SPICOSA

E-NEWS

2009 - SPECIAL ISSUE

EXTENDSIM 5

TABLE OF CONTENTS

TOWARDS A GENERIC MODEL LIBRARY

Purpose of a generic model library

The 18 study sites in the SPICOSA program are now well underway with the design of their ExtendSIM® models to describe a variety of coastal processes ranging from eutrophication, fisheries, and beach attractiveness to clam culture. A generic model library is a complete set of reusable model components which can be used to (re)build coastal models for new study areas, not much unlike the well-known Lego bricks.

The population of such a model library is one of the key challenges of the project and serves several purposes:

- existing models are easier to expand or maintain
- new models are easier to build
- quick exchange of models within the scientific community
- avoidance of unnecessary modelling (not “reinventing the wheel”)

The principles of generic modelling have been elaborated during the SPICOSA Cluster Workshops that were held from February 17-20 in Amsterdam and Copenhague. For more information you can contact the WP8 leader, Jean-Luc de Kok (jeanluc.dekok@vito.be). See also project deliverable D8.8 for a more detailed discussion of this topic (internal document for project partners).

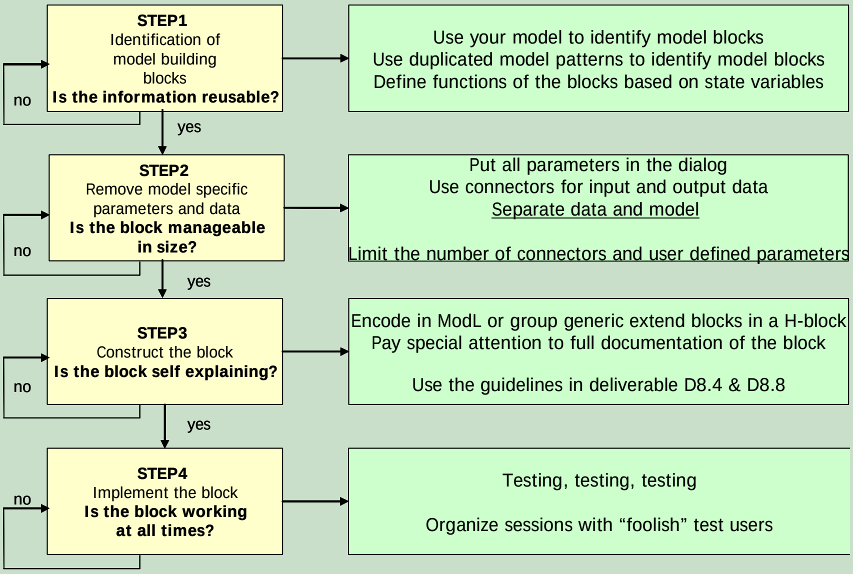
What makes a model building block reusable ?

Reusability is a criterion which cannot be quantified precisely, but it is clear that the degree of case specificness and level of detail are important aspects. The more specific a component is the more reusable it is. For example, a mathematical operator is completely reusable, but a coastal model based only on such “low-level” components will quickly become complex and difficult to understand, and it is much better to build in different levels of hierarchy in the model. Extend offers the option to restructure models in so-called hierarchical or H blocks. On the other hand, very large components capturing, for example, the complete coastal economy with all it details will become very case dependent thereby limiting their usefulness for application to other study areas or problems. So an appropriate, intermediate level of hierarchy needs to be identified for the model components. The examples discussed later will demonstrate this. In addition to the level of detail there are other important aspects:

- the function and use of the model component should be well documented
- the number of in- and outgoing connectors should be limited
- the implementation is hidden (encapsulation) but documented
- data should separated from the model
- the block has been tested and is robust

How to design a reusable model component ?

