Tesnim Abdellatif & Mohamed Hibti

EDF R&D November 2015

Andromeda Comparison

A small tutorial for a quick start

Contents

1 Introduction **3**

2 Why do we need the comparison functionality? **3**

3 Concepts **4**
3.1 Matches . . . 4
3.2 Detailed Differences . . . 4
3.3 Filters . . . 5
3.4 Profiles . . . 5

4 Before beginning the tutorial :required input data **5**

5 Use cases **6**
5.1 Comparing Parameters Types . . . 6
5.1.1 Application Launch . . . 6
5.1.2 Tool Navigation . . . 8
5.1.3 Comparison Results Exploration . . . 9
5.1.4 Export of the comparison result . . . 10
5.2 Comparing Fault Trees types . . . 13
5.2.1 Application Launch . . . 13
5.2.2 Profile Creation . . . 16
5.2.3 Filters Use . . . 17
5.2.4 Web export . . . 19
5.3 Comparing Event trees types . . . 24
5.3.1 Application Launch . . . 24
5.3.2 Results Exploration . . . 25
5.3.3 Advanced Filters . . . 27
5.3.4 Review Export . . . 33

List of Figures

1 Andromeda provides different view boxes for models and the comparison result 8
2 Model and Folder views . . . 11
3 Attribute view for parameter named "AAR"* . . . 12
4 Comparison result of the parameters example . . . 12
5 Overview of the Fault trees example exploration . . . 14
6 Fault trees folder view . . . 15
7 Result view for the fault tree example . . . 16
8 Component filtering selection List . . . 17
9 Filtering result for the fault tree example . . . 18
10 Textual comparison of fault trees . . . 19
11 Graphical comparison of fault trees . . . 20
12 Web view start page . . . 21
13 Web view Fault trees modifications . . . 22
14 Graphical comparison of fault trees in the web view . . . 22
15 Textual comparison of fault trees in the web view . . . 23
16 Andromeda view boxes for Event Tree components . . . 25
17 Event trees comparison result . . . 26
18 Example for sequence modifications . . . 27
19 Dialog box for the advanced filter configuration . . . 28
20 Single Type Filter Components list . . . 29
21 Multi Type Filter Components list . . . 30
22 Extract of a review in word format . . . 34
23 Extract of the review in HTML Format . . . 35

1 Introduction

PSA models are widely used in the field of probabilistic risk assessment for complex physical systems such as nuclear power plants. Most of these fault trees and event trees models are quite complex. Unfortunately, their increasing size makes difficult to develop them any further. Especially today, where often multiple engineers develop in parallel on the same PSA model, it can become difficult to trace and verify modifications.

To be aware about model differences is a crucial preliminary step for several tasks:

- To verify, analyze, cross-check model modifications: The differences between models give an important feedback "what has been done" or "what has been modified" since an earlier point in time (e.g. a previous model version).
- To fusion models: Model fusion consists in merging the differences between models.
- To automatically generate modification reports: Typically, PSA model modifications are to justify to control authorities. Generating reports automatically is not only efficient, it also ensures consistency between models and documentation.

In this tutorial, we present how to compare PSA models in order to give engineers important feedback about model modifications. The main objective of this small guide is to give the user some key elements to explore the functionalities provided by Andromeda for comparison purposes. It is based on a how to approach for practical reasons. The tutorial do not cover all the functionalities of Andromeda which are available in the tools and may be used by the user beyond comparison functions. These functionalities are not under quality assurance and may not work as expected. The user can also refer to the user guide (cf. [1]) for more detailed explanations about the tools features.

2 Why do we need the comparison functionality?

PSA models are generally very complex and require a good quality assurance. One of the aspects of this quality assurance is the efficient control of the model evolutions to guaranty compliance with standards and to ensure that models reflect the reality of plants. However, in the database architecture of currently used PSA tools, only meta-data information can be obtained concerning model modifications. Analysts (users), developers and reviewers may need to have deep insights on different model transitions (set of modifications), and then go through details in order to verify and justify (for example to safety authorities) the set of modifications applied to a PSA model. Currently, those activities are performed manually and can be time-consuming and error-prone since PSA models may contain dozens of thousands of model objects.

Moreover, for a better version control of PSA models, comparison is the first main step to get it done. Therefore, teams may then work on a distributed basis and have the possibility to merge models for a better organization.

3 Concepts

The Andromeda functionality for comparing models is a possibility to determine the differences between two models "Model A" and "Model B", to visualize them and to export them in various formats.

3.1 Matches

The result of a model comparison is represented as a set of matches. Each match states which model component of "Model A" corresponds to which one of "Model B".

Matches themselves can contain further matches, referred to as submatches. In case of PSA models, this is the case for fault and event tree matches:

- Gate matches are submatches of fault tree matches.
- Sequence matches are submatches of event tree matches.

Each match provides several information about its matched components CA and CB:

- Match Type: Gives feedback about the kind of differences:
  - `EQUAL`: CA and CB do not differ. Matches of this type are never displayed.
  - `MODIFIED_SLIGHTLY`: CA and CB differ "slightly" not impacting quantification results.
  - `MODIFIED`: CA and CB differ and may impact quantification results.
  - `A_ONLY`: CA could not be matched to a component in "Model B": CB is "null".
  - `B_ONLY`: CB could not be matched to a component in "Model A": CA is "null".
- Severity: Gives feedback about the severity of the differences:
  - `MAJOR`: the differences may impact quantification results.
  - `MINOR`: the differences do not impact quantification results.

3.2 Detailed Differences

Matches provide an overview about what has been changed and about the kinds of modifications. However, an analyst may want to analyze a match further in order to obtain deeper insights about modiights about alyze a match further in order to obtain deeperout the kinds of modifications. However, an herefore, teams may then work.

3.3 Filters

Analysts are typically interested in focusing on a particular subset of matches. Andromeda provides three kind of filters for this purpose:

1. Component Filter: The "Component Filter" specifies a set of component types. Matches are then filtered due to their component type.

2. Simple Filter: The "Simple Filter" permits to filter matches by name, by match type, by severity etc. It is called simple filter as it requires few editing efforts (from a useres perspective).

3. Advanced Filter: The "Advanced Filter" permits to express more complicated constraints. Those complicated constraints are Boolean formula over criteria. And each criterion is a predefined Boolean function that may or may not be satisfied by matches.

All three filters can be activated / deactivated. A match is required to satisfy all activated filters at a time (AND logic).

3.4 Profiles

As the specification of filters can be time-consuming (in particular the "Advanced Filter"), Andromeda offers a possibility to save and load filters via so-called "profiles". Technically, a profile

is a configuration file that stores filter information but also further information for example those relevant for exporting comparison results.

The following information are stored in a profile:

- Profillle name and description
- Filter (Component Filter, Quick Filter and Advanced Filter)
- Export Options for Web Export and Review Export

4 Before beginning the tutorial :required input data

The content of the folder provided with andromeda includes the Andromeda binaries for different OS platforms (windows 32 and 64, Linux 32 and 64 and MacOS). This tutorial include a number of exercices for which you can find the different models in the directory `./EXERCICES/`. These models have the extension `.psa` and can be used in this andromeda version to illustrate the comparison concepts.

5 Use cases

In this section, we will explain through different use cases how to use the comparison tool in order to compare different kind of PSA models. The models we are dealing with in the next sections are only dedicated to the demonstrations and as you may notice do not represent necessarily real systems. The parameters are also virtual and may be fictive.

In the first sub-section 5.1, we use a simple case study dealing with models that define components of the same type, which are parameter components.

In the second sub-section 5.2, we will explain some comparison features through a bigger model, containing fault trees and data that are used for their construction, that is, basic events, parameters, references and CCFs. We will show the use of profiles and filters (Component and Quick filters) that are very useful for analyzing the results. We will also explain the web export feature in order to explore the results in a web browser outside the tool, when needed.

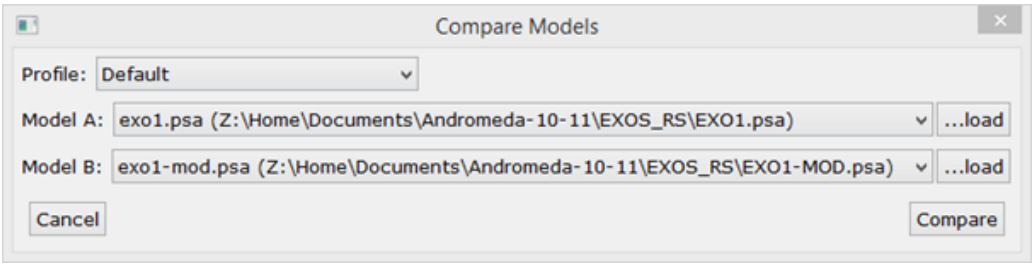
In the last sub-section 5.3, we have chosen to deal with models that include event trees. We will show the use of advanced filters that allow to define more complex filter expressions on components and we will show how to export the results in a Word document or HTML format in order to generate reports and share more easily the results.

5.1 Comparing Parameters Types

Through this first use case, we will see how modifications are managed by the comparison tool in order to understand and interpret the different informations given by the comparison result. We will examine a small model containing a certain number of parameters. The model contains only parameters which are not related to any other objects, but generally, parameters are associated to basic events with failure probabilities or failure probability distributions in order to be used in system risk analysis.

5.1.1 Application Launch

Launch the comparison for EXO1.psa and EXO1-mod.psa which are provided in the folder *Exercises* joined with this tutorial. You can launch the comparison application when you click on the comparison item in the toolbox. You can then download the two models via the following dialog box. We will generally say that the first model is the one selected for Model A and the second model is the one selected for Model B.


