Replay a Log on Petri Net for Performance/Conformance Plug-in

Package: PNetReplayer, PNetAlignmentAnalysis

Arya Adriansyah

Last updated on 8 October 2012 (draft version)

Contents

1. Introduction

This plug-in accept a Petri Net/Inhibitor/Reset/ResetInhibitor Net and an event log to create advanced alignments between each trace in the log and the net. Note that the concept of advanced alignments is derived from [1].

2. Requirements

To get the most benefit of this plugin, these requirements need to be satisfied:

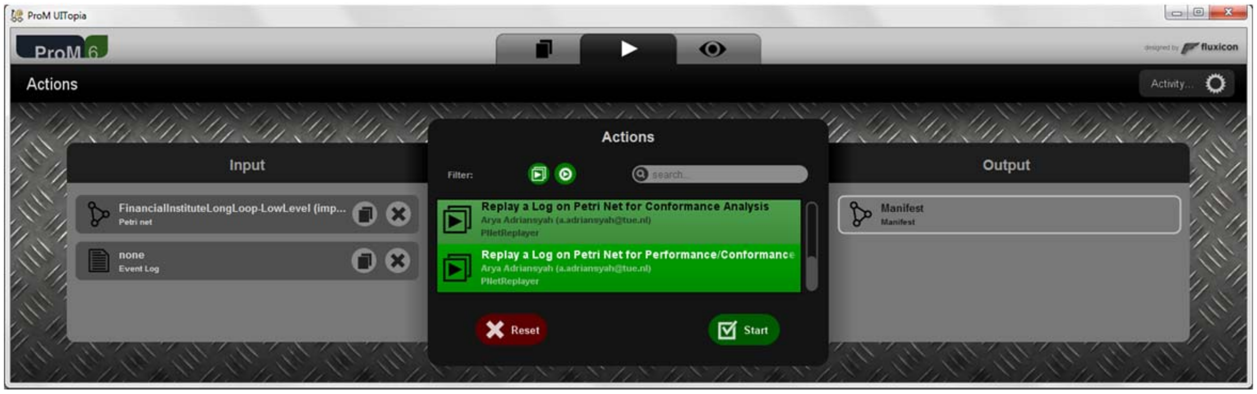
1. The net:
    a. Has a non-empty initial marking, and
    b. Has a final marking. Without any final marking, it is assumed that all deadlock states/reachable states in the net are final markings.
2. The log has more than 0 (zero) number of traces

3. Known Issue (!)

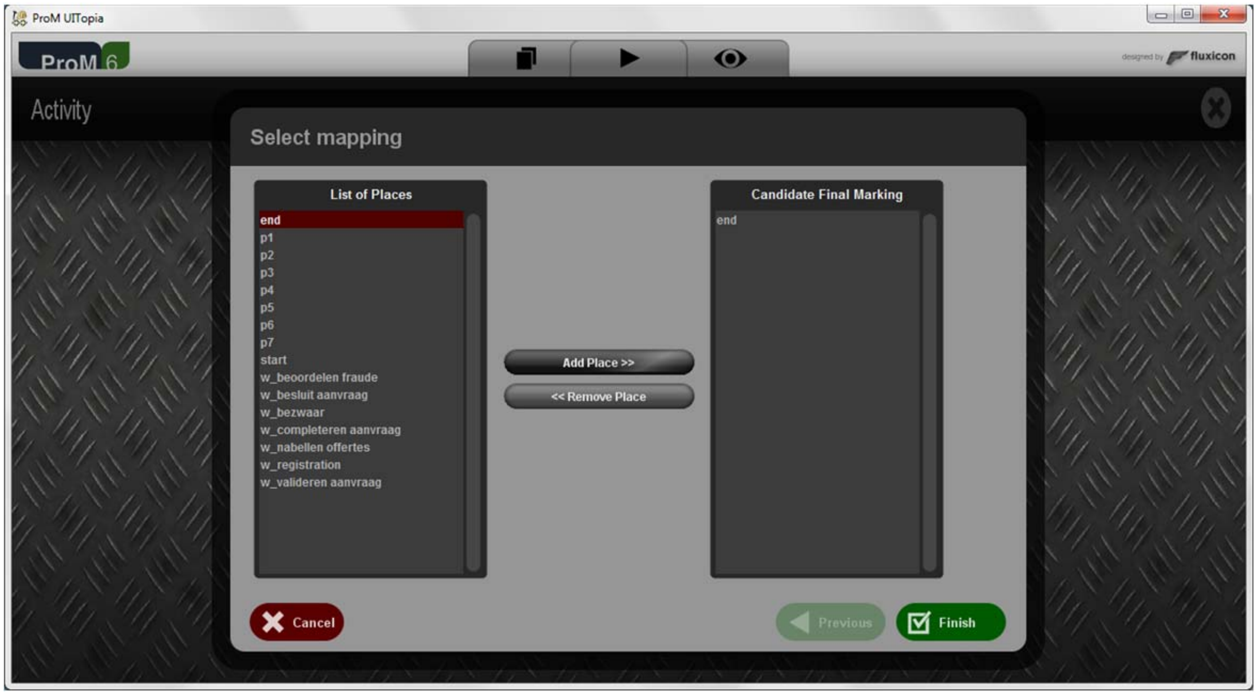
ProM 6 “PNetReplayer” package is derived from the **obsolete**-and-will-be-removed “Replayer” package. Existence of both packages within one ProM installation may result in inconsistency problem. Hence, please remove the “Replayer” package using the ProM Package Manager and ensure that the “Replayer” package has been removed from .ProM folder (located in C:\Users\<user name>\.ProM\packages in Windows OS) before you use this plug-in.

4. Example use of the plug-in

The plug-in requires a Petri Net/Inhibitor/Reset/ResetInhibitor Net and an event log. Therefore, we first need to load both of them (see figure below). Click the “Start” button


ProM UITopia
ProM 6
designed by fluxicon
Actions
Activity...
Input
Actions
Filter:
search...
Output
FinancialInstituteLongLoop-LowLevel (imp...
Petri net
none
Event Log
Replay a Log on Petri Net for Conformance Analysis
Arya Adriansyah (a.adriansyah@tue.nl)
PNetReplayer
Replay a Log on Petri Net for Performance/Conformance
Arya Adriansyah (a.adriansyah@tue.nl)
PNetReplayer
Manifest
Manifest
Reset
Start

If the net does not have any final marking, (i.e. no FinalMarkingConnection object that connects the net to a marking object), a popup panel that asks whether you want to create one is shown. Answering yes to the question triggers the “Create Final Marking” plugin. This plug-in can also be executed separately from the “Replay a Log on Petri Net for Performance/Conformance Analysis” plug-in. Answering no to the question will cancel the plugin.

