEWS-module	This project	Likely full implementation; see Section 3.4.

3. FEWS is Delft Hydraulics' Flood Early Warning System (see www.wldelft.nl/soft/fews/index.html). It has a calibration module that offers a configurable choice for the Goodness Of Fit evaluation, and a configurable choice for the Calibration Criteria. It also has a Graphical User Interface to modify these settings (that are stored in XML-files).

Component	Label	Available implementation(s)	Additional work / remarks	Project bearing the cost	Remarks
(Soft) Constraints	"must have"	--	Implement constraints definition and evaluation	This project	Likely full implementation, see Section 3.5.
Stopping Criteria	"must have"	--	Implement criteria definition and evaluation	This project	Likely full implementation
	alternative option (not selected)	FEWS calibration module.	Isolate criteria configuration and evaluation-implementation from FEWS-module	Not applicable	
Parameters file and Result file definition	"must have"	Proposal by SHiVa validation instrument project.	Define file format/contents in co-operation with the validation instrument project (see section 4.1).	This project	50% cost to H4918.70[4] (SBW Wadden Sea sub-project Validation Instrument (SHiVa)), assuming SHiVa is not lagging behind C-I
Statistical Measures Component	"must have"	Developments in SHiVa validation - instrument project.	Develop module for statistics on results together with the validation instrument developers (see section 4.3 and Appendix D).	H4918.70 (SBW Wadden Sea sub-project Validation Instrument (SHiVa))	assuming SHiVa is not lagging behind C-I

4. This has been agreed with the project leader of the development of the Validation Instrument (SHiVA) sub-project (H4918.70)

In summary, the above Table presents for all C-I modules and functionality elements whether they are labelled “must have”, “should have” or “nice to have but not in this release”.

Moreover, options for implementation or adjustment of existing components are made clear and the selection of the specific option to be realised is indicated, plus an assessment of the relative effort expected to be involved. The selections were made based on experience and estimation of the suitability and efficiency of implementation of the options.

Equally, the table clarifies which parts of the implementation are justifiably attributed to the present project and which cost is borne by other projects since those projects require the functionality earlier or more specifically. In the latter cases, e.g. project AQUASTRESS, SBW sub-project SHiVa (H4918.70).

The table should be viewed as a specification of the elements in Figures 1 and 2, without going into the details of the actual implementation aspects.

3.4 The Goodness-of-Fit optimisation criterion

During the calibration cycle, the simulation results for different settings of model parameters (and boundary inputs) are evaluated using a Goodness-of-Fit (GoF) least squares norm, see also Appendix D. In terms of the spectral parameters mentioned in Appendix D, the following GoF is formulated:

$$GoF = \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{H_{m0}}\left(H^i_{m0,obs} - H^i_{m0,sim}\right)/\sigma_{H^i_{m0,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{Hm0}}^2} +$$

$$+\frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{T_{m-1,0}}\left(T^i_{m-1,0,obs} - T^i_{m-1,0,sim}\right)/\sigma_{T^i_{m-1,0,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{T_{m-1,0}}}^2} + \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{T_p}\left(T^i_{p,obs} - T^i_{p,sim}\right)/\sigma_{T^i_{p,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{T_p}}^2} +$$

$$+\frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{T_{m02}}\left(T^i_{m02,obs} - T^i_{m02,sim}\right)/\sigma_{T^i_{m02,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{T_{m02}}}^2} + \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{\bar{\theta}}\left(\bar{\theta}^i_{obs} - \bar{\theta}^i_{sim}\right)/\sigma_{\bar{\theta}^i_{obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{\bar{\theta}}}^2},$$

in which N_i is the number of spatial calibration data points (possibly summed over a number of applications), w^i_φ is a weight, and $\sigma^i{}_{\varphi_{obs}}$ the standard deviation of the observed parameter in the particular location, accounting for the uncertainty or quality of the respective observations. An appropriate directional measure would be the angle from the mean wave direction $\bar{\theta}$ (ensuring proper accounting for directional periodicity).

The formulation is a variation on the formulation in Appendix D, and brings it in the form required for a maximum Likelihood interpretation and associated uncertainty assessment. The weights can be used to exclude a specific parameter (e.g. T_{m02}), or to account for differences in spatial distribution of the measurement locations. The module for calculation of the GoF will be realised as part of the C-I activities, and will be shared with the validation instrument SHiVa.

3.5 Formulation of (soft) constraints on parameter ranges

During a calibration cycle in which a number of parameters are jointly optimised, individual parameters may gradually drift away from the range that is considered acceptable from a physical point of view. The variation of individual parameters can be constrained by adding a so-called soft constraint term (SC) to the GoF in the previous section. This SC term has the form:

$$SC = +\frac{1}{2}\sum_{p=1}^{P} w_p \left(\frac{\alpha_p - \alpha_p^{ref}}{\sigma_p^{ref}}\right)^2,$$

in which α_p is a parameter that is being calibrated, α_p^{ref} is an initial best guess for α_p, and σ_p^{ref} is a measure for the variation that is allowed in α_p^{ref}; w_p is a weight factor with value 0 or 1, to indicate whether a constraint for parameter α_p is applied. Furthermore, p = 1, …., P is the index for the parameter considered.

For the list of the SWAN calibration parameters with their default values, see Appendix E. It is noted that not all parameters need to be selected for a calibration cycle; in general, a sub-set will be selected, based on prior knowledge on behaviour and relevance.

4 Relations with the validation instrument (SHiVa)

The C-I has several relations with the validation instrument SHiVa (SWAN Hindcast and Validation instrument), in the sense that they share functionality requirements that still need to be developed, see the Table in Section 3.3. The part of the shared and jointly defined functionality which is fully realised within the sub-project Validation Instrument (SHiVa) is not included in the table of Section 3.3. This chapter describes the relations and the requirements for those functionality developments.

