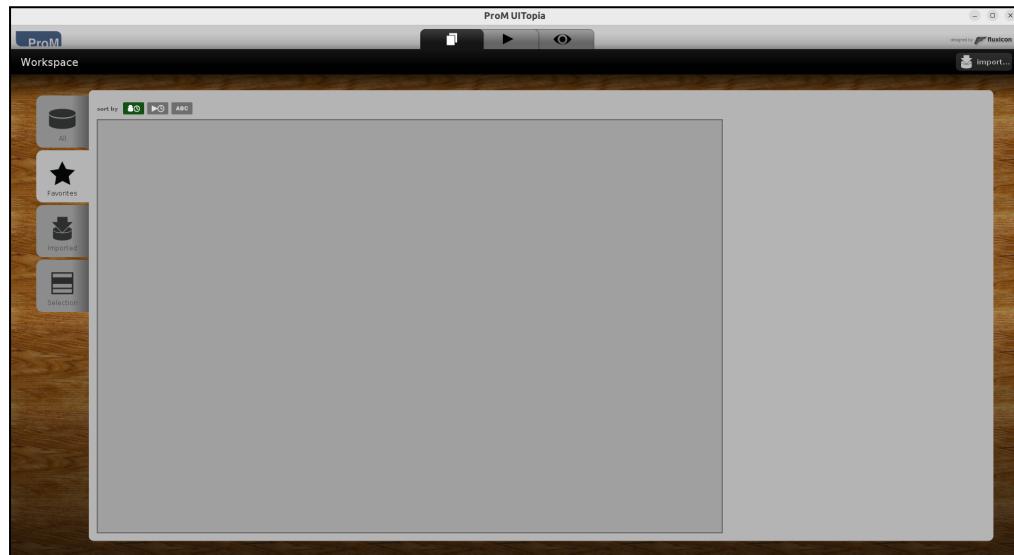


User Guide

Here is a user guide with an example using the process model for a cardiologist:

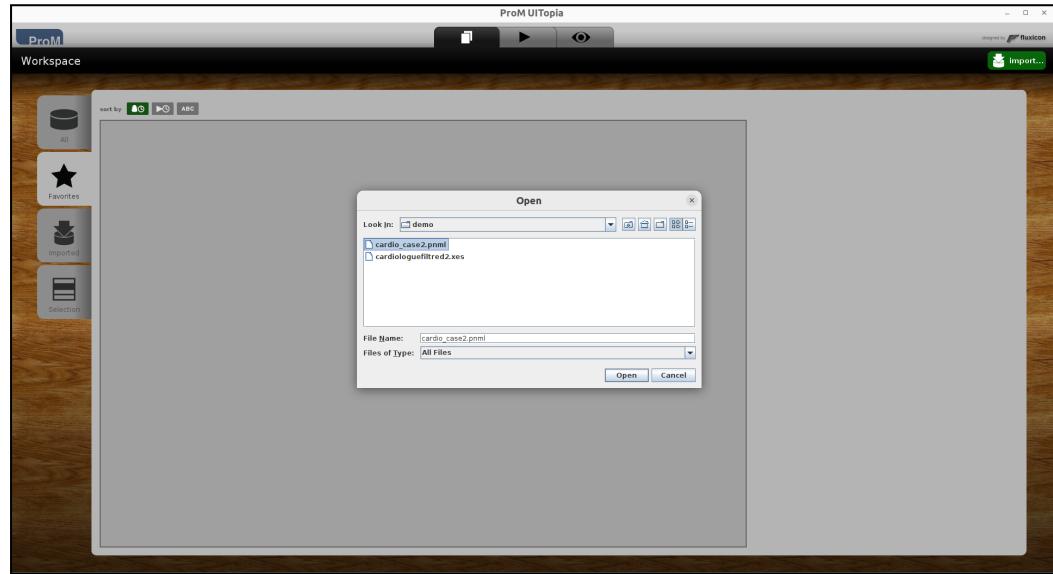
1. The plugin takes two inputs, a LOG and a PETRINET. These two can be introduced in 2 different ways:

- a. **Variant I :** when the Petri net and the log are already provided in ProM, for example : if they are results of other plugins (e.g., filtering plugins), or if they are imported directly from ProM like this:



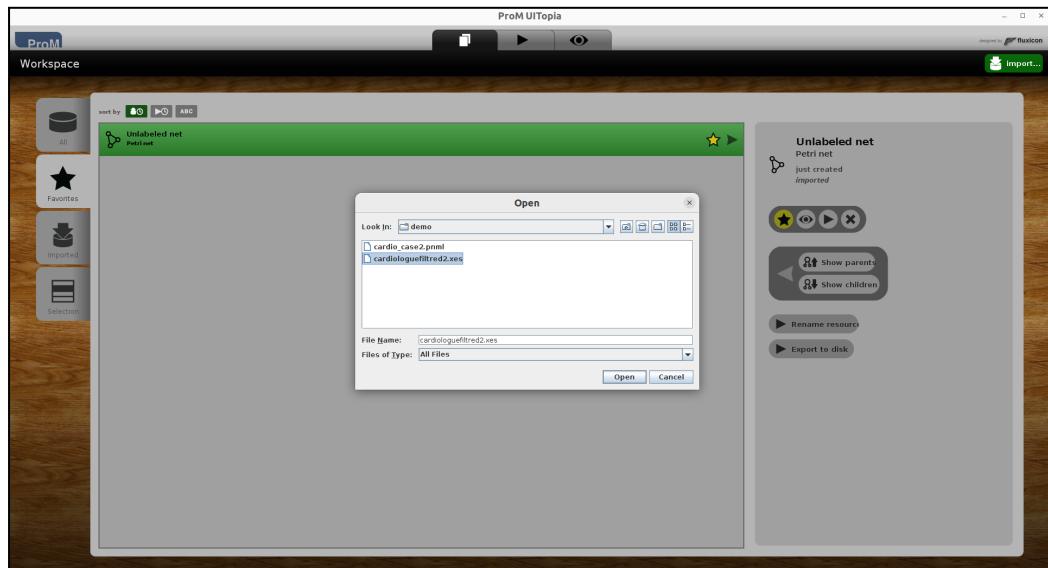
Interface (Variant I) - Button Import

- Click on "*Import*".



Interface (Variant I) - Importing the model

- Import the process model (.pnml extension).



Interface (Variant I) - Importing the event log

- Import the event log (.xes extension).



Interface (Variant I) - Plug-in creation

- Click on "►" and type the name of the plug-in: *Interoperability Solver*.

b. Variant II: the plug-in will take care of importing the files.



Interface (Variant II) - Plug-in creation

- Click directly on "►" and type the name of the plug-in: *Interoperability Solver*.



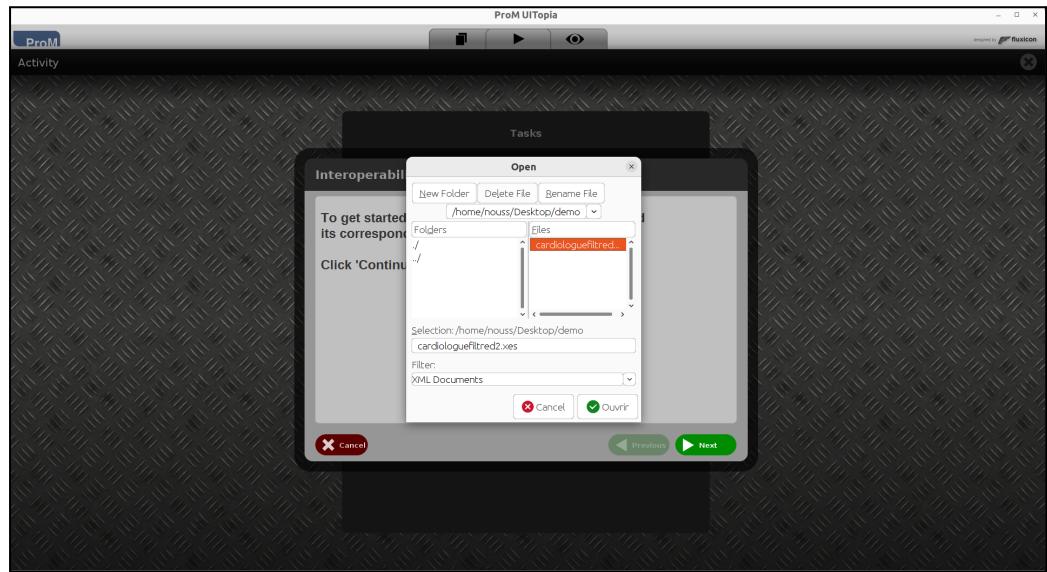
Interface (Variant II) - Variant selection

