

Andromeda

A new OPEN PSA Workbench

Friedlhuber T.

2012-12-10 Mo

Topic

- 1 What is Andromeda
- 2 Andromeda - current Applications
- 3 Andromeda core concepts
- 4 OPSAMEF Challenges

Andromeda Objectives

- Application Framework
- Reducing complexity of models
- Reducing complexity of applications
- Open, Extensible Application Architecture
- Connecting applications
- Connecting models
- Simplify to develop new modelling applications

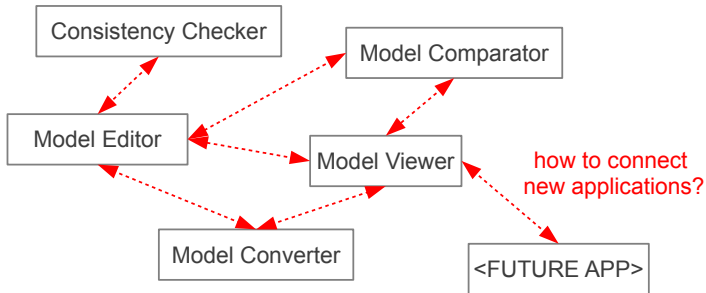
PSA related goals:

→ Understanding risk models

→ Improvement of qualitative risk assessment

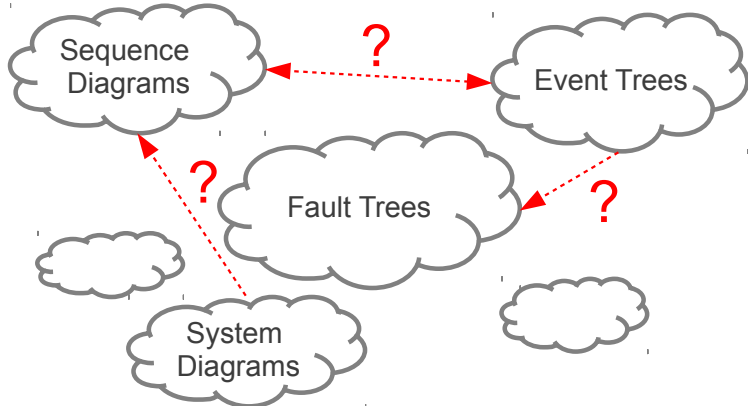
Application philosophy

- Each application is dedicated to fulfill a certain task
- Applications are kept simple
- Applications are interconnected



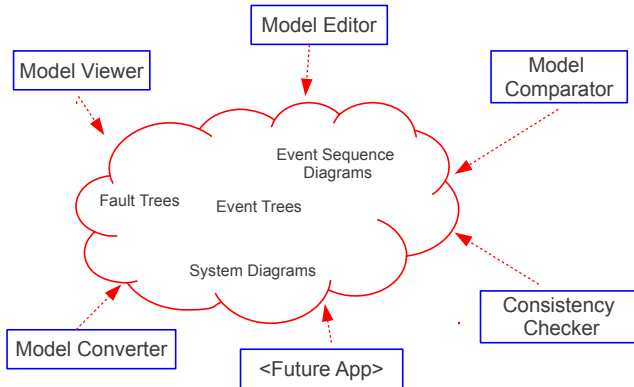
Modelling philosophy

- Models have often dependencies between each other
- Andromeda respects having an interconnected model world



Applications operate together on models

- Models are shared between applications
- Models are no longer treated independently from each other



Topic

- 1 What is Andromeda
- 2 Andromeda - current Applications**
- 3 Andromeda core concepts
- 4 OPSAMEF Challenges

Model Viewer

The screenshot shows the Model Viewer interface with the following components:

- Navigator:** A tree view on the left showing a hierarchy of sequences and attributes. The selected element is 'Attributes' under 'Sequence --RPURB-E-1&C02'.
- Diagram:** A central area displaying a sequence diagram with colored blocks (red, green, yellow) and connecting lines. The diagram is zoomed to 29x.
- Attributes Table:** A table at the bottom left showing details for the selected element:

name	value
name	--RPURB-E-1&C02-1&C03-
label	
modified	2009-02-19T12:57:16
modifiedBy	BDE
- Search Results:** A panel on the right showing search results for 'Search (101)'. It includes a 'Type Filter' set to '<ALL>' and a 'Name Filter' field. Below is a table of found elements:

