MicroWave: a generic framework for micro simulationbased *ex ante* policy evaluation[1]

Jacques Wolfert*, Jan Lepoutre**, Wietse Dol*, Steven Van Passel**, Hennie van der Veen*, and Foppe Bouma*

*LEI, Agricultural Economics Research Institute, Burgemeester Patijnlaan 19, P.O. Box 29703, 2502 LS The Hague, The Netherlands
**Stedula, Policy Research Centre for Sustainable Agriculture, Potaardestraat 20, B-9090 Gontrode, Belgium

Abstract

This paper presents the MicroWave approach that has been developed to improve the process of modeling in the context of micro simulation. It leads to a more efficient model development, better quality of models and their output and improvement in knowledge management. A conceptual framework has been developed and translated into a hierarchical structure of GAMS program code. Besides, several software applications and other tools have been developed for support. These products are presented and some examples illustrate how MicroWave can be applied. MicroWave is especially useful in interdisciplinary research in which different persons are involved in the modeling process and when different models have to be combined.

keywords: Micro simulation framework, reusable models, knowledge registration, model quality

1 Introduction

In agricultural economic research, models are used to support policy evaluation, in particular *ex ante* evaluation. LEI has a long tradition in using micro simulation models that simulate individual farms and aggregate the results to (sub)sector level. Existing examples of these types of models are FES (Mulder, 1991, 1994) and APPROXI (Hennen, 1995). FES focuses on financial-economic simulation but includes several specific extensions, e.g. for simulations concerning Agenda 2000 and the Mid Term Review. APPROXI includes farmer's behavior in a more detailed way allowing different styles of farming and was until now mainly focused on environmental effects of manure policy.

FES, APPROXI and other LEI models are models that are based on the micro (farm) level. They are simulating the same phenomena, but do this in their own way and they distinguish themselves by focusing on specific aspects. There is a growing need for combining these different models in a flexible way, urged by the EU *cross compliance* policy (see e.g. the Seamless project; http://www.seamless-ip.org/index.htm).

In the past, attempts were made to refurbish existing models and to make a quick (and dirty) coupling between them. These attempts were often hampered by problems caused by:

- poor model structure, because researchers themselves did both the modeling and programming work and no 'technical' modeling experts were involved;
- poor documentation of models;
- lack of a sound and complete mathematical description – very often the contents of the model are tied together with the software environment and calculations are only available in program code or spreadsheet formulas;
- the fact that model knowledge is bound to one or a few person(s) and the model is rarely subject to peer review – with the risk that most knowledge is lost when a person leaves the institute or project;

[1] To be presented at the 89th EAAE Seminar, 3 – 5 Feb. 2005, Parma, Italy

These problems are identified by many authors in all kinds of scientific research where which simulation models are used (see e.g. Scholten et al., 2000). If these problems are not solved, coupling of models will be like coupling two black boxes, leading to results for which it is very hard to guarantee their quality. This coupling is not a sustainable and efficient solution in the long term.

Beside the wish to combine models, there is also a need to extend the application possibilities of these models. For example, the way of modeling investment and financing decisions for manure application measures in dairy farming should principally not be different from modeling the same phenomena for energy-saving measures in horticultural firms. The difference lies in the data input and coefficients but not in the calculation itself. The same is true for application of such models in different regional or international contexts. Many EU countries, and also countries worldwide, are faced with very similar problems. A higher efficiency can be obtained by reusing existing models or parts of them.

This search for efficiency drew the Flemish Policy Research Centre for Sustainable Agriculture (Stedula) to LEI, resulting in a joint plan to develop micro simulation models. Together they worked on a new approach that had to tackle the aforementioned problems. This approach should integrate the following topics:

- model development and coupling of models can be done in an efficient way;
- knowledge that is locked up in models can be easily shared and reused;
- the quality of models and their output can be assured.

This paper describes this integrative approach, which was called *MicroWave*. The *Micro* stands for micro simulation models and the *Wave* is the result of the complete new approach of model building. Also a microwave oven can be seen as a generic tool that can do all kinds of useful things in a household.

First, a conceptual framework for micro simulation is described, that formed the basis for the MicroWave approach. Then a total picture of the MicroWave approach is outlined and after that each part is described in detail. Next, the MicroWave approach is illustrated by several examples. Finally, the approach is discussed and several conclusions are drawn.

2 Conceptual framework development for micro simulation

2.1 Development and requirements

A farm simulation model is the core of each micro simulation model. Instead of each model having its own farm model, it would be more efficient to have a generic farm model that is valid for every type of farm and can be used for every research theme that is applicable to micro simulation. Such a generic model is usually referred to as a framework. In MicroWave, the framework comprises the basic structure for a micro simulation application and can be easily tailor-made for specific situations.

To define the framework, several activities were carried out: desk research, interviews, brainstorm sessions and workshops in which interaction took place between people that can be roughly divided into three categories:

1. **domain experts** – having in-depth knowledge about specific topics (e.g. manure application, energy use); they tend to model exceptions for each specific situation in their domain and want to include many details; usually they are also model users and have close contacts with clients of applied research and therefore want their models to be practical;
2. **technical model experts** – often having a background in mathematics and/or computer science and always looking for the possibility to describe something as compact and abstract as possible;
3. **scientific methodologists** –always looking for the best methodology for describing or simulating phenomena and keen on scientific approval of these methodologies; usually they are less interested in practical application of models.

From these activities, requirements for the framework were identified of which most important are:

- **Micro simulation**. At an aggregated (sector) level, the framework model should be able to simulate the dynamics of a set of indicators based on several exogenous factors for a given time horizon (often about 3 to 15 years), usually resulting in several scenarios. It should account for the diversity between farms connected with the diverse risk perception among farmers. So, it must be able to include farmer's behavior. The framework model should include the order and impact of policy measures for each individual farm in order to get a better idea of when and how a farmer will react on certain exogenous factors. The framework model should also account for differentiation between short and long-term policy measures.
- **Farm model**. The farm model must be simple and transparent. Detailed, in-depth knowledge must be handled by substitutable modules, connected with a knowledge base. The model should account for the fact that farms can be part of a larger enterprise and can consist of several establishments and divisions. At the same time, a household can be distinguished that is sometimes partly and sometimes totally connected with the farm; the model should at least account for off-farm income.
- **Input**. The main input of the model must be a farm accountancy database, but the framework should not be dependent on a very specific database. Modelers must be able to use a detailed database (e.g. LEI-BIN) or a simple database (e.g. EU-FADN). The starting point is that the minimum level of input is the data provided by the FADN database, which is the level of detail that is collected by all EU countries. To estimate individual farm dynamics and farmer's behavior, it is desirable that the database contains farm-historical data.
- **Output**. The model should calculate effects by means of financial-economic and environmental variables. Another type of output variables are what measures, and to which degree, are taken by farmers.