Compare Models
Profile:
Default
Model A:
exo1.psa (Z:\Home\Documents\Andromeda-10-11\EXOS_RS\EXO1.psa)
...load
Model B:
exo1-mod.psa (Z:\Home\Documents\Andromeda-10-11\EXOS_RS\EXO1-MOD.psa)
...load
Cancel
Compare

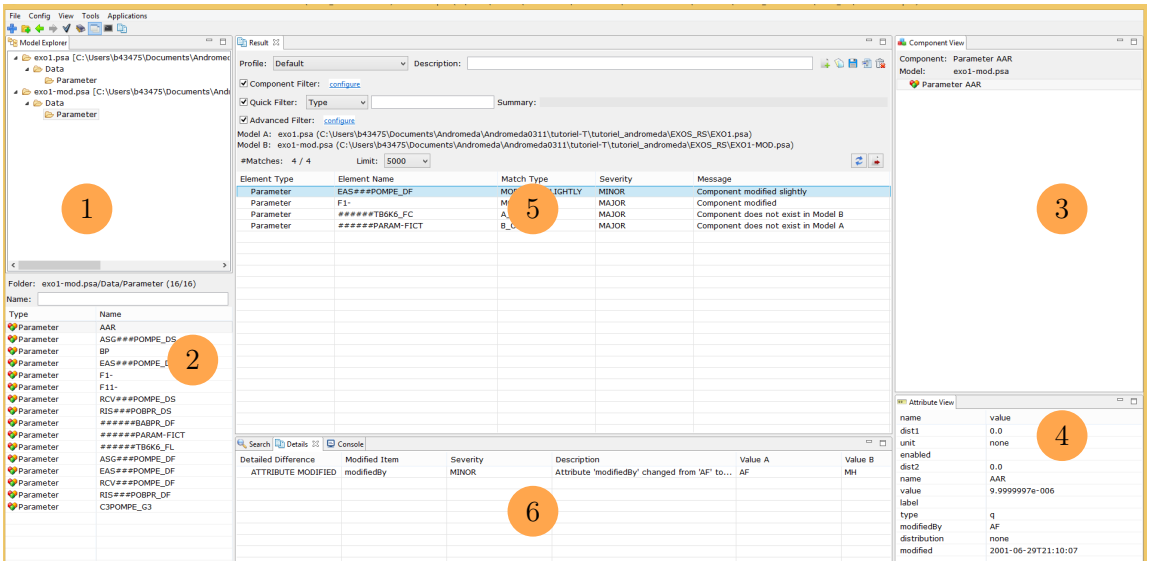


Figure 1: Andromeda provides different view boxes for models and the comparison result

5.1.2 Tool Navigation

The list of matches is given in the result window of the tool. Some basic information is also given by the match type. It gives a first quantification about the importance of differences between the compared components (`EQUAL`, `MODIFIED_SLIGTHLY`, `MODIFIED`, `A_ONLY`, `B_ONLY`). The severity gives feedback about the severity of the differences (`MAJOR` or `MINOR`). For each match, we also obtain a deeper insight of modifications with details such as the changes in the values or the date of their modifications. Those modification details are given in the "detailed analysis" view box of the tool.

Before we explain the comparison result, let's recall the aim of the different view boxes in order to understand how to navigate in the models with Andromeda (see Figure 1).

The IHM provides six view boxes:

1. In the model Explorer view, you can notice that parameters components are stored in the folder **Data -> Parameter**. This folder manages the types of components that compose the model.

2. In the folder view, you can see the content of the Parameter Folder containing the list of the model parameters (see Figure 2).

3. In the component view, you can see the information details for each item of the list whenever you click on it. In our example, the component parameter is a simple component that doesn't have any composed component in it.

4. In the attribute view, you can see the different attributes of each selected component. In our example, we can see the parameter type attributes for parameter "AAR" (unit, enabled, dist2 ..) (see figure 3)

5. In the result view, we can see filters configuration section and the set of matches of the comparison result. We should note that there is a limit the user can specify to avoid charging a huge number of differences if it applies.

6. In the details view, we can view for each match their detailed modifications, that is, the detailed difference, the modfified item of the component, the severity and modified values.

5.1.3 Comparison Results Exploration

In principle, the comparison consists in finding *matches* between two models A and B. The result of the comparison rises different types of changes or **Match Types.** In the example, four parameters have been changed (see figure 4). Those parameters are `EAS###POMPE_DF`, `F1-`, `######TB6K6_FC` and `######PARAM-FICT`. For each parameter that has been changed, the match type indicates the type of modification that has been found.

Here are an analysis regarding the match types of the matches:

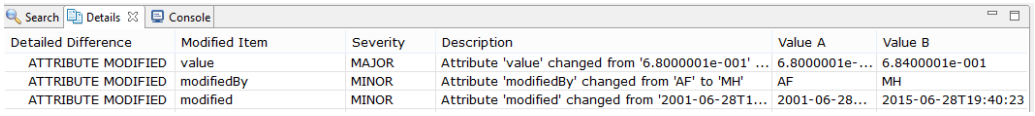
Search | Details | Console

Detailed Difference	Modified Item	Severity	Description	Value A	Value B
ATTRIBUTE MODIFIED	modifiedBy	MINOR	Attribute 'modifiedBy' changed from 'AF' to 'MH'	AF	MH

- Parameter `EAS###POMPE_DF` has been `MODIFIED_SLIGHTLY`: The parameter in model A matches the parameter in model B with "minor" differences. Slight differences represents those with minor impact such as label differences. Generally, slight modifications

concern component descriptions that do not have any impact on risk quantification. In our example, we can see in the detailed view section that the attribute `modifiedBy` has changed from `AF` to `MH`.

- Parameter `F1-` have been `MODIFIED`: That is, the parameter in A and B have major differences with major impact on risk quantification. In our example, the major severity impact of the modification is the change in the attribute `value`. It has a value "6.8000001e-001" in the first model and a value "6.8400001e-001" in the second one. Other attributes like `modifiedBy` and `modified` have also changed. They correspond to minor changes in respectively the name of the user that have done the modifications and the date of modifications.



Detailed Difference	Modified Item	Severity	Description	Value A	Value B
ATTRIBUTE MODIFIED	value	MAJOR	Attribute 'value' changed from '6.8000001e-001' ...	6.8000001e-...	6.8400001e-001
ATTRIBUTE MODIFIED	modifiedBy	MINOR	Attribute 'modifiedBy' changed from 'AF' to 'MH'	AF	MH
ATTRIBUTE MODIFIED	modified	MINOR	Attribute 'modified' changed from '2001-06-28T1...	2001-06-28...	2015-06-28T19:40:23

- Parameter `######TB6K6_FC` is `A_ONLY`: The parameter in A could not be matched to a parameter in B because it is only declared only in the first model.
- Parameter `######PARAM-FICT` is `B_ONLY`: The parameter in B could not be matched to a parameter in B because it is declared only in the second model.

5.1.4 Export of the comparison result

In addition to the Andromeda interface, it is possible to view the comparison result either in the web view (cf. section 5.2.4) in order to navigate further in the result or in a document to use in a report for instance (cf. section 5.3.4).

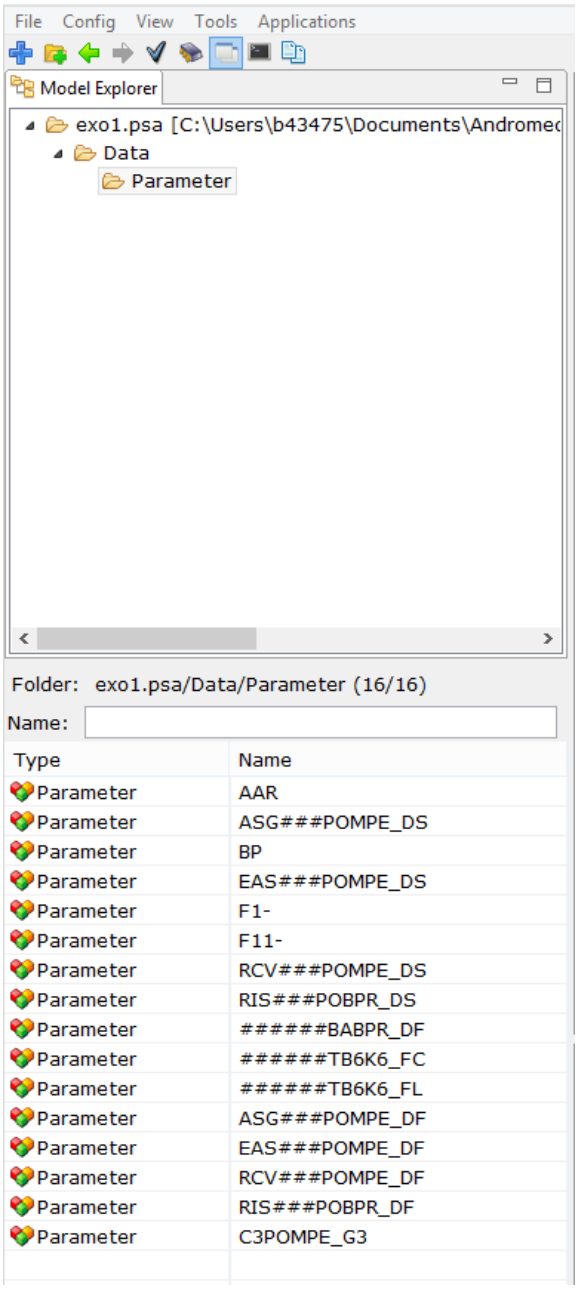


Figure 2: Model and Folder views

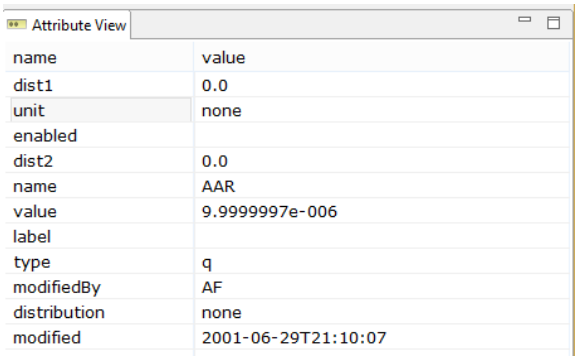


Figure 3: Attribute view for parameter named "AAR"*

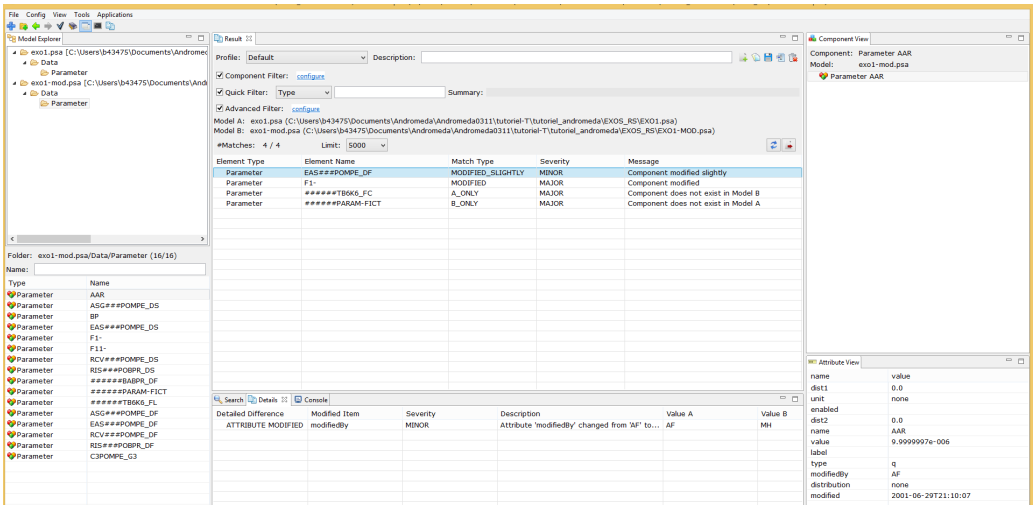


Figure 4: Comparison result of the parameters example

5.2 Comparing Fault Trees types

A fault tree encodes a Boolean formulae over events in order to express the likelihood of a so-called *top event*. Comparing fault trees involves the comparison of its constituting components, that is, its gates, basic events, CCFs, properties and parameters. In this section, we will focus on presenting the comparison result of fault trees involving those component types. We will also experiment the use of different filters in order to manage the comparison result. Finally, we will use the export function to generate the adequate documents.