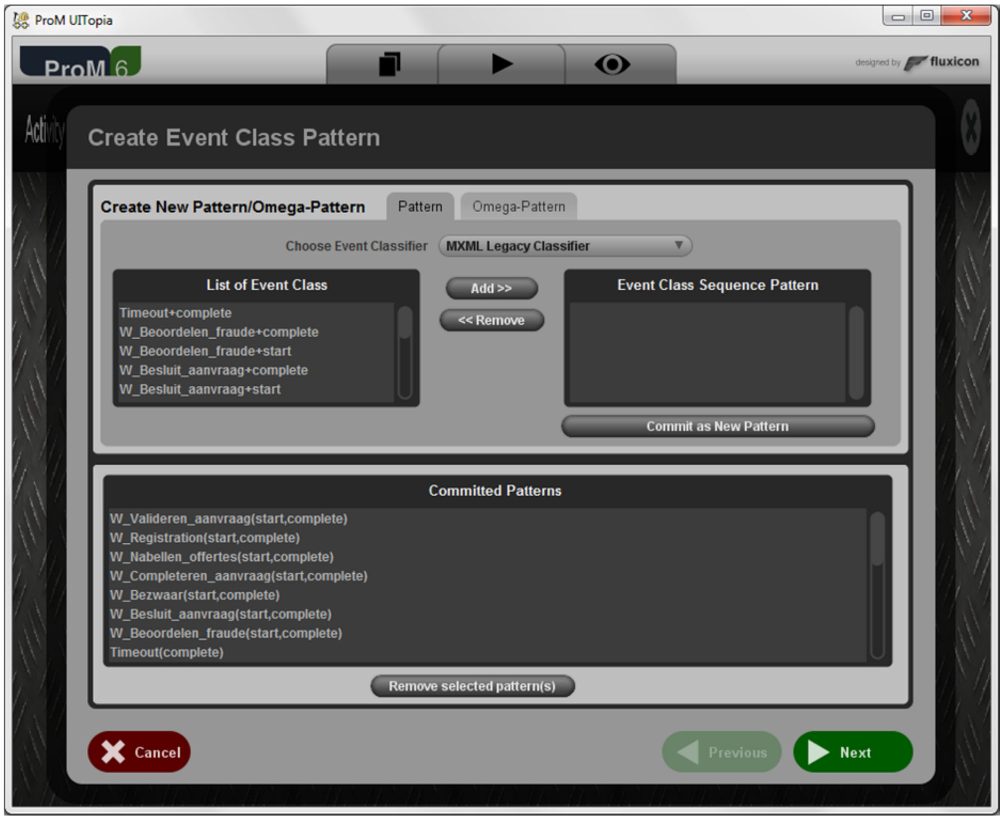


List of Places:
end
p1
p2
p3
p4
p5
p6
p7
start
w_beoordelen fraude
w_besluit aanvraag
w_bezwaar
w_completeren aanvraag
w_nabellen offertes
w_registration
w_valideren aanvraag

Candidate Final Marking:
end

After finish creating a final marking, click “Finish”, you’ll be asked to re-run the plug-in.

A panel to create event class pattern will be shown (see below). Choose an appropriate classifier and create pattern of event class that transitions in the net may have in the log. For example, execution of a transition “W_Registration” in the net may be recoreded as two events in the log: W_Registration+start and W_Registration+complete. In this case, use this panel to create a pattern W_Registration+start, followed by W_Registration+complete, then commit it. Only committed patterns can be mapped to transitions in the next steps.



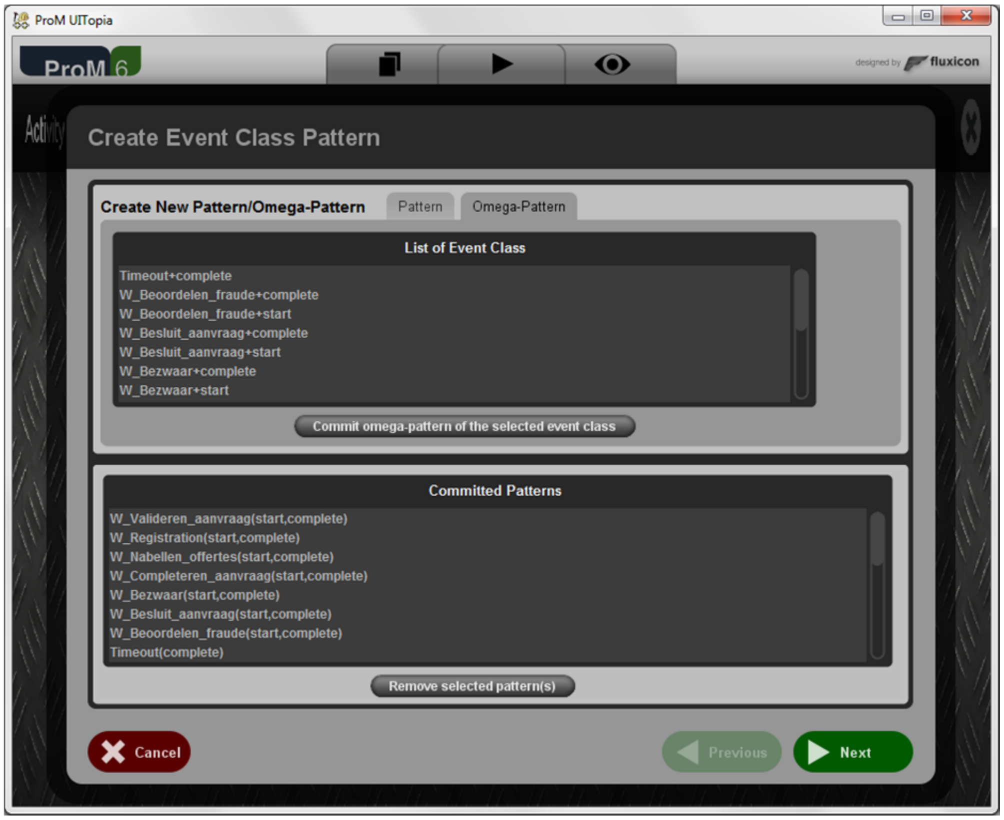
Choose Event Classifier: MXML Legacy Classifier

List of Event Class
Timeout+complete
W_Beoordelen_fraude+complete
W_Beoordelen_fraude+start
W_Besluit_aanvraag+complete
W_Besluit_aanvraag+start

Event Class Sequence Pattern

Committed Patterns
W_Valideren_aanvraag(start,complete)
W_Registration(start,complete)
W_Nabellen_offertes(start,complete)
W_Completeren_aanvraag(start,complete)
W_Bezwaar(start,complete)
W_Besluit_aanvraag(start,complete)
W_Beoordelen_fraude(start,complete)
Timeout(complete)

There is also a special pattern that called “Omega-Pattern”. This pattern is true if mapped to any other than the ones mentioned in the pattern. For example, an omega pattern of activity “Timeout+complete” represents all activities other than “Timeout+complete”.


ProM UITopia
ProM 6
designed by fluxicon
Create Event Class Pattern
Create New Pattern/Omega-Pattern
Pattern
Omega-Pattern
List of Event Class

Timeout+complete
W_Beoordelen_fraude+complete
W_Beoordelen_fraude+start
W_Besluit_aanvraag+complete
W_Besluit_aanvraag+start
W_Bezwaar+complete
W_Bezwaar+start

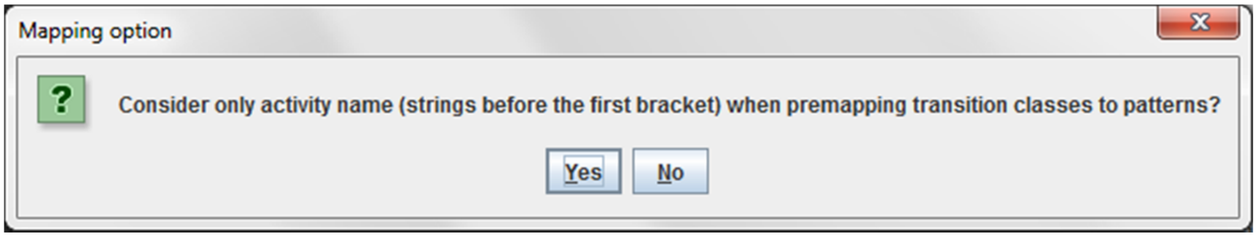
Commit omega-pattern of the selected event class
Committed Patterns

W_Valideren_aanvraag(start,complete)
W_Registration(start,complete)
W_Nabellen_offertes(start,complete)
W_Completeren_aanvraag(start,complete)
W_Bezwaar(start,complete)
W_Besluit_aanvraag(start,complete)
W_Beoordelen_fraude(start,complete)
Timeout(complete)

Click “Next” to continue.