2.2 Conceptual framework

From these requirements, a conceptual framework was defined that is depicted as a scheme in Figure 1. The following subsections describe this scheme in more detail; corresponding definitions of words in the figure will be in bold in the text. First, a quick overview of the scheme is provided.

A first structural element is the division between (1) data initialization (normalization) and (2) the iterative loop, separated by the dashed line in Figure 1. The actual simulation takes places in the iterative loop (*t, t+1, …t+T*), where T is equal to the simulation horizon. The state of an aggregated set of enterprises (indicated by the shadow boxes) changes over time as a consequence of external factors: public policy and autonomous developments. This is indicated by the transition of the left state block (*t*) to the right one (*t+1*), which is in fact an iterative loop that runs a number of times dependent on the time interval t and the simulation horizon *T*. How this transition takes place, is determined by the period simulation model, indicated by the curved arrows and entities in between.

2.3 Initialization and normalization procedures

The initialization and normalization procedure is separated from the actual simulation procedures in order to ensure that data input can vary depending on the type of data that is provided (accountancy network data, representative farms or other) and the data required by the specific model application.

The initialization and normalization procedures are responsible for preparing the data in such a way that the original state of the enterprise can be described and all necessary data for the exogenous state variables are available.

Input data is the collection of data provided to the model, consisting of at least the data that is collected in the FADN. **Enterprise models** translate rough data from the database to an array of "facts" that describe the original state of the enterprise. For example, from the total area (ha) of sugar beets and the total yield of sugar beets, sugar beet yield per hectare can be calculated, financial ratios can be calculated, based on the balance sheet, etc. The calculated and derived data, together with the original data, are "facts" that describe the state of the enterprise at the beginning of the simulation. If desired, normalization of input data, e.g. to obtain more representative results for the start situation, can be applied.

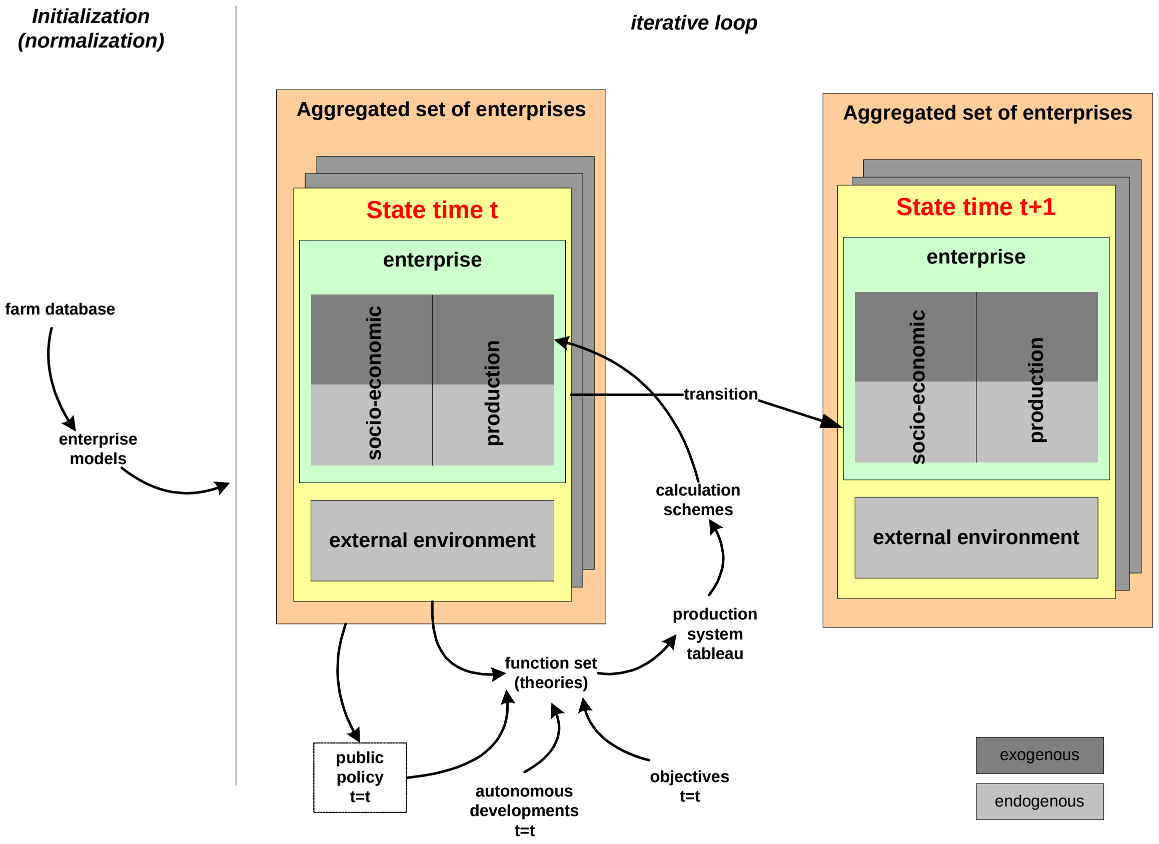


Figure 1 - The MicroWave conceptual framework

2.4 The enterprise as the central object

The **iterative loop** is the dynamic part of the framework, with the enterprise as its central object of simulation. Many effects concern only a part of the enterprise that might even play a role at different geographical locations, and because accounting for off-farm income was one of our requirements, a distinction is made between the farm production system and the household as an independent entity. Hence, an **enterprise** consists of a firm and a household, the former being the entity where production takes place, the latter providing capital, labor and the manager to the firm. The firm can consist of several establishments (parts of the firm in different locations), each of which in their turn can consist of several divisions. This is schematically represented in Figure 2.

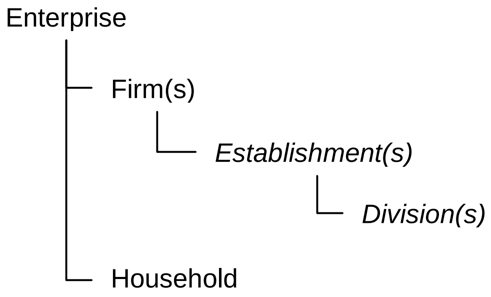


Figure 2 - The enterprise as the object of study and how it can be split up into different parts. An enterprise always consists of a firm and a household and can possibly exist of different establishments and divisions.