5.2.1 Application Launch

Launch the comparison the comparison for `exo2.psa` and `exo2_mod.psa` that are provided in the **Exercices** folder joined to the tutorial.

In this example, we use the EFWS system (ASG) model and its fault trees corresponding to the expression of the loss of 2 out of 3 trains of the system. In the Model Explorer view, you can notice that a *Data Folder* contains all the Data components that are used in the construction of Fault Tree components. **Folder Basic-Events** contains the basic events used in the trees, **Folder Parameter** contains the parameter components that can be referenced by Basic events components for instance and so on (See 5). You can explore the list of the model fault trees by clicking on *Folder Fault-Trees* and navigating through the list in the folder view (see Figure 6). You can see that 15 fault trees are used in the second model such as Fault tree `ASG`, `ASG_VOIE2`, `ASG_VOIE1` and so on. Note that a filter can be applied in the search folder to limite by a regexp on the name the list of displayed results. When you click on a tree from the list, its diagram opens in a new window next to the result view. You can click on the different boxes corresponding either to gates that are also fault trees or basic events in order to see their details. The component view gives the list of sub-components of the selected fault tree/gate. The attribute view gives its attributes. You can also click on each sub-component in order to see its position in the diagram view (it becomes colored) and its attributes.

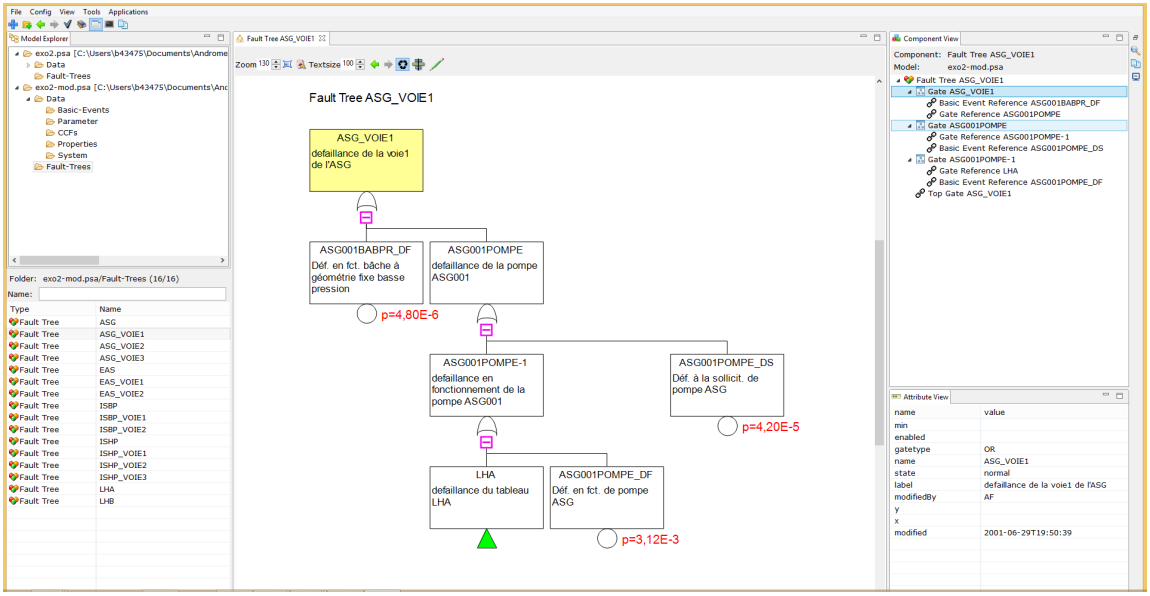


Figure 5: Overview of the Fault trees example exploration

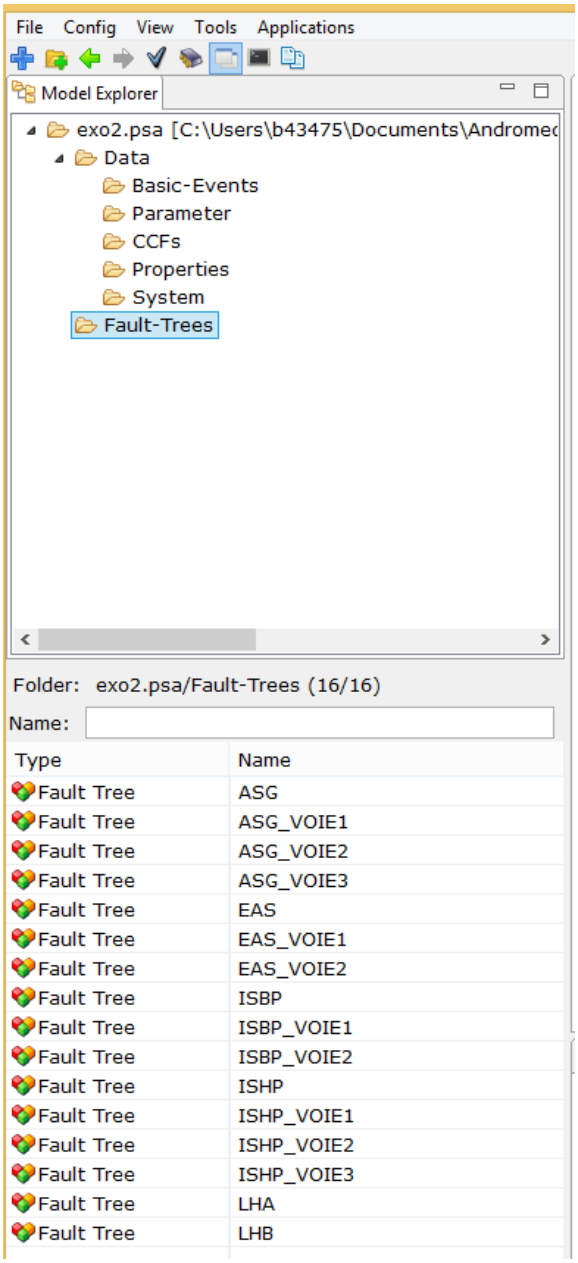
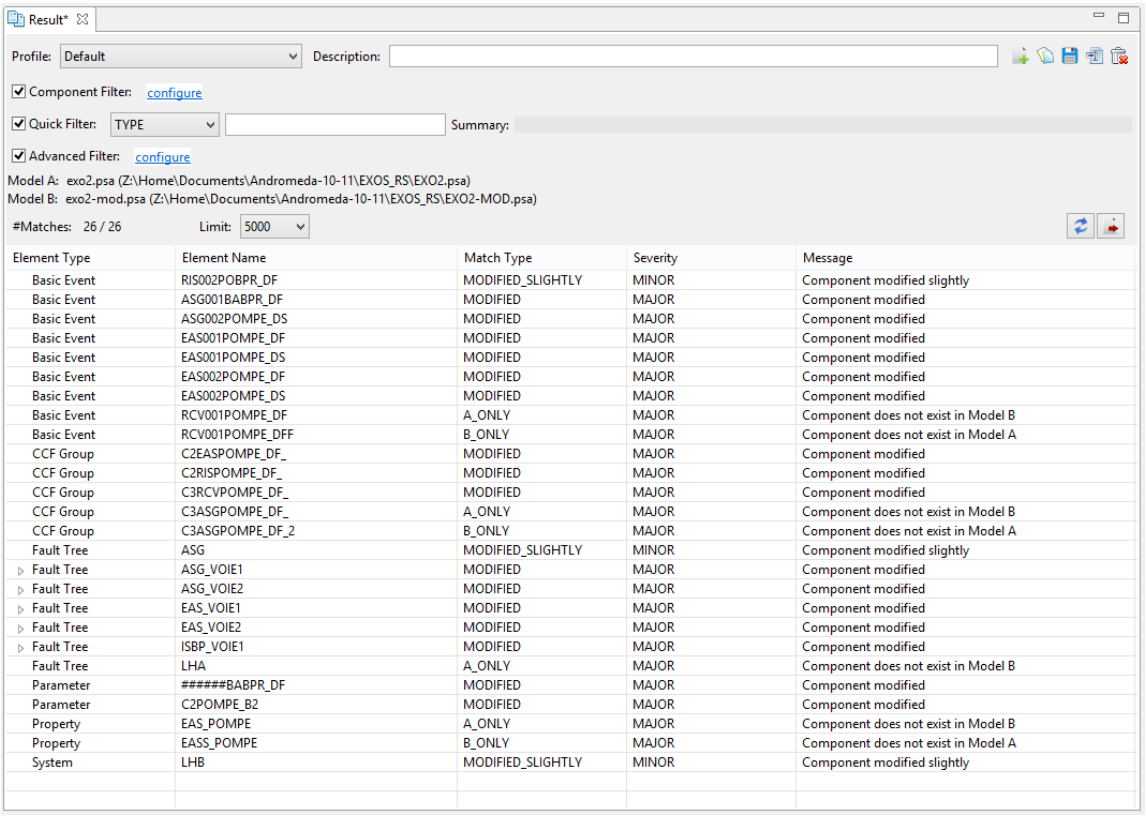


Figure 6: Fault trees folder view

5.2.2 Profile Creation

Figure 7 gives an overview of the comparison result. You can notice that the comparison result involves different types of modifications for which we are going to apply filters capabilities. Thus, we will create a new profile to obtain a part of the comparison result that interests us most (the **default** profile gives all the result of the comparison without any filtering). There are three types of filtering that are the "*Component Filter*", the "*Quick Filter*" and the "*Advanced Filter*". We will focus only on the first two categories in this use case.



