SysML Models Animation (using Rhapsody tool)

Jean-Michel Bruel <bruel@iut-blagnac.fr>

Revision History

Revision 1.0	February 2012	JB

Table of Contents

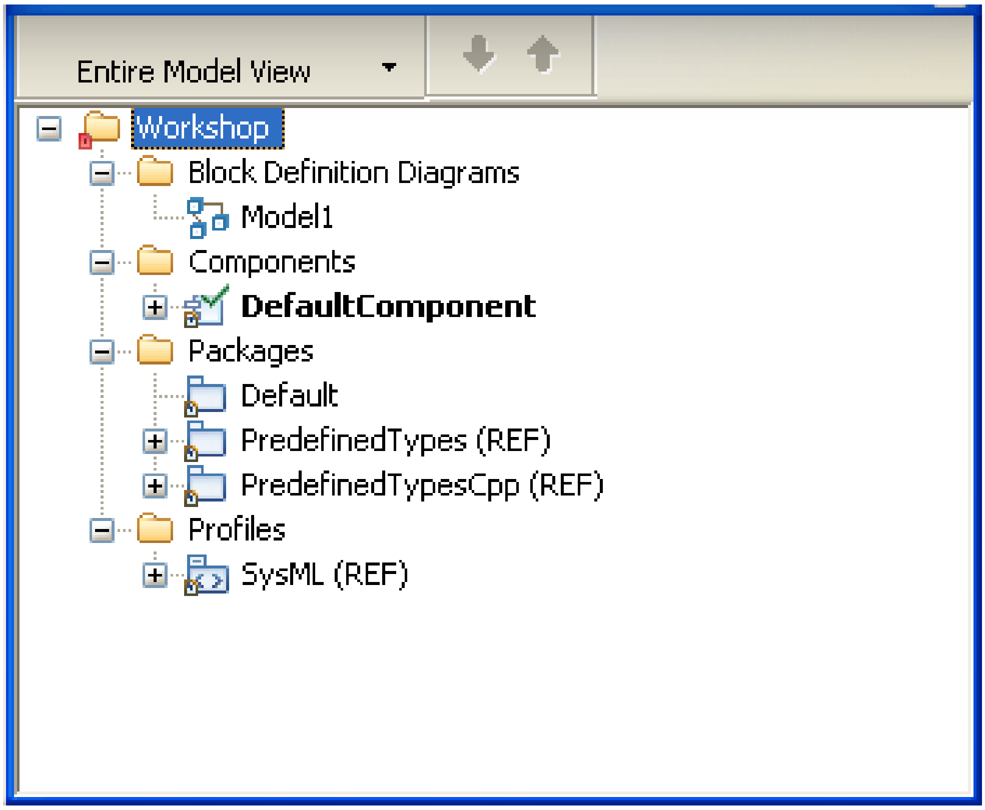
In this practical work, we are going to experiment the Rhapsody [http://www-01.ibm.com/software/rational/products/rhapsody/developer/] tool, by implementing a simple example of a Stack data structure. This tutorial was originally presented at "Universidad Autonoma de Guadalajara" (UAG [http://www.uag.mx/]) for the undergraduated students in engineering.

1. Tools

Verify that you have the corresponding tool on your machine: Rhapsody [http://www-01.ibm.com/software/rational/products/rhapsody/developer/] 7.6

1.1. Let’s start

- Open a new project (select the SysML kind)
- Add some organisation (packages) if needed (right-click#add#)



- An initial (Block Definition) Diagram is automatically generated and ready for you to start describing your system. If it is another diagram, add a new Block Definition Diagram by right-clicking in your project in the "Model view" (left part) and Add New → Diagrams → Block Definition Diagram.

