

**Client:**  
**Environment Agency, UK**

# **National Groundwater Modelling System**

**Phase 2 – Detailed Architecture Design  
(Change Control Note 2005/03)**

**User Interface Specification Document  
Version 1.0 (final product Phase 2)**

01 February 2006



**wL | delft hydraulics**



| <b>CLIENT:</b>                 | Environment Agency<br><br>UK   |               |                                |                             |   |  |
|--------------------------------|--|---------------|--------------------------------|-----------------------------|---|--|
| <b>TITLE:</b>                  | National Groundwater Modelling System (NFFS CCN 2005/03)<br>Phase 2 – Detailed architectural design<br>User Interface Specification Document |               |                                |                             |   |  |
| <b>REFERENCES:</b>             | EA Contract Reference Number 11918-1-8<br>EA Order Number 30056390 dated 21/06/2005  |               |                                |                             |   |  |
| VER.                           | ORIGINATOR   | DATE          | REMARKS                        | REVIEW                      | APPROVED BY                               |  |
| 0.1                            | Gijsbers   | August 2005   | draft                          |                             |   |  |
| 0.2                            | Gijsbers   | October 2005  |                                |                             |   |  |
| 0.3                            | Gijsbers   | November 2005 |                                |                             |   |  |
| 1.0                            | Gijsbers   | February 2006 |                                |                             |   |  |
| <b>PROJECT IDENTIFICATION:</b> |  |               |                                |                             |   |  |
| <b>KEYWORDS:</b>               | NFFS, NGMS, National Groundwater Modelling System, IT architecture   |               |                                |                             |   |  |
| <b>NUMBER OF PAGES</b>         | 32   |               |                                |                             |   |  |
| <b>CONFIDENTIAL:</b>           | <input checked="" type="checkbox"/> YES, until (date)  |               |                                | <input type="checkbox"/> NO |   |  |
| <b>STATUS:</b>                 | <input type="checkbox"/> PRELIMINARY   |               | <input type="checkbox"/> DRAFT |                             | <input checked="" type="checkbox"/> FINAL |  |

## Preface

Groundwater Models have been in use by the Agency and its predecessors for many years and these have been initiated and implemented at Regional level. These models were not linked in a national context. Groundwater models are also costly and have not been easily linked to the needs of operational customers. The fragmented approach to modelling was highlighted by the development of source protection zones in the early 1990's. It was recognised that a centrally co-ordinated, strategic approach was required to avoid duplication and reduce the risk of challenge.

To address this situation, a Strategic Review of Groundwater Modelling (R&D Project W6-034, R&D Technical Report W214; Brown and Hulme, 2001) was undertaken. The main output of the Strategic Review was the *Environment Agency Framework for Groundwater Resources Conceptual and Numerical Modelling* (R&D Technical Report W214) which contained a nationally consistent technical approach and programme for regional groundwater resources assessment and modelling.

The Head Office Hydrogeology Team presented a summary *Implementation Plan* for this work jointly with the Science Group which was accepted by the national Water Resource Management Team (WRMT) in October 2004. Regional modelling strategies were also recognised by WRMT as strategic Water Resources capital programmes. This work therefore supports the Streamlining Abstraction Processes (SAP) and Restoring Sustainable Abstraction (RSA) programmes managed by the national Water Resources Regulation team.

The Head Office Hydrogeology team is now developing a more detailed *Implementation Strategy* comprising a series of measures to support groundwater modelling, ensuring appropriate national consistency, improving efficiency and accessibility by customers. The Implementation Strategy will address concerns like national planning of model development, benefit realisation, succession planning, business efficiency, IS performance, and customer accessibility to models.

An *IT Strategy for Groundwater Resource Assessment and Modelling* is being prepared to address the infrastructure and IS performance issues.

The National Flood Forecasting Projects (NFFS) is currently implementing an IT architecture for a centrally hosted flood forecasting system for the Environment Agency (EA). It is recognised that there are strong links between the proposed IT Strategy for Groundwater Modelling and the NFFS. A feasibility study has been conducted which concluded that the IT-backbone, named National Groundwater Modelling System (NGMS), can be based on the NFFS architecture (and software components) if some minor modifications and extensions are implemented. This outcome was the starting point of phase 2, the detailed architectural design. In this phase the required modifications and extensions need to be made explicit.

Phase 2 of the NGMS will produce the following documents...

- Update of phase 1 User Requirements Document (URD)
- Update of phase 1 Software Requirements Document (SRD)
- Architecture Design Document (ADD)

User Interface Specification Document (UISD)  
Interface Definition Document (IDD)  
Hardware and Infrastructure Design document (HID)  
Update of Project Implementation Plan (PIP)

The *User Interface Specification Design* is presented in this document. It identifies the required User Interface components, specifies the interaction flows and a possible screen layout.

### **Guide to the reader**

The document lying before you provides the user interface specification for the *National Groundwater Modelling System*. This specification document is intended to provide an answer to the following question: “*What modifications are proposed to the NFFS user interface and how will the user be able to interact with the system?*”

The level of detail that is contained within this document is limited to providing the reader an overall picture of the systems operation, to enable an understanding of the proposed implementation of the system.

*Chapter 1* describes the user interface architecture as applied in the NFFS.

*Chapter 2* introduces the high-level user interface design as proposed for the NGMS..

*Chapter 3* specifies in more detail the functionality of the individual components.

### **New in this document**

Version 0.1 is the initial version

In version 0.2 the NGMS explorer description has been added

In version 0.3 all plug-ins have been added

In version 1.0 review comments have been incorporated.

## Contents

|          |  |            |
|----------|--|------------|
| <b>1</b> | <b>Introduction.....</b>   | <b>1-1</b> |
| 1.1      | Introduction.....  | 1-1        |
| 1.2      | The NFFS and its user interface .....                                | 1-1        |
| 1.3      | Analysing the NFFS user interface capabilities.....                  | 1-3        |
| <b>2</b> | <b>Proposed outline of the NGMS user interface.....</b>              | <b>2-1</b> |
| 2.1      | Introduction.....  | 2-1        |
| 2.2      | The NGMS explorer .....  | 2-2        |
| 2.2.1    | The Explorer map .....   | 2-2        |
| 2.2.2    | Zoom extent .....  | 2-3        |
| 2.2.3    | Plug-in toolbar/Explorer tasks .....                                 | 2-3        |
| 2.2.4    | Filters - Navigation panel .....                                     | 2-5        |
| <b>3</b> | <b>Plug-ins .....</b>  | <b>3-1</b> |
| 3.1      | Reuse plug-ins with minor modification .....                         | 3-1        |
| 3.1.1    | GridDisplay.....   | 3-1        |
| 3.1.2    | TimeSeriesDialog .....   | 3-1        |
| 3.1.3    | System Monitor .....   | 3-2        |
| 3.1.4    | DataExport.....  | 3-2        |
| 3.1.5    | Report .....   | 3-2        |
| 3.2      | Reuse plug-ins with major modification.....                          | 3-3        |
| 3.2.1    | GridDisplay → ModuleInputBrowser .....                               | 3-3        |
| 3.2.2    | ManualForecast → TaskRunDialog.....                                  | 3-5        |
| 3.2.3    | Forecast → DataDownloadDialog.....                                   | 3-7        |
| 3.3      | New plug-ins.....  | 3-8        |
| 3.3.1    | AbstractionLocator .....   | 3-8        |
| 3.3.2    | AbstractionTmsDialog / WhatIfScenarioDialog /<br>ScenarioWizard..... | 3-10       |
| 3.3.3    | Slice Locator / SliceWizard .....                                    | 3-12       |
| 3.3.4    | OutputFilter.....  | 3-15       |
| 3.3.5    | ModuleRun Journal .....  | 3-15       |
| 3.3.6    | Administrative plug-ins .....  | 3-16       |
| 3.3.7    | Configuration plug-ins.....  | 3-16       |



# I Introduction

## I.1 Introduction

The NGMS user interface is the view of the user on the client-server based system. The most important tasks to be supported are:

- inspection of module input data sets (UC ...)
- definition of what-if scenarios including modification of abstraction time series, or addition of new abstraction locations.

## I.2 The NFFS and its user interface

As the NGMS will build upon the NFFS architecture, a proper understanding of the rationale of the NFFS user interface is required to understand the capabilities and limits to build upon this system.