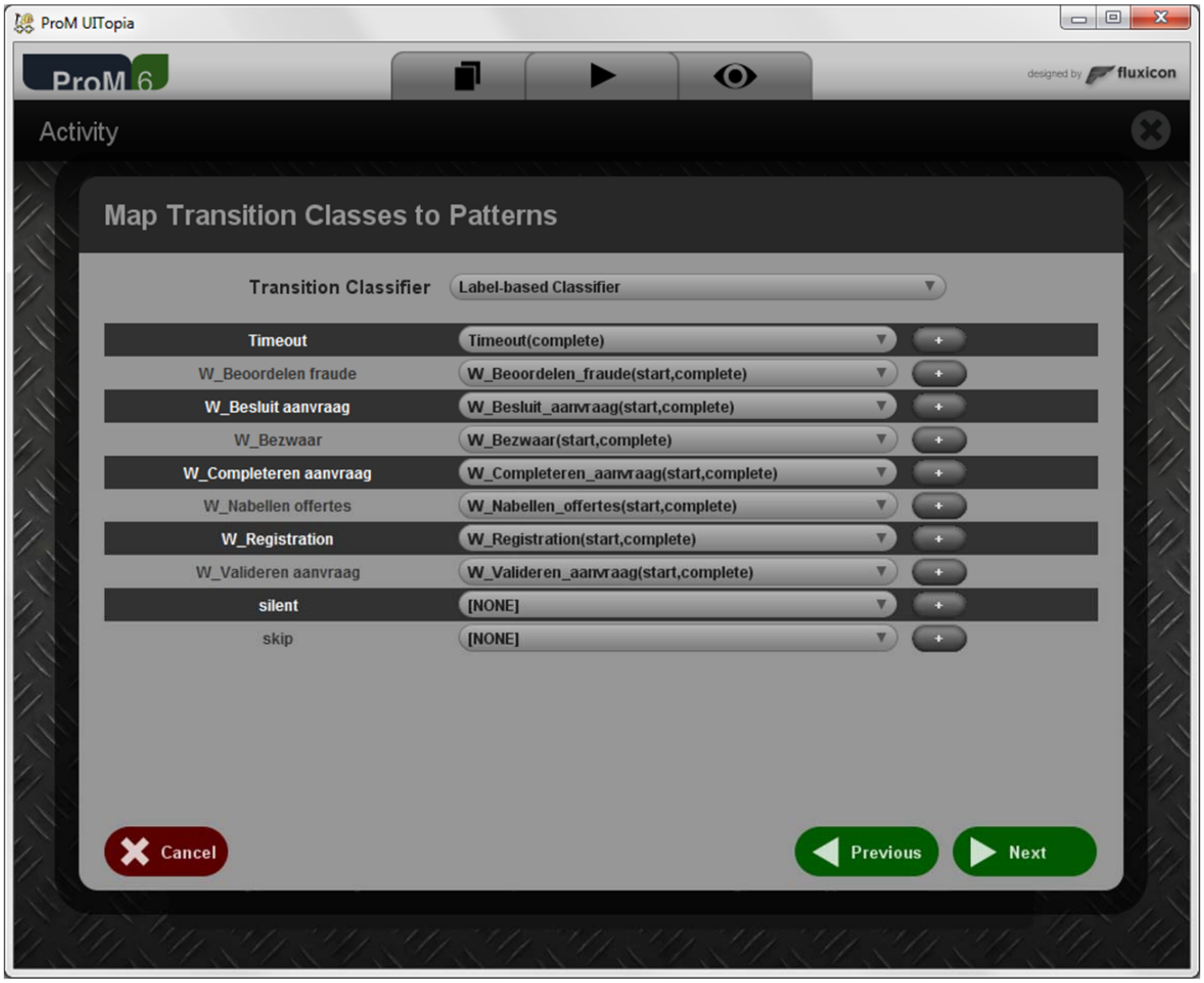
Note: The mapping panel automatically identified patterns based on standard lifecycle of activities, i.e. Start-Complete and Schedule-Start-Complete. For example, if the log records the start and complete of activity W_Registration, a pattern W_Registration(start,complete) will be identified and shown in the set of committed patterns.

A popup will be shown, asking whether the mapping suggestions between transitions and event class patterns should be based on only event class name before brackets, or the whole label. If the execution of each transition is recorded as only one event in the log, it’s suggested to choose “No”. If the execution of each transition is recorded as more than one event, e.g. start-complete, schedule-start-complete, it’s suggested to choose “Yes”.


Mapping option

Consider only activity name (strings before the first bracket) when premapping transition classes to patterns?

The next panel is where transition class should be mapped to event class patterns, defined before. In this panel, we can set a transition to have more than one pattern. For example, if execution of a transition “W_Registration” can be logged as either one event (indicating only completion) or two events (indicating start and completion), use this panel to map the transitions to two patterns W_Registration (complete) and W_Registration(start,complete). Click “Next” to proceed.

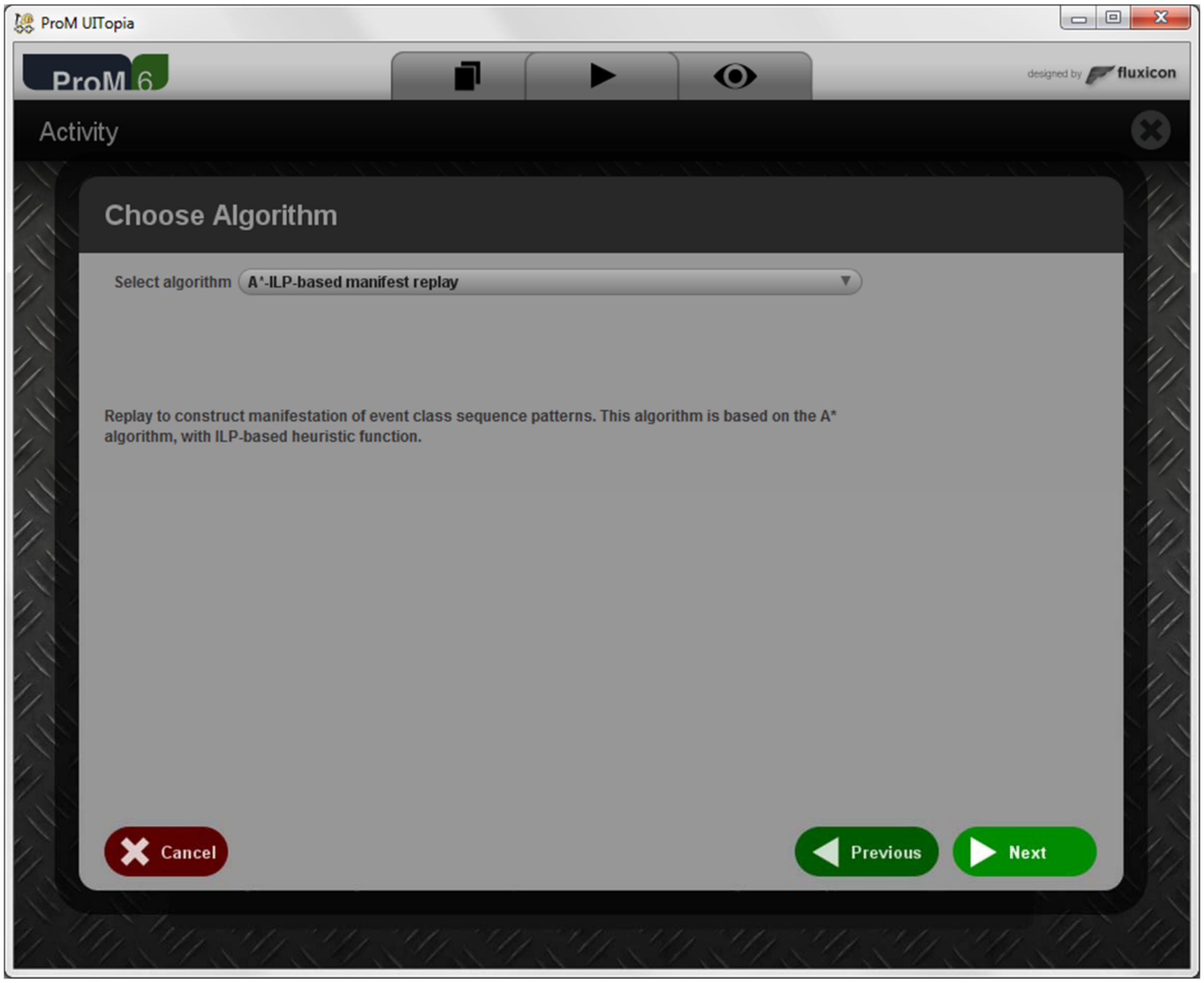


Map Transition Classes to Patterns

Transition Classifier: Label-based Classifier

Timeout	Timeout(complete)
W_Beoordelen fraude	W_Beoordelen_fraude(start,complete)
W_Besluit aanvraag	W_Besluit_aanvraag(start,complete)
W_Bezwaar	W_Bezwaar(start,complete)
W_Completeren aanvraag	W_Completeren_aanvraag(start,complete)
W_Nabellen offertes	W_Nabellen_offertes(start,complete)
W_Registration	W_Registration(start,complete)
W_Valideren aanvraag	W_Valideren_aanvraag(start,complete)
silent	[NONE]
skip	[NONE]

