

Cross Compliance Assessment Tool (CCAT)



CROSS-COMPLIANCE ASSESSMENT TOOL

Policy-oriented research:
Scientific support to policies SSP

Specific Targeted Research Project (STREP)

Deliverable 5.3: Technical Design of Final CCAT Tool

Due date of deliverable: [25-09-2009]

Actual submission date: [25-10-2009]



Cross Compliance Assessment Tool (CCAT)

Authors of this report and contact details

Name:

Foppe Bouma

Berien Elbersen

Janneke Roos-Klein Lankhorst

Partner acronym

LEI

Alterra

Alterra

Contact details of first authors/editors:

Name: Foppe Bouma

Address:

Email: Foppe.Bouma@wur.nl

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Cross Compliance Assessment Tool (CCAT)

Disclaimer:

“This publication has been funded under the CCAT project, EU 6th Framework Programme, Priority 8.1 (European Commission, DG RTD, contract no. 44423-CCAT). Its content does not represent the official position of the European Commission and is entirely under the responsibility of the authors.”

"The information in this document is provided as it is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability."

Table of Content

I. Acronyms and Definitions5

1 General7

2 Objectives of the CC assessment final tool8

3 Architectural design of the final Tool9

 3.1 *The CCAT Client-Application*10

 3.2 *Application User Types*11

 3.2.1 *Application for End Users*11

 3.2.2 *Application for Researchers*11

4 Design of Model Framework Run System12

 4.1 *Preprocess Manager*13

 4.2 *Scenario Run Manager*13

 4.3 *Postprocess Manager*14

5 Design of Scenario Definition System16

 5.1 *Creating a new Scenario*16

 5.2 *Editing a new Scenario*16

6 Design of Output Presentation System17

 6.1 *Output Indicators*17

 6.2 *Output Data*18

 6.3 *Presentation types: Tables, Graphs, Maps*18

REFERENCES19

Table of Figures:

Figure 1. Overview of the CCAT Final Tool10

Figure 2. Model Framework Run System architecture of CCAT12

Figure 3. Scenario Run Manager architecture of CCAT13

Figure 4. Scenario Definition System16

Figure 5. Output Presentation System17

I. Acronyms and Definitions

	Abbreviation	Full term
A	ANSI	American National Standards Institute
C	CC	Cross Compliance
	CCAT	Cross compliance Assessment Tool
D	DOW	Description of Work (of CCAT)
E	EU	European Union
G	GAEC	Good Agricultural and Environmental Condition
	GAMS	General Algebraic Modelling System
	GDX	GAMS data exchange (format)
	GIS	Geographical Information System
	GSE	GAMS Simulation Environment
	GUI	Graphical User Interface
N	NUTS	Nomenclature of Territorial Units for Statistics
S	SMR	Statutory Management Requirements
W	WP	Work Package

1 General

The analytical tool to be developed enables the assessment of impacts of Cross Compliance (CC) given different implementation pathways and specific national and regional conditions. Due to the challenges outlined in the description of work (DOW) of the project the final tool will build as much as possible on relevant existing projects.

With special regard to the final users requirements the final tool will be implemented as a standalone desktop application that the user has to install on his or her computer. The installation package of the software tools is intended to be self contained, meaning that all required elements for the successful operation will be included.

This report aims to describe the technical design of the final tool based on the overall approach and selection of available models as described in other CCAT deliverables:

- Deliverables 2.1 and 2.2 : General approach to the assessment of the impacts of CC in the EU and list of indicators
- Deliverable 2.3: Report describing the operationalisation of the first selection of indicators into impact indicators for the implementation of these in the first prototype of the analytical tool.
- Deliverable 2.4: Report describing the data and information requirements to be provided by WP3
- Deliverable 2.5: Report describing the design of the analytical tool (prototype 1) specifying the types of output to be produced and the functionalities required to produce the final impacts
- Deliverable 2.6_7: Report describing the operationalisation of the second selection of indicators into impacts of Cross Compliance and the implementation in the final analytical tool
- Deliverable 3.2.3: Estimating compliance levels and costs of compliance
- Deliverable 4.1.1: Report on the design and development of the economic impact generator of CC
- Deliverable 4.2.1: Report on the design and development of the environmental impact generator of CC
- Deliverable 4.2.3.1: Derivation of EPIC meta-models to evaluate the impact of cross compliance measures on leaching and runoff of nitrogen and soil erosion at European scale
- Deliverable 4.2.3.2: Derivation of DNDC meta-models to evaluate the impact of cross compliance measures on leaching and emissions of nitrogen at European scale