Within the NFFS, a major role is played by the DelftFEWS software application. This application can be operated stand-alone or in a client-server system. On the server side, it performs all data processing and computational jobs upon request of the Master controller. On the client side, DelftFEWS provides the user interface to the user. Each instance of DelftFEWS has its local data store which is synchronized with the central data before a new activity is undertaken.

Figure 1-1 illustrates the main window (the ‘Explorer’) of the NFFS user interface. The Explorer holds a navigation panel on the left, a plug-in bar on the top, a message window on the bottom and a base map display in the middle right.

### *Navigation panel*

The function of the navigation panel is to enable quick access to the available data in the system, either observed or calculated. Filters are applied to organised data by location type (i.e. type of monitoring entity or forecast entity) and data type (i.e. variable). Once a location and data type have been selected, the data set is identified and can be displayed on the time series display. The red cross in the figure indicates that the data set is outdated (i.e. expired) for real-time application.

### *Map Display*

The Map Display is used to locate the various monitoring and forecast stations. ID-information is provided via a tooltip. The background of the map is composed of GIS-layers (shape files) that can be switched on and off. No data can be accessed by clicking on the map. The zoom extent can be managed by mouse scrolling or by selecting the extend level.

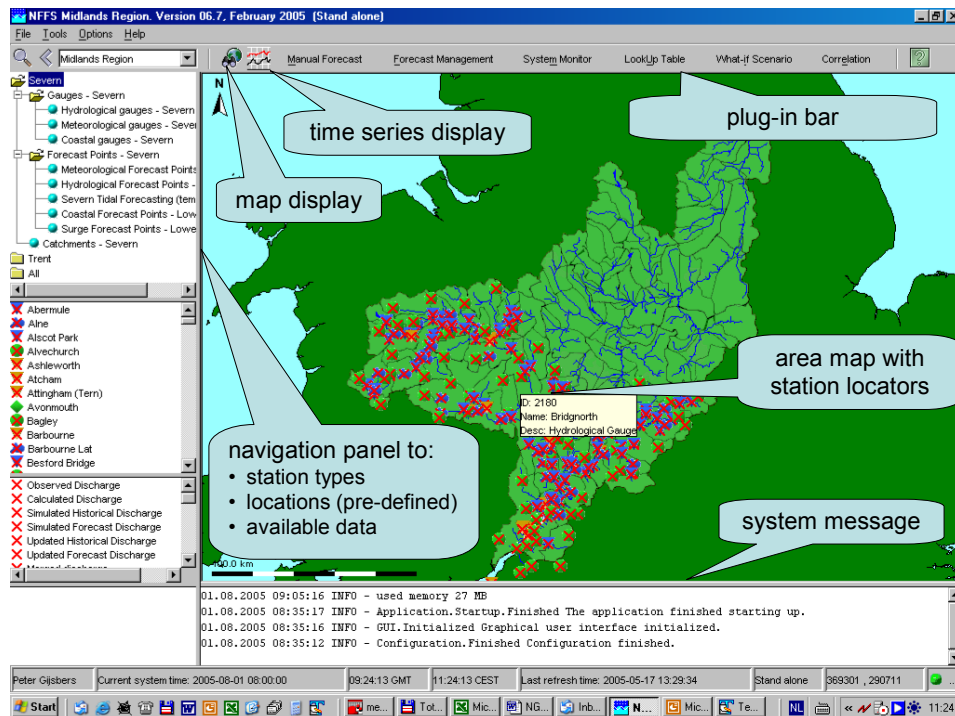


Figure 1-1 The DelftFEWS user interface for NFFS

### Log Viewer

The Log Viewer provides information about the status of tasks run by the system. The messages that are displayed by the Log Viewer – called System Messages - are meant for both end users and system managers. Typical system logs are the successful start and finish of a module run.

### Plug-in bar

The plug-in bar (as well as the Tools menu) provides access to the user interface components that are dedicated for specific jobs. The following plug-ins are available for the NFFS:

- *Data Editor*: By selecting Data Editor, time series can be displayed and edited in the form of tables.
- *Data Display*: By selecting Chart, time series can be displayed in graphs.
- *Grid Display*: By selecting Grid Display, Meteorological Forecast grids or Hydrological Flood Maps can be displayed on a map (see also Grid Display)
- *Manual Forecast*: Via the Manual Forecast Display, the user can define forecasts that should be run in manual mode.
- *System Monitor*: The System Monitor allows the user to view selections of system messages and acknowledge severe and fatal errors.
- *Lookup Table*: The Lookup Table Display allows for working with lookup tables interactively and run what-if scenarios on them.
- *What-if Scenario*: Via the What-if Scenario Display, the user can define what-if scenarios which can be run via the Manual Forecast Display.
- *Forecast Management*: Via the Forecast Management Display, the user can download forecasts from the Central Database and make forecast active for viewing. The current forecast can be set and archived forecasts can be retrieved.



- **Correlation:** The Correlation display allows for working with interactive regression functions and makes alternative (off-line) forecasts.
- **Calibration:** The Calibration display allows a user to automatically optimize the model calibration parameters.
- **Reports:** The Reports menu item activates Internet Explorer, through which any reports that have been made can be viewed.

The functionality offered by the Tools menu may also be available via the Button Bar. Figure 1-2 illustrates some NFFS plug-ins, namely Manual Forecast (upper left); Forecast Management (upper right); System Monitor (lower right); What-if Scenario (lower right).

Each of the plug-ins shown fulfils a task that is desired for NGMS as well, namely:

- what-if/boundary conditions set definitions
- run definition and execution
- module data management
- system monitoring

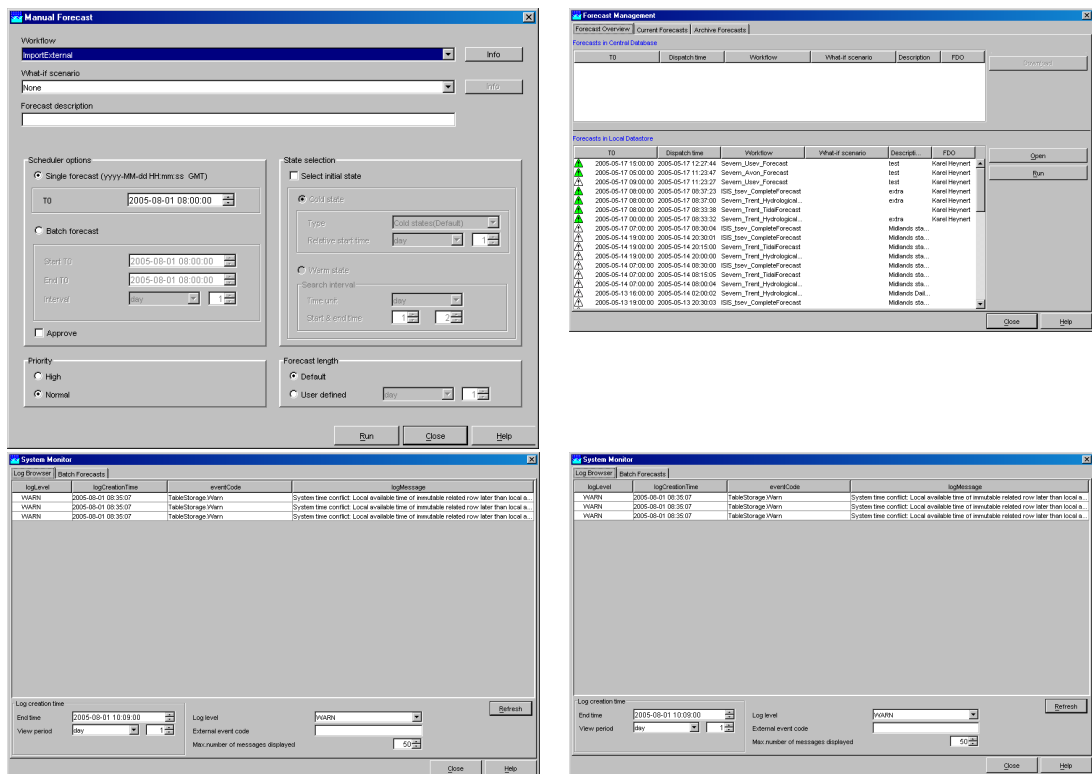


Figure 1-2 Illustrative screen shots of NFFS plug-ins to define, manage and monitor forecasts. (upper left): Manual forecast; (upper right): Forecast Management; (lower left): System Monitor; (lower right): What if-Scenarios

### 1.3 Analysing the NFFS user interface capabilities

When analyzing the entire user interface architecture in more detail, it shows that a large conceptual assumption underlying the NFFS is not very practical from the NGMS perspective.