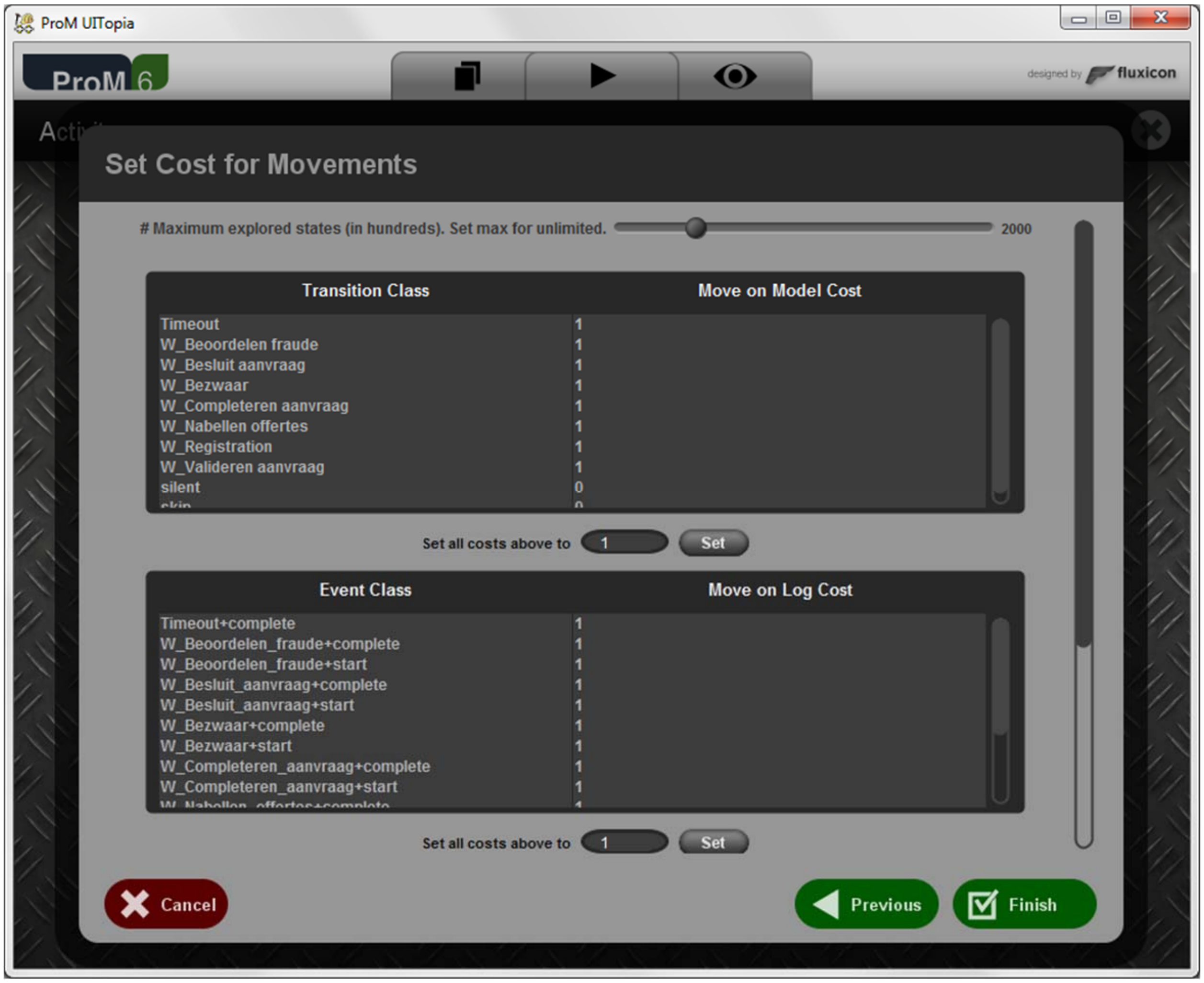
Choose a replay algorithm from the provided combo box. Press “Next” to continue.



Select algorithm: A*-ILP-based manifest replay

Replay to construct manifestation of event class sequence patterns. This algorithm is based on the A* algorithm, with ILP-based heuristic function.

Assuming that you choose the default algorithm (A* ILP-based manifest replay), you can configure the cost of violation and the limit of calculation that you allow to come up with reliable results (see below).



Set Cost for Movements

Maximum explored states (in hundreds). Set max for unlimited. 2000

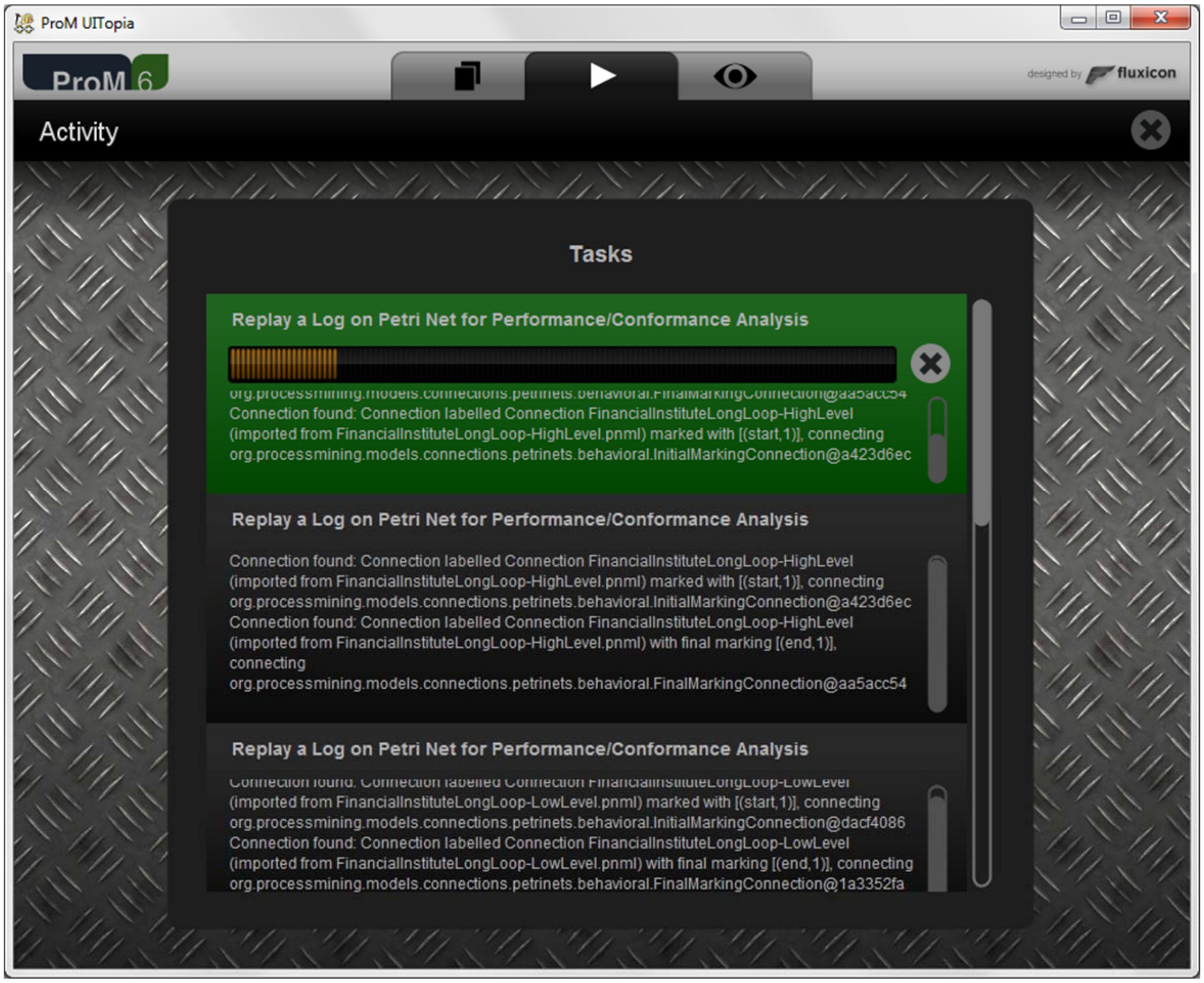
Transition Class	Move on Model Cost
Timeout	1
W_Beoordelen fraude	1
W_Besluit aanvraag	1
W_Bezwaar	1
W_Completeren aanvraag	1
W_Nabellen offertes	1
W_Registration	1
W_Valideren aanvraag	1
silent	0

