CONFIGURATION OF SIMULATION ALGORITHMS in ADOxx

Simulation Functionality in ADOxx

When working with graph-type systems, simulation functionality is needed for calculating and evaluating the behaviour of the systems in different situations.

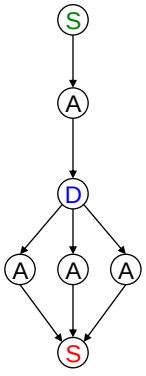
A typical graph system consists of:

-One start object

-Several „active“ objects

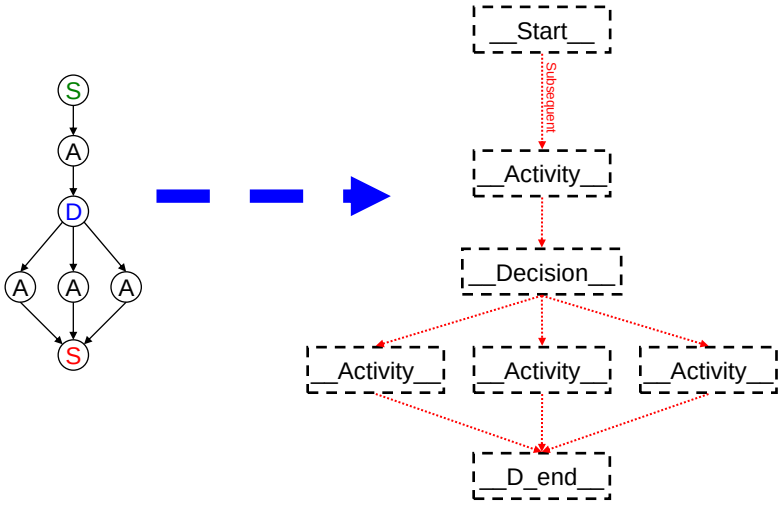
-One or more „decision“ objects

-One final object



Simulation Functionality in ADOxx

ADOxx has predefined abstract classes and relation classes and four predefined algorithms for enabling model simulation.



Simulation Functionality in ADOxx

ADOxx Simulation algorithms

1. Path analysis – simulates a dynamic model alone
2. Capacity Analysis – simulates a dynamic model AND the corresponding static model(s)
3. Workload Analysis – simulates a dynamic model AND the corresponding static model(s)

Path analysis in ADOxx

- "Path analysis" is a predefined simulation algorithm provided by ADOxx 1.3 UL1
  - It enables the user to evaluate your models containing „active" classes without regard to other models
  - It returns path-related results for a specific model
  - The user can access all possible paths depending on their frequency of occurence or any other criteria defined by the method developer
  - The Modelling Toolkit can display the paths both graphically and in text format
  - When executing the path analysis, the paths of the selected model are run through and the sum of all defined parameters are calculated
- In order to make simulation functionality available, the method developer must:
  - Define a model type that can be simulated
  - Configure the path analysis algorithm (mappings)

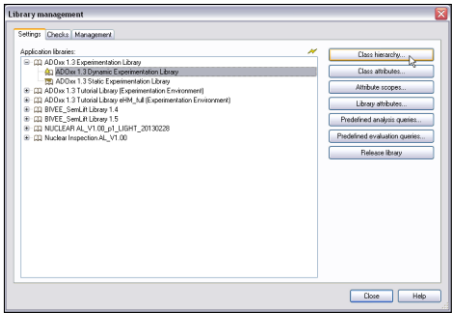
Path analysis in ADOxx

- In order to perform the path analysis, a model type needs:
  - A class that will represent the „initial point" of the model, derived from the abstract class „_Start_"
  - A class that will represent the „final point" of the model, derived from „__D_end__"
  - One or more classes that will represend the „active" objects of the model, derived from „__Activity__"
  - A class that will represent the „points of decision" in the model, derived from „__Decision__"
  - The „Subsequent" relation, already defined in the ADOxx metamodel, for connecting the objects
  - To be simulateable; the keyword „not-simulateable" must not be used in the definition of the model type

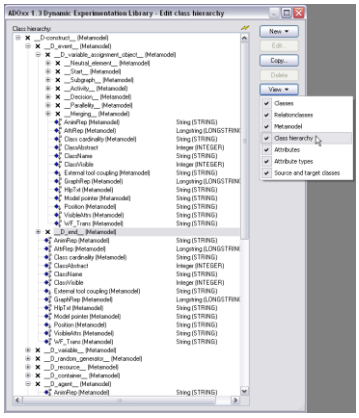
Path analysis in ADOxx

- The standard parameters for the path analysis algorithm need to be configured according to the classes defined for the model type to be simulated, by changing the following library attributes (available in the „Simulation“ tab):
  - "Simtext" contains some user-specific expressions used by ADOxx to label simulation results
  - "Simmapping" contains the definition of the input sets for the Simulation and a group of classes which are then used in simulation-related Actions.
  - "Sim result mapping" defines which simulation results are written back into which attributes of a model when you click on the "Evaluation" button.