- Deliverable 4.3.1: Report on the design and development of land use change, landscape and biodiversity impact generator as a component of the analytical tool
- Deliverable 5.1: Technical design of prototype 1

2 Objectives of the CC assessment final tool

One of the main objectives of the CCAT project is to make an interactive computer application to estimate the effects of a variety of Cross Compliance (CC) measures in terms of agricultural markets, producer's income, consumer's welfare, land use, soil, water, air, climate, biodiversity and landscapes, as well as food safety, animal welfare and health. The application should seek to integrate existing knowledge and models as far as feasible.

To start the creation a first prototype was worked out. This prototype focussed on the effects of the Nitrate Directive and included the CAPRI-MITTERA model chain (see for further specifications D4.1.1 and D4.2.1).

The prototype served the following aims:

- a) Verification of the collection of use-cases as provided by the various project participants
- b) Verification and testing of a real system integration between the models
- c) Performance tests & validation

Now the final version is being created. In this version most SMR obligations and all GAECs are included, and relevant indicators are added. For the description of these indicators and their operationalisation into the final analytical tool we refer to Deliverable 2.6_7. In this Deliverable 5.3 the technical design will be outlined.

3 Architectural design of the final Tool

The architectural design of the final CCAT Tool follows the (software) concept of loose coupling of modules¹. The graphical user interface (GUI) that is used in CCAT is a totally model independent GUI. But also the coordinating modules (model managers) which take care of the communication towards and between models are very loosely coupled². This is a very important feature which makes the model builder totally independent of the GUI and of all other modules and models in the system. Very loosely coupled systems have the added advantage that they tend to build more quickly and also can be maintained, reused and scaled more easy³.

In Deliverable D 5.1 of CCAT Prototype I was stated that CCAT should make use of an existing GUI named GSE (GAMS Simulation Environment). But in developing CCAT the need grew to build a new GUI which meets better the requirements of the final CCAT Tool than GSE. This new GUI named GModelBuilder is however also built on the GSE modelling philosophy⁴ and has extended model building, model linking and testing facilities and also better facilities for presentation of results, especially mapping features.

The interfaces between the GUI and the model(s) are controlled by meta-information files, batch files and process managers (to process the run-off of the models). The GUI can run on top of any (set of) model(s).

The GUI allows defining new scenarios, inheriting from previous ones, running those scenarios, and if needed run all defined scenarios in a row (during the night or in weekends if this procedure is very time consuming).

The GUI also allows to present results in Tables, Graphs and Maps. Tables and Graphs are presented by a multidimensional data viewer which allows exporting results towards a variety of formats like Word, Excel, etc. Maps are presented by means of GIS (e.g. Nuts2-polygons). A largely appreciated feature is scenario comparison.

The built-in model scenario control (with additional documentation possibilities) guarantees a full reproducibility of work done with the tool in the past.

The GUI (GModelbuilder) has been developed at LEI (Agricultural Economic Institute) to be used not only for CCAT but for all kinds of models.

¹ See: www.sei.cmu.edu/architecture

² ISO/IEC 42010:2007

³ Krause. A 2006

⁴ Dol.W & F. Bouma 2004

3.1 The CCAT Client-Application

The CCAT Client-Application has been built on the GModelBuilder software, being a model independent GUI. The actual functionality and layout of the GUI is defined by two definition files: an Ini-file and an Excel file, containing all kinds of content definitions.