- Choose the variant in *green*.



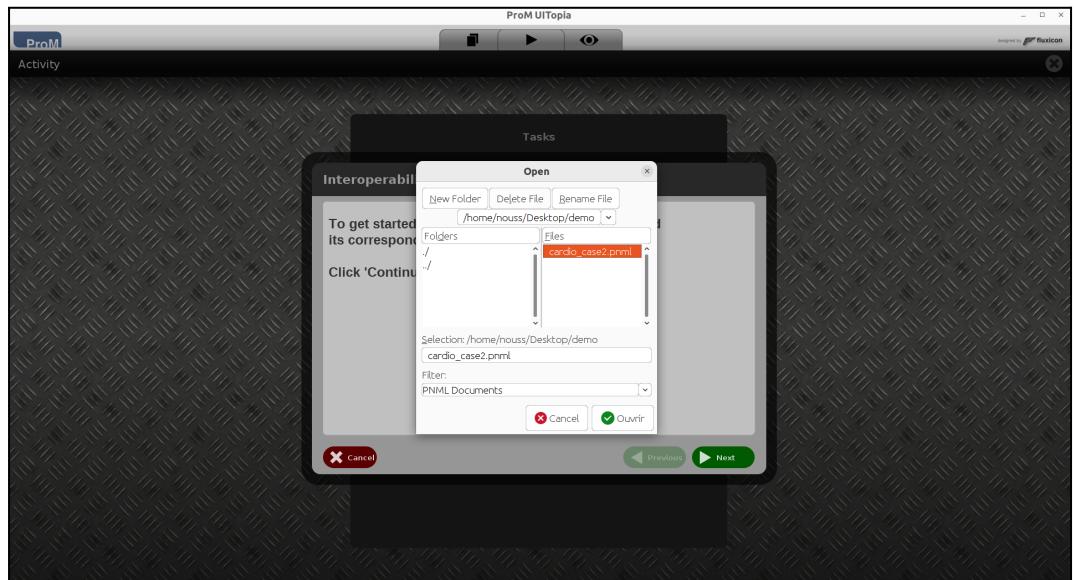
Interface (Variant II) - Button Next to continue

- Click on "Next".