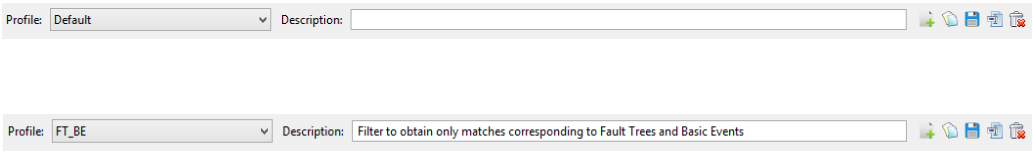
Element Type	Element Name	Match Type	Severity	Message
Basic Event	RIS002POBPR_DF	MODIFIED_SLIGHTLY	MINOR	Component modified slightly
Basic Event	ASG001BABPR_DF	MODIFIED	MAJOR	Component modified
Basic Event	ASG002POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	EAS001POMPE_DF	MODIFIED	MAJOR	Component modified
Basic Event	EAS001POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	EAS002POMPE_DF	MODIFIED	MAJOR	Component modified
Basic Event	EAS002POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	RCV001POMPE_DF	A_ONLY	MAJOR	Component does not exist in Model B
Basic Event	RCV001POMPE_DFF	B_ONLY	MAJOR	Component does not exist in Model A
CCF Group	C2EASPOMPE_DF_	MODIFIED	MAJOR	Component modified
CCF Group	C2RISPOMPE_DF_	MODIFIED	MAJOR	Component modified
CCF Group	C3RCVPOMPE_DF_	MODIFIED	MAJOR	Component modified
CCF Group	C3ASGPOMPE_DF_	A_ONLY	MAJOR	Component does not exist in Model B
CCF Group	C3ASGPOMPE_DF_2	B_ONLY	MAJOR	Component does not exist in Model A
Fault Tree	ASG	MODIFIED_SLIGHTLY	MINOR	Component modified slightly
Fault Tree	ASG_VOIE1	MODIFIED	MAJOR	Component modified
Fault Tree	ASG_VOIE2	MODIFIED	MAJOR	Component modified
Fault Tree	EAS_VOIE1	MODIFIED	MAJOR	Component modified
Fault Tree	EAS_VOIE2	MODIFIED	MAJOR	Component modified
Fault Tree	ISBP_VOIE1	MODIFIED	MAJOR	Component modified
Fault Tree	LHA	A_ONLY	MAJOR	Component does not exist in Model B
Parameter	######BABPR_DF	MODIFIED	MAJOR	Component modified
Parameter	C2POMPE_B2	MODIFIED	MAJOR	Component modified
Property	EAS_POMPE	A_ONLY	MAJOR	Component does not exist in Model B
Property	EASS_POMPE	B_ONLY	MAJOR	Component does not exist in Model A
System	LHB	MODIFIED_SLIGHTLY	MINOR	Component modified slightly

Figure 7: Result view for the fault tree example

We will create a new Profile to manage only basic events and fault trees, using the profile toolbox. We name the profile `FT_BE` and we will switch on it in order to configure the filters parameters.

- You have to click on the "create new profile" icon in the following toolbox in order to create the new profile. A dialog box opens where you have to give a name to the

created profile.



5.2.3 Filters Use

Let's configure the *component Filter* by clicking on "/configur/e" in front of Component Filter field. We Select "Fault tree" and "basic event". You can later try the use of other types of filtering related to the "Fault tree layer" or "other types" (see figure 8).

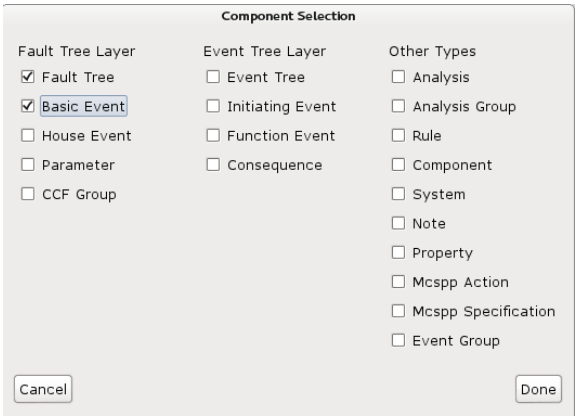


Figure 8: Component filtering selection List

In order to reduce the list of matches and focus on the important changes, we will use the "*Quick Filter*" capabilities and filter over two match types: the `Severity` and the `Match Type`. Edit the severity pattern by switching the dropdown menu on severity and write "MAJOR" in order to filter only on the major changes then switch to `Match Type` and write `MODIFIED, B_ONLY` in order to focus on modified components and components that figure only in the second model.

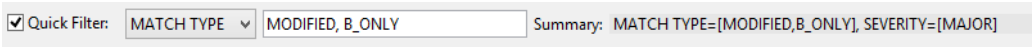


Figure 9 shows the list of the modifications that fit the filtering criteria which are 19 out of 26 of the modifications. You can see the different types of modifications in the detailed view or by right clicking on the match element.

Let's take for instance, the first Basic Event `ASG001BABPR_DF`. The detailed view indicates that the model attribute have been changed from "non-repairable" to "repairable".

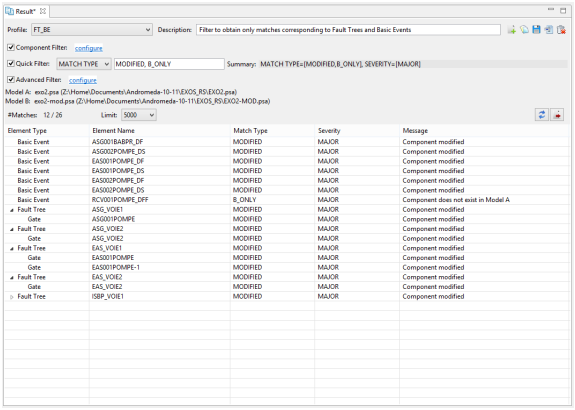


Figure 9: Filtering result for the fault tree example

Details

Detailed Difference	Modified Item	Severity	Description	Value A	Value B
ATTRIBUTE MODIFIED	model	MAJOR	Attribute 'model' changed from 'non-repair	non-repairable	repairable
ATTRIBUTE MODIFIED	modifiedBy	MINOR	Attribute 'modifiedBy' changed from 'AF' to	AF	TA
ATTRIBUTE MODIFIED	modified	MINOR	Attribute 'modified' changed from '2001-06	2001-06-28T20:	2015-11-10

For Basic Events `EAS001POMPE_DF`, `EAS001POMPE_DS` and `EAS002POMPE_DS`, we can see that List of `property reference` has been modified, but since we have filtered the results, the references components are no longer visible in the list.

#Matches: 12 / 26 Limit: 5000

Element Type	Element Name	Match Type	Severity	Message
Basic Event	ASG001BABPR_DF	MODIFIED	MAJOR	Component modified
Basic Event	ASG002POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	EAS001POMPE_DF	MODIFIED	MAJOR	Component modified
Basic Event	EAS001POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	EAS002POMPE_DF	MODIFIED	MAJOR	Component modified
Basic Event	EAS002POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	RCV001POMPE_DFF	B_ONLY	MAJOR	Component does not exist in Model A
Fault Tree	ASG_VOIE1	MODIFIED	MAJOR	Component modified
Gate	ASG001POMPE	MODIFIED	MAJOR	Component modified
Fault Tree	ASG_VOIE2	MODIFIED	MAJOR	Component modified
Gate	ASG_VOIE2	MODIFIED	MAJOR	Component modified
Fault Tree	EAS_VOIE1	MODIFIED	MAJOR	Component modified
Gate	EAS001POMPE	MODIFIED	MAJOR	Component modified
Gate	EAS001POMPE-1	MODIFIED	MAJOR	Component modified
Fault Tree	EAS_VOIE2	MODIFIED	MAJOR	Component modified
Gate	EAS_VOIE2	MODIFIED	MAJOR	Component modified
Fault Tree	ISBP_VOIE1	MODIFIED	MAJOR	Component modified

Search Details Console

Detailed Difference	Modified Item	Severity	Description	Value A	Value B
LIST MODIFIED	Property Reference	MAJOR	List of 'Property Reference' modified		

When we right click on the modified basic event `ASG002POMPE_DS`, we can choose to see the textual differences in red. We can see that the attribute `initiator` has been modified from `both` to `enabler-only`. We can also see by who and when the modifications have been made.