2.5 The state and transition of the enterprise: the iterative loop

The total simulation horizon *T* will usually vary between about 3 to 15 years and the time step *t* is equal to a production period *P*. In arable and animal production systems, *P* will usually be equal to one year, but in other systems, e.g. horticulture, this could be less then a year. In principal P is variant, although this should comply with the datasets that are used in a specific application.

Two discrete points are identified: at the beginning of *P* and at the end of *P*. At the beginning of *P*, state variables that define the situation at the start of production period are derived (e.g. capital, labor, current regulations). In the first year, they are derived from the database; in the next years they are derived from the previous state of the enterprise. These state variables are indicated by the exogenous part of the state variables in Figure 1. They are used as input for the 'function set'.

The **function set** uses facts from the state of the enterprise in relations and functions. These are timeless and not specific for an individual enterprise. The function set can be considered as a set of theories about enterprise development. Depending on the purpose of the study, or the theoretical assumptions on which the simulation is based, a set of functions is chosen. Examples of functions are: relations that link all production areas with the available area, production functions, and investment rules. Other input for these functions are public policies, autonomous developments and objectives of the farmer.

Public policies are sets of measures and regulations that are meant to direct the developments and behavior of enterprises and/or entire sectors towards a certain public goal. This means that governments can adapt a set of measures, depending on the individual or aggregated reaction of entrepreneurs to this. Public policies can either be introduced in the model as model input, or defined by the model itself, based on the state of the farm/sector at the beginning of period *P*. This is why the entity public policy in Figure 1 is surrounded by a dashed rectangle, which means that it can be based on either model input or calculated output from the previous year.

Autonomous developments are relations that can change elements of the state of the enterprise. Examples are technological novelties and improvements, expected price developments and market behavior.

Central elements in the decision-making process are the **objectives** of the entrepreneur. This can be simple profit maximization, or more complex, including e.g. risks minimization or minimization of external capital. Thus, autonomous developments, public policy and objectives are combined with the exogenous state variables to simulate the production and socio-economic decisions, which takes place in the function set. All functions eventually lead to a set of equations that are used to fill the production system tableau.

The **production system tableau** describes all inputs, outputs and related restrictions and links them with the objectives of the farmer, autonomous developments and public policy. The format will be similar to an LP-tableau, but will not necessarily be used for LP-purposes. Obviously, the dimensions of the tableau are variable, as the amount of functions applied to fill the tableau may vary as well. Once the tableau is filled, a number of calculation schemes are used to fill the endogenous state variables of the state of the enterprise.

Calculation schemes consist of calculations that calculate what is produced at the end of a production period P and the consequences for socio-economic state variables (e.g. financial results, solvency, etc.)

The full cycle of how the enterprise is simulated has now been described. To summarize, the state of the enterprise is defined as follows. The **state of the enterprise** describes the characteristics of the enterprise at time *t*. The enterprise is represented by endogenous and exogenous state variables and by state-specific external factors. **Endogenous state variables** represent the result at point *t* of the actions that were taken during production period *P*. **Exogenous state variables** remain constant during period *P*: they describe the state of the enterprise at the beginning of production period P and are exogenous to the decision-making process. Also exogenous to the decision-making process are the state-specific **external environment state variables** that are related to the enterprise, but which are not enterprise-specific.

State variables can be divided into socio-economic variables and production variables. Both socio-economic and production variables are subdivided in exogenous and endogenous variables. The modeler should decide to place each variable in one of these four quadrants.

Socio-economic exogenous state variables are exogenous state variables that refer to the socio-economic part of the enterprise. Examples are available labor, farm infrastructure, capital, etc.

Socio-economic endogenous state variables are endogenous state variables that represent the result of all socio-economic decisions. Examples are new investments, replacement investments, employ extra workmen, marketing decisions, etc.

Exogenous production state variables are exogenous state variables that include any information related to production. Examples are quota, enterprise specific manure regulation, nature restrictions, rotation schedules, etc.

Endogenous production state variables are endogenous state variables that describe the inputs used and the produced outputs. Examples of inputs are input of pesticides, seeds, concentrates, etc. Examples of outputs are milk production, fattening pigs, bovines, crop yields etc.

External environment is the environment which is faced by the enterprise, but which is not specific to the enterprise. For example, a farm can be located in a sandy region, but other enterprises can be located in the sandy region as well. Other external factors are location-specific manure or nature legislation, prices, etc.

2.6 Aggregation

At the end of each time step, **aggregation** takes place of variables that are relevant in the research concerned. Initially, aggregation will mean only adding up the results of the simulated farms in order to have an impression of the simulated dynamics at meso or macro level. The framework, however, also allows a link with meso or macro level. Ideally this link will lead to a dynamic interaction between decisions made by individual farmers and the effect of these decisions at the aggregated level. This is, however, beyond the initial scope of this conceptual framework.

3 The MicroWave approach – the total picture

Developing the conceptual framework was a major step in improving the modeling process for micro simulation. However, it is inadequate to guarantee well-documented and structured models of which the quality can be assured and that knowledge is shared and reused. Therefore, the framework should be translated into a software environment with predefined structures that guides a modeler to develop his model according to the MicroWave approach. Such an environment was developed resulting in several tools, documents and pieces of program code that are shown with their interrelationships in Figure 3. This scheme also indicates how they are connected with research questions. This total picture can be considered as the MicroWave approach.