Interface (Variant II) - Importing the event log

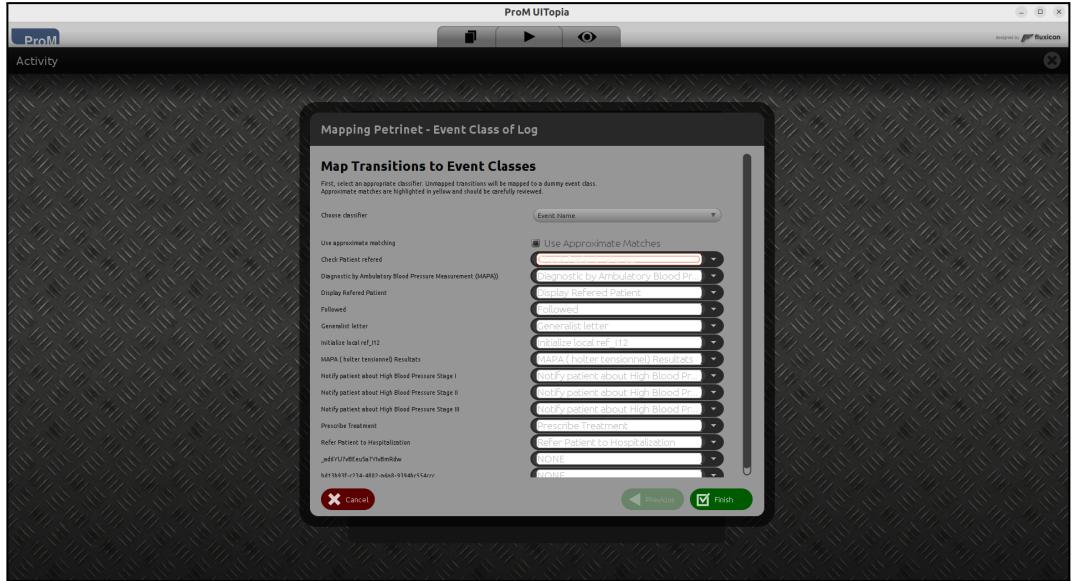
- Import the event log (.xes extension).



Interface (Variant II) - Importing the process model

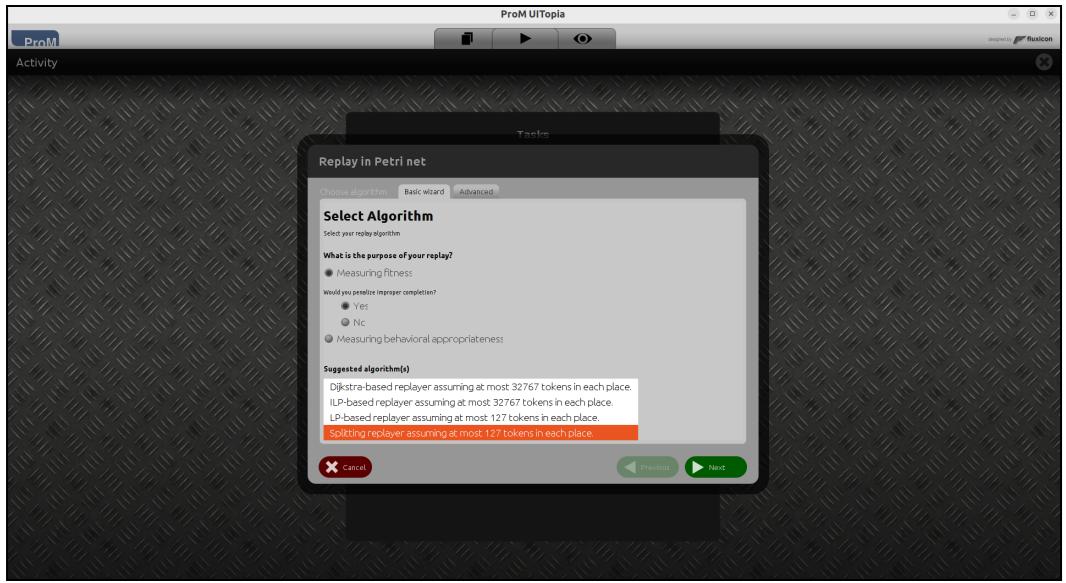
- Import the process model (.pnml extension).

2. Given the 2 inputs, the plug-in starts by doing the alignment, while giving the user the possibility to customize the different parameters.



Interface (Variant II) - Check/modify the mapping

- Check/modify the mapping between the *Event Class* of the log and the *Transitions* of the model (Here we have classified the events according to their names).



Interface (Variant II) - Selecting the Replay algorithm

- Select the desired *replay* algorithm (here we have chosen the algorithm *Splitting Replayer* which measures the fit between each trace and the model).