As shown below, relatively little additional work is needed to enhance the integration and mutual use of modules and data structures, which is beneficial for both the development stage and the later use of the C-I and the SHiVa.

4.1 Evaluation of simulation results against measurements

In SHiVa as well as in the C-I the simulated results have to be compared with, or rather, evaluated against independent information, generally in the form of measured values (the observations).
This means that in both SHiVa and C-I (e.g. in an evaluation of the results of the Calibration Initialisation Support Tool, or the results of a first stage of a calibration cycle) options must be available for:

- modules for selection and retrieval of required measurements from storage.
- modules for structured storage of measurements and simulation results.
- visualisation of results against measurements.
- statistical quantification of the results against measurements, or of results against results of an earlier model simulation.

In the generic DATOOLS/COSTA environment, the exchange of measured and simulated data is defined by specifying 'get-functions' as part of the interface of the observations component and the model component. However, to avoid that every piece of software *always* has to make use of these get-functions and that there is no way to access the data otherwise, it is useful to define the format and content of one or more file types that can contain these measurements. In this way, an external utility (e.g. a Matlab visualization routine) may also directly read the files.
Especially in the case of the SWAN calibration and validation for SBW Wadden Sea, where all data is spatially varying but time-independent (assumption in SBW Wadden Sea), it is quite convenient if these datasets can be stored in simple ASCII-files.

The C-I and SHiVa sub-projects have jointly defined and will use file types that contain (spatially varying) measurements. Wave measurements are performed at a number of locations (typically in the order of 10 for a field case). The following distinction among types of measurement data can be made:

- integral wave parameters. These parameters are scalar values. Examples are: significant wave height H_{m0}, peak wave period T_p, spectral wave periods $T_{m-1,0}$ and T_{m02} and mean wave direction $\overline{\theta}$.

- 1D spectra. A 1D energy spectrum consists of a table of energy density values versus frequency. In some cases, also the mean wave direction and directional spreading per frequency is derived from the measurements, e.g. when wave data have been collected with a directional waverider, see (Kuik et al., 1988).
- 2D spectra. A 2D spectrum can be considered as a rectangular array of spectral density values as a function of frequency and direction. It is noted that 2D spectra cannot be directly measured from directional waveriders or directional wave gauge arrays, but only reconstructed using e.g. MLM techniques.

Note the following:
- In most measurement locations, not all types of measurement data are available. In particular, directional information is not always available.
- The number of available integral wave parameters may differ per measurement location.
- The frequency axis (range and spacing) for the spectra may differ per measurement location.
- The frequency domain and computed frequency values may vary per measurement location. The software must be capable of dealing with this kind of generality.

SWAN can, as one may expect from a discrete computational spectral wave model, provide all of the above mentioned types of data in all computational grid points.

The realisation of the modules for selection and storage will be part of the activities to realise the SHiVa instrument. In the C-I the measurement file(s) will be accessed by a small component that reads these file(s) and provides the values according to DATOOLS/COSTA interface. In the C-I, the model results will be produced in the format required by the SWAN model wrapper. This format conforms to the standard method of providing SWAN results, as used in the SHiVa instrument.

4.2 Post-processing and visualisation of simulation results against measurements

In the C-I, the post-processing and visualisation of model results against measurements will be used (see WL (2006c), in particular Section 4.8). These are:
- The integral parameters, comprising significant wave height (H_{m0}), peak period (T_p), spectral wave period ($T_{m-1,0}$) and mean direction $\overline{\theta}$ are determined in the SWAN simulation for all grid points, and will be shown visually by means of 2D plots. Information on the high end of the spectrum is represented by T_{m02}.
- Visualisation of observed and computed spectra in all observational locations
- Visualisation of the spatial distribution of the observed and computed integral parameters in all observational locations
- Scatter diagrams showing computed against measured parameter values.

In the SHiVa sub-project, the post-processing modules and visualisation modules will be re-defined, adjusted and extended if needed. The SHiVa sub-project will bear the cost of this; the C-I will share these modules.

4.3 General statistical measures and Goodness-of-Fit

Both in SHiVa as well as in the C-I various statistical measures of the model run results have to be provided. The following statistical measures will be calculated for each model simulation result: RMS error (systematic and non-systematic parts), bias, spread, scatter index, correlation coefficients between simulation results and observations, indices of agreement d1 and d2.
Current implementations that generate these measures for SWAN are fully integrated in the existing software. They can be used by the C-I without adjustment; any adjustment considered necessary at a later stage will be realised as part of the SHiVa sub-project.

The very general form of the configurable Goodness-of-Fit criterion that is implemented as part of the present C-I sub-project, see Sections 3.3 – 3.5, is also shared with the SHiVa sub-project.

4.4 Use of the SWAN model wrapper

The SWAN wrapping was described in Sections 2.3 and 3.1. In the C-I, the results of SWAN simulations with different parameter settings are evaluated (using the GoF criterion) and based on this evaluation, one or more new sets are derived, after which the computation is performed with the new set(s), and the results are again evaluated. This process is repeated until some convergence stopping criterion is met. The SWAN model wrapper realises this exchange of parameter settings and retrieval of results between simulation model and C-I.

The newly proposed parameters are provided to the SWAN model wrapper as a DATools/COSTA TreeVector (see Appendix C). To facilitate the feeding of these parameters to the SWAN model wrapper, a simple ASCII-based straightforward file format will be defined for the parameters. This way, the SHiVa sub-project can make use of the SWAN wrapper as well. The implementation of the SWAN wrapper is prepared in commission of RIKZ and a version that has been shown to work for the provided SWAN test cases (Beji Battjes Bar and Friesche Zeegat) will be available to the SBW project by mid-August 2007. Preparation of possible remaining small adjustments on the DATools-COSTA side is the responsibility of the C-I, see Section 3.3.