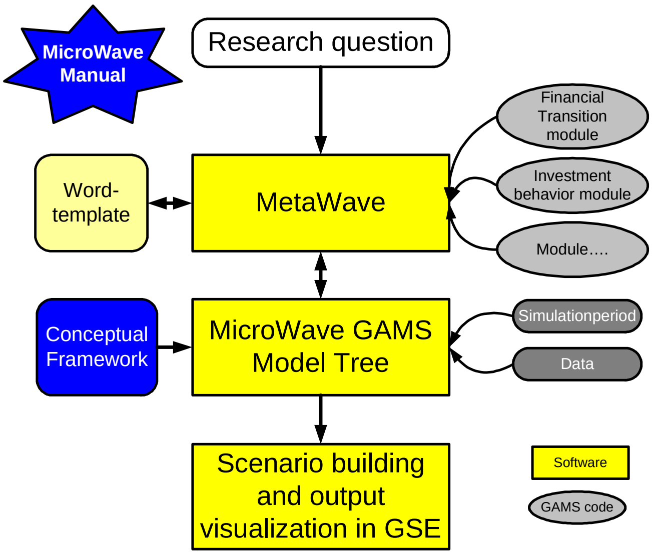

MicroWave Manual
Research question
Financial Transition module
Word-template
MetaWave
Investment behavior module
Module….
Conceptual Framework
MicroWave GAMS Model Tree
Simulationperiod
Data
Scenario building and output visualization in GSE
Software
GAMS code

Figure 3: The MicroWave approach: the total picture

The MicroWave conceptual framework was translated into the MicroWave GAMS Model Tree. Here, the actual simulation models are built in program code for a predefined simulation period and also actual data from a data source are read. The Tree is described in section 5. The software application MetaWave was developed as a tool to support knowledge registration and to search and browse through the modules that are built in MicroWave. Parallel to MetaWave, a Word template was defined that describes how modules should be documented. MetaWave is described in section 4. GSE is a tool that provides a user interface for GAMS models in which scenarios can be easily defined and analysed. GSE is an existing software application and was adopted in the MicroWave approach; it is described in section 6. The MicroWave manual is a document that describes a protocol on how the MicroWave approach should be applied by different users and at different stages; it is described in section 7.

When MicroWave is applied for using an existing model or developing a new one in order to answer a certain research question, MetaWave should be used to look for existing knowledge that can be reused or to register new knowledge. Then, the MicroWave GAMS Model Tree is used to build an actual simulation program for that specific research question. Finally, GSE can be used for running different scenarios with the model and to analyze the output. An illustrative example of this procedure is described in section 8.2.

4 MetaWave

When we started working on MicroWave, working together as a large group and everyone working on very specific elements of the framework, knowledge registration became of crucial importance. Registration of knowledge is the first step in sharing knowledge with others. There are two reasons for sharing or registering knowledge. First, by documenting your model, it becomes possible to peer review its scientific contents. Secondly, as a result of these documentation efforts, it becomes easier to reuse parts of a model in other models.

We started with a MS-Word template to document models (for a table of contents, see Figure 4). In MicroWave, we decided that all models would be documented according to this format and would be peer reviewed. Reviewing is done both scientifically and technically, i.e. the first review deals with the scientific contents, whilst the second looks at the technical implementation (computer code) of the model.

Module summary
1 Introduction
  1.1 Goal
  1.2 Calculation basics
  1.3 Domain and range
  1.4 Important Decisions
  1.5 Domain
  1.6 Relation with other modules
2 Organization of the calculation
  … Mathematical rules are described here …
3 References

Appendix 1 Sets
Appendix 2 List of used variables
Appendix 3 External data
Appendix 4 Comments by Scientific reviewers
Appendix 5 Comments by technical reviewers
Appendix 6 Test plan and results
Appendix 7 To-do list and history

Figure 4: Table of contents for the MicroWave model documentation

The team members of the MicroWave group have much experience and a well-established reputation in model building and it is no wonder that we got many documents describing models. There was a growing need to centralize the documentation of the models. To register knowledge we decided not only to work with a MS-Word template but also to build a software application called MetaWave (a contraction of Meta-information and MicroWave).

MetaWave is a database application. For each module, all information as indicated in Figure 4 is stored in a database. In a user interface, information can be viewed and modified. This can be done at several different levels as indicated in Figure 5.

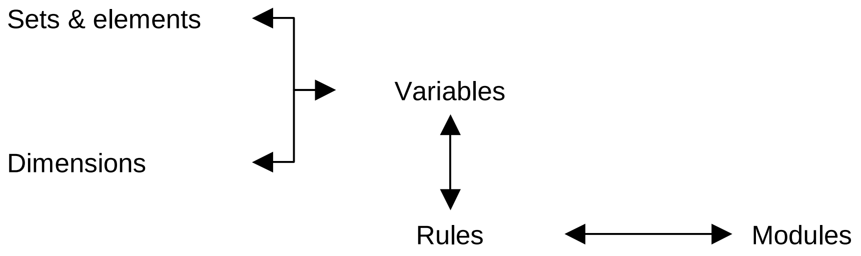


Figure 5 The information levels in MetaWave

The module level is the highest level in which information is represented in a coherent way. The core of a module consists of rules that represent a calculation. Rules consist of operators (+, – , =, etc.) and variables. Variables have a certain dimension and can be related to a certain set. In a micro simulation context, these sets relate to a domain the variables apply to. This division makes it possible that information can be reused at different places. For example, a variable can be part of different rules and rules can be used in different modules. The database however, takes care of consistency, because information of each element is stored at only one place. For each piece of information, associated meta information can be stored using regular editing facilities like text, graphics, tables, etc.

MetaWave makes it possible to browse and search through the information in a convenient way, because it is electronically available. However, occasionally (e.g. for peer reviewing) it would be more convenient to have an integral document of a module. Hence, a feature was developed that generates a Word document exactly in the same format as the Word template (see Figure 4).

MetaWave is not yet able to generate programming code automatically; at the moment only the sets and elements can be generated in GAMS code. However, we don't want MetaWave to become a compulsory tool in developing micro simulation programs. It is especially meant for those activities in which knowledge usage and sharing and discussion between experts with a different background (see section 2.1) is involved. Section 8 will illustrate how first gains can already be made without using MetaWave.

MetaWave is useful to register and reuse models, search for models and knowledge within them, and makes it possible to register knowledge in a uniform way (independent of model, model type, model builder). MetaWave facilitates looking at models and extending them by using them for new research questions. These features also make it a perfect tool to apply for ISO 9000 certification for model use and development.

5 The MicroWave GAMS Model Tree

So far, we emphasized the role of knowledge in the modeling process. MetaWave helps the modeler to put knowledge at the right place in the framework and make it subject to review. As mentioned in the previous section, MetaWave doesn't generate a complete set of program code, so it is necessary to define a structure for the GAMS program files that hold the actual program code, in accordance with the conceptual framework that was defined in section 2. This structure was called the MicroWave GAMS Model Tree. Besides, a software application, GTree, was developed that makes it easy to build up and visualize a tree structure of a set of GAMS files and at the same time provides an editor to read and modify the GAMS source code. GTree however, can also be used outside the scope of the MicroWave approach.