Name	Type	Li
%1/A1	Basic_Event	Li
%1/A2	Basic_Event	Fr
%1/A3	Basic_Event	Fr
%1/B1	Basic_Event	Fr
%1/B2	Basic_Event	Fr
%1/B3	Basic_Event	Fr
%1/B4	Basic_Event	Fr
%1/CA1	Basic_Event	Fr
%1/CA2	Basic_Event	Fr
%1/CA3	Basic_Event	Fr
%1/CA4	Basic_Event	Fr
%1/CB1	Basic_Event	Fr
%1/CB2	Basic_Event	Fr
%1/DB1	Basic_Event	Fr
%1/D2	Basic_Event	Fr
%1/E	Basic_Event	Fr
%1/F	Basic_Event	Fr
%A/ABCD	Basic_Event	Ti
%A1/ABCD	Basic_Event	Ti
%A2/ABCD	Basic_Event	Ti
%A3/ABCD	Basic_Event	Ti
%B/ABCD	Basic_Event	Ti



Model Editor

The screenshot displays the Andromeda Model Editor interface. The main window shows a fault tree diagram with several gates and events. A 'Create Gate' dialog box is open, allowing the user to create a new gate. The dialog contains the following fields:

- *Name: new_Gate
- Label:
- Parent: Fault_Tree %COEF_B1B2B3

The dialog has 'Cancel' and 'Create' buttons. The main diagram shows a fault tree structure with gates like '%COEF_B31' (highlighted in yellow) and '%COEF_B1B2B3'. The diagram also includes events like '%B2 Sous-état b2', '%B3ABCD Time spent in state B3 (20h) / Time spent in state ABCD (8406h)', and '%B3 Sous-état b3'. The interface includes a Navigator on the left, a Search bar at the top right, and a list of found elements on the right.

Found elements:

Name	Type
%COEF_A1A2A3	Fault_Tree
%COEF_A1A2A3_2	Fault_Tree
%COEF_ABCD-1	Fault_Tree
%COEF_ABCD-2	Fault_Tree
%COEF_B1B2B3	Fault_Tree
%COEF_B1B2B3_2	Fault_Tree
%COEF_B1B2B3B4	Fault_Tree
%COEF_CA1CB2	Fault_Tree
%COEF_CA1CB2_2	Fault_Tree
%COEF_D1D2	Fault_Tree
%COEF_D1D2_2	Fault_Tree
%COEF_EF	Fault_Tree
%COEF_EF-2	Fault_Tree
%COEF_RH_A	Fault_Tree
%COEF_RH_B	Fault_Tree
%COEF_RH_C	Fault_Tree
%COEF_RH_D	Fault_Tree
---FLOOP-AD	Fault_Tree
---FLOOP-EF	Fault_Tree
---FLSBO-D	Fault_Tree
---FLSBO-EF	Fault_Tree
---FLUHS-AD	Fault_Tree
---FLUHS-EF	Fault_Tree

Attributes:

name	value
name	@%COEF_B31
label	Coefficient of the substae
modified	
modifiedBy	
gatetype	and

Consistency Checker

The screenshot displays the Consistency Checker (v1.0.1_int_qualifier) interface. The main window shows a list of unresolved references for the model `/home/friedlhu/models/EPS_EPR_BK_V_JANVIER.opsamef` in the `SUPER SET` variant. The list includes various parameter references such as `X3PORTES_CF_DS`, `X1CLAPET_CF_DS`, and `XFT09_NEXT_LOC_FAIOP`, all marked as `MAJOR` errors. The interface also features a Consistency Navigator on the left, an Attributes table at the bottom left, and a toolbar with Check, Reset, Export, and Memory buttons.

Type	Name	Sort	Element Type	Limit
Basic_Event	X3PORTES_CF_DS	MAJOR	Parameter Reference 'X3CLAPETS_CF_DS': Unresolved Reference	100
Basic_Event	X1CLAPET_CF_DS	MAJOR	Unresolved referenced Parameter Reference 'X3PORTES_CF_DS': Unresolved Reference	100
Basic_Event	XFT09_NEXT_LOC_FAIOP	MAJOR	Parameter Reference 'XCLAPET_CF_DS': Unresolved Reference	100
Basic_Event	XFT71_NEXT_E_LOC_01M	MAJOR	Parameter Reference 'XCLAPET_CF_DS': Unresolved Reference	100
Basic_Event	XFT71_NEXT_H_LOC_01M	MAJOR	Unresolved referenced Parameter Reference 'XFT09_NEXT_LOC_FAIOP': Unresolved Reference	100
Basic_Event	XFT71_NEXT_M_LOC_01M	MAJOR	Parameter Reference 'XFT09_NEXT_LOC_FAIOP': Unresolved Reference	100
Basic_Event	XFT71_NEXT_S_LOC_01M	MAJOR	Parameter Reference 'XFT71_NEXT_E_LOC_01M': Unresolved Reference	100
Basic_Event	XFT71_NEXTTEM_LOC_01M	MAJOR	Parameter Reference 'XFT71_NEXT_H_LOC_01M': Unresolved Reference	100
Basic_Event	XFT71_NEXTTPC_LOC_01M	MAJOR	Parameter Reference 'XFT71_NEXT_M_LOC_01M': Unresolved Reference	100
Basic_Event	XFT71_NEXTTF_LOC_01M	MAJOR	Parameter Reference 'XFT71_NEXT_S_LOC_01M': Unresolved Reference	100
Basic_Event	XFT72_NEXT_E_LOC_02M	MAJOR	Parameter Reference 'XFT71_NEXTTEM_LOC_01M': Unresolved Reference	100
Basic_Event	XFT72_NEXT_H_LOC_02M	MAJOR	Parameter Reference 'XFT71_NEXTTPC_LOC_01M': Unresolved Reference	100
Basic_Event	XFT72_NEXT_M_LOC_02M	MAJOR	Parameter Reference 'XFT71_NEXTTF_LOC_01M': Unresolved Reference	100