Set all costs above to 1 Set

Event Class	Move on Log Cost
Timeout+complete	1
W_Beoordelen_fraude+complete	1
W_Beoordelen_fraude+start	1
W_Besluit_aanvraag+complete	1
W_Besluit_aanvraag+start	1
W_Bezwaar+complete	1
W_Bezwaar+start	1
W_Completeren_aanvraag+complete	1
W_Completeren_aanvraag+start	1

Set all costs above to 1 Set

Setting the maximum explored states to maximum (unlimited) means that you allow the algorithm to explore up to (2^{31} – 1) states to find an optimal alignment for each trace in the log. This may cause OutOfMemory exception in most personal computer after a long computation time (may take hours!) if the net and trace is really complex. The limit of 200,000 states is typically good enough to deal with real-life logs and models with moderate complexity. Click “Finish” when you’re done. A progress bar should be shown as follows.



Replay a Log on Petri Net for Performance/Conformance Analysis

```
org.processmining.models.connections.petrinets.behavioral.FinalMarkingConnection@aa5acc54
Connection found: Connection labelled Connection FinancialInstituteLongLoop-HighLevel
(imported from FinancialInstituteLongLoop-HighLevel.pnml) marked with [(start,1)], connecting
org.processmining.models.connections.petrinets.behavioral.InitialMarkingConnection@a423d6ec
```

Replay a Log on Petri Net for Performance/Conformance Analysis

```
Connection found: Connection labelled Connection FinancialInstituteLongLoop-HighLevel
(imported from FinancialInstituteLongLoop-HighLevel.pnml) marked with [(start,1)], connecting
org.processmining.models.connections.petrinets.behavioral.InitialMarkingConnection@a423d6ec
Connection found: Connection labelled Connection FinancialInstituteLongLoop-HighLevel
(imported from FinancialInstituteLongLoop-HighLevel.pnml) with final marking [(end,1)],
connecting
org.processmining.models.connections.petrinets.behavioral.FinalMarkingConnection@aa5acc54
```

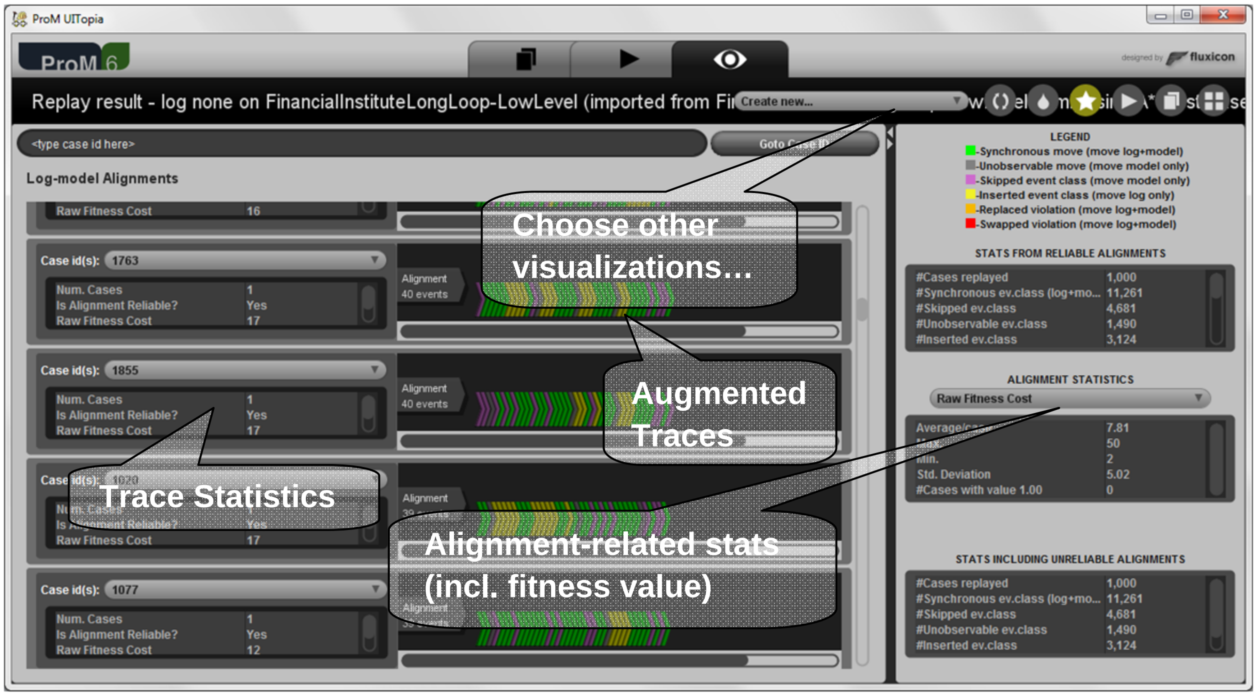
Replay a Log on Petri Net for Performance/Conformance Analysis

```
Connection found: Connection labelled Connection FinancialInstituteLongLoop-LowLevel
(imported from FinancialInstituteLongLoop-LowLevel.pnml) marked with [(start,1)], connecting
org.processmining.models.connections.petrinets.behavioral.InitialMarkingConnection@dacf4086
Connection found: Connection labelled Connection FinancialInstituteLongLoop-LowLevel
(imported from FinancialInstituteLongLoop-LowLevel.pnml) with final marking [(end,1)], connecting
org.processmining.models.connections.petrinets.behavioral.FinalMarkingConnection@1a3352fa
```

5. Result

Result of the plug-in is a Manifest object that contains all alignments between the log and net. There are several visualizations of a Manifest. The work in [2] shows a study case of how some of these visualizations are exploited to analyze the logged process.

If you only installed the PNetReplayer package, the alignments are visualized with "Projects Alignments to Log" visualization, similar to the one shown below.



Replay result - log none on FinancialInstituteLongLoop-LowLevel (imported from Fi

<type case id here>

Log-model Alignments

Raw Fitness Cost	16

Case id(s): 1763 — Alignment 40 events

Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	17

Case id(s): 1855 — Alignment 40 events

Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	17

Case id(s): 1020 — Alignment 39 events

Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	17

Case id(s): 1077 — Alignment 38 events

Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	12

LEGEND
- 🟩-Synchronous move (move log+model)
- ⬛-Unobservable move (move model only)
- 🟪-Skipped event class (move model only)
- 🟨-Inserted event class (move log only)
- 🟧-Replaced violation (move log+model)
- 🟥-Swapped violation (move log+model)

STATS FROM RELIABLE ALIGNMENTS

#Cases replayed	1,000
#Synchronous ev.class (log+mo...	11,261
#Skipped ev.class	4,681
#Unobservable ev.class	1,490
#Inserted ev.class	3,124

ALIGNMENT STATISTICS

Raw Fitness Cost

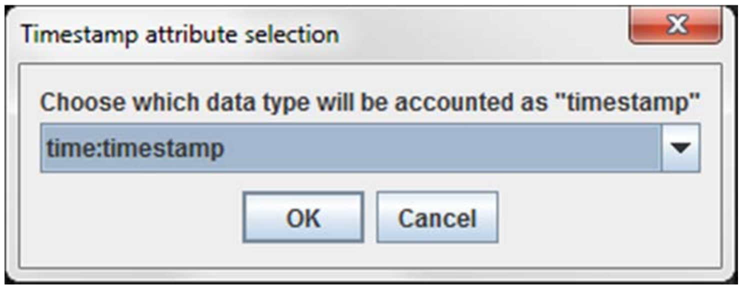
Average/case	7.81
Max.	50
Min.	2
Std. Deviation	5.02
#Cases with value 1.00	0

STATS INCLUDING UNRELIABLE ALIGNMENTS

#Cases replayed	1,000
#Synchronous ev.class (log+mo...	11,261
#Skipped ev.class	4,681
#Unobservable ev.class	1,490
#Inserted ev.class	3,124

If you installed the PNetAlignmentAnalysis package, you more visualization options and have different default visualization (Performance projection to model).