From the conceptual framework, the following major phases can be identified (also depicted in Figure 6):

1. Reading input data into a data structure;
2. Using data as input for different functions;
3. Simulations using a set of generic functions;
4. Putting outcomes of the ‘functions’ into the data structure

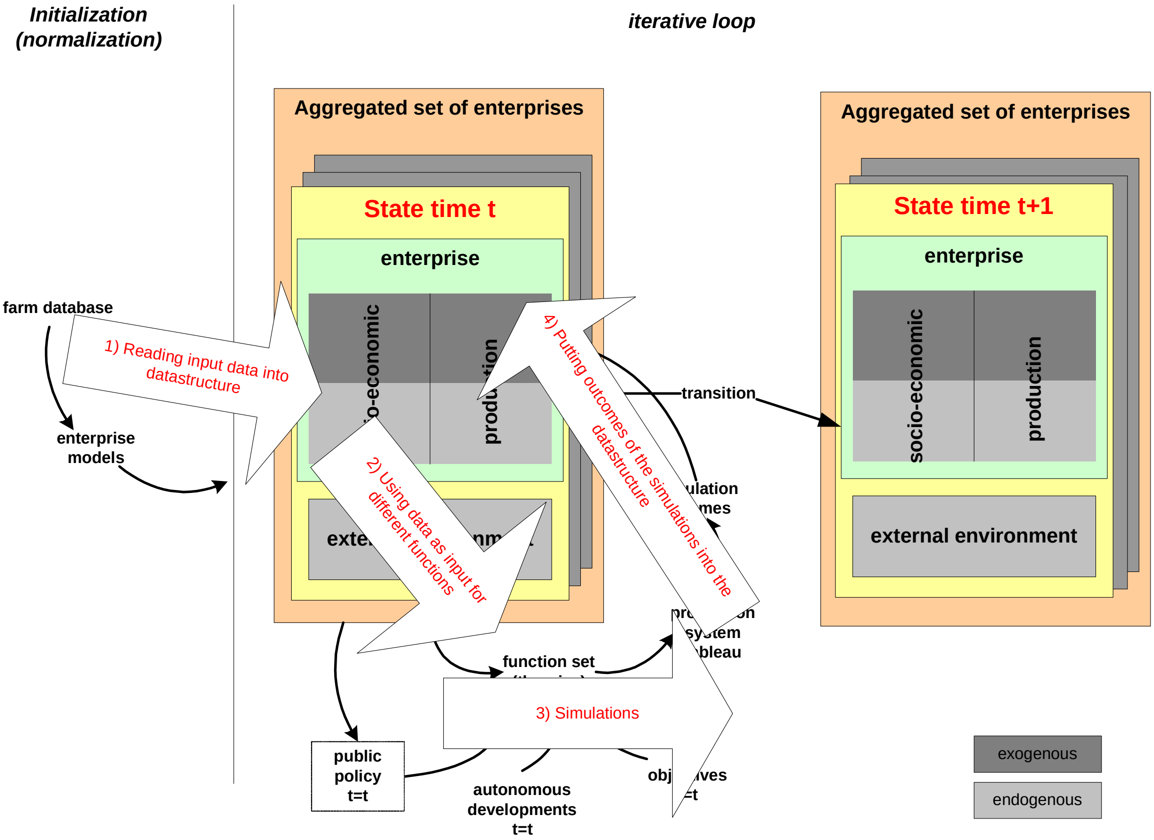
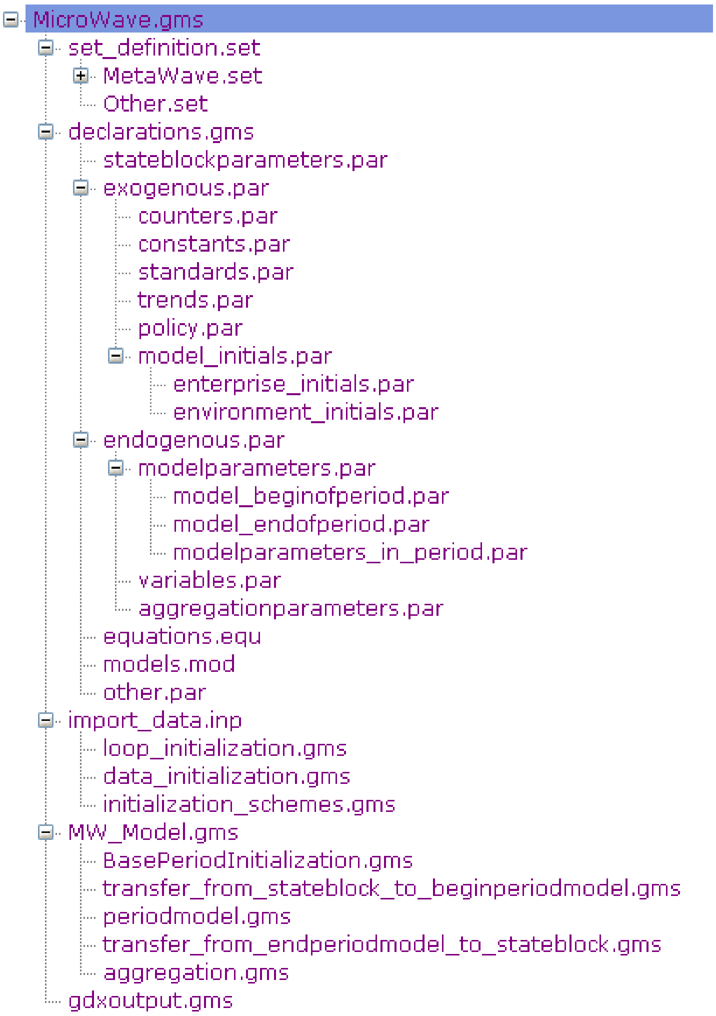


Figure 6 The four major phase for the GAMS tree in the conceptual framework

These four steps form the basis for the structure of the Tree. Besides, GAMS modeling syntax requires that all its consisting elements are being defined in advance: sets, parameters (exogenous and endogenous), variables, equations and models. From these requirements the Tree was defined in several branches and sub branches as shown in Figure 7. Each branch corresponds with a physical GAMS program file. The following paragraphs will describe only the main branches in more detail.



```
MicroWave.gms
    set_definition.set
        MetaWave.set
        Other.set
    declarations.gms
        stateblockparameters.par
        exogenous.par
            counters.par
            constants.par
            standards.par
            trends.par
            policy.par
            model_initials.par
                enterprise_initials.par
                environment_initials.par
        endogenous.par
            modelparameters.par
                model_beginofperiod.par
                model_endofperiod.par
                modelparameters_in_period.par
            variables.par
            aggregationparameters.par
        equations.equ
        models.mod
        other.par
    import_data.inp
        loop_initialization.gms
        data_initialization.gms
        initialization_schemes.gms
    MW_Model.gms
        BasePeriodInitialization.gms
        transfer_from_stateblock_to_beginperiodmodel.gms
        periodmodel.gms
        transfer_from_endperiodmodel_to_stateblock.gms
        aggregation.gms
    gdxoutput.gms
```

Figure 7 The MicroWave GAMS Model Tree (as viewed with GTree)

In the tree, 5 main branches can be identified:

1. **Set_definition.gms**

   All sets and their subsets are defined in this file.