A general procedure for designing the block is outlined below:



STEP1
Identification of model building blocks
Is the information reusable?
no
yes

Use your model to identify model blocks
Use duplicated model patterns to identify model blocks
Define functions of the blocks based on state variables

STEP2
Remove model specific parameters and data
Is the block manageable in size?
no
yes

Put all parameters in the dialog
Use connectors for input and output data
Separate data and model

Limit the number of connectors and user defined parameters

STEP3
Construct the block
Is the block self explaining?
no
yes

Encode in ModL or group generic extend blocks in a H-block
Pay special attention to full documentation of the block

Use the guidelines in deliverable D8.4 & D8.8

STEP4
Implement the block
Is the block working at all times?
no

Testing, testing, testing

Organize sessions with “foolish” test users

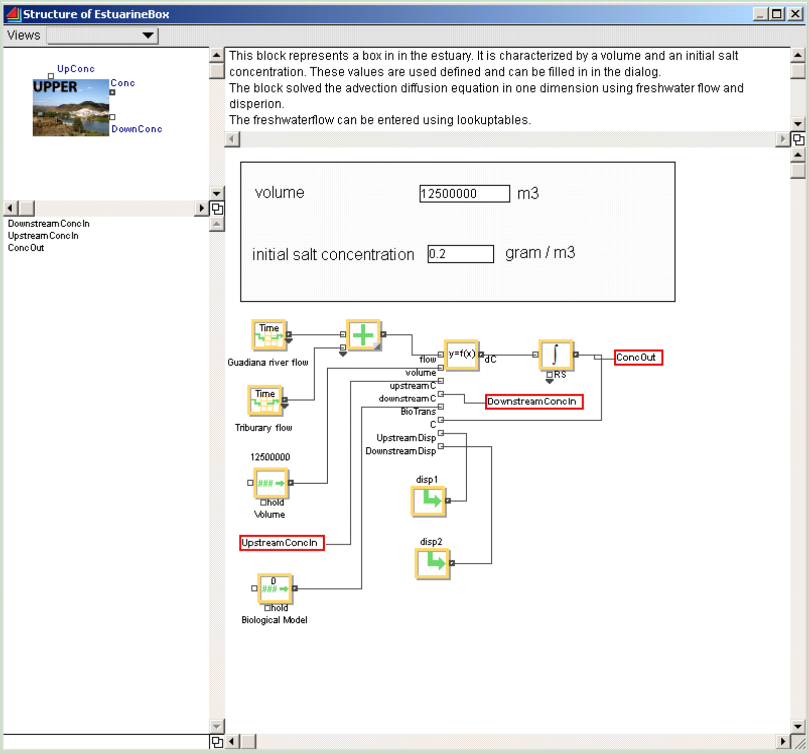
In principle Extend offers two ways to design the reusable model components: bottom-up by starting from scratch with the design of the block, or top-down by restructuring an existing model in hierarchical blocks. The design of a new block can be carried out with the structure window for designing H blocks, or in the C-like ModL programming language, which is more technically demanding but also offers more flexibility to add additional information for the users such as dialog boxes. A good compromise, used by many study sites, is the Equation block from the Extend Value library: it comes with a useful comment tab to add documentation.

An important topic discussed during the workshop was data management. Although no model component can function without data it is not recommended to place model specific data in the block itself. This makes the block less comprehensible, and will cause problems when people try to reuse the block and are not aware of the parameters included. Instead, it is much better to let the model component read its data from a database, which can be external (e.g. and Excel file) or an internal Extend database. The general rule of thumb is that (model-specific) parameters should be read from a database, and that time-dependent state variables should be passed via the connectors.

A detailed explanation on how to set up, manage and use an Extend database has been presented during the Cluster Workshop and can be found on the SPICOSA server under the WP8 directory (internal document for project partners).

Examples of candidate blocks for the model library

Over twenty useful potential candidate blocks for the model library were identified during the Cluster Workshops. Here we only present some examples. Note that these blocks still require some restructuring to meet the criteria mentioned above.