Choose the type of event attributes that indicates timestamp. By default of XES standard, it is “time:timestamp”.

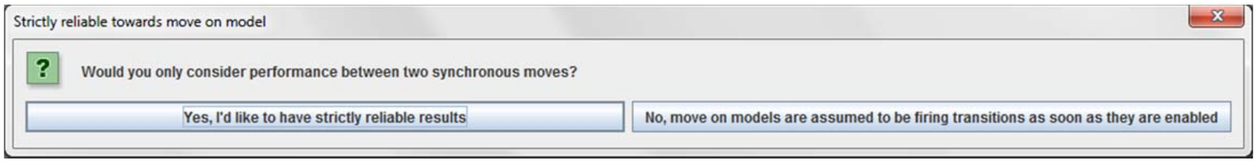


Timestamp attribute selection

Choose which data type will be accounted as "timestamp"

time:timestamp

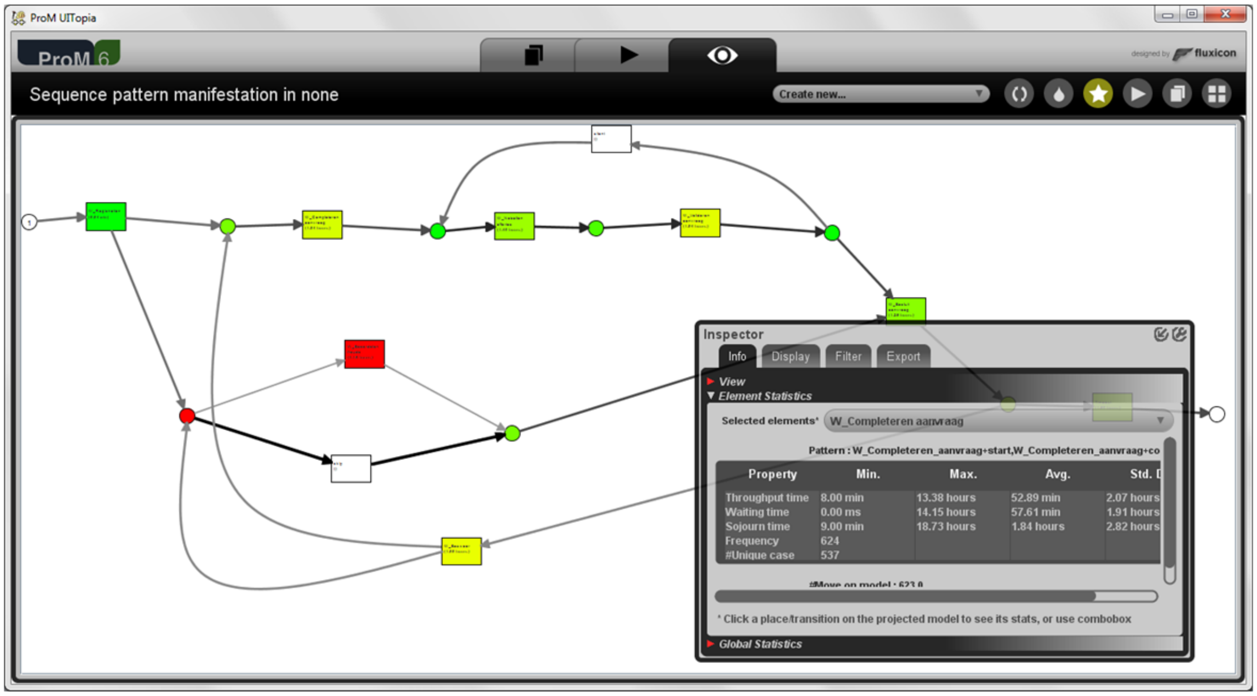
Then, a popup panel will be shown, asking whether you’d like strictly reliable performance measurements, or measurements where skipped tasks are assumed to immediately execute whenever they are enabled (see [2]). Otherwise, it’ll only measure performance based on two consecutive synchronous moves.


Yes, I'd like to have strictly reliable results
No, move on models are assumed to be firing transitions as soon as they are enabled

Strictly reliable towards move on model

Would you only consider performance between two synchronous moves?

You should have performance measurement like the followings



Selected elements* W_Completeren aanvraag

Pattern : W_Completeren_aanvraag+start,W_Completeren_aanvraag+co

Property	Min.	Max.	Avg.	Std. D
Throughput time	8.00 min	13.38 hours	52.89 min	2.07 hours
Waiting time	0.00 ms	14.15 hours	57.61 min	1.91 hours
Sojourn time	9.00 min	18.73 hours	1.84 hours	2.82 hours
Frequency	624			
#Unique case	537			

#Move on model : 623.0

* Click a place/transition on the projected model to see its stats, or use combobox

This visualization projects performance information onto the original model. It uses seamless coloring scheme from green (low value) to red (high value). By default, transition color shows the average sojourn time of transitions, place color shows the average waiting time of places, and arc thickness shows frequency of passing tokens. However, this color scheme can also be modified from the Display tab. Click an element in the model (place or transition), and see its stats in the Info tab, Element Statistics panel.

This visualization also allows you to separate quickly finished cases with cases that take so much time to complete. Use the Filter and Export tab for this purpose.

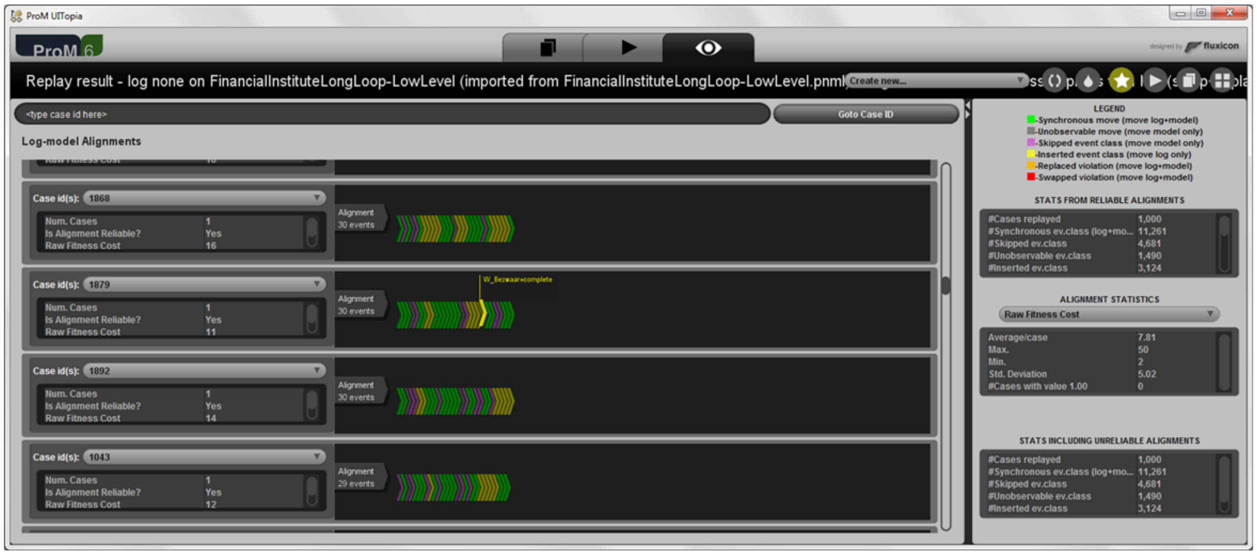
Apart of the two default visualizations, there are other visualizations to analyze conformance/performance from various perspectives. Readers are referred to [3] for a study case that shows the benefit of using various visualizations of alignments. To date, these are the list of all available visualizations:

1. *Project Manifest to Log*

. In this perspective, you can inspect per trace, which deviations occurred, and what are their possible causes. Use the “Goto Case ID” button to jump into your case of interest.