- Click on the images of this tutorial to enlarge them

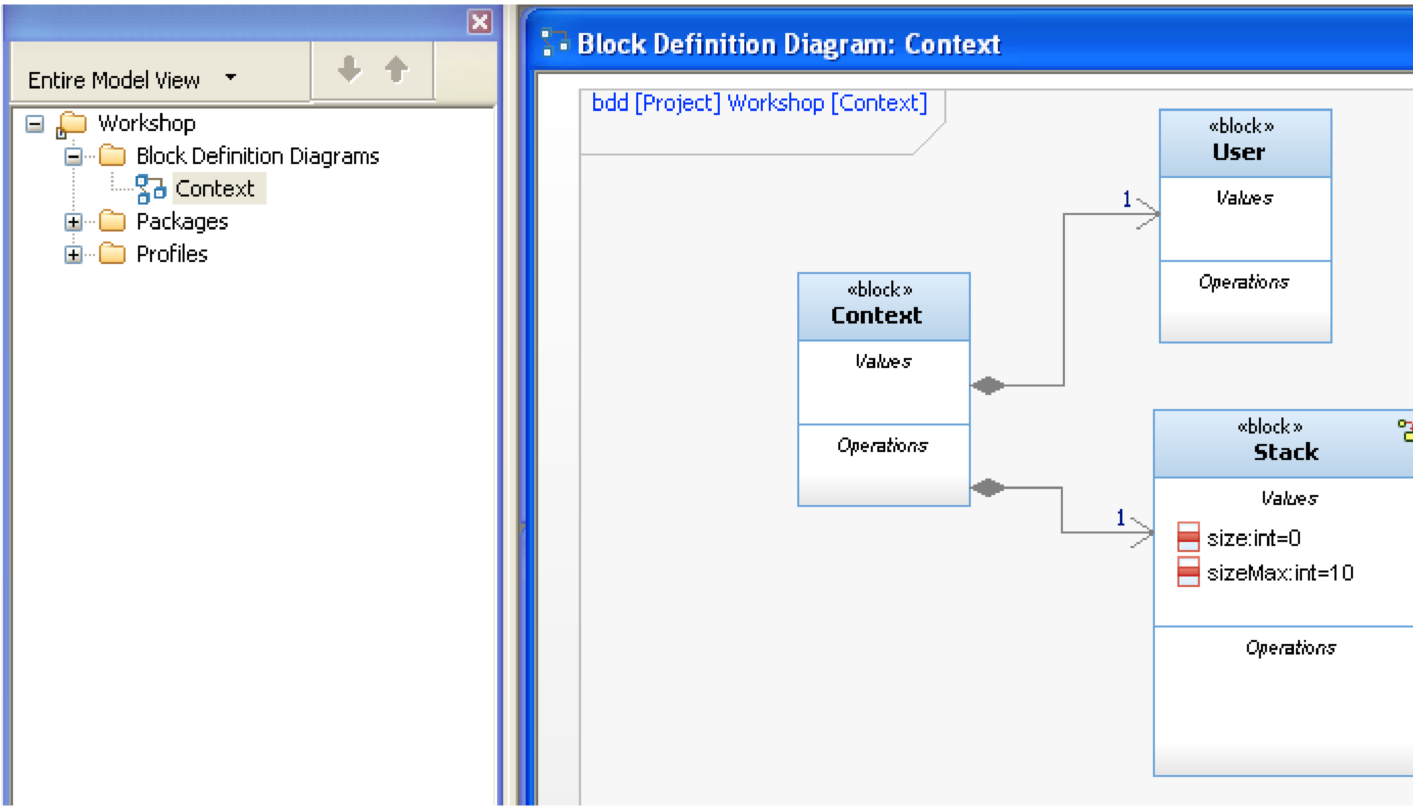
1.2. Block Definition Diagram

Start with a description of the system in its environment (most of the time called "context"):

- name your bdd into "Context"

- using the panel on the right, select a block and place one in the diagram

- notice the presence of the block you just defined in the model elements (Default Package)



- add the details of the stack (size, sizeMax)
- try different methods:
  - right-click on your block in the diagram and "Add New" → "Attribute"
  - right-click on yout block in the model view and do the same. What do you notice?
  - play with the display property of your block in the diagram to show the last attribute.

> There is two different "delete" in modeling tools: one delete from the diagram, the other from the model (all its occurences). Pay attention when deleting things. And as always save regularly. Tolls crash (especially on Windows ;-)

1.3. State Machine Diagram

We are now going to define the behavior of our Stack through a state machine.

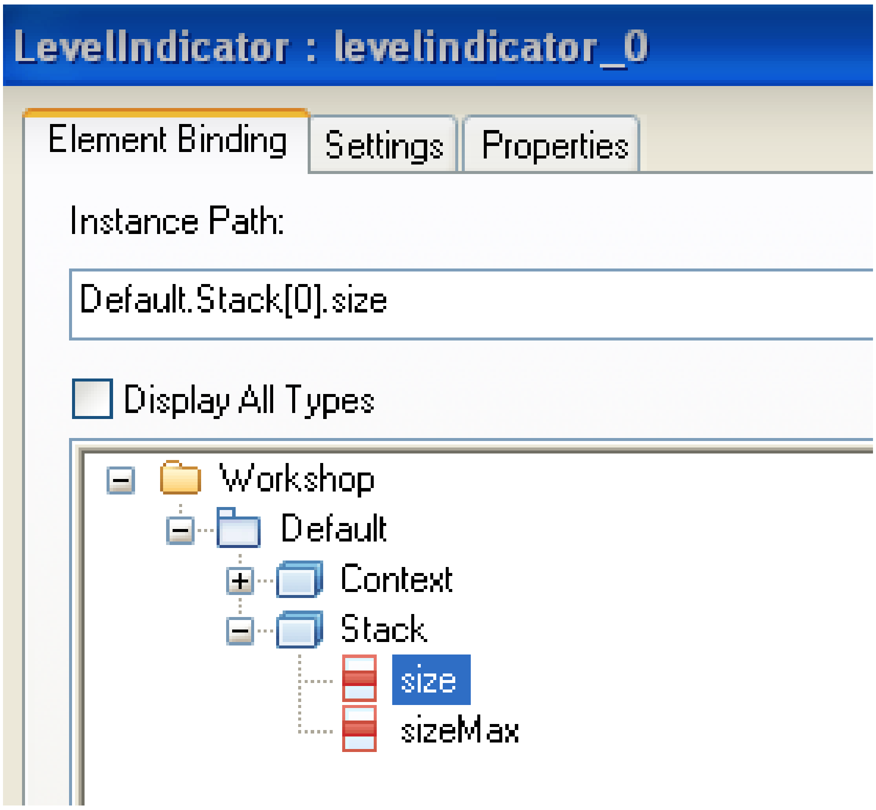
- Right-click on the block and add an new statechart
- Add 3 states (Empty, Full, Regular), as well as the entry (pseudo-)state
- Add transitions between them (according to push/pull events and according to the value of size and of sizeMax). If you really can’t find the state machine, here’s an answer [file:images/Rhapsody/under-stm.png].

In order to animate the model (see below), we need to provide precise default values and results of the events in the state machine (size++ when a push occurs for example). Make sure you provide those information.

1.4. Preparing the animation of your model

We are provide the model elements used to show the results of the animation of the state machine.