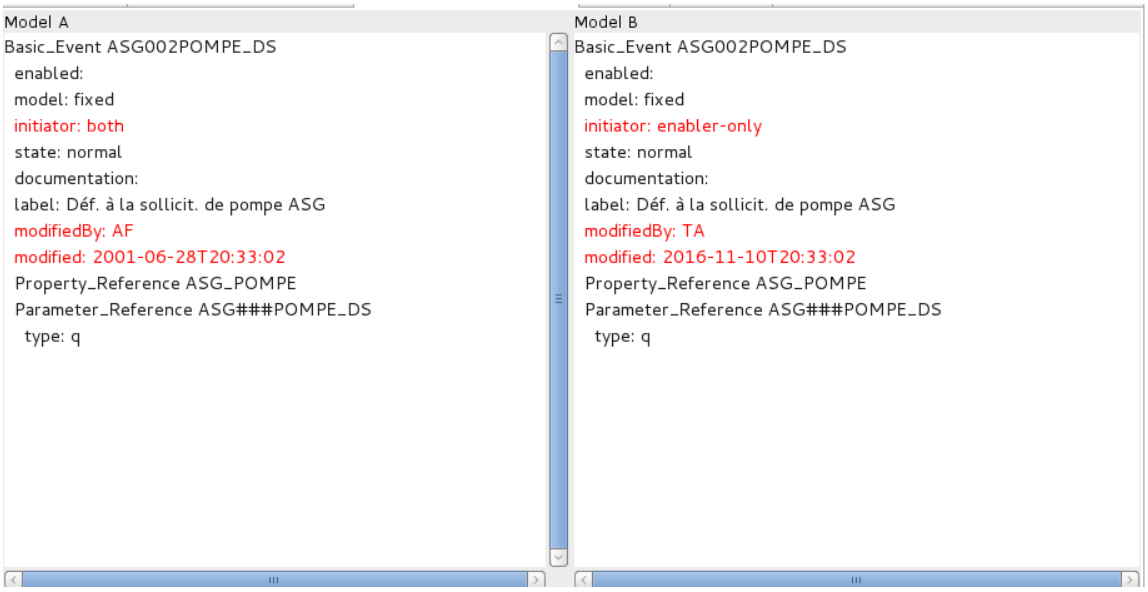


Figure 10: Textual comparison of fault trees

For fault trees, we can also see the sub matches that are not subject to filters, that is, gates modifications. We can explore the differences by looking at the detailed view or textual view but it is also possible to have a graphical view by right clicking on the item and selecting the "Compare Fault Tree diagrams". For example, for fault tree `ISBP_VOIE1`, we can see that modifications have been made in a gate reference. We should note that all the differences are not necessarily highlighted in this graphical view. Some modifications may remain hidden (for instance when a parameter relates to some basic events changes it is only highlighted in the paralmeter views and not within the fault-tree display to avoid duplicated information). But those related on the structure are explicitely highlighted. Since we have a lot of information that we want to explore, we will switch to the web view navigation.

5.2.4 Web export

In order to open the web view, you have to click on the export icon  from the result toolbox and choose "Web export" in order to view and explore the result in a web browser. In order to open the web export, open the file "index.html" that has been generated in the result folder you have given in order to store the result. In the main page, you can see that two types of components are proposed "Basic Events" and "Fault Trees" regarding

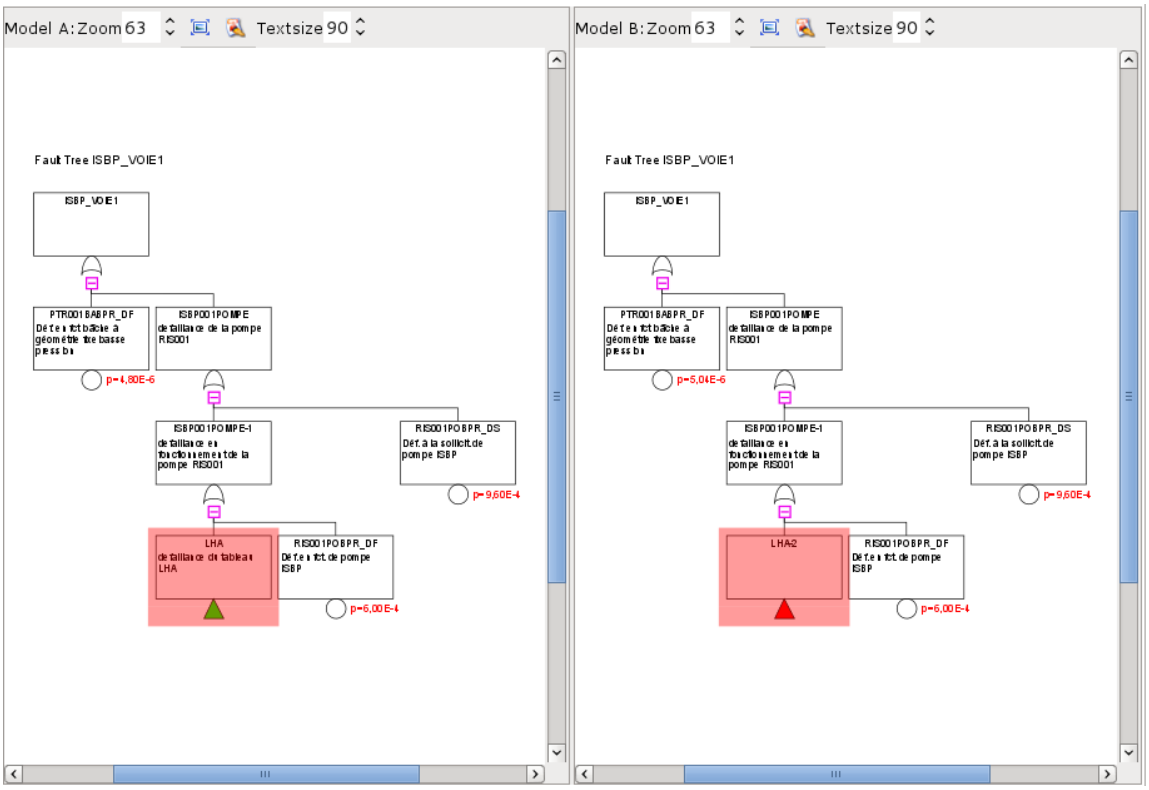


Figure 11: Graphical comparison of fault trees

the filtering configuration we have chosen (see Figure 12).

The fault tree differences are given in a table either in overview view or detailed view. In the detailed view, we can see all the detailed differences of fault trees and its sub matches as well as their detailed differences (see figure 13). Any detailed differences are colored red to increase readability. Sub matches are indicated by a leading # symbol before the Component Type (note the `#Gate` entries in the example). For example, in fault tree `ASG_VOIE2`, gate `ASG_VOIE2`, the gate type attribute has been modified from value "or" to "nor". In fault tree `EAS_VOIE1`, gate `EAS001POMPE` has its state and enabled attributes changed as well as its gcs list modified.

It is also possible to open textual comparison through this icon or graphical comparison through this icon . If we click on the graphical icon of fault tree `ISBP_VOIE1`, we can see the graphical differences of the two models (see Figure 14). We can notice that it corresponds to the same graphical comparison as figure 11. If we click on the textual icon of its subcomponent gate `ISBP001POMPE-1`, you can see the detailed textual information of the changes that have been made on the gate reference from value `LHA` to `LHA-2` (see Figure

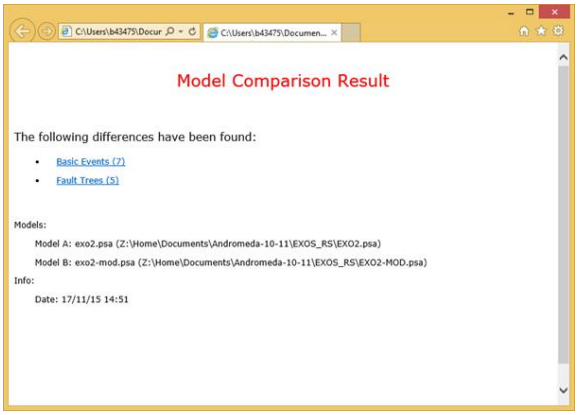


Figure 12: Web view start page

15).

Finally, a search field allows the filtering of matches by name and a top navigation bar provides the necessary links to switch to other tables types or to return back to the main page.

Remark. In the web interface the zoom applicability is managed by your browser. That is you may —depening of your browser— use the `Ctl - -` and `Ctl - +` for respectively zoom out and zoom in.

go back Basic Events (7) Fault Trees (5)

Fault Trees

Search: (5/5) *Overview* *Detailed View*

List of differences

Type	Name	Match Type	Severity	Comparison	Detailed Difference	Modified Item	Severity	Detailed Comparison	Value A	Value B
Fault Tree	ASG_VOIE1	MODIFIED	MAJOR							
# Gate	ASG001POMPE	MODIFIED	MAJOR		LIST MODIFIED	gcs	MAJOR			
Fault Tree	ASG_VOIE2	MODIFIED	MAJOR							
# Gate	ASG_VOIE2	MODIFIED	MAJOR		ATTRIBUTE MODIFIED	gatetype	MAJOR		OR	nor
Fault Tree	EAS_VOIE1	MODIFIED	MAJOR		ATTRIBUTE MODIFIED	modifiedBy	MINOR		AF	TA
					ATTRIBUTE MODIFIED	modified	MINOR		2001-06-29T19:55:49	2015-10-29T19:55:49
# Gate	EAS001POMPE	MODIFIED	MAJOR		ATTRIBUTE MODIFIED	enabled	MAJOR			true
					ATTRIBUTE MODIFIED	state	MAJOR		normal	true
					ATTRIBUTE MODIFIED	modifiedBy	MINOR		AF	TA
					ATTRIBUTE MODIFIED	modified	MINOR		2001-06-29T19:54:01	2015-11-10T19:54:01
					LIST MODIFIED	gcs	MAJOR			
# Gate	EAS001POMPE-1	MODIFIED	MAJOR		LIST MODIFIED	gcs	MAJOR			
Fault Tree	EAS_VOIE2	MODIFIED	MAJOR		ATTRIBUTE MODIFIED	modifiedBy	MINOR		AF	TA
					ATTRIBUTE MODIFIED	modified	MINOR		2001-06-29T19:58:41	2014-10-29T19:58:41
					ELEMENT MODIFIED	Top Gate	MAJOR			
# Gate	EAS_VOIE2	MODIFIED	MAJOR		ATTRIBUTE MODIFIED	gatetype	MAJOR		OR	and
Fault Tree	ISBP_VOIE1	MODIFIED	MAJOR							
# Gate	ISBP001POMPE-1	MODIFIED	MAJOR		LIST MODIFIED	gcs	MAJOR			