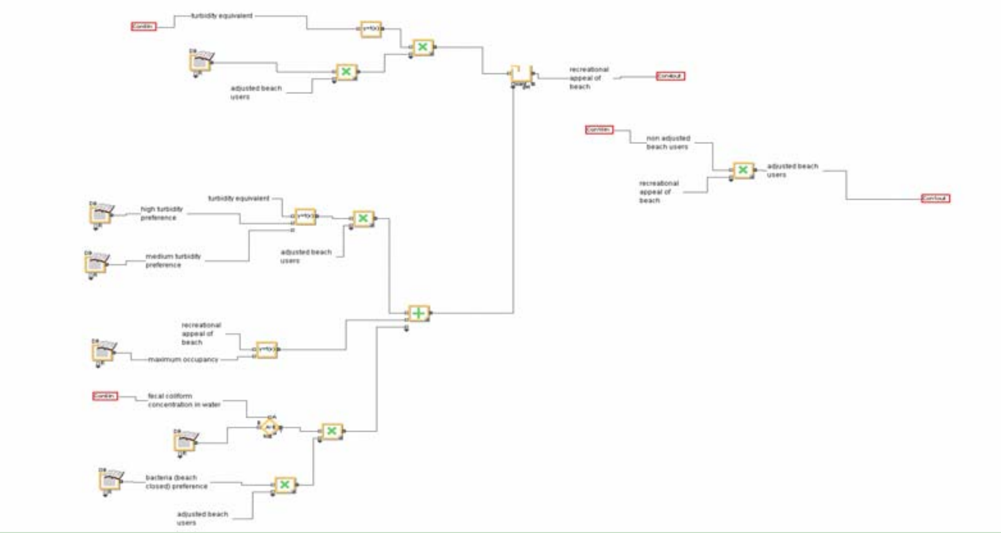
Structure of EstuarineBox

This block represents a box in in the estuary. It is characterized by a volume and an initial salt concentration. These values are used defined and can be filled in in the dialog.
The block solved the advection diffusion equation in one dimension using freshwater flow and disperion.
The freshwaterflow can be entered using lookuptables.

volume 12500000 m3

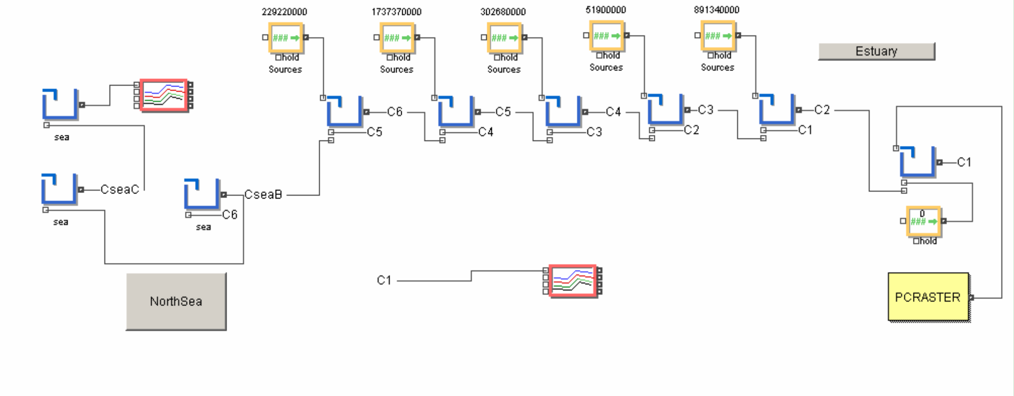
initial salt concentration 0.2 gram / m3