2. **Declarations.gms**

   In MicroWave, the model itself is separated from all declarations and input definitions. The difference between a declaration and a definition is that a declaration only notifies GAMS that a certain element exists, whereas a definition also associates a certain value to the element. All entries that are used in a model are declared in the section “declarations”. The advantage of this approach is that (1) it reduces the effort to find an entry definition and characteristics and (2) entries that are not used in the model can be identified and removed.

   The declarations file is further subdivided into the following files:

   a. Stateblockparameters.par
   b. Exogenous.par
   c. Endogenous.par
   d. Equations.equ
   e. Models.mod
   f. Other.par

Once this structure is defined, the initial input data are read into the Tree in the third branch:

3. **Import_data.inp**
    a. Loop_Initialization.gms
    b. Data_Initialization.gms
    c. Initialization_Schemes.gms

In the next branch, the relationship between the conceptual framework and the Model Tree becomes clear by replacing the term *data structure* with the term *state block* in the conceptual framework. Both terms refer to the same structure to keep all information that is collected about a certain *enterprise* (or its consisting parts) in a certain *period*. In this part the model is separated from the data.

4. **MW_Model.gms**
    The MW_Model.gms file is the core of the MicroWave model. The basic approach and underlying principle is: **'develop generic modules that can be used and reused independent of time or place they are called in the model, as long as they are provided with the right input'**.
    All modules are structured in these basic steps:
    a. BasePeriodInitialization.gms
    b. Transfer_from_stateblock_to_begin_periodmodel.gms
    c. Periodmodel.gms
    d. Transfer_from_end_periodmodel_to_stateblock.gms
    e. Aggregation.gms
    The first two files are used to read the original and associated data from the database and the time and space indications are stripped from the entry data.

A last branch is used to generate output that can be used in the GDXviewer that comes with GAMS or in the GSE-interface (see section 6).

5. **GDXoutput.gms**

These 5 branches make up the backbone, the stem of the MicroWave GAMS Model Tree, and should be used in all modules that are built using the MicroWave approach. Complementary to MetaWave, which supported the modeling process especially for knowledge registration, the MicroWave GAMS Model Tree aided by GTree software supports programming activities, gently forcing the programmer to modularity and generic pieces of code.

6 Model versions and scenarios using GSE

Once a model is developed as a computer program, you want to experiment with it by defining scenarios that use varying data input or settings of certain parameters. It is a good practice to keep your basic model version separated from your scenarios. Besides, you never build one version of your model and after a while you wonder which version you have used to solve scenarios for a certain project. Not only you but also your customers require reproducibility of your model and scenarios and hence good bookkeeping of them is necessary. Also storing, showing, and changing the input/output data and performing scenario comparisons are very important. Having a good user interface will help the model builder in developing his model (tracking errors) and will be of great help when a model is used for policy advise.

GSE is a software tool that is extremely useful for quickly getting a user interface for an existing GAMS model and keeping your scenarios separated from your base model. The great advantage of GSE is that a model builder can spend his time/effort on the quality of the model and scenarios using a user interface with full functionality for handling input and output data and version and scenario control. For every version and scenario you can add additional information to the GSE database like text, pictures, word documents etc., hence registering your knowledge about the current model version and scenario.

After registering your knowledge in MetaWave and building your program in the MicroWave GAMS Model Tree (partly generated from MetaWave) you only need to insert some comments/tag lines (also done by GTree and MetaWave) and use the Import2GSE wizard to get a fully functional GSE user interface that can run and compare versions/scenarios. After running a few scenarios you probably want to update your model. This is done by registering your new knowledge in MetaWave (and do a peer review), go to the MicroWave GAMS Model Tree and change the GAMS code and finally import the new version into GSE.

Like GTree, GSE was not developed exclusively for the MicroWave approach; it can also be used for other modeling activities in which GAMS is used for computer programming. However, GSE perfectly fits in the MicroWave philosophy of developing efficient, highly qualitative models where knowledge registration plays a key role. A more extensive description of GSE can be found elsewhere in this book.

7 The MicroWave Manual

So far, a conceptual framework and several supporting tools or predefined structures were presented that should overcome the problems that were mentioned in the introduction. Obviously, the MicroWave approach supposes a modeling process in which several persons are involved. Communication between persons by definition has a potential risk of loosing information and thus quality of the final result. To minimize this risk, we developed a protocol that was described by the Microwave Manual. In the following paragraphs, the most important parts of this manual are summarized.

Each simulation model in MicroWave consists of one or more modules. Core to such a model is a main module by which the simulation can be started. A module can be defined as a mathematical algorithm, consisting of either calculation rules or other modules. It is clear that the approach to such a model, or its individual consisting modules, depends on their purpose. Depending on whether you are a model user, model developer or model manager, you will be engaged in either of the following four main procedures (see also Figure 8).

1. **Model using -** using finished modules for scenario-analysis;
2. **Model building -** developing new modules;
3. **Model management and administration -** organization of the IT environment and of model management issues;
4. **Model revision -** changing or adapting modules.

The first two procedures are executed *ad hoc*: they are independent projects or project activities. The third procedure, management and administration of the simulation model, is not related to one activity in particular and therefore needs to be managed at institution level.

The first procedure, model using, includes an analysis of the *ad hoc* problem, an exploration of suitable modules and the data that is necessary in this context. In absence of an appropriate module, the procedure ‘Building the simulation module’ is started, involving a sequence of steps needed for building modules. All modules that are ready for (re)usage are managed in the procedure ‘Management and administration of the simulation models’. If a specific model needs to be revised or rebuilt, the procedure ‘revising a simulation module’ is called.