Model Comparison

Model Comparison

Model A: /home/friedlhu/models/EPR2.opsamef Variant: SUPER SET
 Model B: /home/friedlhu/models/EPR.opsamef Variant: SUPER SET

Type: <ALL> Name: Sort: Element Type Limit: 100

Comparison Result:

Type	Element	Severity	Differences
Fault_Tree	%COEF_CA2/CA	MAJOR	show Basic Event Reference '%CA2/CA' removed from Fault_Tree '%COEF_CA2/CA'
Event_Tree	#PDS-PSP1	MAJOR	show Fork removed from Event_Tree '#PDS-PSP1' (Model B)
Fault_Tree	%COEF_B1/AC	MINOR	show Attribute 'label' of Gate '%COEF_B1/AC' changed from 'Fraction of the year in
Fault_Tree	%COEF_CA4/CDE	MAJOR	show Gate Reference '@%COEF_CA4/CDE-1' removed from Fault_Tree '%COEF_CA4'
Fault_Tree	%COEF_B1/B	MAJOR	show House Event Reference '%B1' removed from Fault_Tree '%COEF_B1/B' (Mode
Event_Tree	#PDS-PSP1	MAJOR	show Sequence Reference '#PDS-PSP1-#PDS-LABEL-PSP1' added to Event_Tree '#

Attributes Compare

name	value (Model A)	value (Model B)
name	%COEF_B1/B	%COEF_B1/B
label	Fraction of the state in	Fraction of the state in
modified		
modifiedBy		

Compare Diagram

Model A: Zoom 86 Textsize 80
 Model B: Zoom 55 Textsize 80

Variant Management

The screenshot displays the Andromeda Model Editor interface. The main window shows a sequence diagram with a timeline at the top and a state transition diagram below. The timeline is divided into segments with different colors (red, green, blue). The state transition diagram shows various states and transitions between them, with some states highlighted in blue and red.

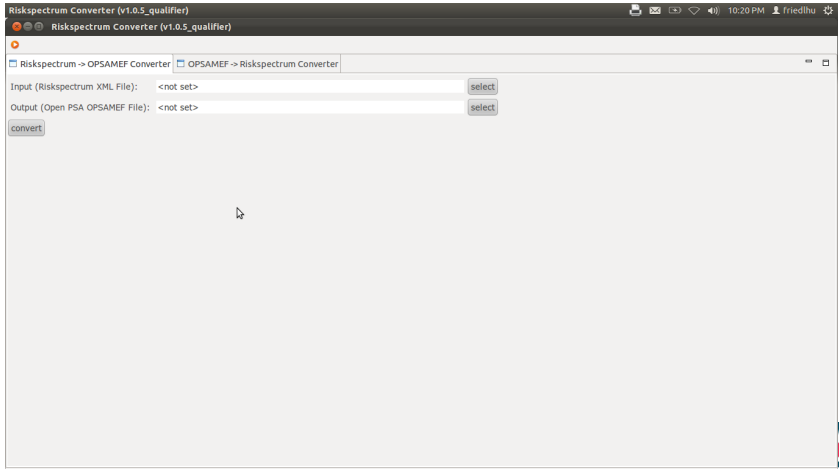
On the left, there is a Navigator pane showing a tree structure of sequences and attributes. Below it, an Attributes pane shows details for a selected attribute:

name	value
name	--HRPDL-E-I&C03-
label	
modified	2009-04-02T12:41:08
modifiedBy	BDE

On the right, a Search pane shows the results of a search for "Event Tree". The results are listed in a table:

Name	Type	Label
--FLOOP-AD	Event_Tree	Total I
--FLOOP-EF	Event_Tree	Total I
--FLSBO-AD	Event_Tree	Total I
--FLSBO-EF	Event_Tree	Total I
--FLUHS-AD	Event_Tree	Total I
--FLUHS-EF	Event_Tree	Total I
--FP3TB-AD	Event_Tree	Break
--FPCC1-AD	Event_Tree	Total I
--FPCC1-EF	Event_Tree	Total I
--FPCC2-AD	Event_Tree	Total I
--FPCC2-EF	Event_Tree	Total I
--FPCIL-AD	Event_Tree	Total I
--FPCIL-EF	Event_Tree	Total I
--FPLUH-AD	Event_Tree	Total I
--FPLUH-EF	Event_Tree	Total I
--FPMTB-AD	Event_Tree	Break
--FPMTB-E	Event_Tree	Break
--FPURB-AD	Event_Tree	Break
--FPURB-E	Event_Tree	Break
--HEPWT-AD	Event_Tree	Water
--HRPDL-E	Event_Tree	Spurio
--HRPDL-E	Event_Tree	Involu

Model Converter



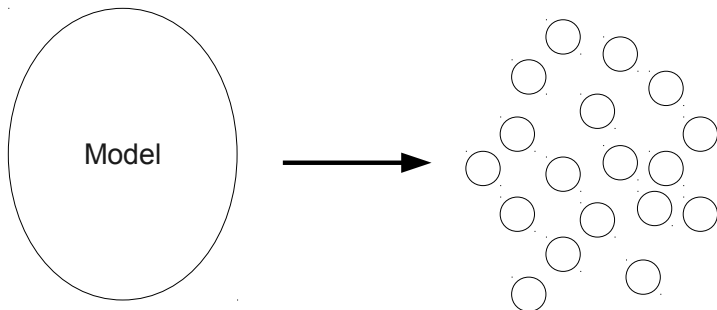
Topic

- 1 What is Andromeda
- 2 Andromeda - current Applications
- 3 Andromeda core concepts**
- 4 OPSAMEF Challenges

Modular PSA

Modular PSA splits model into pieces (modules)

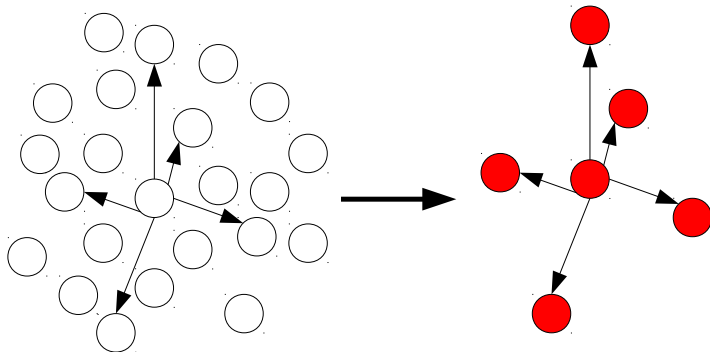
Each piece is treated rather independently



Connecting modules

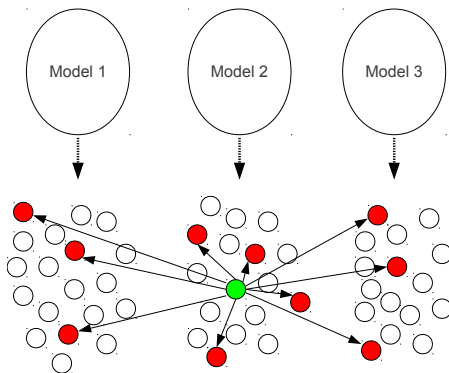
Related model components (the modules) can be linked together

Model gets reduced to a less complex one:



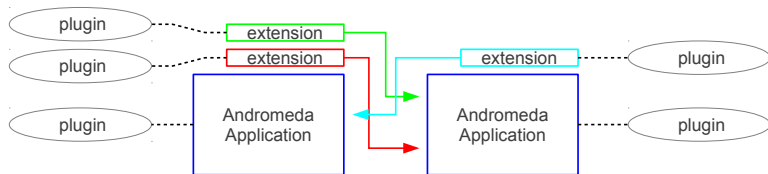
Connecting various models

Idea can be extended to connect modules of various models



Extension points

- Andromeda is based totally on plugins
- Each application can be extended
- Interconnection between applications done by adding new extension points



Topic

- 1 What is Andromeda
- 2 Andromeda - current Applications
- 3 Andromeda core concepts
- 4 OPSAMEF Challenges

What are the current challenges of integrating OPSAMEF in Andromeda?

Format is often too complicated:

- Gate definitions
- Event Tree formula collection
- Non-Existence of Alternatives, Exchange Events, Mutual Exclusions, Delete Terms
- Substitutions are not intuitive

But on the other hand compatibility needed!

Goal: Find right compromise, only provide what is really needed!