- Create a Panel Diagram (Right-click on your Project and Add#Diagrams#PanelDiagram).
- Find the corresponding diagram in the viewer and open it
- Using the panel on the right, place one "Level Indicator" and 3 "leds"
- Bind them to the corrsponding models elements:
  - bind the level indicator to the stack’s size
  - define the max value and the number of division in the "settings" folder
  - do the same for the three leds (for the 3 states, play with colors)

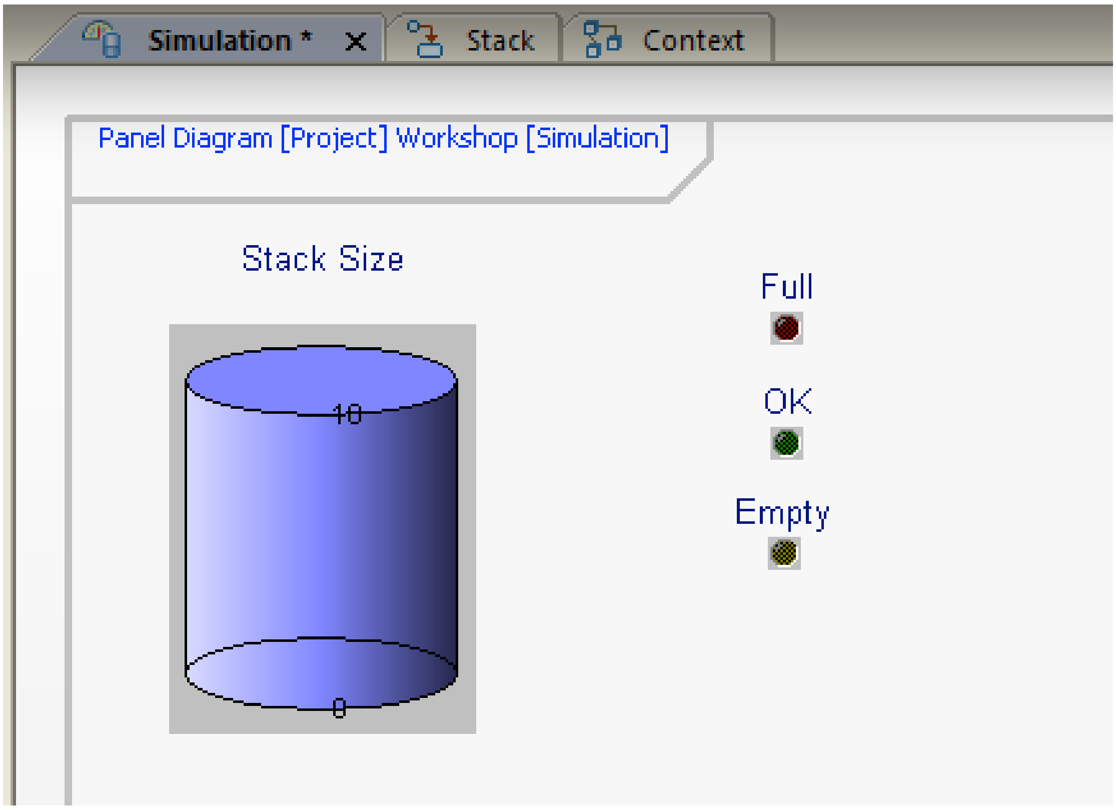


LevelIndicator : levelindicator_0

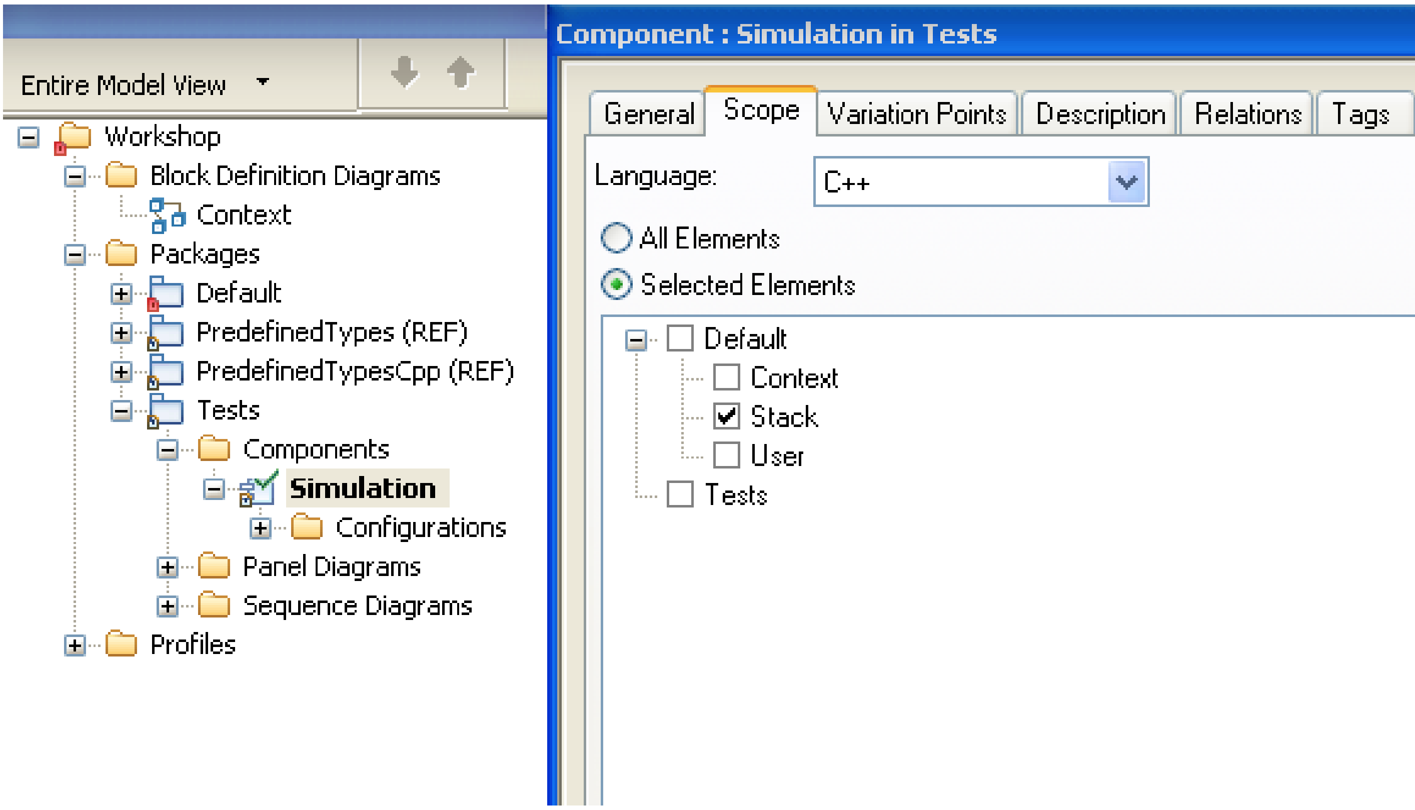
Instance Path:

Default.Stack[0].size

□ Display All Types



- Create a new Package "Test", and add a new component (Add New → UML → Component)
- Setup the C++ language and choose your Stack as the "scope" of the component.
- Setup the configuration of the component by choosing the Stack as initialized instance.



Entire Model View

- Workshop
  - Block Definition Diagrams
    - Context
  - Packages
    - Default
    - PredefinedTypes (REF)
    - PredefinedTypesCpp (REF)
    - Tests
      - Components
        - **Simulation**
          - Configurations
      - Panel Diagrams
      - Sequence Diagrams
  - Profiles

Component : Simulation in Tests

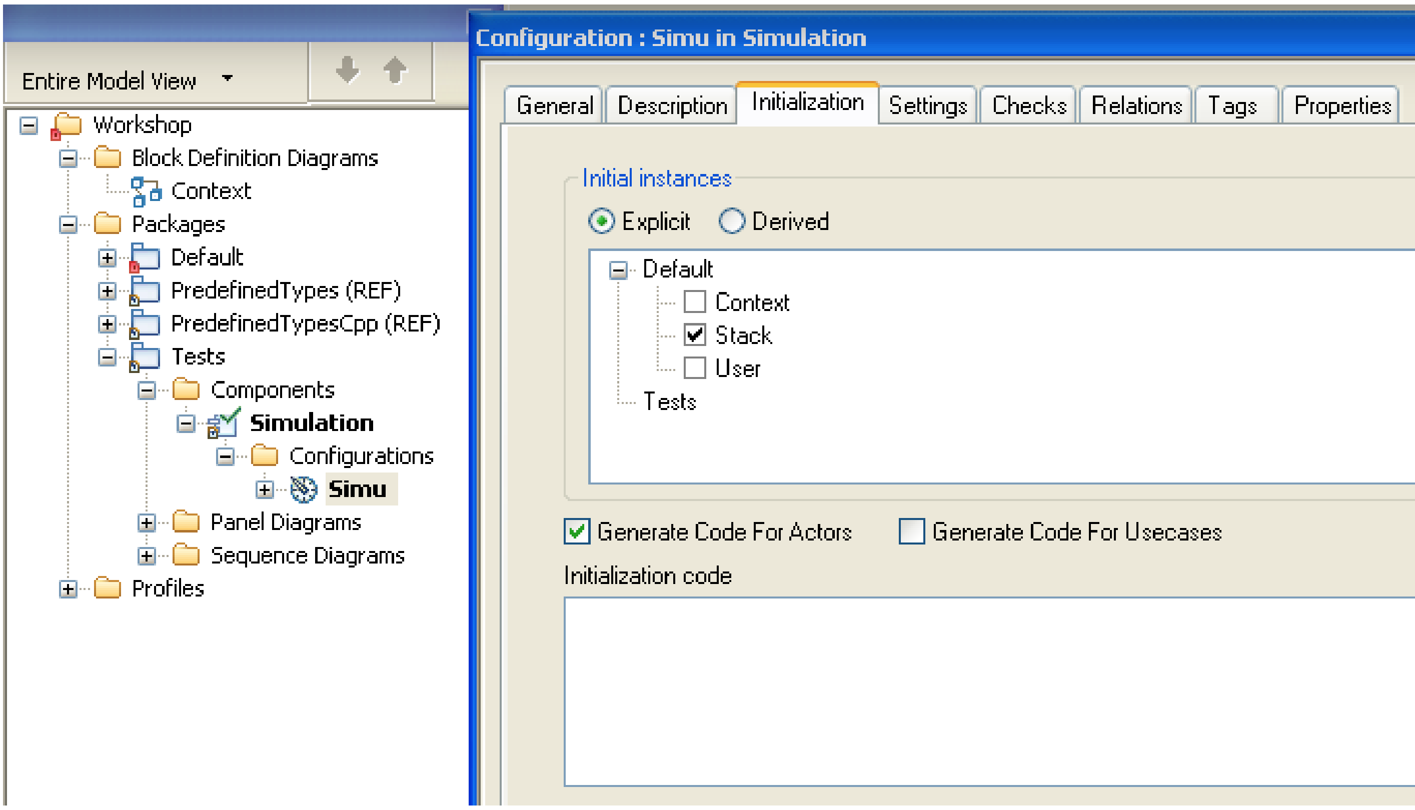
General | Scope | Variation Points | Description | Relations | Tags

