Software Requirement Specification (SRS)

Appendix C

Contents

Figures

Tables

1 INTRODUCTION

1.1 Purpose

This document describes the Consultants perception of the required functionality of the Nile Basin DSS (NB DSS) The specifications at this point in time define the full system but only at a high level.

Over the three development cycles the specifications will be refined. Each cycle will start with a detailed requirement analysis and software design stage, and during this stage the specifications will be detailed on a functionality area by functionality area basis. For example, specifications for time series handling shall be included in development cycle 1 while detailed specifications for multi-criteria analysis (MCA) will be included in development cycle 3.

The specifications established in the Terms of References (/2/) or developed through the Use Cases (/3/) are not in all aspects sufficient for a detailed design of the system. Information that must be established or clarified is marked in the document with a reference to Chapter 8 Issue List. Clarification of these issues will be made during the detailed analysis and design stages of the 3 development cycles.

1.2 Document Conventions

The document makes use of various types of diagrams in order to best convey the description of the system. Some of these diagrams are formatted according to the UML specifications; such diagrams are marked with a "UML" indication in the figure caption.

The document refers at several places to documents referenced in section 1.5. Whenever text is directly quoted from these documents, it will be shown in italics.

A document reference is made using a /n/ notation where the 'n' indicates the reference number. I.e. /2/ refers to the Contract (including the Terms of References). Section 1.5 lists all references.

1.3 Intended Audience and Reading Suggestions

The document is written with the intention to provide a functional description of the NB DSS and as such targets all people – both on the Client's and Consultant's side - having an interest in understanding the system's functionality.

It, however, specifically targets the Client's DSS Core team with the purpose of ensuring a common understanding of the system's functionality by the Client and the Consultant.

The Document is an Inception Phase deliverable and constitutes Appendix C of the Inception Report.

Chapter 2 provides an overview of the system. Although the functional components are not formally established before chapter 1, the overview lists and briefly describes them.

The main chapters in the document are chapter 1, which provides an analysis of the supplied use cases (/3/) and chapter 1, which describes the main system features. Chapter 1 uses the use cases together with the functional requirements to establish a number of functional components. A functional component is a logical grouping of related functionality. The functional components are mapped to the functional requirements provided it in /2/ (see section 3.2). This mapping identifies a number of issues – e.g. uncertainties with respect to understanding the requirements, request for further information. These issues which are summarised in Chapter 1 shall be resolved during the detailed analysis and design stages of the development cycles.

1.4 Project Scope

The project purpose is to establish the NB DSS conforming to the requirements established in the ToR /2/.

The Consultant envisions the NB DSS to be system that can serve as a shared knowledge base, provide analytical capacity, and support stakeholder interaction, for cooperative planning and management decision making for the Nile River Basin.

The ToR /2/ provides the following high-level vision of the NB DSS.

- *A Graphical User Interface (GUI) providing the end-user a simple to use yet powerful and flexible application*
- *A database serving as a central place to store data*
- *A Data Management component utilized to prepare input data for simulations and analyze simulation output*
- *A Dynamic Water Budget and Allocation Model and a Scenario Management component to easily study the effects of various developments in the Nile basin*
- *An Analysis Component for consolidating various types of information and data in multi-criteria analyses.*

1.5 References

/1/ Request for Proposals for Consultancy Service To Develop and Deploy The Nile Basin Decision Support System (DSS). / Nile Basin Initiative, 2008

/2/ The Contract entered between DHI and NBI including the Terms of References. / DHI and NBI. May 2009

/3/ The "4+1" Use cases as communicated to DHI on the 10th September 2009 (included in Appendix B of the Inception Report)

/4/ "Essential Technical Diagrams and key information for assessing design" as communicated to DHI during the Cairo workshop.

/5/ Software Architecture Document. /NBI and DHI, October 2009. (Appendix D of the Inception report)

1.6 Terminology

The Terminology used in this document is listed in Table 1.1.

Table 1.1 Terminology

Name	Description
Actor	In UML an actor specifies a role played by a user or any other system that interacts with the subject.
Adapter	In the context of NB DSS a piece of software that makes it possible for the DSS Front-end and Model Tools to integrate. The name relates to from the software engineering Adapter pattern. The Adapter pattern translates one interface of a component into a compatible one (see e.g http://en.wikipedia.org/wiki/Adapter_pattern). In this case part of the model tool interface is translated into an interface that the NB DSS can leverage
BLOB	Binary Large Object. In a database BLOB fields can be added, changed, and deleted. However, they cannot be searched and manipulated with standard database commands.
Catchment	A catchment is an area where water is collected by the natural landscape. In a catchment, all rain and run-off water eventually flows to a creek, river, lake or ocean, or into the groundwater system.
CBA	Cost-benefit analysis.
Component, functional	A logical grouping of related functionality.
Component, software	An identifiable part of a larger program. In software engineering a component is a reusable program building block that can be combined with other components in the same or other computers in a distributed network to form an application.
CRUD	Create, Read, Update and Delete – the canonical database interaction types.
DAO	Data Access Object, typically referring to a programming pattern for interaction with a database. See http://en.wikipedia.org/wiki/Data_access_object
DSS	Decision Support System.
DSS Front-end	The custom Windows application developed to the NB DSS. Serves as the front-end (entry point) of the NB DSS.
Ensemble	In this context an Ensemble is equivalent to a group of time series typically used as inputs for an ensemble simulation
Ensemble modeling	Refer to applying mathematical model with ensembles of time series as input
Ensemble scenario	Refer to defining scenarios that involves Ensemble Modeling
GUI	Graphical user interface, synonymous with UI.

Name	Description
Hydraulic Model, Hydrological Model, Water balance and allocation model	Mathematical models with specific scientific focus (see also Mathematical Model)
Hydro Objects	Hydro objects are entities related to modelling of water related processes, e.g. reservoirs and irrigation schemes.
IDE-style user interface	A UI style where multiple child windows reside under a single parent window. Child windows are dockable and collapsible and can be tabbed and resized.
Interface	A protocol or interface is what or how unrelated objects use to communicate with each other.
IS	Information System.
Layer	A logical part of an application providing a set of specific functionalities.
Leaf use case	The term leaf use case is used in Chapter 3 to identify the most specific use cases defined in the “4+1” use cases.
Linked Model	A model that involves two or more models linked through data transfers from one model to another model.
Management Scenario	Describes the present or possible future conditions resulting from alternative water resources management and development strategies or changed climatic conditions (see also Scenario)
Mathematical Model	A set of mathematical expressions and logical statements combined in order to simulate certain characteristics of the natural system. The “Model” in this document refers to a Mathematical Model. A Mathematical model may also consist of a number of linked models.
MCA	Multi-criteria analysis.
Member	A function or method in a software class or interface description.
Model	Synonymous with Mathematical model
Model Setup	A model setup is a definition of all data required for a simulation, including input data, configuration of the physical infrastructure, management strategies, such as reservoir operation rules, and all output specifications.
Model Tool	A generalised mathematical model such as MIKE11 and MIKE BASIN. May include proprietary as well as public domain systems.
Model Tool Engine	The engine is considered the full suite of executables and Dynamic Link Libraries (dll) controlling the simulation of system behaviour, including pre- and post-processing of Model Data and parameters and solving of the underlying mathematical equations.
Model Tool User Interface (UI)	The Model Tool UI gives access to the functionality of the Model Tool Engine.
Modelling System	A suite of Model Tools.
Modelling System UI	The Modelling System UI gives access to the functionality of the Modelling System.

Name	Description
Module	A module takes care of a specific well defined functionality within the DSS front-end. An example of a module is the Timeseries Manager
NB DSS, NB DSS system	The complete DSS system delivered to the Nile Basin Initiative.
Parameter	A parameter is a quantity characterizing a (physical or conceptual) property of a system. Within the context of hydrologic mathematical models, examples of a parameter are hydraulic conductivity, channel roughness/resistance, storage delay time, and time of concentration. A parameter may or may not be constant in time.
Pivoting pattern	This pattern describes a flexible way of modeling attribute-pair values in a database schema, see /5/
PostGIS	Extension to PostgreSQL for handling spatial data.
PostgreSQL	Object-oriented relational database management system.
PROJ4	Cartographic Projections library
Reference Scenario	Describes past conditions and is used for comparing impacts of present and future water resources development and management strategies.
Scenario	A Model Setup or a set of linked models designed to analyse a specific combination of water resources development strategies and water resources management strategies. A scenario is typically used for simulation of conditions for ground water, surface water, water quality, water allocation etc. under a given combination of water resources development strategies, water resources management strategies and climatic conditions. Two principally different scenarios exist, namely *reference scenarios* and *management scenarios*.
Schema pattern	The Schema pattern is data model pattern where an entity is modelled through a type identifier and an aggregated field. The aggregated field can be validated through a type specific schema stored elsewhere in the database. This pattern can be used for modelling entities that not necessarily are known at system construction time. It makes it possible to extend a live system with new types of entities. The pattern is not an officially known pattern; but a pattern established by the Consultant
Simulation	A time-varying description of certain behaviour of the natural system as computed by the mathematical model. A simulation will produce outputs referred to as Simulation Results.
SRTEXT	An OpenGIS Consortium standard for representing of a spatial reference system, based on WKT
Study	A logical grouping of data, typically stored within a dedicated data compartment.
System	Any structure, device, scheme or procedure, real or abstract, that interrelates in a given time reference, an input, cause, or stimulus, of matter, energy, or information and an output, effect, or response of information, energy or matter.

Name	Description
The natural system	The entirety of the socio-economic, environmental/ecological and water resources (hydrological) sub-systems and includes the hydrological cycle or parts of it as we currently conceive it.
ThinkGeo	In this document shorthand for the MapSuite Desktop product from the Company called ThinkGeo. The product is map control for rendering of spatial data.
Tier	Hardware where a specific layer is deployed.
UI	User Interface.
UI object	A software object (instance of a class) residing in the user interface part of the software code.
UML	Unified Modelling Language – A language for the specification, visualization, construction, and documentation of the components of software systems.
Use case	A use case is a description of how an actor will use a software code. It describes a task or a series of tasks that an actor will accomplish using the software, and includes the responses of the software to the actions. In this report use cases may be textual descriptions or diagrams (UML).
USLE/RUSLE	Universal Soil Loss Equation and Revised Universal Soil Loss Equation
UTOPIA/NADIR	Ideal and worst values of objectives.
Variable	A variable is a characteristic of the natural system that may be measured and that may assume different numerical values at different times. It can be a series of inputs and outputs from the model, but also a description of conditions in some component of the model (state of the system).
Water Resources Development Strategy	A set of water management options that include structural changes to the existing system such as the construction or modification of reservoirs and irrigation schemes.
Water Resources Management Strategy	A set of water management options that include operational changes to the system such as changes of operation rules for reservoirs.
WKB	Well-known binary, an OpenGIS Consortium standard for GIS geometry specification
WKT	Well-known text, an OpenGIS Consortium standard for GIS geometry specification

2 OVERALL DESCRIPTION

This chapter provides an overview of the functional components constituting the NB DSS with focus on the component's main functionality and how they relate to, and interact with, each other.

The chapter does not present great detail in the descriptions – but aims at providing an overview that will ease the understanding of the more elaborate description given in Chapter 1 through Chapter 7. The rationale behind the existence of the listed functional components is offered in chapter 1.

2.1 Product Perspective

The NB DSS is as specified in the ToR /2/ a new system that shall support NBI in fulfilling their business mission.

NBI has already one information system in operation, the Nile IS system. The NB DSS will augment Nile-IS by publishing information to the system

2.2 Product Features

Seen from a functional point of view, the NB DSS consists of the components shown in Figure 2.1 and briefly described in Table 2.1

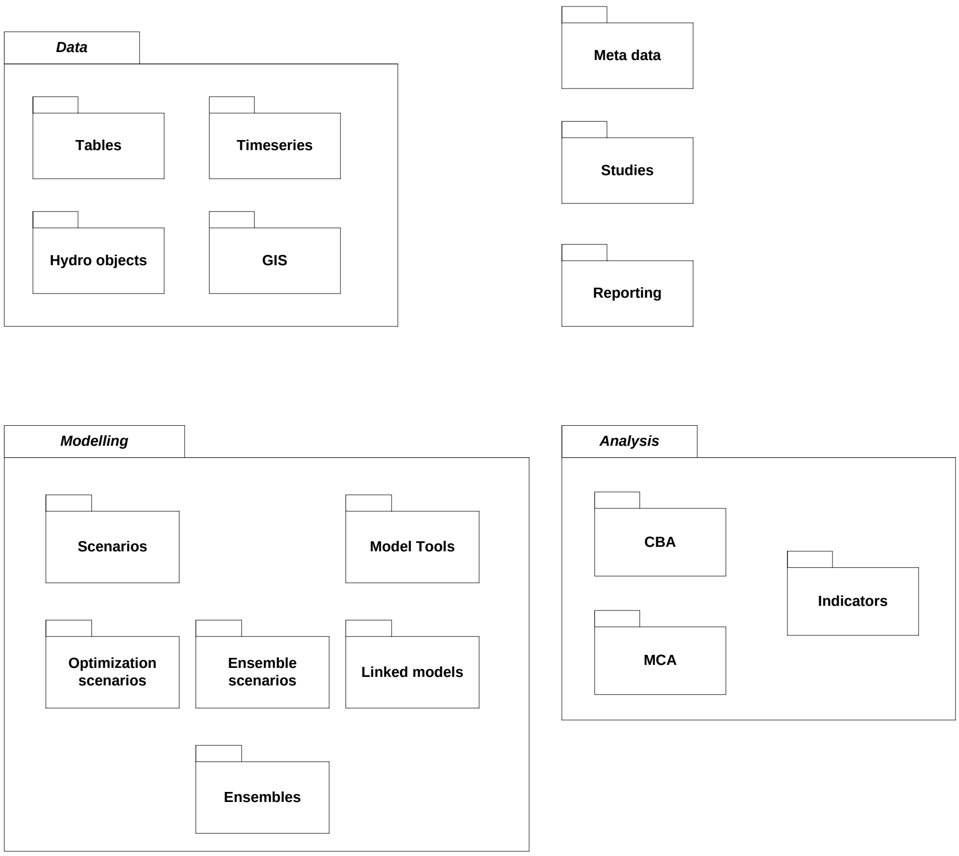

Figure 2.1 Functional components (UML)

Note the following in the figure:

- The figure uses a grouping mechanism where functional components are grouped in packages, i.e. the Data group consists of the Time series, Tables, GIS and Hydro objects functional components.

- All functional components persists their definitions and data in logical partitions within the DSS database called Studies.

- The Data functional component refers to the functionality provided for the time series, GIS, Hydro objects and Tables data types.

Table 2.1 Data related functional components

Functional component	Short description
GIS	Functionality for visualizing, persisting, querying and processing GIS data.
Time series	Functionality for visualizing, persisting, querying and processing time series data.
Tables	Functionality for visualizing, persisting, querying and processing tables. Tables are 2-dimensional matrices of data, used, e.g., for storing corresponding Q/H values for deriving rating curve. Tables are in this context not to be mistaken for database tables.
Hydro objects	Functionality for visualizing, persisting, querying and processing hydro objects. Hydro objects are objects related to the modelling of water related processes, e.g. reservoirs and irrigation schemes.

Table 2.2 Modelling related functional components

Functional component	Short description
Scenarios	Functionality for registering models, creating and executing and comparing scenarios (through variations in input data) on top of registered models.
Optimization scenarios	Functionality for managing scenarios that include simulation based optimization.
Ensemble scenarios	Functionality for managing scenarios that involve ensemble

	modelling.
Linked model	Functionality for managing configuration (linking) and execution of linked models A linked model involves execution of more than one model tool. Typically, the first model tool simulation generates output time series to be used as input time series for the next model tool simulation.
Ensembles	Functionality for managing and creating ensembles of time series, i.e. groups of time series.
Model tools	Represents the functionality provided by modelling tools like MIKE BASIN, MIKE 11 and MIKE SHE Studio.

Table 2.3 Analysis related functional components

Functional component	Short description
Indicators	Functionality for defining and creating indicators that are being used by the CBA and MCA functional components.
CBA	Functionality for creating and running cost benefit analyses.
MCA	Functionality for creating and running multi-criteria analyses.

Table 2.4 Other functional components

Functional component	Short description
Meta data	Functionality for creating, editing, querying and displaying meta data. Meta data are typically attributes describing data entities.
Studies	Functionality for creating studies and managing studies. A study is a logical partition of data in the database that shares the same security model and logically belongs together. Study management comprises activities related to associating users to a Study and defining access rights.
Reporting	Functionality for creating, publishing and storing reports over data stored within the DSS Database

In order to provide the users with an integrated and coherent experience, the functional components interact and provide their functionality to other components. It will e.g., be possible to filter data to just show output from one or more scenario simulations. Figure 2.2 below depicts the overall dependencies among the functional components.

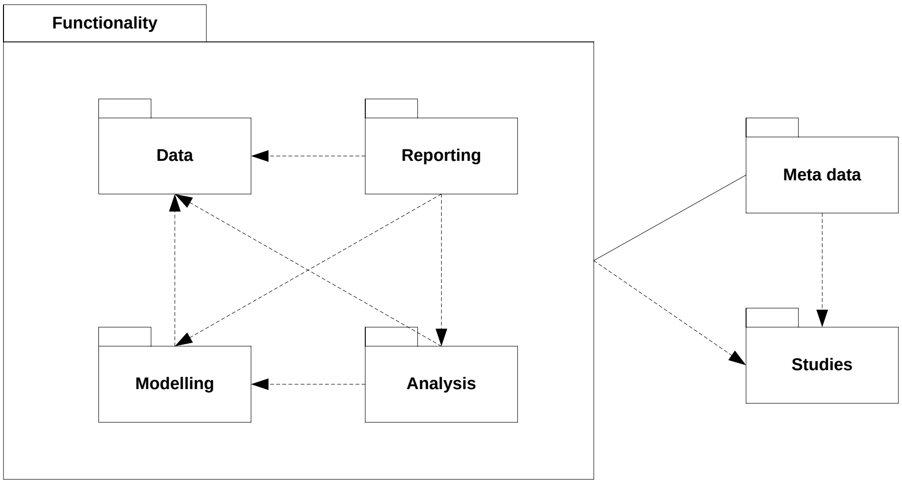

Figure 2.2 Overall dependencies among the functional components (UML)

The dashed lines in the figure represent dependency – or uses of – relations, e.g. Modelling will make use of Data[1].

The non-dashed line represents an attribute based relation. In the figure the solid line indicates that Meta data are a type of attributes associated with all entities.

Additional notes regarding the figure:

- Modelling uses Data – typically for resolving references to time series, i.e. reading the actual time series values.
- Analysis uses Data and Modelling – typically for obtaining references to output time series from scenario simulations and resolving these references.
- Reporting can generate reports based on data directly obtained from the Data functional component or indirectly through the Modelling and Analysis functional components.
- Meta data is related to all types of data, i.e. it can be associated to any data stored in the database.
- All functional components depend on studies in the sense that a Study is a grouping mechanism of data within the database, thus providing access to those data

Figure 2.3 below depicts the dependencies among the Modelling related functional components.

[1] Note Data here refers to the Data package as defined in Figure 2.1

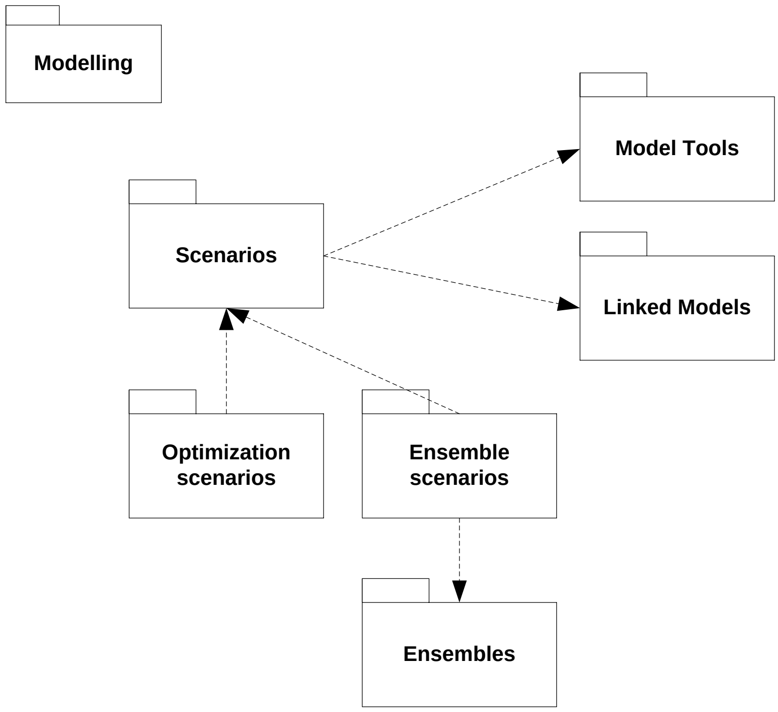

Figure 2.3 Modelling related functional components (UML)

Note from the figure that

- Optimization scenarios, Ensemble scenarios and Linked models are sub-groups of scenario-related functionality
- Ensembles provide the group of times series (ensembles) that the Ensemble scenarios functional component works upon
- Scenarios uses the Model Tools to perform the actual simulation.

Figure 2.4 below depicts the dependencies among the Analysis related functional components.

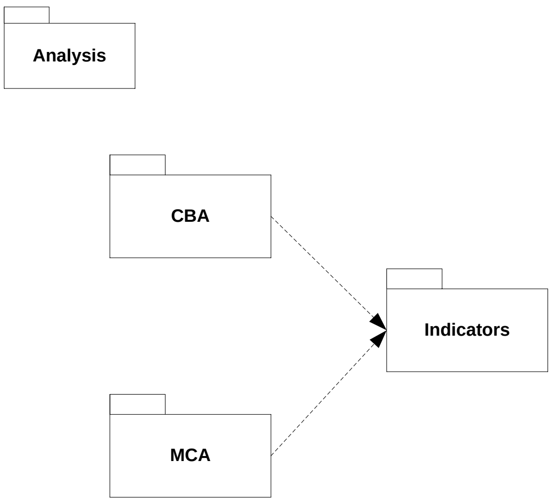

Figure 2.4 Analysis related functional components (UML)

Note from the figure that:

- The Indicators functional component provides functionality for defining and generating the indicators.

- Both the MCA and CBA functional component use the indicators established through the Indicators functional component.

2.3 User Classes and Characteristics

The use cases provided in /3/ and analysed in chapter 1 involve the actors listed below together with short descriptions (deduced from /3/).

1. Hydrologist – focusing on data preparation

2. GIS Specialist – focusing on GIS processing

3. Modeller – setup and execution of models and scenarios, generates reports

4. Decision maker –decision making based on reports

5. Water resource planner – analysing model simulation output

6. System administrator – exporting and importing study data, managing system users and studies

7. Water resources economist – defining optimization objectives and constraints

8. Software engineer – coding plug-ins to integrate model tools with the NB DSS.

Note:

- The Water resource planner and the Water resources economist are not the only actor in any of use cases in /3/. They are - whenever appearing collaborating with a Modeller.

- The Decision maker is involved in the use cases as reviewer and takes decisions based on generated reports.

- The Hydrologist and the Modeller are in many of the use cases combined actors.

- The GIS specialist and the Hydrologist are in many of the use cases combined actors.

This reflects the Consultants experience that the users of a DSS often span over more than one role. I.e. often a user is both hydrologist and modeller. For these reasons the users of the NB DSS are described more in terms of their data access rights than their working role.

Users of the system will access and work with data that logically exist at 2 different levels:

1. Study level – data that are pertinent to users associated with a specific study

2. Global level – data that are available for all users

When a user logs on to a study, all the data that is being created will be associated with the specified study. Studies will be discussed in more details in section 4.15. Users that log on to the NB DSS without selecting a study will only have access to data at the global level.

The users of NB DSS will belong to the following classes:

1. Reviewer – a user that can read data at the global level. The user cannot modify data; but is allowed to make simple reports and analyses. This user cannot access data at the study level

2. Study Reviewer – a Reviewer who can read data for studies to which he is associated.

3. Study Member – a Study Reviewer who can change, delete and create data for studies to which he is associated.

4. Study Lead – a Study Member who can associate users as Study Reviewer or Study Member with the studies to which he is appointed Study Lead.

5. Data Owner – a Reviewer who can add, change and delete data at the global level.

6. Administrator – can perform backup, restore, data synchronization, add and delete users, set user access levels (Reviewer, Data Owner and Administrator) and can create study areas.

There is no difference between the Corporate and Professional versions with respect to user classes and roles.

2.4 Operating Environment

The system consists of 3 deployment parts as depicted in Figure 2.5 below

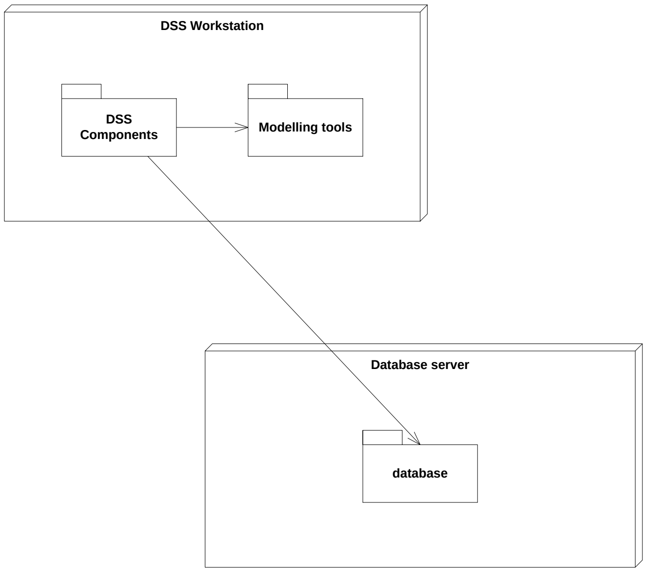

Figure 2.5 Deployment parts (UML)

Note from the figure that:

- The components shown within the nodes are not functional components but software components. These are discussed in more detail in the SAD /5/
- The DSS workstation is the interactive workplace for NB DSS users. The workstation must run the Microsoft Windows operating system. The exact version of the operating system will to be decided in consultation with the Client.
- The Database server hosts the NB DSS dataset. The server can run either Microsoft Windows or Linux. Exact versions of the operating systems will be decided in consultation with the Client.
- The associations between the 3 components represent visibility, i.e. the DSS components can access the database and the model tools.

The deployment shown in the above figure is not the only deployment possible. Two additional deployment options exist:

1. The Model Tools component could be installed on a dedicated simulation server. Operating system requirements for this server are the same as for the DSS Workstation.
2. All 3 deployment components could be installed on a single workstation. Operating system requirements for this workstation are the same as for the DSS Workstation.

These options are discussed in more details in /5/.

2.5 Design and Implementation Constraints

Design constraints comprise:

- The NB DSS will be designed as a 3-layer and 2-tier application. While it should be possible to convert the system into a 3-tier application, this will likely require a major rewrite and will have implications on how the Model Tools can be deployed.

- The Model Tools are envisioned to be external to the DSS Front-end. This implies that there are operating system constraints on the selection of Model Tools.

Implementation constraints comprise:

- PostGIS imposes constraints with respect to the types of GIS data that can be stored within the database. Currently PostGIS does not fully support storage and processing of raster data; however open source development project aiming at lifting this constraints exist. Solutions for incorporating raster storage and processing will be established during the detailed design phases.

- The system will be developed using Microsoft .NET 3.5. This limits the operating system to Windows XP or later versions.

2.6 User Documentation

The system will be delivered with:

1. On-line help, supplied in Microsoft Help 2 format (the latest help file format from Microsoft). This compact hyper text format is used by most Microsoft products.

2. User manuals, supplied in PDF format.

3. System documentation, supplied in Microsoft Help 2 format.

2.7 Assumptions and Dependencies

The system is anticipated to incorporate the external components listed in the tables below.

External components come in 2 groups:

1. Those that are external to the DSS Front-end. This comprises the modelling tools and the database

2. 3rd party products that are an integrated part of the DSS Front-end but for which the Consultants project group cannot access the source code or where there are restrictions associated with modifying the source code. This comprises miscellaneous programming tools such as the ThinkGeo GIS map control.

The MIKE systems are standardised, professional software systems that are developed and owned by DHI. As such DHI has the ability to modify the source code if required.

Model Tool components are listed in Table 2.5 to Table 2.7.

Table 2.5 Model tool components that are external to the front end

Model Tool	Description
MIKE 11	MIKE 11 is a fully hydrodynamic model
MIKE BASIN	MIKE BASIN is a water allocation and budget model
MIKE SHE Studio	MIKE SHE Studio is a semi-distributed rain-fall runoff model
DHI.Optimization	Library of optimization functionality

The model tools, integrated with the DSS Front-end[2] through a model adapter , provide the modelling capabilities of the NB DSS system. Section 4.10 provides more details on these components.

Interactions between the NB DSS and the model tools comprise model registration and scenario simulation. Figure 2.6 and Figure 2.7 below depict the data flow for these two interactions.

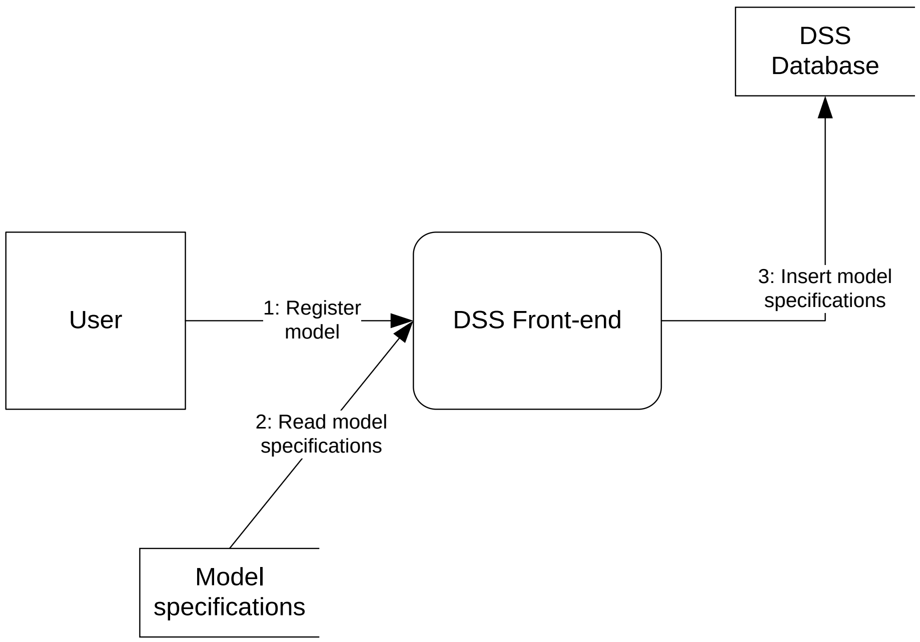

Figure 2.6 Data flow at model registration (Gane-Sarson)

Note from Figure 2.6:

- When the user registers the model with the NB DSS, the model specifications (the model setup) are inserted in the DSS database

[2] /3/ defines the NB DSS System as consisting of model tools, the DSS Database and the DSS Front-end

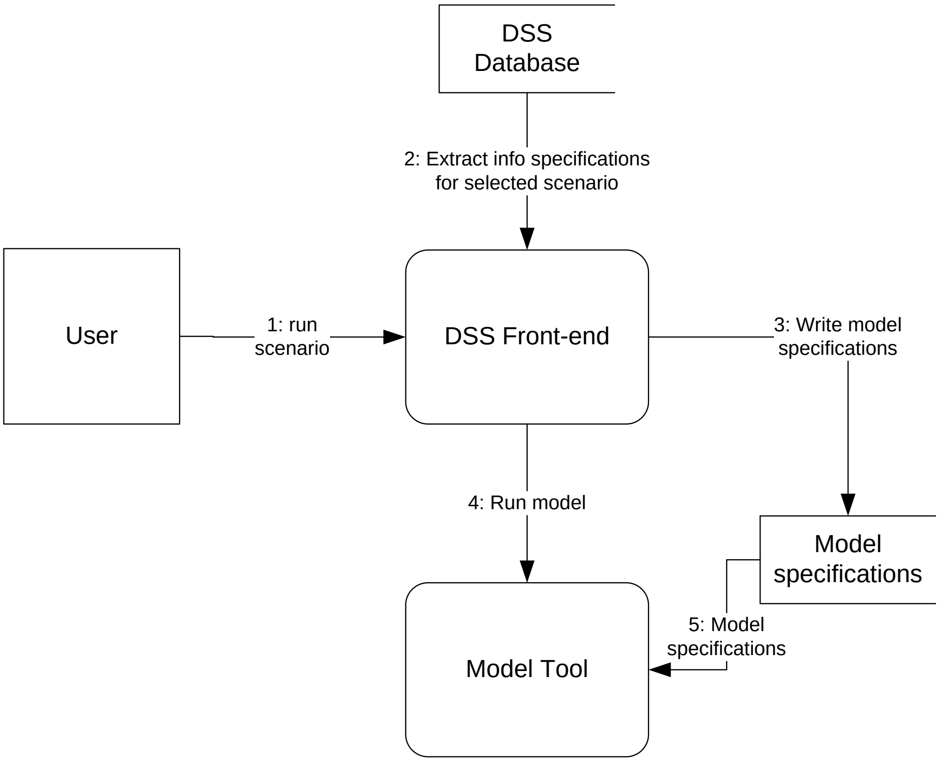

Figure 2.7 Data flow at scenario simulation (Gane-Sarson)

Note from Figure 2.7 that when the user initiates a scenario simulation, the DSS Front-end restores the original model setup – i.e. the setup that was present when registering the model – and substitute's specifications in the restored model setup with those defined for the scenario. All in all resulting in a model setup that differs from the one that original was registered. Then the simulation is started.

Interactions between the DSS Front-end and the model tools are described in more details in the SAD /5/.

Table 2.6 External database related components

Model Tool	Description
PostgreSQL	Relational database, see http://www.postgresql.org/
PostGIS	Enables the PostgreSQL server to be used as a backend spatial database for geographic information systems, see http://postgis.refractions.net/
NPGSQL	.Net Data Provider for PostgreSQL, see http://npgsql.projects.postgresql.org
NHibernate	NHibernate handles persisting plain .NET objects to and from an underlying relational database, see https://www.hibernate.org/343.html.

The external database related components are described in more detail in Section 5.2.1 and in the SAD /5/.

Table 2.7 External programming related components

Model Tool	Description
ThinkGeo MapSuite Desktop	MapControl providing functionalities for viewing and processing GIS functionality, see http://www.thinkgeo.com/
EUM	DHI library for managing engineering data types and units. Used by all MIKE products
MIKE Objects	DHI library for managing time series data (also known as TSObject)
DotNetMagic	Library of user interface controls, see http://www.dotnetmagic.com/

The external programming components are discussed in more detail in section 0. The use of ThinkGeo MapSuite Desktop and EUM are discussed in further in /5/.

3 USE CASE AND FUNCTIONAL COMPONENT ANALYSIS

This chapter presents the analysis of the “4+1” use cases and requirements (see /3/ and /2/). The purpose of the analysis is:

- *To establish generalised use cases by identifying patterns of common behaviour in the “4+1” use cases.* The generalised use cases will be further used during the detailed analysis and design stages of the development cycles in order to inform the design. Generalised use cases are use cases that draw from specific use to more general use; e.g. instead of describing how to visualise time series of water levels and time series of discharges there will be a description of how to visualise time series.

- *To establish functional components based on the generalised use cases.* The functional components are used as “input” to the SAD /5/ for establishing the overall software architecture. They together with the functional requirements form the basis for identifying the software components the system is being built upon.

This flow is depicted in Figure 3.1 below.

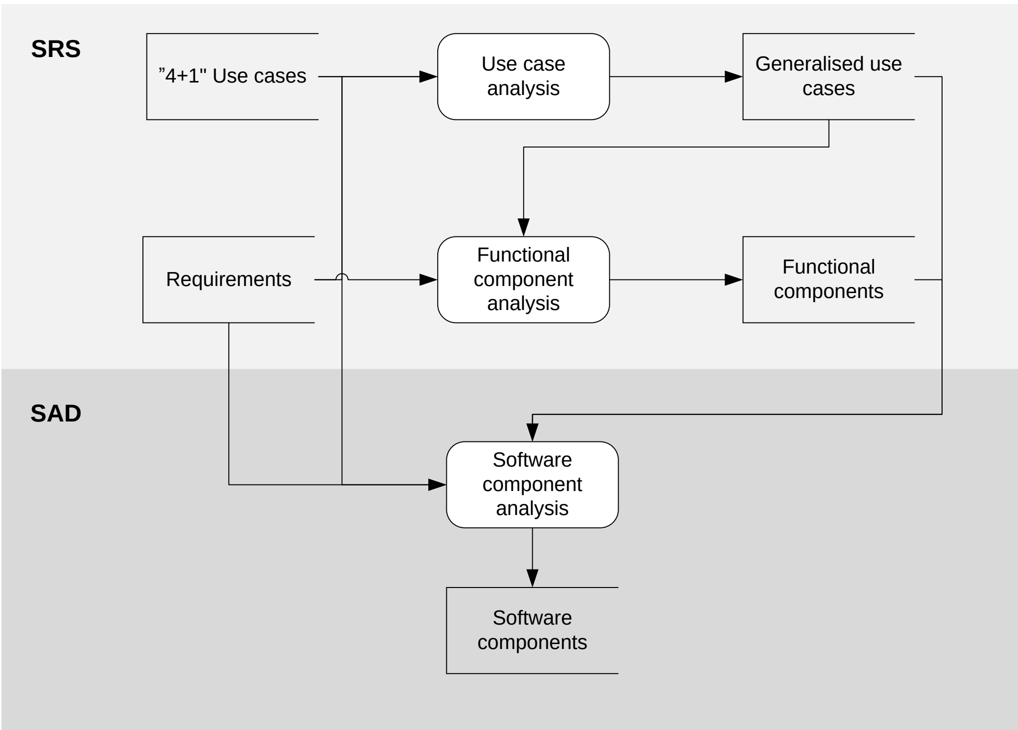

Figure 3.1 From Use cases and requirements to functional and software components

The analysis of the use cases was performed in 3 steps. First all activities in the supplied use cases were analysed and converted to generalised use cases. In a next step, the generalised use cases derived from the “4+1” use cases were consolidated and finally functional components were established.

The use case analysis is shown in tables and use case diagrams in sections 3.1.1 to 3.1.5 Section 3.1.6 uses the generalised use cases together with requirements to establish functional components.

3.1 Use Cases

This section presents the analysis of the “4+1” use cases. The analysis established generalised use cases from all of the leaf use cases[3] defined in the “4+1” use cases. The textual use case descriptions are attached in Appendix B of the Inception Report /3/. Note that:

- The use case diagrams shown in the following sections all shows the identified generalised use cases as use cases extending the “4+1” use cases. The “4+1” use cases are named according to the descriptions given in /3/.

- The descriptions and activities of the generalised use cases given in Table 3.1 through Table 3.5 are not exhaustive but rather indicative. They will be made further analysed during the detailed analysis and design stages of the development cycles.

The section concludes with a table mapping the generalised use cases to the requirements.

3.1.1 NB-DSS UC-01: Determine Causes for Declining of Lake Victoria Water Level

Background*: There is public concern regarding the falling levels of the Lake Victoria. The drop in lake levels has affected the socio-economic activities in the three east African countries of Uganda, Kenya and Tanzania that directly depend on the Lake’s water resources notably, through the prevailing power cuts, fall in fish supplies, unsafe docking of lake transport vessels, hanging domestic water supply intakes, etc. It has also caused environmental effects such as the drying of wetlands on the lakeshore line which are breeding grounds for fish.*

Key questions*: the following questions are expected to answered after implementing this use case:*

- *Does the net basin supply show significant downward trend?*
- *Is the departure from the release curve agreed by countries concerned (commonly known as ‘agreed curve’) the primary cause of decline in Lake water level?*
- *Can Lake Victoria sustain the releases as determined by the agreed curve?*

Figure 3.2 Brief description of the use case quoted from /3/

The use case UC-01 is described in /3/ in the form of a table, which is translated into the overview use case diagram shown in Figure 3.3.

Figure 3.4 to Figure 3.16 show how use case UC-01 was used to identify generalised use cases.

[3] Leaf use case here refer to the most specific use cases defined in the “4+1” use cases.

This analysis has identified the use cases listed in Table 3.2. Note that some of the generalised use cases have been identified and described in previous sections; they are included here for reference.

Table 3.1 Use case 01 - Identified generalised use cases

Generalised use cases	Description
Calibrate MIKE BASIN	This is a specific case of the “Calibrate model” use case when the model is related to the MIKE BASIN modelling tool. See “Calibrate model” in Table 3.2
Create scenario	Creating a new scenario (possibly from another scenario) within the DSS front-end. Activities involved comprise selecting existing scenario, specifying input time series – including e.g. ensembles - to be applied, initial conditions and output time series.
Filter time series	Defining search criteria for looking-up time series in the database. Activities involved comprise specifying search parameters like name, type of time series, time series generated from specified scenarios, associated with selected features etc.
Import Time series	Importing time series data to the database. Activities involved comprise formatting the input time series file to a supported layout, specifying the data type and unit for the input data, providing the time series with a name and associating it with GIS features.
Inspect GIS layers	Looking at GIS features, their attributes and meta-data. Activities involved comprise looking-up GIS feature layers, selecting features, visualising features, showing features as attribute tables and showing feature associated meta-data.
Inspect Time series	Looking at time series attributes and meta-data Activities involved comprise looking-up time series, selecting time series, showing time series attributes and showing associated meta-data.

Generalised use cases	Description
Publish report	Creating reports from different data sources – references to existing or “to be generated” data - within the database. Activities involved comprise defining report templates, by grouping controls on reports, associating controls with data sources and “running” the report template in order to create the report.
Register model	Importing a model setup to the database. Activities involved comprise selecting the model setup by communication with the relevant model tool, identifying input time series and identifying hydro objects.
Run scenario	Making a simulation using pre-specified input data and model components. Activities involved comprise selecting a scenario and executing it.
Setup MIKE BASIN	This is a specific case of the “Setup model” use case when the model is related to the MIKE BASIN modelling tool. Activities involved comprise defining network, defining user nodes with the network, specifying boundary conditions, setting other model parameters, specifying simulation period.
Use GIS catchment delineation tool	Establishing catchments based on GIS data Activities involved include selecting GIS feature layer, selecting the delineation tool, setting delineation parameters, performing the geo-processing and storing the delineated catchments.
Use rainfall runoff model	Calculating catchment runoff based on meteorological input time series, catchment properties and initial hydrological conditions. Activities involved comprise selection of input data, setting model parameters, calibrating the model, executing the model, model and storing the output time series.

Generalised use cases	Description
Use table tool	Process data stored in a table Activities involved comprise selecting the table, selecting table tool(s), specifying tools parameters, running the tool(s) and storing the output.
Use Time series tool	Processing time series. Activities involved comprise selecting of one or more time series to process, selecting the appropriate tool(s), setting tool specific parameters, executing the tool to modify the time series' data and storing the output in the database.
Visualise GIS layers	Show GIS features as map layers Activities involved comprise selection of GIS layers, adding the layer to a new or existing map and customising the map (e.g. symbology, labels, scale, projection)
Visualise tables	Display tables Activities involved comprise selection of tables and opening of tables to view tabular data.
Visualise Time series	Display time series Activities involved comprise selection of time series, adding time series to an existing or new chart or table and customising the chart or table.

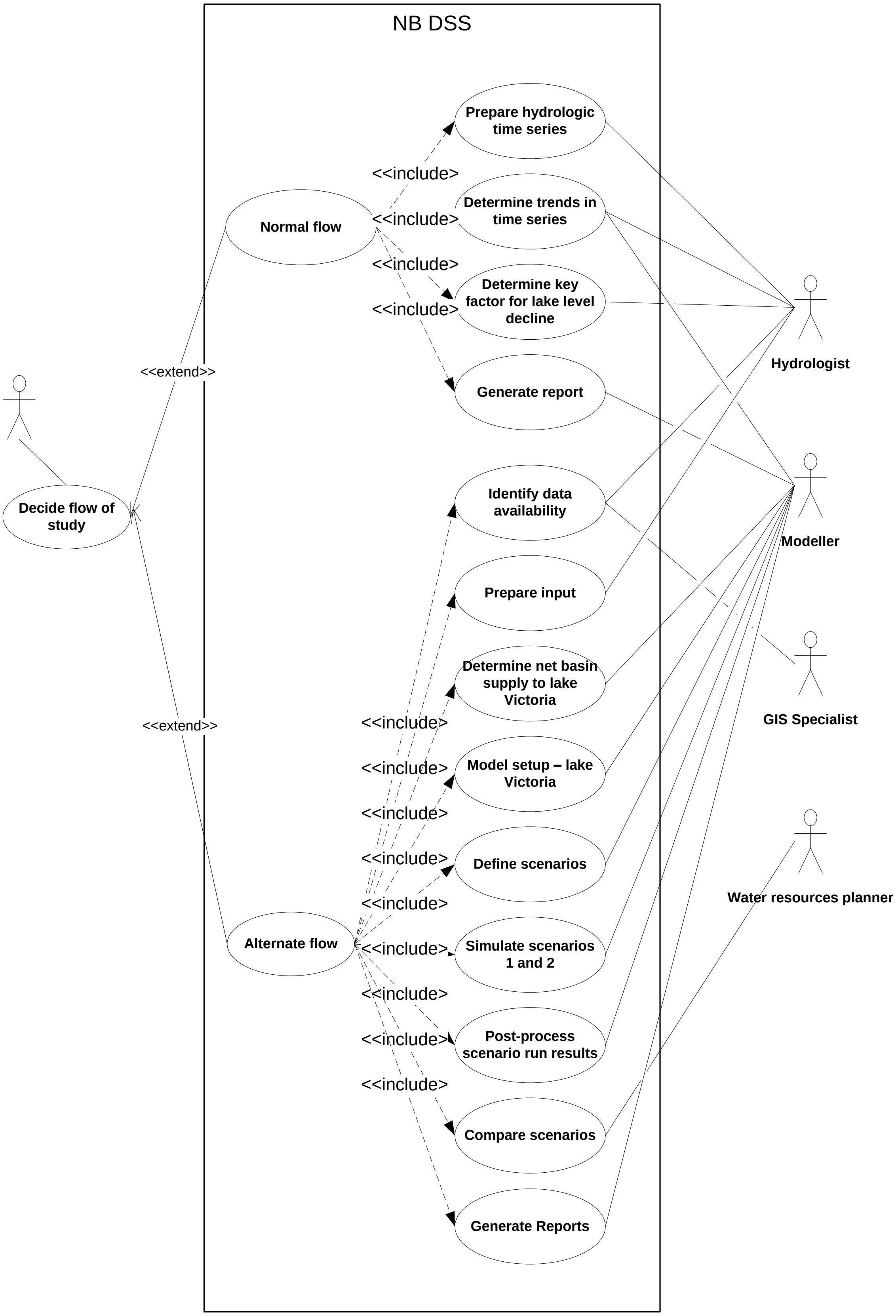

Figure 3.3 Use case 01 – Overview (UML)

Note in Figure 3.3 that:

- The course of this study depends on the available data – the normal flow data (where available) is sufficient to reach the required conclusions or an alternate flow where modelling has to be included. The decision making process is not included in the diagram.