SIMULATION CONFIGURATION HANDS-ON



1. Open the „Class hierarchy“ for the Dynamic library.



2. Activate de „Metamodel“ view with „Class hierarchy“

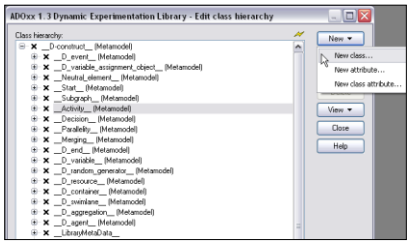
3. Add a class derived from the abstract class „__Activity__“.
This will be the main class used for simulation.

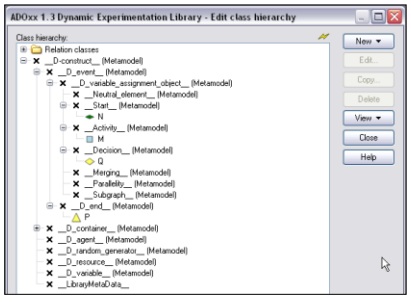
4. Add a class derived from the abstract class „__Start__“.
This will be the class where the simulation starts.

5. Add a class derived from the abstract class „__D_end__“.
This will be the class where the simulation ends.

6. Add a class derived from the abstract class „__Decision__“.
This will be the class used for making path decisions in the simulation.

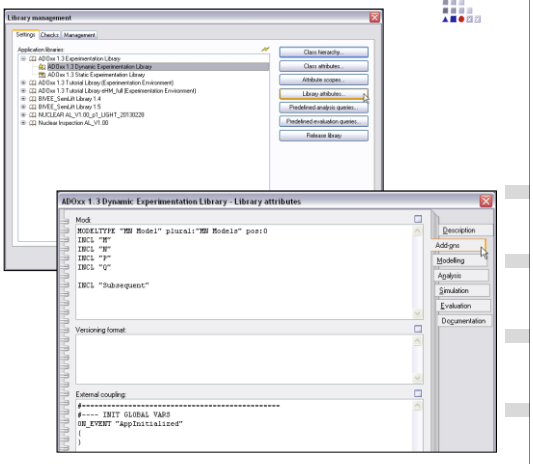
7. For the classes you defined, add the attributes that you want to use in your simulation. These attributes will have to be mapped in the „Simmapping“ library attribute.





8. Create a new model type that contains these classes.

9. Add the relationclass „Subsequent“ to the model type created at step 7.

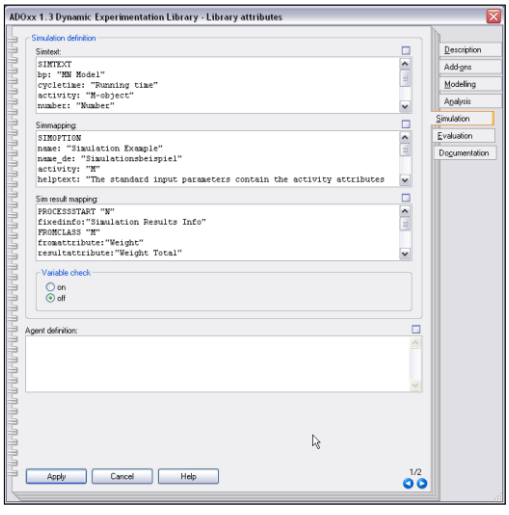


Sample definition of model type

```
MODELTYPE "MN Model" plural:"MN Models" pos:0
INCL "M"
INCL "N"
INCL "P"
INCL "Q"

INCL "Subsequent"
```

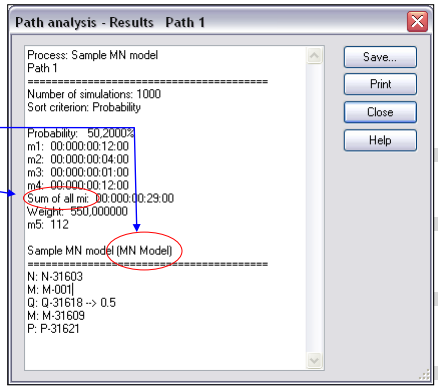

10. In the „Simulation“ tab of the Library attributes, add the simulation texts (Simtext), the simulation attributes mapping (Simmapping) and the rules for mapping the results of the simulation (Sim result mapping) for your specific classes and model type.