5 Testing of key aspects of the calibration

As described in the previous chapters, the system design of the calibration instrument is based on the design of DATools and COSTA. Both these environments are proven technology by now, as shown above in Section 2.1. The same section describes that the DATools and COSTA environments are currently being unified, by making both adhere to the OpenDA standard.

Given the fact that the existing DATools and COSTA components are working well, there is one remaining question that has to be answered to be sure that C-I system design meets the requirements.

This question is:

With all required DATools and COSTA components available in an OpenDA compliant way, can the C-I indeed be implemented by using these components and there interfaces?

The best way to prove that this question can be answered affirmative, is by implementing one or more parts of the C-I, based on the interfaces.
Since the DUD calibration algorithm is one of the key parts of the C-I, this algorithm has been fully prototyped using the various interface-functions. The main part of this prototype code is listed in Appendix F.
To run this prototype code, every called function has been implemented in such a way that it reports its action to a logfile. So when the algorithm has run, the logfile contains all performed steps.

The answer to the question is:

Results of the DUD tests have shown that the system indeed performs as intended using these OpenDA compliant components and interfaces.

6 Summary and follow up

In the preceding Chapters a detailed specification or blueprint for the software realisation of the Calibration instrument has been given. The specification can be summarised in compact form as:

- An introduction with the argumentation why the development of a calibration instrument for SWAN within the SBW Waddenzee project will benefit efficient and structured parameter optimisation for SWAN in applications for the Wadden Sea (and, for that matter, elsewhere).
- A description of the existing DATools and COSTA software environments for data assimilation and model calibration and their use sofar in projects of RIKZ, Delft Hydraulics and TU Delft.
- An underpinning of the selection of the COSTA-DATools environment (which are presently being unified using the OpenDA interface standards) as a logical basis for the calibration instrument.
- A detailed description of the architecture, components and interfaces of COSTA-DATools.
- A list of the "must have", should have" and "nice to have but not in this release" functionalities, with the status of their development and what still needs to be done for their realisation, including information on parallel projects that contribute to realisation of some of the functionalities.
- The detailed formulations of the Goodness-of-Fit criterion in the optimisation routines and the soft constraints on parameter ranges
- A description of the required functionalities that are used jointly with the validation instrument SHiVa, including the information on the interfacing with the SHiVa subproject for the realisation.
- A description of the additional software tests with prototype code, performed to confirm that the proposed calibration process runs correctly with components and processes that conform to the OpenDA interface standards; this confirms the choice for COSTA-DATools as basis for the instrument.

The next step will be the software realisation and documentation of the calibration instrument. This is proposed as Activity 8.4 of SBW Waddenzee. This will contain the following activities:

A: realisation and software testing of the calibration instrument with three options for calibration methods (DUD, Powell's UOBYQA, Neldr-Mead Simplex method), making use of modules and functionalities that are developed in paralle in the SHiVA subproject;
B: preparation of a concise User Manual with examples (possibly a twin experiment);
C: conducting and concise reporting of two applications: (A) a simple model (triads; Beji-Battjes bar) and a field application (Friesche Zeegat).

With the completion of Activity 8.4 the calibration instrument will be completed. The instrument can then be made available for general use within the SBW project.

7 References

Alkyon, 2003. Calibration of SWAN 40.20 for field cases Petten, Slotermeer and Westerschelde. Alkyon Report A1168 prepared for Rijkswaterstaat/RIKZ, December 2003, 46 pages, 14 tables, 183 figures, 2 appendices (on CD-ROM).

Battjes, J. A., and Stive, M. J. F., 1985. Calibration and verification of a dissipation model for random breaking waves. J. of Geophys. Res., 90 (C5), 9159-9167.

Booij, N. n.d. ARCADIA Primer. Digital Hydraulics, 14 pages.

Booij, N. 1995. The ARCADIA System. User Manual version 30.04. Digital Hydraulics, July 1995, 53 pages.

Booij, N., R. C. Ris, L. H. Holthuijsen, 1999. A third-generation wave model for coastal regions, 1. Model description and validation. J. Geophys. Res., 104 (C4): 7649-7666.

Dingemans , M.W., 1997. Water wave propagation over uneven bottoms. Part 1 – Linear wave propagation. World Scientific. 471pp.

El Serafy, G.Y., H. Gerritsen, S. Hummel, A.H. Weerts, A.E. Mynett and M. Tanaka, 2007. The generically applicable data assimilation toolbox DATools – application in portable operational forecasting systems. Accepted for publication in Ocean Dynamics, May 2007, 27 pages.

Kuik, A.J., G.Ph. van Vledder, L.H. Holthuijsen, 1988. A method for the routine analysis of pitch-and-roll buoy wave data. J. Phys. Oceanography, Vol. 18, No. 7, 1020-1034.

Nelder, J.A. and R. Mead, 1965. A simplex method for function minimization. Computer Journal, 7, 308-313.

Powell, M.J.D., 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer Journal, 7, 155-162.

Powell, M.J.D., 2002. UOBYQA: Unconstrained Optimization By Quadratic Approximation. Mathematical Programming, B92, 555-582.

Press, W.H., B.P. Flannery, S.A. Teukolsky, W.T. Vetterling: Numerical Recipes. The Art of Scientific Computing. Cambridge University Press, Cambridge, 1986.

Ralston, M.L. and R.I. Jennrich, 1978. Dud, a derivative-free algorithm for nonlinear least squares. Technometrics, 20, 7-14.

Ruessink, B.G., D.J.R. Walstra and H.N. Southgate, 2003. Calibration and verification of a parametric wave model on barred beaches. Coastal Engineering, 48, 139-149.