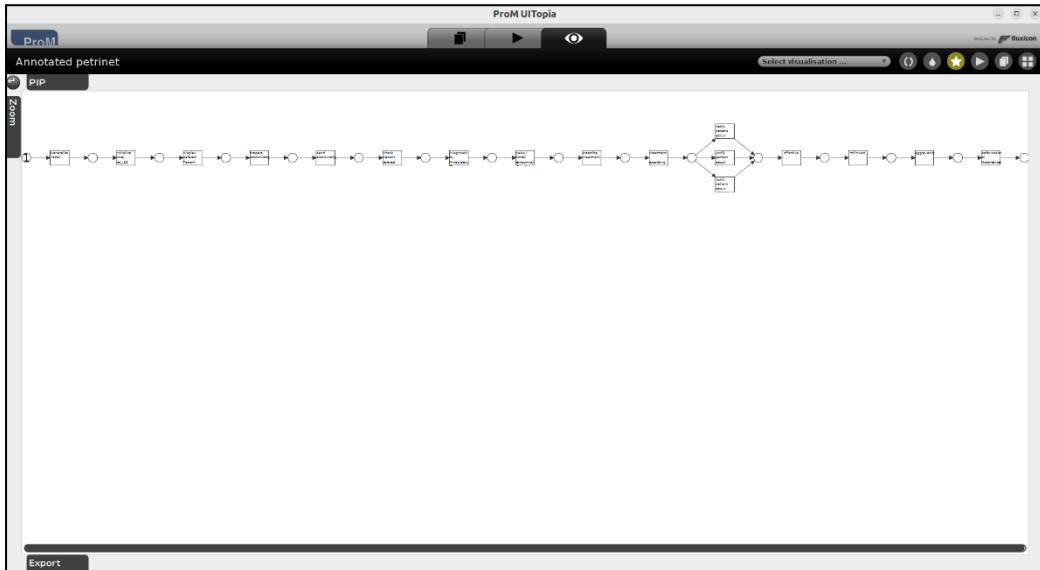
Interface (Variant II) - Customize the cost of each deviation

- Customize the cost of each deviation between the traces and the model
 (Customized Cost Function): Here, we have not emphasized one deviation over another (all costs are set to 1, except for *Move synchronous* which are set to 0 since they do not have deviations).

PS:

- **Move on log**: it corresponds to an activity in the log that is not in the model. This type of *move* is inappropriate and indicates a deviation between the trace and the model.
- **Move on Model**: it corresponds to an activity in the model that is not in the trace, if it does not imply an invisible transition, it signals a deviation between the trace and the model.
- **Move synchronous**: the classification of the event corresponds to the transition, in this case the trace and the model advance in the same way during the replay.

3. The model will then be repaired and annotated:



As we have not implemented a visualization module, we will visualize the different labels on the console as follows:

```
eclipse-workspace - GettingStarted/src/org/processmining/plugins/interoperability/InteroperabilitySolver.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Annotation.java SolverDialog.java
Console x Problems Debug Shell
ProM with UI Topia (GettingStarted) [Java Application] /home/nouss/Downloads/ire1.8.0_341/bin/java [Oct 6, 2022, 8:37:52 PM] [pid: 19576]
83 * @UITopiaVariant
84 {
85     affiliation = "CDTA"
86     author = "Bachiri I"
87     email = "ji_bachiri"
88     )
89 }
90 @PluginVariant
91     variantLabel = "Int"
92     requiredParameterLa
93     requiredParameterLa
94 )
95 public LabelledPetrinet sol
96 XLog log = null;
97 Petrinet net = null;
98 SolverDialog dialog = n
99 InteractionResult result
100 if (result != Interacti
101     return null;
102     }
103     return solveGiven(conte
104 }
105 /**
106 * The actual work.
107 */
108 Diagnostic by Ambulatory Blood Pressure Measurement (MAPA):
109 private LabelledPetrinet so
110     // Alignment
111     PNRepResult aligned = A
112     // Repair
113     Petrinet repairedNet =
114     // Annotation
115     LabelledPetrinet result
116     return result;
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }

CodeMix
```

Viewing the different labels