- The modeller is involved in all the use cases in the diagram. This reflects experience by the Consultant that modelling and data processing is often done by the same person.

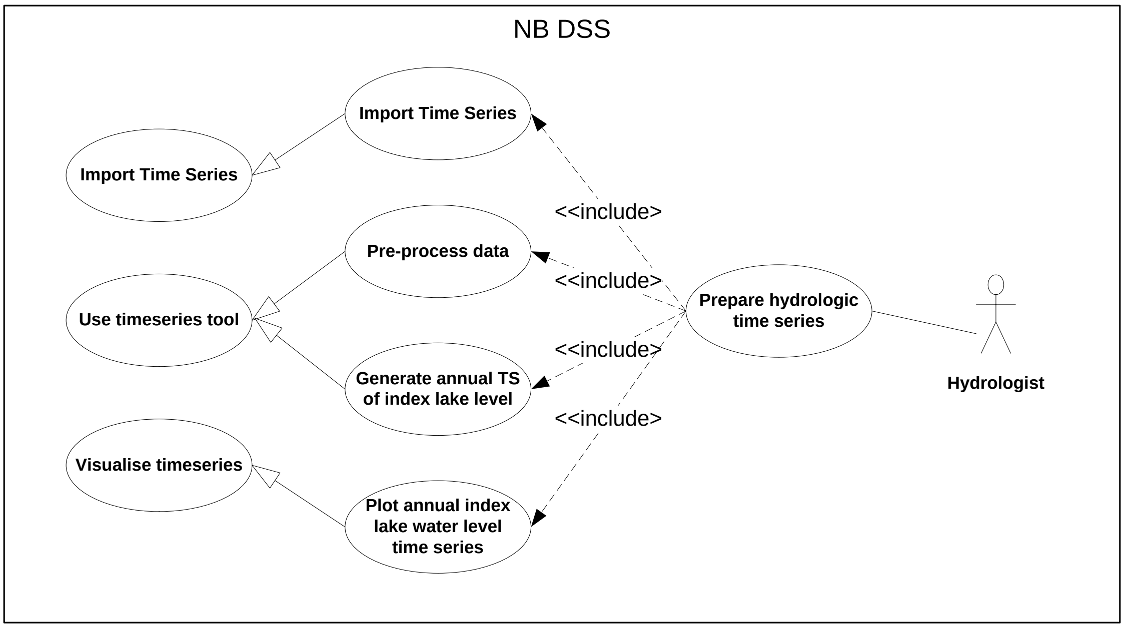

Figure 3.4 Use Case 01 – Prepare hydrologic time series (UML)

Note in Figure 3.4 that:

- “Prepare hydrologic time series” uses 3 time series related generalised use cases
  - “Import time series“ to load data into the database.
  - “Use time series tool” which includes the use of any of the standard pre- and post-processing time series tools like gap filling, re-sampling etc.
  - “Visualise time series” to visualise time series data either as graph or as table.

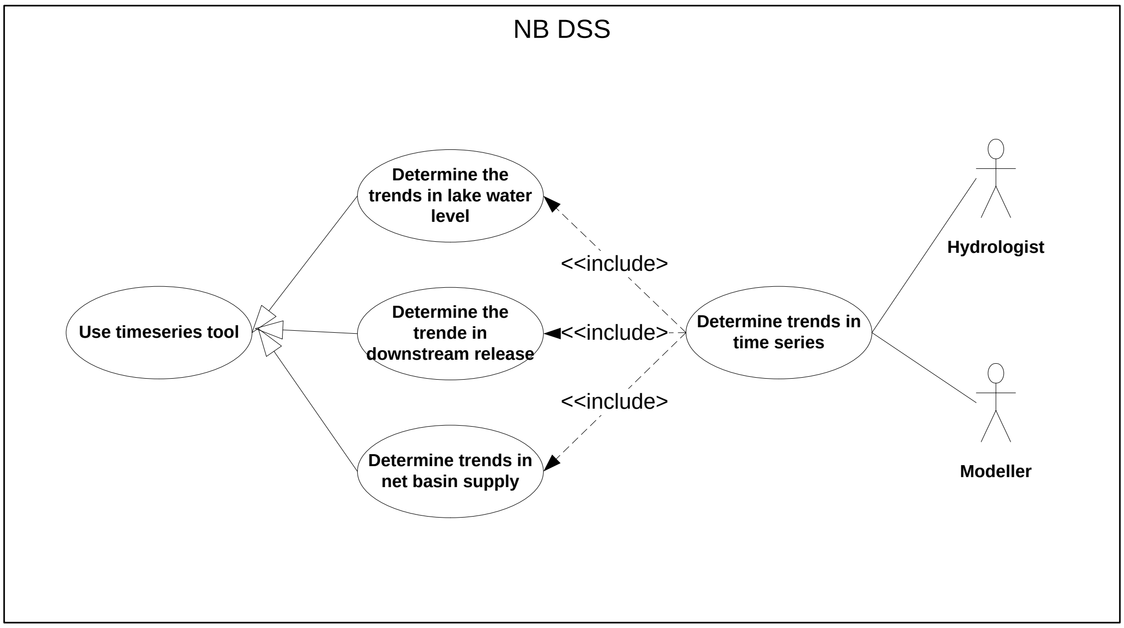

Figure 3.5 Use case 01 – Determine trends in time series (UML)

Note in Figure 3.5 that all trend analysis is performed by standard time series analysis tools.

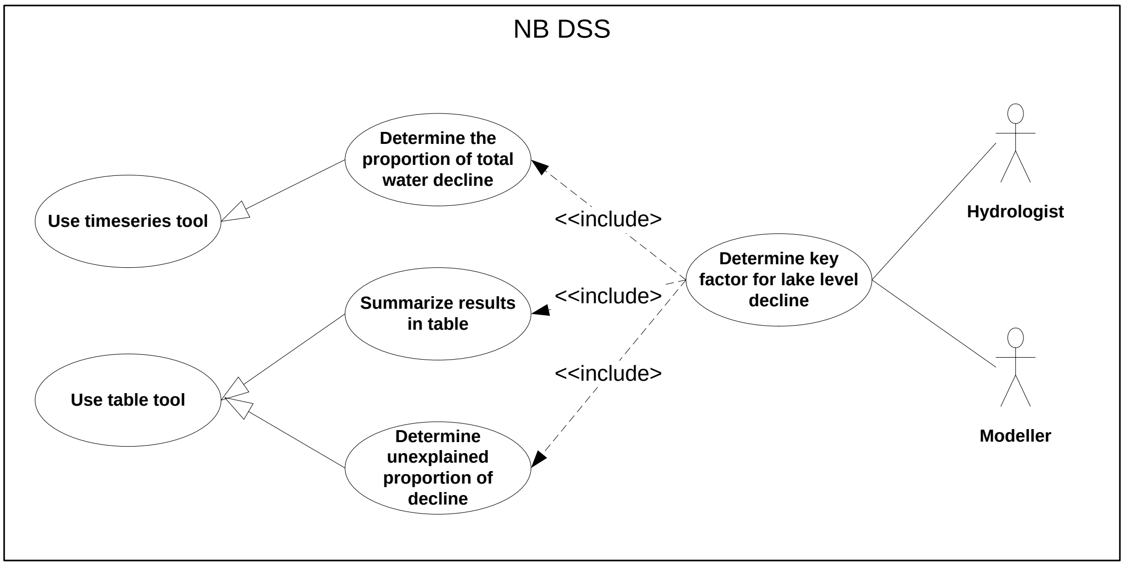

Figure 3.6 Use case 01 – Determine key factor for lake level decline (UML)

Note in Figure 3.6 that:

- Summarizing the results is performed by the table tool. This involves creating a table and entering the summarized values.
- Determine the unexplained portion of the lake level decline by inspecting the summary established in the table tool

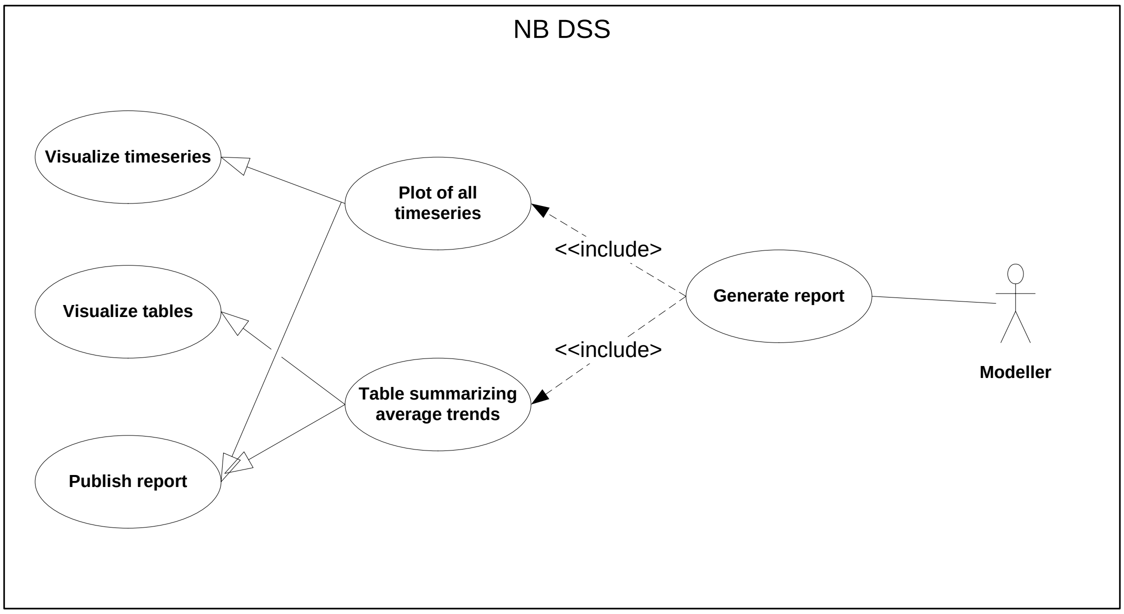

Figure 3.7   Use case 01 – Generate report (UML)

Note in Figure 3.7 that reporting is made by *using* the "Visualize time series" and "Visualize tables" use cases together with the "Publish report" use case.

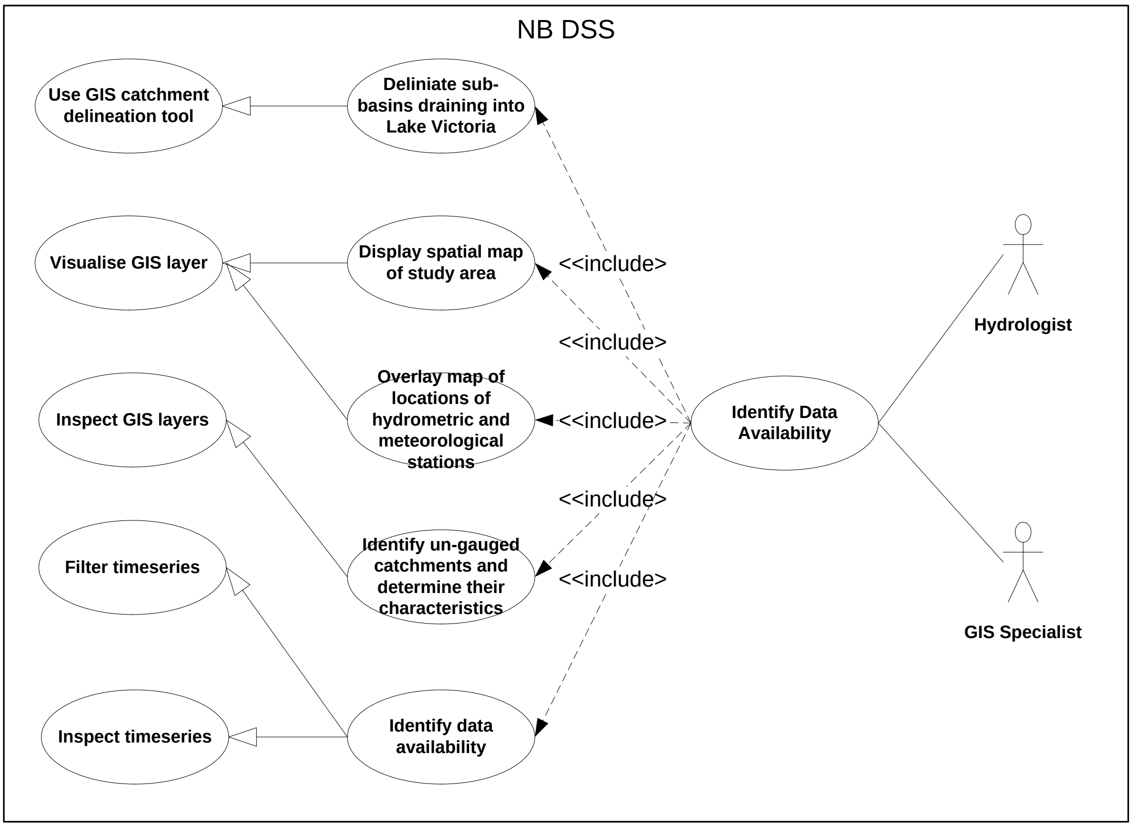

Figure 3.8   Use case 01 – Identify data availability (UML)

Note in Figure 3.8 that:

- "Inspect GIS layers" is used to identify un-gauged catchments characteristics. This involves inspecting feature attributes and their meta-data properties.

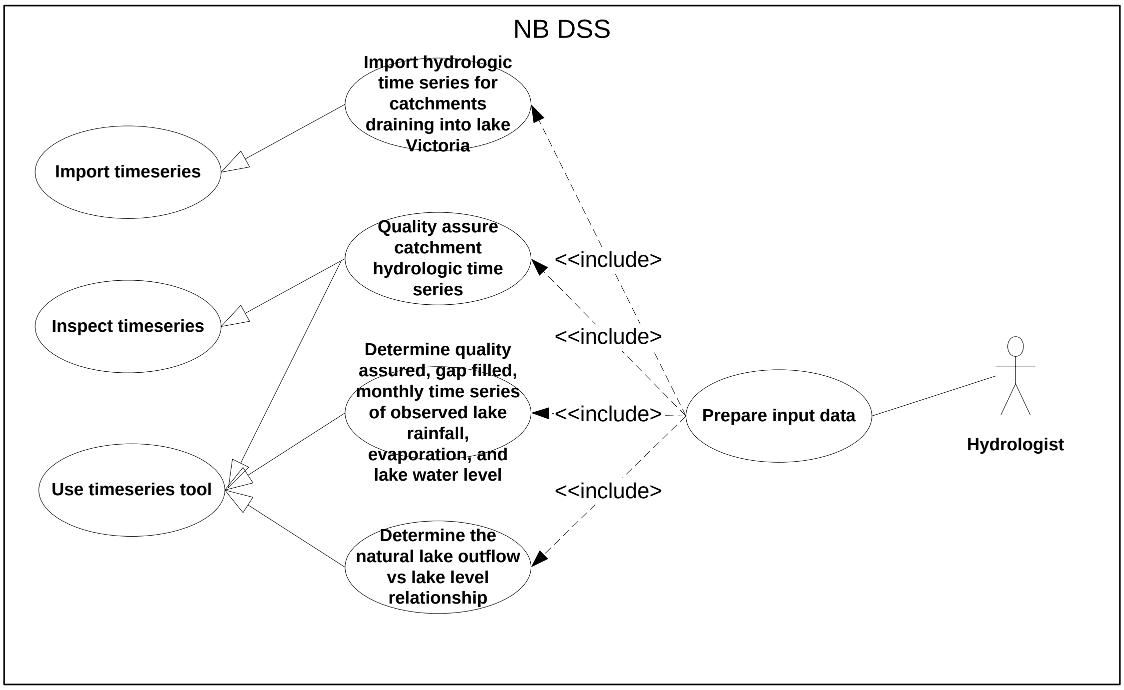

Figure 3.9 Use case 01 – Prepare input data (UML)

Note in Figure 3.9 that the "Prepare input data" use case is only concerned with time series data, importing, looking at attributes and meta-data ("Inspect time series") and using various time series tools.

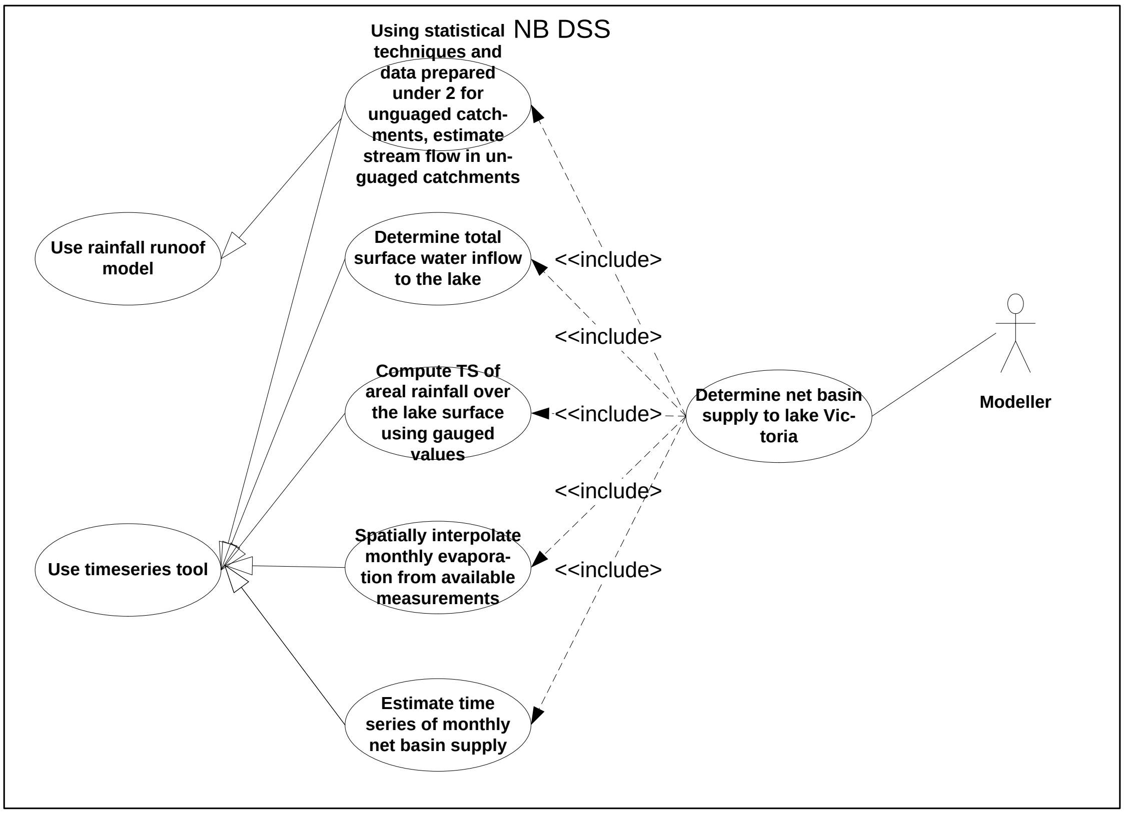

Figure 3.10 Use case 01 – Determine net basin supply to lake Victoria (UML)

Note in Figure 3.10 that:

- A rainfall-runoff model is assumed to be used for estimating the stream flow in un-gauged catchments.
- Various time series tools are used for the remaining parts, e.g. for creating rain-fall maps.

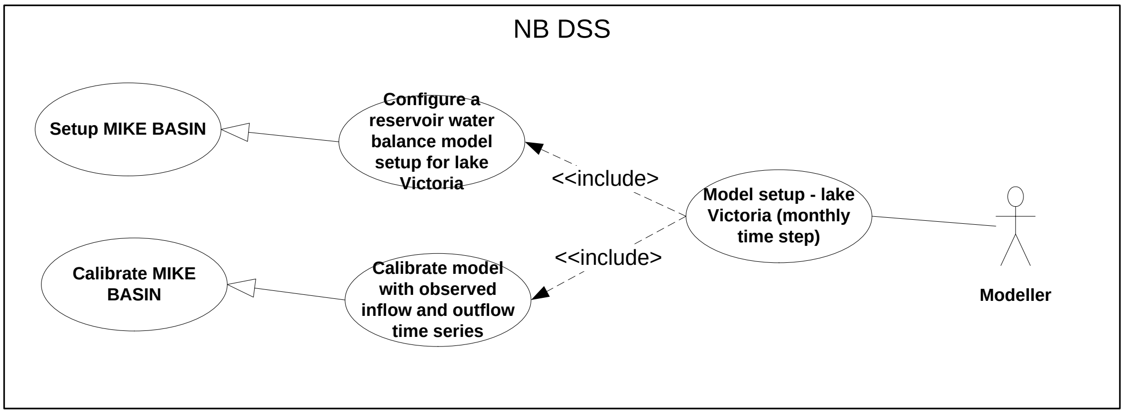

Figure 3.11 Use case 01 – Model setup – lake Victoria (monthly time step) (UML)

Note in Figure 3.11 that MIKE BASIN is used as model tool for reservoir water balance modelling

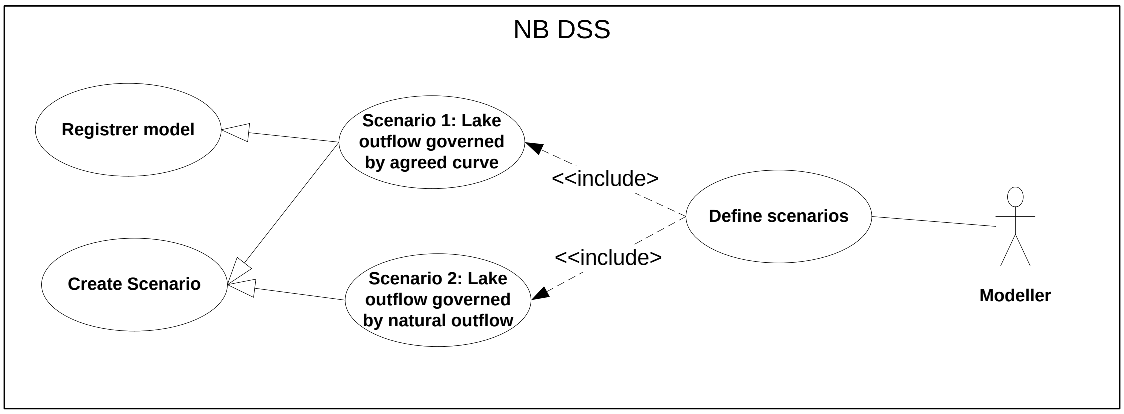

Figure 3.12 Use case 01 – Define scenarios (UML)

Note in Figure 3.12 that defining scenarios require that the model has been registered with the DSS. In this case, 2 scenarios are being created; however, the model must be registered just once.

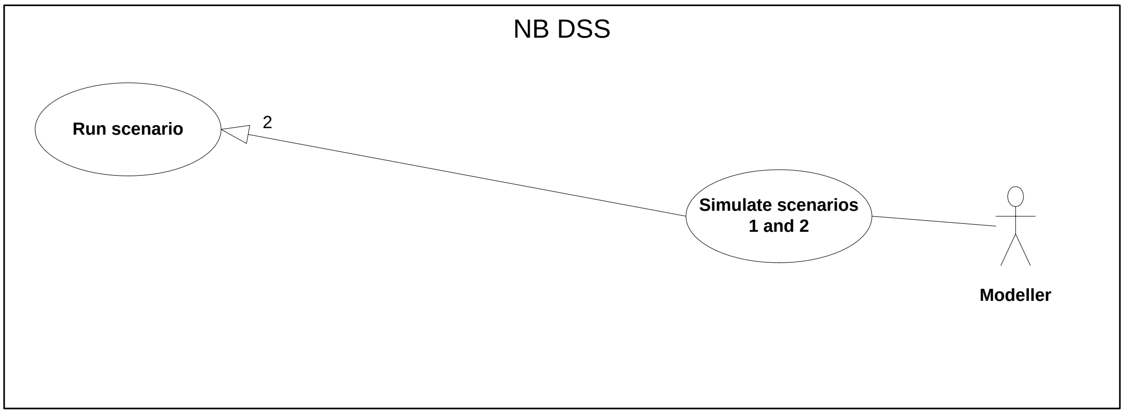

Figure 3.13 Use case 01 – Simulate scenarios 1 and 2 (UML)

Note in Figure 3.13 that "Run scenario" is performed twice.

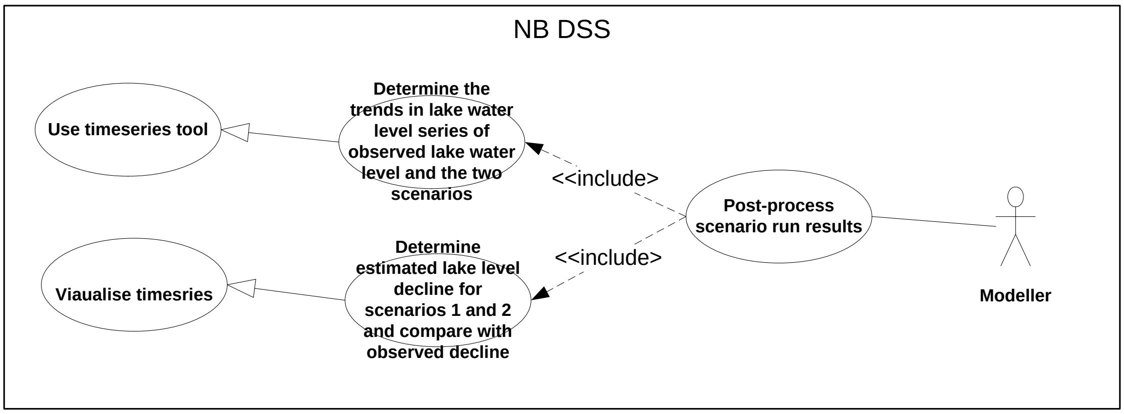

Figure 3.14 Use case 01 – Post-process scenario run results (UML)

Note in Figure 3.14 that post-processing of the scenario results is made by applying time series tools and visualising the data. This corresponds to the activities made when working with the data during the “Normal” branch of the whole use case, see Figure 3.5.

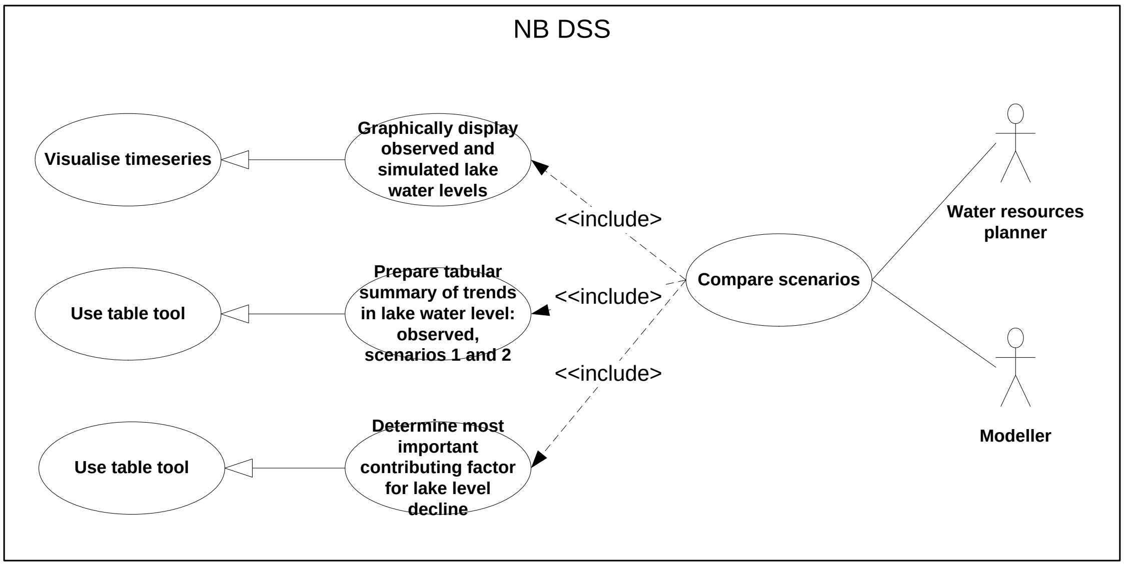

Figure 3.15 Use case 01 – Compare scenarios (UML)

Note in Figure 3.15 that scenario comparisons are made through the use of time series visualisation and applying the table tools. This corresponds to the activities made when working with the data during the “Normal” branch of the whole use case, see Figure 3.6.

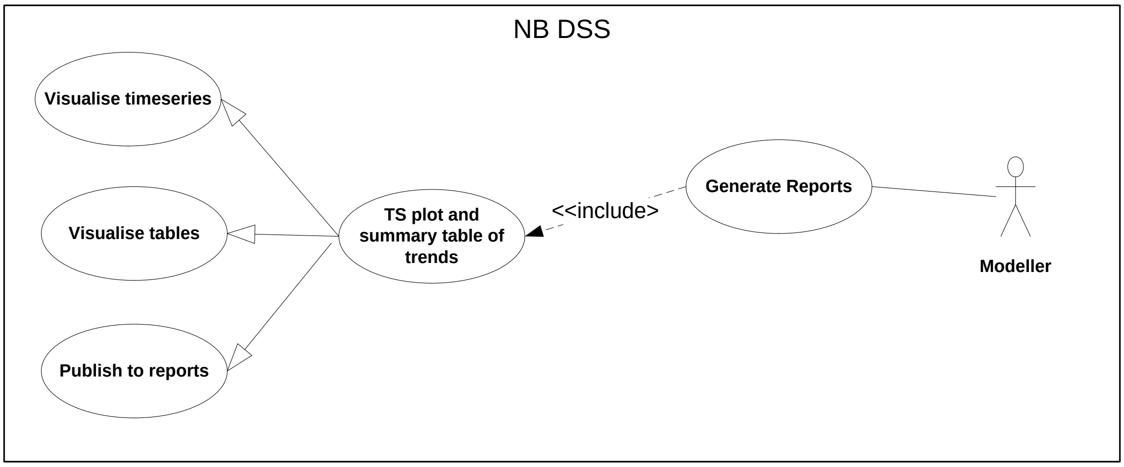

Figure 3.16 Use case 01 – Generate reports (UML)

Note in Figure 3.16 that report generation is made through publishing visualised time series and tables to reports. This corresponds to the activities made when working with the data during the “Normal” branch of the whole use case, see Figure 3.7.

3.1.2 NB-DSS UC-02: Select best option for Jonglie Canal

***Background**: The primary objective of the Jonglie canal Project was to conserve around 4.7 billion meter cube of water that is lost in the Sudd swamp mainly through evaporation and make it available for downstream use. The project was terminated in 1983 after completion of 260 km out of the total 360km. The EN-COM has realized the benefits of gaining additional water for various uses downstream and agreed to investigate the consequences, both positive and negative, that might occur due to the completion of the canal.*

***Key questions**: the following questions are expected to be answered through the implementation of this use case:*

- *How much water can be conserved if the Jonglei canal project is completed?*
- *What would be the total changes in the inundation patterns of the Sudd wetland (spatial extent, depth, etc)?*
- *What are the estimated impacts in community livelihoods?*
- *What would be the approximate volume of water to be conserved to keep environmental and socio-economic impacts to acceptable limits?*

Figure 3.17 Brief description of the use case quoted from /3/

The use case UC-02 is described in /3/ in the form of a table, which is translated into the overview use case diagram shown in Figure 3.18.

This analysis has identified the use cases listed in Table 3.2. Note that some of the generalised use cases have been identified and described in previous sections; they are included here for reference.

Table 3.2 Use case 02 - Identified generalised use cases

Generalised use cases	Description
Calibrate model	Model calibration includes adjusting model parameters such that the model provides, for selected input data, satisfactory outputs, typically by simulating historic events. Comparison/plotting of result time series with/against historically measured time series are used to assess result performance. This is typically done in within the GUI of the Model Tool, but may use external (to the Model tool) comparison tools as well.
Clone and modify MIKE BASIN model and scenario	Establishing a new model setup with defined scenarios (input definitions) in the NB DSS on the basis of an existing MIKE BASIN model setup. In the NB DSS this involves both cloning and editing of database content as well as use of the Model tool to alter an existing model setup configuration.
Create linked model	The process of linking NB DSS models where one model uses output from another model as input.
Create study	The process of establishing definitions in the NB DSS database of a Study entity, including assigning a Study owner account to manage the Study

Generalised use cases	Description
Define indicator	The process of determining the required input and processing needed to calculate an indicator value from data in the NB DSS. This task is performed outside of the NB DSS and which indicators are chosen depends on the objective(s) of the planned CBA or MCA.
Edit GIS data	Perform GIS operations on existing GIS data (extract, combine, modify properties, etc.) on exiting GIS layers to create new or modified layer information in the NB DSS database.
Import GIS data	Transfer from other sources (e.g. shape files) GIS information into the NB DSS database. The import may include projection of data, transformation of data, sub-selection of data and aggregation of data.. The outcome can be new layers/feature classes and/or new features in existing layers.
Import time series	Identified and described in Table 3.1
Manage study	The process of maintaining lists of access rights for user accounts attached to a Study. When a user works within a Study all the data created as part of the work is automatically associated with the study. This implies that the user implicitly manages the Study with respect to data.
Register MIKE 11 model	The process for bringing the MIKE 11 model definitions into the NB DSS, i.e. allowing the NB DSS system to interact with the model setup and define scenarios for execution of the model setup. This process is a special version of the “Register model” use case described in Table 3.1.
Register MIKE BASIN model	The process for bringing of the MIKE BAIN model definitions into the NB DSS, i.e. allowing the NB DSS system to interact with the model setup and define scenarios for execution of the model setup. This process is a special version of the “Register model” use case described in Table 3.1.

Generalised use cases	Description
Run MCA	The process of populating the MCA decision matrix or reference point graph with actual values for criteria values defined during the MCA setup. This task comprises setting of criterion weights for one or more stakeholder(s) (or stakeholder groups), selecting scenarios to be included, initiating the computation of the involved indicator values, and finally initiating the computation of all criteria from the indicator values. Run MCA is typically an interactive use of the MCA tool why the differences between Run MCA and Setup MCA is not yet finally defined.
Setup MCA	Defining the components to be used during the MCA Activities involved comprise defining criteria to be used in the analysis and linking said criteria to named indicators (see the use case “Use indicator tool”) through relationship tables (to be defined using the Table tool). Additional tasks include the definition of stakeholder(s) or stakeholder groups associated with criteria weights (which will be set and varied during the actual MCA).
Setup MIKE 11 model	This is a specific case of “Setup model” when the modelling is related to the MIKE 11 modelling tool. The process of defining – in the MIKE 11 model tool – a model setup, by specifying all required input data describing the geometry, physical processes, inputs and outputs.
Setup MIKE BASIN	Identified and described in Table 3.1
Setup semi-distributed rainfall-runoff model (MIKE SHE Studio)	This is a specific case of “Setup model” when the modelling is related to the MIKE Studio modelling tool. The process of defining – in the MIKE SHE Studio model tool – a model setup, by specifying all required parameters describing the geometry, physical processes, inputs and outputs.
Make decision	Take a decision based on data or simulation output The use case is managed outside the NB DSS system.
Use GIS tool	Identified and described in Table 3.1

Generalised use cases	Description
Use indicator tool	A user uses the indicator tool (which is a collection of processes) to define named indicators in NB DSS. Each indicator is described by data sources, references and processing specifications required for the calculation of the indicator value from data available in the NB DSS.
Use table tool	Identified and described in Table 3.1
Use time series tool	Identified and described in Table 3.1
Visualise time series	Identified and described in Table 3.1
Visualize GIS data	Identified and described in Table 3.1

Figure 3.19 to Figure 3.26 show how the UC-02 use case was used to identify generalised use cases.

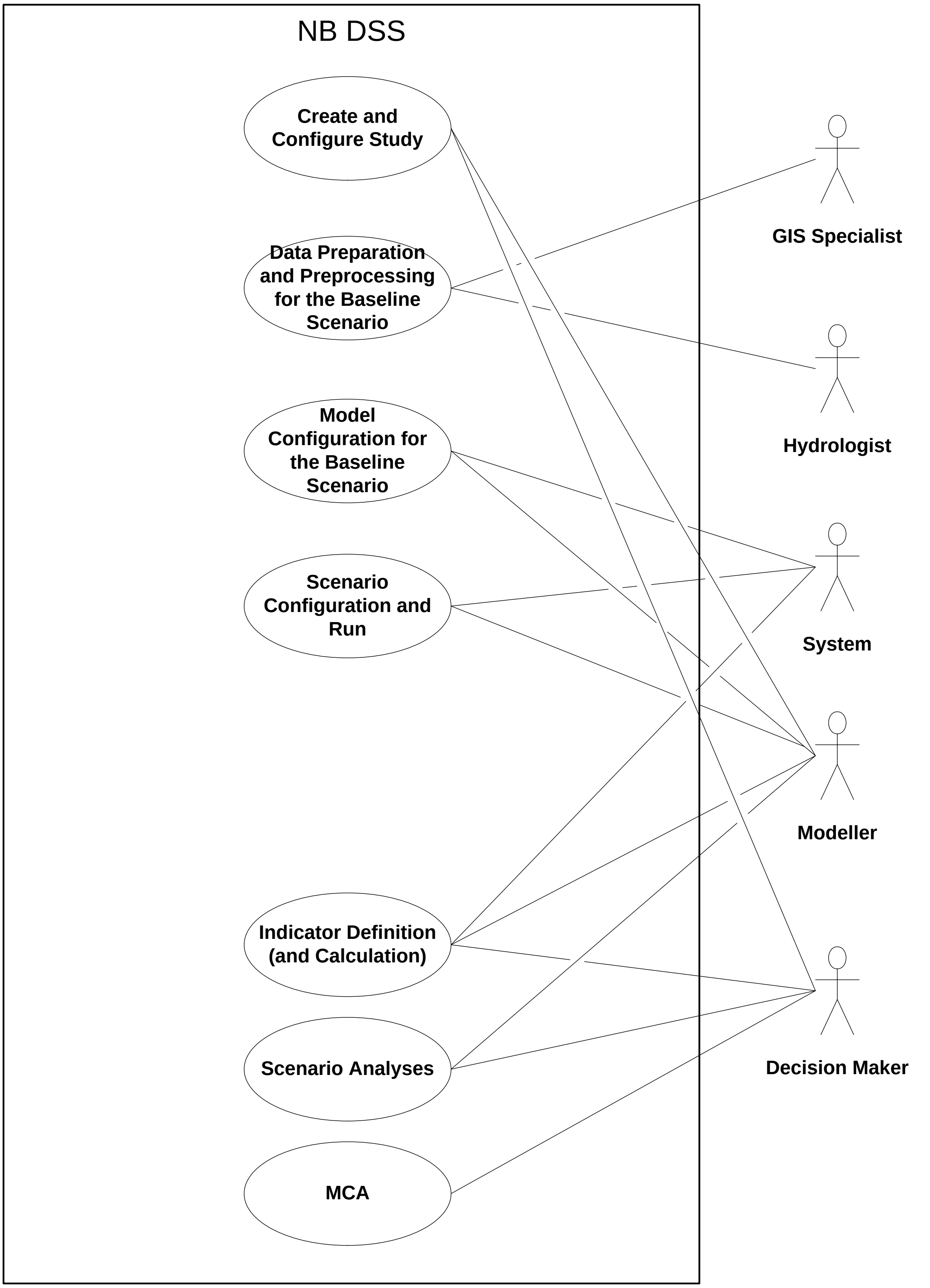

Figure 3.18 Use case 02 – Overview (UML)

Note in Figure 3.18 that:

- Only a normal flow is described in the study

- The Decision Maker is involved from beginning to end in the study by assisting in the definition and configuration of a well defined analysis.

The individual activities from Figure 3.18 are illustrated in the following diagrams.

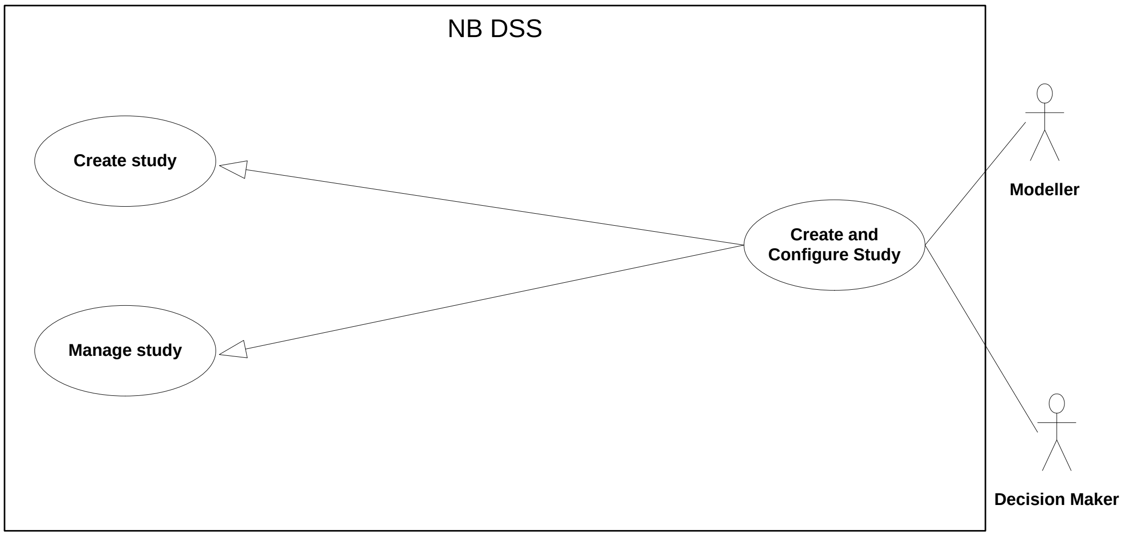

Figure 3.19 Use case 02 - Create and Configure Study (UML)

Note in Figure 3.19 that:

- The different users can create a study
- Creation and Management of study are primarily concerned with setting up user permissions and defining basic properties of a study. The data and other definitions contained within the study are established later.

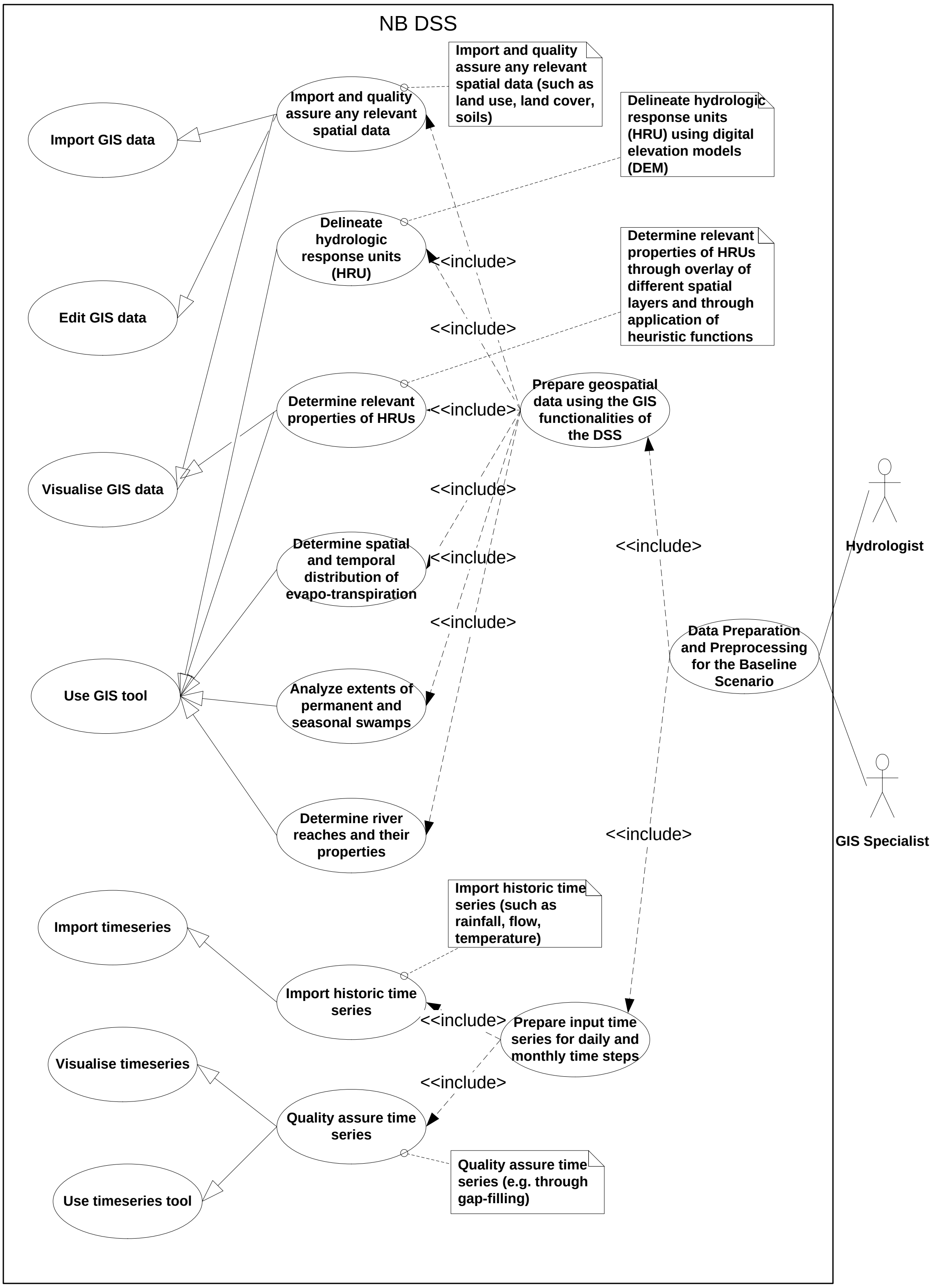

Figure 3.20 Use case 02 – Data Preparation and Preprocessing for the Baseline Scenario (UML)

Note in Figure 3.20 that:

- The preparation of data includes both time series and GIS data. Both follow a similar pattern of

- Import (“Import time series” , “Import GIS data”)
- Edit (“Edit GIS data” or editing using a time series too)
- Use of tools (“Use time series tool”, “Use GIS Tool”)
- Visualisation (“Visualise time series”, “Visualise GIS data”)

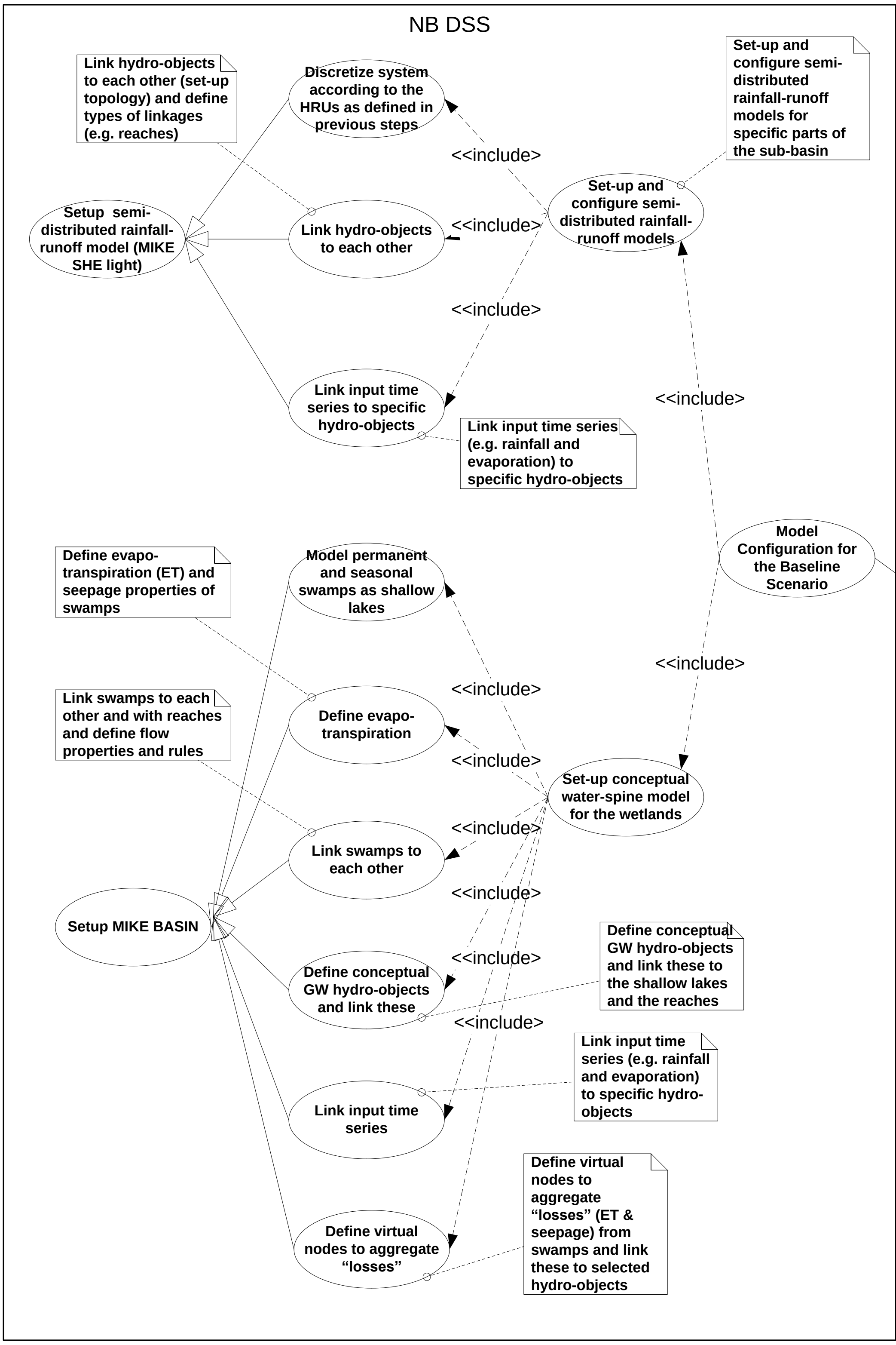

Figure 3.21 Use case 02 – Model Configuration for the Baseline Scenario, part ½ (UML)

Note in Figure 3.21 that

- This figure is the first half of the activities involved in setting up the baseline scenario, illustrating some of the tasks performed by the Modeller actor.