1. View on an open H-block structure window for a model component describing 1D advection-diffusion.

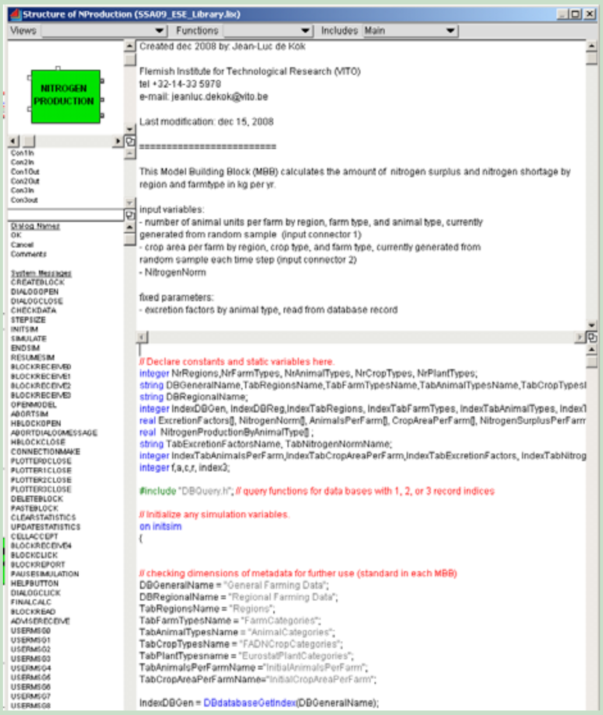


turbidity equivalent
adjusted beach users
recreational appeal of beach
non adjusted beach users
adjusted beach users
recreational appeal of beach
turbidity equivalent
high turbidity preference
medium turbidity preference
adjusted beach users
recreational appeal of beach
maximum occupancy
fecal coliform concentration in water
bacteria (beach closed) preference
adjusted beach users

2. Candidate model block for the attractiveness of a beach for tourism related to the water transparency.



3. Example of a candidate block with a box model to describe the estuarine dispersal of dissolved organic nitrogen in the Scheldt estuary.



```
Created dec 2008 by: Jean-Luc de Kok

Flemish Institute for Technological Research (VITO)
tel +32-14-33 5978
e-mail: jeanluc.dekok@vito.be

Last modification: dec 15, 2008

=========================

This Model Building Block (MBB) calculates the amount of  nitrogen surplus and nitrogen shortage by
region and farmtype in kg per yr.

input variables:
- number of animal units per farm by region, farm type, and animal type, currently
generated from random sample  (input connector 1)
- crop area per farm by region, crop type, and farm type, currently generated from
random sample each time step (input connector 2)
- NitrogenNorm

fixed parameters:
- excretion factors by animal type, read from database record
```

```
// Declare constants and static variables here.
integer NrRegions,NrFarmTypes, NrAnimalTypes, NrCropTypes, NrPlantTypes;
string DBGeneralName,TabRegionsName,TabFarmTypesName,TabAnimalTypesName,TabCropTypesI
string DBRegionalName;
integer IndexDBGen, IndexDBReg,IndexTabRegions, IndexTabFarmTypes, IndexTabAnimalTypes, IndexT
real ExcretionFactors[], NitrogenNorm[], AnimalsPerFarm[], CropAreaPerFarm[], NitrogenSurplusPerFarm
real  NitrogenProductionByAnimalType[] ;
string TabExcretionFactorsName, TabNitrogenNormName;
integer IndexTabAnimalsPerFarm,IndexTabCropAreaPerFarm,IndexTabExcretionFactors, IndexTabNitrog
integer f,a,c,r, index3;

#include "DBQuery.h"; // query functions for data bases with 1, 2, or 3 record indices

// Initialize any simulation variables.
on initsim
{


// checking dimensions of metadata for further use (standard in each MBB)
DBGeneralName = "General Farming Data";
DBRegionalName = "Regional Farming Data";
TabRegionsName = "Regions";
TabFarmTypesName = "FarmCategories";
TabAnimalTypesName = "AnimalCategories";
TabCropTypesName = "FADNCropCategories";
TabPlantTypesname = "EurostatPlantCategories";
TabAnimalsPerFarmName ="InitialAnimalsPerFarm";
TabCropAreaPerFarmName="InitialCropAreaPerFarm";

IndexDBGen = DBdatabaseGetIndex(DBGeneralName);
```

4. Example of a candidate block programmed in ModL language to describe the production of organic nitrogen by cattle farming in the Scheldt catchment. The farm types and animal categories are read from a database and can be changed without compromising the functionality of the block.

SPICOSA

COLOPHON

This EXTEND Special NEWS ISSUE has been prepared by Jean-Luc de Kok, WP8 (Model Support) SPICOSA Partner 6 (VITO) jeanluc.dekok@vito.be

An integrated project under the EU´s 6th Framework Programme for Research (FP6) of the European Commission

SIXTH FRAMEWORK PROGRAMME