Language: C++

○ All Elements

● Selected Elements

- □ Default
  - □ Context
  - ☑ Stack
  - □ User
- □ Tests



Entire Model View

- Workshop
  - Block Definition Diagrams
    - Context
  - Packages
    - Default
    - PredefinedTypes (REF)
    - PredefinedTypesCpp (REF)
    - Tests
      - Components
        - **Simulation**
          - Configurations
            - **Simu**
      - Panel Diagrams
      - Sequence Diagrams
  - Profiles

Configuration : Simu in Simulation

General | Description | Initialization | Settings | Checks | Relations | Tags | Properties

Initial instances

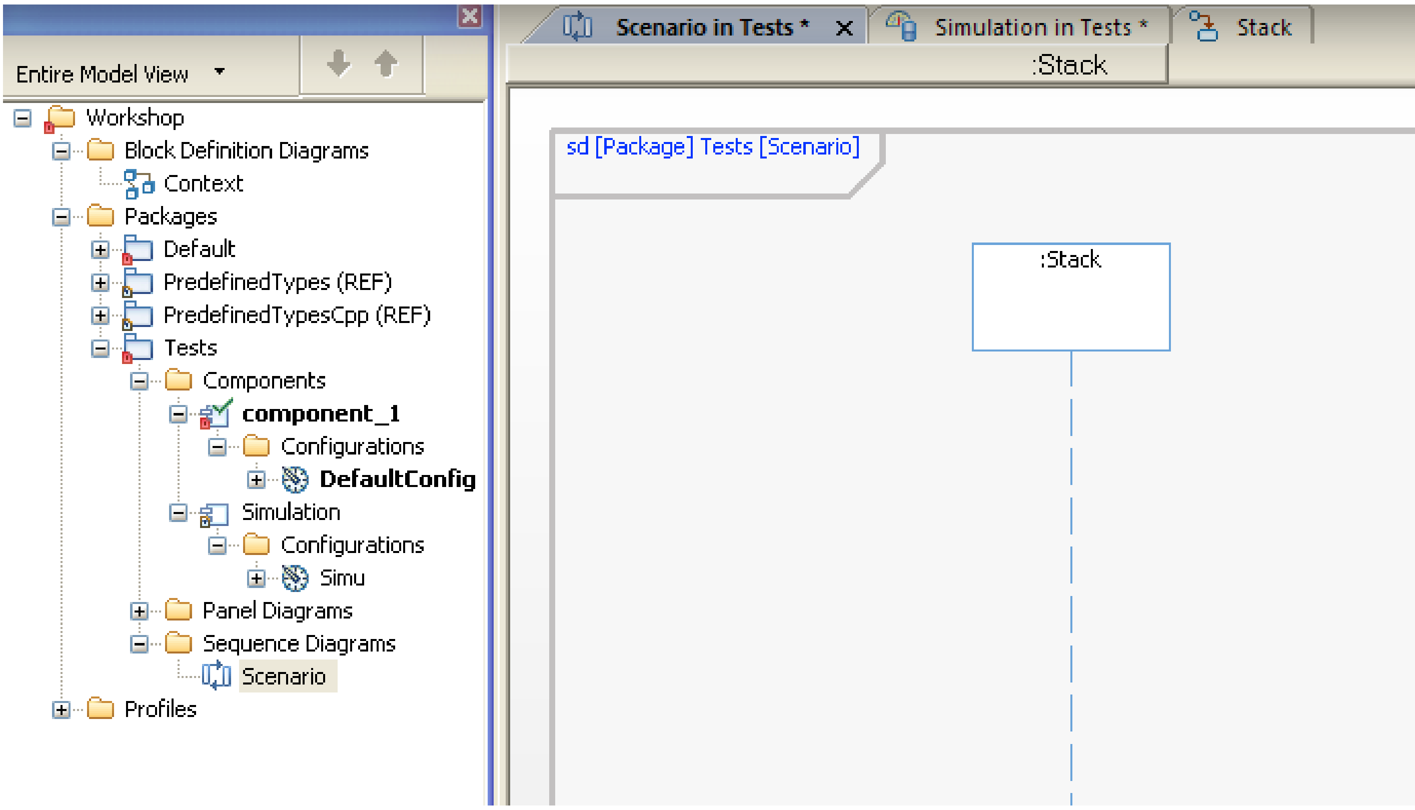
● Explicit ○ Derived

- Default
  - □ Context
  - ☑ Stack
  - □ User
- Tests

☑ Generate Code For Actors □ Generate Code For Usecases

Initialization code