SBW, 2005. Projectplan Sterkte en Belasting Waterkeringen. SBW-2005-003-P concept. December 2005 (G.A. de Raat).

SWAN website: http://130.161.13.149/swan/

Van den Boogaard, H.F.P., 2003. SUPERDUD – Technical Reference Document. Delft Hydraulics Research Report X0286, December 2003, 47 pages

Van Velzen, N., 2006. COSTA, A problem solving environment for data assimilation. Computational Methods in Water Resources, 18-22 June 2006, Copenhagen, Denmark.

Van Velzen, N. and M. Verlaan, 2007. COSTA a Problem Solving environment for Data Assimilation Applied for Hydrodynamical Modelling. Accepted for publication in Meteorologische Zeitschrift, Vol. 16, 2007, 10 pages.

Van Vledder, G.Ph., 1993. Evaluation of model performance. Methodology, examples, subroutines. Delft Hydraulics Research Report H1224, 42 pages, excluding appendices.

VTV, 2004. Ministry of Transport, Public Works and Water Management The safety of the primary water defences in The Netherlands; Regulations for Safety Assessment for the second round of assessments 2001 - 2006 (VTV). (De veiligheid van de primaire waterkeringen in Nederland; Voorschrift Toetsen op Veiligheid voor de tweede toetsronde 2001 - 2006.) January 2004, DWW-2004-009, ISBN 90-369-5558-0 (in Dutch). Note: An updated / revised version is due in 2006.

Warmink, J., 2007. Data assimilation in Delft FewsNL-Rhine. Delft Hydraulics Research Report Q4141, April 2007, 28 pages.

Weerts, A.H., 2004. Calibration tool. Implementation and use within FEWS. Delft Hydraulics Research Report Q3436, 21 pages, December 2004.

Weerts, A.H. and G.Y. El Serafy, 2006. Particle filtering and ensemble Kalman filtering for state updating with hydrological conceptual rainfall-runoff models. Water Resources Research, Vol. 42, W09403, doi: 10.1029/2005WR004093, 17 pages.

Weerts, A.H., G.Y. El Serafy, S. Hummel, J.D. Dhondia, H. Gerritsen, 2007. Application of generic data assimilation tools (DATools) for flood forecasting purposes. Journal contribution (draft version).

WL, 2002. Kwaliteit randvoorwaardenboek en kwaliteit SWAN. WL|Delft Hydraulics Rapport H4061, Januari 2002.

WL, 2006a. Uitvoering Plan van Aanpak SBW-RVW Waddenzee. Definitieve versie.

WL, 2006b. Gerritsen, H., A.J. van der Westhuysen, G. Ph. van Vledder, J. Groeneweg, G. Ruessink en H.F.P. van den Boogaard. Calibration tool for SWAN. Activity 8.1 and 8.2 of SBW project Waddenzee. Part 1: analysis of requirements. WL|Delft Hydraulics, Report X0346, 31 pp.

WL, 2006c. Wenneker, I. Validation instrument for SWAN. Operational requirements and structure. Activity 8.5 of SBW project Waddenzee. WL|Delft Hydraulics, Report H4803.80, 44 pp.

WL, 2007a. Van den Boogaard, H.F.P., G. Ruessink, H. Gerritsen, G. Ph. van Vledder, A.J. van der Westhuysen, J. Groeneweg. Calibration tool for SWAN. Activity 8.2 of SBW project Waddenzee. Part II: Selection of appropriate calibration techniques. WL|Delft Hydraulics, Report X0346, 54 pp.

WL, 2007b. Van Dongeren, A. Aangepaste offerte in antwoord op offerteaanvraag RIKZ1797, April 2007, 78 pages.

WL, 2007c. Evaluation of FewsNL-Rhine and the forecast enhancement using DATools. Delft Hydraulics Report for RIZA (in preparation).

A Note on ARCADIA and DATools

The text below is a full and unchanged copy of a note on the ARCADIA software, prepared for the Hydraulic Review Team in April 2007.

On 28 March 2007 the Hydraulic Review Team met with the SBW Waddenzee project leaders, discussing progress and plans for 2007 of the project.

The provided materials on Activity 8.3 – Calibration Instrument gave rise to the following specific question by the HRT:

Why did the project team not consider the existing ARCADIA software as a basis for the calibration instrument?

The present two-page note answers this question and assesses the consequences.

Answers:

1. The proposal stage:

We regret that no one in the project team for the calibration instrument was aware of ARCADIA as a candidate tool (ARCADIA is not used at WL Delft or Alkyon).

2. Follow up actions:

ARCADIA has been discussed with Dr. Sofia Caires, who has used ARCADIA when working on her master thesis and jointly with Nico Booij and Leo Holthuijsen in the period 1995 – 1996. She provided detailed information on ARCADIA in the form of the Primer[5] and the User Manual[6]. Furthermore, the project team has contacted Dr. Nico Booij, author of ARCADIA to discuss the background and present status of ARCADIA.

3. The main characteristics of ARCADIA:

ARCADIA assists the user by providing a software environment for efficient execution of repeated model runs, to generate the necessary input files for these runs, and provides desired postprocessing. This functionality is not part of the "calibration tool" we are after.
ARCADIA has modes for sensitivity analysis (linear variation), calibration, and uncertainty analysis (called "probability transformation"). This is the functionality of interest here.
ARCADIA was designed for use with HISWA and/or DOLPHIN, but can also be used for SWAN, DUCHESS, or flow models. Its user base is small.
ARCADIA is FORTRAN-77 based, and was designed in the end of the 1980's. It is batch-oriented; with tabular and HPGL-type graphical output. There have been no developments since the mid-1990's.

[5] N. Booij, ARCADIA Primer. Digital Hydraulics, (n.d.), 14 pages.
[6] N. Booij. The ARCADIA System. User Manual version 30.04. Digital Hydraulics, July 1995, 53 pages.