- Setting up the model for the baseline scenario has been divided into a rainfall-runoff model and a water spine model. Depending on the tools employed this may be one step; hence this part of the study also pre-empts some implementation.

- The complete set of steps for setting up the two models largely match what is required for an integrated modelling tool

- Some data prepared previously may have to be transferred from the NB DSS to the model tool for inclusion in the model configuration. This is not illustrated in this use case.

- Additional modelling activities may be part of setting up the model in the model tool. This depends on the model tool employed. Hence, not necessarily all activities involved in setting up a model are included in this study. However, the NB DSS will be able to provide the required data to the model tool to define the numerical definitions required.

- The semi-distributed rainfall-runoff model is translated into the MIKE SHE Studio model tool.

- The “conceptual water spine model” is translated into the MIKE BASIN model tool.

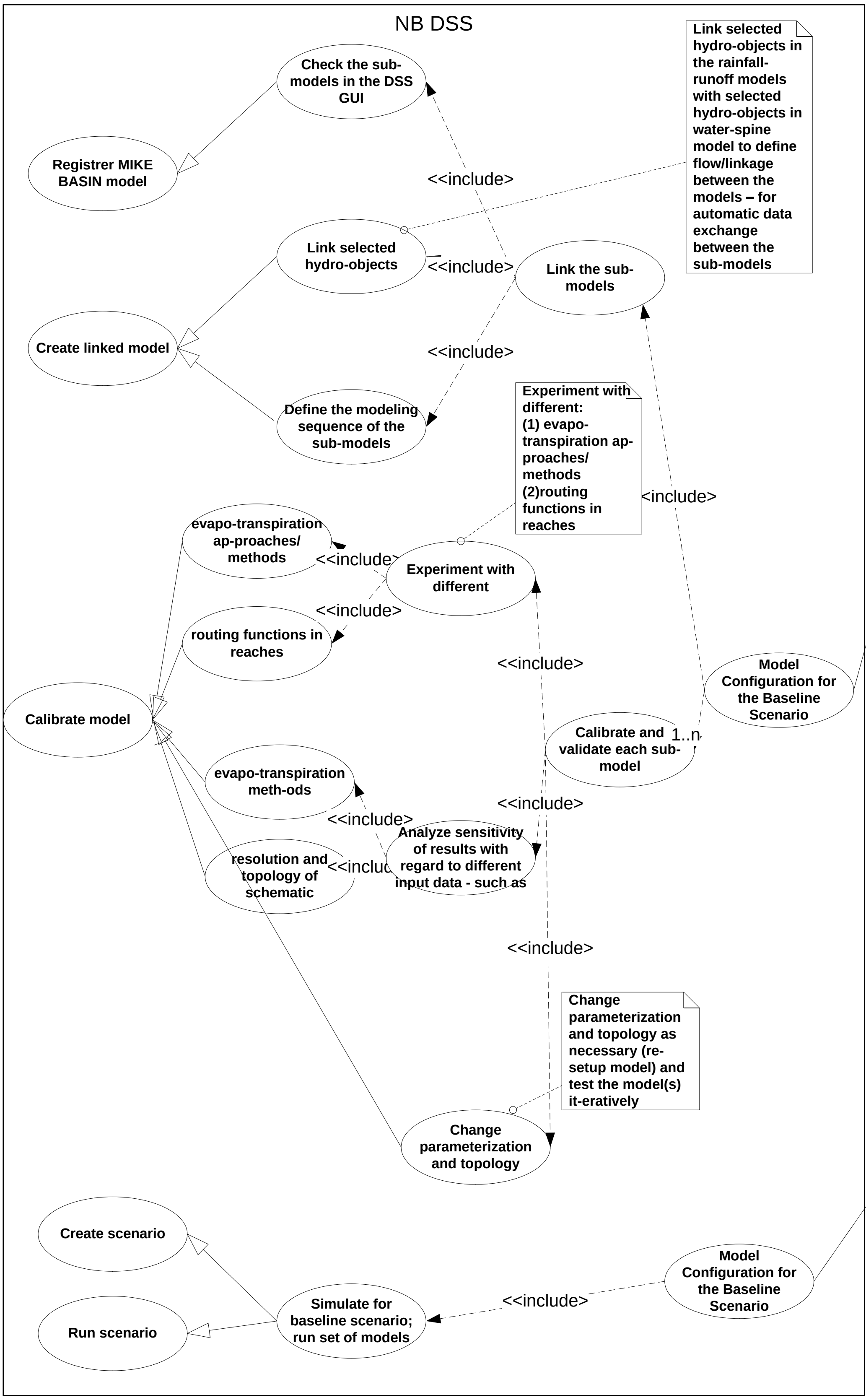

Figure 3.22 Use case 02 – Model Configuration for the Baseline Scenario, part 2/2 (UML)

Note in Figure 3.22 that:

- It shows the second half of the activities required for the baseline scenario configuration, illustrating the remaining tasks performed by of the Modeller actor and those of the System actor.

- Use case “Indicator Definition” (and Calculation) appears twice in the diagram for layout reasons, highlighting subsets of activities by different actors

- “Check sub-model in the DSS GUI” relates to displaying model related details created during registration of the model setup in the NB DSS database.

- “Calibrate” each sub-model details (some of) the activities which can be undertaken for defining the best model. The Consultant sees the majority of this best be performed using a model tool (external to the front end) rather than within the DSS front-end, although some output analysis is available via the NB DSS time series tools.

- “Linking selected Hydro Objects” is interpreted as linking output time series from one model to input nodes in another model, i.e. that linking of models within the NB DSS is performed via time series.

- The actor System is in this use case a proxy for the actor Modeller.

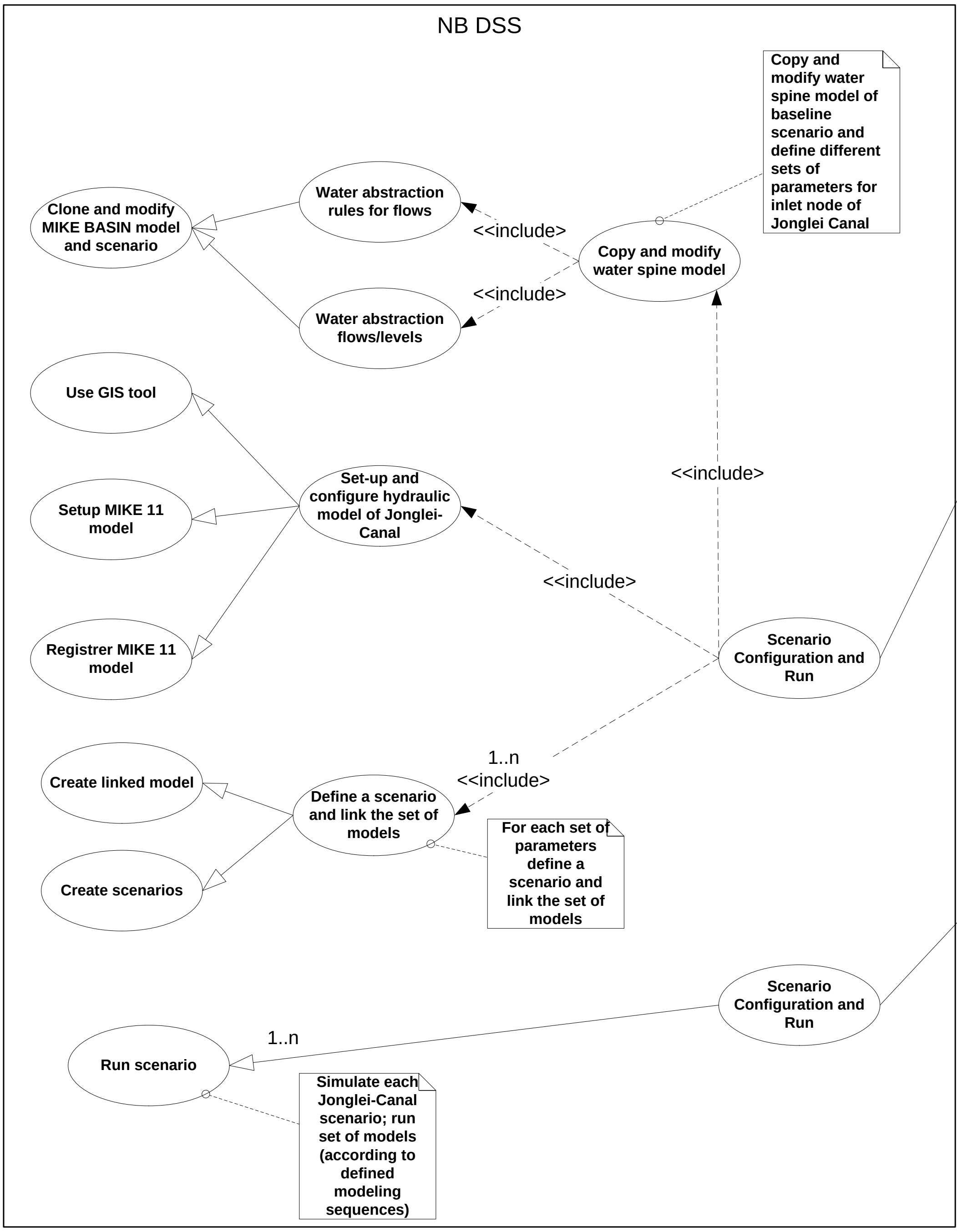

Figure 3.23 Use case 02 – Scenario Configuration and Run (UML)

Note in Figure 3.23 that

- The hydraulic model is translated into the MIKE 11 model tool.
- The same generalised use cases are used to define scenarios for setting up the baseline scenario.
- “Copy and modify water spine model” in NB DSS context includes replicating an existing model setup and scenario followed by a modification of the model setup to include variations in parameters and components.

The actor System is in this use case a proxy for the actor Modeller.

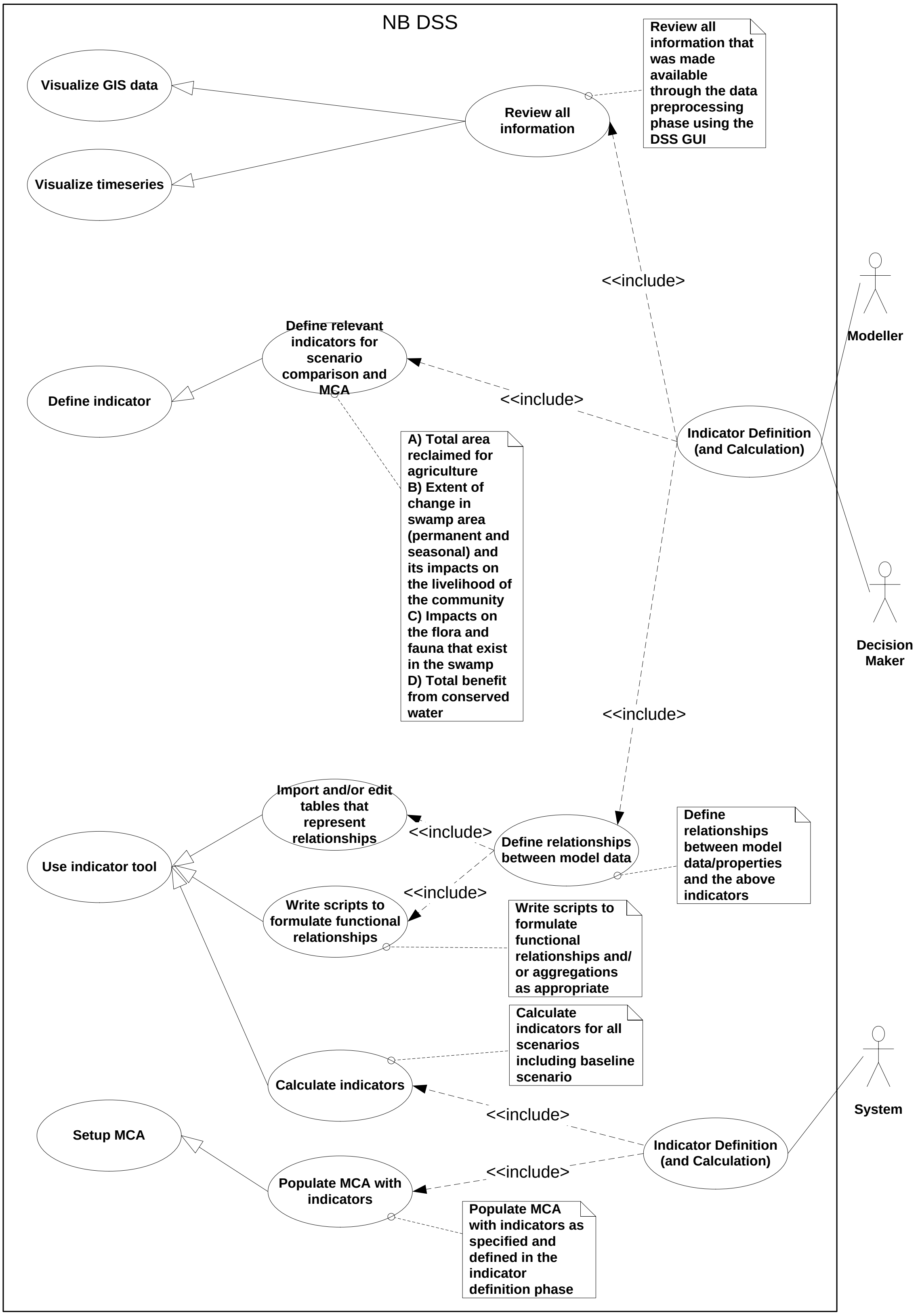

Figure 3.24 Use case 02 – Indicator Definition (and Calculation) (UML)

Note in Figure 3.24 that

- Use case "Indicator Definition (and Calculation)" appears twice in the diagram for layout reasons, highlighting subsets of activities by different actors.
- The "Review of Information" is largely a manual and visual inspection of the available data aided in part by visualization tolls provided by the NB DSS front-end.
- "Calculate indicators" and "populate MCA tool with indicators" are both part of the MCA tool functionality.
- The hydraulic model is translated into the MIKE 11 model tool.

The actor System is in this use case a proxy for the actor Modeller and/or actor Decision Maker.

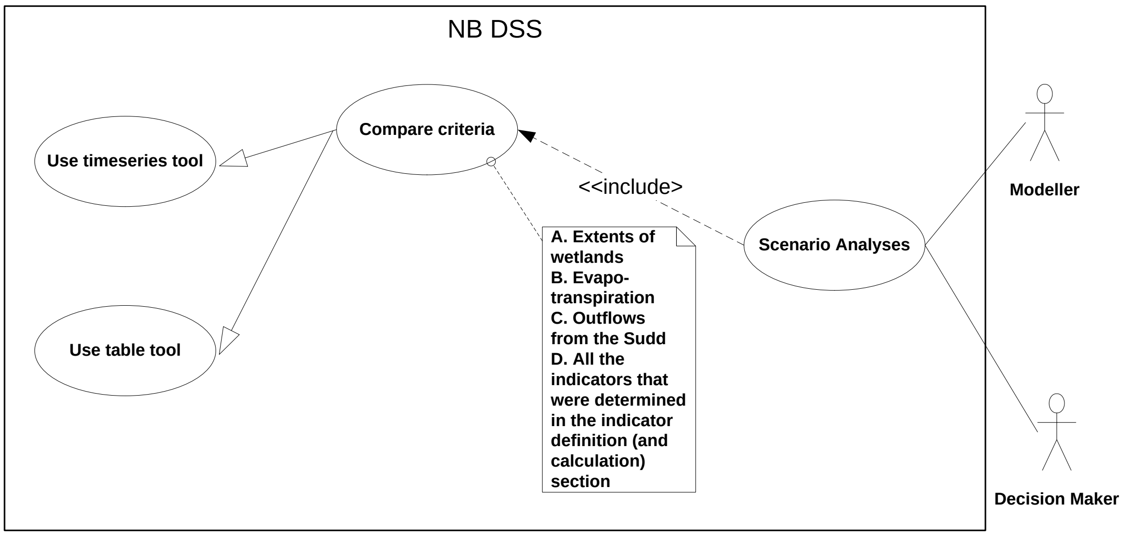

Figure 3.25 Use case 02 – Scenario analyses (UML)

Note in Figure 3.25 that

- The Scenario analysis basically is a comparison of model output data and indicators, i.e. time series and numbers (organised in tables).

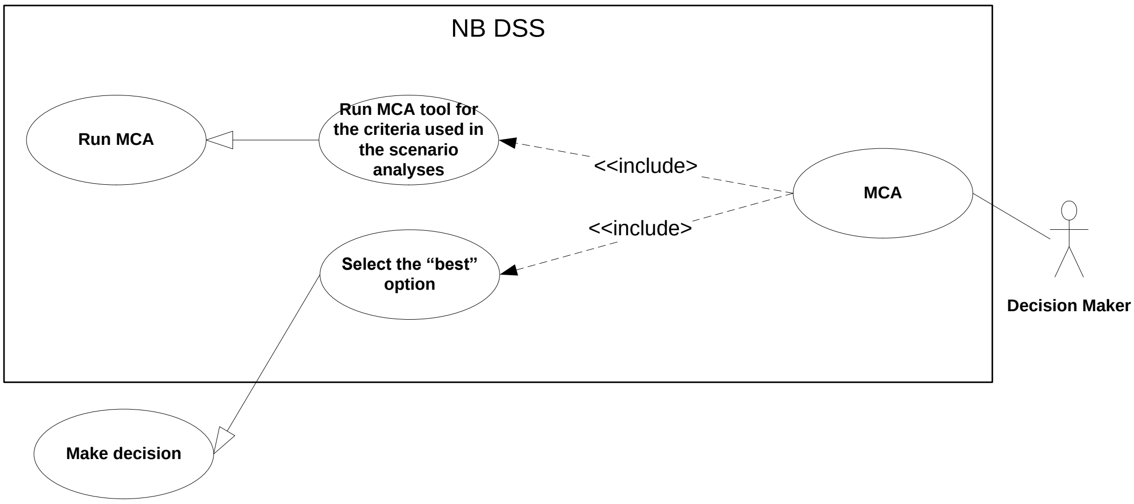

Figure 3.26 Use case 02 – MCA (UML)

Note in Figure 3.26 that

- Running the MCA tool implies that the MCA analysis has been setup and defined in prior steps. Depending on the implementation this may be different, such that setting up is also part of running the MCA tool.
- "Make decision" of selecting the best option is an out-of-system use case involving the Decision Maker.

3.1.3 NB-DSS UC-03: Identify the Preferred Cascade of Power Development and first investment for the Eastern Nile Joint Multipurpose Program (JMP)

> *Under the Nile Basin Initiative (NBI), the Eastern Nile Technical Regional Office (ENTRO) is currently planning a major investment program under the title: 'Join Multipurpose Pr***ogram (JMP)'. The JMP shall include major regulation dams in the Eastern Nile (Blue-Nile – Main Nile sub-basins) with power development, irrigation, flood damage mitigation, improving rural livelihoods and other benefits to the three countries, Egypt, Ethiopia and Sudan. The JMP shall be one of the key application cases for the Nile Basin Decision Support System.**
>
> *This use case focuses on the identification of a preferred cascade for hydropower development and selection of the investment sequencing of the regulation works in the cascade. The exercise shall depend on assessment of potential benefits and impacts under conditions of hydrologic variability to arrive at preferred choice.*
>
> *Being a joint undertaking of three countries, technical experts and decision makers from all three countries and ENTRO shall closely collaborate using the DSS to identify the preferred cascade and the investment sequencing of the dams in the cascade.*

Figure 3.27 Use case 03 - Brief description of the use case quoted from /3/

The use case UC-03 is described in /3/ in the form of a table, which is translated into the overview use case diagram shown in Figure 3.28.

Figure 3.29 to Figure 3.64 show how the use case has been used to identify generalised use cases.

This analysis has identified the use cases listed in Table 3.3. Note that some of the generalised use cases have been identified and described in previous sections; they are included here for reference.

Table 3.3 Use case 03 - Identified generalised use cases

Generalised use cases	Description
Create hydroobject	Creating a new hydro object and store it in the DSS database. Activities involved could comprise selection of hydro object, type, entering characteristics, associating with GIS feature and saving it.
Calibrate MIKE 11	This is specific case of the "Calibrate model" use case when the model is related to the MIKE 11 modelling tool. See "Calibrate model" in Table 3.2
Calibrate MIKE BASIN	Identified and described in Table 3.1
Clone and modify MIKE BASIN model and scenario	Identified and described in Table 3.2

Generalised use cases	Description
Clone and modify model and scenario	Creating a new scenario in the NB DSS as a clone of an existing scenario. Activities involved comprise selection of an existing scenario, cloning it, exporting it as a model setup, using the model tool user interface (UI) to modify the setup and registering the model setup as a new scenario in the NB DSS.
Compare scenarios	Comparing scenarios by analysing the results from the simulated scenarios. Activities involved comprise selection of results from the scenarios, displaying output in charts, calculating and comparing statistics (e.g. in tables) and calculating and comparing indicators.
Create linked model	Identified and described in Table 3.2
Create study	Identified and described in Table 3.2
Create time series	Creating new time series in the DSS Database. This can be accomplished by importing time series, using time series tools, running models or directly by creating them within the NB DSS. This use case is concerned with the latter. Activities involved include the definition of a time axis, data type, data units, entering values, saving the time series under a specified name and associating it with GIS features.
Define scenario	Defining a scenario – in the sense planning the scenario objectives and input. The actual use case takes place outside the NB DSS but could be supported by different data analysis use cases like “Visualise time series” and “Visualise GIS”
Edit GIS data	Identified and described in Table 3.2
Edit hydroobject	Modifying an existing hydro object Activities involved comprise searching for hydro objects, selecting a hydro object, modifying characteristics and saving it

Generalised use cases	Description
Edit time series	Modifying a time series Activities involved comprise selecting the time series, adding it to a data table, editing time stamps and/or values and saving it. This can happen as a manual or automated process.
Export GIS data	Export GIS data from the NB DSS database to a native modelling tool or the file system. Activities involved comprise selection of the features that shall be exported and selection of output format.
Export study	Exporting study data Activities involved comprise selecting the study, initiating an export wizard, selecting the data that shall be exported and executing the actual export.
Export time series data	Export time series data from the NB DSS database to a native modelling tool or the file system. Activities involved comprise selection of the time series that shall be exported and selection of output format
Import GIS data	Identified and described in Table 3.2
Import Study	Importing study data Activities involved comprise selecting the study to which the data should be imported, initiating an import wizard, browsing for the file with study data to be imported and executing the actual import.
Import tables	Importing tables into the DSS Database Activities involved comprise selection of a table file on the file system, importing the file, validating the file against a format specification, providing table meta-data and storing it in the database.
Import time series	Identified and described in Table 3.1

Generalised use cases	Description
Inspect GIS layers	Identified and described in Table 3.1
Link time series to feature	Associating time series with features Activities involved comprise selection of one or more time series, selection of GIS features and executing the association process.
Publish in report	Identified and described in Table 3.1
Register MIKE 11 model	Identified and described in Table 3.2
Register MIKE BASIN model	Identified and described in Table 3.2
Run ensemble scenario	Execution of a scenario that uses time series ensembles as input Activities involved include selecting a scenario having ensembles as input and initiating the simulation
Run MCA	Identified and described in Table 3.2
Run scenario	Identified and described in Table 3.1
Setup ensemble scenario	Defining a scenario that shall use time series ensembles as input Activities involved comprise selecting a scenario, initiating a “New scenario” wizard, specifying ensemble(s) as input and selecting ensembles
Setup MCA	Identified and described in Table 3.2
Setup MIKE 11 model	Identified and described in Table 3.2
Setup MIKE BASIN	Identified and described in Table 3.1
Make decision	Identified and described in Table 3.1

Generalised use cases	Description
Setup CBA	Defining the components to be used during the CBA Activities involved comprise defining initial and recurring costs for infrastructure components (Hydro objects) and named indicators (see the use case “Use indicator tool”) through relationship tables (to be defined using the Table tool).
Run CBA	Populating the CBA tables with actual costs and benefits Activities involved comprise setting variables required to compute present value amounts (e.g., analysis horizon, interest rates (for loans and savings), inflation rates, time when costs will be incurred or benefits will be realized) and initiating the computation of present value costs and benefits using said variables and the costs/benefits defined during the “Setup CBA” use case.
Use demand calculator tool	Calculating crop water demands. Activities involve comprise setting up the crop water demand calculator tool (based on FAO56) by defining required inputs such as climate, crop type. Irrigation technology and soil characteristics.
Use ensemble generator tool	Generation of ensembles of time series Activities involved could comprise starting the tool, specifying a master input time series, define perturbations, define the number of time series that shall be generated and running the tool
Use GIS tool	Identified and described in Table 3.1
Use rainfall runoff model	Identified and described in Table 3.1
Use soil erosion tool	Calculate sediment loads from catchment(s). Activities involved will be related to specify the required input for the soil erosion model. A soil erosion model for calculating long term average erosion rates will be built into the NB DSS (e.g. based on USLE/RUSLE). Inputs such as land slope, soil type and climate would be required.

Generalised use cases	Description
Use time series tool	Identified and described in Table 3.1
Visualise tables	Identified and described in Table 3.1
Visualise time series	Identified and described in Table 3.1
Visualize GIS data	Identified and described in Table 3.1

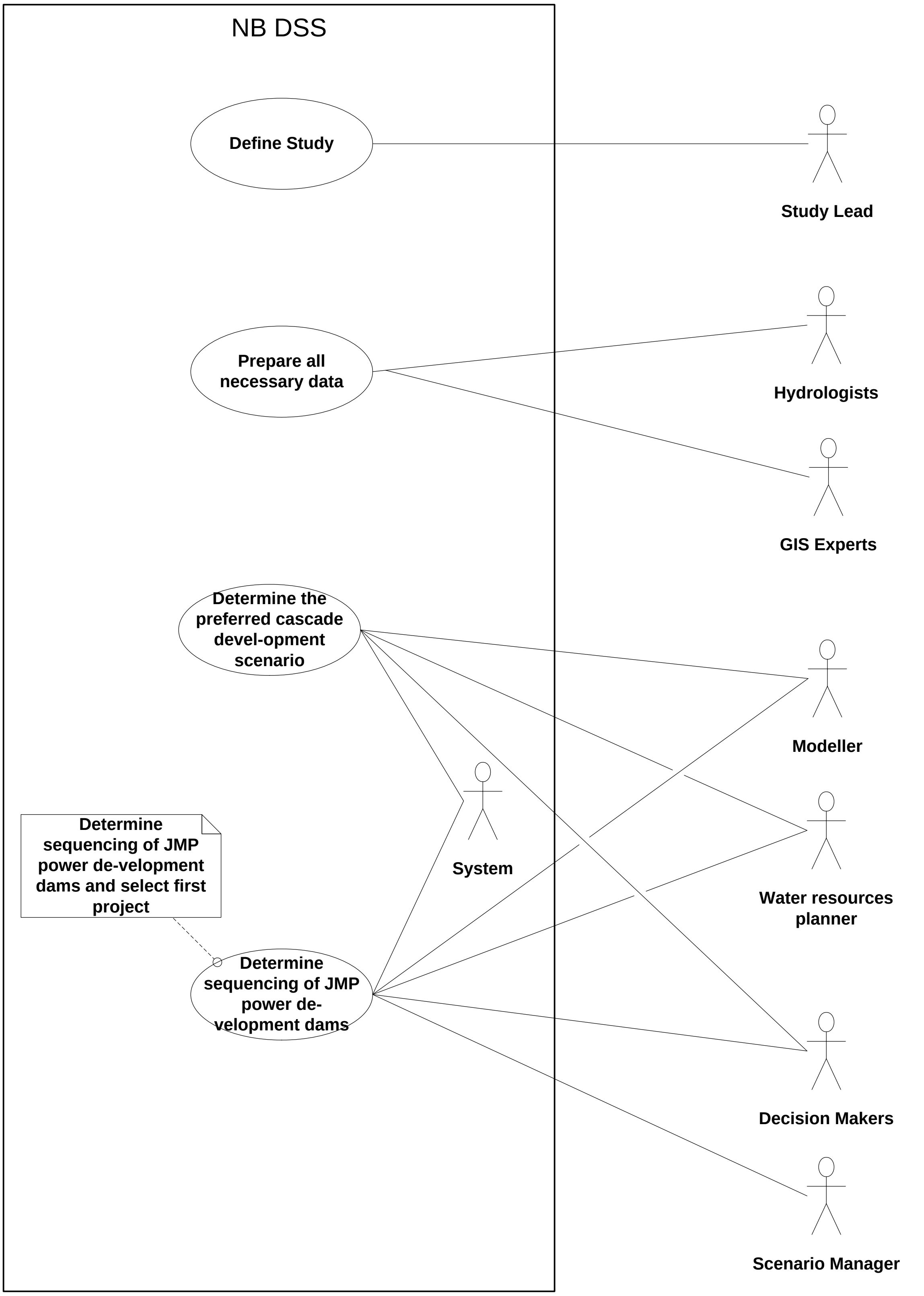

Figure 3.28 Use case 03 – Overview (UML)

Note in Figure 3.28

- The use case consists of 4 sub-use cases and 7 different actors.

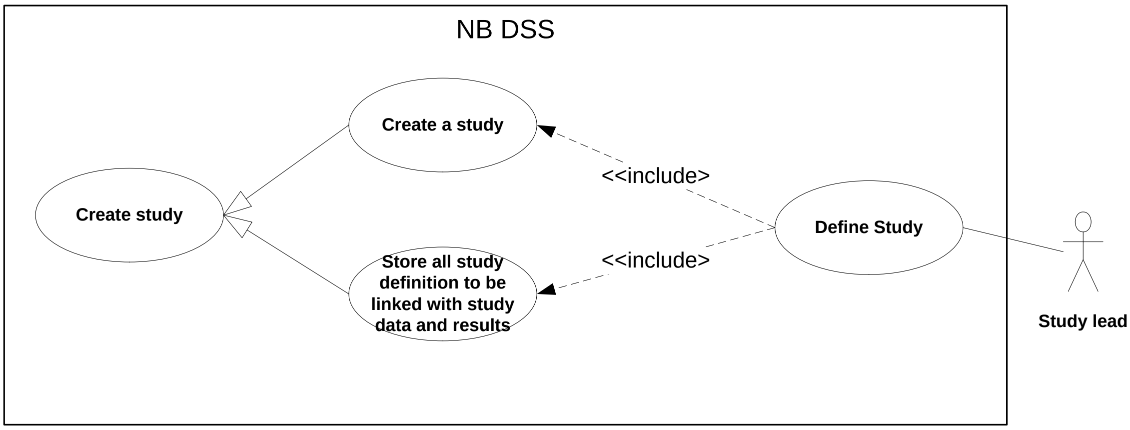

Figure 3.29 Use case 03 – Define study (UML)

Note in Figure 3.29 that "Create study" involves both "Create a study" and "store all study ...". In fact all data created while being logged on to a study will automatically be associated with that study.

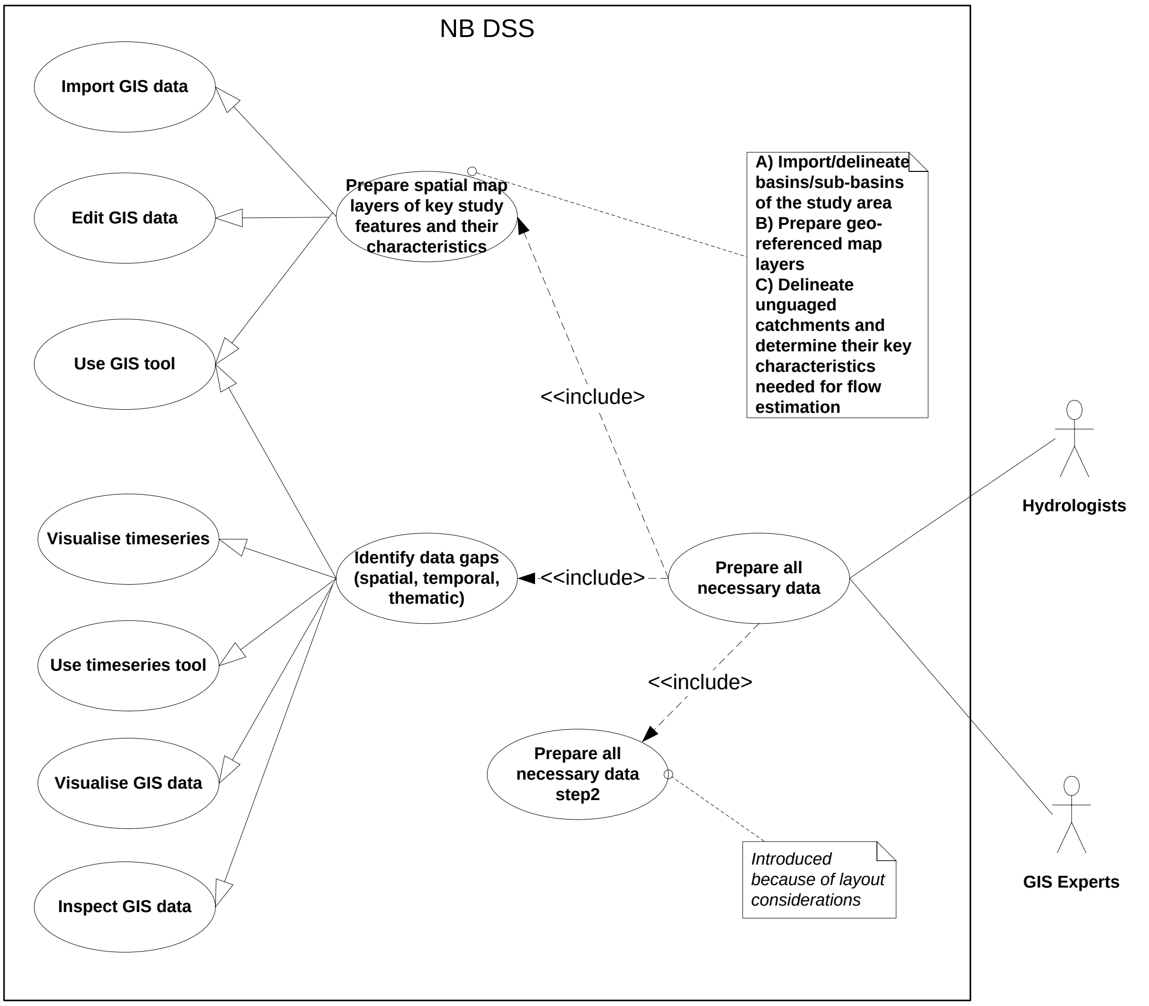

Figure 3.30 Use case 03 - Prepare all necessary data (UML)

Note in Figure 3.30 that

- Data preparation involves visualization, editing, looking at meta-data and processing activities.
- The “Prepare all necessary data step 2” use case is included to split the use cases over two diagrams (Figure 30 and Figure 3.31).

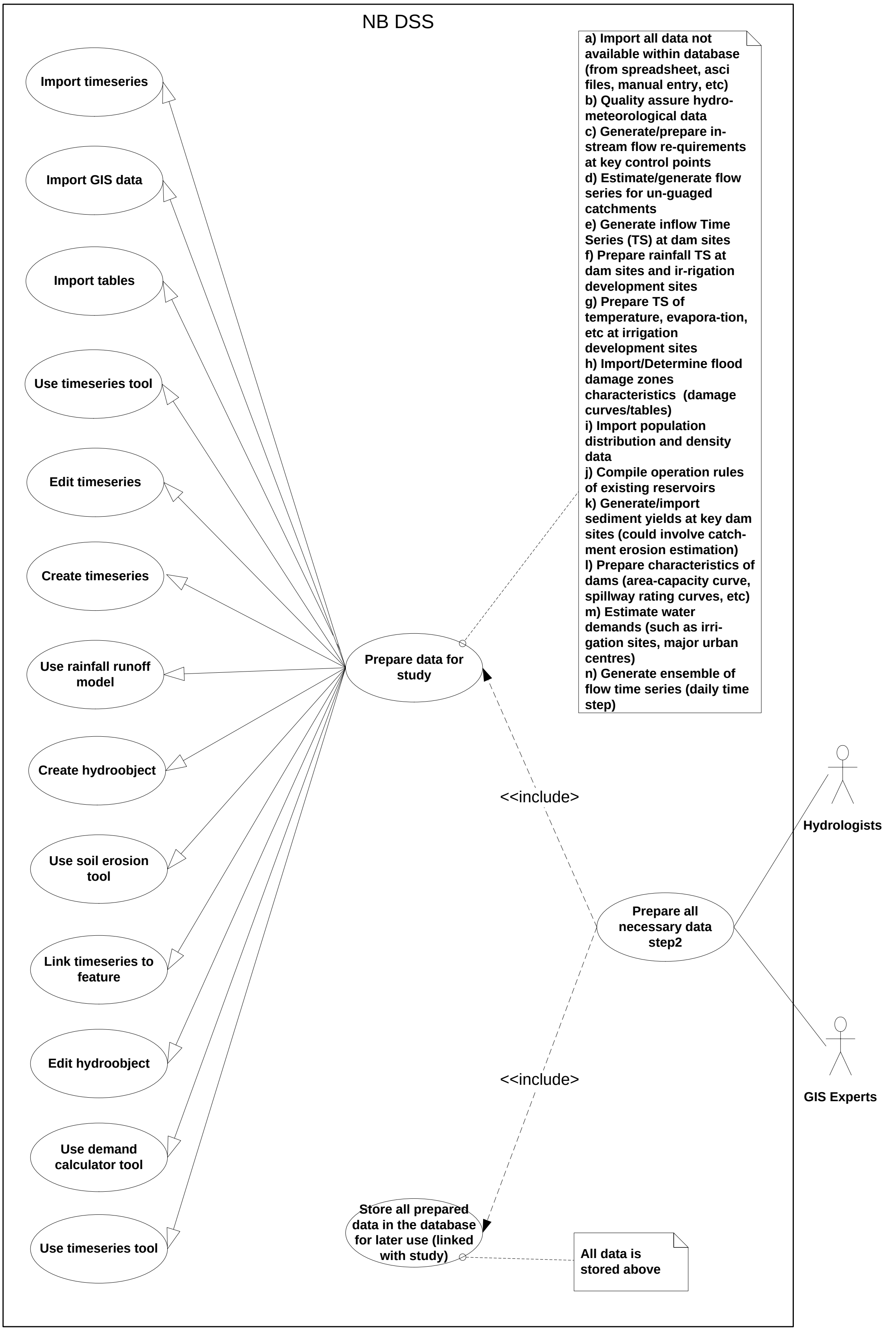

Figure 3.31 Use case 03 - Prepare all necessary data step 2 (UML)

Note in the Figure 3.31 that

- “Prepare data for study” includes a wide variety of activities. In order not to explode the diagram into many more diagrams the activities involved are described in the UML comment and the generalised use cases are linked directly to the “Prepare data for study” use case.

- The “Create hydroobject” and “Edit hydroobject” use cases refer to directly defining hydro objects; i.e., not implicitly through model registration

- The “Prepare all necessary data step 2” use case is included to split the use cases over two diagrams (Figure 30 and Figure 3.31).

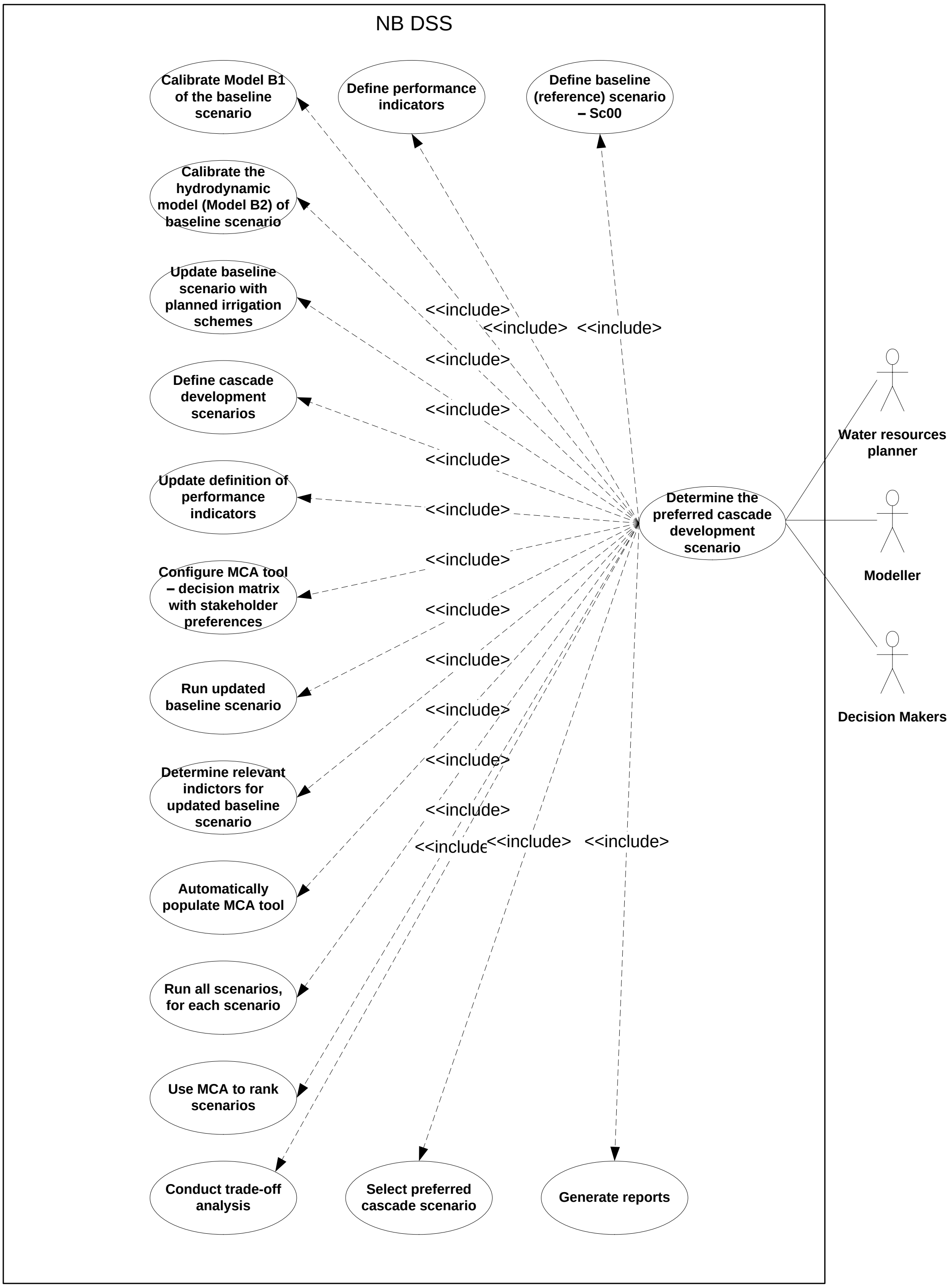

Figure 3.32 Use case 03 – Overview of the "Determine the preferred cascade development scenario" (UML)

Note in Figure 3.32 that the use case includes 16 sub-use cases. Each of these 16 sub-use cases is shown and analysed in Figure 3.33 to Figure 3.48.