### *The NFFS perspective*

The NFFS concept assumes that all locations of the system are known beforehand. This assumption is quite logical for a forecasting system as all monitoring stations and forecast locations already exist in the field. Hence, it is not surprising that pre-defined locations play two important roles in the architecture design.

First of all, all data is referenced via these locations, i.e. the system can only access data via locations that have been defined in the configuration. This prominent role of pre-defined locations is visible throughout the entire user interface. For example, the entire contents of the navigation panel (Figure 1-1) are based on these pre-defined locations, while the Base Map Display is linked to those locations as well.

Secondly, given the required robustness of the NFFS and the fact that station positions seldom change, the system has been designed such way any modification of location information requires a modification of the configuration which can only be incorporated by restarting the system. Such restart to update the system configuration is considered not a big problem if it only happens few times a year.

### *The NGMS perspective*

The fixed location concept does to a limited extend fit the needs for the NGMS as impact analysis on modified operation of existing well fields is an important application field for NGMS. However, concept is less suited for another application field, namely the impact assessment of new abstraction locations.

A frequent system shut down and restart to incorporate new abstraction locations in the system configuration will not be appreciated by its users. Hence other solutions will be proposed to enable specifications of new abstraction locations. These proposals will also have impact on the navigation to data and thus to the user interface.

However, it is proposed to incorporate the important locations of standard scenarios into the region configuration, thus enabling visualization of standard scenario data at selected locations via the NGMS-explorer.

## 2 Proposed outline of the NGMS user interface

### 2.1 Introduction

The proposed outline of the NGMS user interface will be based adopt the NFFS concept by the combination of an Explorer for general tasks and dedicated plug-ins to perform specific tasks.

The general tasks supported by the Explorer will be:

- provide access to the time series of Approved Scenarios via the navigation panel, the central map and the TimeSeriesDialog
- provide access to plug-ins

The following types of plug-ins are proposed

| Plug-in name       | Type  | Origin  | Task   |
|--------------------|-------|---|--|
| GridDisplay        | Grid  | GridDisplay   | Displays sliced or post processed output on a map.   |
| TimeSeriesDialog   | Chart | TimeSeriesDialog  | Displays sliced or post processed output on a chart.   |
| SystemMonitor      | Form  | adaptation<br>SystemMonitor   | Displays system and job status   |
| DataExport         | Form  | adaptation<br>DataExport  | Exports data in Excel and/or Shape format to local disk  |
| Report             | Form  | Report  | Generates and displays reports   |
| ModuleInputBrowser | Form  | extension of<br>GridDisplay<br>functionality                                  | This plug-in allows browsing of available module instances (including data sets), and provides for each module instance the associated meta data as well as a GridDisplay of input data sets |
| TaskRunDialog      | Form  | new Dialog, revised<br>layout, functionality<br>similar to<br>ManualForecast  | specifies the execution instructions and submits the job to the master controller  |
| DataDownloadDialog | Form  | new Dialog, revised<br>layout, functionality<br>similar to<br>ForecastManager | Enables identification of data being synchronized (i.e. downloaded)  |
| AbstractionLocator | Form  | New   | To define new abstraction locations and assign them to a groundwater module layer  |

|  |      |  |  |
|--|------|--|--|
|  |      |  | This plugin could be represented in the form of a wizard   |
| AbstractionTmsDialog /<br>What-If ScenarioDialog /<br>ScenarioWizard | Form | new or adapted<br>Dialog                       | Dialogue to define modifications to existing data sets<br>NB. the AbstractionLocator might be combined with the AbstractionTmsDialog in a ‘ScenarioWizard’   |
| SliceLocator / SliceWizard   | Map  | new  | The SliceLocator defines the horizontal extent of the data slice that can be extracted from a bulky module output.<br><br>This plug-in could be extended to a wizard which creates a complete slice definition |
| OutputFilter   | Form | new, may be combined with<br>TaskManagerDialog | Dialog that specifies the raw-output generation  |
| ModuleJournal  | Form | new  | Dialog to specify the background information on this module data set   |

## 2.2 The NGMS explorer

The main task of the NGMS explorer is:

- to provide access to the Approved Scenarios and associated data set;
- to provide access to all plug-in functionality for specification, deployment and visualization of module runs.

### 2.2.1 The Explorer map

The explorer map can display map layers of the NGMS as well as GIS-layers available in shape format. It is proposed to incorporate the following types of layers:

- General topographic information layers (Ordnance Survey)
  - rivers
  - reservoirs
  - land use
  - catchments
  - coastline
  - geology/soil types
- Environment Agency specific topographic information layers
  - Env.Agency Area(s)
  - CAMS area(s)

- WFD river basin district(s)
- Local Authority areas (for contaminated land purposes)
- Approved Scenario information layers
  - locations (see navigation panel for details)
  - location labels
- Module data set information layers
  - contour boundaries of module data sets
  - network/grid of module data set

The explorer options and the manipulation menu/form can remain the same.

|                            |                               |
|----------------------------|-------------------------------|
| Requested change from NFFS | only changes to configuration |
|----------------------------|-------------------------------|

### 2.2.2 Zoom extent

The zoom extent will be in principle be utilized in the same way as it is used in the NFFS; that is: it will determine the explorer map focus.

Note however that the selected zoom region will be passed as an argument to the Module Data Set Browser, in order to display only the module data sets associated to this zoom extent.

|                            |   |
|----------------------------|---|
| Requested change from NFFS | selected zoom extent will be passed as argument when calling the Module Data Set Browser plugin |
|----------------------------|---|

### 2.2.3 Plug-in toolbar/Explorer tasks

The functionality of the plug-in toolbar will remain the same; namely invoke a plug-in according to the selected Explorer Task.

|                            |  |
|----------------------------|--|
| Requested change from NFFS | only configuration changes to accommodate various new or modified plug-ins (see Section 3)   |
| example configuration      | <pre> &lt;explorerTask name="Module Browser"&gt;   &lt;taskClass&gt;...plugin.spatialdisplay.SpatialDisplay&lt;/taskClass&gt;   &lt;arguments&gt;zoom extent&lt;/arguments&gt;   &lt;arguments&gt;browse instance&lt;/arguments&gt;   ... &lt;/explorerTask&gt;  &lt;explorerTask name="Module Journal"&gt;   &lt;taskClass&gt;...plugin.modulejournal.JournalDialog&lt;/taskClass&gt;   &lt;arguments&gt;ModuleRunID&lt;/arguments&gt;   ... &lt;/explorerTask&gt; </pre> |

|  |
|--|
| <pre>&lt;explorerTask name="Deployment"&gt;     &lt;taskClass&gt;...plugin.manualforecast.RunDialog&lt;/taskClass&gt;     &lt;arguments&gt;ModuleRunOptions&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="System Monitor"&gt;     &lt;taskClass&gt;...plugin.systemmonitor.SystemMonitorDialog&lt;/taskClass&gt;     &lt;arguments&gt;MonitorFilters&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Disk Management"&gt;     &lt;taskClass&gt;...plugin.runmanagement.RunManagementDialog&lt;/taskClass&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Grid Display"&gt;     &lt;taskClass&gt;...plugin.grid.GridDisplay&lt;/taskClass&gt;     &lt;arguments&gt;InputDisplayFilter&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Data Display"&gt;     &lt;taskClass&gt;...plugin.timeseries.TimeSeriesDialog&lt;/taskClass&gt;     &lt;arguments&gt;chart&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Abstraction Locator"&gt;     &lt;taskClass&gt;...plugin.scenario.AbstractionDialog&lt;/taskClass&gt;     &lt;arguments&gt;position&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="What-if Scenario"&gt;     &lt;taskClass&gt;...plugin.scenario.WhatIfScenarioDialog&lt;/taskClass&gt;     &lt;arguments&gt;WhatIfScenarioFilters&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Slice Locator"&gt;     &lt;taskClass&gt;...plugin.mapdisplay.SpatialSliceLocator&lt;/taskClass&gt;     &lt;arguments&gt;SliceFilter&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Output Specifier"&gt;     &lt;taskClass&gt;...plugin.outputfilter.OutputFilterDialog&lt;/taskClass&gt;     &lt;arguments&gt;OutputFilters&lt;/arguments&gt;     ... &lt;/explorerTask&gt; &lt;explorerTask name="Report"&gt;</pre> |
|--|

|  |  |
|--|--|
|  | <pre> &lt;taskExe&gt;\$EXPLORER_HTML_VIEWER\$&lt;/taskExe&gt; &lt;arguments&gt;\$EXPLORER_REPORTINDEX\$&lt;/arguments&gt; &lt;/explorerTask&gt;         </pre> |
|--|--|

## 2.2.4 Filters - Navigation panel

It is proposed that the Navigation panel is the main entry point for time series data related to the Approved Scenarios. These scenarios have an approved and fixed set of abstraction locations, calculation and monitoring points with associated time series (see Figure 2-1). These time series can directly be displayed in a chart. Scenario related Map or grid views have to be accessed via the GridDisplay at the Plugin-toolbar.