In calibration mode, ARCADIA applies linear perturbations to derive a set of linear equations, which are solved by a direct method (Gauss elimination) to minimise a least squares norm. This normalized least squares norm contains the measurement information. It has weight factors which can account for uncertainties. It can also deal with (soft) constraints.
The search algorithm finds a local optimum which is an approximation; for non-linear models such as SWAN iterations are required; various initial parameter sets may need to be investigated in order to avoid missing the actual (global) optimum. ARCADIA does not provide support facilities for this, such as a quick scan option.

4. Assessment of ARCADIA as calibration instrument for SBW Waddenzee:

ARCADIA presently provides many but not all of the functionalities required for the calibration instrument, as described in the report on Activity 8.1: Analysis of requirements[7].
The selection of the present ARCADIA system as calibration (and sensitivity analysis) environment within SBW Waddenzee does not satisfy the SBW requirements, because:
Substantial extensions would need to be realised, notably regarding the definition of constraints, more generic user controlled least squares criteria, and in multiple optimisation algorithms to ensure sufficient robustness for use with the highly non-linear SWAN model in the Waddenzee.
The result would be a largely "fixed" instrument, which would effectively be tailored to its intended use with SWAN.

5. Assessment of the proposed DATools as calibration environment for SBW Waddenzee

The present DATools provides the key functionalities for the calibration instrument and for the sensitivity analysis required in SBW Waddenzee.
DATools is already used by various users within WL Delft. DATools is intended as a strategic software instrument to be used by users in WL Delft, RIKZ and by consultants; it will also be used in calibration studies for model applications based on Delft3D-FLOW, SIMONA, SOBEK, etc. The benefits for SBW are a wider user base, steeper learning curve, more extensive testing and effectively a higher robustness.
DATools (/UATools) has been designed as a sensitivity / calibration and uncertainty analysis environment following present day programming concepts and languages.
DATools also includes data assimilation functionalities such as Ensemble Kalman Filtering, particle filtering, etc. to improve operational forecasting results.
A computationally efficient and generic coupling with SWAN is presently being realized without cost to the SBW Waddenzee project.
Through its coupling with the open source COSTA content, DATools can flexibly use an increasing range of optimization routines (DUD, Powell's method, classical Simplex / Nelder-Mead, and others). Effort in programming these is not necessary.
In various other projects, DATools is continually being developed further and extended with new functionalities, which is of direct benefit to SBW Waddenzee.

[7] WL. Calibration tool for SWAN. Activity 8.1 Part 1: Analysis of requirements. WL report X0346, 2006, 28 pages.

6. Overall conclusion

Based on the arguments above it is concluded that a DATools-based calibration instrument is preferred, rather than ARCADIA which would need effort to modernize and redesign (both regarding the solvers and the IT structure). Rather, we will put our efforts into a code which is fit for purpose, is continuously being developed in other projects and has a large acceptance and user base.

B Implementation environment

The various components in the calibration instrument will be implemented in the following languages:

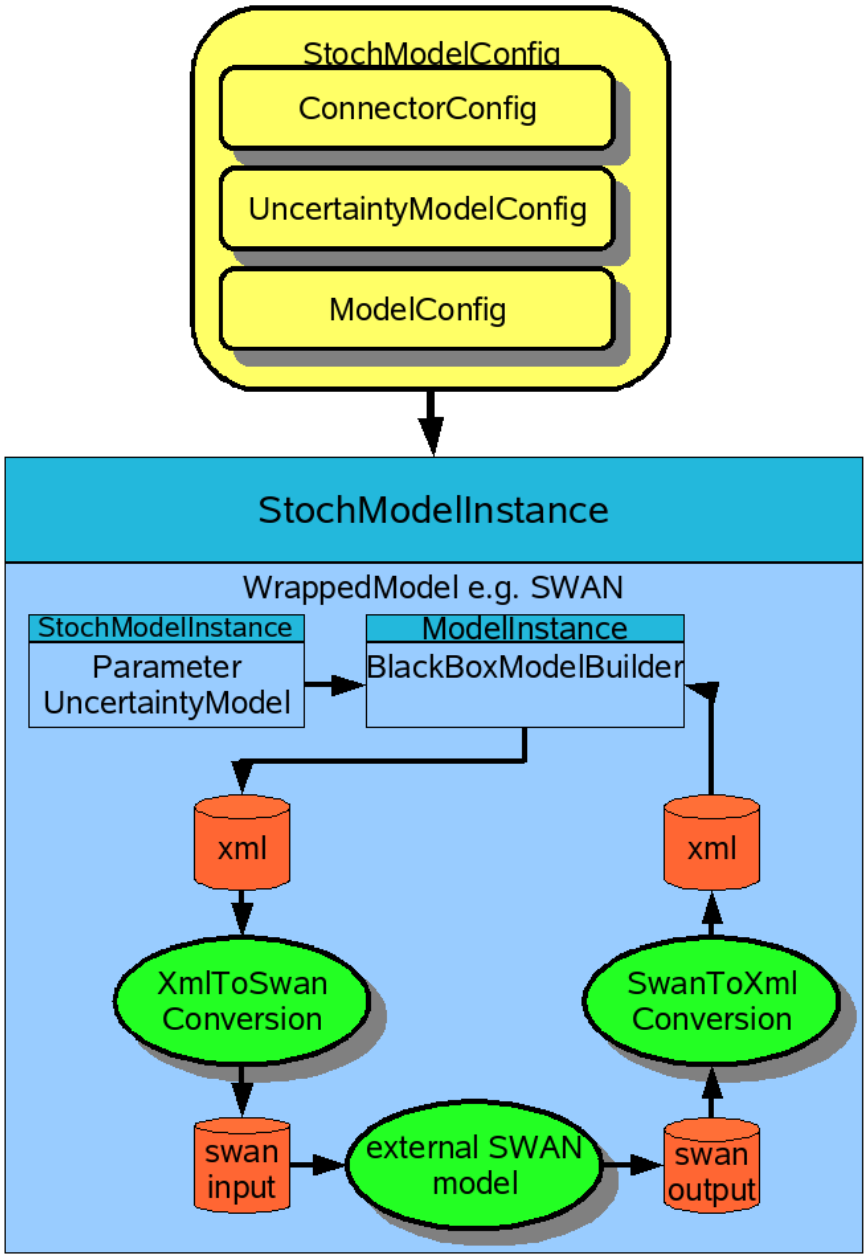
- *Main program*: DATools (see below);
- *Algorithm*: (DUD, others): java;
- *StochasticObserver*: depending on the selected observer: java (DATools observer), C (COSTA observer);
- *Swan ModelWrapper*: C#;
- *DATools / COSTA utilities*: see below;
- *Newly developed components for calibration instrument*: java;
- *Newly developed components for calibration instrument and validation instrument SHiVa:* java and/or matlab.