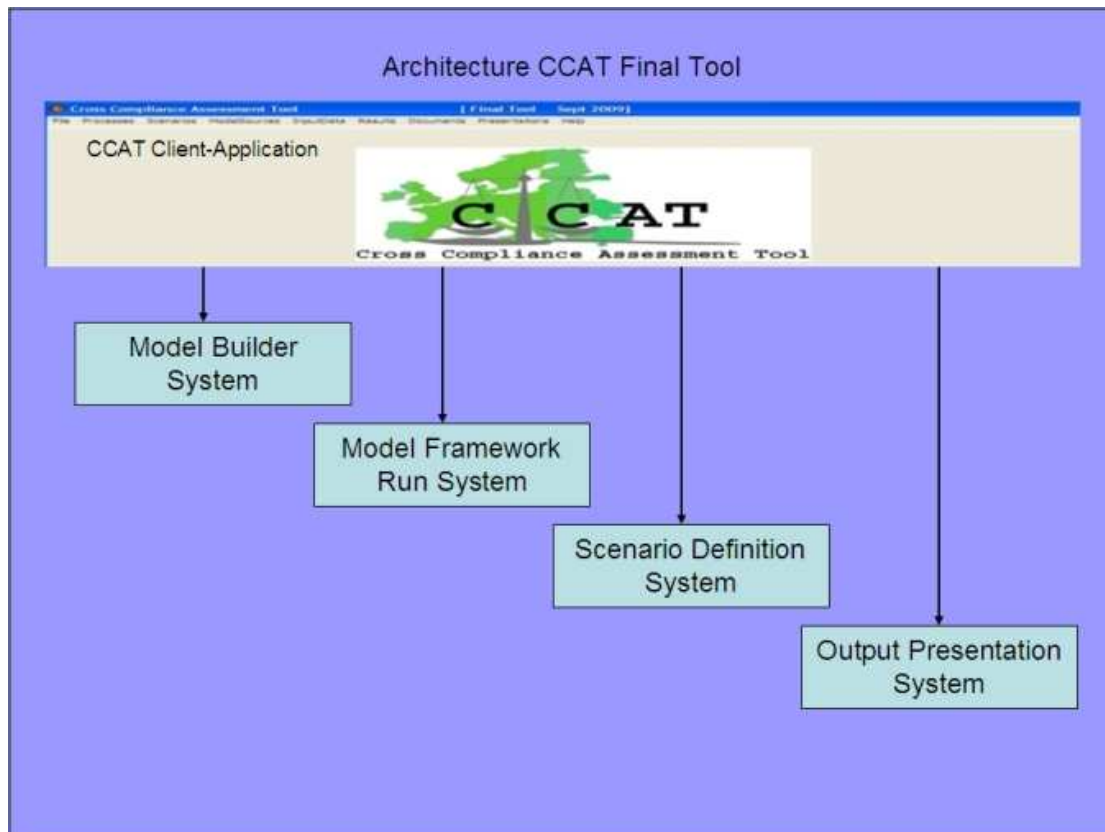


Figure 1. Overview of the CCAT Final Tool

The CCAT Final Tool has the following subsystems:

- Model Builder System: this subsystem allows the model builder(s) to build, run and test their respective models from within the GUI. The GUI calls the so called GamsTools toolkit⁵ to perform these activities, the toolkit contains GTREE and DataExplorer as model development tools. The Model Builder System will not be discussed in more detail in this deliverable (see discussions on user types in chapter 3.2).
- Model Framework Run System: this subsystem enables running (a set of) scenarios on the set of CCAT models. This subsystem will be discussed in more detail in chapter 4.

⁵ See Dol.W 2008

- Scenario Definition System: this subsystem enables defining new additional scenarios to be run on the CCAT models. This subsystem will be discussed in more detail in chapter 5.
- Output Presentation System: this subsystem enables to investigate the results of the scenario runs of CCAT and will be discussed in more detail in chapter 6.
- Some additional features: the GUI gives the user the possibility to investigate interesting or important input data used in the various models, and also gives access to various documents and presentations concerning the CCAT project.

3.2 Application User Types

The CCAT Final Tool comes in three flavours, so-called user levels. The CCAT tool can be used by three different user types: end users, researchers and model builders. Depending on this user type a user has less or more rights and possibilities to use various features of the tool. Restricting features to a specific user type prevents erroneous use (e.g. changing model code or settings). The CCAT Final Tool will be delivered in two user types: end users and researchers. The model builder user type, which has the most extended rights, will not be delivered and therefore will not be discussed further on.