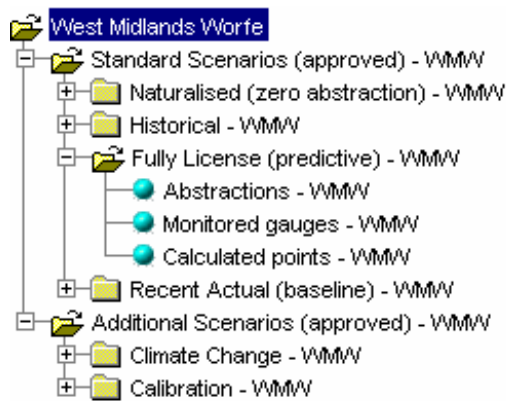


Figure 2-1 Example tree view to access data of Approved Scenarios

| Requested change from NFFS | configuration changes, possibly modifications in relation to data access and user right   |
|----------------------------|---|
| example configuration      | <pre> &lt;filter id="West Midlands Worfe"&gt;   &lt;!-- --&gt;   &lt;child foreignKey="Standard Scenarios (approved) - WMMW"/&gt;   &lt;child foreignKey="Additional Scenarios (approved) - WMMW"/&gt; &lt;/filter&gt; &lt;filter id="Standard Scenarios (approved) - WMMW"&gt;   &lt;!-- --&gt;   &lt;child foreignKey="Naturalised (zero abstraction) - WMMW"/&gt;   &lt;child foreignKey="Historical - WMMW"/&gt;   &lt;child foreignKey="Fully License (predictive) - WMMW"/&gt;   &lt;child foreignKey="Recent Actual (baseline) - WMMW"/&gt; &lt;/filter&gt; &lt;filter id="Additional Scenarios (approved) - WMMW"&gt;   &lt;child foreignKey="Climate Change - WMMW"/&gt;   &lt;child foreignKey="Calibration - WMMW"/&gt; &lt;/filter&gt; &lt;filter id="Naturalised (zero abstraction) - WMMW"&gt;         </pre> |

|  |  |
|--|--|
|  | <pre>&lt;!--All forecasting points--&gt; &lt;child foreignKey=" Gauging stations - WMW"/&gt; &lt;child foreignKey=" Assessment points - WMW"/&gt; &lt;/filter&gt; &lt;filter id="Historical - WMW"&gt; &lt;child foreignKey="Abstraction points - WMW"/&gt; &lt;child foreignKey=" Gauging stations - WMW"/&gt; &lt;child foreignKey=" Assessment points - WMW"/&gt; &lt;/filter&gt; &lt;filter id="Fully License (predictive) - WMW"&gt; &lt;child foreignKey="Abstraction points - WMW"/&gt; &lt;child foreignKey=" Gauging stations - WMW"/&gt; &lt;child foreignKey="Assessment points - WMW"/&gt; &lt;/filter&gt; &lt;filter id="Recent Actual (baseline) - WMW"&gt; &lt;child foreignKey="Abstraction points - WMW"/&gt; &lt;child foreignKey="Gauging stations - WMW"/&gt; &lt;child foreignKey="Assessment points - WMW"/&gt; &lt;/filter&gt; &lt;filter id="Climate Change - WMW"&gt; &lt;child foreignKey="Abstraction points - WMW"/&gt; &lt;child foreignKey="Gauging stations - WMW"/&gt; &lt;child foreignKey="Assessment points - WMW"/&gt; &lt;/filter&gt; &lt;filter id="Calibration - WMW"&gt; &lt;child foreignKey="Abstraction points - WMW"/&gt; &lt;child foreignKey="Gauging stations - WMW"/&gt; &lt;child foreignKey="Assessment points - WMW"/&gt; &lt;/filter&gt;</pre> |
|--|--|



## 3 Plug-ins

### 3.1 Reuse plug-ins with minor modification

#### 3.1.1 GridDisplay

| Item                                    | Details   |
|---|---|
| <b>Class name</b>                       | <code>plugin.grid.gridDisplay</code>  |
| <b>Key Functionality</b>                | Displays module output in form of grid time series  |
| <b>Architecture Description</b>         | Plug-in activated by Explorer   |
| <b>Changes from NFFS Implementation</b> | <ul style="list-style-type: none"> <li>add contour plotting library</li> </ul>                                |
| <b>Issues</b>                           | <ul style="list-style-type: none"> <li>ensure that data from the selected module run is visualized</li> </ul> |

#### 3.1.2 TimeSeriesDialog

| Item                                    | Details   |
|---|---|
| <b>Class name</b>                       | <code>plugin.timeseries.TimeSeriesDialog</code>   |
| <b>Key Functionality</b>                | Displays module output as time series at predefined point locations or at selected grid points  |
| <b>Architecture Description</b>         | Plug-in activated by Explorer or by GridDisplay   |
| <b>Changes from NFFS Implementation</b> | <ul style="list-style-type: none"> <li>chart extensions might be needed to accommodate displays (e.g. bar charts, groundwater unit budgets) as desired for groundwater assessment reports (see also Report-module)</li> </ul> |
| <b>Issues</b>                           | <ul style="list-style-type: none"> <li>ensure that data from the selected module run is visualized</li> </ul>   |

### 3.1.3 System Monitor

| Item                                    | Details  |
|---|--|
| <b>Class name</b>                       | <code>plugin.systemmonitor.SystemMonitorDialog</code>  |
| <b>Key Functionality</b>                | This plug-in displays diagnostics messages as reported by the modules in a workflow.   |
| <b>Architecture Description</b>         | Plug-in activated by Explorer  |
| <b>Changes from NFFS Implementation</b> | <ul style="list-style-type: none"> <li>• accommodate pick up and display of intermediate status messages</li> <li>• synchronize status message with central system upon monitor request (i.e. activation)</li> </ul> |
| <b>Issues</b>                           | <ul style="list-style-type: none"> <li>• synchronization Master Controller → Operator Client</li> </ul>  |
| <b>Explanatory</b>                      |  |

### 3.1.4 DataExport

| Item                                    | Details   |
|---|---|
| <b>Class name</b>                       | <code>plugin.dataExport.runExport</code>  |
| <b>Key Functionality</b>                | This plug-in exports data in a predefined format.   |
| <b>Architecture Description</b>         | Plug-in activated by Explorer with export format argument: <code>eaExport</code> , <code>excelExport</code> , <code>shapeExport</code> , <code>ascgridExport</code>   |
| <b>Changes from NFFS Implementation</b> | <ul style="list-style-type: none"> <li>• Extend export formats to support EA-timesSeries format (existing), excel (new), shape+dbf (new) and possibly ascii/grid</li> </ul>   |
| <b>Issues</b>                           | <ul style="list-style-type: none"> <li>• mapping from time series set to file formats to be defined (either fixed options in code or by configuration file)</li> <li>• contour plot library (see GridDisplay) is needed for export to shape</li> <li>• ensure that data from the selected module run is reported</li> </ul> |

### 3.1.5 Report

| Item                                    | Details   |
|---|---|
| <b>Class name</b>                       | <code>plugin.report.reportController</code>   |
| <b>Key Functionality</b>                | This plug-in reports data in a predefined format and layout.  |
| <b>Architecture Description</b>         | Plug-in activated by Explorer   |
| <b>Changes from NFFS Implementation</b> | <ul style="list-style-type: none"> <li>• chart extensions might be needed to accommodate displays (e.g. bar charts, groundwater unit budgets) as desired for groundwater assessment reports (see also Report-module)</li> <li>• contour plot library (see GridDisplay) is needed to accommodate contour plot reports</li> </ul> |
| <b>Issues</b>                           | <ul style="list-style-type: none"> <li>• ensure that data from the selected module run is reported</li> </ul>   |

## 3.2 Reuse plug-ins with major modification

If the plug-in complexity increases too much for shared application by NFFS and NGMS, it might be decided to develop a new plugin.