Figure 13: Web view Fault trees modifications

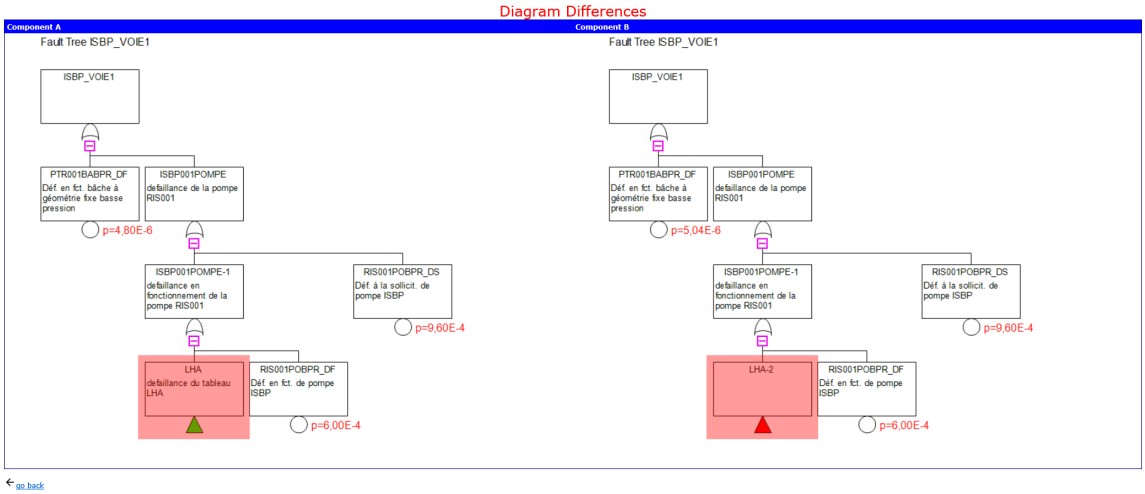


Figure 14: Graphical comparison of fault trees in the web view

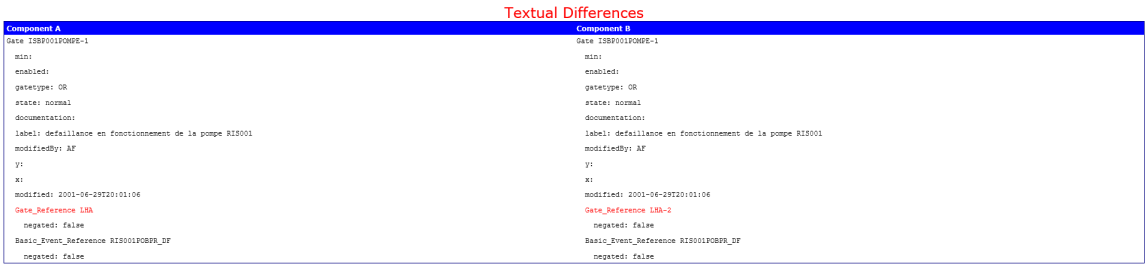


Figure 15: Textual comparison of fault trees in the web view

5.3 Comparing Event trees types

Event trees also encode Boolean formulas. Contrary to fault trees, which follow a deductive (top-down) concept, event trees focus on the evolution of events and thus follow an inductive (bottom-up) concept. Starting from a so-called *initiating event* (the first event to consider in an accident scenario), all consequences events are derived, recursively. A consequential event is an event that occurs due to the occurrence / non-occurrence of another event. Consequential events are called *functional events*. Each function event represents a system mission, human factor or an I&C system to mitigate a critical situation. The mission can be successful or failing. According to whether functions are successful or not, different evolutions (sequences) are deduced. The deduction of events ends is called *sequences*. Each sequence describes one specific event evolution. Sequences can lead to so-called *consequences* which describe a certain system state. Different sequences can lead to the same consequence. Also a sequence can lead to various consequences at a time.

In this exercise, we will focus on the comparison of event trees as well as the main components used for their construction, that is, initiating events, functional events, sequences and consequences. Initiating events and functions events can be linked to fault trees. Thus, in this example we will use the same fault trees as the previous example to construct the event tree.

5.3.1 Application Launch

Launch the comparison the comparison for `exo3.psa` and `exo3_mod.psa` that are provided in the **Exercices** folder joined to the tutorial.

You can navigate in the different view boxes in order to familiarize with the examples (see figure 16). In the model explorer view, you can notice a new folder named **Event-Trees**. This folder contains a single Event Tree called `BP`. When you click on the item, a graphical representation of the tree opens in order to get a graphical visualization. In the component view, you can find the main components of the event trees, that is the functional event references (e.g. `AAR`, `ISHP`, `FH_REF_2`, `ISBP`, `ASG` and `FH_GO` for model `exo3_mod`), the initiating event reference (e.g. `BP` for model `exo3_mod`) and the different sequences and their consequences (e.g. Sequence `BP_ASG_2` in model `exo3_mod` has two consequences references `CI_2` and `CI_H`). The attribute view allows to see the different attributes of each selected component.

Remark. In figure 16 you can notice that the selected sequence is colored red. In general when you select a sequence, it is highlighted and all the function events that are involved are colored red when there failure is expressed and green when they are successful. For consequences, a selection implies the highlighting of all the sequences that end in.

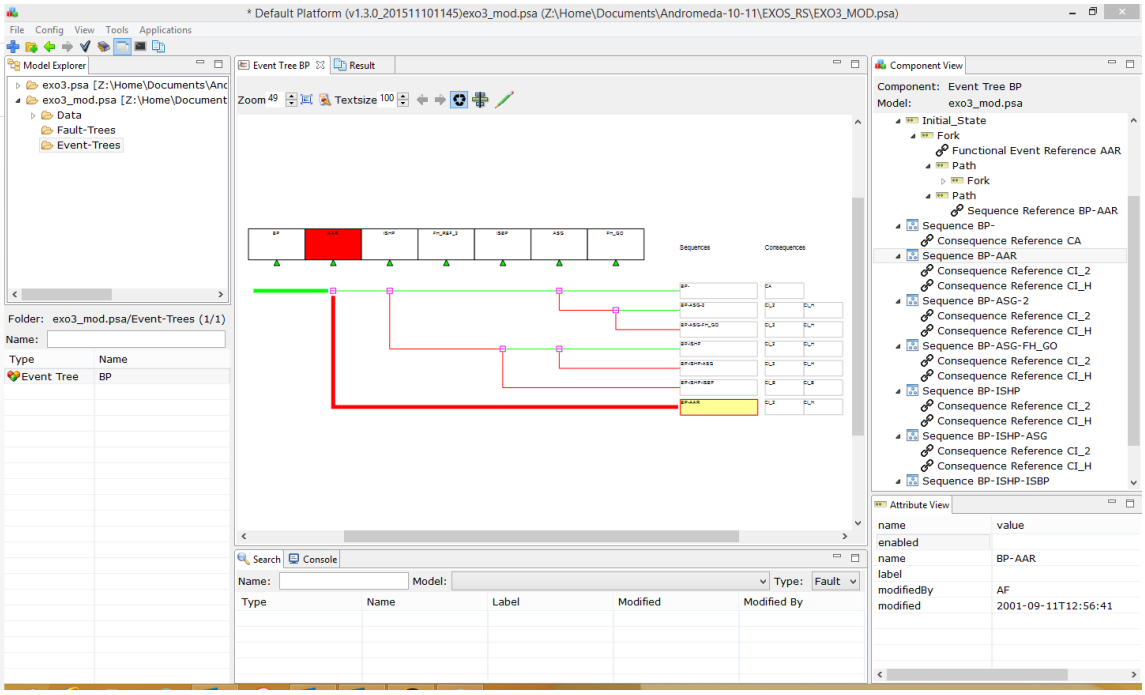


Figure 16: Andromeda view boxes for Event Tree components

5.3.2 Results Exploration

Figure 17 show the overview of the comparison result. Main differences involve changes in the consequences components such as consequences for `CI_H` that has been slightly-modified. Indeed minor modifications are introduced for attributes `modified_by` and `modified` corresponding to the user and date of modification.

Consequence `CI` is only declared in the first model whereas consequence `CI_2` is only declared in the second model.

Modifications have also been done for the sub-components of the event tree `BP` corresponding to sequences (see figure 18). For example sequence `BP-ISHP-ASG` has been modified, since its attributes have been changed. Attribute `enabled` has changed from no value to `true` and its consequence reference has also been changes from `CI` to `CI_2` if we look ore in detail in the textual comparison.

#Matches: 13 / 28 Limit: 5000

Element Type	Element Name	Match Type	Severity	Message
Basic Event	ASG001BABPR_DF	MODIFIED	MAJOR	Component modified
Basic Event	ASG002POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	EAS001POMPE_DF	MODIFIED	MAJOR	Component modified
Basic Event	EAS001POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	EAS002POMPE_DF	MODIFIED	MAJOR	Component modified
Basic Event	EAS002POMPE_DS	MODIFIED	MAJOR	Component modified
Basic Event	RCV001POMPE_DFF	B_ONLY	MAJOR	Component does not exist in Model A
Fault Tree	ASG_VOIE1	MODIFIED	MAJOR	Component modified
Gate	ASG001POMPE	MODIFIED	MAJOR	Component modified
Fault Tree	ASG_VOIE2	MODIFIED	MAJOR	Component modified
Gate	ASG_VOIE2	MODIFIED	MAJOR	Component modified
Fault Tree	ASG_VOIE3	MODIFIED	MAJOR	Component modified
Fault Tree	EAS_VOIE1	MODIFIED	MAJOR	Component modified
Gate	EAS001POMPE	MODIFIED	MAJOR	Component modified
Gate	EAS001POMPE-1	MODIFIED	MAJOR	Component modified
Fault Tree	EAS_VOIE2	MODIFIED	MAJOR	Component modified
Gate	EAS_VOIE2	MODIFIED	MAJOR	Component modified
Fault Tree	ISBP_VOIE1	MODIFIED	MAJOR	Component modified
Gate	ISBP001POMPE-1	MODIFIED	MAJOR	Component modified