3.2.1 Application for End Users

The Application for end users will mainly contain the Output Presentation System. End users can investigate the results of predefined scenarios and have the possibility to extract information from these scenarios to Word, Excel, etc. They do not have the possibility to change and/or rerun these scenarios, nor have the possibility to define and create additional scenarios.

End users however will have insight from within the tool in most documents and presentations and also some relevant input data.

3.2.2 Application for Researchers

The Application for researchers will contain the Output Presentation System, but also the Scenario Definition System and the Model Framework Run System. Researchers, therefore, can create and define new scenarios by inheriting from existing scenarios. And, of course they can run those new scenarios and investigate the results. However researchers cannot change the predefined scenarios, nor can they change the model version code of the various underlying models. How the Scenario Definition System works will be discussed in chapter 5.

Researchers will have insight from within the tool in most documents and presentations and also in all relevant input data.

4 Design of Model Framework Run System

The Model Framework Run System is the real heart of the CCAT Tool. Its architecture tells how the communication between various models of CCAT is established and how the model running system operates.

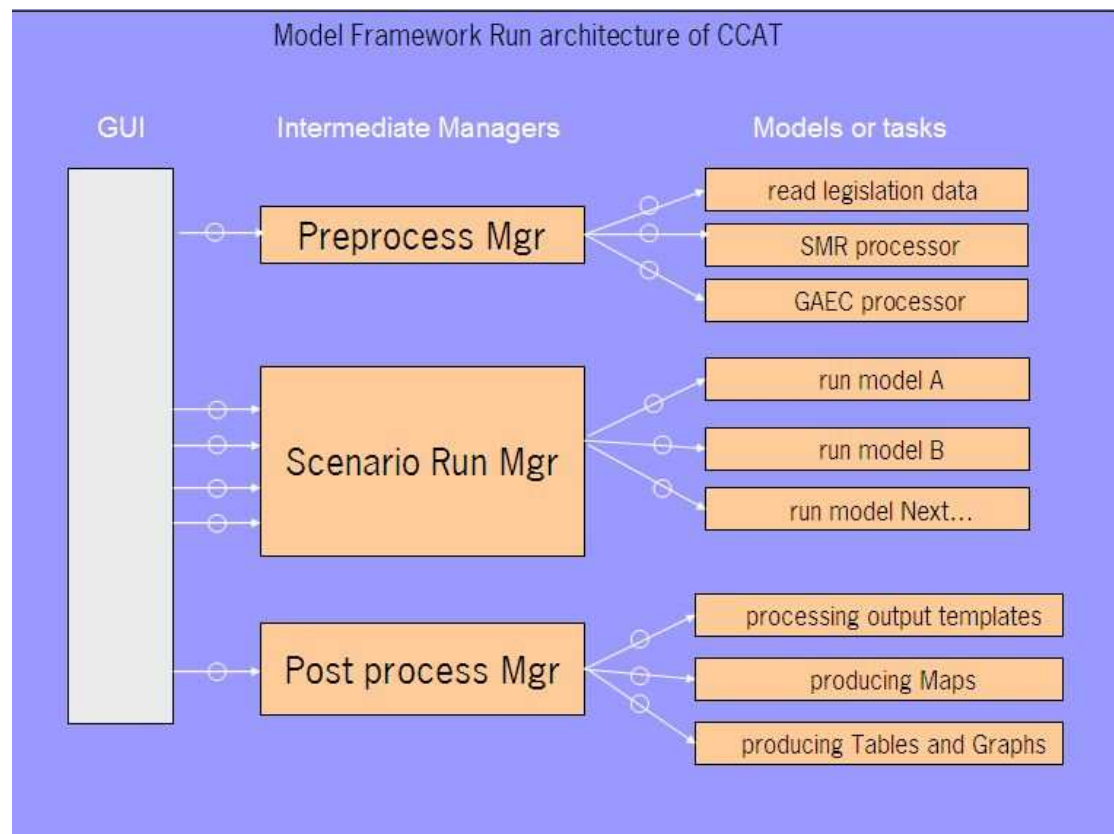


Figure 2. Model Framework Run System architecture of CCAT

The various models of CCAT are not directly controlled by the GUI but instead by means of intermediate managers as figure 2 shows. This architecture enables the GUI to stay model independent. The real specific knowledge of how to deal with the models now is incorporated in the intermediate managers.