### 3.2.1 GridDisplay → ModuleInputBrowser

#### Summary

|               |   |
|---------------|---|
| Plugin Name   | ModuleInputBrowser  |
| ClassName     | plugin.grid.gridDisplay   |
| call argument | zoom extent   |
| activated by  | NGMS explorer toolbar   |
| Task          | This plug-in is an extended version of the GridDisplay. It allows module instance browsing and provides meta data views and displays for each associated parameter/data set |

#### The user's perspective

The ModuleInputBrowser will be a modification to the GridDisplay plugin. The GridDisplay will be used to present the static parameters associated to a module data set. The extension will be:

- addition of a navigation filter (similar to Explorer) to add additional levels for distinction of ModuleDataSets
- a panel to host
  - A MetdataViewer
  - A GridPointProperty Viewer, as a property list preferably supported by a graphical representation.

Figure 3-1 provides a possible outline of the new display.

Support for the following actions is foreseen:

| User action                          | Result  |
|--------------------------------------|---|
| Activate plug-in (from explorer)     | <ul style="list-style-type: none"> <li>• Plug-in activated with zoom extent as selected in Explorer.</li> </ul>   |
| Zoom extent selection                | <ul style="list-style-type: none"> <li>• The list of module data set will be updated (filtered) according to zoom extent.</li> <li>• The associated shape files will be shown as background on the grid/map</li> <li>• The default module data set associated to this zoom extent will be activated.</li> </ul> |
| Module data set selection/activation | <ul style="list-style-type: none"> <li>• The meta data associated to the filter level will be visualized in the meta data panel</li> </ul>  |
| Parameter selection                  | <ul style="list-style-type: none"> <li>• The associated values will be shown by colors on the grid</li> </ul>   |
| Grid point selection                 | <ul style="list-style-type: none"> <li>• The grid point properties will be derived and shown in the panel</li> </ul>  |

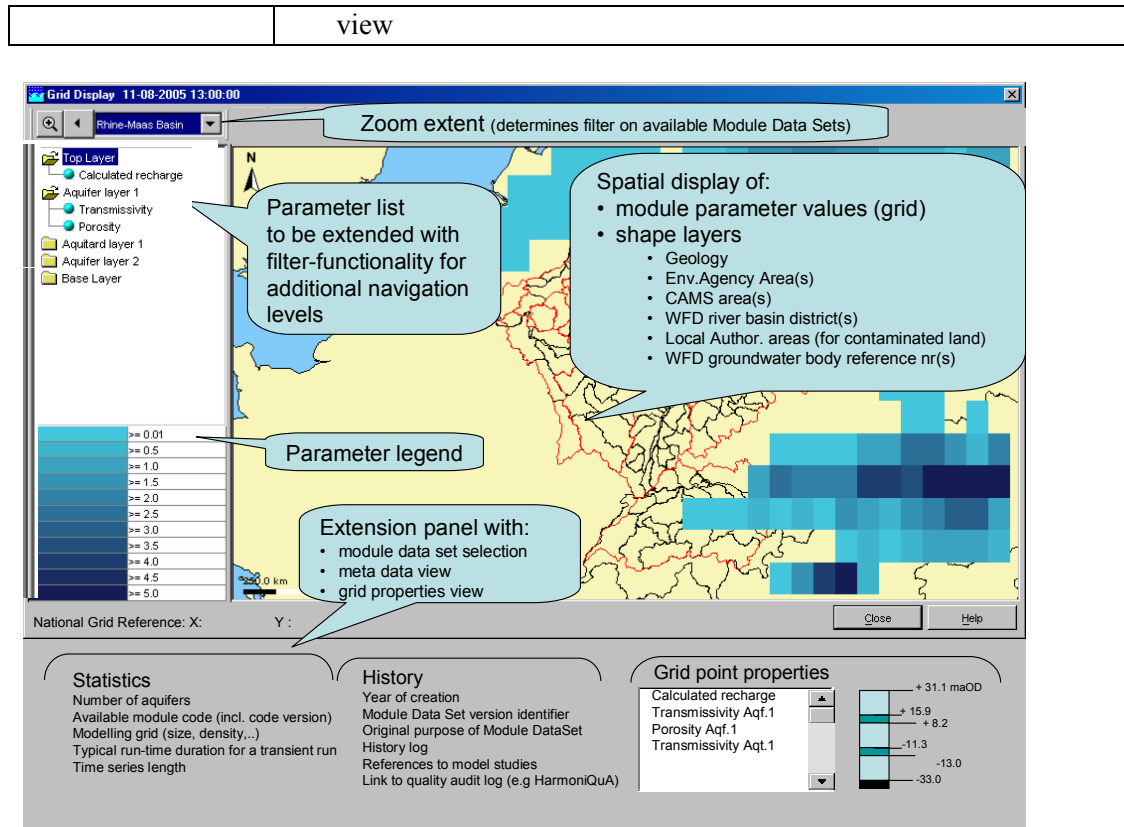


Figure 3-1 Illustration how the grid display might be extended for module input browsing

## The data perspective

- The navigation bare will be extended with the filter functionality of the Explorer. Filters can hold sub-filters or time series.
- Meta data will be associated to FilterIDs
- Each Module Instance, identified by a ModuleInstanceID, refers to a particular data set associated to a module. This ModuleInstanceID is a property of a TimeSeriesSet..
- All static parameter values that may vary over space will be handled as a grid time series, i.e. a mapstack with one timestep and a relativeViewPeriod that starts very early (e.g. 01-01-1900, exact date time to be defined). See pi\_mapstacks.xsd with only one time step).
- All static parameters values which are constant over space and time will be handled as module parameters (see pi\_parameter.xsd).
- For quasi-3D data sets (i.e. vertical layers of a 2D-grid), the vertical layering will be handled by sub-locations; i.e. the layer grouping will be identified by a locationSet-placeholder, while each vertical grid layer will be identified by a locationID-placeholder which is part of the locationSet. The name of the locationID holds a reference to the layer number (starting from top to bottom). In addition the z-property of the location position holds the layer reference. The exact vertical position of the bottom of each grid layer is for each grid cell defined by the coordinates of the grid cell centre points (x,y,z) (see pi\_cells.xsd).

## The class perspective

**Class ModuleInstanceMetaData:** holds the meta data of a ModuleInstance

| Main Methods      | Associated action   |
|-------------------|---|
| .Update(filterID) | <ul style="list-style-type: none"> <li>updates property list of filterID</li> </ul> |

**Class MetaDataViewer: viewer on ModuleInstanceMetaData**

| Main Methods      | Associated action   |
|-------------------|---|
| .Update(filterID) | <ul style="list-style-type: none"> <li>requests update of ModuleInstanceMetaData</li> </ul> |

**Class GridPointPropertyViewer**

| Main Methods   | Associated action  |
|----------------|--|
| .Update(pixel) | <ul style="list-style-type: none"> <li>transfer pixel position into cell position (i.e. x-y coordinate of a grid)</li> <li>obtain all parameter values of all grid layers (associated to this ModuleInstanceID) for this coordinate</li> <li>display properties</li> </ul> |

### Requested change from NFFS-plugin GridDisplay

- add filter functionality to navigation panel (similar to Explorer)
- enable panel to hold a MetaDataViewer and GridPointPropertyViewer
- enable hiding toolbar with time series play options
- notify refresh of zoom extent
- keep display position/zoom extent when selecting another parameter in the navigation list
- enable configuration switch which indicates that a double-click event on the grid should notify the GridPointPropertyViewer instead of the TimeSeriesDialog (i.e. the latter is the built-in behaviour)

### 3.2.2 ManualForecast → TaskRunDialog

Remark: Although the functionality is similar to the manualForecastDialog, this Dialog is likely to become a new plug-in.