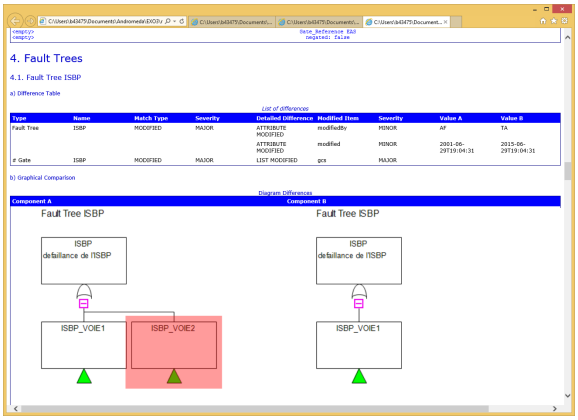
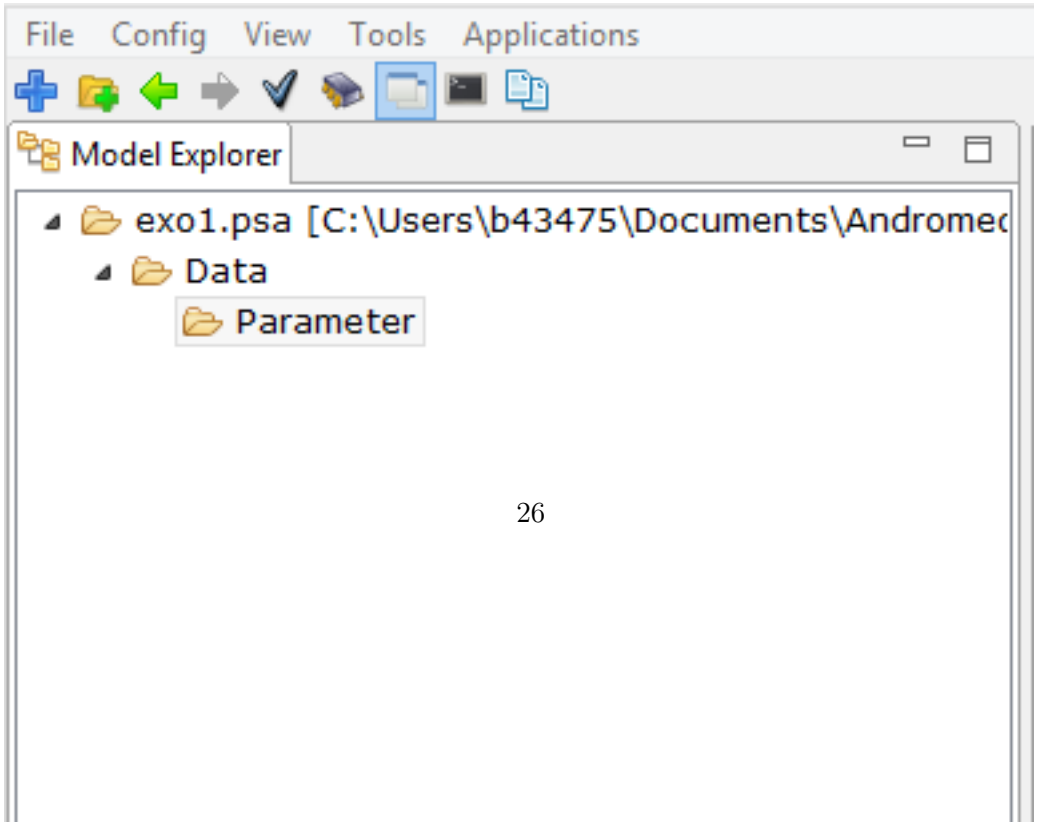


Figure 17: Event trees comparison result



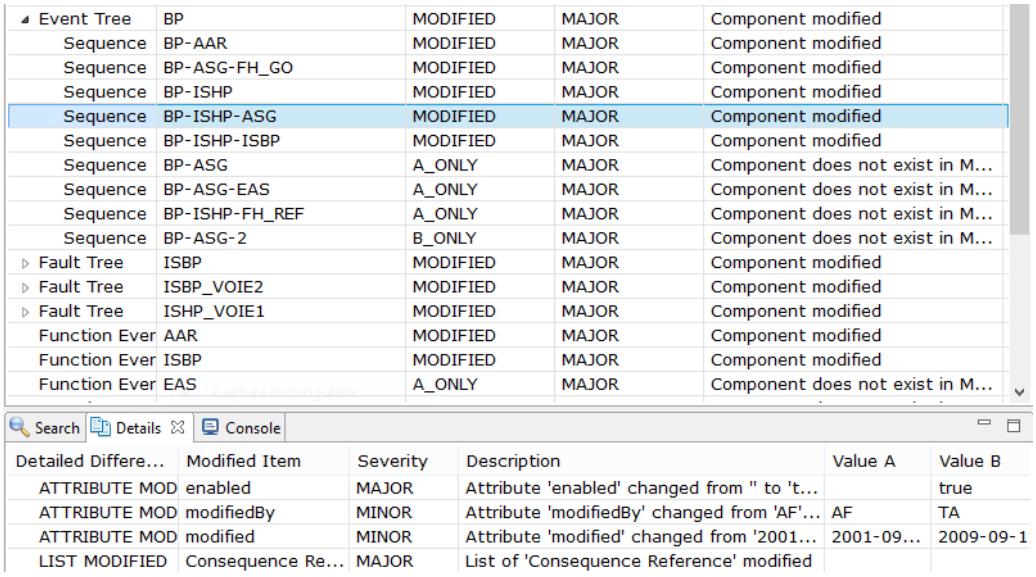


Figure 18: Example for sequence modifications

We can notice that function events have also been modified, such as function event "ARR" for which major modifications are raised in its attribute "enabled" that changed from no value in the first model to "false" in the second model. Attribute "success" has also changed from "logical" to "DeMorgan". In function event "ESBP", the attribute "FeAlternative" have been modified and finally Function Event "EAS" is only declared in the first model. We can notice changes in some fault trees that are linked to functional events but you can see section 1.2 for the result comparison of fault trees.

5.3.3 Advanced Filters

In this section, we will use the advanced filters in order to focus on specific type of changes. First, we are going to create a new profile `Event_tree` in order to save the filters configuration. Then we only check on the "Advanced Filter" box in order to focus on this feature. By appliying this filter on a component attribute, the other components are not impacted, you may notice that the other components remain in the interface. By clicking on "configure", a dialog box opens to begin the configuration (Figure 19). There are two types of filters that we can apply on components in order to specify a particular type of criteria concerning their attributes.

- The Single Type Filter is associated to a specific (and only one) component type.
- The Multi Type Filter is associated to a set of component types.

The difference between the two types of filters is that single type filters can involve specific expressions regarding the selected component (linked mainly to its specific attributes) whereas multi type filters are limited to common properties of the selected components.

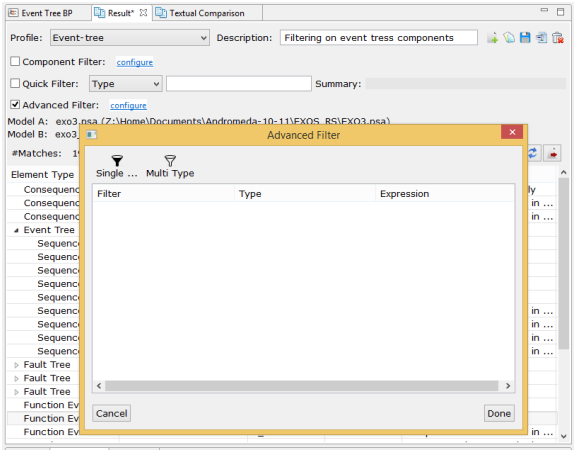


Figure 19: Dialog box for the advanced filter configuration

1. Single Type Filter

   Let's make a new single type Filter over a Function Event type. When you click on the single Type item, a dialog box opens and therefore, you can select only one component.

   Check on Function Event from the Event Tree Layer components list (see figure 20).

   Once you are back to the advanced filter dialog box, you can see that a new single type Filter have been added for Function Event type. Right click on the Single Type Filter. You can either:

   - Edit the name
   - Append a criterion over the date of modification or over an attribute
   - Append an operator in order to define complex expressions. You can choose OR, AND, NOT.
   - Remove the filter.

   We want to filter the Function events that have been modified after 01/01/2010 and for which the attribute "enabled" has been set to false.

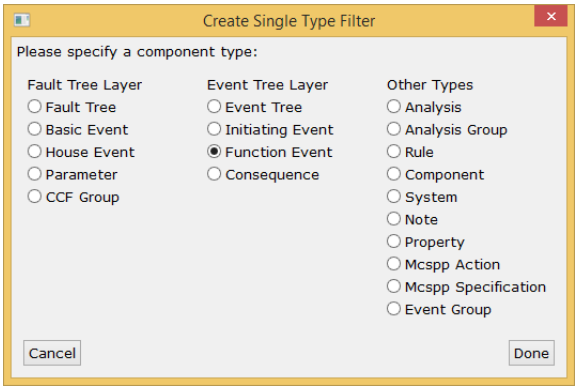
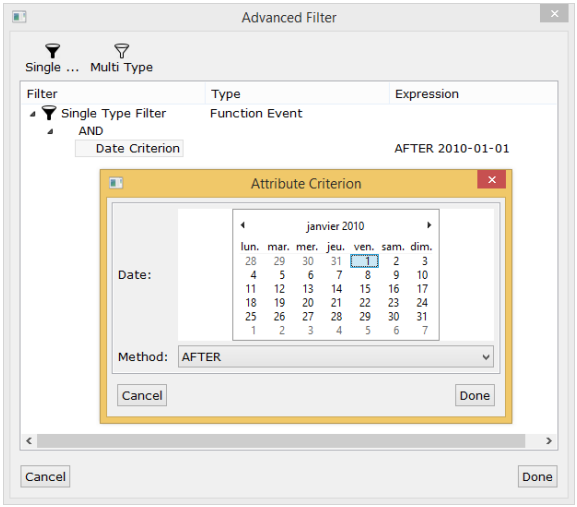


Figure 20: Single Type Filter Components list

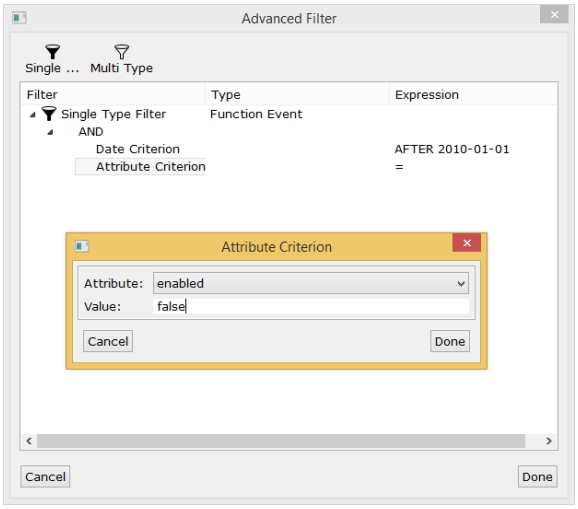
(a) Right click on the single Type Filter and select "Append Operator" then "AND".

(b) Right click on the "AND" operator and select "Append Criterion", then "Date criterion". Each time you want to change a criterion, you have to right click on it.



(a) You can choose the date via a calendar and the comparison method (`BEFORE`, `BEFORE_EQUAL`, `EQUAL`, `AFTER`, `AFTER_EQUAL`) by selecting `AFTER` among the proposed list.

(b) Right click again on the `AND` operator and select "Append Criterion", then "Attribute criterion".