Each modelling process needs three intermediate managers: a Pre-process Manager, a Scenario Run Manager and a Post process Manager. When starting the whole run process from the GUI the Pre-process manager runs only once, irrespective of the number of scenarios that will be (sequentially) run (see figure 2). Next one or more scenarios will be run by starting the Scenario Run Manager sequentially. Finally, after all (selected) scenarios have been run the Post process Manager is started by the GUI, only ones. The specific tasks of the three managers will be discussed in more detail hereafter.

The GUI enables the user to start all processes in a row, or to start just one process, e.g. the Pre-process or the Post process. It is also possible to start only a selected set of scenarios.

The modular approach, as discussed in chapter 3, enables the model builder even to start the processes outside the GUI.

4.1 Preprocess Manager

The Preprocess Manager governs tasks that have to be done first and only once irrespective of the number of scenarios that will be sequentially run. Performing those tasks results in information needed by all scenarios.

For CCAT these tasks consist of:

- Reading collected data on SMR and GAEC legislation into a Gams readable form.
- Processing raw SMR information into CCAT model information.
- Processing raw GAEC information into CCAT model information.
- Running the EPIC metamodel for inner reference scenarios (see Deliverable 4.2.3.1).
- Running the DNDC metamodel for inner reference scenarios (see Deliverable 4.2.3.2).

4.2 Scenario Run Manager

The Scenario Run Manager governs the running of just one scenario by starting the respective CCAT models in a predefined sequence, being responsible for the communication between the models (figure 3).

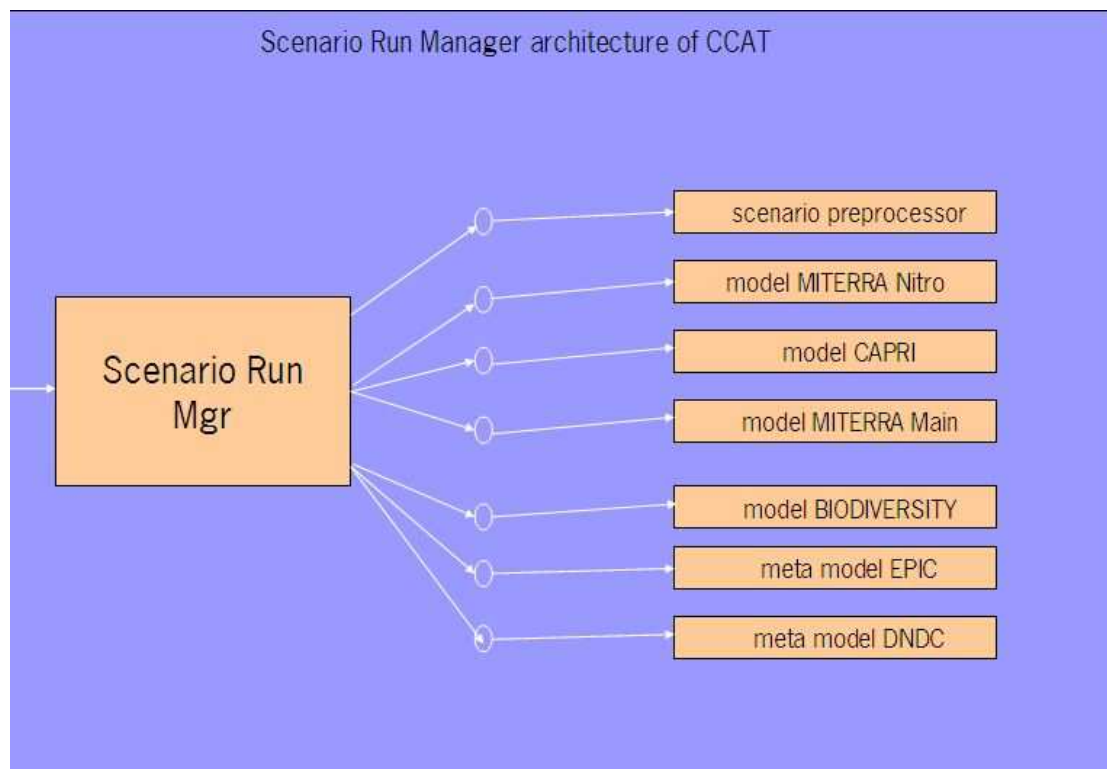


Figure 3. Scenario Run Manager architecture of CCAT

The Scenario Run Manager performs a whole list of tasks which can be described as follows:

- Receiving scenario information from GUI (not in figure 3)
- Translating scenario information for models to run. Information on chosen scenario options as well as different input data.
- Transporting information towards models. In the Scenario Run Manager architecture a specific Data Exchange directory has been designed for that purpose.
- Running models sequentially. (No recursive iteration of models is foreseen).
- Translating information between models. This is in particular needed if the meta information of data is differently phrased (e.g. different nomenclature for NUTS2 regions) or if different model software is used. Mostly GAMS⁶ is used as model software, except for DNDC which uses R⁷ as software. Therefore here a data transformation is required.
- Receiving output from the respective models. This output is defined by a list of output indicators (see further chapter 6.1).
- Transforming the output in a uniform nomenclature for the GUI. All results to be shown in the GUI should have the same definitions and phrasing irrespective of the specific use of nomenclature in the various models.
- Adding the output of the current scenario to the output collection of all scenarios. This output collection resides in one database (a.gdx file⁸) and will be used by the Post process manager.

It is interesting to note that the overall modular approach enables the model builder even to start its own model separately from the manager (the small circles in the middle of the arrows – figures 2 and 3 - stand each of them for a batch file). For further information on the models themselves and the actual content of the model interactions (see Deliverables 4.1.1 to 4.3.1).

4.3 Post process Manager

The Post process Manager governs tasks that have to be done after a number of scenarios have been run in order to prepare the results to be shown by the GUI. For CCAT these tasks consist of:

- Reading output collection of all scenarios and calculation differences compared to a chosen reference scenario.
- Starting a specific devised subprogram⁹ which reads in a predefined list of indicators (see par. 6.1) in order to prepare template files tailored for each indicator individually. These templates are used by the GUI to present tables, graphs and/or maps.
- Starting the production of maps in PNG format according to the map definitions in the templates (optional).

⁶ GAMS is a language for modelling: see www.gams.com

⁷ R is a programming language for statistical computing: see www.r-project.org

⁸ A.gdx file is a binary file format in the GAMS software system

⁹ Written in Delphi, a programming language of Embarcadero: see www.embarcadero.com

Cross Compliance Assessment Tool (CCAT)

An interesting feature is that the user can change the reference scenario to be used in the Post process. In doing so only the Post process has to be rerun. The output of the scenarios themselves (even when rerunning) will not change.

5 Design of Scenario Definition System

The Scenario Definition System gives the researcher and the model builder the opportunity to create and define new scenarios (figure 4). Furthermore, this system enables the user to describe the meaning and purpose of a scenario in text boxes.