#### Summary

|               |   |
|---------------|---|
| Plugin Name   | TaskRunDialog   |
| ClassName     | plugin.manualForecast.taskRunDialog or<br>(new) plugin.manualTaskRun.TaskRunDialog  |
| call argument |   |
| activated by  | NGMS explorer toolbar   |
| Task          | This plug-in specifies the task run instructions (what-if scenario, post-processing instructions etc.) and submits the job to the master controller. Its basic functionality is similar to ManualForecastDialog, but its layout is tailored to running what-if scenarios or post-processing tasks |

## The user's perspective

This dialog mainly utilizes instructions and data sets that have been fixed in the configuration or have dynamically been defined via other plug-ins. The instruction set holds:

- the workflow, i.e. which modules (including their data sets) have to run in what sequence)
- the what-if scenario, i.e. which boundary conditions have to be used
- the output specification, i.e. which output has to be produced (raw output, sliced output, post-processed output, reports)

As the list of workflows and tasks is extensive, it is recommended to introduce a treewiew for organizing the various workflows or tasks that can be run. For each task, associated task run properties may be selected. Figure 3-2 provides a possible layout for such dialog.

This dialog could/should be extended with an tab to collect a set of workflows into a batch.

The screenshot shows a dialog box titled "TaskRunDialog". At the top, there is a "Run description" text field. Below this, the dialog is split into two main sections. The left section, titled "Overview of available tasks and workflows", displays a tree view with a root node "Sample Node" and three sub-nodes, also labeled "Sample Node". The right section, titled "Selected tasks (ordered)", is currently empty. Between these two sections are two buttons: a right-pointing arrow ">" and a left-pointing arrow "<". Below the tree view, there is a section titled "Selected Task run properties" containing five dropdown menus for "What if scenario", "Execution control template", "Slicing template", "Post-processing template", and "Reporting template". Below these dropdowns is a checkbox labeled "Synchronize automatically". At the bottom of the dialog, there are "Schedule options" with two radio buttons: "Now" and "Later". The "Later" option is selected. Next to the "Later" radio button is a text input field. To the right of this field is a "Submit" button.

Figure 3-2 Illustration of potential TaskRunDialog

## The Data perspective

The TaskRun-properties will need to be extended to handle (references to) various new instructions, such as the execution controller template, slicing and post-processing templates.

### Requested change from NFFS-plugin ManualForecast

Although the functionality is similar to the ManualForecastDialog, this Dialog is likely to become a new plug-in.

- new layout
- accommodate combination of individual tasks

- accommodate selection of various task run properties
- NB these changes require modification of the TaskManager as discussed in the ADD-report

### 3.2.3 Forecast → DataDownloadDialog

Remark: Although the functionality is rather similar to the ForecastManager, this Dialog is likely to become a new plug-in for layout reasons.

#### Summary

|               |  |
|---------------|--|
| Plugin Name   | dataDownloadDialog   |
| ClassName     | plugin.forecast.DownloadDialog or<br>(new) plugin.dataDownload.DownloadDialog  |
| call argument |  |
| activated by  | NGMS explorer toolbar  |
| Task          | This plug-in accommodates selection of raw, sliced and post-processed data and reports to be synchronized between the central data-store and the local data store. |

#### The user's perspective

As data set from groundwater models are extensive in size, it is preferable to synchronize/download data only when needed. This dialog enables the user to select the data sets that have been produced by the central system and need to be downloaded to the local data store for visualization of reporting.

It is recommended to introduce a treeview for organizing the data sets, and associated meta-data (e.g. user, production run, size etc.).

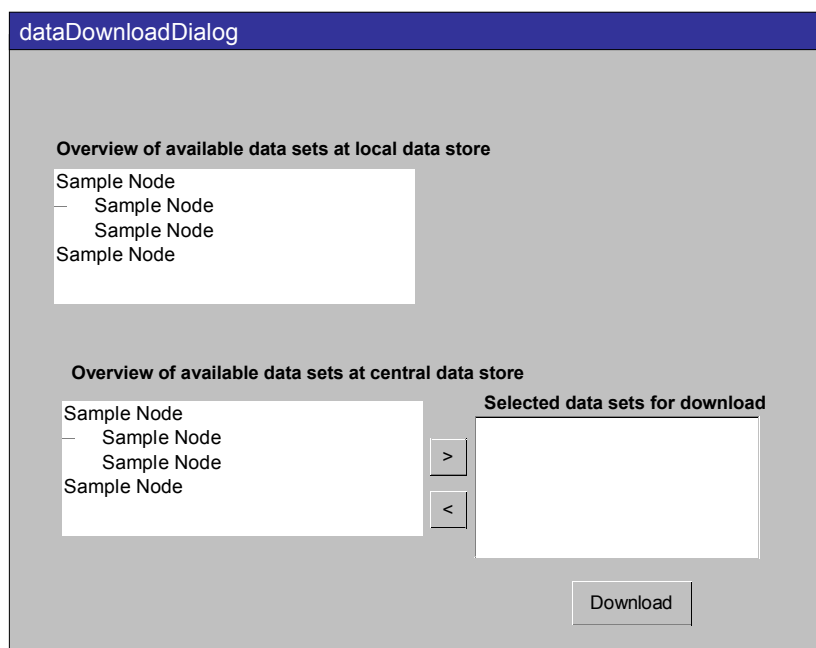


Figure 3-3 Illustration of potential DataDownloadDialog

Note: the visual functionality of this Dialog is similar for the selection of data sets to be archived. When properly designed, reuse should be feasible.

### The Data perspective

The TaskRun-properties will need to be extended to handle (references to) various new instructions, such as the execution controller template, slicing and post-processing templates.

### Requested change from NFFS-plugin Forecast

- new layout
- accommodate selection of individual data sets to be synchronized

NB these changes require modification of the DataSynchronisation API as discussed in the ADD-report.

## 3.3 New plug-ins

### 3.3.1 AbstractionLocator

#### Summary

|               |  |
|---------------|--|
| Plugin Name   | AbstractionLocatorDialog   |
| ClassName     | plugin.scenario.AbstractionLocatorDialog   |
| Call argument |  |
| Activated by  | NGMS explorer toolbar or Map/GridDisplay   |
| Task          | This plug-in enables the addition of new abstraction locations (or artificial recharge locations), including the associated to grid cell centre points.  |
| Comment       | This plug-in might become part of a ‘ScenarioWizard’ where all steps in defining a scenario (abstraction location definition, time series specification, etc.) are combined in one wizard type plug-in |

### The user’s perspective

The user starts the dialog, either by activation from the Explorer toolbar or by clicking in a Map or GridDisplay). The Dialog requires the user to specify the following:

- a location identifier
- the OS-coordinates in the physical system
- the computational coordinates

Note that the Ordinance Survey-coordinates identify the exact position of the well in the field, while the computational coordinates identify the grid point (or computational entity) where the abstraction time series is defined.

Figure 3-4 illustrates how the user might be supported with geological information (map and vertical profile) to support the specification of the exact position.



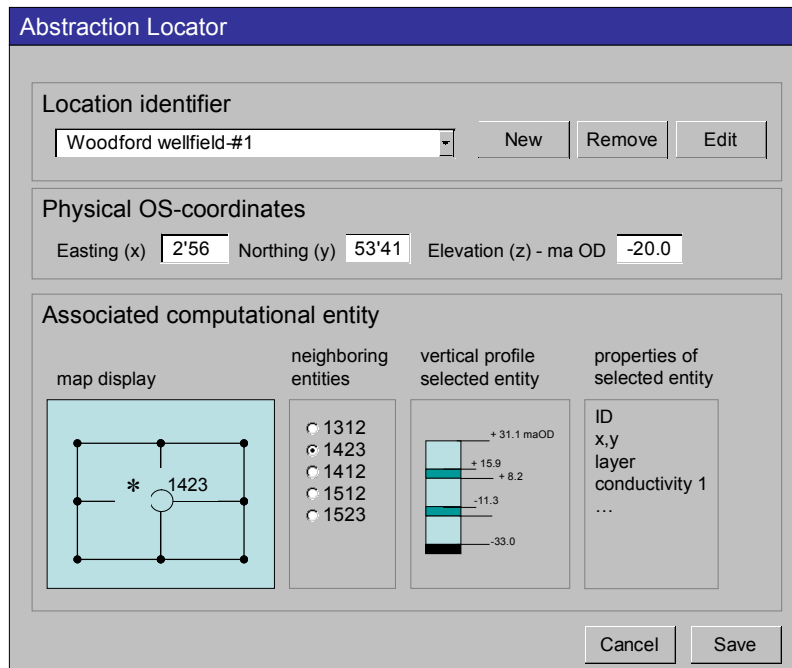


Figure 3-4 Illustration of a potential AbstractionLocator-dialog

### The data perspective

While the module adapter only needs to have information on the computational entity associated to the abstraction time series, the OS-coordinates are useful for the visual support. A lookup-table needs to be populated to associate those properties.