DATools
main program, algorithms, ulitities: java

COSTA
utilities: C will be wrapped in java.
algorithms: C/F77, will be transcribed to java.

C Stochastic model for SWAN

In general, a stochastic model instance exists of a deterministic model and stochastic model defining the uncertainties.
Such a combination is instantiated by reading the configuration file describing which uncertainty model to use, and which deterministic model.
For SWAN, the deterministic model will be the COSTA black box wrapper, configured for SWAN. This black box wrapper reads the parameters (stored in XML-files as TreeVector), inserts them in the SWAN input, executes the model, and delivers the required results.



D Functional Requirements for the C-I

The requirements and recommendations for the calibration tool can be categorised as follows (this text copied with minor changes from WL, 2006b):

Purpose of the calibration tool:
The purpose of the calibration tool is to determine in an efficient and effective way the optimum parameter setting of the SWAN model to compute spectral wave transformation in tidal areas with complex bottom topography, notably the Wadden Sea.
It should result in a single (optimum), physically realistic parameter setting for the Wadden Sea.
The calibration tool should have facilities to deal with constraints in terms of (physically) acceptable parameter ranges.
Due to the various sources of uncertainty the optimum parameter estimates are inherently uncertain. It is therefore recommended to quantify this in some way, e.g. by determining a band around the optimum, or, ideally, the probability density function for the parameters.
It is expected that with the calibrated model, the Hydraulic Boundary Conditions can be computed more reliably than with a non-calibrated model.

Description of the calibration tool:
The calibration tool is a flexible software environment, allowing efficient defining, running and evaluating series of model simulations with the aim of an efficient and effective assessment of the effects of variations in SWAN parameter settings. Key words are: (semi-)automated, quantified, objective and reproducible.
The tool allows the flexible definition of explicit error measures or Goodness-of-Fit criteria (plus non-acceptance criteria) in terms of relevant measurable quantities, taking into account associated uncertainties.
The tool should allow for formal application of penalties in the GoF and/or allow for checks on physics-based constraints.
The above quantitative criteria should be accompanied by well-chosen postprocessing to present visualizations and key statistical quantities.

Calibration – the GoF:
For each simulation the calibration tool will automatically produce one or more GoF's, and associated statistical and graphical presentations. Criteria for monitoring of the convergence are necessary and will be implemented, to enable intermediate interruption of the automated calibration cycle when user evaluation is deemed appropriate, so avoiding continuation with unrealistic settings.
The minimisation technique for the GoF criterion or criteria should be effective and numerically efficient.
A critical a priori assessment of the quality of the calibration and validation data sets used for comparison is necessary, in terms of relevance for the process at hand, measurement accuracy and their spatial/temporal distribution and representativity for the Wadden Sea (or other domain) as a whole.
The findings of this data quality assessment will be reflected in the appropriate weight factors in the GoF criterion.
Based on general practice and availability of reliable data sets, the GoF will in principle be based on a formulation in terms of the integral parameters significant wave height (H_{m0}),

peak period (T_p), spectral wave period ($T_{m-1,0}$) and mean direction $\overline{\theta}$ (with weight terms), but could also provide information on the spectral form (tail). Information on the high end of the spectrum, relevant for various fail mechanisms, is represented by the spectral wave period T_{m02}.
As an example of a GoF or least squares criterion, we refer to Alkyon (2003). This formula could be adapted for the present case to read:

$$GoF = \frac{\sum_{i=1}^{N_i}\left[w_{Hm0}^{i}\left(H_{m0,obs}^{i} - H_{m0,sim}^{i}\right)\right]^2}{\sum_{i=1}^{N_i} {w_{Hm0}^{i}}^2 \sigma^2_{H_{m0,obs}^{i}}} +$$

$$+\frac{\sum_{i=1}^{N_i}\left[w_{T_{m-1,0}}^{i}\left(T_{m-1,0,obs}^{i} - T_{m-1,0,sim}^{i}\right)\right]^2}{\sum_{i=1}^{N_i} {w_{T_{m-1,0}}^{i}}^2 \sigma^2_{T_{m-1,0,obs}^{i}}} + \frac{\sum_{i=1}^{N_i}\left[w_{T_p}^{i}\left(T_{p,obs}^{i} - T_{p,sim}^{i}\right)\right]^2}{\sum_{i=1}^{N_i} {w_{T_p}^{i}}^2 \sigma^2_{T_{p,obs}^{i}}} +$$