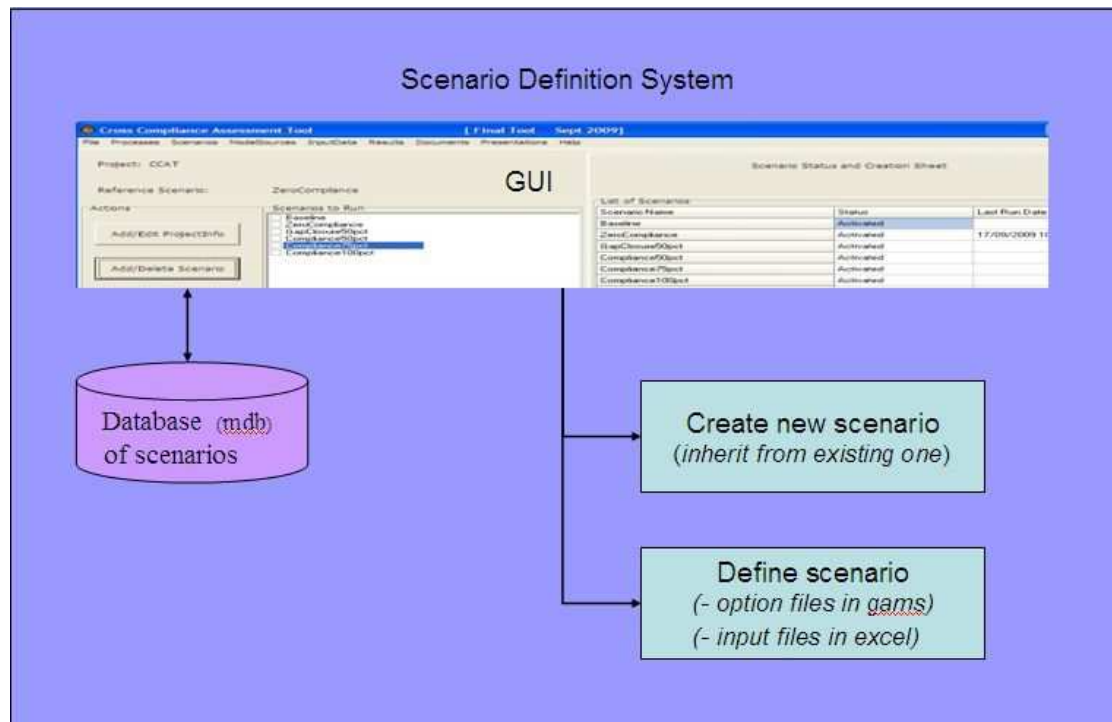


Figure 4. Scenario Definition System

5.1 Creating a new Scenario

The Scenario Definition System emerges by choosing the tab sheet “Scenarios” of the GUI. Here the user will find an option to create a new scenario by giving it a name. Next the user has to choose of which existing scenario he will inherit all scenario files. By choosing an already existing scenario which is most similar to the proposed content of the new scenario the user saves time in defining the new scenario. Having chosen the scenario to inherit from, the new scenario subdirectory will be created and all scenario files are copied from the inherited one.

5.2 Editing a new Scenario

Once the new scenario system exists the user can define the scenario by editing its scenario files. In the CCAT Final Tool these files are all excel files. These files can be accessed from the GUI by choosing the scenario to edit in the “Scenarios to Run” box. Next, by clicking on the button “Edit Scenario Run Settings” a list of excel file descriptions pops up. By clicking on a description the user enters the corresponding excel file to edit.

6 Design of Output Presentation System

The Output Presentation System is available to all user types: end user, researcher and model builder. This system gets its input from a predefined list of so-called indicators and the results of all scenarios delivered by the Post process Manager.

To get access to the output content of an indicator the user chooses the tab sheet “Results” and a box of “Available Indicators” shows up. Then by choosing an output type (table, graph or map) and clicking somewhere in the tree of available indicators the desired output information will be shown (see figure 5).