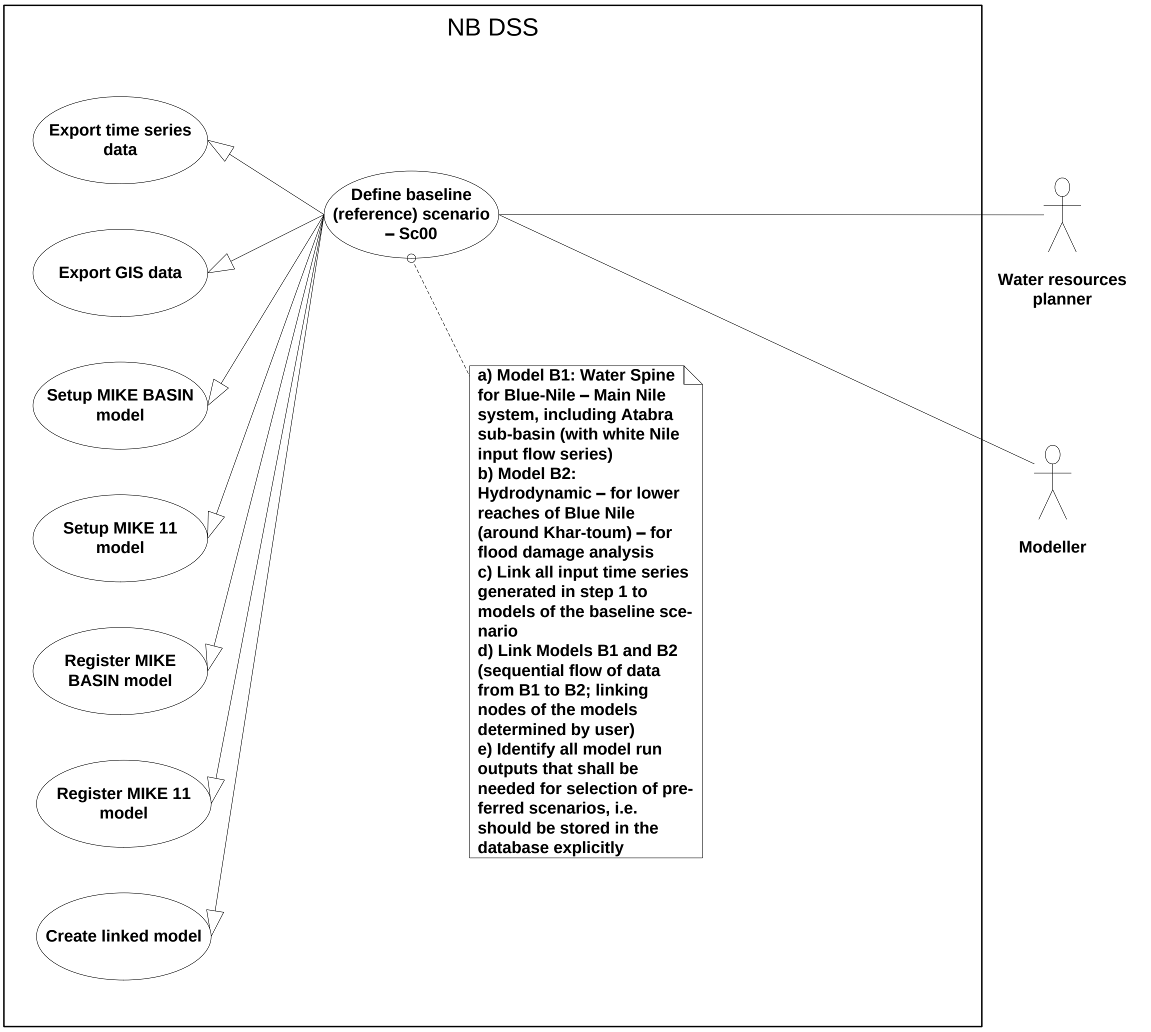

Figure 3.33 Use case 03 -Define baseline (reference) scenario (UML)

Note in Figure 3.33 that defining a baseline scenario includes setting up a model, calibrating it and registering it with the DSS. In this case the 2 registered models (MIKE BASIN and MIKE 11) shall also be linked, which necessitates the "Create linked model" use case. Also note that in order to setup a new model it is often required to export data from the NB DSS database.

The "Define baseline .." use case is described through its activities in a UML comment to prevent exploding` the diagram into many more diagrams.

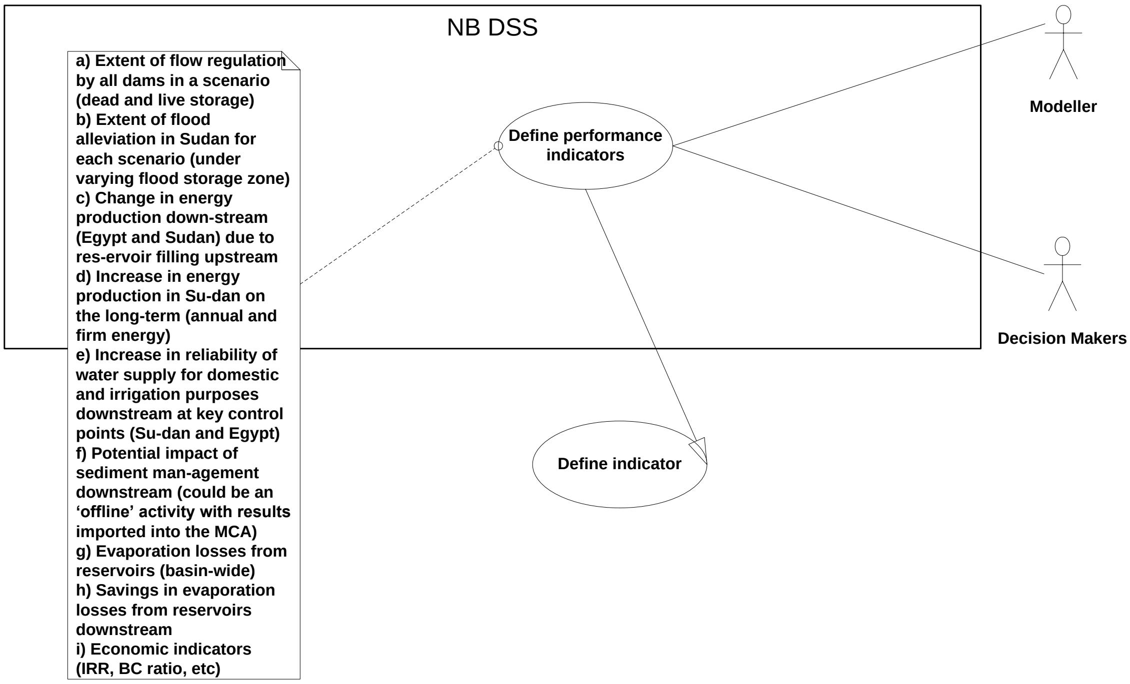

Figure 3.34 Use case 03 - Define performance indicators (UML)

Note in Figure 3.34 that the "Define performance indicator" maps directly to the "Define indicator" generalised use case. Also note that this use case occurs outside the system, see the description of the use case in Table 3.2.

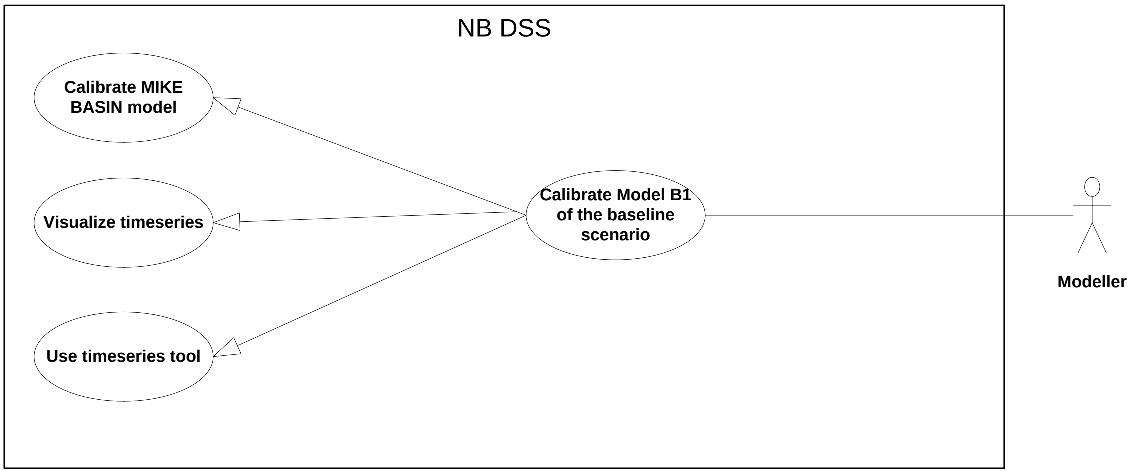

Figure 3.35 Use case 03 - Calibrate model B1 of the baseline scenario (UML)

Note in Figure 3.35 the use of time series visualisation and processing to compare observed and simulated values and to calculate performance statistics.

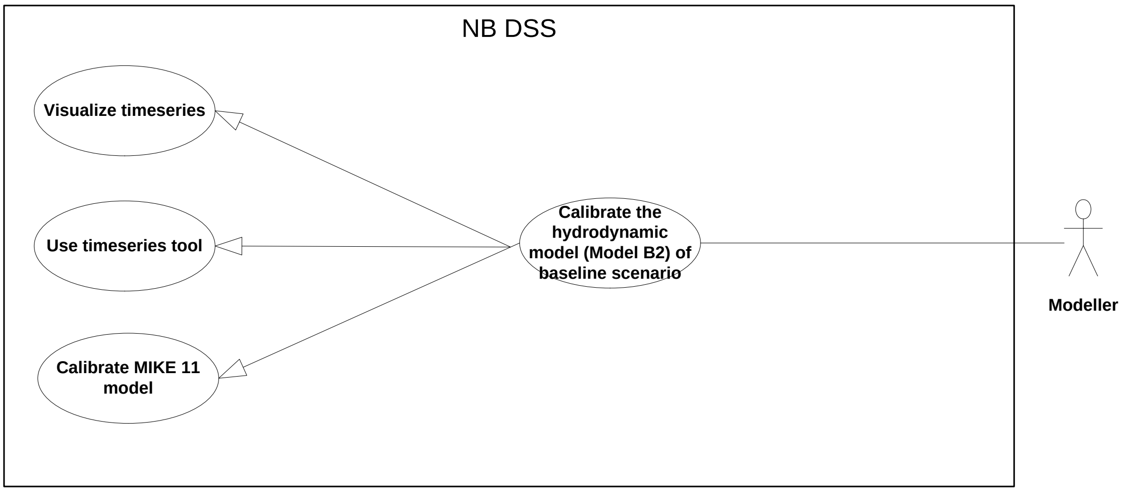

Figure 3.36 Use case 03 – Calibrate the hydrodynamic model (UML)

Note in Figure 3.36 that the calibration of the hydrodynamic model involves the same activities as the calibration of the MIKE BASIN water budget model, see Figure 3.35.

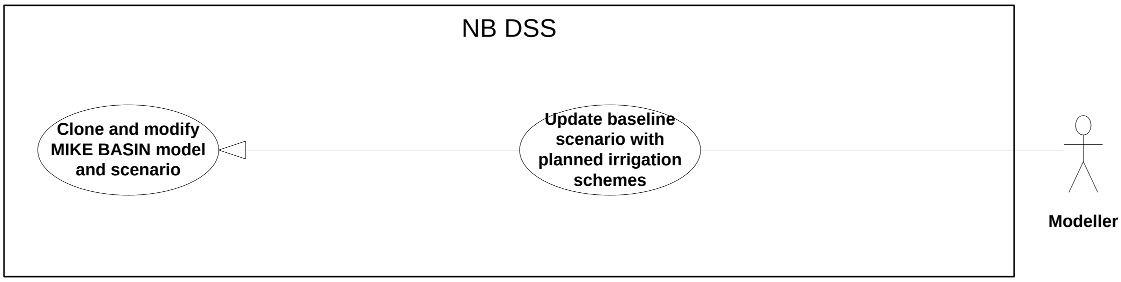

Figure 3.37 Use case 03 - Update baseline scenario with planned irrigation schemes (UML)

Note in Figure 3.37 that updating the baseline scenario with planned irrigation schemes maps directly to the “Clone and modify MIKE BASIN model and scenario” generalised use case.

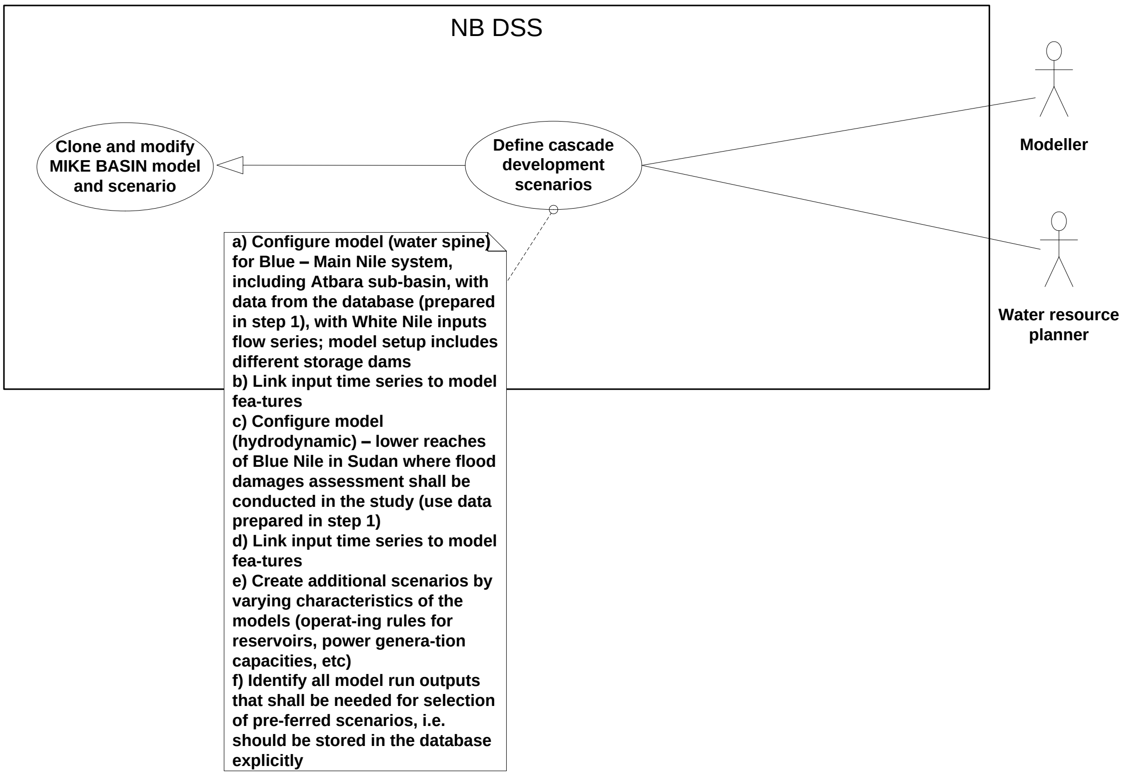

Figure 3.38 Use case 03 - Define cascade development scenarios (UML)

Note in Figure 3.38 has collapsed all activities performed under "Define cascade development scenarios" to the "Clone and modify MIKE BASIN model and scenario" use case. Each of the activities specified in the UML comment is included in the generalised use case.

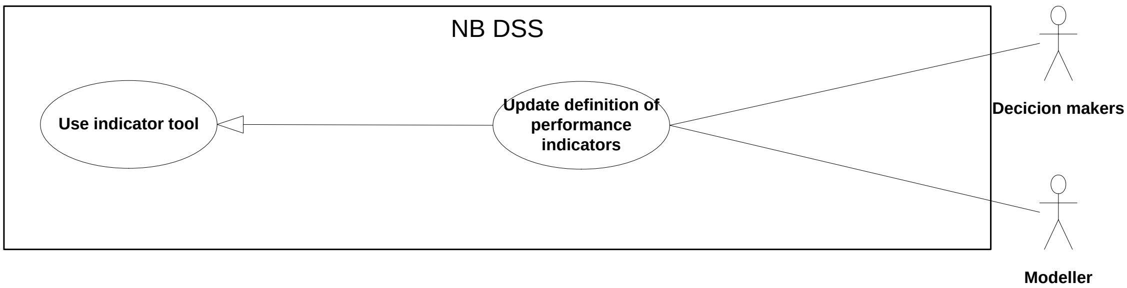

Figure 3.39 Use case 03 - Update definition of performance indicators (UML)

Note in the Figure 3.39 that updating the definition of the performance indicators employs the "Use indicator tool" use case. In this case this involves updating of the data source behind the indicators defined in Figure 34.

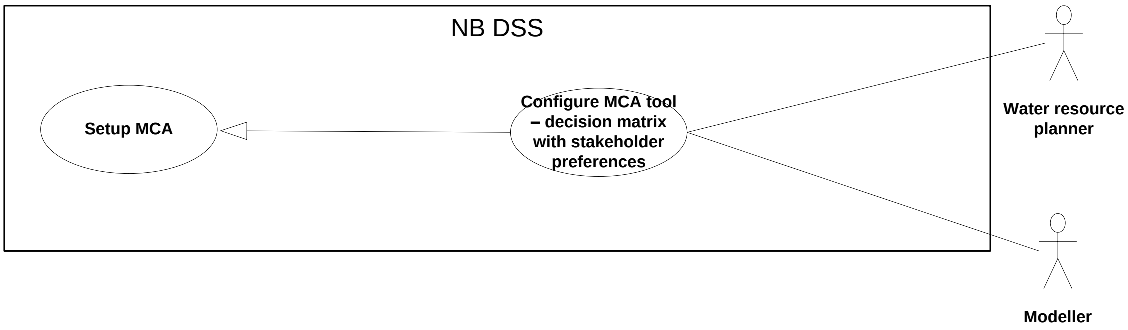

Figure 3.40 Use case 03 – Configure MCA tool – decision matrix with stakeholder preferences (UML)

Note in Figure 3.40 that the “Configure MCA tool ...” use case maps directly to the generalised use case “Setup MCA”. In this case this involves defining the decision matrix.

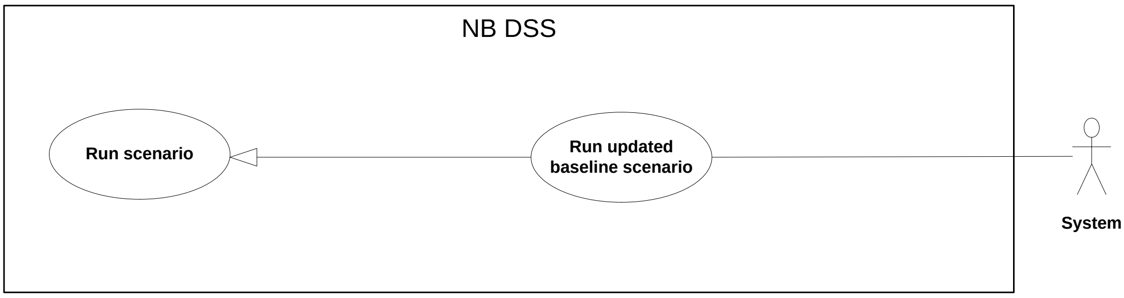

Figure 3.41 Use case 03 – Run updated baseline scenario (UML)

Note in Figure 3.41 that there is a one-to-one correspondence between the specific and generalised use case. The actor “System” shown in the figure is a proxy for the actor Modeller, who is the one actually initiating the use case, see Run Scenario in Figure 3.44

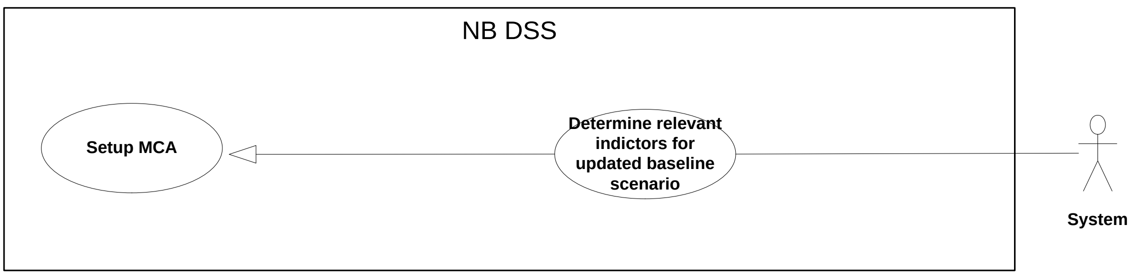

Figure 3.42 Use case 03 – Determine relevant indicators for updated baseline scenario (UML)

Note in Figure 3.42 that the “determine relevant..” use case is understood as updating the data source and/or the data source processing rules. This is included in the “use indicator tool” use case. The actor “System” shown in the figure is a proxy for the actor Modeller, who is the one actually initiating the use case, see Run Scenario in Figure 3.44

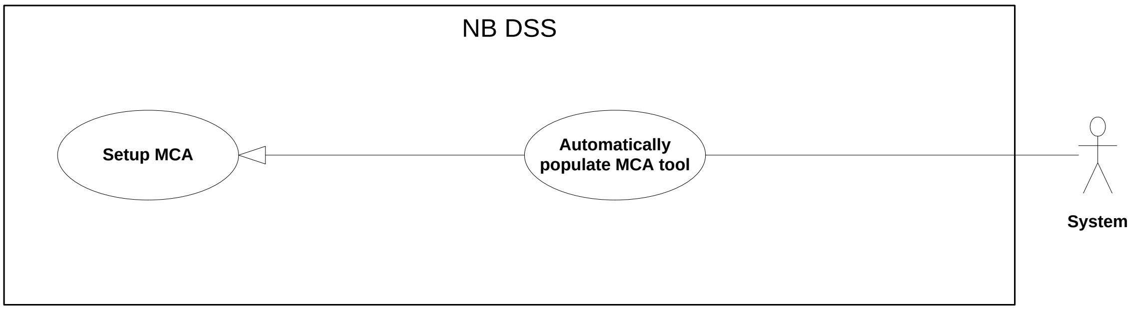

Figure 3.43 Use case 03 – Automatically populate MCA tool (UML)

Note in Figure 3.43 that the automatic population of the MCA happens when setting-up the tool. In this case this happens automatically through applying a user defined function. The actor "System" shown in the figure is a proxy for the actor Modeller, who is the one actually initiating the use case, see Run Scenario in Figure 3.44

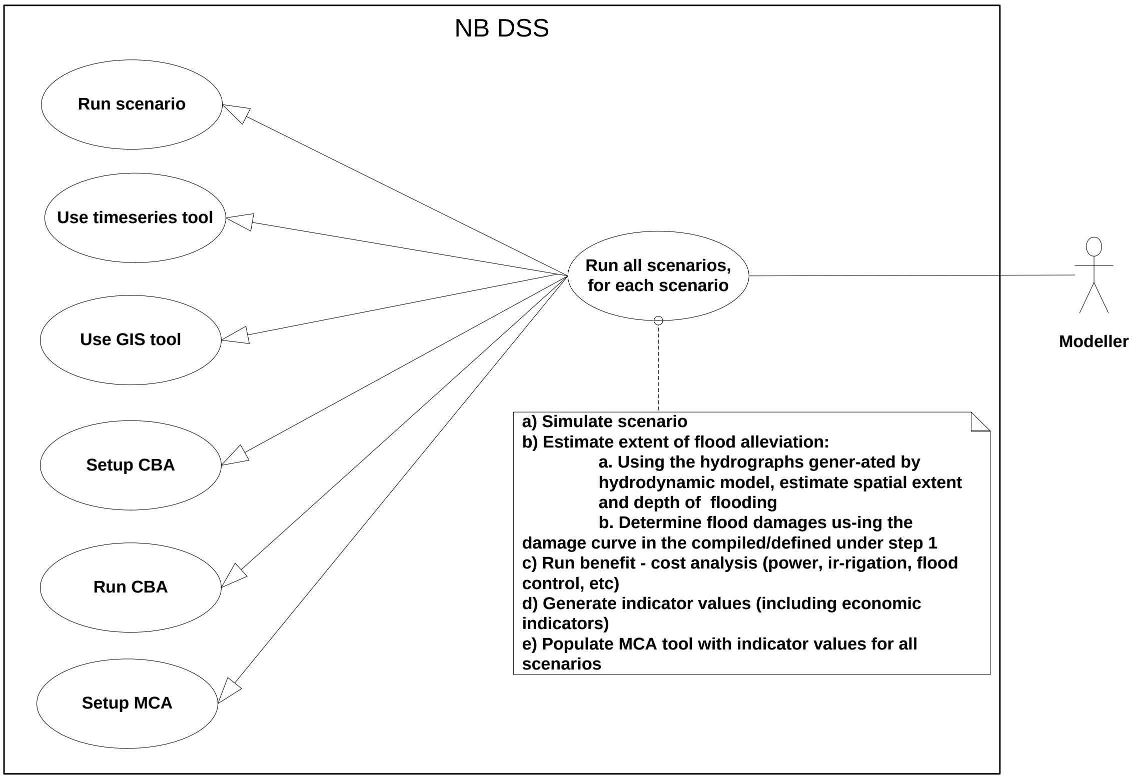

Figure 3.44 Use case 03 – Run all scenarios (UML)

Note in Figure 3.44 that

- "Run scenario" triggers execution of the use cases shown in Figure 3.41, Figure 3.42 and Figure 3.43.
- "Setup CBA" includes the definition of costs for building and maintaining infrastructure components (represented by hydro objects in the NB DSS), and defining indicators that can be related to the expected costs and benefits resulting from the analysed infrastructure scenario.

- “Run CBA” focuses on defining the analysis variables (see Table 3) and calculating actual costs and benefits in monetary units.

- “Run MCA” includes the setting of stakeholder weights, selecting scenarios to be included in the analysis and calculating and visualising the actual criteria values.

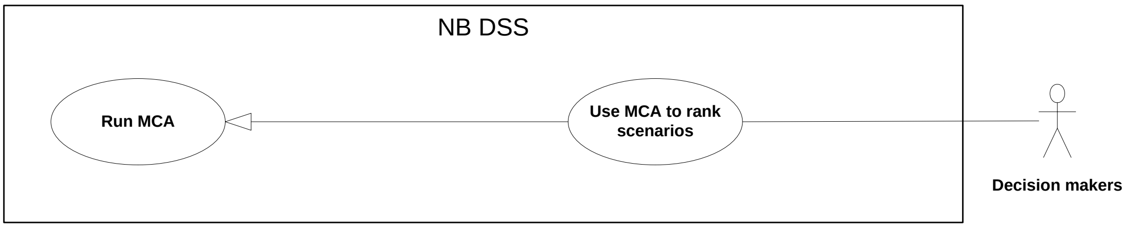

Figure 3.45 Use case 03 - Use MCA to rank scenarios (UML)

Note in Figure 3.45 that the individual criteria values in the decision matrix (or the reference point graph) are used to establish a rank for each scenario.

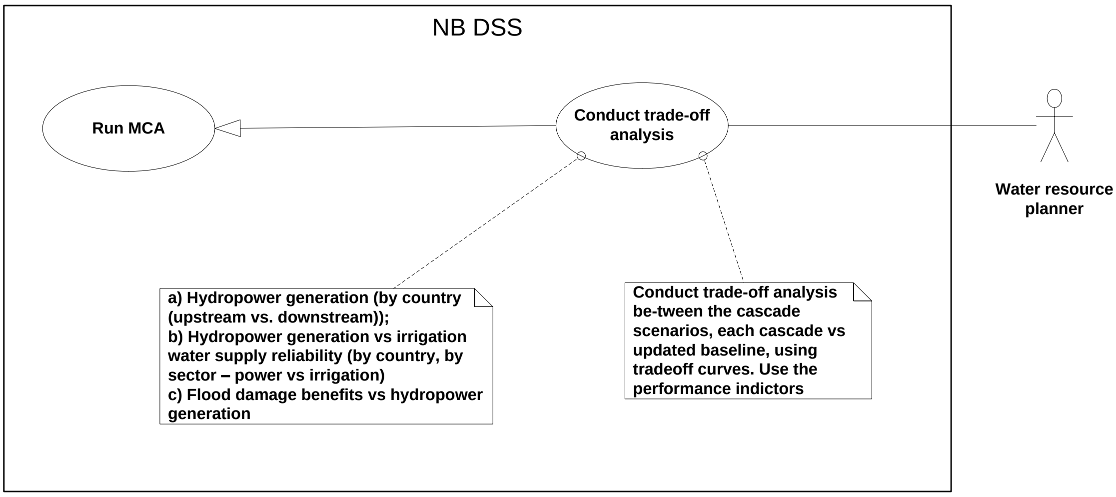

Figure 3.46 Use case 03 – Conduct trade-off analysis (UML)

Note in Figure 3.46 that the MCA rank established for each scenario is strongly dependent on the stakeholder weights, which are varied to conduct a trade-off analysis.

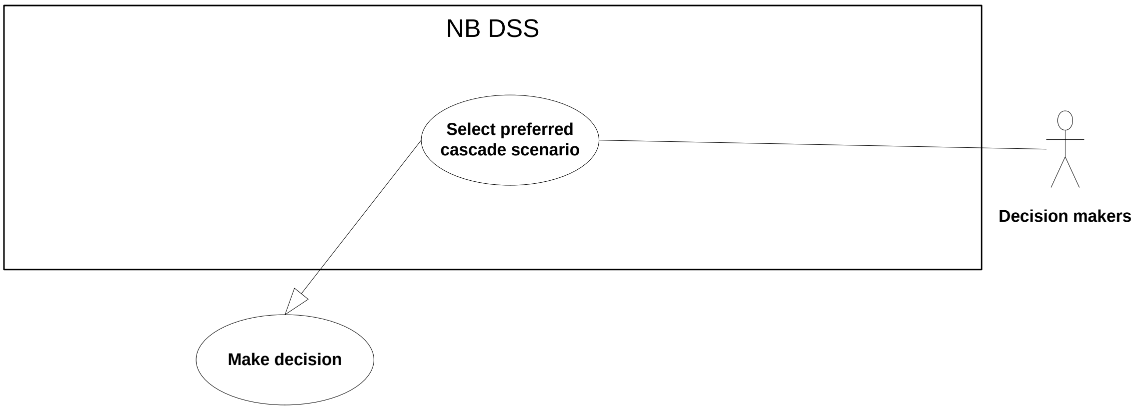

Figure 3.47 Use case 03 – Select preferred cascade scenario (UML)

Note in Figure 3.47 that “Make decision” is a use case that happens outside of the NB DSS system.

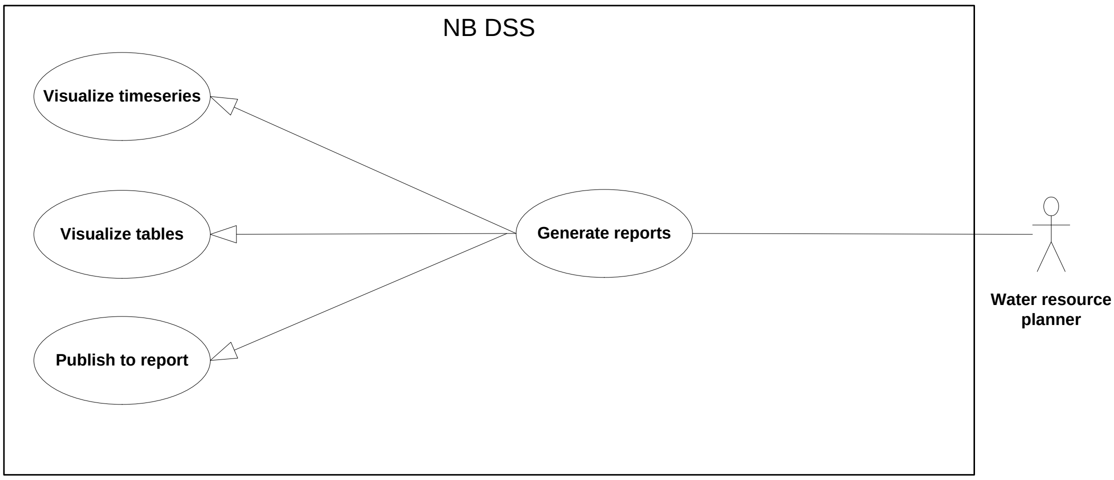

Figure 3.48 Use case 03 – Generate reports (UML)

Note in Figure 3.48 that “Generate reports” is repeating a previous used pattern for report generation, “Visualise time series”, “Visualise tables” and “Publish to report”.

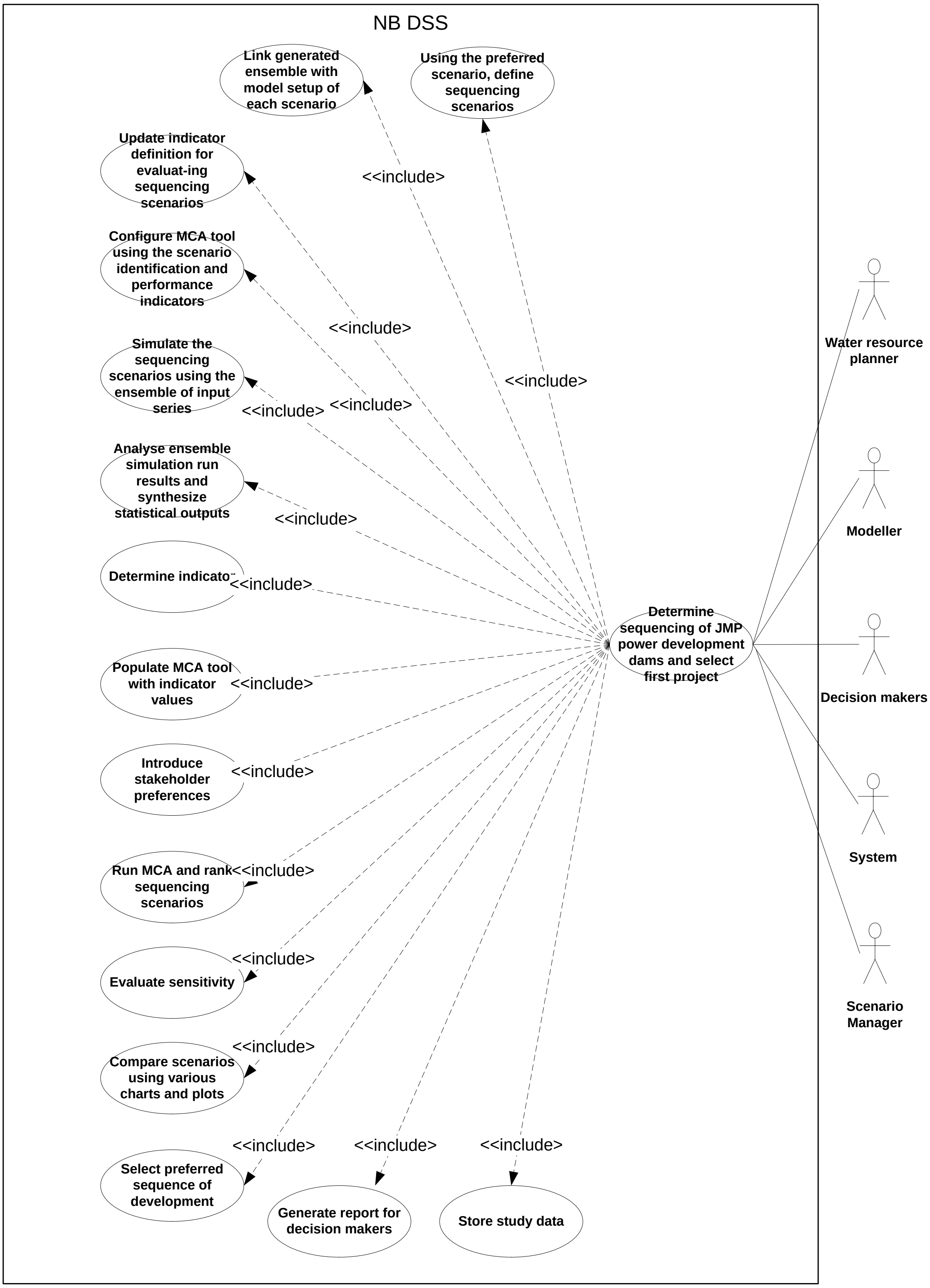

Figure 3.49 Use case 03 – Overview of "Determine sequencing of JMP power development dams and select first project" (UML)

Note in Figure 3.49 that the "Determine sequencing of JMP power development dams and select first project" use case is complex. It involves 15 sub-use cases and 5 actors.

One of those is a System actor that during simulation acts on behalf of one of the other "physical" actors. The use case is analysed in Figure 3.50 to Figure 3.64.

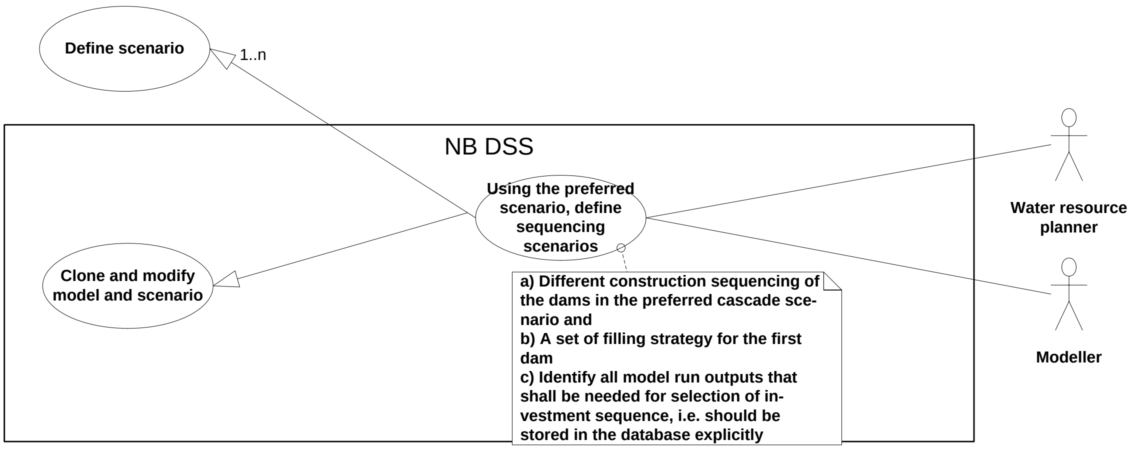

Figure 3.50 Use case 03 – Using the preferred scenario, define sequencing scenarios (UML)

Note in Figure 3.50 that the definition of scenarios, i.e. the planning of the scenarios to establish, is performed outside the system. It merely involves defining sequencing order and filling strategies. After this has been decided, the scenarios shall be prepared. This happens through the "Clone and modify model and scenario" use case.

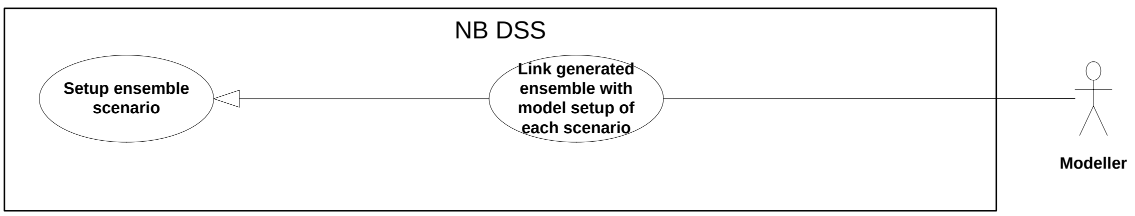

Figure 3.51 Use case 03 – Link generated ensemble with model setup of each scenario (UML)

Note in Figure 3.51 that the "Link generated ensemble with .." use case maps directly to the "Setup ensemble scenario" generalised use case.

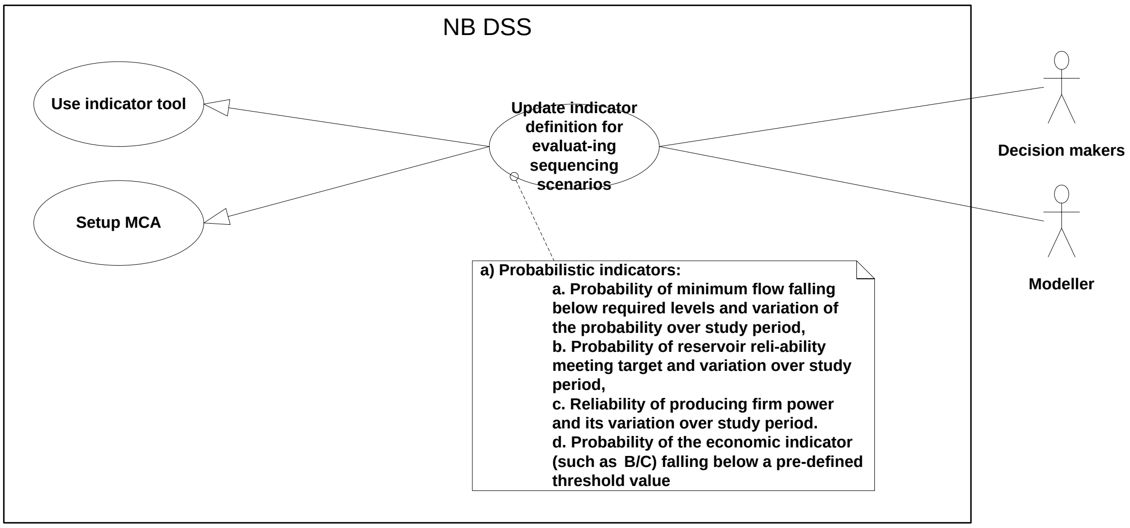

Figure 3.52 Use case 03 – Update indicator definition for evaluating sequencing scenarios (UML)

Note in Figure 3.52 the use of probabilistic indicators, which can be directly used as criteria in the MCA, but require the use of ensemble scenarios. The definition of these indicators occurs outside of the NB DSS, but the defined indicators must subsequently be configured in the NB DSS. This involves the definition of data sources and evaluation processes (e.g., number of times that the flow falls below the required minimum), as described in the “Use indicator tool” use case.

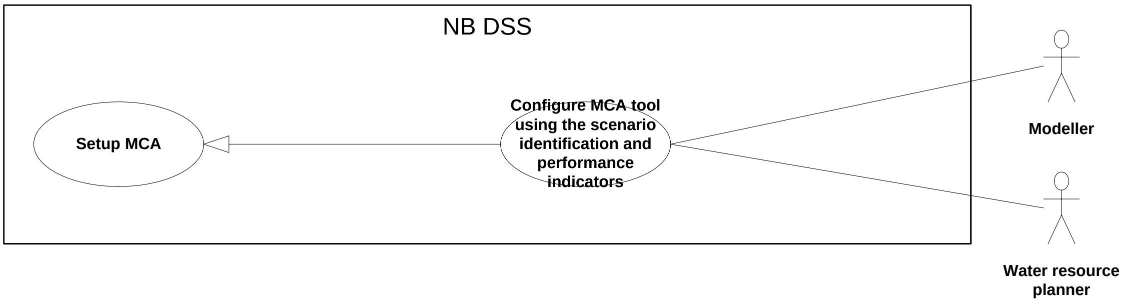

Figure 3.53 Use case 03 – Configure MCA tool using performance indicators (UML)

Note in Figure 3.53 that “Setup MCA” use case focuses on defining the performance indicators to be used in the MCA in the calculation of MCA criteria.

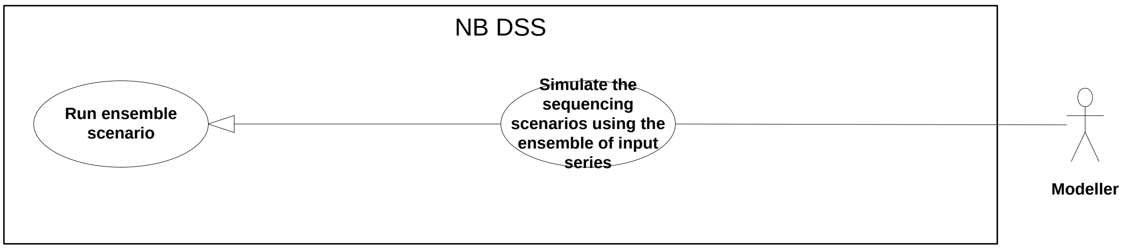

Figure 3.54 Use case 03 – Simulate the sequencing scenarios using the ensemble of input series (UML)

Note in Figure 3.54 that "Simulate the sequencing scenarios .." use case maps directly to the "Run ensemble scenario" generalised use case.

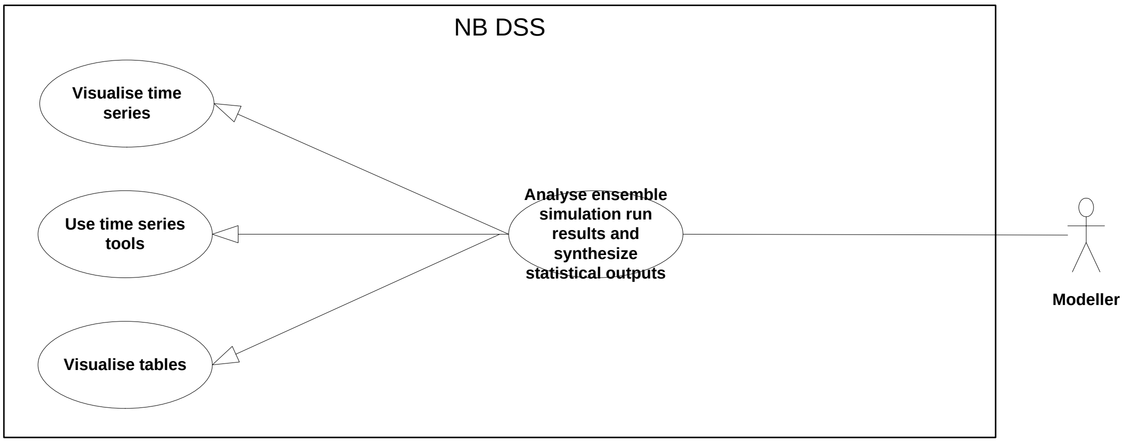

Figure 3.55 Use case 03 – Analyse ensemble simulation run results and synthesize statistical outputs (UML)

Note in Figure 3.55 that the analysis of ensemble situation can be generalised to time series processing and visualisation. The statistical output is synthesized in tables.

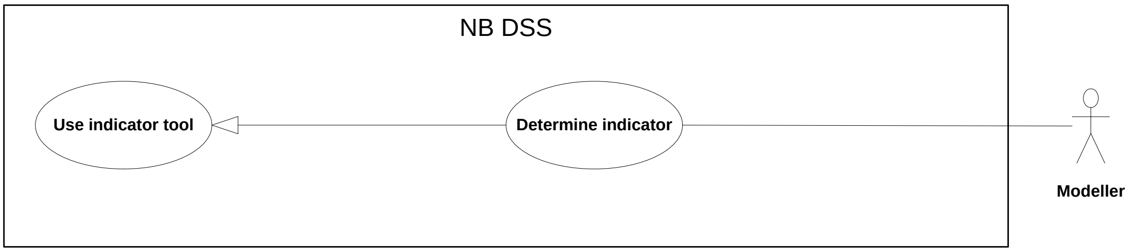

Figure 3.56 Use case 03 – Determine indicator (UML)

Note in Figure 3.56 that the use case "Determine Indicator" maps directly to the use case "Define indicator", which is performed outside the system (see description of that use case in Table 3.2). Once indicators are determined, they must be configured in the NB DSS (use case: "Use Indicator Tool").