The attribute „Simtext“ contains some user-specific expressions. These are used by ADOxx to label simulation results . The Simtext is used in the Path Analysis in the result text.

Sample „Simtext“ attribute:

```
SIMTEXT
bp: "MN Model"
cycletime: "Sum of all mi"
activity: "M-object"
number: "Number"
actor: "Object X"
rescost: "Cost 1"
resource: "Cost 2"
perscost: "Cost 3"
```

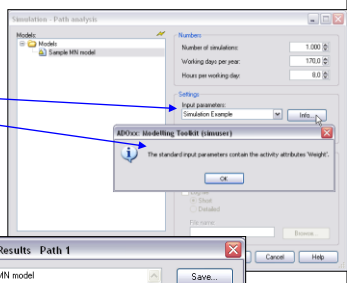


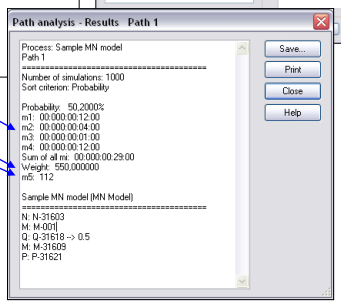
The parameters „activity“, „number“, „actor“, „rescost“ „resource“ and „perscost“ are not used in the path-analysis simulation, but they have to be defined in the „Simtext“ library attribute.

"Simmapping" is an attribute of the Dynamic library. In this attribute, input sets for the Simulation and the Analytical Evaluation are defined. Furthermore, a group of classes can be defined which are then used
in simulation-related Actions.

Sample „Simmapping“ attribute:

```
SIMOPTION
name: "Simulation Example"
activity: "M"
helptext: "The standard input parameters contain the activity attributes 'Weight'."
executiontime: "m1"
waitingtime: "m2"
restingtime: "m3"
transporttime: "m4"
userattribute-1: "Weight"
userattribute-2: "m5"
SIMCLASSES
bp-all
we-none
```





The attribute "Sim result mapping" defines which simulation results are written back into which attributes of a model when you click on the "Evaluation" button.

Sample „Sim result mapping“ attribute:

```
PROCESSSTART "N"
fixedinfo:"Simulation Results Info"
FROMCLASS "M"
fromattribute:"Weight"
resultattribute:"Weight Total"
ACTIVITY "M"
fixedinfo:„Description"
fixednumber:"Number"
FROMCLASS "M"
fromattribute:"Weight"
resultattribute:"Weight Total"
FROMCLASS "M"
fromattribute:"m5"
resultattribute:"m5_total"
```

PROCESSSTART is a keyword used for assigning the name of the class that represents the starting point of the model that you want to simulate; this class must be derived from the class „__Start__“ (see slide 2).

ACTIVITY is a keyword used for assigning the name of the main class used in the model; this class must be derived from the class „__Activity__“ (see slide 2).

FROMCLASS is a keyword used for selecting additional classes (*FromClassname*) and specify values from the fromattribute attribute values (*FromAttributename*) specified. The selected attributes of this class can be transferred back through toattribute into the respective attribute (*ToAttributename*).

11. Open the ADOxx 1.3 UL1 Modelling Toolkit

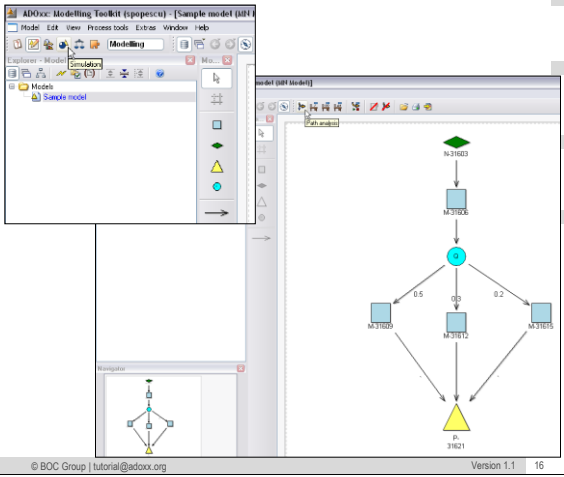
12. Create a new model of the model type you want to simulate

-The model must contain exactly one object of the class you defined as PROCESSSTART
-The model must contain exactly one instance of class you derived from the abstract class __D_end__
-For each instance of the class derived from „__Decision__“, the sum of the transition conditions of all connectors of type „Subsequent“ starting from it has to be 1.
-From each class derived from „“__Activity__“ may start exactly one connector of class „Subsequent“.