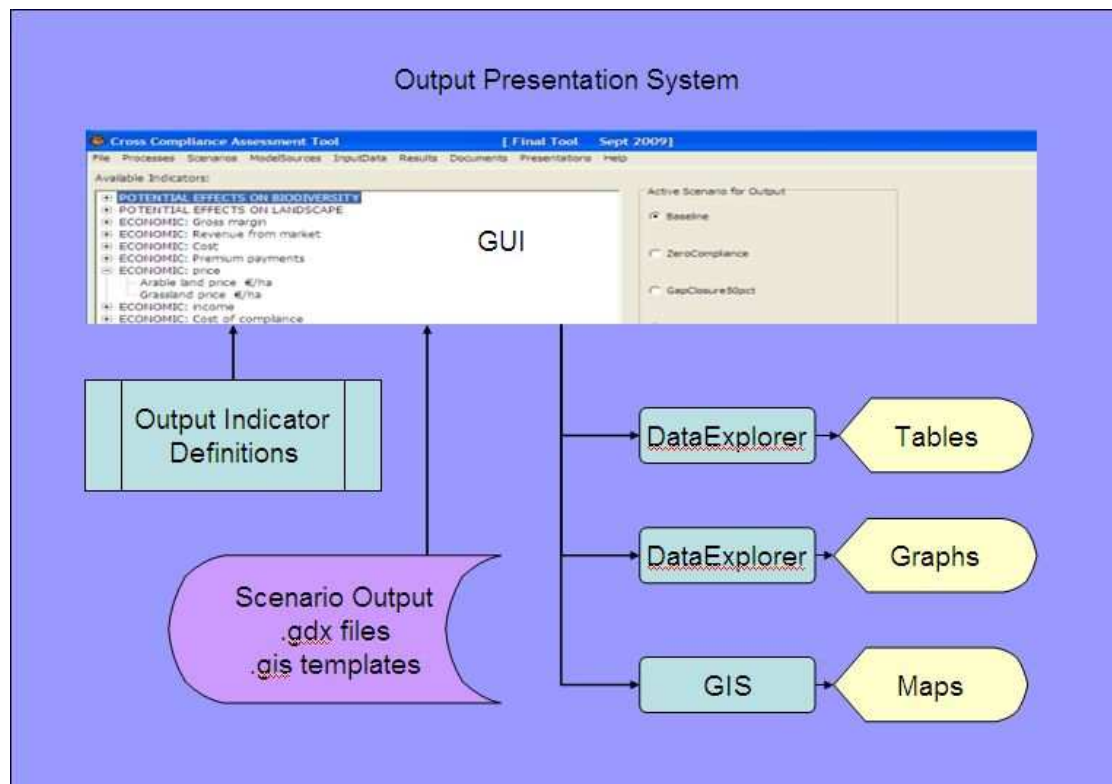


Figure 5. Output Presentation System

6.1 Output Indicators

All output of interest to the users of the CCAT tool is predefined in an Output Indicator Definitions table, which resides in an excel file. This table defines the name of an output variable as it is used in one of the models and its description used in the GUI. The description consists of a group name and a specific variable description. Furthermore the output file (a.gdx file) which holds the variable values is indicated in this table as well as a set of issues used to create template files.

The list of indicators is used by the GUI to build the Tree of Available Indicators.

6.2 Output Data

The output data of CCAT consists of the indicators delivered by each model in CCAT, as well as indicators created as calculated absolute and relative differences of all scenarios compared to a chosen reference scenario. This calculation is done by the Post process Manager. All output indicators reside in.gdx files.

Furthermore the Post process Manager creates for each single indicator a template file, using the information in the Output Indicator Definitions table. These template files are used by the GUI to decide how to layout each individual indicator (indexes in front, colours, boundaries, etc ...) when showing a table, graph or map.

6.3 Presentation types: Tables, Graphs, Maps

The Output Presentation System contains three kinds of output possibilities: Tables, Graphs and Maps. In CCAT each indicator can be shown in those three forms.

With tables and graphs all scenarios of an indicator are shown together and the user can restrict the information shown by choosing a specific country in the GUI. The GUI calls the DataExplorer tool to show tables and graphs.

With maps only the output of one scenario can be shown and the user has to indicate in the GUI which scenario he wishes to investigate. The GIS tool is used by the GUI to present the indicator.

For both tools, DataExplorer and GIS, the pre-processed template file is used to layout the indicator.

Finally, the GUI gives the user the opportunity to choose the desired output value type: indicator value of the chosen scenario(s) or absolute or relative differences to the reference scenario.

REFERENCES

Dol, W (2008) User documentation on DataExplorer; User documentation on GTREE (2006) LEI-WageningenUR, The Hague.

Dol, W & F. Bouma (2006) The GSE philosophy: a concept of model building as a team activity. LEI-WageningenUR, The Hague

ISO/IEC 42010:2007, Systems and Software Engineering -- Recommended practice for architectural description of software-intensive systems. (Note: ISO/IEC 42010: 2007 was formally known as ANSI/IEEE 1471-2000, Recommended Practice for Architecture Description of Software-Intensive Systems.)

Krause, A (2006): Einführung eines GIS für die Landwirtschaftsverwaltungen der BRD auf Grundlage EU-rechtlicher und nationaler Verordnungen. Ibidem, Stuttgart