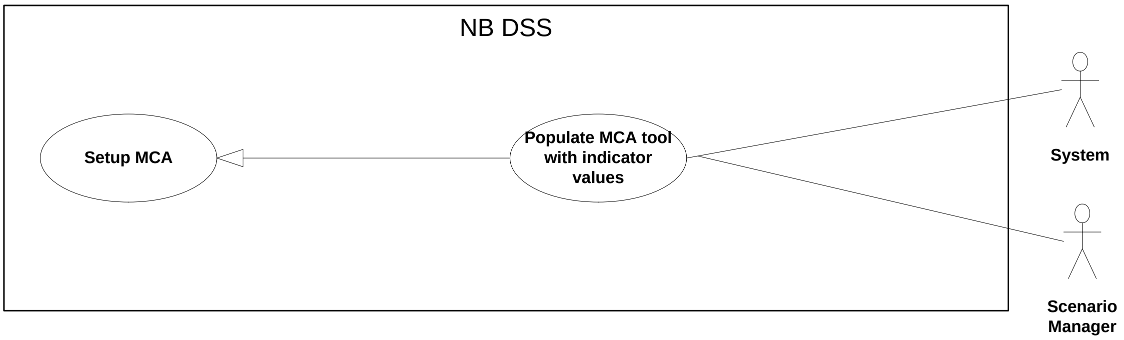

Figure 3.57 Use case 03 – Populate MCA tool with indicator values (UML)

Note in Figure 3.57 that only the indicators are defined during this phase of the MCA setup. Stakeholder and their preferences will be introduced in the next phase (Figure 3.58).

The actor System is in this use case a proxy for the actor Modeller, who is the one actually initiating the use case.

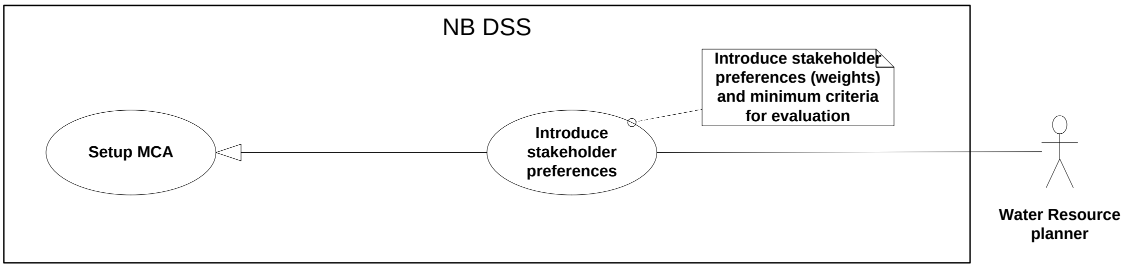

Figure 3.58 Use case 03 – Introduce stakeholder preferences (UML)

Note in Figure 3.58 that stakeholder preferences are considered in the MCA through the definition of

- Criteria important to the stakeholders; these criteria are defined and configured in the "Setup MCA" use case.
- Weights that individual stakeholders (or stakeholder groups) put on the configured criteria. These weights are set and varied in the "Run MCA" use case.

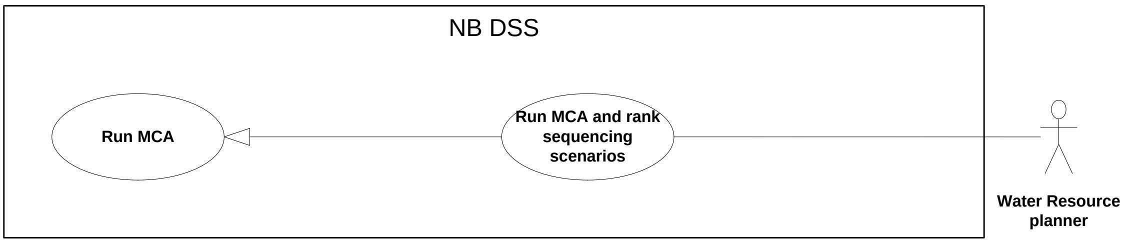

Figure 3.59 Use case 03 – Run MCA and rank sequencing scenarios (UML)

Note in Figure 3.59 that the individual criteria values in the decision matrix (or the reference point graph) are computed in the "Run MCA" scenario and subsequently used to establish a rank for each scenario.

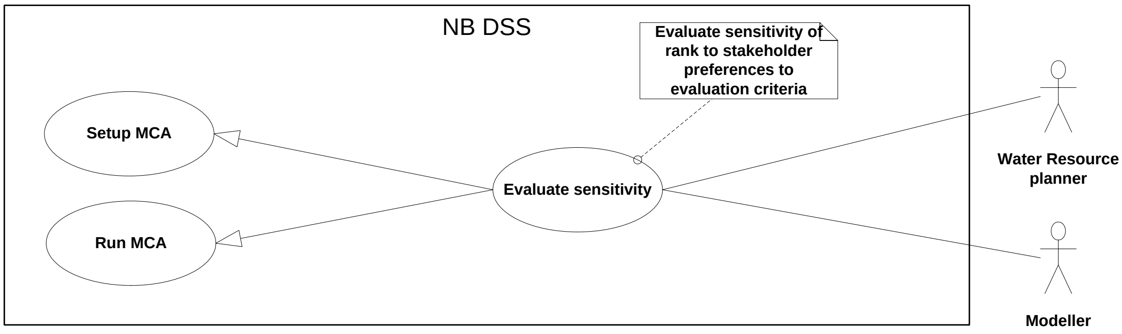

Figure 3.60 Use case 03 – Evaluate sensitivity (UML)

Note in Figure 3.60 that the MCA rank established for each scenario is strongly dependent on the MCA criteria and the stakeholder weights utilized. Including or excluding criteria ("Setup MCA") and varying stakeholder weights ("Run MCA") are used to evaluate the sensitivity of the MCA.

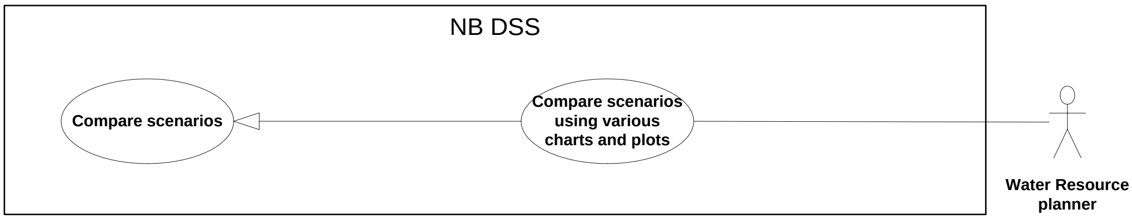

Figure 3.61 Use case 03 – Compare scenarios using various charts and plots (UML)

Note in Figure 3.61 that the specific use case here maps directly to the generalised version.

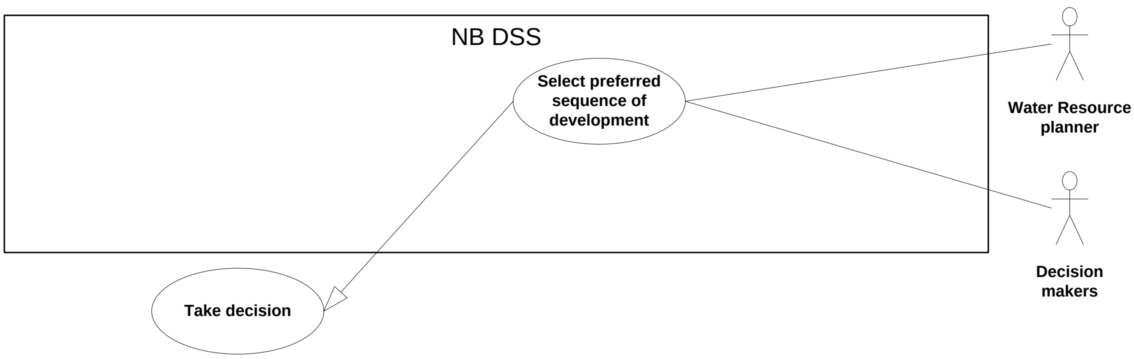

Figure 3.62 Use case 03 – Select preferred sequence of development (UML)

Note in Figure 3.62 that actual decision making occurs outside of the NB DSS system.

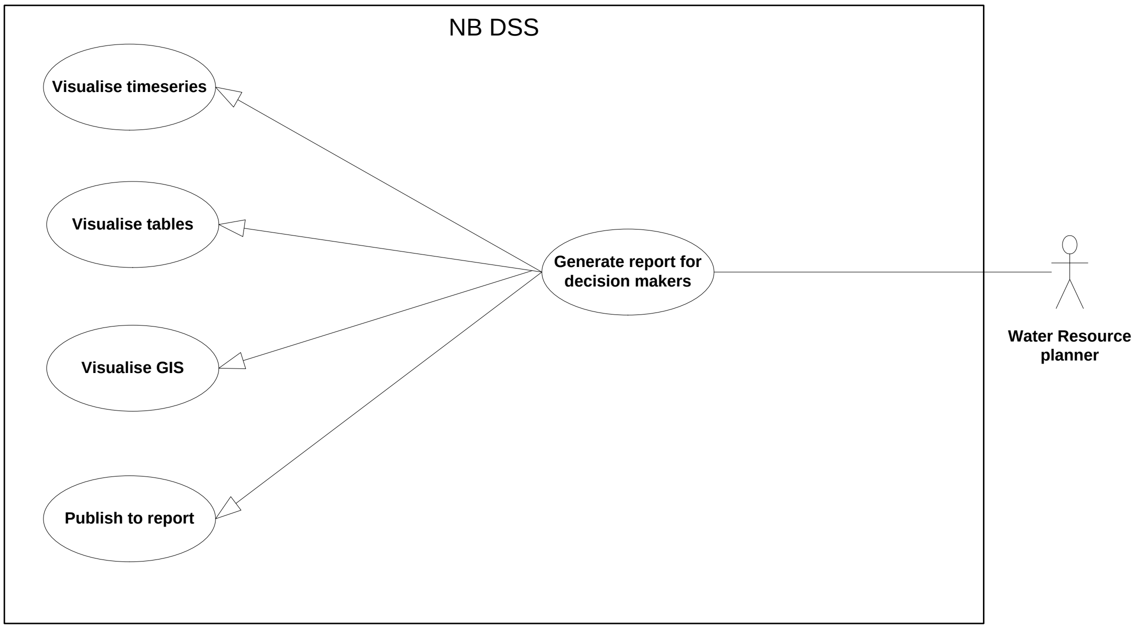

Figure 3.63 Use case 03 – Generate report for decision makers (UML)

Note in Figure 3.63 that reporting – as shown in many of the use cases – involves visualisation of time series, tables and GIS layers and thereafter publishing to reports.

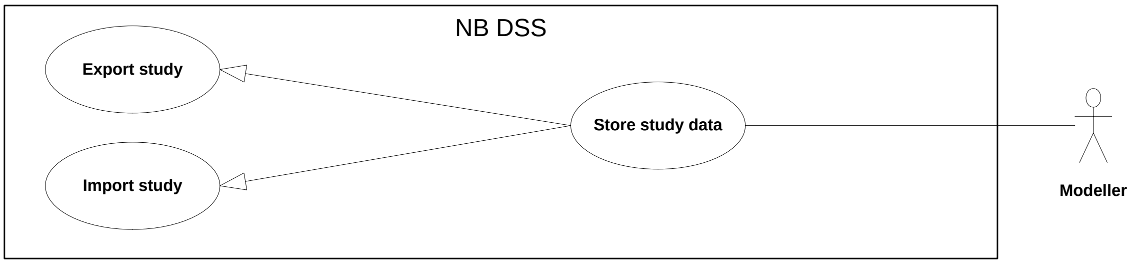

Figure 3.64 Use case 03 – Store study data (UML)

Note in Figure 3.64 that Studies are intended to be shared with other NB DSS installations. This is defined in the "Export study" and "Import study" use cases.

3.1.4 NB-DSS UC-04: Select best investment option for NEL region

***Brief description**: The Nile equatorial region is facing problems of shortage of hydropower, food security and transportation. The region would like to exploit the hydropower potential within the region, use the available lakes as source of water for irrigation, and also as means of transport. This use case focuses on selection of best investment option to address that power and food shortages in the region..*

***Business trigger**: The available water may not be enough to cater all planned activities at once. Therefore there was a need to see different scenarios in which the effect of planned development activities on the existing water uses.*

Preconditions:
a) Agreed scenarios to be investigated (with the availability of the relevant data)
- *Base Case – existing situation*
- *Alternative 1: one multiple reservoir on R. Kagera*
- *Alternative 2: Two medium dams on R. Nzoia*
- *Alternative 3: Water abstraction from lake Victoria for irrigation*
- *Alternative 4: Four new hydropower dams (Lake Victoria- lake Albert stretch)*

b) Agreed set of analysis:
- *Planned hydropower development on existing water fall.*
- *Irrigation water abstraction from Lake Victoria on the water level*
- *Multipurpose reservoirs on flood protection and fishery.*

c) Spatial, hydro-meteorological and system data in the Lake Victoria area exist.

Figure 3.65 Brief description of the use case quoted from /3/

The use case UC-04 is described in /3/ in the form of a table, which is translated into the overview use case diagram shown in Figure 3.66.

Figure 3.67 to Figure 3.71 show how the use case has been used to identify generalised use cases.

This analysis has identified the use cases listed in Table 3.4. Note that some of the generalised use cases have been identified and described in previous sections; they are included here for reference.

Table 3.4 Use case 04 - Identified generalised use cases

Generalised use cases	Description
Assess data availability	This use case is in itself not internal to the NB DSS system, but is likely to make use of several others tools, as it is the process of identifying available data – in the NB DSS as well as outside – with respect to the task(s) at hand. The user may peruse the NB DSS but may also supply additional data and perhaps import it into the NB DSS system as part of this exercise.
Calibrate MIKE 11	Identified and described in Table 3.3
Calibrate MIKE BASIN	Identified and described in Table 3.1

Clone and modify model and scenario	Identified and described in Table 3.3
Create linked model	Identified and described in Table 3.2
Create optimization scenario	The Optimizer is used for single or multi objective optimization often related to optimization of reservoir operations, conjunctive use of surface- and groundwater or for model calibration (automatic calibration) but may also be used for automatic calibration purposes. Creating an optimization scenario in the NB DSS involves that certain inputs for the optimizer is specified including: • Definition of objective functions to be evaluated and optimized • Identification of variables and parameters to be changed during optimization associated with acceptable limits (variables may include rule curves etc.) • Maximum number of simulations and/or stop criteria (the latter typically used for calibration).
Create scenario	Identified and described in Table 3.1 Table 3.2
Define indicator	Identified and described in Table 3.2
Define optimization scenario	The process occurs outside of the NB DSS as a planning routine to define the content of a study with respect to doing optimisation using a model setup, i.e. defining on paper what is required with respect to data, model(s), criteria etc.
Define scenario	Identified and described in Table 3.3
Edit model	A user edits a model (setup) in the designated Model Tool, e.g. MIKE BASIN.
Edit time series	Identified and described in Table 3.3
Export study	Identified and described in Table 3.3
Import Study	Identified and described in Table 3.3
Import time series	Identified and described in Table 3.1
Modify DSS scenario	Modifying the specifications of a scenario in the NB DSS. This can also involve modifying the underlying model setup in the

	Model Tool.
Publish in report	Identified and described in Table 3.1
Register MIKE 11 model	Identified and described in Table 3.2
Register MIKE BASIN model	Identified and described in Table 3.2
Register model	Identified and described in Table 3.1
Run CBA	Identified and described in Table 3.2
Run MCA	Identified and described in Table 3.2
Run optimization scenario	The process in the NB DSS of executing a scenario that includes optimization specifications.
Setup CBA	Identified and described in Table 3.2
Setup MCA	Identified and described in Table 3.3
Setup MIKE 11 model	Identified and described in Table 3.2
Setup MIKE BASIN	Identified and described in Table 3.1
Make decision	Identified and described in Table 3.1
Use ensemble generator tool	Identified and described in Table 3.3
Use time series tool	Identified and described in Table 3.1
Visualise tables	Identified and described in Table 3.1 Table 3.2
Visualise time series	Identified and described in Table 3.1 Table 3.2

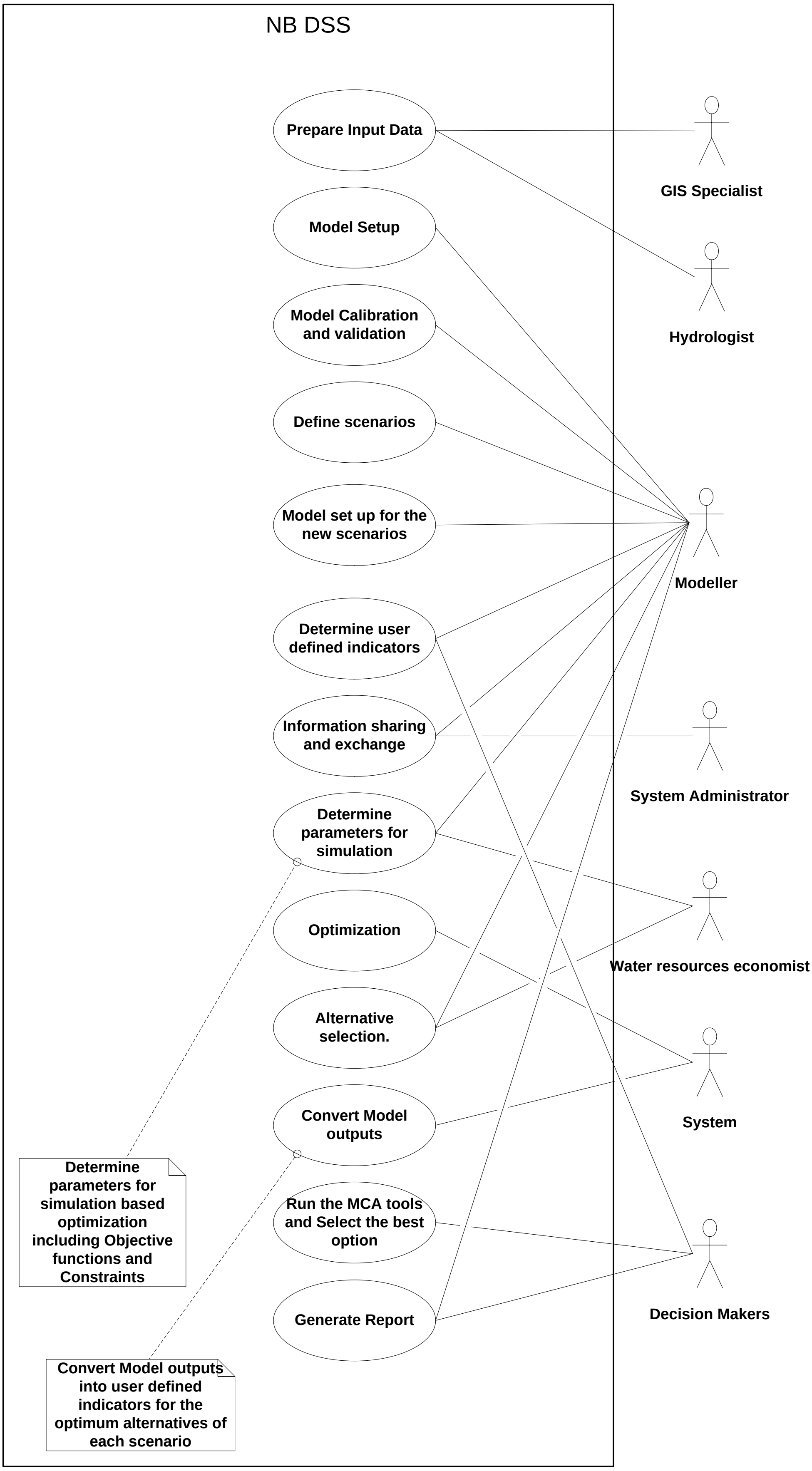

Figure 3.66 Use case 04 – Overview (UML)

Note in Figure 3.66 that

- The use case steps illustrates a standard type of modelling exercise: preparing input, defining the model setup to investigate, specify scenarios, run and compare scenarios, and conduct an MCA analysis before generating a report. The “information sharing and exchange” step illustrates the need to transfer study data between NB DSS sites.

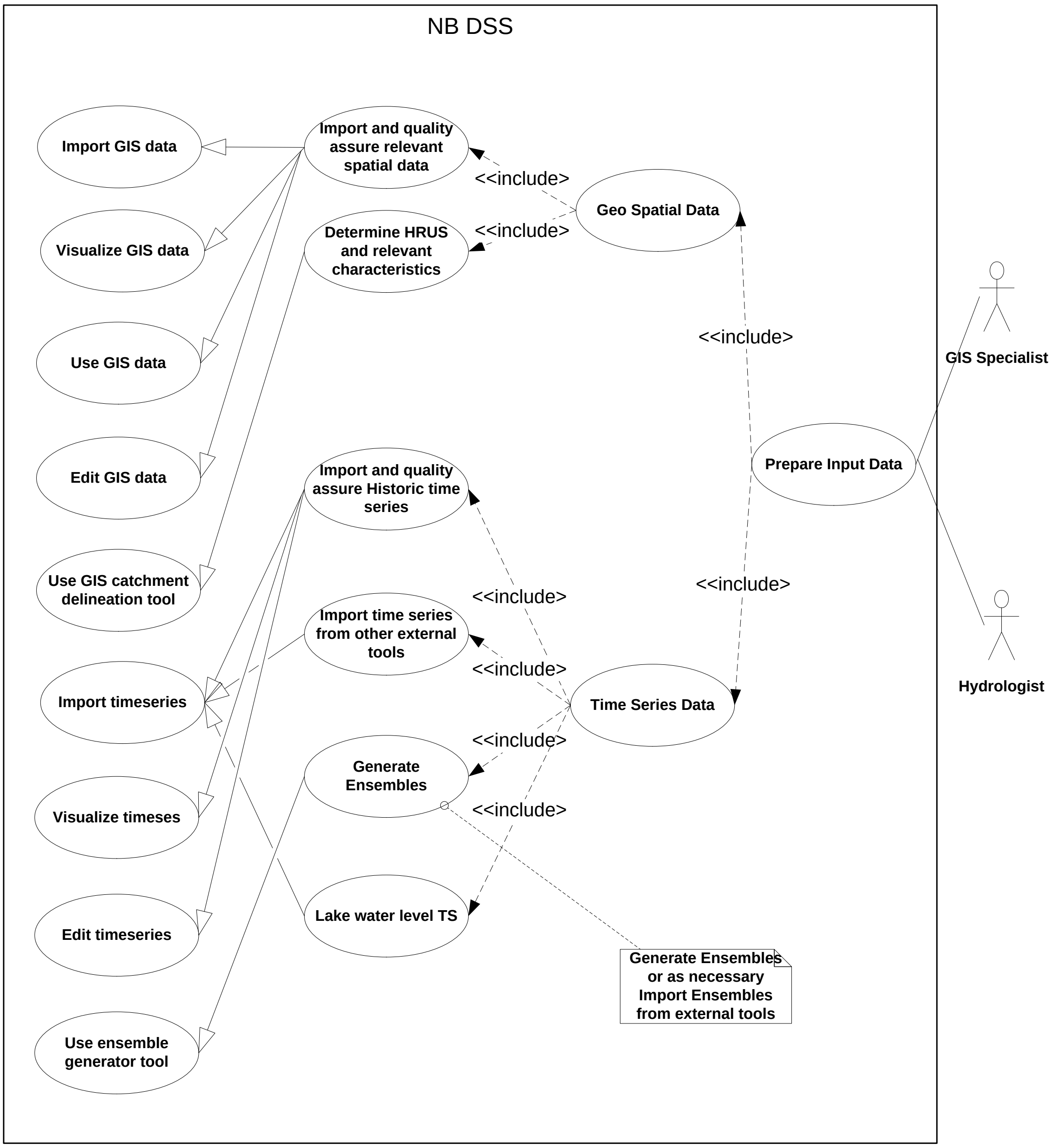

Figure 3.67 Use case 04 – Prepare Input data (UML)

Note in Figure 3.67 that

- Preparing of input data revolves around time series and GIS data. A similar exercise is conducted in the first steps of UC02

- A special type of time series, ensembles, is introduced by the activity “Generate ensembles”. The user may use a tool for establishing these sets of time series or may import predefined time series as ensembles.

- An important GIS activity - catchment delineation - is introduced. Conceptually this is not different from other GIS activities normally carried out using tools, but it will require significant programming to deploy this functionality; hence the special focus.

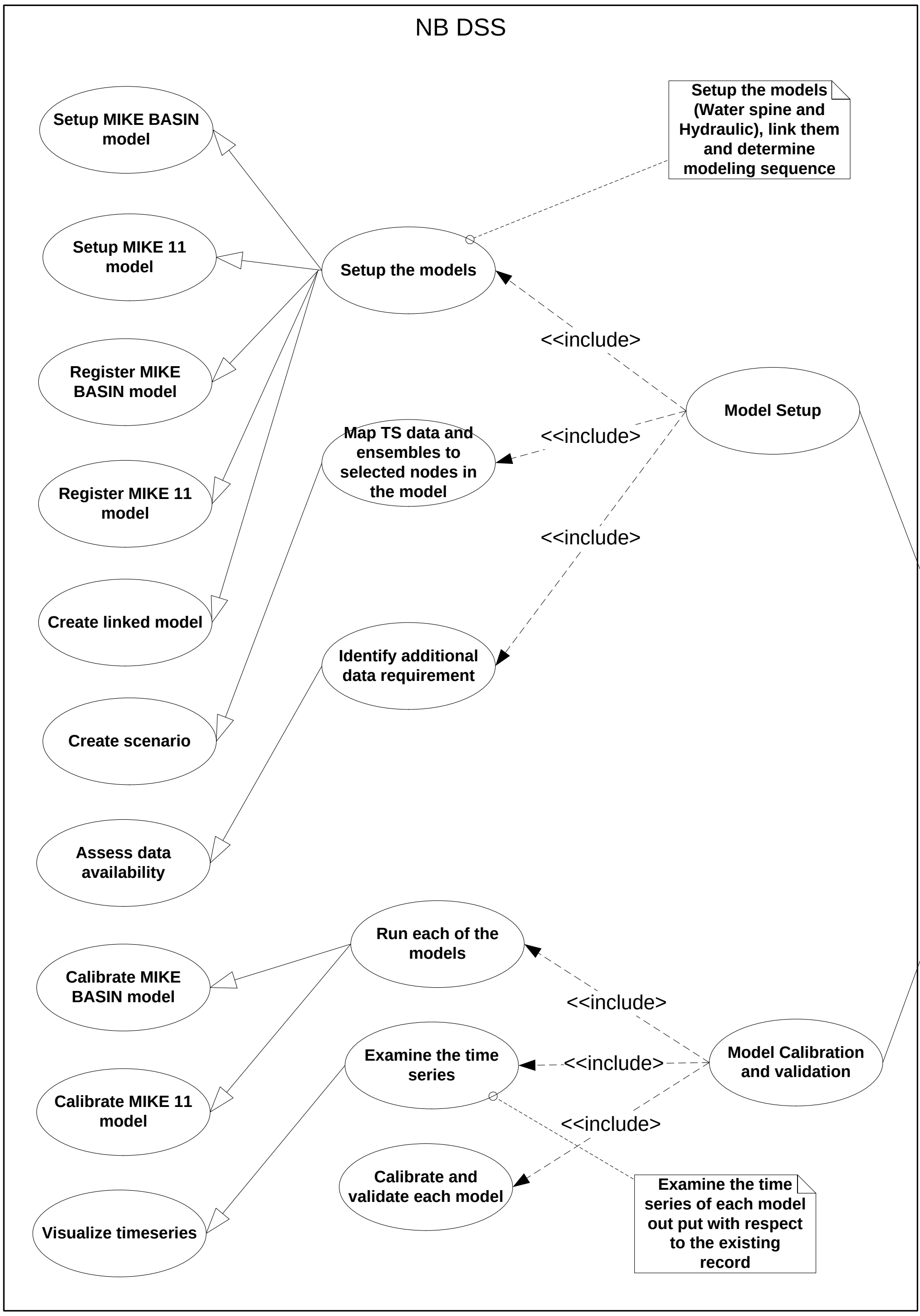

Figure 3.68 Use case 04 – Model Setup and Calibration (UML)

Note in Figure 3.68 that

- The uses cases illustrate what are normal activities when performing modelling – irrespective of the model tool. For the NB DSS the Water spine and hydraulic models are translated into models of type MIKE BASIN and MIKE 11.

- Calibration is an activity by the Modeller involving execution of the models and visualization of time series. These activities can be performed entirely within the model tools – or in conjunction with the NB DSS system. Parameter changes to a model setup are carried out in the relevant Model Tool.

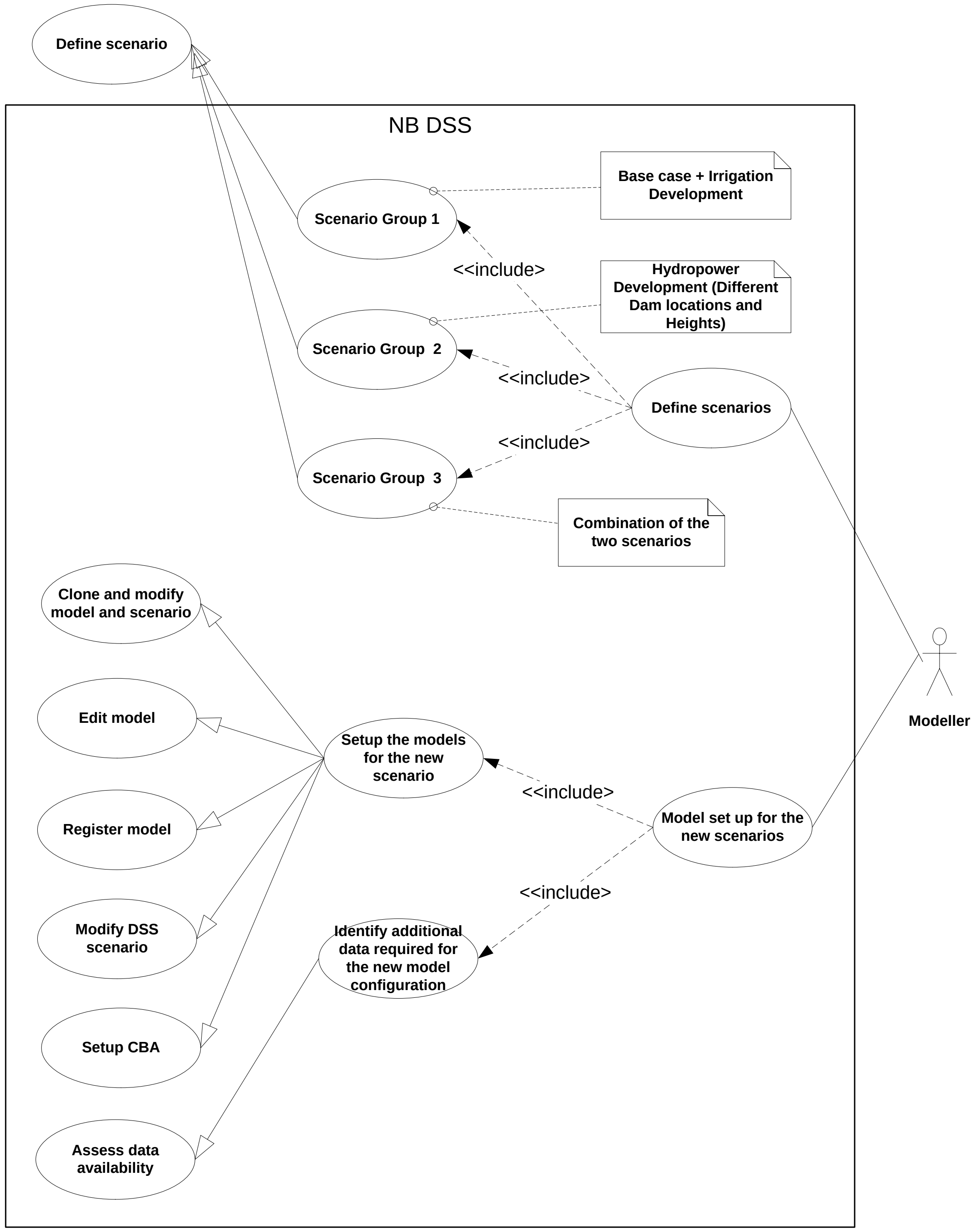

Figure 3.69 Use case 04 – Scenarios (UML)

Note in Figure 3.69 that

- “Define scenario” in the context of the generalised use case is an activity outside the NB DSS system, using data from the system. A distinction between the scenario as seen by the user and scenario as seen by the system exists and this activity relates to the scenarios outside the NB DSS system (but inside the head of the user).

- The above also implies a missing step in the use case where by the scenario(s) as defined by the system are added on top of the models setup(s). This is, however, included in the "Model set up for the new scenarios" as the general use cases include "Modify DSS Scenario" which is about editing the definitions in the NB DSS defining the scenario.

- The use case "Asses data availability" could also be drawn outside the NB DSS system boundary as it is an activity by the user alone and not in itself requiring the system. However, the user will perform other generalised use cases (internal to the NB DSS) to obtain the knowledge for the assessment.

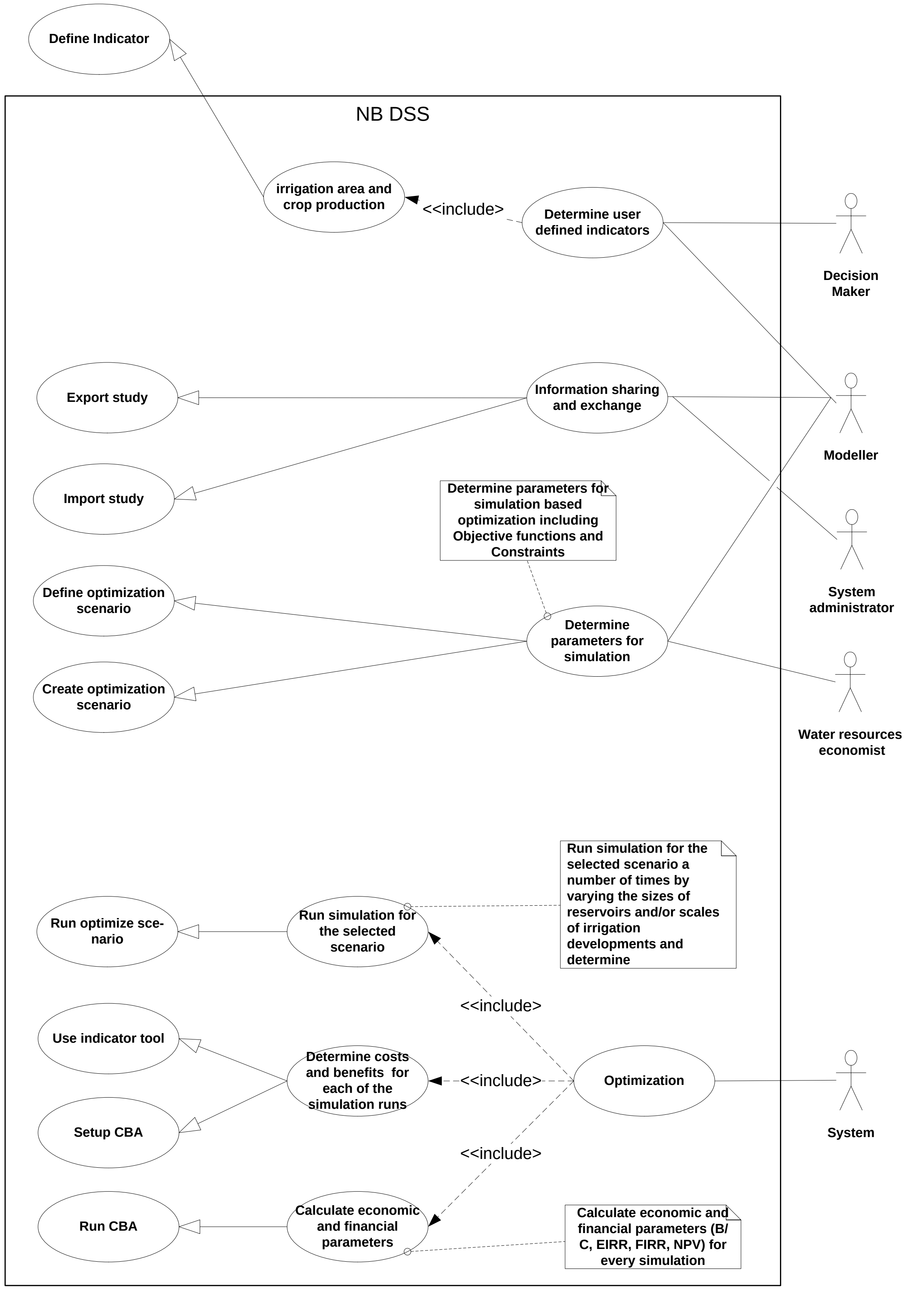

Figure 3.70 Use case 04 – Indicator definition and Optimisation (UML)

Note in Figure 3.70 that

- The use case “Determine parameters for simulation” could be an activity outside the NB DSS, as could (at least) the first of the two derived generalised use cases “Define optimisation scenario”, as this could be a task on paper. The “Create optimisation scenario”, however, involves system activities to establish the links between data, models, scenario and indicators.

- The use case “Optimization” is very broad as it includes running the simulations, performing post-processing, performing performance analysis and changing model components and/or parameters.

- The links between indicators and analysis tool and between indicators and data are described as if the indicators have a life of their own while being used by analysis tools like CBA and MCA. This is indicated by the use case “use indicator tool”, which is being applied irrespective of the final use of the indicator.

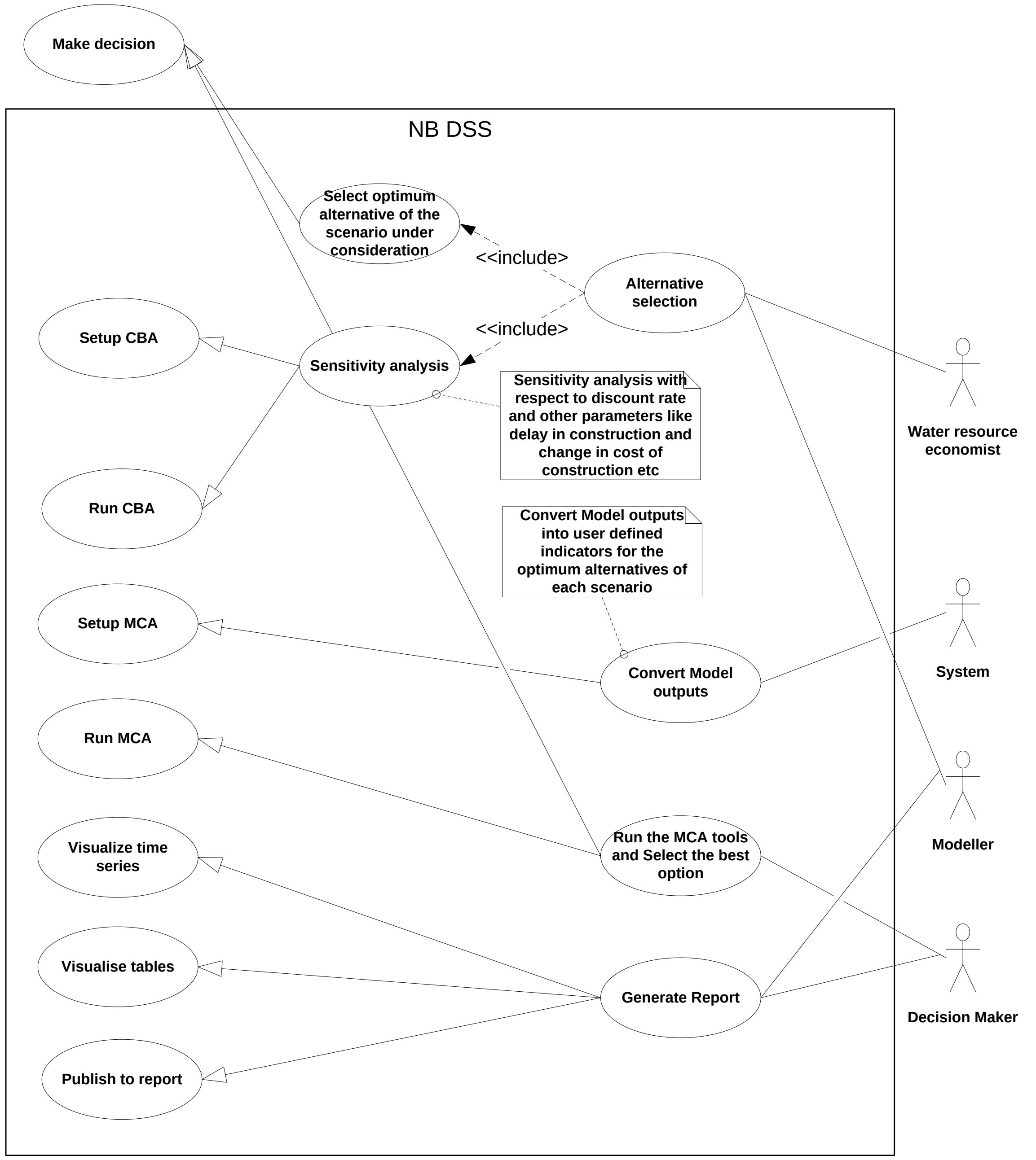

Figure 3.71 Use case 04 – Analysis and report (UML)

Note in Figure 3.71 that

- Doing analysis and making decisions is an iterative process where the user "plays" with different variables in the analysis and indicator calculations. This is why "Setup .." and "Run .." use cases exist for both CBA and MCA analyses.

- The "Publish to Report" use case reflects the ability of the system to export/generate content (e.g., graphs and tables).

- Visualisation of time series and tables (and other data as well) will enable the user to copy/paste information to other media.

3.1.5 NB-DSS UC-05: Integrate a New Model Tool in DSS

One of the requirements of the Nile Basin DSS is that is shall be flexible enough to enable advanced users integrate model tools to solve diverse types of problems in the operational application of the DSS. This use case focuses on integrating a Groundwater model tool into the DSS.

Figure 3.72 Brief description of the use case quoted from /3/

The use case UC-05 is described in /3/ in the form of a table. This table was translated into the overview use case diagram shown in Figure 3.73

This analysis has identified the use cases listed in Table 3.5. Note that some of the generalised use cases have been identified and described in previous sections; they are included here for reference.

Table 3.5 Use case 05 - Identified generalised use cases

Generalised use cases	Description
Analyse model tool inter-communication options	Establish understanding of the model tool's inter-communication possibilities Activities involved comprise establishing understanding of the model tool and inspecting the model tool source code
Analysis of IModelAdapter	Establish understanding of the DSS Front-end model adapter interfaces Activities involved comprise reading the NB DSS documentation
Change model tool	Modifying the model tool in order to support model tool inter-communication Activities involved comprise changing the source code of the model tool, testing and debugging it
Code model tool adapter	Coding the model tool adapter Activities involved comprise writing the source code for the model tool adapter, testing and debugging it

Generalised use cases	Description
Design model adapter tool	Designing the model tools adapter Activities involved include defining the hydro objects that the model tool adapter shall support, defining the optimization parameters shallot be supported, defining whether is shall support ensembles as input and defining how to modify a model setup
Establish understanding of model tool	Establish understanding of the model tool Activities involved include analysing the model tool's data model, analyse the format of input data, analyse how output is created and formatted and analyse what type(s) of hydro object can be used by the model tool
Test create scenario	Test the model tool adapter with respect to scenario creation Activities involved include deployment of the developed model adapter and creating a new scenario to verify that all parameters are correctly communicated from the adapter to the Scenarios functional component.
Test model inter-communication	Test the model tool with respect to model inter-communication Activities involved include testing that data are correctly communicated between the model tools.
Test model linkage	Test the model tool adapter with respect to model linkage Activities involved include deployment of the developed model adapter, linking scenarios and testing that data is parsed correctly by the NB DSS from the first model execution to the next.
Test model registration	Test the model tool adapter with respect to model registration Activities involved include deployment of the developed model adapter, registration of a model setup and verifying that the adapter has parsed the correct information to the NB DSS system.

Generalised use cases	Description
Test run scenario	Test the model tool adapter with respect to running a scenario Activities involved include deployment of the developed model adapter, running a scenario and verifying that the correct in-formation is parsed from the NB DSS to the model tool.

Figure 3.73 provides an overview of the use case and Figure 3.74 and Figure 3.75 show how the use case has been used to identify generalised use cases.