13. Save your model

14. Switch to the „Simulation“ view

15. Click the „Path Analysis“ button to start the path anaysis simulation.



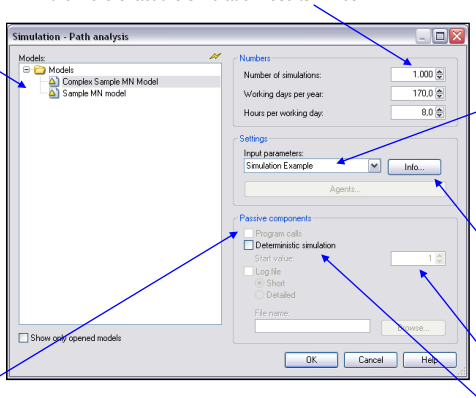
SIMULATION CONFIGURATION DEBUG VIEW

16. Select/enter the values you need for the path analysis simulation.

Select the model you want to simulate

Indicate how many processes are to be "run through". The number of simulations selected affects the accuracy - the higher the number, the more exact the simulation results will be.

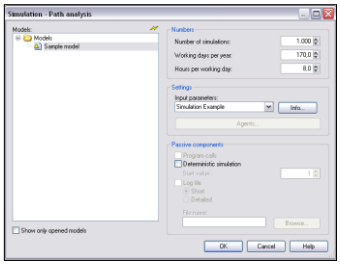
Select the input parameter combination you want to work with. The input parameters are defined in the „Simmapping“ library attribute of the dynamic library in your metamodel, using the „SIMOPTION“ keyword. (see slide 7)



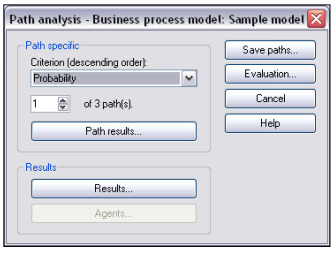
Display information about the selected input parameter combination (SIMOPTION).

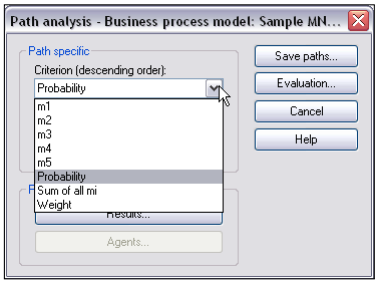
"Program calls" (default setting: deactivated)
If this option is activated, program calls specific to the activity will be carried out during the simulation of each activity. The selected input parameter combination will determine which program calls will be concerned by this .

"Deterministic simulation„ (default setting: disabled)
When enabling this option the simulation is initialised with the same start value. This ensures that with the same start value independent simulation runs will determine the same simulation results.



17. Select the number of simulations and click „OK“

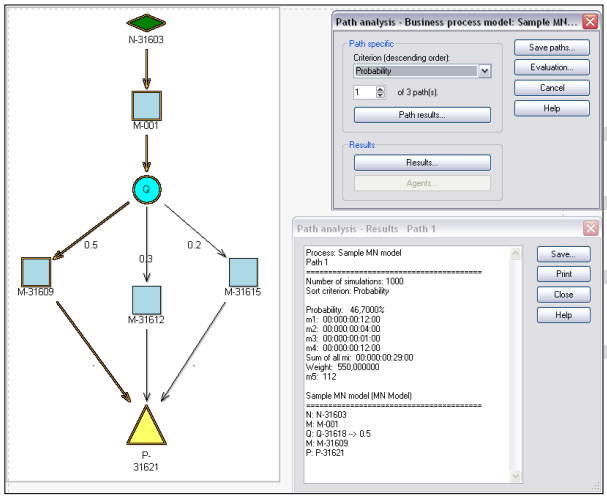




18. You may sort the resulted paths by Probability, or another criteria based on the attributes you defined.

19. Select any path you want and click „OK“ to display information it. The selected path will be marked on your model.

20. Save the data corresponding to the selected pah or all paths for further use.



Further Questions?

tutorial@adoxx.org