$$+\frac{\sum_{i=1}^{N_i}\left[w_{T_{m02}}^{i}\left(T_{m02,obs}^{i} - T_{m02,sim}^{i}\right)\right]^2}{\sum_{i=1}^{N_i} {w_{T_{m02}}^{i}}^2 \sigma^2_{T_{m02,obs}^{i}}} + \frac{\sum_{i=1}^{N_i}\left[w_{\overline{\theta}}^{i}\left(\overline{\theta}_{obs}^{i} - \overline{\theta}_{sim}^{i}\right)\right]^2}{\sum_{i=1}^{N_i} {w_{\overline{\theta}}^{i}}^2 \sigma^2_{\overline{\theta}_{obs}^{i}}},$$

in which N_i is the number of spatial calibration data points (possibly summed over a number of applications), w_{φ}^{i} is a weight function, and $\sigma^{i}_{\varphi_{obs}}$ the standard deviation of the observed parameter in the particular location, accounting for the uncertainty or quality of the respective observations. An appropriate directional measure would be the angle from the mean wave direction $\overline{\theta}$ (ensuring proper accounting for directional periodicity).

Calibration – aspect of constraints and statistical quantities:
Required spectral and other behaviour will be checked in the form of constraints by defining appropriate checks which can be made in a semi-automated way.
The following statistical measures should be calculated for each model simulation result: RMS error (systematic and non-systematic parts), bias, spread, scatter index, indices of agreement, correlation index between simulation results and observations, see (Van Vledder, 1993).
Similarly, the following graphical output should be provided for each model simulation result: time series plot, scatter diagram, error histogram, see (Van Vledder, 1993).

Use of the calibration tool:
The calibration tool does not assume the existence of a rigid hierarchy or ranking of processes, as assumed in rigorous successive and separate generic calibration of key processes and parameters, starting with numerical settings (iteration), deep water parameterisations, shallow water processes, and Wadden Sea application.

Calibrated parameter settings for the deep water processes should only be adjusted within the earlier established acceptable ranges when calibrating shallow water process parameterisations.
In line with Battjes and Stive (1985) and Ruessink et al. (2003), the parameter α_{BJ} in the depth-induced breaking formulation formulation is set to α_{BJ} = 1, to avoid effects of interdependence as shown by Dingemans (1997, pp. 397-398).
For the calibration material used for assessing the quadruplet and triad parameterisations benchmark simulations and flume experiment data will be used, respectively.

Issues not addressed by the calibration tool:
The choice of model domain for an application is treated as fixed and well-founded at the start of the calibration – it cannot be subject of the calibration process.
The same holds for the geographical model resolution; only in case of significant spatial redundancy in independent observations will calibration allow explicit conclusions about the adequacy of the resolution.
The optimum parameter setting obtained in the calibration process will not automatically be valid beyond the range of tested physics, i.e. for general extreme situations not represented in the calibration data sets used. The much more quantified quality assessment that will provided by systematic calibration as proposed with the above defined calibration instrument should, however, provide a much better insight into the question of model quality for extreme conditions.

E Overview of model parameters and input parameters

The following table presents an overview of the SWAN parameters that may be taken into account in the model calibration. The table also includes input parameters (representing boundary forcing or volume forcing in the model) that may be added in the calibration.

The subsequent columns contain the parameter number, its symbol, its acronym in the SWAN code, its default value in the SWAN code, and a range (if available). For a detailed description of the nature of the parameters, see (WL, 2006b).

Nr.	Symbol	Acronym	Default value	Range	Remarks
1	α_{in}				Cannot be calibrated via the command file
2	$C_{ds,wc}$	cds2	2.36E-5		
3	q	powst	2.0		
4	r	powk	0.0		
5	C_{nl4}	Cnl4	3.0E7		
6	λ_{nl4}	lambda	0.25		
7	$C_{ds,bot}$	cfjon	0.067		
8	α_{BJ}	alpha	1.0		To be kept fixed (WL, 2006b)
9	γ_{BJ}	gamma	0.73		
10	α_{EB}	trfac	0.1		
11	$f_{\max,EB}$	cutfr	5.5		
12	α_{DWS}		1.0		Cannot be calibrated via the command file, in case of measured spectrum
13	$\alpha_{windspeed}$		1.0		Cannot be calibrated via the command file, in case of non-uniform wind field
14	$a_{winddir}$		0.0		As above

In the above table, α_{DWS} stands for the uncertainty factor with which the spectrum (measured or assumed uniform) that is prescribed at the open boundary will be multiplied. Furthermore, $\alpha_{windspeed}$ stands for the uncertainty factor in the wind direction, and $a_{winddir}$ stands for the uncertainty in wind direction, expressed in radians or degrees.

Remark 1: It is noted that not all parameters need to be selected for a calibration cycle; in general, a sub-set will be selected, based on prior knowledge on behaviour and relevance. The input parameters (nos. 12, 13, 14) will most likely not be taken into account initially, as we generally assume that these are sufficiently accurately known. If, however, the analysis

of the simulation results shows internal inconsistencies which, based on physical reasoning can be related to the model inputs, parameters 12, 13 and or 14 can be included in a further cycle of the calibration.

Remark 2: Note that we do not include bathymetric uncertainties in the calibration. If there is uncertainty about the bathymetry, this often is formulated in terms of a migration of a channel or similar systematic effect. In such cases, one should consider simulations with an appropriately adjusted bathymetry as such.