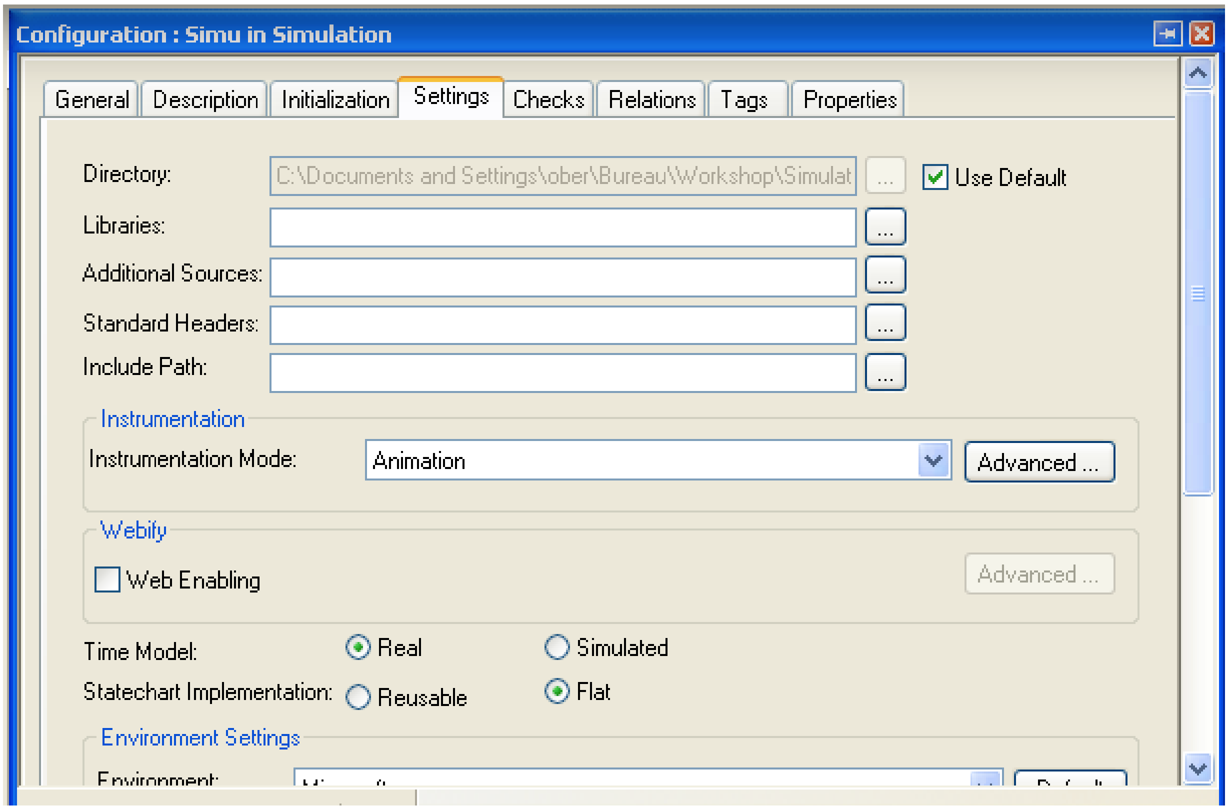
- If you have time only, create also a sequence diagram in your Test package with your Stack as a lifeline. This step is not mandatory.



1.5. Animation of the model

Everything is ready now for animation.

- Go to the "setting" part of your simulation configuration and make sure the Instrumentation mode is in "Animation".



Configuration : Simu in Simulation

Directory: C:\Documents and Settings\ober\Bureau\Workshop\Simulat ☑ Use Default

Libraries:

Additional Sources:

Standard Headers:

Include Path:

Instrumentation

Instrumentation Mode: Animation

Webify

☐ Web Enabling

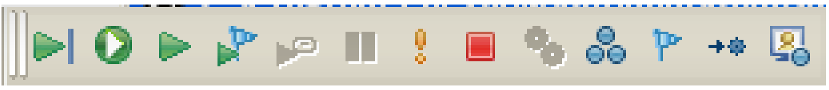
Time Model: ● Real ○ Simulated

Statechart Implementation: ○ Reusable ● Flat

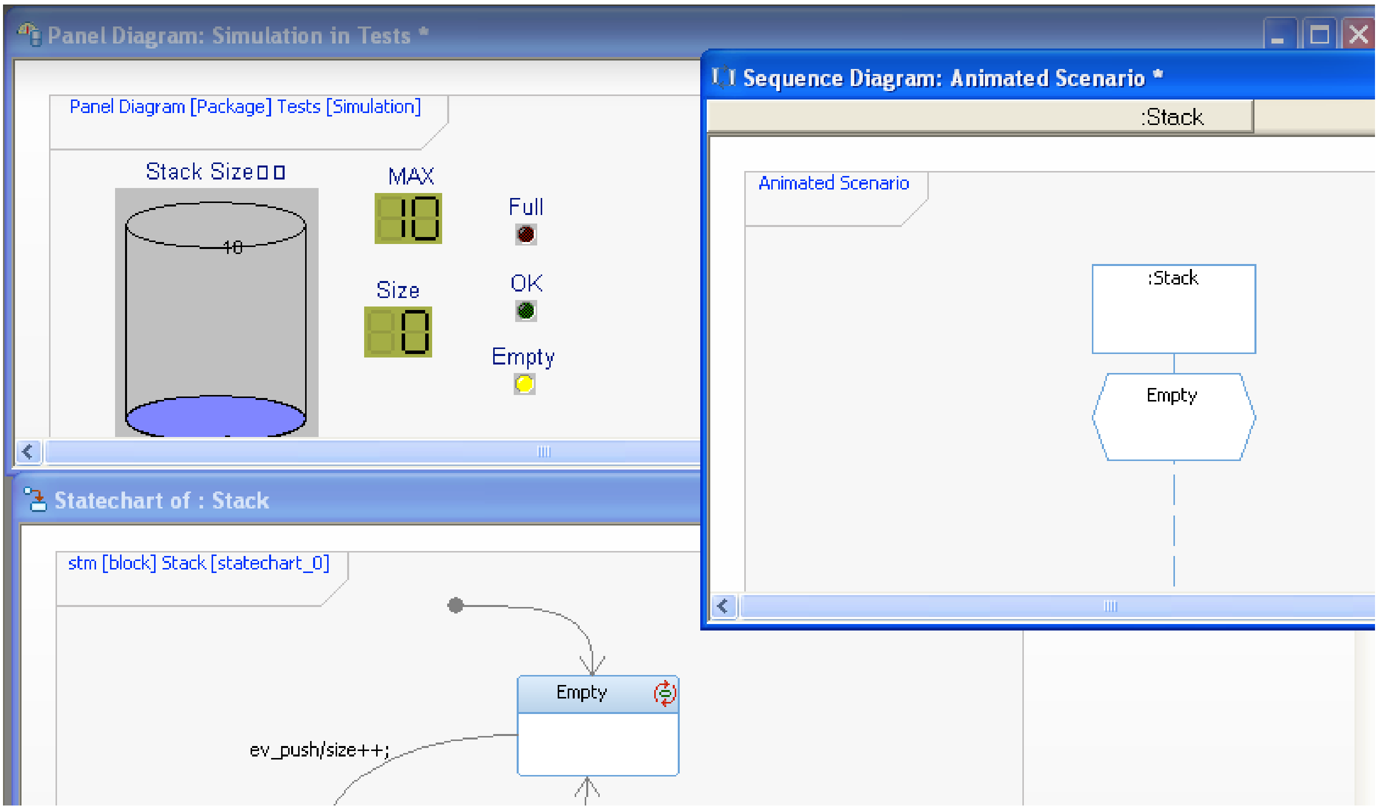
Environment Settings

Environment:

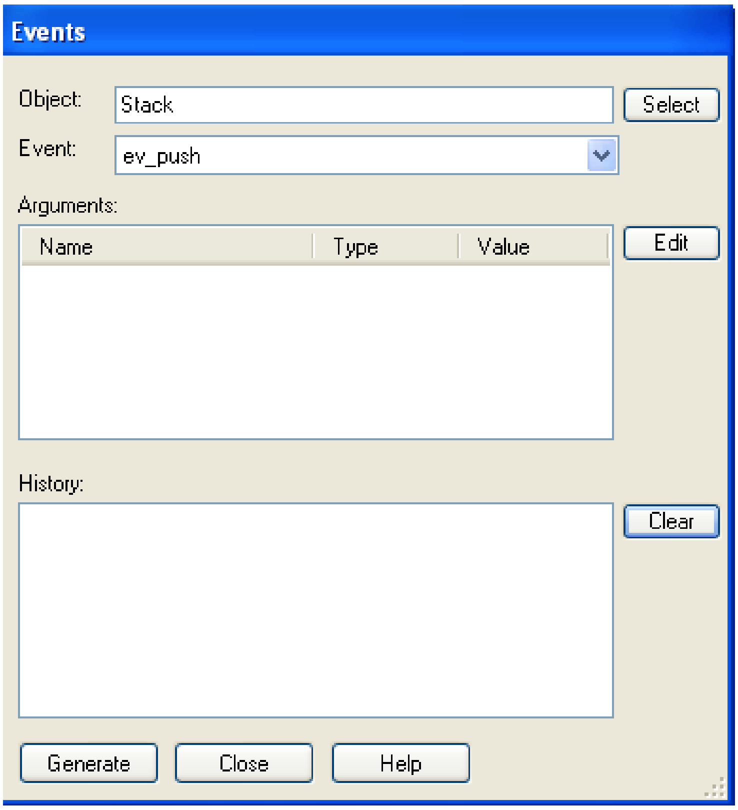
- "Build" (construct all the code generation) by clicking on the Generate/Make/Run Button.
- If it is not opened, open the Animation panel.



- Click on "Go Iddle" to initiate the animation. See the corresponding State Machine, Panel Diagram and Sequence diagram initialisations!


Panel Diagram: Simulation in Tests *
Panel Diagram [Package] Tests [Simulation]
Stack Size
10
MAX
10
Full
Size
0
OK
Empty
Statechart of : Stack
stm [block] Stack [statechart_0]
Empty
ev_push/size++;
Sequence Diagram: Animated Scenario *
:Stack
Animated Scenario
:Stack
Empty

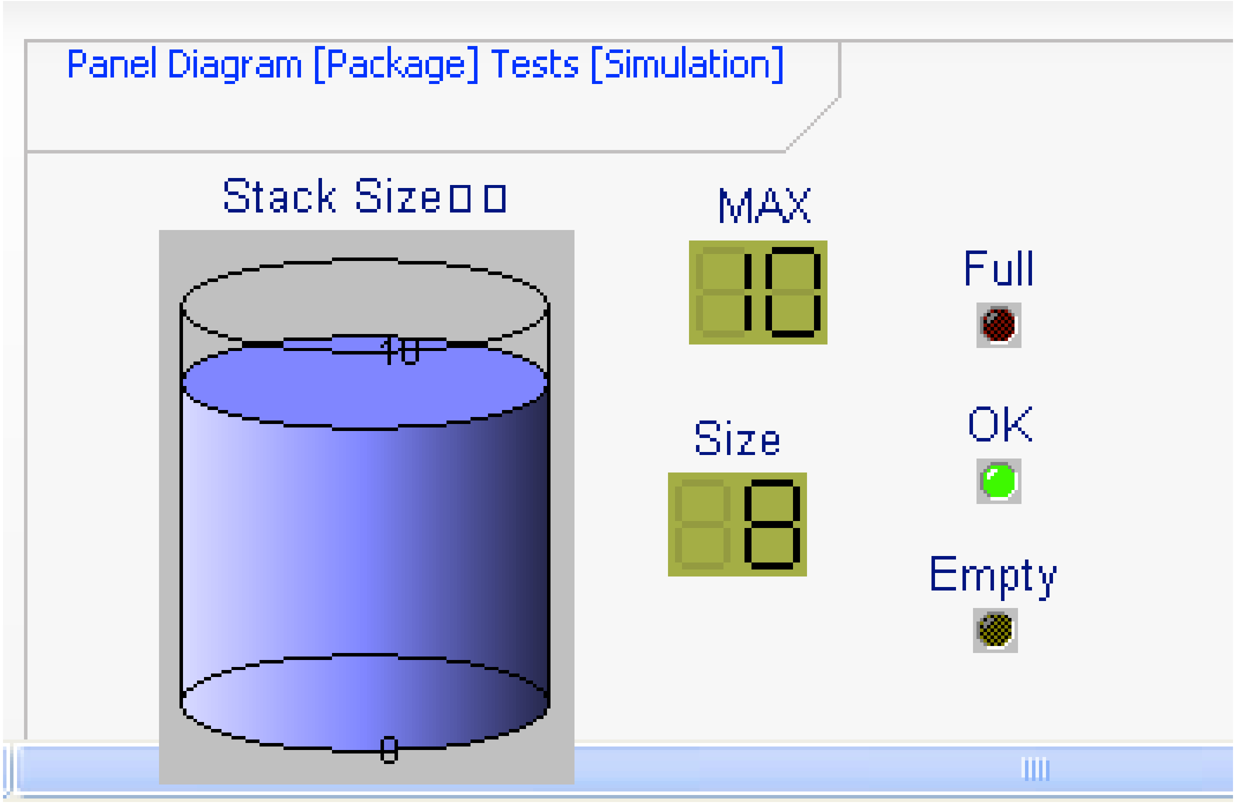
- Click on "Go". the animation is now waiting for an event to occur.
- Click on the "Event Generator" and produce a push for example.