2. *Project Alignments to Log*

This is the standard visualization of alignments. In this perspective, you can inspect per trace, which deviations occurred, and what are their possible causes. Use the “Goto Case ID” button to jump into your case of interest.



Replay result - log none on FinancialInstituteLongLoop-LowLevel (imported from FinancialInstituteLongLoop-LowLevel.pnml

Log-model Alignments

Case id(s):	1868
Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	16
Case id(s):	1879
Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	11
Case id(s):	1892
Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	14
Case id(s):	1043
Num. Cases	1
Is Alignment Reliable?	Yes
Raw Fitness Cost	12

LEGEND

- 🟩-Synchronous move (move log+model)
- ▨-Unobservable move (move model only)
- 🟪-Skipped event class (move model only)
- 🟨-Inserted event class (move log only)
- 🟧-Replaced violation (move log+model)
- 🟥-Swapped violation (move log+model)

STATS FROM RELIABLE ALIGNMENTS

#Cases replayed	1,000
#Synchronous ev.class (log+mo...	11,261
#Skipped ev.class	4,681
#Unobservable ev.class	1,490
#Inserted ev.class	3,124

ALIGNMENT STATISTICS

Raw Fitness Cost

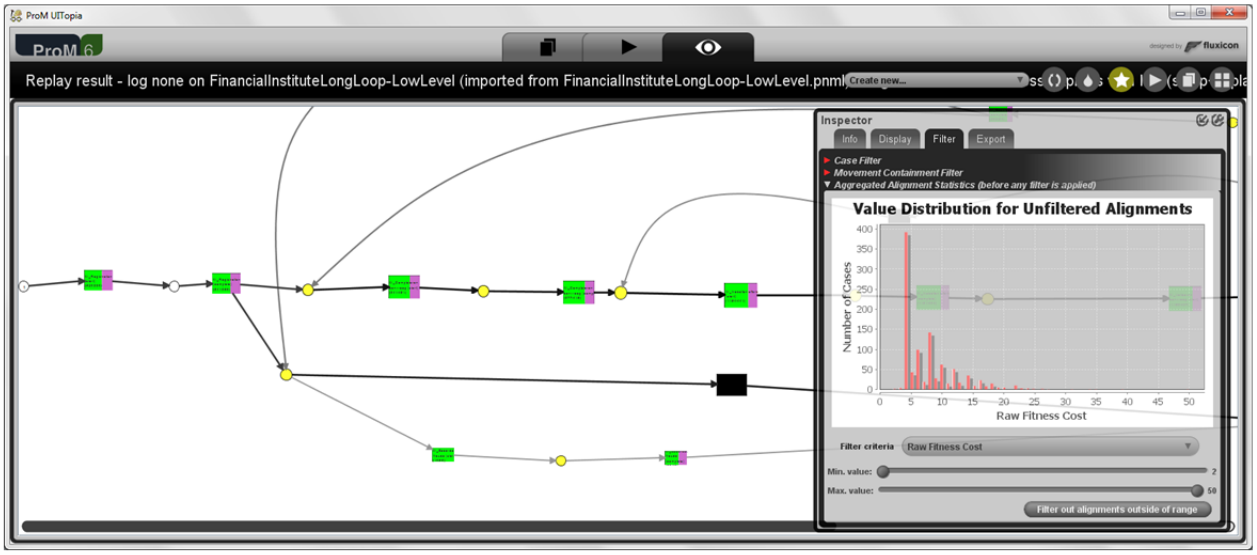
Average/case	7.81
Max.	50
Min.	2
Std. Deviation	5.02
#Cases with value 1.00	0

STATS INCLUDING UNRELIABLE ALIGNMENTS

#Cases replayed	1,000
#Synchronous ev.class (log+mo...	11,261
#Skipped ev.class	4,681
#Unobservable ev.class	1,490
#Inserted ev.class	3,124

3. *Project Manifest to Model for Conformance*

This visualization shows in the original model, which tasks are often skipped, and when extra activities that should not be performed according to the model are performed in reality.


ProM UITopia
ProM 6
designed by fluxicon
Create new...
Inspector
Info
Display
Filter
Export
Case Filter
Movement Containment Filter
Aggregated Alignment Statistics (before any filter is applied)
Value Distribution for Unfiltered Alignments
Number of Cases
Raw Fitness Cost
0
50
100
150
200
250
300
350
400
5
10
15
20
25
30
35
40
45
50
Filter out alignments outside of range

Replay result - log none on FinancialInstituteLongLoop-LowLevel (imported from FinancialInstituteLongLoop-LowLevel.pnml

Filter criteria: Raw Fitness Cost

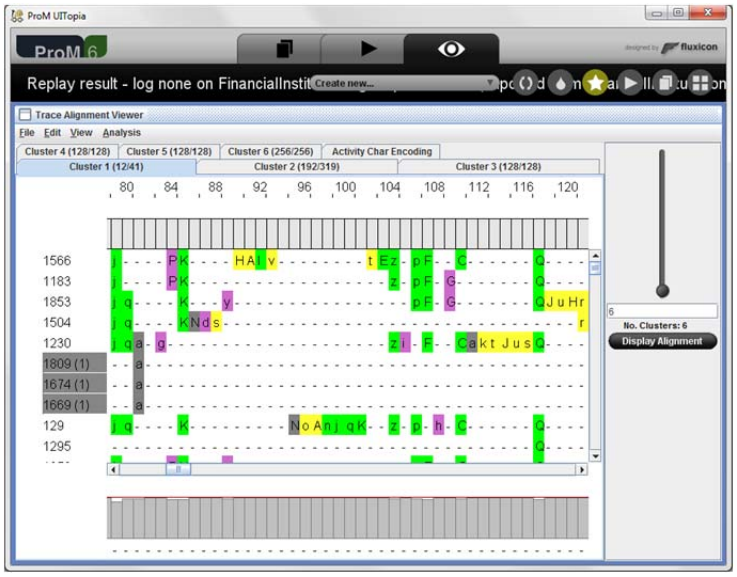
Min. value: 2

Max. value: 50

Use this visualization to get an overview of where deviations are located, separate severely deviating cases from non-deviating ones, and analyze deviations that often co-occur together. For example, you can export cases where deviations are severe as a new log by first filter out non-severely deviating cases using the filter tab, and then export shown cases as a new Log using the export tab.