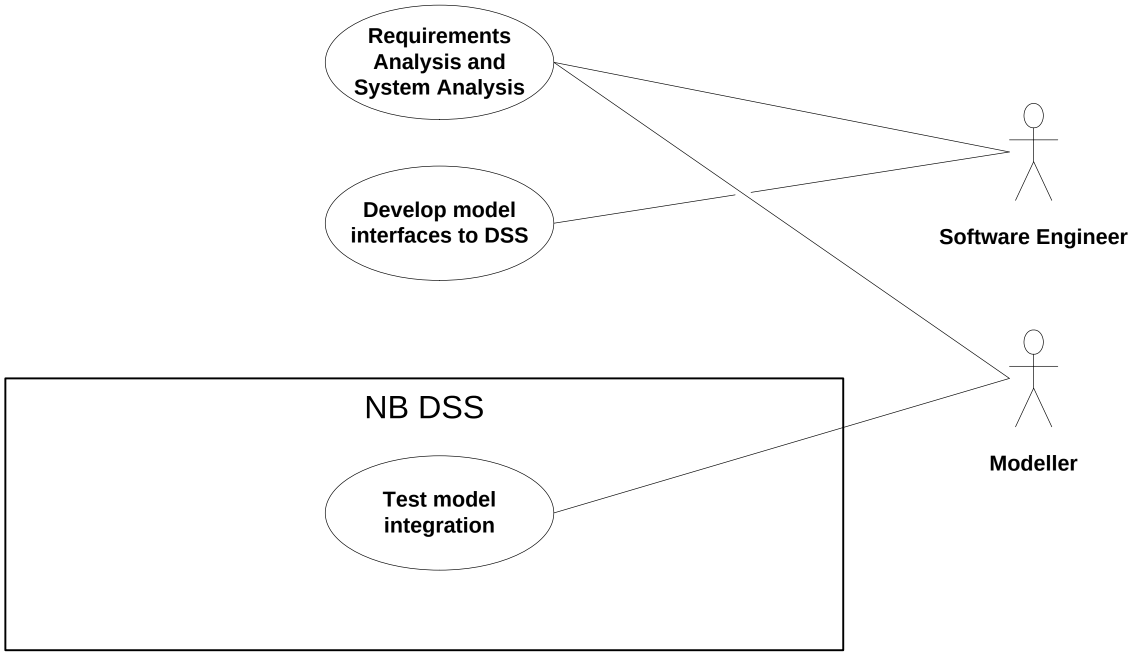

Figure 3.73 Use case 05 – Overview (UML)

Note in Figure 3.73 that

- The use case includes 3 sub-use cases – analysis, development and testing – and 2 actors - a software engineer and a modeller.
- The analysis and development takes place outside the NB DSS system

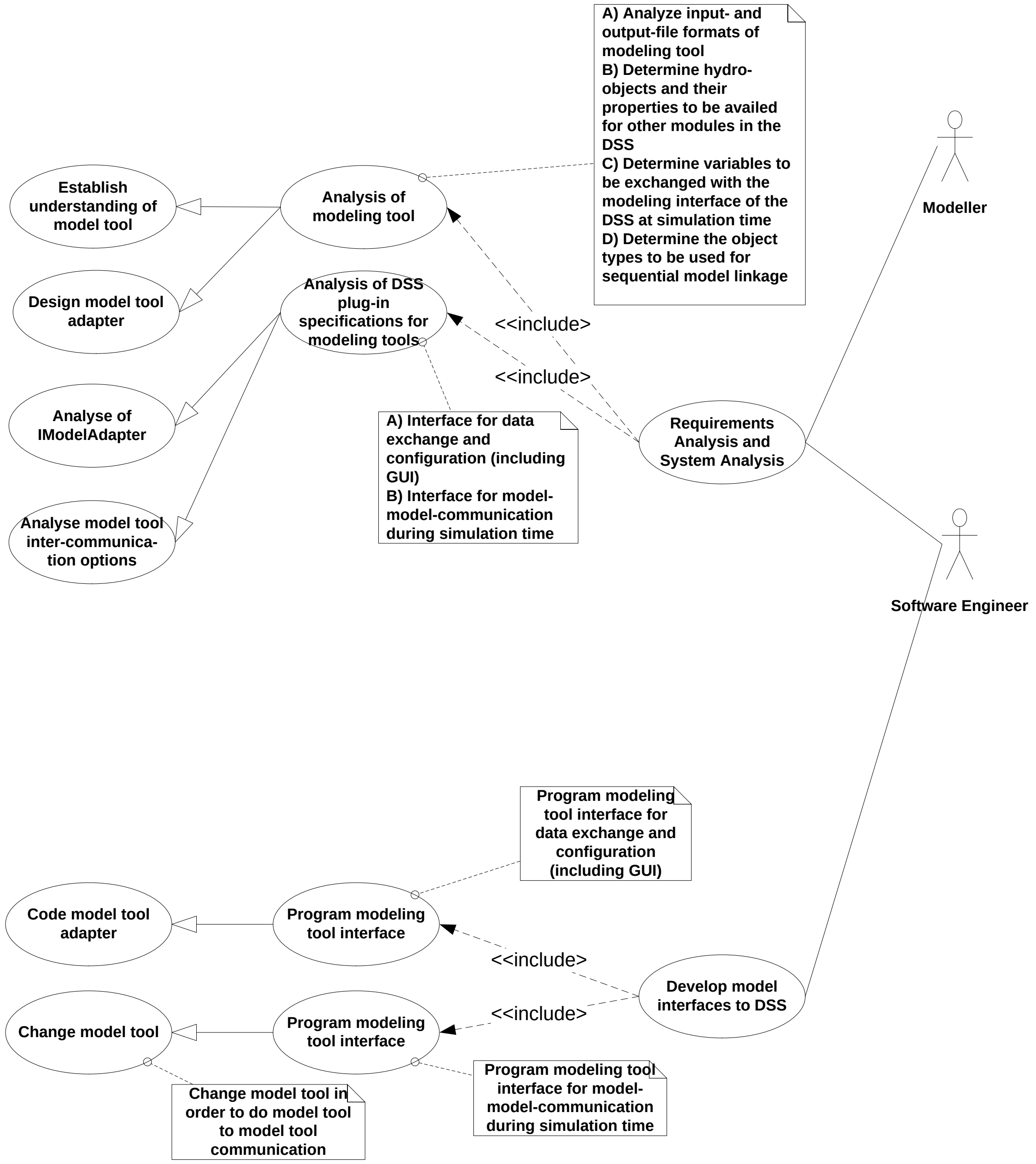

Figure 3.74 Use case 05 - "Requirements analysis and system analysis" and "develop model interfaces to DSS" (UML)

Note in Figure 3.74 that

- Analysis of model tool focuses on its input and output. The central processing part is not relevant with respect to designing and coding the adapter.
- The Software Engineer must understand the interfaces that have to be implemented in order to communicate with the DSS Front-end.
- Model to model inter-communication can only be enabled if each model supports external communication (e.g., providing and accepting model parameters and state variables).

- All the activities takes place outside the NB DSS.

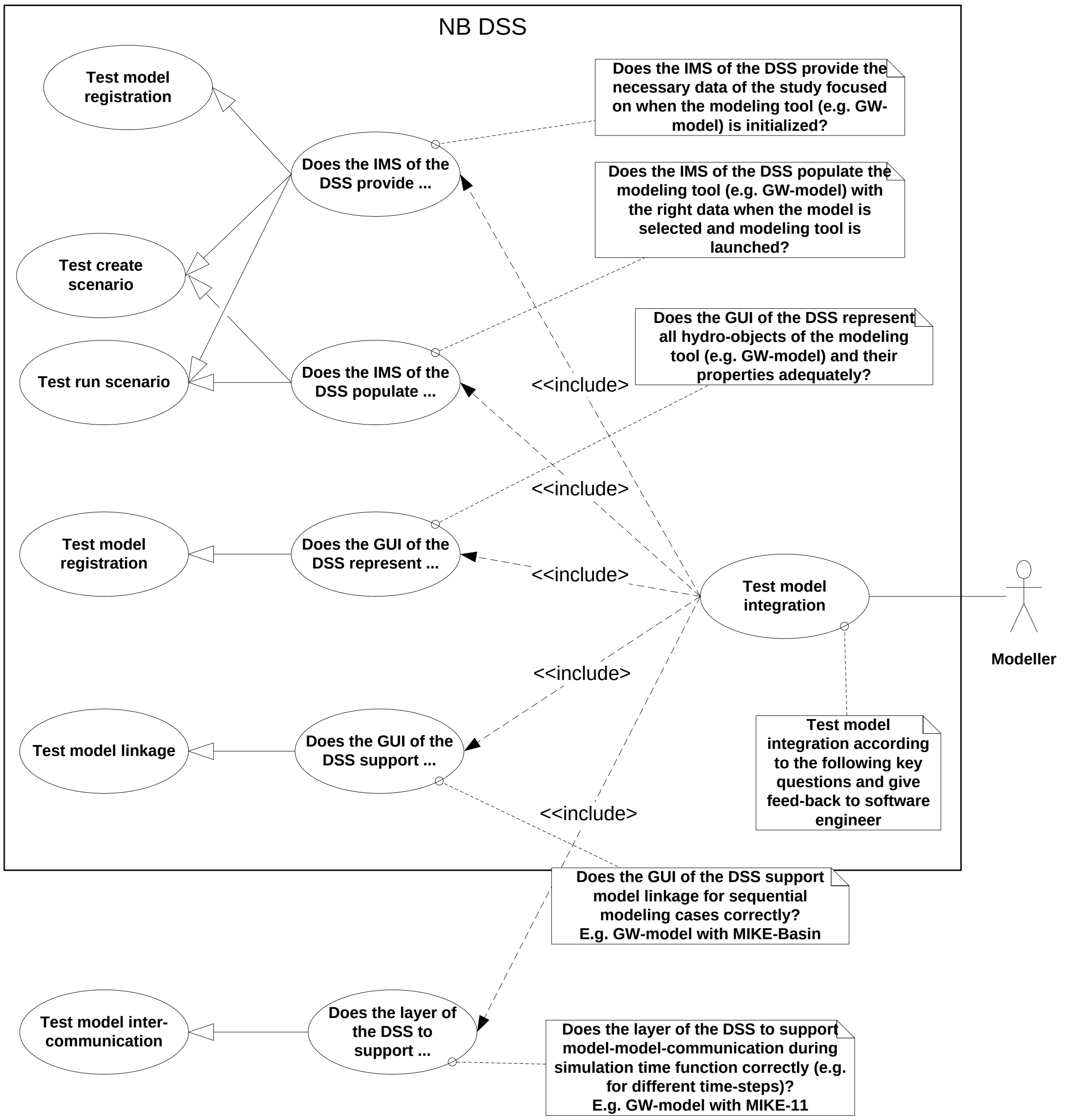

Figure 3.75 Use case 05 – Test model integration (UML)

Note in Figure 3.75 that testing focuses on verifying proper processes involving the adapter, i.e. registration of models, creation of scenarios and execution of scenarios. Also note that model inter-communication is a model tool to model tool concern and testing this communication typically occurs outside the NB DSS system.

3.1.6 Summary

This section maps use cases with requirements in table Table 3.6.

Table 3.6 Use case and requirement mapping

Functional component	Use case	Requirement
CBA	Run CBA	2.1.5.11
	Setup CBA	2.1.5.11
Ensembles	Run ensemble scenario	
	Setup ensemble scenario	
	Use ensemble generator tool	2.1.6.6 (1) 2.1.6.6 (2)
GIS	Edit GIS data	
	Export GIS data	
	Import GIS data	2.1.4.1 2.1.4.2 (1)
	Inspect GIS layers	
	Link timeseries to feature	
	Use GIS catchment delineation tool	2.1.4.3 (3)
	Use GIS tool	2.1.4.2 (2) 2.1.4.3 (1) 2.1.4.3 (2) 2.1.4.4 2.1.4.4 (1a) 2.1.4.4 (1b) 2.1.4.4 (1c) 2.1.4.4 (2a) 2.1.4.4 (2b) 2.1.4.4 (2c) 2.1.4.4 (2d) 2.1.3.3(2a) 2.1.3.3(2e)
	Visualize GIS data	2.1.4.3 (1) 2.1.4.3 (2)
Hydro objects	Create hydroobject	
	Edit hydroobject	
Indicators	Define indicator	
	Use indicator tool	2.1.5.12 2.1.8.3
Linked model	Create linked model	2.1.5.10
MCA	Run MCA	2.1.5.12 2.1.8.1 2.1.8.2 2.1.8.3
	Setup MCA	2.1.8.1 2.1.8.3
Optimization scenarios	Create optimization scenario	2.1.6.4
	Define optimization scenario	
	Run optimization scenario	
Scenarios	Clone and modify MIKE BASIN model and scenario	
	Clone and modify model and scenario	
	Compare scenarios	2.1.6.3

Functional component	Use case	Requirement
	Create scenario	2.1.6.1
	Modify DSS scenario	
	Register MIKE 11 model	2.1.6.1
	Register MIKE BASIN model	2.1.6.1
	Register model	2.1.6.1
	Run ensemble scenario	
	Run scenario	
	Setup ensemble scenario	
	Test create scenario	
	Test model registration	
	Test run scenario	
	Use rainfall runoff model	
Studies	Create study	
	Export study	
	Import Study	
	Manage study	
Tables	Assess data availability	
	Import tables	
	Use table tool	
	Visualise tables	
Timeseries	Assess data availability	
	Create time series	
	Edit timeseries	
	Export time series data	
	Filter timeseries	
	Import timeseries	
	Inspect timeseries	2.1.3.3 (1a) 2.1.3.3 (1b)
	Use demand calculator tool	
	Use soil erosion tool	

Functional component	Use case	Requirement
	Use timeseries tool	2.1.3.1 2.1.3.1 (1) 2.1.3.1 (1a) 2.1.3.1 (1b) 2.1.3.1 (1c) 2.1.3.1 (1d) 2.1.3.1 (1e) 2.1.3.1 (1f) 2.1.3.1 (1g) 2.1.3.1 (1h) 2.1.3.1 (1i) 2.1.3.1 (1j) 2.1.3.1 (2a) 2.1.3.1 (2b) 2.1.3.1 (2c) 2.1.3.1 (3) 2.1.3.1 (4) 2.1.3.1 (4a) 2.1.3.1 (4b) 2.1.3.2 2.1.3.2 (2) 2.1.3.2 (3) 2.1.3.2 (4) 2.1.3.2 (5) 2.1.3.3 (1b) 2.1.3.3 (1c) 2.1.3.3 (1e) 2.1.3.3 (2b.3) 2.1.3.3 (2c) 2.1.3.3 (2c.1) 2.1.3.3 (2c.2) 2.1.3.3 (2c.3) 2.1.3.3 (2d.3)
	Visualise timeseries	2.1.3(1) 2.1.3(2) 2.1.3(3) 2.1.3(4) 2.1.3(5) 2.1.3(6) 2.1.3.3 (1b) 2.1.3.3 (1c) 2.1.3.3 (1d) 2.1.3.3 (1f) 2.1.3.3 (2b) 2.1.3.3 (2b.2) 2.1.3.3 (2b.3) 2.1.3.3 (2c.1) 2.1.3.3 (2c.2) 2.1.3.3 (2d) 2.1.3.3 (2d.1) 2.1.3.3 (2d.2) 2.1.3.3 (2d.3)
Reporting	Publish in report	

Functional component	Use case	Requirement
Model tools	Calibrate MIKE 11	
	Calibrate MIKE BASIN	
	Calibrate model	2.1.6.2 2.1.6.2 (1) 2.1.6.2 (2) 2.1.6.2 (4)
	Clone and modify MIKE BASIN model and scenario	
	Edit model	
	Setup MIKE 11 model	2.1.5 (1) 2.1.5 (3.3) 2.1.5 (3.5) 2.1.5 (4) 2.1.5.1 2.1.5.2 2.1.5.2 (1) 2.1.5.2 2.1.5.2 (1) 2.1.5.2 (3) 2.1.5.2 (4) 2.1.5.2 (5) 2.1.5.2 (7) 2.1.5.3 2.1.5.4 2.1.5.5 2.1.5.8 2.1.5.9 2.1.5.13 2.1.5.14 2.1.5.15 2.1.5.16 2.1.5.17 2.1.5.18 2.1.7.1 2.1.7.2 2.1.7.3 2.1.7.5

Functional component	Use case	Requirement
	Setup MIKE BASIN model	2.1.5 2.1.5(1) 2.1.5(2) 2.1.5(3) 2.1.5(3.1) 2.5.5(3.2) 2.1.5(3.3) 2.1.5(3.4) 2.1.5(3.5) 2.1.5(3.6) 2.1.5(3.7) 2.1.5(3.8) 2.1.5(4) 2.1.5.1 2.1.5.2 2.1.5.2(1) 2.1.5.2(2) 2.1.5.2(3) 2.1.5.2(4) 2.1.5.2(5) 2.1.5.2(6) 2.1.5.2(7) 2.1.5.3 2.1.5.4 2.1.5.5 2.1.5.7 2.1.5.8 2.1.5.9 2.1.5.13 2.1.5.14 2.1.5.15 2.1.5.17 2.1.5.18 2.1.7.1 2.1.7.2 2.1.7.3 2.1.7.5 2.1.7.6 2.1.7.8 2.1.6.2 2.1.6.2(1) 2.1.6.2(2) 2.1.6.2(3) 2.1.6.2(4)

Functional component	Use case	Requirement
	Setup MIKE BASIN	2.1.5 (1); 2.1.5 (2); 2.1.5 (3.1) 2.1.5 (3.2) 2.1.5 (3.3) 2.1.5 (3.4) 2.1.5 (3.5) 2.1.5 (3.6) 2.1.5 (3.7) 2.1.5 (3.8) 2.1.5 (4) 2.1.5.1 2.1.5.2 2.1.5.2 (1) 2.1.5.2 (2) 2.1.5.2 (3) 2.1.5.2 (4) 2.1.5.2 (5) 2.1.5.2 (6) 2.1.5.2 (7) 2.1.5.3 2.1.5.4 2.1.5.5 2.1.5.7 2.1.5.8 2.1.5.9 2.1.5.13 2.1.5.14 2.1.5.15 2.1.5.17 2.1.5.18 2.1.7.1 2.1.7.2 2.1.7.3 2.1.7.5 2.1.7.6 2.1.6.2 (3)
	Setup semi-distributed rainfall-runoff model (MIKE SHE Studio)	2.1.5.13 2.1.5.7 2.1.5.6 2.1.7.8
	Test model registration	
	Use rainfall runoff model	
Meta-data	Assess data availability	
	Inspect GIS layers	
	Inspect timeseries	

Please note from Table 3.6 that

- 47 out of the 72 generalised use cases are not backed by requirements from the ToR /2/. Requirements for those shall be established during the detailed analysis and design stages.

- 6 generalised use cases are backed by more than 10 requirements
- 8 generalised use cases are backed by only 1 requirement. Also these use cases need to be further analysed during the detailed analysis and design stage.

3.2 Functional Components

The generalised use cases defined in Table 3.1 to Table 3.5 are used to establish functional components. Requirements, meetings and initial design considerations have given rise to the establishment of additional functional components. All the functional components are shown in Table 3.7. Note, Chapter 1 describes the functional components in more details

Table 3.7 Summary of the functional components

	Major Features	Requirement Groups[4]
GIS	Import/export of GIS data Creation of features Visualization of GIS Analysis of GIS data Saving GIS data in the relevant study in the database	Pre- and Post-Processors and Data Analysis Tool Embedded GIS functionality
Time series	Import/export of time series Creation of time series Visualization of time series Analysis of time series Association of time series with GIS features Saving time series in the relevant study in the database	Pre- and Post-Processors and Data Analysis Tool
Tables	Import/export of tables Creation of time series Visualization of data Analysis of data Association of tables with GIS features Saving tables in the relevant study in the database	Pre- and Post-Processors and Data Analysis Tool

[4] This refers to the grouping of the requirements used in the ToR /2/

	Major Features	Requirement Groups[4]
Hydro objects	Creation of hydro objects Extraction of hydro objects from models Visualization of hydro objects Association of hydro objects with GIS features Saving the hydro object in the relevant study in the database	Pre- and Post-Processors and Data Analysis Tool Dynamic Water Budget and Allocation Model Process Models
Scenarios	Registration and export of model definitions Creating scenarios Saving in the relevant study in the database Running scenarios Comparing scenarios	General Utilities
Optimization scenarios	Creating optimization targets Using the Scenarios functional component to run the optimization scenario	General Utilities
Ensemble scenarios	Creating ensemble scenarios Using the Scenarios functional component to otherwise manage the ensemble scenario	General Utilities
Linked models	Creating linked models. Using the Scenarios functional component to otherwise manage the models and their scenarios	General Utilities
Ensembles	Creating ensembles Saving ensembles in the relevant study in the database	Pre- and Post-Processors and Data Analysis Tool
Model tools	Creating models Calibrating models	Dynamic Water Budget and Allocation Model Process Models
MCA	Creating multi-criteria analysis setups Saving setups in the relevant study in the database Running setups	Multi-criteria- Analysis
CBA	Creating cost benefits analysis setups Saving setups in the relevant study in the database Running setups	Multi-criteria- Analysis

	Major Features	Requirement Groups[4]
Indicators	Defining indicators for MCA and CBA analysis Saving the indicator definitions in the relevant study in the database Calculating the indicator values	Multi-criteria- Analysis
Reporting	Create reports Publish reports Save reports in the relevant study in the database	Information Management System
Studies	Creating studies Administer study users Import and export studies	Information Management System
Meta data	Manage meta data	Information Management System

Table 3.8 below shows a mapping between functional components and use cases. Note in the table that this is not a one-to-one mapping as:

- Some use cases employ more than one functional component
- Some functional components are used by more than one use case
- Some functional components are not related to any use case, i.e. they have been identified through the requirements, meetings in Addis Ababa or Cairo or initial design considerations.

Table 3.8 Mapping between functional components and generalised use cases

Functional components	Use cases
CBA	Run CBA
	Setup CBA
Ensembles	Run ensemble scenario
	Setup ensemble scenario
	Use ensemble generator tool
GIS	Assess data availability
	Edit GIS data
	Import GIS data
	Inspect GIS layers
	Link time series to feature
	Use GIS catchment delineation tool
	Use GIS tool

Functional components	Use cases
	Visualize GIS data
Hydro objects	Create hydroobject
	Edit hydroobject
Indicator	Define indicator
	Use indicator tool
Linked models	Create linked model
	Run linked scenario
	Test model inter-communication
	Test model linkage
	Run MCA
MCA	Setup MCA
	Assess data availability
Meta data	Inspect GIS layers
	Inspect time series
	Calibrate MIKE 11
Model Tools	Calibrate MIKE BASIN
	Calibrate model
	Clone and modify MIKE BASIN model and scenario
	Edit model
	Setup MIKE 11 model
	Setup MIKE BASIN
	Setup semi-distributed rainfall-runoff model (MIKE SHE Studio)
	Test model registration
	Use rainfall runoff model
	Create optimization scenario
Optimization Scenarios	Define optimization scenario
	Run optimization scenario
	Publish in report
Reporting	Clone and modify MIKE BASIN model and scenario
Scenarios	Clone and modify model and scenario
	Compare scenarios
	Create scenario
	Modify DSS scenario
	Register MIKE 11 model

Functional components	Use cases
	Register MIKE BASIN model
	Register model
	Run ensemble scenario
	Run scenario
	Setup ensemble scenario
	Test create scenario
	Test model registration
	Test run scenario
	Use rainfall runoff model
	Create study
Studies	Export study
	Import Study
	Manage study
	Assess data availability
Tables	Import tables
	Use table tool
	Visualise tables
	Assess data availability
Time series	Create time series
	Edit time series
	Filter time series
	Import time series
	Inspect time series
	Use demand calculator tool
	Use soil erosion tool
	Use time series tool
	Visualise time series
	Analyse model tool inter-communication options
User action	Analyse of IModelAdapter
	Change model tool in order to do model tool to model tool communication
	Code model tool adapter
	Define scenario
	Design model tool adapter
	Establish understanding of model tool
	Take decision

4 SYSTEM FEATURES

This chapter describes the functionality of the NB DSS system based on the functional components established in chapter 1 and briefly introduced in Chapter 2.

The functional components are a logical grouping of functionality, i.e. not concrete implementation components. These are established on basis of the functional components and are discussed in the SAD /5/.

Each functional component is discussed in terms of:

- A general description focusing on the scope of use, applicability in the Nile basin, temporal and spatial resolution, data requirements and parameters

- List of high-level functionalities

- Mapping to TOR requirements

  Each requirement is referenced to the TOR (chapter, section heading and bullets) and also includes a title or text from the TOR. Additionally clarification notes taken from the discussions during the launch workshop and further comments from DHI have been added. Finally the requirement links to a (possible) entry in the issue list.

  Optional requirements have not been taken into account.

- Derived requirements – The TOR lists for some of the functional components few or, in some cases, no relevant requirements. This chapter establishes for such components *derived* requirements, i.e. requirements that have been identified through the use case analysis performed in chapter 1.

  Note: These requirements are all subject to acceptance by the Change Control Board.

In the sections below tables are used to list the requirements related to the described functional components. During the Launch Workshop a number of clarification notes to selected requirements were recorded. These are marked with "***Clarification***" with the relevant requirements. The consultant in some cases added comments to requirements. These are marked with "**Comment**:".

4.1 Functional Component: GIS

The GIS functional component provides functionality for visualizing GIS data in maps and attribute tables and for storing, querying and processing them. In fact, all functionality related to presenting, editing, analysing, importing or exporting spatial data is part of this component.

The GIS functional component is a generic component providing the user various means and tools for working with spatial data. Often, this component will be used at the initial stages of a study to provide an understanding of the study area and to prepare spatial data for use in setting up models. The component is also frequently used towards the end of a study to display geo-referenced results.

Other characteristics of the component include:

- Association between spatial features and other types of data entities, e.g. time series can be related to gauging stations or catchments

- The component will also be used programmatically by other DSS components when accessing data and inserting data in the database.

- The GIS component will store all its data in the database in a fixed spatial reference system.

- Spatial data can be imported from and exported to files in the following formats: for vector formats like Shape, WKT, KML and for raster formats like GeoTIFF and GRID.

The component utilises 2 types of parameters:

1. Parameters that are used in various types of analyses and visualisation definitions.

   These parameters – which typically are input parameters for various analyses – can be persisted and re-used in similar analyses on different GIS data. I.e., a user can save the specifications used for one analysis and reuse them at a later time during another analysis.

2. Parameters that are used for working with the component.

   These parameters comprise:

   - Search filters (system and user defined)

   - User settings (favourites)

   - “Last used” references

High-level functionality related to the functional components comprises the following:

- Import spatial data

- Export spatial data

- Store spatial data in the database

- Convert between spatial reference systems

- Visualise spatial data in layers

- Display the layer attributes in tables

- Provide basic GIS display
- Provide geo-processing capabilities
- Provide catchment delineation functionality
- Provide spatial interpolation functionality

Table 4.1 maps the functional component with the requirements from the ToR /2/.

Table 4.1 Functional component: GIS - requirement mapping

Req. No.	Comments	Issue ref
2.1.4.1	Support of GIS industry standard formats ***Clarification*** *Shapefiles and KML are de-facto standards that need to be supported.*	
2.1.4.2	OGC compatibility and compliance	
2.1.4.2 (1)	OGC compatibility and compliance: Import / Export of spatial information **Comment:** Supported import formats: Shape files and KML	9
2.1.4.2 (2)	OGC compatibility and compliance: Projections	
2.1.4.3	Support of spatial data pre-processing for model inputs ***Clarification*** *NBI just wanted to make sure that the GIS libraries proposed could do the things described above – functionality for pre-processing for hydrologic analysis. Could include working with multiple map layers to derive new layers by combinations and algorithms on other layers.*	
2.1.4.3 (1)	Support of spatial data pre-processing for model inputs: Process geo-referenced vector and raster data	
2.1.4.3 (2)	Support of spatial data pre-processing for model inputs: Pan, zoom and select	
2.1.4.3 (3)	Support of spatial data pre-processing for model inputs: Catchment delineation ***Clarification*** *This implies that we can do some processing with the GIS, and then draw that information into the DSS for inclusion in models, not necessarily directly within the DSS. We want to be able to see the models both from the spatial context, as well as from the topological context.*	
2.1.4.4	Spatial analysis, interpolation (GIS links)	
2.1.4.4 (1a)	Spatial analysis, interpolation (GIS links): Geo-processing: Intersection	

Req. No.	Comments	Issue ref
2.1.4.4 (1b)	Spatial analysis, interpolation (GIS links): Geo-processing: Union	
2.1.4.4 (1c)	Spatial analysis, interpolation (GIS links): Geo-processing: Proximity analysis	
2.1.4.4 (2a)	Spatial analysis, interpolation (GIS links): Spatial inter-polation: Nearest neighbour	
2.1.4.4 (2b)	Spatial analysis, interpolation (GIS links): Spatial inter-polation: Inverse distance	
2.1.4.4 (2c)	Spatial analysis, interpolation (GIS links): Spatial inter-polation: Moving polynomials	
2.1.4.4 (2d)	Spatial analysis, interpolation (GIS links): Spatial inter-polation: Kriging	

4.2 Functional Component: Time Series

The Time series functional component provides functionality for importing, exporting, visualising, storing, querying and processing time series data.

It is a component providing the user various means and tools for working with time series data. Often, the component will be used extensively by the user during a study to prepare data for later model simulations and after model simulations to visualize and analyse simulation output.

The component will also be used programmatically by other DSS modules when accessing data and inserting data in the database.

Other characteristics of the component include:

- Time series data will be stored in the database using Coordinated Universal Time (UTC) as reference. When being displayed, the data it will be converted to the local time zone used by the workstation – i.e. following the Windows regional settings. The data is stored with a precision of one millisecond. However, many of the time series tools and other components will only work with a precision of one second.

- Time series data can be imported from and exported to ASCII files, Excel files and DFS0[5] files.

The module utilises 2 types of parameters:

1. Parameters that are used in various types of analyses and plot definitions.

   These parameters – which typically are input parameters for various analyses – can be persisted and re-used in similar analyses on different GIS data. I.e., a user can save the specifications used for one analysis and reuse them at a later time during another analysis.

2. Parameters that are used for working with the module.

[5] DFS0 is a DHI file format extensively used by the MIKE models.

These parameters comprise:

- Search filters (system and user defined)
- User settings (favourites)
- “Last used” references

High-level functionality related to the functional components comprises the following:

- Import time series data
- Export time series
- Store time series data
- Visualise time series in charts
- Display time series values in tables
- Create time series
- Edit time series
- Provide statistical time series analysis functionality
- Provide time series analysis functionality
- Provide data treatment and QA functionality

Table 4.2 below maps the functional requirements from the ToR /2/ with the functional component.

Table 4.2 Functional component: Time series - requirement mapping

Req. No.	Comments	Issue ref
2.1.3(1)	IMS: Pre- and Post-Processors and Data Analysis Tools - visualization tools	
2.1.3(2)	IMS: Pre- and Post-Processors and Data Analysis Tools - charting functionality	
2.1.3(3)	IMS: Pre- and Post-Processors and Data Analysis Tools - multiple selections	
2.1.3(4)	IMS: Pre- and Post-Processors and Data Analysis Tools - customization of predefined	
2.1.3(5)	IMS: Pre- and Post-Processors and Data Analysis Tools - Charts in IMS	
2.1.3(6)	IMS: Pre- and Post-Processors and Data Analysis Tools - export interface	
2.1.3.1	Embedded Statistical Tools	
2.1.3.1 (1a)	Embedded Statistical Tools: Descriptive statistics: Minimum	

Req. No.	Comments	Issue ref
2.1.3.1 (1b)	Embedded Statistical Tools: Descriptive statistics: Maximum	
2.1.3.1 (1c)	Embedded Statistical Tools: Descriptive statistics: Mean	
2.1.3.1 (1d)	Embedded Statistical Tools: Descriptive statistics: Median	
2.1.3.1 (1e)	Embedded Statistical Tools: Descriptive statistics: Mode	
2.1.3.1 (1f)	Embedded Statistical Tools: Descriptive statistics: Standard deviation	
2.1.3.1 (1g)	Embedded Statistical Tools: Descriptive statistics: Skewness	
2.1.3.1 (1h)	Embedded Statistical Tools: Descriptive statistics: kurtosis	
2.1.3.1 (1i)	Embedded Statistical Tools: Descriptive statistics: Empirical frequency	
2.1.3.1 (1j)	Embedded Statistical Tools: Descriptive statistics: Cumulative frequency distributions	
2.1.3.1 (2)	Embedded statistical tools: Statistical tests	
2.1.3.1 (2a)	Embedded statistical tools: Statistical tests: Stationarity	
2.1.3.1 (2b)	Embedded statistical tools: Statistical tests: Homogeneity	
2.1.3.1 (2c)	Embedded statistical tools: Statistical tests: Randomness	
2.1.3.1 (3)	Embedded statistical tools: Double mass analysis	
2.1.3.1 (4)	Embedded statistical tools: Frequency distributions	
2.1.3.1 (4a)	Embedded statistical tools: Frequency distributions: Common distributions	
2.1.3.1 (4b)	Embedded statistical tools: Frequency distributions: Parameter estimation	
2.1.3.2	Time series analysis analysis tools	
2.1.3.2 (1)	Time series analysis tools: Auto and cross correlation analysis	
2.1.3.2 (2)	Time series analysis tools: Duration curves	
2.1.3.2 (3)	Time series analysis tools: Regression analysis	
2.1.3.2 (4)	Time series analysis tools: Generation of long time-series.	
2.1.3.2 (5)	Time series analysis tools: Projection of future demands.	10
2.1.3.3	Data treatment and quality assurance tools	
2.1.3.3 (1)	Data Treatment and Quality Assurance Tools: Data Validation Tools	
2.1.3.3 (1a)	Data Treatment and Quality Assurance Tools: Data Validation Tools: Physical or numerical limits	

Req. No.	Comments	Issue ref
2.1.3.3 (1b)	Data Treatment and Quality Assurance Tools: Data Validation Tools: rate of rise and fall	
2.1.3.3 (1c)	Data Treatment and Quality Assurance Tools: Data Validation Tools: Comparison against observed behavior	
2.1.3.3 (1d)	Data Treatment and Quality Assurance Tools: Data Validation Tools: Comparison against related variables	
2.1.3.3 (1e)	Data Treatment and Quality Assurance Tools: Data Validation Tools: Comparison between adjacent stations	
2.1.3.3 (1f)	Data Treatment and Quality Assurance Tools: Data Validation Tools: General comparison between different variables	
2.1.3.3 (2)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods	
2.1.3.3 (2a)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Maps	
2.1.3.3 (2b)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Tables	
2.1.3.3 (2b.1)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Tables: All station information	
2.1.3.3 (2b.2)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Tables: Time interval	
2.1.3.3 (2b.3)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Tables: Display statistics	
2.1.3.3 (2c)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Test on extremes	
2.1.3.3 (2c.1)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Test on extremes: upper/lower limits	
2.1.3.3 (2c.2)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Test on extremes: limits in rise and fall	
2.1.3.3 (2c.3)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Test on extremes: User defined rules	
2.1.3.3 (2d)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of temporal variation	
2.1.3.3 (2d.1)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of temporal variation: TS Plot	
2.1.3.3 (2d.2)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of temporal variation: lag/shift	
2.1.3.3 (2d.3)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of temporal variation: Residuals and movi...	
2.1.3.3 (2e)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of longitudinal/spatial variation	
2.1.3.3 (2e.1)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of longitudinal/spatial variation: Displa...	

Req. No.	Comments	Issue ref
2.1.3.3 (2e.2)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of longitudinal/spatial variation: Map pr...	
2.1.3.3 (2e.3)	Data Treatment and Quality Assurance Tools: Graphical Screening Methods: Inspection of longitudinal/spatial variation: Spatia...	

4.3 Functional Component: Hydro objects

Hydro objects describe water resource management related infrastructure such as reservoirs and irrigation schemes. They often play an important role in modelling water related processes within an area – both in providing input and in providing output from a modelling exercise. E.g., the former could be through required water demand per area from an irrigation scheme and the latter optimal reservoir operations calculated based on an optimizing hydrodynamic model.

NB DSS will contain hydro objects originating from 2 different types of input sources:

1. Manually entered – The user can manually enter the hydro object characteristics from the NB DSS user interface.

2. Imported from a model – Hydro objects can be extracted from model setups during a model registration process, see Section 4.10.

Hydro objects are typically characterised through numbers, curves, text and maybe even images. An example could be the geometry of a reservoir, its level/volume curve and textual operations rules. The images could then be photos of the reservoir and its dam.

There does not exist a single taxonomy that describes all type of hydro objects. Different model tools describe the same type of hydro object in different terms - and because the DSS Front-end shall be able to integrate with many different model tools, the data model supporting the hydro objects will naturally have to depend on the model tools. This combined with the requirement that the DSS Front-end shall support integration of new model tools without having to be modified or even re-compiled necessitates a flexible data model that probably needs to be partly supported by a schema definition[6] of the hydro object characteristics.

I.e. support for that new hydro objects can be added to the system over time is required.

Hydro objects are associated with GIS features and –for the ones being imported with models (scenarios). This implies that the user will be able to view the location of the hydro objects on a map and can access the hydro object characteristics from navigating the map.

[6] A description of how to interpret the hydro object characteristics, e.g. as an XMl schema.

High level functionality comprises the following:

- Create hydro objects
- Import hydro objects from model setups
- Visualise hydro objects on a GIS map
- Display hydro object characteristics
- Search for hydro objects

The concept of hydro objects originates from discussions between the Client and the Consultant and is therefore not clearly established in the ToR /2/. The “4+1” use cases do neither define the context of concept clearly. Table 4.3 maps the concept to relevant requirements from the ToR /2/ and Table 4.4 establishes requirement based on the mentioned discussions.

Table 4.3 Functional component: Hydro objects – requirement mapping

Req. No.	Comments	Issue ref
2.1.5.2	Basic set of pre-defined node types	24
2.1.5.2 (1)	Start or input nodes, representing ...	24
2.1.5.2 (2)	Demand nodes, representing areas of water use ...	24
2.1.5.2 (3)	Structural components ...	24
2.1.5.2(4)	Control nodes representing points in the network which are used for monitoring, compliance, calibration purposes...	24
2.1.5.2 (5)	Geometry nodes ...	24
2.1.5.2 (6)	Aquifers ...	24
2.1.5.2 (7)	End nodes...	24

Table 4.4 Functional component: Hydro objects – derived requirements

Req. No.	Comments	Issue ref
X.1	It shall be possible to manually create hydro objects	
X.2	It shall be possible to extract hydro object from model setups that are being registered	
X.3	It shall be possible to associate hydro object with GIS features	

Req. No.	Comments	Issue ref
X.4	It shall be possible display hydro object characteristics	
X.5	It shall be possible to add new types of hydro objects to a system without having to change the source code of the system or re-compile it	
X.6	It shall be possible to search for hydro objects on type and name in the database	

4.4 Functional Component: Tables

The Tables functional component provides functionality for visualizing, storing, querying and processing tables. Table objects are objects related to semi-structured data like summaries of processed data.

The table manager of the DSS manages any kind of information that is to be represented in tabular form in order to reflect the way users see that information. We used the terms "virtual tables" or "organized assembly of scalars" as synonyms.

With regard to functionalities the "virtual tables" used in the Table Manager can be distinguished as follows:

1. Tabular information that contains primary information to be stored in the database: Examples are area-capacity or rating curves. The user will be provided with forms that allow editing (insert, update and delete) cell/scalar values organized in a tabular format. This information can be stored in the RDBMS. The structure of these "virtual tables" is not the same as the structure of the static data model in the RDBMS. Hence the Table Manager has the "knowledge" how the information contained in the "virtual Tables" is to be mapped to the tables/attributes in the RDBMS.

2. Derived information that results from queries and other data manipulation processes: For example the user may want to see statistical properties of several time series in a tabular form, or aggregated scalars in a specific matrix-form to be used subsequently in other tools such as the MCA-Tool. The Table manager shall enable the user to retrieve and use this information. Depending on the complexity of the data manipulation process to generate this kind of "virtual table" the Table Manager shall give the following options:

    i. The derived information that will populate the "virtual table" will always be produced in real-time – just in the moment the generation of the "virtual table" is launched (analogy to views).

    ii. The derived information that will populate the "virtual table" will be stored in the RDBMS. It will be updated when any part of source data to generate this virtual table is amended (analogy to materialized views).

Hence the Table Manager has two types of "knowledge":

1. For primary information: Mapping of attributes in the “virtual table” to attributes in the static data model in the RDBMS

2. For derived information: The data manipulation process to generate the content in the “virtual table”.

The Table Manager shall provide a mechanism for the user to define “virtual tables” and manage the respective “knowledge”. This functionality (the knowledge definition for the Table Manager) can be realized with a scripting interface that enables the user to define rules how to (1) map, (2) retrieve and (3) transform data to be represented in form of “virtual tables”.

Note the Tables functional component has not been directly mentioned in the ToR /2/, but is the result of the use case analysis (see chapter). Specifically, the following use cases listed in /3/ operate with tables:

- User story 1, use case 3, step b
- User story 1, use case 4, step b
- User story 1, use case 8, step b
- User story 1, use case 9
- User story 3, use case 2.3, step a

High-level functionality related to the functional component comprises the following

- Create a table
- Set values at table cells
- Perform simple spreadsheet-like calculations
- Store the table in the database
- Associate tables with GIS features
- Defined fixed table layouts for e.g. indicators and rating curves
- Programmatically access the table cells

Requirements related to the overall architecture of the NB DDS for the functional components are established in Table 4.5.

Table 4.5 Functional component: Tables - derived requirements

Req. No.	Comments	Issue ref
X.7	It shall be possible to associate tables with features	

Req. No.	Comments	Issue ref
X.8	It shall be possible to define fixed table layouts, either through database meta data or through attributes in the database table storing the table	
X.9	It shall be possible for other components to access the tables and read and set values at specific cells within the table	
X.10	It shall be possible to associate script functions to individual cells in order to calculate the cell value through the script	
X.80	It shall be possible to associate tables to hydro objects	

4.5 Functional Component: Scenarios

The Scenario component provides the core functionality related to managing, analysing and comparing scenarios. Different scenarios are typically variations of a certain model and different scenarios may involve variations of any input data including parameters, variables and model topology and geometry. A specific scenario may include a single model, linked models (which is just another model) as well as optimization and ensemble simulations.

The model setup will be created in the native model tool user interface, and when completed, it shall be registered in the Scenario component. During registration of a model, the model boundaries, properties and parameters are brought into the DSS Database and when registration is completed, they can be accessed from the DSS Front-end.

Once a model setup has been registered, the scenario manager provides functionality to clone and modify (i.e. derive new scenarios). This involves selection and editing of the input time series data that shall be modified, as well as selection of the results that shall be made available and stored in the DSS database after execution of the scenario. Examples of input time series data that can be modified during the creation of a derived scenario is climate variables, reservoir operation rules, channel flow capacities, environmental flow requirements and water user demands. Changes in the model topology require the use of the respective model tool UI.

The Scenario component includes functionality to visualize and compare results of one or more scenario simulations. The result presentation will be based on a GIS Map with a spatial display of the model features (e.g. river reaches, water users and reservoirs). The results produced by a scenario simulation will be associated with the relevant spatial features. Results may subsequently be visualised, for example, by clicking on the relevant spatial feature. Results may be further processed into indicators and summary statistics using the Time Series-, Tables- and Indicators components, and if appropriate, interpolated into spatial maps using the GIS Manager.

Sensitivity of model output to changes in model input data will be handed by comparing the output of pertubated scenarios in the Scenario manager.

The result of a scenario comparison can, if desired, be saved as a separate entity in the database for later retrieval and reporting.

Finally, the Scenario component contains the functionality to delete existing scenarios, and optionally, along with the input- and output data that is associated with it.

The high-level functionality is listed below:

- Registration of models
- Creation of derived Scenarios.
- Execution of scenarios
- Retrieval and visualization of scenario simulation results.

Table 4.6 Functional component: Scenarios – requirement mapping

Req. No.	Comments	Issue ref
2.1.6.1	Model scenario management ***Clarification*** *Scenario discussion should be clear now, based on discussions during the workshop.*	
2.1.6.3	Direct scenario comparison **Comment:** A wizard-like facility allowing comparison of heterogeneous outputs between different scenarios of different models is needed. The tool should aid the comparison by proposing items to compare based upon mode scenario, data type, location, but allow the user to determine exactly what can be compared. The result of the comparison shall be stored in the database.	
2.1.6.5	Sensitivity Analysis	21

Table 4.7 Functional component: Scenarios - derived requirements

Req. No.	Comments	Issue ref
X.11	It shall be possible to register a model setup as a scenario.	
X.12	It shall be possible to create a derived scenario from the DSS Front end.	
X.13	It shall be possible to edit selected scenario input data from the DSS front end	
X.14	It shall be possible to execute a scenario from the DSS front end.	
X.15	It shall be possible to retrieve results from selected scenario executions	
X.16	It shall be possible to associate the results produced during a scenario execution to the feature that produced them in a GIS Map.	
X.17	It shall be possible to view/edit an existing scenario.	
X.18	It shall be possible to delete an existing scenario, optionally along with all the results that were produced by execution of that scenario.	

4.6 Functional Component: Linked Models

The Linked models functional component includes functionality for linking different models together in a sequential manner. Models linked in this way is for the NB DSS conceptually still just a single model although it consist of a number of linked models.

The NB DSS supports linked models similar to non-linked models with respect to creating scenarios, running scenarios and comparing scenarios.

When a scenario based on a linked model is executed, all the "single model setups" constituting the linked model is executed sequentially, and the results of the "single models" are imported in the NB DSS database and can be used for e.g. scenario comparisons. I.e. similar to what is described in section 4.5

The high-level functionality is listed below:

- Establishes a link between two registered models

Table 4.8 Functional component: Linked models – requirement mapping

Req. No.	Comments	Issue ref
2.1.5.10	Model nesting, hierarchical linkage of networks	

Table 4.9 Functional component: Linked models - derived requirements

Req. No.	Comments	Issue ref
X.19	It shall be possible to create a link between two registered models, such that the output of the first model simulation serves as input to the next model.	
X.20	It shall be possible to define and execute scenarios based on a linked model from the NB DSS.	
X.21	It shall be possible to retrieve results from all the "single model simulations" when running a linked model based scenario	

4.6.1 Dynamically Linked Models

NB DSS supports dynamic linking – time step by time step - of models. A dynamically linked model is a model that makes use of 2 or more model tools that exchange data during the simulation. This in contrast to sequentially linked models where one model tool simulation provides the input for the next model tool simulation

OpenMI is a standard for how to enable such model tool integration – also across model tools from different vendors. A model tool is OpenMI compliant if it just implements one OpenMI integration interface; but one interface is typically not sufficient in order to communicate all kinds of data among the integrated model tools. The MIKE tools do not all have full support for OpenMI (MIKE11 and MIKE SHE are OpenMI compliant while MIKE BASIN is not). Although two OpenMI models may be linked, it is by no means a trivial task. It typically requires a substantial effort and expert knowledge about the internal part of the model tool including the numerical solution of the governing equations.

Seen from the NB DSS point of view there is no difference between a dynamically linked model and a "single" model. In both cases there has to exist an adapter that the NB DSS can use in order to read and write model setups, run simulations etc.

This adapter will need to know about the OpenMI specifics as well as the internals from the included model tools, i.e. it has to be a separate adapter from the adapters that "just" knows about a model tool.

4.7 Functional Component: Ensembles

In the NB DSS context, an ensemble is a collection of time series that represents different realizations of the same variable (e.g. daily rainfall, monthly reservoir inflow, and water user demands).

The NB DSS Ensembles component provides functionality to generate and manage (edit/delete) ensembles of time series that, once established, can be used as input data in an Ensemble Scenario (section 4.8).

Ensembles are generated, for instance, by repeatedly applying stochastic generation techniques, which are normally configured using statistical properties of historical data (e.g. climate data)The stochastic methods provided in the component includes methods that can take temporal- as well as spatial correlations (multi-site) into consideration, and the generated ensembles can consist of any number of time series, covering any period of time.

The high-level functionality is listed below:

- Provides methods for generation of long, and/or multiple time series.
- Provides methods that take the spatial and temporal correlation between variables into consideration.
- Manages the collection of ensembles, and allows the user to edit and delete existing ensembles.

  The Consultant has not identified requirements in the ToR /2/ that are clearly referring to ensembles, but the ToR does contain requirements that would be required for ensemble modelling.

Table 4.10 Functional component. Ensembles – Requirement mapping

Req. No.	Comments	Issue ref
2.1.3.2 (4)	The data analysis tool shall able to generate long time-series for sensitivity analysis and impact assessment.	

Table 4.11 Functional component: Ensembles - derived requirements

Req. No.	Comments	Issue ref
X.22	It shall be possible to create a new time series ensemble consisting of one or more time series using one of the available stochastic generation techniques.	
X.23	It shall be possible to manage (view/edit/delete) existing ensembles	

4.8 Functional Component: Ensemble scenarios

An Ensemble Scenario is similar to a regular scenario (see Section 4.5) the only difference being that one or more input time series have been replaced by an Ensemble of time series (see Section 4.7).

Replacing one or more input time series by ensembles of time series implies that when an Ensemble Scenario is executed, several executions of the underlying model tool is required.

If an ensemble scenario takes more than one ensemble as input, the user will be presented with different options for defining how many executions that should be used to cover different combinations of the time series contained in different ensembles to obtain a reasonable accuracy in the produced output.

The output of an ensemble scenario can either be the full set of results produced for all model tool executions, and/or a statistical summary of the produced results.

The high-level functionality is listed below:

- Link selected scenario inputs to time series ensembles.
- Execution of ensemble scenarios.
- Apply post processing rules as per user definition.