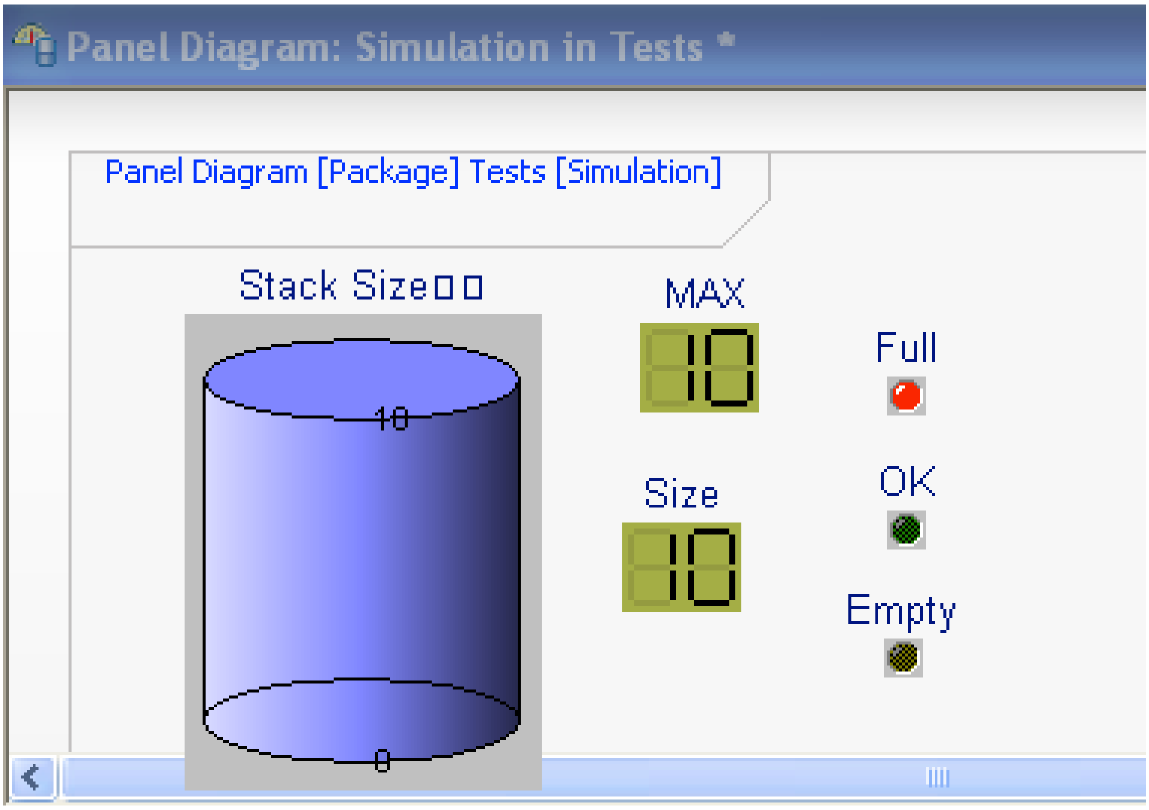
Object: Stack

Event: ev_push

- See your Stack growing as much as you push.



- Until it reaches the "Full" state.


Panel Diagram: Simulation in Tests *
Panel Diagram [Package] Tests [Simulation]
Stack Size□□
MAX
10
Full
OK
Size
10
Empty
10
0

1.6. Other SysML diagrams

Play around with the other SysML diagrams.

1.7. Report on your project

You can generate the documentation associated to your project (as a "souvenir" ;-):

- Click on Tools → ReporterPLUS → Report on all models elements,
- Select "Generate Micro$oft Word Document",
- Select "SysMLreport.tpl" as a report (for example),
- Choose a name (and a place) for your report.
    - You can modify the documentation template
    - You can generate also Powerpoint presentations instead!

2. Tips and tricks

save regularly
watch out the Murphy’s law

check regularly
use the power of tools

3. Links

- A set of slides (in French) of the RadioClock case study from Pascal Roques [http://www.prfc.com/] is available here [http://goo.gl/GVhhX].
- A video of an animation is also available here [http://dl.dropbox.com/u/946731/PRFC_RhapsodySimulation_1.0_Demo.mp4] in case you could not make it.

4. About…

Document generated with AsciiDoc [http://www.methods.co.nz/asciidoc/] by Jean-Michel Bruel [mailto:bruel@iut-blagnac.fr]. licence Creative Commons Paternité - Partage à l'Identique 3.0 non transposé [http://creativecommons.org/licenses/by-sa/3.0/].