(a) You can choose the attribute "enabled" among the list of proposed attributes (`documentation`, `enabled`, `label`, `modified`, `modified_by`, `name` and `success`) and set the value to False.

2. Multi Type Filter

   Let's make a new multi type Filter over the event tree, the initiating event, the consequences and properties. Check on those components via the dialog box **(see Figure 21)**.

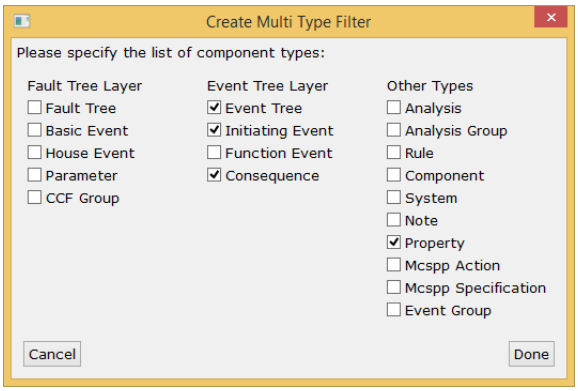


Figure 21: Multi Type Filter Components list

Once you are back to the advanced filter dialog box, you can see that a new Multi type Filter have been added for the selected components. Right click on the Multi

Type Filter and you will notice that you have the same operators as the single type filter:

- Edit the name
- Append a criterion over the date of modification or over an attribute
- Append an operator in order to define complex expressions. You can choose OR, AND, NOT.
- Remove the filter.

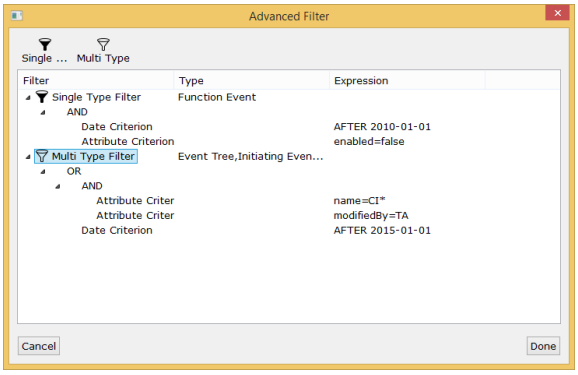
Nevertheless, concerning the attributes criteria, you can notice that only allowed criteria over attributes `label`, `modified_by`, `modified` and `name` which are the common attributes of the selected components.

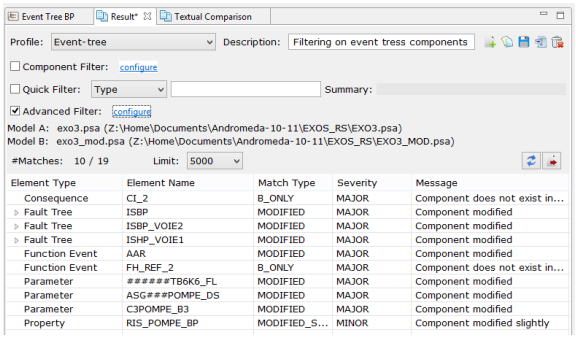
We want to filter components that either have been modified after 01/01/2015 or that their name contains "CI" and have been modified by user "TA".

(a) First add the `OR` operator.

(b) Then add a new operator `AND` in order to create the attributes criteria's over the attributes name = `CI*` and `modified_by` = `TA`.

(c) Add a new attribute criteria for the "OR" operator and select the date via the calendar and choose the comparison method AFTER.





You can notice in the comparison result that we have filtered 10 over 19 matches that correspond to our filters.

5.3.4 Review Export

The review export aims to create a single document that protocols the modifications. It can be used to assist engineers in analyzing the results or to generate reports that can be easily shared. In this section, we will use the default profile, in order to generate all types of components.

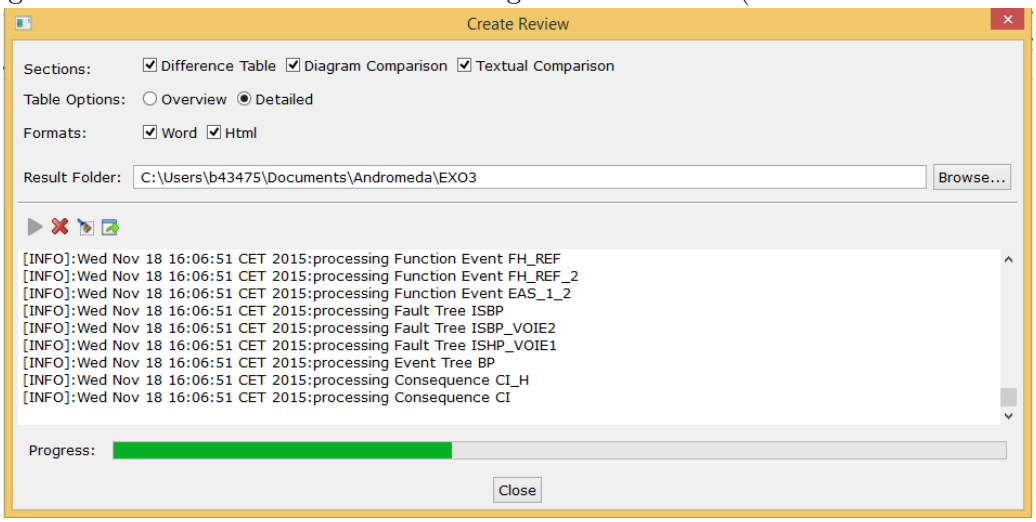
Launch the review export function via the export icon in the result toolbox and select "create review". You can choose your own export options:

- *Difference Table* in order to generate tables that contain the matched components.

You can also find Table options in order to precise the level of detail you need to generate. This case is ignored if there are no tables to produce.

- The overview exports only main characteristics of matches (match name, match type, severity . . .)
- The detailed exports matches and sub matches and their respective detailed differences.
- *Diagram Comparison* in order to export the graphical comparisons for fault trees and event trees.
- *Textual comparison* in order to export textual differences of the matched components.
- *Formats* corresponds to the output format of the produced documents, that is, either a word document or an HTML format which can be visualized in a web navigator.

In the example, we choose to generate the difference tables, diagram and textual comparison with its detailed options in a document and HTML format. Click on to launch the generation and click on close when the generation is over (it is mentioned in the log).



The main chapters of the document and the HTML format contain the different component types for which matches have been exported. Chapter 1 is for parameters, chapter 2 is for initiating events and so on. Below each chapter, the corresponding matches are listed one by one with a respective section of each chapter. For example, chapter 3 concerns functional events, section 3.1. describes the functional event "AAR" and section 3.2. describes the functional event "ISBP" etc. For each section, the following sections are produced:

- Difference Table: Table listing the match, its sub matches and detailed differences.
- Text comparison: Textual comparison of the respective match.
- Graphical comparison: Graphical comparison of the respective match (for fault trees and event trees only).

Figure 22 shows an extract of a review in word format.
Figure 23 shows an extract of a review in HTML format.

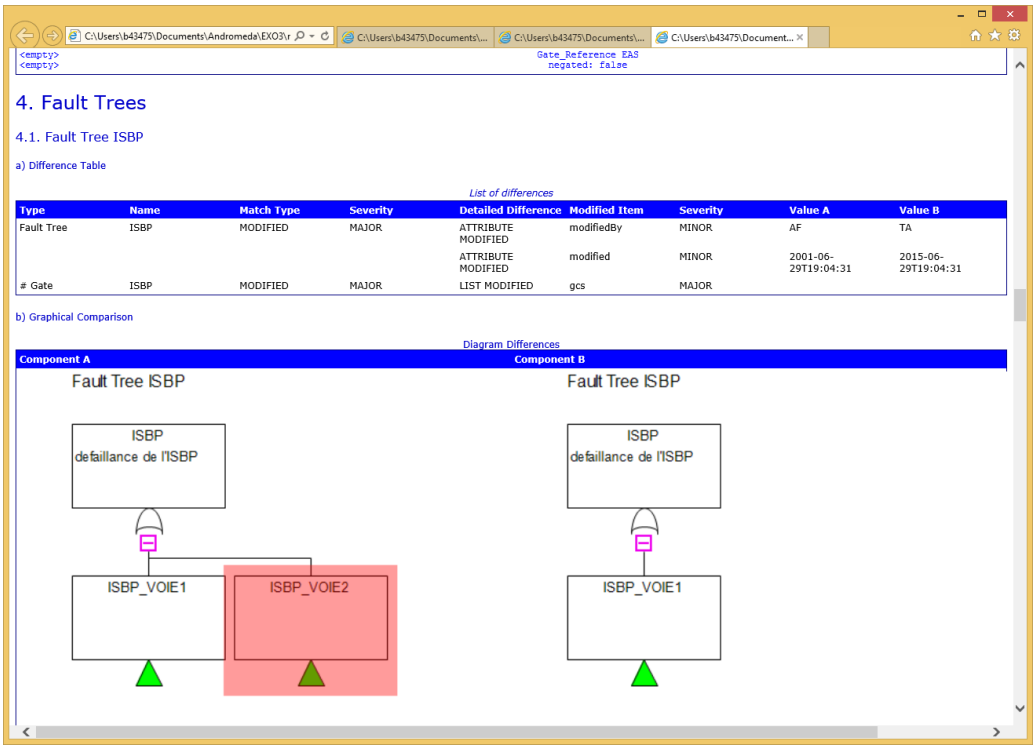


Figure 22: Extract of a review in word format

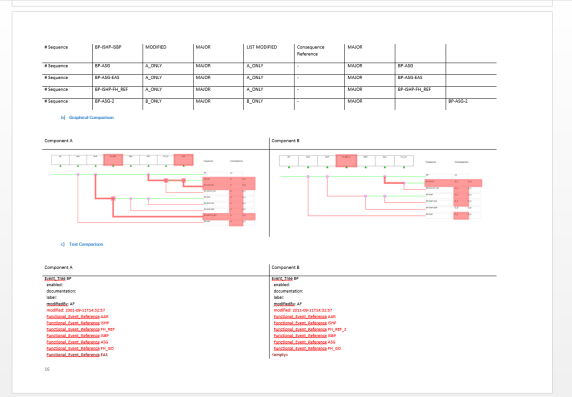


Figure 23: Extract of the review in HTML Format

References

[1] Thomas Friedlhuber. User guide andromeda model comparison v1.3. Technical report, EdgeMind, 2015.