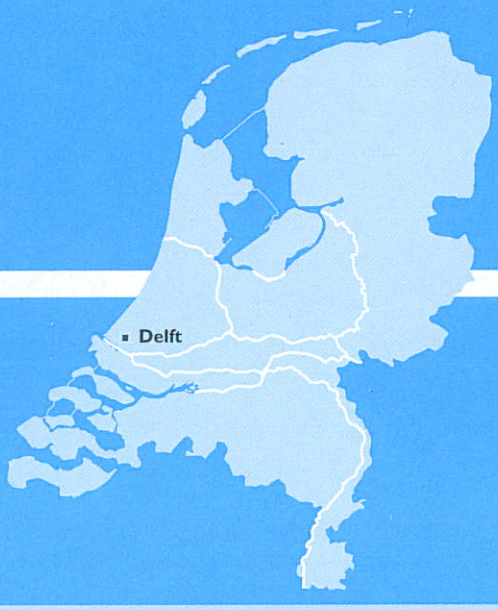
F Prototype test environment (java code)

```
package nl.wldelft.openda.algorithms;

import org.openda.interfaces.*;
import sun.reflect.generics.tree.Tree;


/**
 * OpenDA Dud algorithm
 */
public class DuD implements Algorithm {

    private StochModelInstance bestEstimate = null;

    public void run(Tree algorithmConfigTree,
                    StochModelFactory stochModelFactory,
                    StochObserver stochObserver) {

        // Get initial info
        StochModelInstance stochModelInstance =
stochModelFactory.getInstance();
        TreeVector initialParameters =
stochModelInstance.getParameters();
        int nParameters = initialParameters.getSize();

        // Evaluate cost for initial parameters
        double initialCosts =
this.computeCost(initialParameters,

stochModelFactory,

stochObserver);

        // Keep track of n+1 candidates for the parameters
and their costs
        TreeVector[] parameters = new TreeVector[nParameters
+ 1];
        double costs[] = new double[nParameters + 1];

        // First candidate is the initial set of parameters
        costs[0] = initialCosts;
        parameters[0] = initialParameters;

        // Next nParameters candidates are founfd by
perturbing the
        // initial parameters in each direction scaled with
the covariance
        Vector[] sqrtcovariance = stochModelInstance.

getParameterUncertainty().getSQRTCovariance();
```

```
        TreeVector newParameters =
parameters[0].duplicateTreeVector();
        for (int i = 0; i < nParameters; i++) {
            newParameters.axpy(1.0, sqrtcovariance[i]);
            costs[i + 1] = this.computeCost(newParameters,

stochModelFactory,

stochObserver);
            parameters[i + 1] = newParameters;
        }

        // Sort costs in descendig order
        sortCostsAndParameters(costs, parameters);

        // Main iteration loop
        int maxIterations = 3;              // todo: Read
from algorithmConfigTree
        double delta = 0.01;                // todo: Read
from algorithmConfigTree
        double[] alpha =                    // coefficients
for making linear combinations
                new double[nParameters + 1];//
of previous candidates

        for (int iter = 0;
             (costs[0] - costs[nParameters]) > delta && iter
< maxIterations;
             iter++) {

            // change parameters
            newParameters =
parameters[nParameters].duplicateTreeVector(); //
            newParameters.scale(0.0); //set to zero but keep
the structure
            for (int i = 0; i <= nParameters; i++) {
                alpha[i] = 0.0; // todo: This selection
depends on the algorithm.
                                //       This dummy algorithm
uses dummy values
                newParameters.axpy(alpha[i], parameters[i]);
            }
            // compute cost

            double newCosts = this.computeCost(newParameters,

stochModelFactory,

stochObserver);
            // replace worst candidate
            costs[0] = newCosts;
            parameters[0] = newParameters;

            // Sort costs in descendig order
```

```
                sortCostsAndParameters(costs, parameters);
            }

            // save best estimate so far, rerun this best
estimate to
            // be able to evaluate the results
            // todo: let computeCost not throw away the model
results
            //       to avoid this inefficient rerun.
            bestEstimate = stochModelFactory.getInstance();
            bestEstimate.setParameters(parameters[nParameters]);
            ObservationDescriptions observationDescriptions =
                    stochObserver.getObservationDescriptions();

bestEstimate.announceObservedValues(observationDescriptions);

bestEstimate.compute(stochModelInstance.getTimeHorizon().getE
ndDate());

        }

        public StochModelInstance getBestEstimate() {
            return bestEstimate;
        }

        private double computeCost(TreeVector parameters,
                                   StochModelFactory
stochModelFactory,
                                   StochObserver stochObserver) {
            // Initialize model
            StochModelInstance stochModelInstance =
stochModelFactory.getInstance();
            // Set parameters
            stochModelInstance.setParameters(parameters);
            // announce observations to model
            ObservationDescriptions observationDescriptions =

stochObserver.getObservationDescriptions();

stochModelInstance.announceObservedValues(observationDescript
ions);
            // run model

stochModelInstance.compute(stochModelInstance.getTimeHorizon(
).getEndDate());
            // get model values for observations
            Vector predictedVector =

stochModelInstance.getObservedValues(observationDescriptions)
;
            // compare to observations
            Vector observedVector = stochObserver.getValues();
            // residuals
            Vector residuals = observedVector.duplicateVector();
```

```
        residuals.axpy(-1.0, predictedVector);
        // scale with covariance
        //  (todo: assumes independent measurements - Check
if this is true)
        Vector standarddeviation =
stochModelInstance.getParameterUncertainty().getStd();
        residuals.divide(standarddeviation);
        // dot-product for computing cost
        return residuals.nrm2();
    }

    private void sortCostsAndParameters(double[] costs,
TreeVector[] parameters) {
        if (costs.length != parameters.length) {
            throw new IllegalArgumentException("#costs not
equal to #parameters");
        }
        // todo: invoke sort algorithm, sort costs and
parameters in descendig order
        //       for now: keep algorithm going until max.
#iters
        costs[0] = 1;
        costs[costs.length - 1] = 0;
    }
}
```



WL | Delft Hydraulics

Rotterdamseweg 185
postbus 177
2600 MH Delft
telefoon 015 285 85 85
telefax 015 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

Rotterdamseweg 185
p.o. box 177
2600 MH Delft
The Netherlands
telephone +31 15 285 85 85
telefax +31 15 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