Table 4.12 Functional component: Ensemble scenarios – Requirement mapping

Req. No.	Comments	Issue ref
2.1.6.6 (3)	The stochastically generated sequences shall be used for deterministic applications. **Comment:** Time series ensembles are available as model inputs in a similar manner as other time series	

Table 4.13 Functional component: Ensemble scenarios - derived requirements

Req. No.	Comments	Issue ref
X.24	It shall be possible to replace a reference to a single time series with a reference to a time series ensemble in a scenario definition.	
X.25	If a scenario contains references to more than one ensemble, it shall be possible to specify how many scenario executions that should be used to cover different combinations of ensemble time series to obtain a reasonable accuracy of the results.	
X.26	It shall be possible to execute an ensemble scenario, systematically according to the options provided in 3.8.2.	
X.27	It shall be possible to retrieve the result of an ensemble simulation as a set of summary statistics (per time step, per model feature).	
X.28	It shall be possible to retrieve all results produced during each model tool execution in an ensemble scenario execution.	

4.9 Functional Component: Optimization Scenarios

The Optimization Scenarios component contains the functionality required for optimisation of individual scenarios (models), and provides the user with functionality to find optimal solutions for defined objectives and constraints.

A range of optimisation algorithms, including Shuffled Complex Evolution and Non-Dominated Sorting Genetic Algorithm, will be available. The component includes functionality for defining forms for defining objective functions, constraints, initial values and stop criteria.

When executed, the Optimisation Scenario produces a list of the optimum parameters and performance statistics, as well as the result of the scenario when executed with the optimised set of parameters and/or variables.

The Optimization Scenario component extends the functionality of the Scenarios component.

The high-level functionality is listed below:

- Enables the user to optimize individual scenarios (models)
- Select optimization algorithms.
- Define objective functions and constraints

Table 4.14 Functional component: Optimization scenarios – requirement mapping

Req. No.	Comments	Issue ref
2.1.6.4	Simulation based optimization	

Table 4.15 Functional component: Optimization scenarios - derived requirements

Req. No.	Comments	Issue ref
X.29	It shall be possible to apply optimization on the individual scenarios.	
X.30	It shall be possible to define objectives, constraints, initial guesses and stop criteria in the DSS front end	
X.31	It shall be possible to retrieve the results of the optimized scenario, along with the performance statistics of the optimizer.	
X.32	It shall be possible to register an optimized scenario as a new scenario.	

4.10 Functional Component: Model Tools

The DSS Front-end is open ended with respect to model tools. This is done to ensure that the system can be extended with new model tools, should the need arise. As part of the NB DSS project the Consultant supplies a number of Model tools but any other Model tool may, in principle, be linked to the DSS front-end. The linkage is ensured through an integration mechanism called Model Adapters.

The functionality of the Model tools and their relations to requirements are described in Section 4.10.1.1 to Section 4.10.1.4 and the Model Adapters in section 4.10.2.

4.10.1 Specific Model Tools

The actual model tool components delivered with the system comprises:

1 MIKE BASIN

2 MIKE 11

3 MIKE SHE Studio

4 DHI.Optimization

The following sections describe these NB DSS model tools in terms of functionality and the type of data they consume and produce. Section 4.10.1.5 describes existing couplings among relevant MIKE models.

4.10.1.1 MIKE BASIN

MIKE BASIN is a stand-alone, network-based, dynamic, water budget and allocation model that represents the interactions between water supply and demand for given set of apportionment rules. Water users and sources are represented by a set of predefined nodes connected to the river network by link channels. The model supports conjunctive use of surface and groundwater, and a wide range of water allocation and management rules are available.

Structural components such as weirs, reservoirs and diversion nodes are represented by a separate set of nodes that can be attached directly to the river network, and operated using the available management rules.

Runoff from the individual catchments can either be prescribed as time series or by linking MIKE BASIN to a rainfall-runoff model tools such as the NAM Rainfall-Runoff model (lumped) and MIKE SHE Studio (semi-distributed). The links between MIKE BASIN and the rainfall runoff models are sequential.

With the WQ Module, MIKE BASIN can simulate steady-state and unsteady-state reactive transport of the most important nutrient substances that affecting water quality.

MIKE BASIN comes with a large number of predefined management and water allocation rule types, but these, as well as the behaviour of the individual nodes and reaches may be further customized using the MIKE BASIN COM programming interface.

The MIKE BASIN COM Interface

A MIKE BASIN Engine object can be instantiated using the MIKE BASIN COM Interface (IEngine). The Engine object comprises the possibilities for customizing MIKE BASIN programmatically and provides access to the entire model setup and simulation information. The model setup itself is composed of IModelObjects, which are accessed through IEngine. IModelObjects can be physical features or network elements, such as nodes, reaches, and catchments. Additional implementations of IModelObjects are logical or computational entities, such as allocation rules and water quality models.

The engine part of MIKE BASIN is a C++ console based application that exposes a programming interface through Microsoft COM. This interface – which is described in the MIKE BASIN documentation – is depicted in Figure 4.1.

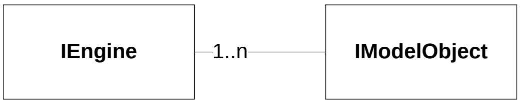

Figure 4.1 MIKE BASIN Engine interface

Input to MIKE BASIN is provided through a single input files, and the MIKE BASIN Data Access component is responsible for all communication between the user interface, data storage and model engine. This is depicted in Figure 4.2.

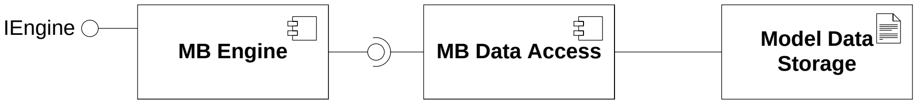

Figure 4.2 MIKE BASIN Data Access

MIKE BASIN input is currently stored in a Microsoft Access file, but this is currently being redesigned by DHI. Whether a different storage approach will be selected has not yet been decided.

Table 4.16 below lists the high level MIKE BASIN functionalities.

Table 4.16 MIKE BASIN high-level functionalities

High-level functionalities	Comments
Network-based, dynamic, water budget and allocation model.	Multiple water allocation and sharing algorithms are available.
River Reaches and Link Channels	Several routing options available. Optionally leakage and evaporation can be assigned.
Water Users	The general Water User Node can be customized to represent different types of users (Industry, Cities, Waste Water Treatment Plants etc.). Demands are prescribed.
Irrigation	Dynamic calculation of irrigation demand and crop yield estimation for individual fields as well as for entire irrigation schemes (FAO56/FAO33).
Reservoirs	Three types of reservoirs are supported: Lakes (not operated), Rule Curve Reservoirs, and Allocation Pool Reservoirs (capacity sharing). Inter-reservoir transfer of water supported.
Hydropower	Detailed Hydropower simulation.
Water Quality	Broad scale reactive transport modelling of the most important nutrient substances that affecting water quality.
Customizable	Well documented Engine API (COM).

Table 4.17 lists the functional/non-functional requirements related to MIKE BASIN.

Table 4.17 MIKE BASIN requirement mapping

Req. No.	Comments	Issue ref
2.1.5	Dynamic Water Budget and Allocation Model **DHI Comment:** Requirement split in sub-requirements The interpretation of the 2.1.5.* requirements depend very much on the individual modelling tools and the approach for the implementation of the integration of the modelling tools in the DSS	
2.1.5 (1)	Dynamic Water Budget and Allocation Model - multiple years	
2.1.5 (2)	Dynamic Water Budget and Allocation Model - surface+groundwater	16
2.1.5 (3)	Dynamic Water Budget and Allocation Model – water allocation **Comment:** Requirement split in sub-requirements	
2.1.5 (3.1)	Dynamic Water Budget and Allocation Model - water allocation per user	
2.1.5 (3.2)	Dynamic Water Budget and Allocation Model - source priority per user	
2.1.5 (3.3)	Dynamic Water Budget and Allocation Model - operation rules	
2.1.5 (3.4)	Dynamic Water Budget and Allocation Model - groundwater management rules	
2.1.5 (3.5)	Dynamic Water Budget and Allocation Model - operation rules of the diversion structures	
2.1.5 (3.6)	Dynamic Water Budget and Allocation Model - water allocation based on targets	17
2.1.5 (3.7)	Dynamic Water Budget and Allocation Model - proportional water allocation **Comment:** MIKE BASIN supports proportional allocation of available water at abstraction nodes and at individual water users	18
2.1.5 (3.8)	Dynamic Water Budget and Allocation Model - re-use of drainage water	
2.1.5 (4)	Dynamic Water Budget and Allocation Model - explicit transfer scheme	19
2.1.5.1	Data driven, user specified, interactive network configuration **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.2	Basic set of pre-defined node types	

Req. No.	Comments	Issue ref
	Comment: The node types available are those implemented in the model tools.	
2.1.5.2 (1)	start or input nodes representing sub-catchment and groundwater	
2.1.5.2 (2)	Demand nodes	
2.1.5.2 (3)	Structural components	
2.1.5.2 (4)	Control nodes	
2.1.5.2 (5)	Geometry nodes	
2.1.5.2 (6)	Aquifers	
2.1.5.2 (7)	End nodes	
2.1.5.3	Geo-referenced network geometry **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.4	Explicit routing of flow **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.5	Temporal scope and time step **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.6	Explicit mass budget, error statistics	
2.1.5.7	Explicit groundwater representation and coupling	
2.1.5.8	Multiple reservoirs, including hydropower generation	
2.1.5.9	Variable reach geometry	13
2.1.5.13	Lateral inflow, lateral catchments, floodplain representation	
2.1.5.14	Hydropower production	
2.1.5.15	Hydraulic model (1D) **Comment:** The hydraulic routing is provided by MIKE Basin and MIKE 11.Likewise with water quality and sediment transport functionalities.	
2.1.5.16	Sediment transport in river networks and reservoir siltation	

Req. No.	Comments	Issue ref
2.1.7.1	Rainfall-runoff models (lumped, semi-distributed) **Comment:** RR model are like other models – and are available in a similar manner. The model tool configuring the rainfall-runoff model defines the type of process.	
2.1.7.2	Irrigation water demand estimation, crop production model	
2.1.7.3	Water quality model (DO/BOD, conservative, first order decay **Comments:** Water quality in rivers and other surface water bodies are simulated considering transport and degradation of various pollutant substances When using simple conceptual lumped/semi-distributed groundwater models water quality simulations are typically done using a simple "catchment load" approach where the pollution loads is calculated per catchment based on aggregated non-point and point pollution loads from urban areas, industries and agriculture (crop production and live-stock). These loads are then converted to concentrations by associating them with lateral inflows to rivers calculated, for instance, by the catchment run-off model. Alternatively, water quality may be considered in the linear reservoir calculations. The latter is however often problematic due to the assumption of complete mixing within the reservoirs.	
2.1.7.5	Multiple evapotranspiration estimation methods **Comment:** FAO56 reference evapotranspiration is implemented	
2.1.7.6	Support for user defined process representation/algorithms ***Clarification*** *We want a feature that allows the user to write some code for a "rule-based" simulation, where the user can influence the execution of the simulation. may include a rule editor, rule language, and interaction with the engine.* **Comments:** MIKE BASIN includes COM interfaces that allows a user to define custom process behaviour	
2.1.6.2	Embedded calibration methods with error statistics. **Comment:** Model calibration is carried out in the respective model tools using whatever facilities are available. Additional comparison of model time series with observations and other time series are handled by the general time series analysis within the Timeseries functional compo-	

Req. No.	Comments	Issue ref
	nent.	
2.1.6.2 (1)	The system shall provide embedded calibration tools **Comment:** See above	
2.1.6.2 (2)	The calibration tools shall support calibration of any part of the basin under consideration **Comment:** The calibration related to the entire model as defined by the model tool.	
2.1.6.2 (3)	The system shall provide appropriate objective functions **Comment:** As calibration is carried out from within the model tool, the DSS will not provide this facility.	
2.1.6.2 (4)	The calibration tools shall enable the users to determine model parameter sets **DHI Comment:** The calibration in the DSS does not know about model parameters, but is limited to time series analysis and comparison. The available calibration tools are determined by the model tools.	

4.10.1.2 MIKE 11

MIKE 11 is a stand-alone hydrodynamic model for simulating flow, water level, water quality and sediment transport in rivers, flood plains, irrigation canals, reservoirs and other inland water bodies.

The hydrodynamic module of MIKE 11 provides a library of computational methods for steady and unsteady flow in branched and looped channel networks as well as quasi two-dimensional flow simulation on flood plains.

Among the available Rainfall-Runoff models that can be used to simulate inflow to the MIKE 11 river network are the NAM rainfall-runoff model (sequentially or dynamically linked), and the distributed MIKE SHE Studio model (dynamically linked).

MIKE 11 can be used in applications ranging from simple design investigations to large forecasting projects, including the simulation of complex hydraulic structure operation policies.

Typical applications include flood risk analysis, real time flood and water quality forecasting, dam break analysis, sediment transport, and salt intrusion studies.

The engine part[7] of MIKE 11 is a console application coded in CodeGear Delphi. Input to MIKE 11 is provided through a number of ASCII and binary input files. This is depicted in Figure 4.3 and Figure 4.4.

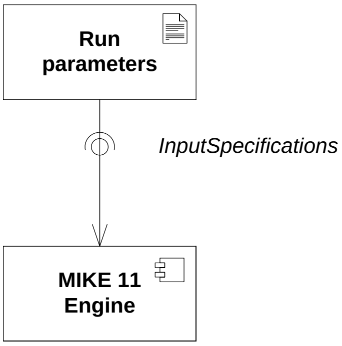

Figure 4.3 MIKE 11 input interface

Figure 4.3 illustrates that the interface to the engine part of MIKE 11 is a file with *run parameters*. This file references other input files as depicted in Figure 4.4.

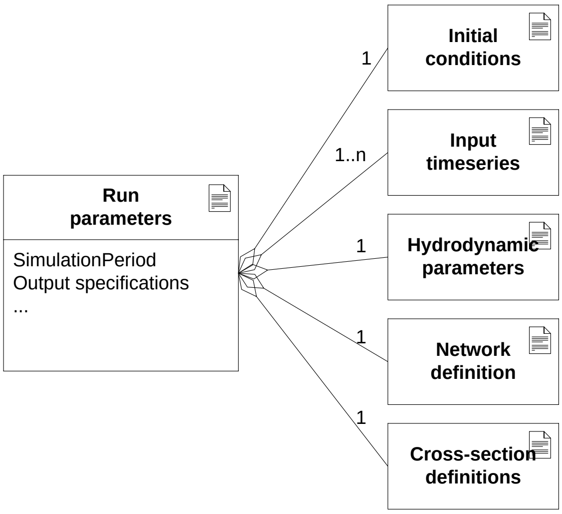

Figure 4.4 MIKE 11 Input files

The Run parameter, Hydrodynamic parameter and network definition files are ASCII files adhering to the DHI Parameter File System (PFS) format specifications. PFS is a format similar to XML. The layouts of the MIKE 11 PFS files are defined through a PFS template file (similar to an XML schema file). MIKE 11 does neither provide a programming interface to set and retrieve input parameters nor expose an API for controlling the model execution.

[7] The engine part is the part that performs the actual simulation

Table 4.18 below lists the high level MIKE 11 functionalities.

Table 4.18 MIKE 11 high-level functionalities

High-level functionalities	Comments
Hydrodynamic routing	Flexible and robust solution of the non-linear St. Venant Equations for open channel flows.
Structure operation	Including weirs, culverts, bridges, pumps, energy loss and tabulated structures.
Stratified flow	Modelling of vertical density differences, such as salinity or temperature in two - or multi-layered stratified water bodies.
Real time flood forecasting	Modelling of real time flood forecasting including state updating and data assimilation features.
Sediment transport modelling	Transport, erosion and deposition of Sediments (cohesive as well as non-cohesive), including morphological changes on river bed topography.
Water Quality	Transport and spreading of conservative pollutants and constituents with linear decay (including heat).

Table 4.19 lists the functional/non-functional requirements related to MIKE 11.

Table 4.19 MIKE 11 requirement mapping

Req. No.	Comments	Issue ref
2.1.5 (1)	Dynamic Water Budget and Allocation Model - multiple years	
2.1.5 (3.3)	Dynamic Water Budget and Allocation Model - operation rules	
2.1.5 (3.5)	Dynamic Water Budget and Allocation Model - operation rules of the diversion structures	
2.1.5 (4)	Dynamic Water Budget and Allocation Model - explicit transfer scheme	19
2.1.5.1	Data driven, user specified, interactive network configuration **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.2	Basic set of pre-defined node types **Comment:** The node types available are those implemented in the model tools.	
2.1.5.2 (1)	start or input nodes	
2.1.5.2	Basic set of pre-defined node types **Comment:** The node types available are those implemented in the model tools.	
2.1.5.2 (1)	start or input nodes	

Req. No.	Comments	Issue ref
2.1.5.2 (3)	Structural components	
2.1.5.2 (4)	Control nodes	
2.1.5.2 (5)	Geometry nodes	
2.1.5.2 (7)	End nodes	
2.1.5.3	Geo-referenced network geometry **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.4	Explicit routing of flow **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.5	Temporal scope and time step **Comment:** This is achieved via MIKE Basin/MIKE 11	
2.1.5.8	Multiple reservoirs, including hydropower generation	
2.1.5.9	Variable reach geometry	13
2.1.5.13	Lateral inflow, lateral catchments, floodplain representation	
2.1.5.14	Hydropower production	
2.1.5.15	Hydraulic model (1D) **Comment:** The hydraulic routing is provided by MIKE Basin and MIKE 11.Likewise with water quality and sediment transport functionalities. The tools can be used off-line.	
2.1.5.16	Sediment transport in river networks and reservoir siltation	
2.1.5.17	Multiple routing methods (data dependent)	
2.1.5.18	Open (user defined) list of node types **Comment:** The node types are defined by the applied model tools (e.g. MIKE Basin/MIKE 11) and limited by their capabilities	15
2.1.7.1	Rainfall-runoff models (lumped, semi-distributed) **Comment:** RR model are like other models – and are available in a similar manner. The model tool configuring the rainfall-runoff model defines the type of process.	
2.1.7.2	Irrigation water demand estimation, crop production model	
2.1.7.3	Water quality model (DO/BOD, conservative, first order decay	

Req. No.	Comments	Issue ref
	Comments: It does not make sense to address WQ in groundwater in a lumped or even semi distributed model. It requires a fully distributed model to simulate groundwater WQ. Normally, base flow WQ is determined by calibration.	
2.1.7.5	Multiple evapotranspiration estimation methods **Comment:** FAO56 reference evapotranspiration is implemented	
2.1.6.2	Embedded calibration methods with error statistics. **Comment:** Model calibration is carried out in the respective model tools using whatever facilities are available. Additional comparison of model time series with observations and other time series are handled by the general time series analysis in Data Management.	
2.1.6.2 (1)	The system shall provide embedded calibration tools **Comment:** See above	
2.1.6.2 (2)	The calibration tools shall support calibration of any part of the basin under consideration **Comment:** The calibration related to the entire model as defined by the model tool.	
2.1.6.2 (3)	The system shall provide appropriate objective functions **Comment:** As calibration is carried out from within the model tool, the DSS will not provide this facility.	
2.1.6.2 (4)	The calibration tools shall enable the users to determine model parameter sets **Comment:** The calibration in the DSS does not know about model parameters, but is limited to time series analysis and comparison. The available calibration tools are determined by the model tools.	

4.10.1.3 MIKE SHE Studio

MIKE SHE Studio is a stand-alone, grid-based, semi-distributed hydrological modelling framework that includes a range of numerical methods for each hydrological process, with full water balance accounting for all hydrological processes. The hydrological processes and numerical methods can be combined depending on the requirements of the application and the availability of data.

The groundwater component of MIKE SHE Studio consists of a system of interconnected linear reservoirs, and hence emphasis has been put on making an efficient tool for distributed rainfall-runoff modelling and basin-wide water balance and water management studies.

MIKE SHE Studio is linked dynamically with MIKE 11, and sequentially with MIKE BASIN.

The engine part of MIKE SHE is a console application coded in FORTRAN. Input to the MIKE SHE Engine is provided through a single main input files, referred to as the ".SHE file". The .SHE file is an ASCII files adhering to the DHI Parameter File System (PFS) format specifications. This is depicted in Figure 4.5

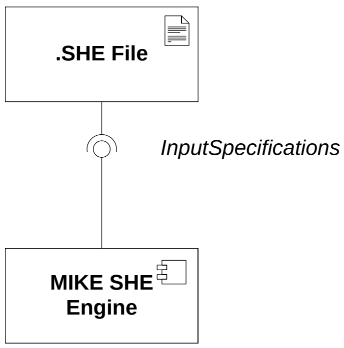

Figure 4.5 MIKE SHE Input Interface

The .SHE File references again other input files as depicted in Figure 4.6.

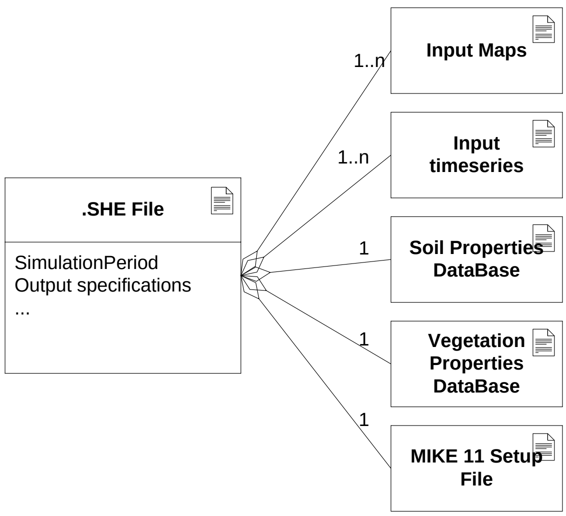

Figure 4.6 MIKE SHE Input Files

Some of the files listed in are ASCII formatted files others are binary.

Table 4.20 below lists the high level MIKE SHE functionalities.

Table 4.20 High-level MIKE SHE functionalities

High-level functionalities	Comments
Fully distributed, grid based, catchment model.	The grid based structure allows for an easy utilization of remote sensing data.
Overland flow	MIKE SHE Studio includes a simple, semi-distributed overland flow method for rainfall-runoff modelling. It also includes a 2D, diffusive wave, finite difference method for detailed runoff and flood modelling.
River Flow	Dynamically coupled to MIKE 11, including option for flooding to and from the river.
Unsaturated flow	Includes a 1D, multi-layer gravity flow model, a two-layer root zone model, as well as a Green and Ampt infiltration approach.
Evapotranspiration (ET)	MIKE SHE uses a vegetation based ET model approach that takes soil moisture content, vegetation development stage, as well as evaporation from interception, snow and ponded water into consideration.
Groundwater	MIKE SHE Studio includes a semi-distributed/lumped linear reservoir groundwater method suitable for basin-wide water balance and management.
Water Balance	Comprehensive and flexible water balance calculation of individual parts or the full hydrological cycle for any given time period and area.
Link to MIKE BASIN	Simulated catchment runoff can be transferred to MIKE BASIN in a sequential linkvia the NB DSS.

Table 4.21 below lists the functional/non-functional requirements related to MIKE 11.

Table 4.21 MIKE SHE requirement mapping

Req. No.	Comments	Issue ref
2.1.5.13	Lateral inflow, lateral catchments, floodplain representation	
2.1.5.7	Explicit groundwater representation and coupling	
2.1.5.6	Explicit mass budget, error statistics	
2.1.6.2	Embedded calibration methods with error statistics. **Comment:** Model calibration is carried out in the respective model tools using whatever facilities are available. Additional comparison of model time series with observations and other time series are handled by the general time series analysis in Data Management.	

Req. No.	Comments	Issue ref
2.1.6.2 (1)	The system shall provide embedded calibration tools **Comment:** See above	
2.1.6.2 (2)	The calibration tools shall support calibration of any part of the basin under consideration **Comment:** The calibration related to the entire model as defined by the model tool.	
2.1.6.2 (3)	The system shall provide appropriate objective functions **Comment:** As calibration is carried out from within the model tool, the DSS will not provide this facility.	
2.1.6.2 (4)	The calibration tools shall enable the users to determine model parameter sets **Comment:** The calibration in the DSS does not know about model parameters, but is limited to time series analysis and comparison. The available calibration tools are determined by the model tools.	

4.10.1.4 DHI.Optimization

DHI.Optimization is a library of generic methods to be used for optimization of any model, across differences in input/output file formats and programming languages.

The library has a well defined API, is extensible, and supports a range of optimization algorithms. To make the functionality available across differences in programming languages, the library has been developed in .NET and provided with a COM interface. The interface has been designed such that it is possible to define and solve an optimization problem with just a few lines of code.

Typically, once the specific optimization tasks have been identified, a UI tailored for these specific tasks is developed, and DHI.Optimization will provide the underlying algorithms. DHI.Optimization itself does not include any UI components (see section 4.19 Optimization Scenarios).

DHI.Optimization is used by the functional component Scenarios and the functional component Optimisation scenarios for performing the actual optimisations

Table 4.22 below lists the high-level DHI.Optimization functionalities.

Table 4.22 High-level DHI.Optimization functionalities.

High-level functionalities	Comments
API Documentation	The API is extensively documented, including examples on how to use it.
Optimization Algorithms	The list includes Shuffled Complex Evolution, Non-dominated sorting genetic algorithm II (Multi-objective) and Dynamically Dimensioned Search algorithm (planned)
Extensible	The API supports easy implementation of new algorithms, stop criteria, parameter constraint rules and objective functions.
Model independent	DHI.Optimization can communicate with models (input/output) either through Model APIs, or through input/output files.
Objective functions	DHI.Optimization includes a number of pre-defined objective functions. Additional ones can be added.

Table 4.23 below lists the functional/non-functional requirements related to DHI,Optimization.

Table 4.23 DHI.Optimization Requirement Mapping

Req. No.	Comments	Issue ref
2.1.6.4	Simulation based optimization	

4.10.1.5 MIKE Model Tool Couplings

MIKE11 and MIKE BASIN model tools come with built in catchment runoff models. The current versions of the model tools include the following options:

MIKE 11

The NAM rainfall-runoff model is a lumped conceptual (linear reservoir) model. NAM is delivered with the MIKE11 model tools and may be included in a MIKE11 model (if desired). NAM is coupled sequentially to MIKE11 implying that the rainfall-runoff simulation is executed for the full simulation period prior to the execution of the river hydraulics.

The MIKE SHE Studio semi-distributed conceptual/physically based model. MIKE SHE Studio comes as an integral part of the MIKE11 Tool and may be used for more sophisticated catchment runoff modelling than provided by the lumped NAM model. The MIKE SHE studio is dynamically linked to MIKE11's hydraulic engine implying that simulation results are exchanged on a time-step by time-step basis.

MIKE BASIN

The NAM rainfall-runoff model is delivered with the MIKE BASIN model. As for MIKE11 the coupling is sequential.

MIKE BASIN also comes with its own groundwater model (MIKE BASIN GW). MIKE BASIN GW is dynamically coupled with the river network simulation engine implying that simulations results are exchanged on a time-step by time-step basis (typically daily to monthly time-steps). MB GW is a lumped, conceptual linear reservoir routing model. Input for MB GW is recharge (i.e. MB GW does not include a recharge model). Recharge may be calculated using different recharge model – for instance recharge may be calculated using MIKE SHE studio.

A sequential coupling is basically just a transfer of data from one model to another. As such all the catchment runoff models that are mentioned above may be coupled sequentially to MIKE11 or MIKE BASIN. The dynamic couplings however are much more complex and tailored for the specific model tool. The coupling specifications reside in the Model Tools and are brought into the NB DSS through the model adapters.

Sequential, couplings or simply direct specification of catchment runoff (e.g. specific runoff) may be sufficient for many purposes. In catchments where conjunctive use of groundwater and surface water is important dynamic couplings are often necessary.

An overview of the couplings that is currently available for MIKE11 and MIKE BASIN is shown in Table 4.24.

Table 4.24 Overview of built-in couplings between network models and catchment run-off models

	Catchment Runoff Models		
	MIKE SHE Studio	**NAM**	**MB GW**
MIKE BASIN	*none*	*Sequential*	*Dynamic*
MIKE 11	*Dynamic*	*Sequential*	*none*

In all cases the combined model setups will be controlled by one model tool and is as such regarded as a single model in the NB DSS context.

4.10.2 Model Adapters

Integration of Model tools and model setups with the DSS Front-end is accomplished through the use of Model tool adapters. An adapter resides between the DSS Front-end and the Model tool as shown in Figure 4.7.

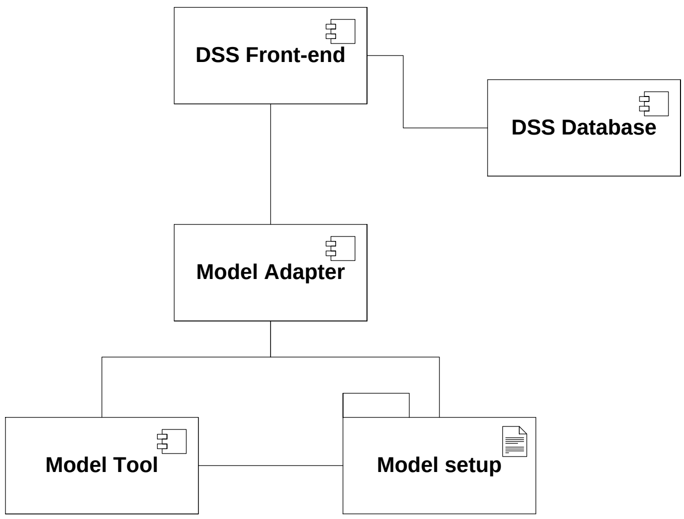

Figure 4.7 Model adapters (UML)

In order to provide a standardised way of integrating Model tools with the DSS Front-end, the adapter must adhere to a standardized interface towards the DSS Front-end. This interface defines the kind of information that can be communicated between the model tool and the DSS Front-end.

The Consultant will develop adapters for the Model Tools that are supplied as part of the Contract /2/. Adapters for any other Model Tool may be developed by the NBI, by other Consultants or by any other for whom it may be relevant.

A model adapter needs to be able to parse and prepare model setups for the model tools. This implies that an adapter is specific to a specific model tool.

In fact, two types of adapters are needed: A configuration adapter and a run-time adapter.

- The configuration adapter is used for registration of a model setup in the DSS Front-end. Registration of the model setup is required to port details of the setup into the DSS database including hydro objects and boundary conditions (input time series). During registration, the full model setup will be parsed and hydro objects, simulation period, initial conditions, input time series and output will be identified and stored in the database

  In addition, the configuration adapter informs the NB DSS about parameters within the model tool that can be varied in an optimization simulation. These parameters are named optimization parameters.

- The run-time adapter is required to control model simulations using the Model tool. The purpose of the run time adapter is to prepare a model setup for simulation, execute the Model tool and finally extract selected simulation outputs to be stored in the DSS Database. This is needed for all types of model simulation also those that may include ensemble and optimization simulations. .

  In addition, the run-time adapter shall be able to modify the model setup with defined optimisation parameters.

High-level functionality related to the functional components comprises the following:

- Parse a model setup
- Re-create a model setup
- Return a list of optimization parameters
- Initiate a model simulation
- Set optimisation parameters in the model setup

Requirements related to the overall architecture of the NB DDS for the functional component are established in Table 4.25

Table 4.25 Functional component: Model tools (adapters) - derived requirements

Req. No.	Comments	Issue ref
X.33	The configuration adapter shall be able to parse a model setup and extract hydro object, name or location of input time series, name or location of output time series, name or location of initial conditions.	
X.34	The run-time adapter shall be able to re-create a model setup using the full model setup stored in the database but with different input time series, different output and different initial conditions	
X.35	The run-time adapter shall be able to extract the time series from a simulation run that shall be inserted in the database	
X.36	The run-time adapter shall be able to modify a model setup in order to change the simulation period.	
X.37	The configuration adapter shall be able to inform the NB DSS about what parameters (optimization parameters) in a model setup that can be varied in an optimization simulation.	
X.38	The run-time adapter shall be able to set the optimisation parameters according to rules defined by the DSS front-end.	

4.11 Functional Component: Indicators

The Performance Indicators functional component provides functionality for defining and creating indicators to be used, for example by MCA functional components (see Section 4.12 and Section 4.13). The Indicators functionality shall be accessible from the NB DSS to calculate indicators whenever needed. As such it shall not be available only from within the MCA component but as a general functionality within the NB DSS.

Performance Indicators describe certain effects of an evaluated scenario on its environment as single scalar values and are, in the context of the NBI-DSS, derived from observed or (typically) modelled data in the DSS. For example, if the scenario is "construction of a reservoir at location *X'*, then possible performance indicators could be "the number of days the flow below the dam is above 350 cms during the dry season", or "the reduction in maximum stage during the flood season". As such, Performance Indicators are calculated as statistics (or as combinations of statistics) of the associated time series, such as "modelled river water depth" or "modelled stream flow".

The Performance Indicators functional component will provide the users logged into a Study (Section 4.15) with an interface to

- Configure the metadata for a performance indicator in the database, such as a location and an identifier for the indicator (for example "reduction in maximum stage during the flood season at location *X*")

- Define the time series, the statistic method, the evaluation period, and additional calculations required to compute the performance indicator (for example, "maximum[base scenario stage a location *X* in July] – maximum[modelled scenario *n* stage a location *X* in July])"

The scenario(s) for which the Performance Indicators are computed will be defined in the CBA and MCA functional components.

High-level functionality related to the functional components comprises the following:

- Configure a Performance Indicator

- Store Performance Indicators to the database associated with the relevant study.

- Import Performance Indicators

- Export Performance Indicators

Table 4.26 below maps the functional component with the requirements from the ToR /2/.

Table 4.26 Functional component: Indicators – requirement mapping

Req. No.	Comments	Issue ref
2.1.5.12	Tools for converting model outputs into desired criteria using user-defined methods **Comment:** Indicators are the first step in this conversion process	14
2.1.8	Multi-Criteria-Analysis (MCA) Tools	
2.1.8.3	Automatic model linkage **Comment:** This requirement for the MCA will be satisfied through the use of performance indicators that are derived from data computed by models for user-defined scenarios	
2.1.5.19	Yield/reliability analysis for reservoirs and catchments	

4.12 Functional Component: MCA

The Multi-Criteria Analysis (MCA) functional component provides functionality for configuring and running multi-criteria analyses.

Inputs to an MCA are:

- Performance indicators (see Section 4.11) computed from data in the DSS (internal indicators),
- Data tables or arithmetic processes to convert these performance indicators to comparable performance criteria (internal criteria),
- Performance criteria that were developed outside the DSS (external criteria),
- Identifiers for one or more stakeholders or stakeholder groups, and criterion weights for each stakeholder - performance criteria combination.

The MCA functional component will allow users logged into a study to create a decision matrix for each stakeholder or stakeholder group, similar to what is shown in Figure 4.8.

Matrix for Stakeholder *k*	Internal Criterion 1 (*c1* – derived from internal indicator *i1*)	Internal Criterion 2 (*c2*– derived from internal indicator *i2*)	External Criterion 3 (*c3*– entered directly)	Criterion *n* (*cn*)	Scenaria Rank
Scenario 1 (*s1*)	$c1_{s1} \cdot \text{Weight}_{k,c1} = f(i1_{s1}) \cdot \text{Weight}_{k,c1}$	$c2_{s1} \cdot \text{Weight}_{k,c2} = f(i2_{s1}) \cdot \text{Weight}_{k,c2}$	$c3_{s1} \cdot \text{Weight}_{k,c3}$	$cn_{s1} \cdot \text{Weight}_{k,cn}$	*Sum of criteria*
Scenario 2 (*s2*)	$c1_{s2} \cdot \text{Weight}_{k,c1} = f(i1_{s2}) \cdot \text{Weight}_{k,c1}$	$c2_{s2} \cdot \text{Weight}_{k,c2} = f(i2_{s2}) \cdot \text{Weight}_{k,c2}$	$c3_{s2} \cdot \text{Weight}_{k,c3}$	$cn_{s2} \cdot \text{Weight}_{k,cn}$	*Sum of criteria*
Scenario 3 (*s3*)	$c1_{s3} \cdot \text{Weight}_{k,c1} = f(i1_{s3}) \cdot \text{Weight}_{k,c1}$	$c2_{s3} \cdot \text{Weight}_{k,c2} = f(i1_{s3}) \cdot \text{Weight}_{k,c2}$	$c3_{s3} \cdot \text{Weight}_{k,c3}$	$cn_{s3} \cdot \text{Weight}_{k,cn}$	*Sum of criteria*
Scenario *m* (*sm*)	$c1_{sm} \cdot \text{Weight}_{k,c1} = f(i1_{sm}) \cdot \text{Weight}_{k,c1}$	$c2_{sm} \cdot \text{Weight}_{k,c2} = f(i1_{sm}) \cdot \text{Weight}_{k,c2}$	$c3_{sm} \cdot \text{Weight}_{k,c3}$	$Cn_{sm} \cdot \text{Weight}_{k,cn}$	*Sum of criteria*

Figure 4.8 Sample MCA matrix

Once the performance criteria are defined and linked to the appropriate scenarios, weighted criterion values are computed (from internal performance indicators if necessary). Changing the stakeholder weights will allow for the analysis of their effects on the ranking of the included scenarios and hence the final outcome.

The MCA functional component also includes an interface to define a Reference Point analysis and to visualize and analyze solutions between UTOPIA and NADIR outcomes.

High-level functionality related to the functional component comprises the following:

- Configure the components of the MCA functional component
- Store the MCA configuration to the database
- Perform the MCA by setting criteria weights and scenarios
- Visualize the results of the MCA analysis
- Export the MCA configuration and results
- Import the MCA configuration and results

Table 4.27 maps the functional component with the requirements from the ToR /2/.

Table 4.27 Functional component: MCA – requirement mapping

Req. No.	Comments	Issue ref
2.1.5.12	Tools for converting model outputs into desired criteria using user-defined methods **Comment:** The MCA functional component is the second step in this conversion process	14
2.1.8.1	Multiple MCA methods	
2.1.8.2	User defined open list of criteria	
2.1.8.3	Automatic model linkage **Comment:** This requirement for the MCA will be satisfied through the use of performance indicators that are derived from data computed by models for user-defined scenarios	

4.13 Functional Component: CBA

The CBA functional component provides functionality for weighing total expected costs against total expected benefits for one or multiple scenarios in order to choose the best or most profitable option.

CBA is a flavour of MCA, where the performance criteria used are normalized to monetary units.

The CBA functional component resembles the MCA functional component (Section 4.12) but will include interfaces to convert future expected costs and benefits to present value amounts and to define, access and aggregate initial and recurring costs and revenues for all infrastructure components in an area of interest (such as existing and proposed reservoirs, irrigation developments, etc.).

High-level functionality related to the functional component comprises the following:

- Define costs for (existing or considered) infrastructure components
- Define and configure the computation of costs and benefits resulting from the realization of a infrastructure component
- Define variables to perform present value calculations
- Visualize the results of the CBA analysis
- Export the CBA configuration and results
- Import the CBA configuration and results

Table 4.28 maps the functional component with the requirements from the ToR /2/.

Table 4.28 Functional component: CBA – requirement mapping

Req. No.	Comments	Issue ref
2.1.5.11	Economic analysis of scenarios (CBA)	

4.14 Functional Component: Meta data

The meta-data functional component provides functionality for creating, editing, querying and displaying meta-data.

The discussions, especially during the Inception Workshop, on this matter have lead to the following observations.

Meta data is typically referred to as "data about data", i.e. information which is used to inform the user of the data he/she is working with. A further analysis into this reveals a grey zone as information acting as meta data in some cases is "real" data in others (see for instance http://en.wikipedia.org/wiki/Metadata).

Likewise with systems for holding meta data – they can be very specific or be general systems linking to anything.

The consultant sees meta-data for the NB DSS as attributes of entities in the system, which is not part of the explicit data model. The meta data therefore are kept "on the side" in relation to the actual data itself which take part in the processes. The reason for this is that meta data in most cases is additional information serving a purpose to the user in assessing the type, origin, quality etc of the data he/she is working with and using for decisions, while the explicit data model (i.e. the table structure and data access objects) contains the data. The explicit data model should be stable over time and serve the functionality of the system.

- The meta-data could also describe a structure for user privileges and access for uses like data sharing.

- Meta-data could be used to establish traceability with respect to use of data, e.g. by having the meta-data describing the changes that have happened to the data. In this respect meta-data becomes part of a possible tools integration solution by keeping a change log on how data has been modified through a series of tools invocations. An example of such a series of tool invocations could be:

    1. Time series TS1 is created/imported in the system. Apart for the time series data additional meta-data describing it and the process of entering it can be added, e.g. name of user, time of entry, source of origin.

    2. A user edits a spike in the data and updates TS1. The edit process adds meta information like time, user id and perhaps prompts for a descriptive text from the user about the change.

3. A user re-samples the data into a new time interval and creates new time series, TS2. meta data describing the time, the user, the process and the parameters of the process and a link to the input time series is created for the output time series

4. TS2 is used by a scenario run which creates an output time series, TS3 (or more like it). When storing TS3 it is accompanied by meta data linking TS2 to the scenario and perhaps also to the input time series.

5. The output time series is used by a decision making tool for producing a result (number or report). The output has meta-data referring to the TS3 as well as the tool configuration which is assumed is in the database.

- The aim is to have a kind of meta data browser in the GUI which can display the meta information for a selected item and also point to related inputs used in creation of the item. Meta data is then available for the linked items and this way the user can "step backwards" in the processing via the meta-data-browser. A forward stepping mechanism presented in a similar way would be useful.Automatic/manual - the system shall allow creation of meta-data by processes as well as allow the user to add/supply meta-data manually (hopefully in a general fashion)

- Simple data types - only types like string, text, number, date and time, guid, enumerations (for classification purposes) are expected to be part of the meta data. A special one like enumeration could be necessary to allow selection of values from a predefined list (drop-down box)

- Limited/well defined list of meta-data keys - the meta data is a list of key-value pairs. The keys are taken from a published list of defined keys in the system in order to standardize the meta-data descriptions across installations. NBI as an organisation would define the set of accepted meta data keys.

- Language - the meta data keys should be language sensitive (e.g. English and French), i.e. the list of keys shall include definitions in supported languages and the logic shall use the application language to display the proper versions.

- Editing parts of the chain. If a user edits TS1 above after it has been used by the process leading to the output in 5) then the chain of information is broken as the same results can no longer be derived from the original data using the described processes. If the chain of events is part of testing/setting up the processes then this is ok, but if the output is used as part of a decision it is not desirable. The system should be able to flag an item if it has been used - and prompt the user before altering data (perhaps offer saving as a new separate item instead of overwriting). It is the user which decides if the chain is to be broken.

  The system could support this by having a indicators (flag, severity) of the process so that the user may know if the data has been used in the decision process or not, and whether it is OK to overwrite or not. This will require all processes to be able to be run in the context of severity in order to create severe/not severe meta-data.

High-level functionality related to meta-data comprises the following

- Associate meta data keys and values with data entities
- Display and edit the meta data

Table 4.29 and Table 4.30 lists the identified and derived requirements.

Table 4.29 Functional component: Meta-data - requirement mapping

Req. No.	Comments	Issue ref
2.1.2.3	Standard META data model ***Clarification*** *Metadata is the description of the data in the database – who created the data? When? What was its purpose? Etc. so that one might have an idea of what is appropriate to do with the data.* *DHI should provide a data model that allows the user to define the metadata structure for given data. Some metadata may be system defined, other may be user defined. The structure of the metadata or the actual metadata fields will be determined in cooperation with NBI and WP2.*	
2.1.2.3 (1)	Standard META data model: General requirements	
2.1.2.3 (2)	Standard META data model: Main data types	
2.1.2.3 (3)	Standard META data model: Authorization and user access	
2.1.2.3 (4)	Standard META data model: Business process	

Table 4.30 Functional component: Meta- data - derived requirements

Req. No.	Comments	Issue ref
X.39	The meta data should hold a change log covering all data entities	
X.40	Each entry in the change log shall include either the input parameters used during the change operation, a reference to the operation that changed the data or descriptive text entered by the user	
X.41	It shall be possible from the GUI to browse the change log	
X.42	The keys used in the meta data model shall be limited to a positive list of possible keys and their types.	
X.43	Meta data keywords shall be translatable with respect to the language(s) of the application	

4.15 Functional Component: Studies

The Studies functional component provides functionality for creating studies and managing studies. A study is a logical partitioning of all the data in the database that shares the same security model.

Studies are logical groupings of data within the database:

3. All data within a study shares the same access control lists.

4. All data within a study relates to the work a dedicated work group undertakes.

An access control list defines how a specific user or user group can interact with the data within a study, i.e. defines who that can read and write data and who that can only read data.

A special study area is the Global data area. Data within this area are either:

- System data, e.g., data item and unit definitions

- Domain data that should be available in all Enterprise installations. This could comprise base lines hydro objects, e.g. list of reservoirs in operation, base line models, measurements etc.

Users - when logging on to the system – can optionally select the study they want to log on to.

- If logging on to a study, the user will have access to – according to the access control list for the study belongs to – all data within the study and all the data belonging to the global area. Typically, access to global data is only granted in read-only mode, this however depends on the role of the user, see Section 2.3. The user will not have access to data in studies other than the one logged on to.

- If not logging on to a study, the user will be given access to all the data belonging to the global area. Typically, access to global data is only granted in read-only mode, this however depends on the role of the user, see Section 2.3

The installations of Corporate edition of the NB DSS shall have identical global data areas. This requires - in addition to a technical synchronization solution - also backing by policies and a central managing organization.

The architecture supporting synchronization of the global data area is described in /5/.

A study can be exported in part or in whole and imported in another DSS installation. This provides for supporting disconnected work teams.

High-level functionality related to the functional components comprises the following:

- Create a study

- Associate users with a study

- Set user access rights
- Export a study
- Import a study

Note the Studies functional component has not been directly mentioned in the TOR, but comes out of the use case analysis. Specifically the following use case cases from /3/ deals with tables:

- User story 2, use case Create and Configure Study
- User story 3, use case Define Study

Table 4.31 lists suggested requirements derived from the use cases.

Table 4.31 Functional component: Studies - derived requirements

Req. No.	Comments	Issue ref
X.44	It shall be possible synchronize global data	
X.45	It shall be possible to share a study across installations	
X.46	If shall be possible to import or export a study in full or parts	
X.47	It shall be possible to associate DSS users with a study and in accordance with the DSS user profile model	
X.81	It shall be possible to update a study, i.e. synchronize through the import mechanism or through the standard PostgreSQL synchronization mechanisms	26

4.16 Functional Component: Reporting

The Reporting functional component provides functionality creating, publishing and storing reports visualizing data from the database, primarily time series, GIS data, tables, hydro objects and text.

The requirements in the ToR /2/ correspond to reporting functionality provided by most standard database systems[8]. I.e. functionality for

- Defining a report template by
    - Choosing a predefined query
    - Specifying fields to display
    - Specifying selection criteria

[8] An example could be Microsoft Access

- Specifying aggregation

- Saving a report template
- Editing a report template (potentially saving it as a new template)
- Running a report (execute the underlying query)

A report consists of a tabular view showing the retrieved data.