Each main procedure is split up into several subprocedures that are not further described in this paper. For each procedure, its objective, explanation, necessities and end products are described in detail. Another step that has to be taken, is assigning specific responsibilities to persons in order to meet the requirements for ISO 9000 certification.

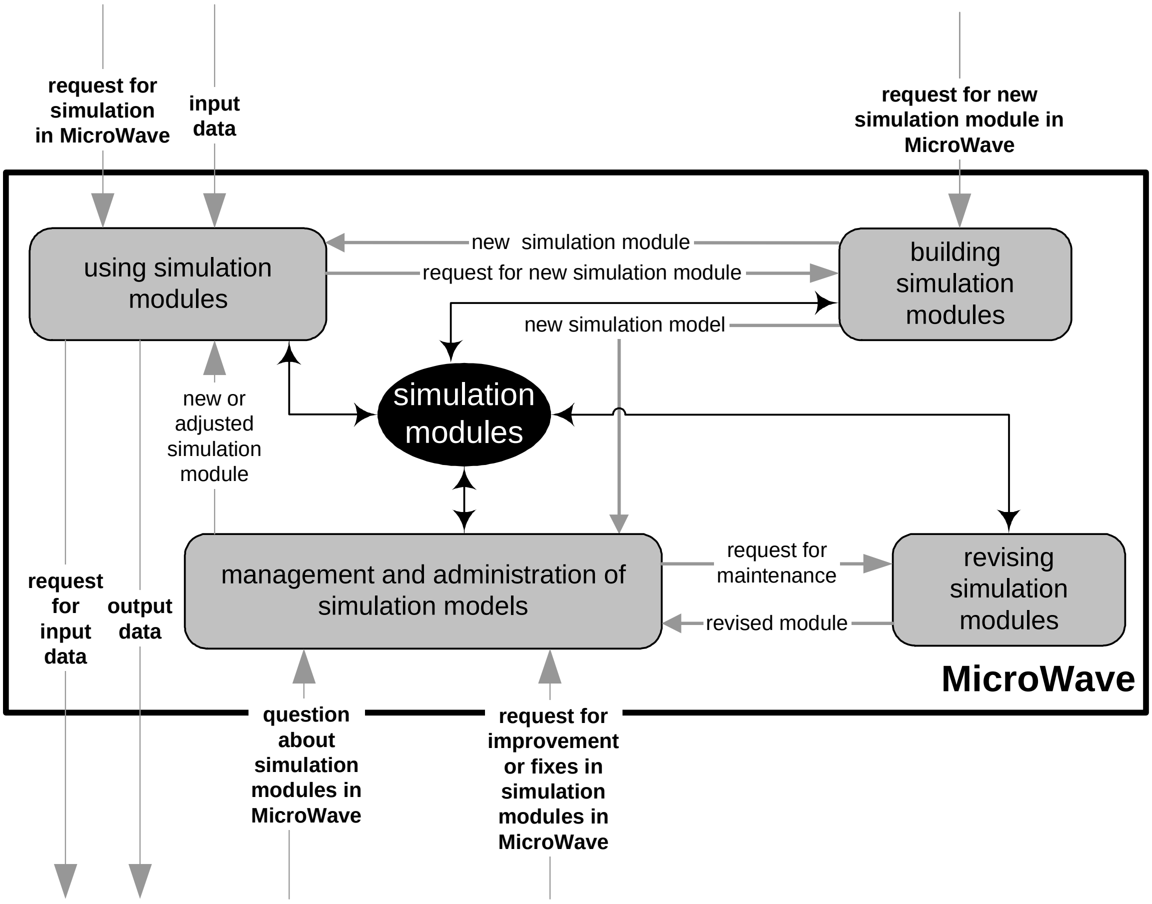

request for simulation in MicroWave
input data
request for new simulation module in MicroWave
using simulation modules
new simulation module
request for new simulation module
building simulation modules
new simulation model
simulation modules
new or adjusted simulation module
management and administration of simulation models
request for maintenance
revised module
revising simulation modules
request for input data
output data
MicroWave
question about simulation modules in MicroWave
request for improvement or fixes in simulation modules in MicroWave

Figure 8 – A workflow diagram for managing modules within MicroWave.

8 Illustrations of the MicroWave approach

The MicroWave approach has now been completely described by the previous sections. In this section we will provide some examples that illustrate how the approach has been used so far. Because we are working in the context of economic research, the first modules we developed deal with investment and financing behavior. Secondly, we used an existing research question on energy use in Dutch horticulture to test the complete MicroWave approach. Finally, an existing GAMS model on analyzing the economic potential of Flemish farms for conversion to organic agriculture was partly re-engineered by putting it into the MicroWave GAMS Model Tree.

8.1 Investment and financing modules

Simulation of the future financial-economic situation of a farm cannot be carried out without considering choices concerning investing in new assets, replacing old assets and other strategic measures such as changing the production plan. The objective of the 'investment and financing' component is to appraise different alternative measures and evaluate them against the background of the financial space available at the firm. The methodology should be general, so that it can be used for every (type of) farm. The output of this component is what choices are made and implemented and their financial implications.

The investment and financing component involves the following modules:

1. **Measure input** – calculating the effects (can be technical and/or financial) of each measure for the specific firm that is considered
2. **Package generator** – determination of all possible packages of measures, relevant to the firm's context.

3. **Annuities** – calculation of the annuity of each combination of measures grouped in a package.
4. **Financial space** – determination of the financial space, based on several financial indicators.
5. **Optimisation** – from all packages that can be financed, the entrepreneur chooses the package with the highest annuity. The implementation of that packages leads to changes in the value of assets, loans, liquidities, cash flow and - if relevant - technical indicators.

All modules are documented in MetaWave, reviewed by experts and implemented in GAMS in accordance with the conceptual framework and the MicroWave GAMS Model Tree. Thus, other model builders can easily judge and reuse the modules or they might want to upgrade a certain module or replace one or more modules by others. For example, in the currently implemented modules, the most optimal package is selected by using the 'Net Present Value method', but another optimization method can be applied by replacing this module. In this way, this set of modules can also be considered as a kind of research agenda and a structured way of knowledge development for this particular area of knowledge on investing and financing.