The lookup table should hold

- Location identifier(name)
- OS-coordinates (x,y,z)
- Computational coordinates (ID, x,y,layer)

The Abstraction time series itself will be captured in a grid time series object, which holds the computational coordinates of the associated location.

### The class perspective

The graphical representation of the vertical profile and the property view on the grid cell (computational entity) can be shared with the ModuleInputBrowser.

### 3.3.2 AbstractionTmsDialog / WhatIfScenarioDialog / ScenarioWizard

#### Summary

|               |  |
|---------------|--|
| Plugin Name   | AbstractionTimeSeriesDialog  |
| ClassName     | (new) plugin.scenario.AbstractionTimeSeriesDialog or plugin.scenario.WhatIfScenarioDialog  |
| call argument |  |
| activated by  | NGMS explorer toolbar or AbstractionLocator  |
| Task          | This plug-in enables the specification of a grid time series of abstractions by relating a time series to the abstraction location   |
| Comment       | This plug-in might become part of a ‘ScenarioWizard’ where all steps in defining a scenario (abstraction location definition, time series specification, etc.) are combined in one wizard type plug-in |

#### The user’s perspective

Within this plug-in, the user can select the abstraction location and specify the new time series or the modification of an existing time series. From a layout perspective, the user interface may function similar as the NFFS WhatIfScenarioDialog (see Figure 3-5). However, alternative dialogs, e.g. more tightly linked to the AbstractionLocator dialog could be developed as well.

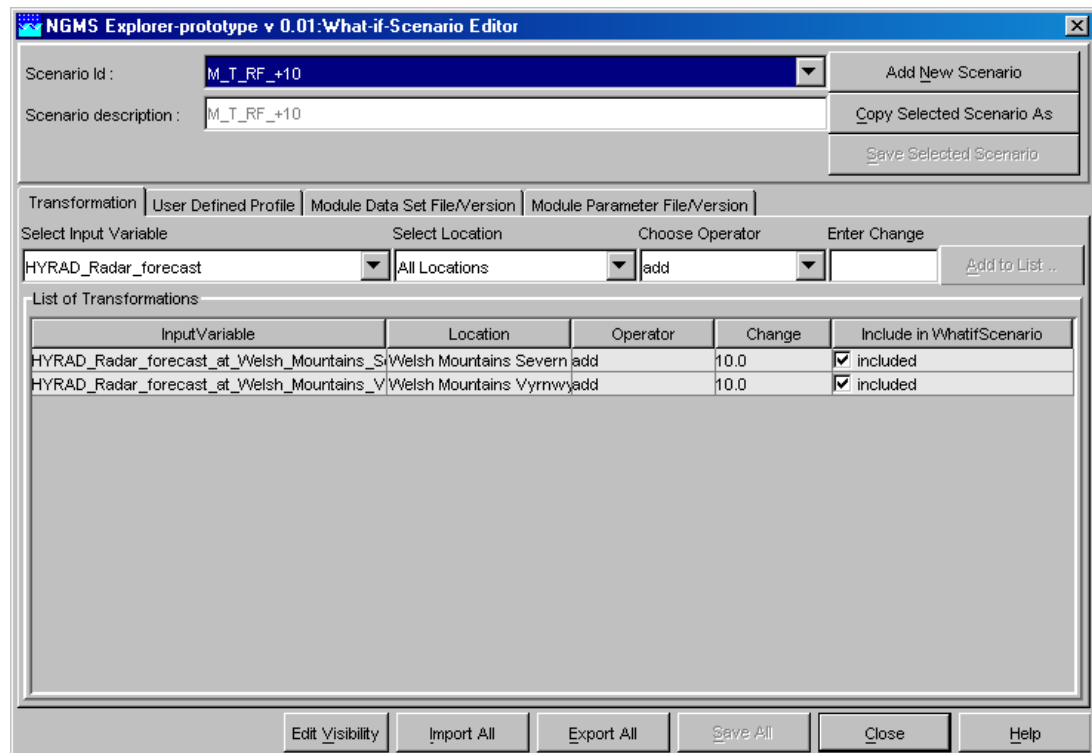


Figure 3-5 Illustration of the NFFS What-If Scenario Editor

## The data perspective

From the data perspective, the existing WhatIfScenarioDialog cannot be reused as this dialog requires that all abstraction points are in predefined in a locationSet.

Hence, functionality needs to be created which, per scenario, creates one grid time series with all abstractions (existing ones and new or modified ones). This requires that the GUI has to display the abstraction location identifiers from the lookup table, as defined by the AbstractionLocator. Once the user has defined the appropriate abstractions (i.e. selected locations and specified the associated time series), the data needs to be transferred into one grid time series-object which can be attached to a scenario.

## The class perspective

The plug-in should be designed in a way that the GUI is separated from the business logic.

The business logic could be divided in the following functionality:

- a class, which holds the lookup table and provides GridCell-coordinates for a selected abstraction location identifier
- a class which holds a grid time series-object with existing abstractions and adds the newly defined abstraction time series data (+ associated grid coordinates) to this grid time series-object
- specific attention should be paid to the ability to transformation time series of an (existing) single grid point without modifying all time series at all grid points in a grid time series-object.

## Relation with the What-IfScenarioDialog functionality

As indicated, it is expected that the additional functionality will make the WhatIfScenarioDialog too complex. If the AbstractionSpecificationDialog will fully replace the NFFS-WhatIfScenarioDialog, it should also enable:

- specification of a scenarios ID/name
- selection of recharge boundary conditions (transformable)
- selection of surface water levels as parameter sets (editable)
- selection of alternative module data sets (on the same grid)

## Combination in a ScenarioWizard

A step-by-step ScenarioWizard could be created by combining the AbstractionLocator with the AbstractionTimeSeriesDialog could be combined with the AbstractionTimeSeries and the other scenario dialog items into a ScenarioWizard (see Figure 3-6).

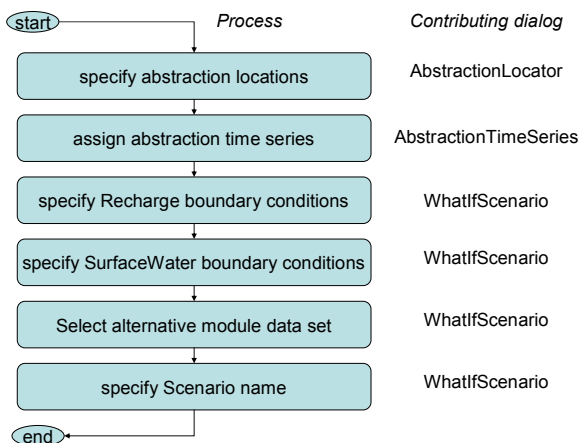


Figure 3-6 Steps in scenario specification (ScenarioWizard)

### 3.3.3 Slice Locator / SliceWizard

#### Summary

|               |   |
|---------------|---|
| Plugin Name   | SliceLocator  |
| ClassName     | plugin.slicespecifier.spatialSliceLocator   |
| Call argument |   |
| Activated by  | NGMS explorer toolbar   |
| Task          | This plug-in enables the specification of the spatial extent of a slice   |
| Comment       | This plug-in might become part of a 'SlicingWizard' where all steps in populating a slice template (input/output parameters, spatial extent, temporal extent.) are combined in one wizard type plug-in<br><br>Such SlicingWizard may also be activated from the TaskRunDialog |

#### The user's perspective

The SliceLocator is a map-based dialog with the groundwater model grid as a layer on the background. It should accommodate the specification of the spatial extent of a slice by:

- x,y-coordinates
- import of slice boundaries overlay from shape file (custodian can limit nr. of polygon overlays, as well as size of data slice).
- map based
  - multi-point selection (location set in form + pointed by mouse)
  - streamline (end node pointed by mouse)
  - entire model area
  - polygon sub-area (outer 3+ points pointed by mouse)
  - circular sub-area (centre pointed by mouse, radius by form edit)
  - upstream catchment area (downstream cell pointed by mouse, for top layer only)

Figure 3-7 provides an illustration of a potential slice locator.