Reporting in Microsoft Access and similar system is based on SQL queries; i.e. the user defines an SQL string that is used to retrieve the data that shall go into the report. This could prove a problematic approach to take for NB DSS reports. Reasons being:

- Spatial queries are not simple to establish with respect to a standard query language like SQL as it uses an extension to the SQL language. The underlying data involves geometry specifications that in no practical way can be entered manually; this would require that the user should enter a WKT text string which often is several thousand bytes long.
- A query may consist of a join of many tables – and could potentially demand too many processing resources from the database in order to complete. Hence, free querying may jeopardise the performance of the database
- The clarifications obtained during the inception phase indicate that expert users shall be able to create new queries. This has additional ramifications with respect to security because free querying will make it possible to circumvent the data access control lists. This is clearly not acceptable why free querying only can be implemented if backed by some kind of row-level security – which PostgreSQL does not support.

The conclusion of this is that the definition of the query that underlies the report needs to be based on an intermediate layer separating the user queries from the database. This could be database views, stored procedures or maybe .NET assemblies. Decision on this should be based on a detailed analysis of the reporting requirements.

Table 4.32 maps the functional component with the requirements from the ToR /2/ and Table 4.33 establishes derived requirements.

Table 4.32 Functional component - Reporting requirement mapping

Req. No.	Comments	Issue ref
2.1.2.4	User defined report generation Comment: A report is a tabular view of data stored in the data-base.	
2.1.2.4 (1)	Report configuration ***Clarification*** It was not envisioned that the user would be creating queries. Rather the user chooses to prepare a report, and is presented with available fields from which to choose for column headings, etc, and can then save the report configuration for later updating (for a given type of report, of which there may be many).	10
2.1.2.4 (2)	IMS shall enable the user to customize predefined re-port templates **Comment:** A report template is a query and some formatting rules (fonts and colours).	
2.1.2.4 (3)	All reports shall be stored and managed in the IMS.	
2.1.2.4 (4)	Export of all reports to standard applications such as office software packages/formats shall be possible.	11

Table 4.33 Functional component - Reporting derived requirements

Req. No.	Comments	Issue ref
X.48	Expert users shall be able to create queries	
X.49	A report must be able to include a map, i.e. the selected GIS data is formatted and displayed as a map	
X.50	a report must be able to include a chart, i.e. the selected data is formatted and displayed as a chart	
X.51	Generated reports shall be saved in the database	
X.52	It shall be possible to associate report templates with studies	
X.53	It shall be possible to associate report templates with global data	
X.54	It shall not be possible to circumvent the security model when creating report queries	

5 EXTERNAL INTERFACES

5.1 User Interfaces

The DSS Front-end will be a Windows application. It shall have an IDE-style[9] user interface (UI) where all windows reside under a single parent window. The IDE-style UI comprises for example dockable and collapsible child windows, tabbed windows and splitters for resizing of child windows.

All controls comprising the NB DSS will be categorized as either Explorer Controls or Data View Controls to maintain a transparent and easily recognizable UI structure. These controls are further grouped into modules, and hence, from a UI perspective, a DSS Module will consist of a collection of Explorer- and Data View Controls.

Explorer Controls will provide access to the module functionality. When the user wishes to perform a task using a specific module, the task will be initiated in an Explorer Control that belongs to the specific module. At this point, the Data View(s) tailored for the initiated task will appear and allow the user to complete the task.

Example: A module is responsible for querying and retrieving data from the database, as well as for editing, visualising and analysing the retrieved data. In that case, the Explorer could contain functionality to construct a query and list the retrieved data, as well as presenting the user to a range of options for visualizing, editing and analyzing the data. Once selected, the appropriate Data Views (e.g. Charts, tables and maps) will open and allow user to complete the selected task.

To avoid lengthy context menus and complex pop-up dialogs, and to maintain consistent structure across Modules and Controls, the DSS Shell shall offer a set of standard controls that can interface with the module controls. These include a Property Control and a Toolbox Control.

When modules are loaded into the system, the controls shall automatically be grouped and displayed by type. This is illustrated in Figure 5.1.

[9] Integrated Development Interface: Similar to Multiple Document Interfaces (MDI), except that the child windows in an IDE-style interface can be enhanced with additional functionality, which is not ordinarily available in MDI-style interfaces.

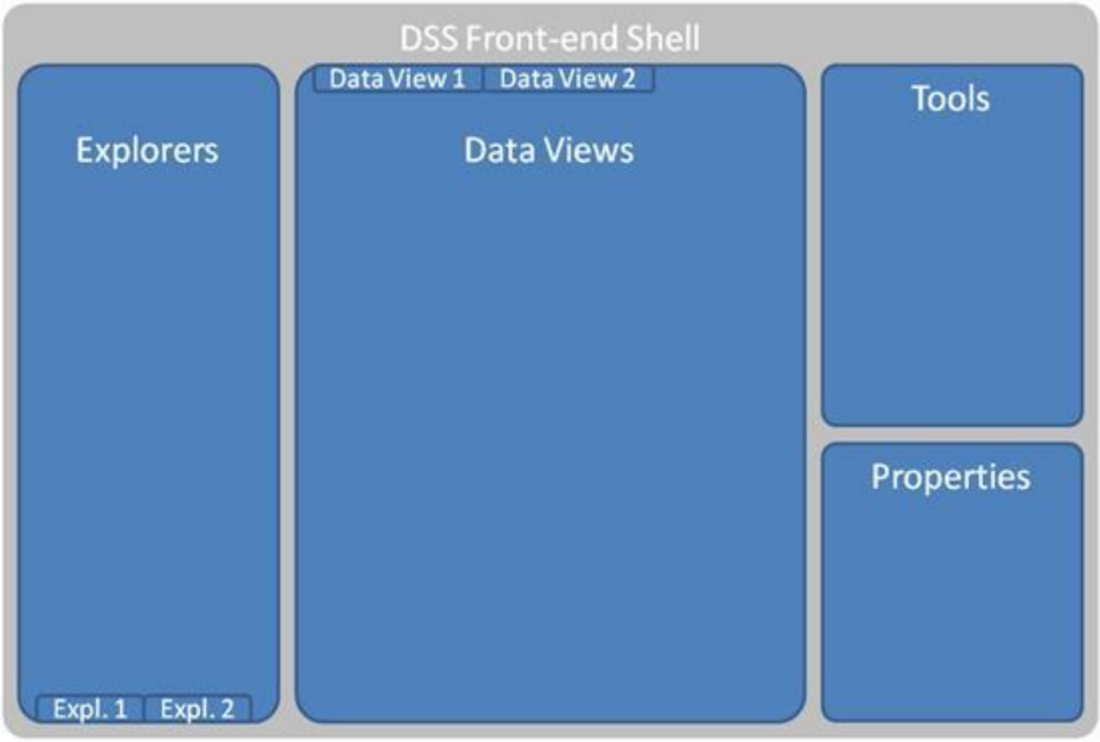

Figure 5.1 Conceptual description of the DSS UI structure.

The task of the Property Control is to expose the properties of the "object(s)" that are currently selected in the active control. "Objects" can be any "selectable" UI element residing in either an Explorer or Data View control, e.g. a line in a chart, a GIS layer, or a column in a table. Object(s) shall be selected by clicking the appropriate object(s) in the Explorers and Data Views. It is up to a given Explorer or Data View to determine which objects are selectable, and which properties are exposed in the Property Control for each selectable object type. It shall, at any time, be clear to the user which object(s) are currently selected.

The Toolbox Control shall, at any time, display a list of tools that can be applied to the currently selected object(s). This means that if a time series is selected, the toolbox shall list the available time series tools, if a GIS layer is selected, it will list the available GIS tools that can operate on a GIS layer etc. When a tool is selected is shall be configured in the Property Control. It shall be possible to store the settings for a single tool. Moreover, it shall be possible to link tools, such that the output of one tool provides the input to the next tool(s) in a sequence, and to store a sequence of configured tools for later use.

The Toolbox shall make it possible to easily transfer output from one tool to the next in a sequence of tool executions. This could happen directly through sequencing of tool executions – similar to the UNIX pipe concept – or through the use of the Windows Clipboard. A tool can display output in a Window – e.g. as a new time series chart or a value in a table - and it can persist the data in the database. In case of the latter the user shall include an identification of the data such that it later on can be retrieved and used as input in another tool.

It shall be possible for the user to rearrange the controls, and optionally to save the preferred configuration.

The NB DSS will serve as a tool for configuration as well as operation. This means that it will be an application targeted at multiple user profiles – varying from expert users setting up and configuring DSS solutions to end-users that need to use the application for daily operations. It shall be possible to configure the system such that the appearance (i.e. the visibility and position and configuration of specific controls) adapts to the need of specific user profiles and tasks.

The DSS Shell shall make a Message Framework available for the individual controls, such that Error Messages, Warnings, Information, and Confirmation dialogs are presented in a consistent way across the different modules and controls.

Moreover, the DSS Shell shall make an Online-help framework available for the individual controls and modules, such that the context sensitive help functionality is handled consistently.

The user interface shall be expandable, such that, if the need should arise after the release of the system, it will be possible to extend the system with new Explorers, Data Views and Tools that are tailored for specific tasks, without having to access the DSS Front-end source code.

All controls, as well as the DSS framework itself will be localizable and French and English shall be supported. It shall be possible to extend the list of supported languages with minimum effort.

Table 5.1 User Interface requirement mapping

Req. No.	Comments	Issue ref
2.1.1	Graphical User Interface	
2.1.1.1	Interactive, fully menu driven graphical, hyperlinked	
2.1.1.2	Multi-language support (English, French)	
2.1.1.3	On-line, context-sensitive help facility with an online hierarchical & cross-linked help system in HTML ***Clarification*** *HTML is requested to make sure that the content of the help system can include hyperlinks to other parts of the documentation.* *When error messages are presented, it will be helpful for the help system to present help on the warning or error message that is present.* *If DHI wishes to propose alternatives, it should do so and provide an explanation of the advantages or reasons for the alternative approach.*	3
2.1.1.1 (1)	The GUI shall be as powerful and interactive as possible for creating, locating and inter-connecting the various components.	

Req. No.	Comments	Issue ref
2.1.1.1 (2)	The GUI shall allow data entry by users with features to add, modify and delete data according to predefined user privileges	
2.1.1.1 (3)	The GUI shall have a menu structure which is always evident to the user.	
2.1.1.4 (1)	The GUI shall be consistent with regard to screen lay-outs, messages, text and graphic positions etc. to ease learning of the ...	
2.1.1.4 (2)	The GUI shall have readable text that is easy to under-stand, even for non-native speakers.	
2.1.1.4 (3)	The GUI shall have text supported by meaningful graphical hints to ease understanding.	
2.1.1.4 (4)	The GUI shall have the ability to make adaptations due to regional differences in line with the platform and in-frastructure... ***Clarification*** *This requirement is to avoid having the gui represent dates and formats in a way that will negatively impact the use of the system, due to inconsistent formats be-tween the gui and the user expectation (like you get sometimes with U.S. based software).* **Comments:** The UI will follow the the user's regional settings (for-matting of date, time, numbers etc)	4
2.1.1.5	Error/action messages for wrong entries ***Clarification*** *This is intended to provide a reasonable level of secu-rity against accidental changes to the database. DHI should propose a balance that is reasonable.*	
2.1.1.6	Web browser support (Optional) **Comment:** Not applicable	

Table 5.2 Derived Requirements – User interface

Req. No.	Comments	Issue ref
X.55	The DSS Front-end shall have an IDE style interface, where all modules and controls reside in a single par-ent window.	
X.56	The Shell shall make a set of standard controls avail-able for the module to help ensure a similar layout, structure and workflow for the individual modules.	
X.57	The appearance of the user interface shall be config-urable such that it can adapt to the needs of a specific	

Req. No.	Comments	Issue ref
	user profile or tasks.	
X.58	A Toolbox control shall maintain a list of the tools that are available to the user at any time during the execution of a task.	
X.59	A Property Control shall be used by all modules to expose properties and for configuration purposes in order to ensure a consistent layout and workflow across the different modules.	
X.60	A standard Message framework shall be made available to the modules to ensure a consistent presentation of errors, warnings and information.	
X.61	The user interface shall be expandable such that it will be possible to add new Explorers, Data Views and Tools without accessing the DSS Front-end source code.	
X.82	It shall be easy to use output from one tool execution as input in another tool	
X.83	It shall be possible to persist tools output in the database	
X.84	It shall be possible to retrieve persisted tools output from the database and use it as input in another tool execution.	

5.2 Software Interfaces

The NB DSS utilises a number of external software components. These can – as briefly described in Section 2.7– be classified as follows:

- Database related components
- Programming related components

The following sections describe the components in more details.

5.2.1 Database Related Components

The database related components are:

- PostgreSQL – the relational database used in the NB DSS system.
- PostGIS – extension to PostgreSQL that makes it possible to maintain and process GIS data managed in the PostgreSQL database.
- NPGSQL – ADO.NET driver for PostgreSQL.
- NHibernate – An object-relation mapper.

These components interact with each other in order to connect the DSS front-end with the database. This is depicted in Figure 5.2 below showing the information flow when issuing a database query through the NB DSS.

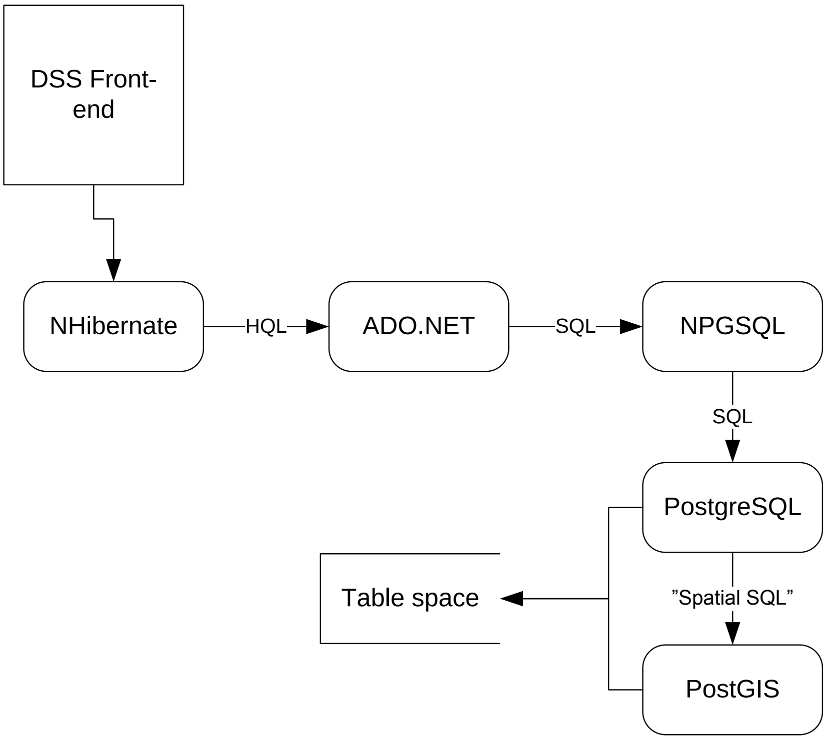

Figure 5.2 Query based data flow (Gane-Sarson)

The following sections will in more details describe each of these components.

5.2.1.1 PostgreSQL

PostgreSQL is an object-relational database management system. It is released under a Berkeley Software Distribution (BSD)-style license and is thus free and open source software. As with many other open source programs, PostgreSQL is not controlled by any single company, but has a global community of developers and companies to develop it. The development of PostgreSQL dates back to the early 1980s.

The main features of PostgreSQL comprise:

- Stored procedures can be written in high-level languages like Python, C++ and Java
- Indexes – based both on column values and expressions. Partial indexes are also supported
- Triggers can also be coded in high-level languages
- Multi-version concurrency control which provides individual user snapshots of the database
- Updatable views
- A wide variety of data types
- User defined objects

- Inheritance – tables can inherit characteristics from a parent table. This can be used to implement table partitioning

The PostgreSQL database is described in much more details on the PostgreSQL home page http://www.postgresql.org/.

The NBI DSS will use PostgreSQL to store all data as shown in Figure 5.3. The input to PostgreSQL is SQL statements and the output are result sets from the executed SQL statements.

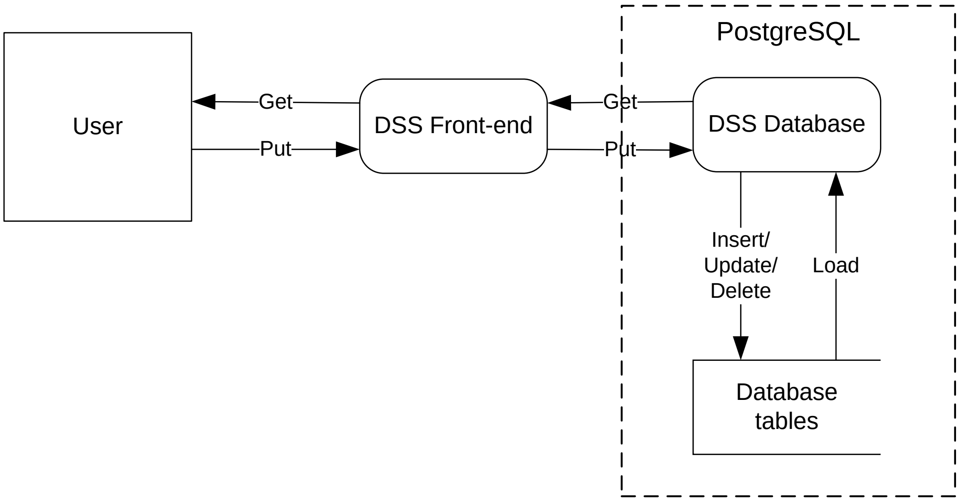

Figure 5.3 Database dataflow

Table 5.3 maps the PostgreSQL component to the relevant requirements from the TOR.

Table 5.3 PostgreSQL requirement mapping

Req. No.	Comments	Issue ref
2.1.2	IMS: Database Management system and Communication	
2.1.2 (1)	IMS: - any time period	6
2.1.2 (2)	IMS: - gregorian calendar	
2.2.1.9	Embedded backup tools and backup strategy	23
2.1.2.1	Standard RDMS with database level application clustering feature	

Req. No.	Comments	Issue ref
2.1.2.1 (1)	The DBMS shall store all relevant basin data as well as all data to manage and run the NB DSS ***Clarification*** *Include standard image formats that Postgres supports, as well as other postGIS formats. This may involve using some work-arounds for storage and processing, with anticipation that the DB capability in Postgres/PostGIS will be enhanced in the next few years. It is understood that Postgres complies with this requirement. Further clarification on item g – remove requirement g.*	
2.1.2.1 (2)	The DBMS shall support data definition and manipulation according to state of the art standards for data base managements sys...	
2.1.2.1 (3)	Database corruption and losses ***Clarification*** *NBI understands that Postgres complies with these requirements. Item d refers to record locking and transaction processing that ensures data integrity for multi-user access. We want to make sure that "undo" feature. Postgres uses "delta compression".* *These features of the database are partly specified to assure that in the future NBI is not limited by the initial selection of the database.* *Remember that the professional editions don't include any synchronization, only import and export.* *If there are contradictions between requirements, DHI should notify.*	
2.1.2.1 (4)	DBMS data type requirements	
2.1.2.1 (5)	DBMS geo-referenced spatial information requirements	
2.1.2.1 (6)	DBMS shall efficiently operate on voluminous time series data	
2.1.2.1 (7)	IMS access to DBMS - interfaces to different programming languages and other third-party applications.	
2.1.2.1 (8)	DBMS synchronization requirements	
2.1.2.2	Data model, entity relationship model description explicit	
2.1.2.6	Support various database functionalities	

Req. No.	Comments	Issue ref
2.2	Non-functional requirements **Comments** The RDBMS shall be used in both the Professional and Corporate editions	
2.2.1.8	Implementation platform The RDBMS shall run on Linux	

5.2.1.2 PostGIS

PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium. The first version of PostGIS was released in 2001.

The main features of PostGIS comprise:

- Geometry types for points, linestrings, polygons, multipoints, multilinestrings, multipolygons and geometrycollections.
- Spatial predicates for determining the interactions of geometries using the 3x3 Egenhofer matrix (provided by the GEOS software library).
- Spatial operators for determining geospatial measurements like area, distance, length and perimeter.
- Spatial operators for determining geospatial set operations, like union, difference, symmetric difference and buffers (provided by GEOS).
- R-tree-over-GiST (Generalised Search Tree) spatial indexes for high speed spatial querying.
- Index selectivity support, to provide high performance query plans for mixed spatial/non-spatial queries.
- Raster (gridded data) support is under development through the WKTRaster project. The goal of WKTRaster is to implement a RASTER type consistent with the existent GEOMETRY type in PostGIS and to offer a single set of overlay SQL functions (like ST_Intersects) operating seamlessly on vector and raster data.

PostGIS is described in much more details on the PostGIS home page http://postgis.refractions.net

The NB DSS uses PostGIS to maintain spatial data in the PostgreSQL database by leveraging its geometry and spatial reference system functionality[10]. Part of the Geo-

[10] Ability to change between projection systems

processing functionally provided by PostGIS for performing spatial unions, intersects etc may be used to meet the functional requirements for GIS processing. Other options such as using functionality provided by the ThinkGeo component (see section 0) or custom coding it might be utilised in the implementation of the GIS functional component. This will be decided upon during the detailed analysis and design stages of the development cycles.

Table 5.4 maps the PostGIS component to the relevant requirements from the TOR.

Table 5.4 PostGIS requirement mapping

Req. No.	Comments	Issue ref
2.1.2.1 (1)	Standard RDBMS – GIS layers and remote Sensing images	
2.1.4.1	Support of GIS industry standard formats	
2.1.4.2	OGC compatibility and compliance	
2.1.4.2 (1)	OGC compatibility and compliance: Import / Export of spatial information	
2.1.4.2 (2)	OGC compatibility and compliance: Projections	

5.2.1.3 Npgsql

Npgsql is a .NET data provider for PostgreSQL. It allows any program developed for the.NET framework to access the PostgreSQL database server.

Npgsql will be implemented as pure infrastructure component to allow the DSS front-end data access components to interact with the PostgreSQL database.

Table 5.5 maps the Npgsql component to the relevant requirements from the TOR. These focus on the ability to transfer necessary types of data between the database server and the NB DSS.

Table 5.5 Npgsql derived requirements

Req. No.	Comments	Issue ref
X.62	It shall be possible to transfer all data types stored in the database between the database and the NB DSS Front-end	

5.2.1.4 NHibernate

NHibernate[11] is an Object-relational mapping (ORM) solution for the Microsoft .NET platform: it provides a framework for mapping an object-oriented domain model to a traditional relational database. Its purpose is to relieve the developer from a significant portion of relational data persistence-related programming tasks. NHibernate is a port of

[11] The Consultant has with success used NHibernate in another large enterprise type of project (www.chemmanager.com) but has no experiences with other ORM implementations. The Consultant feels confident about the use of NHibernate; but if the Client has specific issues concerning the use of NHibernate, the Consultant suggests discussing them during the detailed design stage.

the Hibernate ORM solution for Java. The development of Hibernate dates back to 2001 and the development of NHibernate started in2005.

NHibernate is free as open source software that is distributed under the GNU Lesser General Public License.

It is described in much details on the NHibernate home page http://nhforge.org/Default.aspx.

The data access part of the DSS front-end uses NHibernate to map entities stored in the PostgreSQL database to .NET classes.

Table 5.6 maps the NHibernate component to the relevant requirements from the TOR. They focus on the ability to transfer the necessary types of data between the database server and the NB DSS.

Table 5.6 NHibernate derived requirements

Req. No.	Comments	Issue ref
X.63	It shall be possible to map tables and views to objects	
X.64	It shall be able to handle the data types stored in the database	
X.65	It shall be able to manage transactions	
X.66	It shall be possible to connect to the PostgreSQL data-base	

5.2.2 Programming Related Components

The programming related components are:

- ThinkGeo MapSuite Desktop (ThinkGeo) – A user interface control for displaying and interacting with spatial data.
- DotNetMagic – A user interface component providing different UI controls.
- Engineering Unit Management system (EUM) – A DHI library for handling item types and units.
- MIKE Objects – A DHI library for manipulating time series.

Each component is described below in terms of functionality.

5.2.2.1 ThinkGeo

ThinkGeo is a .NET native set of GIS software controls that lets .NET developers add rich, interactive maps to Windows desktop applications. It provides:

- Support for a large variety of spatial data types – vector as well as raster
- Functionality for zooming, spatial queries, projections etc.

- Options for annotations to the display through symbology, icons shapes etc
- Geometry functions like unions, intersects etc

ThinkGeo is presented in more details on http://gis.thinkgeo.com/Products/GISComponentsforNETDevelopers/MapSuiteDesktopEdition/GISFeaturesBenefits/tabid/675/Default.aspx.

NB DSS will use ThinkGeo for displaying GIS data in maps and for basic GIS functionality like unions and intersects.

Table 5.7 maps the ThinkGeo component to the relevant requirements from the TOR.

Table 5.7 ThinkGeo requirement mapping

Req. No.	Comments	Issue ref
2.1.4.1	Support for GIS industry standards	
2.1.4.3(2)	Provide basic GIS functionalities such as pan, zoom and select	

5.2.2.2 DotNetMagic

DotNetMagic from Crownwood is a collection of user interface controls. The product is described at http://www.dotnetmagic.com/index.html.

NB DSS uses the control for handling windows docking; i.e. snapping of windows to other windows within the user interface.

Table 5.7 maps the DotNetMagic component to the relevant requirements from the TOR and

Table 5.8 DotNetMagic requirement mapping

Req. No.	Comments	Issue ref
2.1.1.1	Interactive, fully menu driven graphical, hyperlinked	
2.1.1.1 (1)	The GUI shall be as powerful and interactive as possible for creating, locating and inter-connecting the various components.	

Table 5.9 DotNetMagic derived requirements

Req. No.	Comments	Issue ref
X.67	It shall be possible to dock windows to other windows	

5.2.2.3 EUM

EUM is DHI library for engineering unit management. It provides enumeration for item types like rainfall, water level, temperature etc and units like metre, degree Celsius etc. Additionally it provides API functions for:

- Tests whether a given unit can be used with a given item type; e.g. if degree Celsius is a valid unit for water levels
- Convert between units; e.g. given an item measured in metres the function can convert to millimetres

As of today EUM supports 485 different data types and 349 units. The system is extendable with respect to types, units and conversion rules, thus allowing the NBI to define/control which ones to use in the NB DSS

The NB DSS will use EUM by associating data items in the database with EUM item types. The system will also maintain a system table (Global data) with a preferred unit for each item type. I.e. the database will always store data in the system preferred unit. A user can overwrite the preferred unit in the system table by configuring the system to display specific data types in other units. This will only affect the display of the values, data will continue to reside in the database in the system preferred unit.

Table 5.10 EUM requirement mapping

Req. No.	Comments	Issue ref
2.1.2 (3)	The IMS shall manage units including their conversions. ***Clarification*** *NBI wants to avoid that any parameter has units that are implied instead of explicit. The system should include a number of units specifications for each data type, with options for the user to add a unit specification (and its associated conversion factor from the standard unit) for that data type.*	

Table 5.11 EUM derived requirements

Req. No.	Comments	Issue ref
X.68	NBI shall be able to add new data type and units to the system.	

5.2.2.4 MIKE Objects

MIKE Objects (or time series package) is a package of Microsoft COM components for handling time series. It is used to load, save, visualize - graphically or tabular – and process time series data. The package employs the EUM component described above and is used intensively by the MIKE products.

The NB DSS will use MIKE Objects as an in-memory storage mechanism for time series, i.e., after a time series has been read from the data base it will be converted into a time series objects. This enables the NB DSS to use all statistical and other tools included in the MIKE Objects package. MIKE Objects also include a number of “bridges” for accessing time series stored in various formats, e.g., ASCII files, Excel spreadsheets etc. NB DSS will leverage this functionality to import time series data into the database.

Table 5.12 MIKE Objects requirement mapping

Req. No.	Comments	Issue ref
2.1.3.1 (all)	Embedded statistical tools	
2.1.3.2	Time series analysis tools	

Table 5.13 MIKE Objects derived requirements

Req. No.	Comments	Issue ref
X.69	It shall be possible to provide an object-oriented interface to in-memory time series	
X.70	It shall provide bridges for importing time series in a wide variety of input formats	

5.3 Communications Interfaces

The NB DSS has the following external communication interfaces:

- With the Nile IS system
- With other installations of the NB DSS Enterprise editions

These interfaces will be described further in the sections below.

5.3.1 Nile IS System

NB DSS shall be able to upload data and reports to the Nile IS system.

Table 5.14 Nile IS requirement mapping

Req. No.	Comments	Issue ref
2.2.2	Interface with Nile-IS	12

5.3.2 Other Installations of the NB DSS Enterprise Editions

The NB DSS Enterprise editions form a “connected” system in the sense that data on user request shall be fully or partial synchronized among installations.

In a geographical dispersed organization like NBI, it is important to keep the global data (see Section 4.15) synchronized between the participating installations. The NB DSS will include functionality to export data from the global data area in a format allows easy import into other NB DSS installations. Individual data items can be tagged for inclusion or exclusion in such exports, i.e., providing a mechanism for retaining private data from participating in data synchronization.

Synchronising global data between geographically dispersed sites requires backing by policies and by a central managing organisation. One way to ensure that all participants have identical global data compartments is to establish a hub-and-spoke synchronisation structure. All installations (the spokes) submit change requests for the global data to a designated installation (the hub) that maintains a master version of the global data area. The hub will then make the changes in the master version and disseminate data deltas to the spokes.

Table 5.15 Other installation of the NB DSS Enterprise editions requirement mapping

Req. No.	Comments	Issue ref
2.1.2.1(8)	The DBMS shall support partial and full synchronization of the databases at different installation centers with the central database (at the regional DSS center) **Comment:** Synchronization of data between Enterprise installation will be based on logic within the NB DSS.	

Table 5.16 Other installation of the NB DSS Enterprise editions derived requirements

Req. No.	Comments	Issue ref
X.71	Policies for data synchronization shall be established by NBI	
X.85	Defining “private” data items; i.e. data that shall not participate in synchronization	

5.3.3 Scripting Interface

The NB DSS shall include a scripting facility that makes it possible to automate data processing activities that otherwise can be reached from the user interface. The scripting option shall also make it possible to interact with those modelling systems that provide scripting facilities.

Req. No.	Comments	Issue ref
2.1.2.7 (2)	The IMS shall have a scripting/macro language to automate data processing activities	25
2.1.2.7 (3)	The IMS shall store and manage coded procedures (scripts/macros) support automating recurrent activities	
2.1.7.6	Support for user defined representations/algorithms	

6 OTHER NONFUNCTIONAL REQUIREMENTS

6.1 Performance Requirements

This section addresses the requirements related to the performance of the IT system (as opposed to the performance of the models within the system).

Specific areas of the system where a discussion on performance is natural are:

- The Database

  How well does the database perform in returning data to the users/clients?

  - Time series data
  - GIS data
  - Data in general

  How well does the database update rows and handles large amounts of data?

- The Client

  How fast and accurately does the client application(s) respond to user input?

  How fast and accurately does the application update the GUI?

- The Model

  How fast can the use expect a model to complete a simulation?

All of the above questions are relevant to ask, but are very difficult to quantify. A user knows when something is unbearable – but to define a response time or reasonable limits for all situations is very difficult. Models run times, for instance, depend significantly on the hardware (CPU, memory, disk drives network etc.), the software, the size and extent of the model setup, and even the skill of the modeller defining the setup. I.e., a precise criterion for accepting performance of a model run is not possible. A user may accept longer time to plot time series she knows contain a lot of data, but will otherwise expect a faster response time

The MIKE Model Tools that will be used by the DSS are used by many different users for many different purposes all over the world and must be regarded well proven with respect to computational efficiency and they are tailored to handle large data sets including grids as well as long, high resolution, time-series.

The ToR /2/ contains very little specific requirements for performance of the system. Statements like “ease of use”, “optimized for Windows” has bearing on the performance, but more illustrate the resulting perception by the user of the quality of the system than details how this quality is to be measured.

Table 6.1 lists the explicit requirements related to performance. Table 6.2 lists derived requirements aiming at handling user response expectations.

Further specification of quantifiable measures for judging performance is required, especially since performance is a criterion for test cases.

Table 6.1 Requirements in ToR related to performance

Req. No.	Comments	Issue ref
2.2.1.1	Optimized for Windows ***Clarification*** *This is for the corporate edition.* **Comment:** In the corporate edition the server is not Windows, hence he optimisation is for the windows client application.	
2.2.6.2	Support a smooth transition to a backup machine with minimum disruption	

Table 6.2 Derived requirements related to performance

Req. No.	Comments	Issue ref
X.72	The application should show a progress bar for operations that potentially will take longer than 1 minute to complete.	
X.73	The application should show an hour-glass for operations that potentially takes longer than 3 seconds.	
X.74	The application should warn the user if database queries due to filter specifications may take “long” time to complete.	

6.2 Safety Requirements

The ToR /2/ does not explicit mention safety requirements and it has neither been subject for discussions during the inception period.

There are, however, situations where misuse of the system or the data within it may lead to wrong decisions which in theory could jeopardize the safety of people in the Nile Basin. Examples of this are:

- The system may contain erroneous data, created either through import of erroneous data or through incorrect transformation

- The models in the system may not represent the physical problems sufficiently (e.g. a model is used to simulate physical phenomena that are not adequately supported by the model tool or the skills of the modeller is inadequate to undertake a particular study)

- The models may not be used correctly for the problem at hand

- The results of the models and analysis tools may be interpreted incorrectly by the user

These potential issues call for procedures governing the use of the system to ensure review and control of all data and processing. Also, adequate user skills, obtained through education and training in the use of the features of the system, are required.

The latter is addressed via explicit requirements for training during the work packages 1 and 3. The former is an institutional issue to be resolved outside the NB DSS software.

Review and control processes are supported by requirements for traceability and metadata as they aim a creating a transparent process with respect to data and data processing.

Table 6.3 Derived requirements related to safety

Req. No.	Comments	Issue ref
X.75	NBI must establish procedures for using the system with respect to review and control of data and processing as well as modelling quality measures (goodness-of-fit performance measures).	

6.3 Security Requirements

The ToR /2/ includes a number of requirements related to security. These requirements are shown in Table 6.4. They reflect that the system must protect access to data. This includes

- Storage, i.e. the database which stores all the data must be robust and resilient in maintaining the data and preventing unauthorized access. The use of PostgreSQL as database will meet this level of security.

- Transfer, i.e. the communication between the nodes in the system, specifically across networks must be secure to prevent unauthorized access. Storage of data on local hard drives for processing (e.g. by model tools) must occur with credentials allowing only authorized users and processes to access the data. Synchronisation and export/import of data must also involve security measures. Especially the latter, which enables user defined transfers of data between different installations of the system, must employ rules regarding who can import/export what.

- Access, i.e. the application shall provide a user access rights system enabling segregation of users to allow access to only permitted data.

Discussions during the inception period have highlighted a further need for the NB DSS to separate data into two types: global and study data. Global data is common data available in the enterprise version of the system giving a starting point for a user to work with Nile basin relevant and approved data. Study data is data related to a study involving one or more persons.

Synchronising global data between geographically dispersed sites requires backing by policies and by a central managing organisation, as also mentioned in Section 4.15. One way to ensure that all participants have identical global data compartments is to establish a hub-and-spoke synchronisation structure. All installations (the spokes) submit change requests for the global compartment to a designated installation (the hub) that maintains a master version of the global compartment. The hub will then make the changes in the master version and disseminate data deltas to the spokes.

Table 6.4 Security Requirements

Req. No.	Comments	Issue ref
2.1.2.1 (3)	In order to avoid data corruption and losses the DBMS shall ensure a) data security, b) data integrity, c) data portability, d) multi-version concurrency control, and e) easy maintenance.	
2.2 (3)	The Bidder shall specify the hardware and software communication requirements, and suggest network tools, protocols, and security measurements to be used for data transfer, communication, administration and management.	
2.2 (4)	Integration and scope of the RDBMS in the Corporate Edition a. Accommodates Client/Server Application b. Contains very large datasets c. Has high concurrency d. Has high security level (including recovery mechanisms)	
2.2.6.1	Hierarchies of access privileges with institution level defined security features	

Table 6.5 Derived requirements related to security

Req. No.	**Comments**	**Issue ref**
X.76	NBI should establish an organisational unit with responsibility for the Global data.	
X.77	Procedures must exist supporting hub-and-spoke synchronization	

6.4 *Software Quality Attributes*

An overall ambition with the NB DSS is that the systems is open, both with respect to the data and the methodologies available but also with respect to the possibilities to customise, tailor and extend the system, should the initial set of functionalities prove inadequate – and at the same time be easy to use and maintain.

The development of the NB DSS will employ a number of techniques to obtain this flexibility and quality of the software.

- Extensibility

  The NB DSS will allow extensions to functionality

  - Model integration using the adapter pattern. By defining adapter interfaces for configuration and execution new 3rd party specialist model tools can be integrated into the NB DSS in a common manner to leverage the need for modelling capabilities. The potential drawback to this is that since all model tools then are treated equal by the NB DSS that special features of a particular tool may not be utilized unless exposed by the adapter. Hence, the adapter interfaces must be elaborate.
  - Ability to add tools dynamically. The framework of the NB DSS solutions will allow tools adhering to a well defined interface to easily plug-in into the application. They can be deployed from a central location very easily. This allows for 3rd party developers to provide new tools as they become necessary.
  - Possibility for the user to define scripts accessing the programming model of the NB DSS. This will enable users to combine the functionality exposed via API and/or .NET classes to new facilities in the NB DSS. Sharing this type of functionality will require a technique for "signing" or verifying scripts to work in other installations.
  - Using the reflection functionality of .NET to use assemblies added to a live system; without having to modify the existing functionality.

- Usability

  The out-of-the-box NB DSS Front-end will allow the user to work with data in a consistent manner as defined by the standard controls and windows. However, the user can modify the application layout (e.g. organise window positions, choose language) to her preferences. The application will then remember its appearance at the next login. A concept of "Favourites" – direct links to often used facilities - will be included.

  This requires the application to store such information either locally at the machine – or in the database.

Furthermore, the NB DSS Front-end framework shall be open with respect to dynamic inclusion of new displays with new functionalities should this be necessary.

- Maintainability

  A simple security model will make it easy for users to share data and simple for administrators (globally and per study) to administer the permissions.

  Dividing data into compartments of global and study data (see Section 4.15) will make it easier for users to work with data and share data, as the overview of the data in focus will be clearer.

Table 6.6 lists the requirements dealing with the above mentioned aspects. Table 6.7 lists additional requirements.

Table 6.6 Requirements regarding software quality aspects

Req. No.	Comments	Issue ref
2.1.2.8	Data Management Interfaces (DMI) ***Clarification*** *Perhaps the user will want to use other models in the future; if those models comply with the DMI, they should be able to be integrated within the system. This might allow flexibility for future (and existing) models to be more tightly integrated and potentially to be able to share time-step to time-step information.* *DHI should help to find flexibility and identify limitations in the requirements.* *DHI needs to express advantages and disadvantages of the approaches that will be proposed for how models communicate with one another and with the database and how future models may be integrated.*	
2.1.2.8 (1)	The DMI shall ensure seamless communication/integration between modules/models of the DSS.	
2.2	Non-functional Requirements ***Clarification:*** *We want to make sure that the design document addresses the noted concerns. –* *Inputs to the selection/identification of:* *Hardware platforms and all peripheral equipment required* *Management tools (database, software administration and configuration) required for the software system management*	

	Hardware and software communication requirements, and suggest network tools, protocols, and security measurements to be used for data transfer, communication, administration and management **Comment:** The differences between the Corporate and Professional editions are solely the data in the database, i.e. functionality-wise they are identical Professional is intended to be installed on a single Windows workstation	
2.2.1	System Architecture ***Clarification:*** *If comments are not clear, you need to ask*	
2.2.1 (1)	System Architecture: Reliability, portability, scalability, interoperability, and maintainability.	
2.2.1 (2)	System Architecture: Employ the use of industry standard interfaces for communication among the software components.	
2.2.1.2	Client-server architecture **Comment:** The server is a pure database server that only handles database interactions. Business logic is not anticipated to be placed here.	
2.2.1.3	Modular implementation	
2.2.1.3 (1)	The system shall have a capability to seamlessly add system functionalities in the future	
2.2.1.3 (2)	Layered architectures, modular design, well defined interfaces, object-orientation and component-based development	
2.2.1.4	Centralized processing and decentralized processing mode **Comment:** The server is a pure database server. It is not expected logic will be executed on the server	

2.2.1.5	Multi-user and multi-processor based	
2.2.1.6	Audit trail and log files for controlling purposes ***Clarification*** Comment as noted previously, record locking, concurrent processes, etc.	22
2.2.1.7	Highly cohesive and loosely coupled design ***Clarification*** *Please address comments in the design document.*	
2.2.6.2	Support smooth transition to a backup machine with minimum disruption	
2.3.*	CMMI **Comments:** These requirements are addressed in the Project Manual process.	

Table 6.7 Derived requirements related to software quality attributes

Req. No.	Comments	Issue ref
X.78	The NB DSS shall allow new displays and new functionality to be added ad deployed dynamically	
X.79	User specific settings like language and window positions shall be kept with the user profile.	

7 OTHER REQUIREMENTS

Table 7.1 below lists requirements not catalogued elsewhere in the document. Some include clarifications as formulated during the Launch workshop and/or comments by the consultant. In some cases the analysis of a requirement has lead to the conclusion that further analysis is required.

Some requirements relate to the project execution and deliverables and have no bearing on the system design.

Table 7.1 List of requirements not listed elsewhere

Req. No.	Comments	Issue ref
2.1.2.2	Data model, entity relationship model description explicit	
2.1.2 (4)	For systematic tracing of results the IMS shall support logging of all processes in the DSS and interactions of the user with the DSS	7
2.1.2.6	Support various database functionalities **Comment:** NBI has chosen the RDBMS. The consultant sees no limitations with respect to these requirements as a result of this choice.	
2.1.2.7 (4)	Batch data processing features: Traceability of outputs	2
2.1.2 (4)	IMS: - tracing **Comment** Further analysis is required.	
2.1.2.7	Batch Data Processing	
2.1.2.5	Support access to system performance statistics **Comment** Further analysis is required	
2.1.2.7 (1)	The IMS shall support the user to run all applications in the DSS in batch mode.	5
2.2.1.8	Implementation platform (platform independent)	1
2.1.2.8 (2)	The DMI shall be the integration base for module-module and module-database communication and interaction for all tools, mode... **Comment:** The model tools will typically not communicate directly with the DSS (or DSS database); but rather work on files exported from the DSS or generating files that automatically will be imported in the DSS after simulation has ended.	
2.1.2.8 (3)	In addition, the data management interface (DMI) shall	

Req. No.	Comments	Issue ref
	support plug-ins of modules that meet the require-ments of the communic...	
2.1.7.4	Catchment erosion process modelling **Comment** Further analysis required	
2.1.2.7 (1)	Batch data processing features: Batch mode require-ments **Comment** Further analysis required	
2.1.2.7 (2)	Batch data processing features: Macro/scripting **Comment** Further analysis required	
2.1.2.7 (3)	Batch data processing features: Storage and man-agement of coded procedures **Comment** Further analysis required	
2.1.2.7 (4)	Batch data processing features: Traceability of outputs **Comment** Further analysis required	
2.2.3	Documentation	
2.2.3.1	Manuals	
2.2.3.2	Help-system **Comments:** The help system will be according to current standards for windows online help, e.g. Microsoft Office or similar where context sensitivity and search facilities are available..	
2.2.3.3	Manuals and documentation in French	
2.2.3.4	Tutorial material, test data sets and example results **Comments:** Not applicable	
2.2.4.*	General Administrative **Comments:** These requirements are managed as part of the overall project management	
2.2.5	User Training **Comments:** Discussed elsewhere	
2.2.6	Security Requirements and Data Recovery	

8 ISSUES LIST

Table 8.1 below list the issues identified during the use case and requirement analysis. Besides the issue description the table shows the ID of the issue (the reference use elsewhere), the Ref (where the issue is discovered) and for issues related to specific requirements, the reference to this.

Table 8.1 List of issues

ID	Description	Ref	Requirement
1	Version and distribution of Linux needs to be decided. Version of Windows Vista needs to be confirmed	SRS 7	2.2.1.8
2	The implications on performance and database size of providing full traceability on data need to be clarified and discussed.	SRS 5.1	2.1.2.7 (4)
3	Design of online help need to be elaborated during the design phases	SRS 7	2.1.1.3
4	Managing 2.1.1.4 (4) by following Windows locale settings needs to be agreed upon	SRS 5.2.1	2.1.1.4 (4)
5	The term “All Applications” needs to be further detailed. The implications on performance and database size of logging all interactions need to be clarified and discussed.	SRS 7	2.1.2.7 (1)
6	The term “any time period” is unclear. Does it refer to time zones or to reference systems?	SRS 5.2.1	2.1.2 (1)
7	It is unclear if the requirement deals with both logging and changes, i.e. shall it be possible to revert previous made changes? The requirement can potentially have severe impact on performance and database size.	SRS 7	2.1.2 (4)
8	Algorithm for projection of future demands needs to be defined.	SRS 4.2	2.1.3.2 (5)
9	Decision on file formats to be supported needs to be confirmed.	SRS 4.1	2.1.4.2 (1)
10	Agreement on the query language is needed	SRS 4.16	2.1.2.4 (1)
11	Decision on supported export formats need to be taken	SRS 4.16	2.1.2.4 (4)
12	Agreement on interface to Nile IS needs to be taken	SRS 5.3.1	2.2.2
13	Meaning of requirement is unclear	SRS 5.2.1	2.1.5.9
14	Requirements for the tool for converting model output to indicator variables needs to be further detailed	SRS 4.11 SRS 4.12	2.1.5.12
15	Node types are not determined by the DSS, but by the applied model tools.	SRS 5.2.1	2.1.5.18
16	Groundwater management options need to be further defined	SRS 5.2.1	2.1.5 (2)
17	The requirement is unclear	SRS 5.2.1	2.1.5 (3.6)
18	The requirement is unclear	SRS 5.2.1	2.1.5 (3.7)
19	The term “bio-physical” is in this context not understood	SRS 5.2.1	2.1.5 (4)

ID	Description	Ref	Requirement
20	Neither MIKE 11 nor MIKE BASIN currently includes catchment erosion process modelling. Requirements for this feature needs to be further defined.	SRS 7	2.1.7.4
21	Sensitivity analysis need to be further defined (is it optional when the tools are optional?)	SRS 4.5	2.1.6.5
22	Audit trail and log requirements need to be established	SRS 6.4	2.2.1.6
23	Backup and restore shall be based on standard functionality coming with the database	SRS 5.2.2	2.2.1.9
24	An exhaustive list of hydro objects including their characteristics shall be established	SRS 4.4	2.1.5.2 2.1.5.2 (1) 2.1.5.2 (2) 2.1.5.2 (3) 2.1.5.2 (4) 2.1.5.2 (5) 2.1.5.2 (6) 2.1.5.2 (7)
25	Scripting language has not been identified	SRS 5.3.3	2.1.2.7
26	It shall be investigated whether the Study synchronization can be based on standard database synchronization only.	SRS 5.6	X.81