8.2 Energy model for horticulture

The general research question in this case was to analyze the impacts of energy market developments and changing climate policy on Dutch horticulture. The MicroWave approach was used to support this analysis. The objective of the energy model that was developed is to calculate cash flows of energy-investments and to calculate the penetration degree of energy assets such as energy screens, heat buffering, etc. in Dutch horticulture.

After analyzing the research question, MetaWave was used to search for useful existing knowledge or modules. In this case, the financial transition modules and the investment and financing modules have been reused.

The next step was to build a specific 'energy component', consisting of several modules with in-depth expert knowledge for this specific field of knowledge. At the moment, this knowledge is not yet completely documented in MetaWave and it is partly programmed in GAMS using Gtree, consistent with the MicroWave GAMS Model Tree. Although the energy model is still under construction and partly programmed, it is already possible to use GSE for viewing and discussing preliminary results.

This case illustrates how existing knowledge, i.e. financial-economic components, could be reused and how new components could be incorporated in accordance with the conceptual framework and the MicroWave GAMS Model Tree. To obtain the maximum gain in improvement of the modeling process, the MicroWave approach should be applied completely. However, this case shows that certain gains can already be obtained by applying parts of the approach. It is not necessarily a strict, obligatory procedure that must be followed from the beginning to the end.

8.3 Bio-conversion model

The bio-conversion model is a micro simulation model build and owned by the Flemish Centre of Agricultural Economics (CAE). This model analyses the economic potential of conversion to organic agriculture using LP-modeling. It is already programmed in GAMS.

We used this model as a test case to convert an existing model into the MicroWave framework. In this conversion, three steps can be identified:

1. analyze the existing model by reading the GAMS code, report and the model builders' notes;
2. cut the bio-conversion in different pieces and place those parts in the generic MicroWave GAMS Model Tree;
3. register the model knowledge in MetaWave and/or documents according to the template and adjust GAMS code so that it becomes more generic code.

At the moment, only the first two steps were carried out. This already resulted in a more structured model and then it appeared that quite some pieces of code were made redundant and could be left out. The readability of the model was also enlarged, which makes peer reviewing possible. This case also illustrates that the structure of the MicroWave GAMS Model Tree could be easily applied to an arbitrary chosen micro simulation model.

9 Discussion

This paper presented the MicroWave approach that is an infrastructure and associated supporting tools that guide modelers in their modeling process in order to meet our goals in attaining a higher efficiency in model development, a better quality of models and their output and an improvement in knowledge management. In this section, we will discuss how far we succeeded.

The examples of the energy model and bio-conversion model show that MicroWave leads to improvement that especially can be ascribed to the conceptual framework, implemented in the MicroWave GAMS Model Tree. The framework helps to define the model in a structured, uniform way, which enlarges the transparency and understandability of the model. It also makes it possible to concentrate on isolated pieces of the model without being bothered by the whole. By using a standard way of working, researchers that wish to understand the working of a model, or contribute to its extension, can rely on their previous experiences with the MicroWave structure to quickly understand and interpret an existing model. This will increase the efficiency of model building greatly.

Moreover, the fact that two examples from different fields of research and application fit in the same framework, are evidence of the generic properties. The energy model reused several existing modules. It can be expected that this type of efficiency gain can increase when more modules are being developed in MicroWave.

Because of the modular approach and strong focus on generic pieces of code that are made independent from specific datasets or research themes, coupling of models becomes easier and will lead to results of a higher quality. In fact, within the MicroWave approach, the focus is not on coupling of two separated models, but on combining the right modules to answers a specific research question. However, we admit that this aspect still has to be tested by developing more models/modules in MicroWave.

It was explained how documenting in MetaWave highly contributes to a better management of knowledge. This also contributes to efficiency because by emphasizing peer reviewing and aiming at a central documentation point, there is less chance that the wheel is re-invented at different places. Peer reviewing also improves the quality of the model in general. The whole MicroWave approach, but MetaWave in particular, opens possibilities for efficient co-operation in multi- or interdisciplinary research leading to more effective results.

Obviously, MicroWave stands for a group model building approach in which each person plays a different role. This requires that knowledge is made explicit, which implies quality improvement. However, this requires coordination and a certain discipline. The protocol in the MicroWave Manual is meant to facilitate this process. When applying this protocol, quality certification like ISO 9000 is within arm’s reach, but therefore it should be implemented officially at institutional level. It could be argued that these kinds of protocols hamper creativity of modelers. However, it was illustrated that the MicroWave approach doesn’t necessarily have to be applied as a whole. Depending on aspects like time, budget or other priorities it can be partly applied, still leading to improvement. One can also think of defining different official statuses depending on the degree the MicroWave approach was applied.

10 Conclusions and future plans

In this paper MicroWave is presented as an approach that can greatly improve the modeling process. It is developed for the context of micro simulations, but several concepts and tools are also applicable for a wider scope. The MicroWave infrastructure and supporting tools are not a formula that automatically leads to model improvement. This still depends very much on people that use it and work together in model building teams. It was also made clear that the success of MicroWave still has to be proven by implementing more models/modules in the framework to obtain a critical mass of knowledge that can be reused to increase efficiency. For that reason LEI and Stedula have decided to continue their cooperation in a consortium and invite others to join.

11 Acknowledgements

The authors would like to thank all persons from the Flemish Policy Research Centre for Sustainable Agriculture (Stedula), LEI and Wageningen University that contributed to the development of the MicroWave approach.

12 References

Hennen, W. H. G. J. (1995). Detector : knowledge-based systems for dairy farm management support and policy analysis : methods and applications. *In* "Onderzoeksverslag / Landbouw-Economisch Instituut", pp. 205. LEI, The Hague.

Mulder, M. (1991). Financiële analyse en continuïteitsvoorspelling: een rekenmodel op basis van de LEI-boekhouding *(Financial analysis and continuation prediction: a calculation model based on the LEI farm accountancy data network)*. Rep. No. 4.127. LEI, The Hague.

Mulder, M. (1994). Bedrijfstakverkenning en financiële analyse: een simulatiemodel voor de glastuinbouw *(Sector exploration and financial analysis : a simulation model for horticulture under glass)*. Doctoral Thesis, Erasmus University, Rotterdam.

Scholten, H., Van Waveren, R. H., Groot, S., Van Geer, F. C., Wösten, J. H. M., Koeze, R. D., and Noort, J. J. (2000). Good Modelling Practice in water management. *In* "Proceedings HydroInformatics2000. International Association for Hydraulic Research", Cedar Rapids, Iowa, USA.