4. Trace Alignment of Manifests

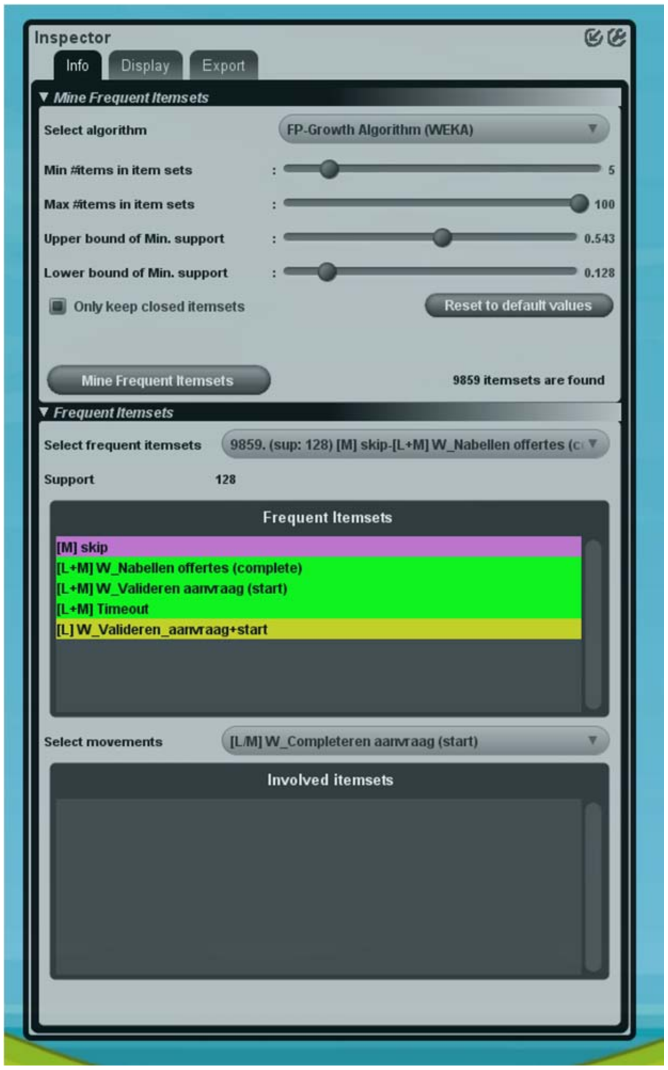
This visualization uses the trace alignment technique of [4]. Use this visualization to get an overview of the context in which deviations occur.



5. xFrequent Movement Sets Mining for Deviation Analysis

This visualization use Weka’s data mining library to mine frequently occurring deviations from all alignments. Typical analysis that you may obtain with this visualization is: “If task A is skipped, task B is also skipped”, or “If event class C occurs (although it should not according to the model), task D is skipped”.

Use the “Info” tab, the Mine Frequent Itemsets pane to mine frequently occurring patterns (shown below). Then, choose an itemset in the Frequent Itemsets panel to see its projection onto the process model. This way, you know where is the itemset.



Inspector

Info | Display | Export

▼ Mine Frequent Itemsets

Select algorithm: FP-Growth Algorithm (WEKA)

Min #items in item sets : 5

Max #items in item sets : 100

Upper bound of Min. support : 0.543

Lower bound of Min. support : 0.128

Only keep closed itemsets

Reset to default values

Mine Frequent Itemsets

9859 itemsets are found

▼ Frequent Itemsets

Select frequent itemsets: 9859. (sup: 128) [M] skip-[L+M] W_Nabellen offertes (c

Support 128

Frequent Itemsets

[M] skip
[L+M] W_Nabellen offertes (complete)
[L+M] W_Valideren aanvraag (start)
[L+M] Timeout
[L] W_Valideren_aanvraag+start

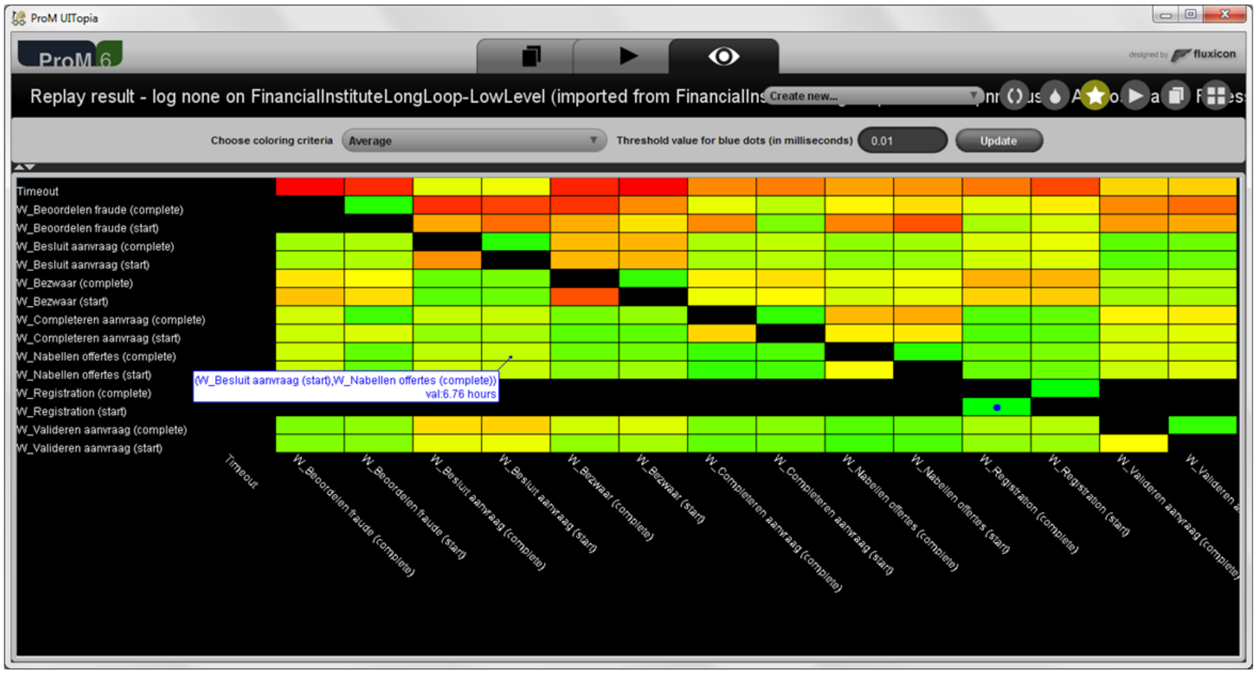
Select movements: [L/M] W_Completeren aanvraag (start)

Involved itemsets

The projection onto the model only shows the synchronous moves and move on model (skipped tasks).

6. Synchronous Transitions Analysis

This visualization is useful to identify time elapsed between tasks and tasks which are synchronous/batched. Given a collection of alignments, this visualization only take **the first occurrences** of all tasks (only their synchronous moves) in each alignment, and measure time between the moment a task occurs and the moment others occur. The color of cells indicate the value of the measured time compared to other (red = high, green = low). Read the matrix from absis then ordinate, e.g. in the example below, the time elapsed between W_Besluit aanvraag (start) and W_Nabellen offertes (complete) is 6.76 hours.


Replay result - log none on FinancialInstituteLongLoop-LowLevel (imported from FinancialIns

Choose coloring criteria: Average

Threshold value for blue dots (in milliseconds): 0.01

(W_Besluit aanvraag (start),W_Nabellen offertes (complete))
val:6.76 hours

ProM UITopia
ProM 6
designed by fluxicon
Create new...
Update
Timeout
W_Beoordelen fraude (complete)
W_Beoordelen fraude (start)
W_Besluit aanvraag (complete)
W_Besluit aanvraag (start)
W_Bezwaar (complete)
W_Bezwaar (start)
W_Completeren aanvraag (complete)
W_Completeren aanvraag (start)
W_Nabellen offertes (complete)
W_Nabellen offertes (start)
W_Registration (complete)
W_Registration (start)
W_Valideren aanvraag (complete)
W_Valideren aanvraag (start)

Set a dot threshold value from the textfield on the top right of the screen. If the measured time of a cell in the visualization goes below the threshold, a blue dot is shown on top of the cell. Such blue dots are useful to identify automated tasks/tasks that always occur within short period.

6. References

[1] W. van der Aalst, A. Adriansyah and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *WIREs Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 182-192, 2012.

[2] A. Adriansyah, B. van Dongen, D. Piessens, M. Wynn and M. Adams, "Robust Performance Analysis on YAWL Process Models with Advanced Constructs," *Journal of Information Technology Theory and Application (JITTA),* vol. 12, no. 3, pp. 5-26, 2012.

[3] A. Adriansyah and J. Buijs, "Mining Process Performance from Event Logs: The BPI Challenge 2012 Case Study," BPM Center Report BPM-12-15, BPMcenter.org, 2012.

[4] J. Bose and W. van der Aalst, "Process diagnostics using trace alignment : opportunities, issues, and challenges," *Information Systems,* vol. 37, no. 2, pp. 117-141, 2012.