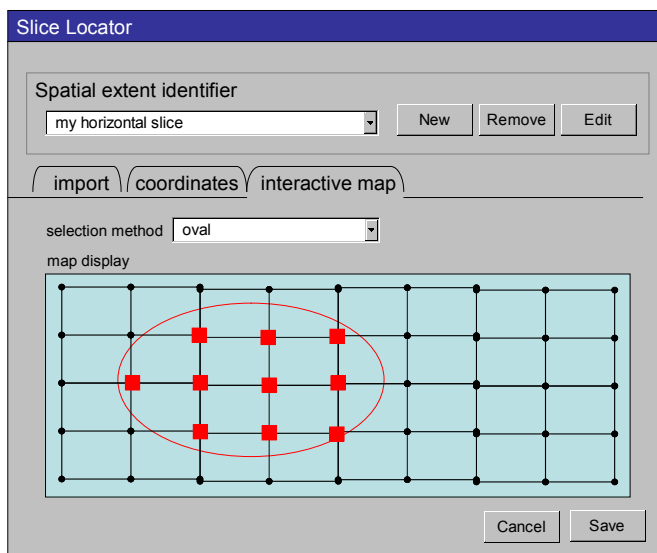


Figure 3-7 Illustration of a potential SliceLocator-dialog

Note that it has been agreed<sup>1</sup> that a slice only incorporates the cell centres that reside within the slice boundary.

### The data perspective

The ADD indicated that a SliceTemplate needs to hold information on the input and output time series (i.e. parameters), the spatial extent of the slice and the temporal extent of the slice.

The SliceLocator populates the spatial extent of the slice and assigns a name to the spatial slice. This spatial extent is held as an irregular grid definition (i.e. a collection of grid cell centre points).

### The class perspective

The business logic for this plug-in should be concentrated in a class/set of classes that have the following functionality:

- be able to collect all grid cell centre points that fit within the boundary specified in the GUI.
- store the points as a single object (the 'spatialExtent'-object within a slice template).

### Remark: diagonal slicing

In most cases a horizontal or vertical slice will be sufficient to meet user requests. In principle, the grid definition could hold a diagonal slice, while interpolation functions could be utilized to calculate the appropriate data values in such slice. From a GUI perspective it becomes much more complicated (and thus expensive) to create visualization which is

<sup>1</sup> Phase 2, third workshop (9 September 2005)

intuitive in defining such diagonal slices. Hence such sophistication is not incorporated in this design.

**Remark: ImportSlice plug-in**

The Import-tab in Figure 3-7 could hold the ImportSlice-plugin as foreseen to be developed in Phase 3. This plug-in allows the user to define the spatial extent by associating a shape-file to the slice. In the background, a SliceFeeder (class or set of classes) will pick up this file and transform it into a grid cell centre specification, holding all grid points that join the slice. Most of the business logic of the SliceFeeder corresponds to the business logic of the GUI described above.

**Extension from SliceLocator to a SliceWizard**

The presented SliceLocator does not yet fill the complete SliceTemplate (see ADD).

A step-by-step SliceWizard could be created to combine the SliceLocator with other dialogs to populate the slice template with input and output time series, as well as the spatial and temporal extent of the slice (see Figure 3-8). One might include task submission as well, thus enabling direct visualization of the (synchronized) data.

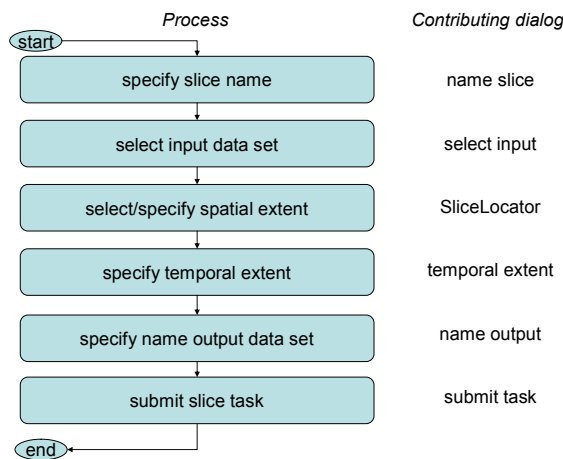


Figure 3-8 Steps in slice specification (SliceWizard)

### 3.3.4 OutputFilter

#### Summary

|               |   |
|---------------|---|
| Plugin Name   | OutputFilter  |
| ClassName     | plugin.outputfilter.outputFilterDialog  |
| Call argument |   |
| Activated by  | NGMS explorer   |
| Task          | This plug-in could be seen as a ‘pre-slicer’, i.e. it filters the parameters and temporal extent of the data being produced by a module. The plug-in populates a ‘raw output generation template’.  |
| Comment       | By default the computational code produces output as defined in its module control file.<br><br>This filter is only relevant if the associated module adapter can overwrite this module control file based on the information which is held in the raw output generation template (which is populated by the OutputFilter-plugin) |
|               |   |

#### The user’s perspective

The template holds a list of all output parameters that can be produced. This list is presented in a form to the user, which then selects the relevant output data. Finally, the user defines the temporal extent for which output needs to be produced. It is assumed that module control files can handle no spatial scoping of output production.

#### The data perspective

The underlying data object is the ‘raw output generation template’ as discussed in the ADD (section 3.5.1).

### 3.3.5 ModuleRun Journal

#### Summary

|               |  |
|---------------|--|
| Plugin Name   | ModuleRun Journal  |
| ClassName     | plugin.modulerunjournal.JournalDialog  |
| call argument |  |
| activated by  | NGMS explorer toolbar  |
| Task          | This plug-in accommodates viewing and/or editing the journal of (critical) module runs |

#### The user’s perspective

The module run journal provides the user and viewer some background information about critical module runs. It can be inspected to determine which module run needs to be access, i.e. which data needs to be synchronized.

Figure 3-9 shows the possible contents of a module run journal, as well as the actor which could provide the data to this journal

| Field  | Filled in by actor                     | When                              |
|--|--|-----------------------------------|
| Run name/Number  | TaskRunDialog (plug-in)                | at deployment                     |
| Date of upload   | Archiver                               |                                   |
| Date of last run   | TaskRunDialog (plug-in)                | after run                         |
| Module data set version  | TaskRunDialog (plug-in)                | at deployment                     |
| ModuleInstance (code) version  | Custodian /<br>TaskRunDialog (plug-in) | at configuration<br>at deployment |
| Comments:<br><ul style="list-style-type: none"> <li>Purpose of run.</li> <li>Brief summary of change from previous critical run version</li> </ul> | User/custodian                         |                                   |
| Custodian  | TaskRunDialog (plug-in)                |                                   |
| Standard Scenario Name   | User/custodian                         | at archiving                      |
| Reference to associated documentation (PDF)  | User/custodian                         | at archiving                      |
| Run start- end date  | TaskRunDialog (plug-in) ?              |                                   |
| Export of input files (incl. date + time created and by whom)  |  |                                   |
| Link to output files (incl. date + time created)   | Archiver                               |                                   |
| Agency project number  | User/custodian                         |                                   |
|  |  |                                   |

Figure 3-9 Contents of the module run journal

### The data perspective

The journal is a straightforward XML-file. In addition to above mentioned properties, the module run journal holds a reference to the ModuleInstanceID, as well as a reference to the associated ModuleDataSet ID

#### 3.3.6 Administrative plug-ins

Various administrative plug-ins, related to user account/disk quota management and system administration will be needed as well. These plug-ins are rather straightforward forms with list-boxes or property grids for editing.

#### 3.3.7 Configuration plug-ins

Currently NFFS configurations are managed with a ConfigurationManager, whose main task is to validate any configuration update. The editing of configuration files is mostly done in a text editor or an XML-support tool. Dedicated plug-ins might be desired to handle configuration updates in relation to uploading new module data sets, boundary condition sets and alike.