

# Statechart ProM Plugin – User Manual

Statechart Workbench: Process Exploration Workflow Tool  
Supporting Hierarchy, Recursions, Cancellations and Software Logs

M. Leemans (m.leemans@tue.nl)

November 30, 2017

## 1 Introduction

The Statechart Workbench can be used to discover models with *hierarchy*, *recursion* and *cancellation* from ordinary event logs. The models can be displayed as *statecharts*, *sequence diagrams*, *Petri nets* and other visual formalisms. The Statechart Workbench is a process exploration workflow tool and it provides an easy and guided approach to apply a collection of advanced algorithms.

The original purpose of the Statechart Workbench was to analyse event logs obtained from instrumented software, but the tool is able to handle any ordinary (business) event log. To start analysing your own software, create a software event log using the instrumentation tool available at: <https://svn.win.tue.nl/repos/prom/XPort/>.

## 2 Installation

There are two ways to install and use the Statechart Workbench. The default way is to install the *Statechart* plugin via the ProM Package Manager. Recent nightly builds should already have this plugin installed by default. The ProM nightly build can be found at: <http://www.promtools.org/doku.php?id=nightly>.

Alternatively, one can checkout the source code repository, and load the Eclipse launch file *ProM with UITopia (Statechart-hg).launch*. Note: you can import this codebase as an existing Eclipse project.

## 3 Getting Started

In this section, step-by-step instructions are given to get started with the Statechart Workbench. For this guide we use JUnit software event log, publicly available at: <https://doi.org/10.4121/uuid:cfed8007-91c8-4b12-98d8-f233e5cd25bb>.

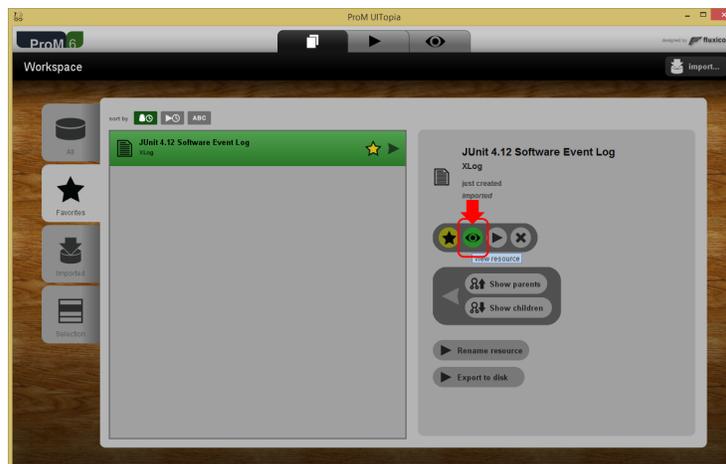
### 3.1 Step 1: Load an event log

To get started, first load an event log into ProM as usual. You can either drag an XES log onto ProM, or use the *Import* button on the top right.

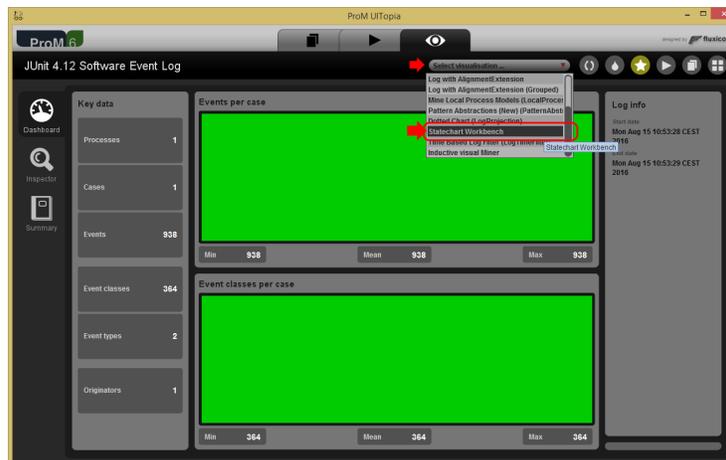
Next, launch the Statechart Workbench. The preferred way to do this is to simply launch the *Statechart Workbench* log visualizer:

1. View the event log resource (Figure 1(a)).
2. Launch the *Statechart Workbench* log visualizer (Figure 1(b)).

Alternatively, one can start the *Discover using the Statechart Workbench* plugin action (Figure 2).



(a) Step 1 – Viewing an event log



(b) Step 2 – Launching the *Statechart Workbench* log visualizer.

Figure 1: Launching the *Statechart Workbench* log visualizer.

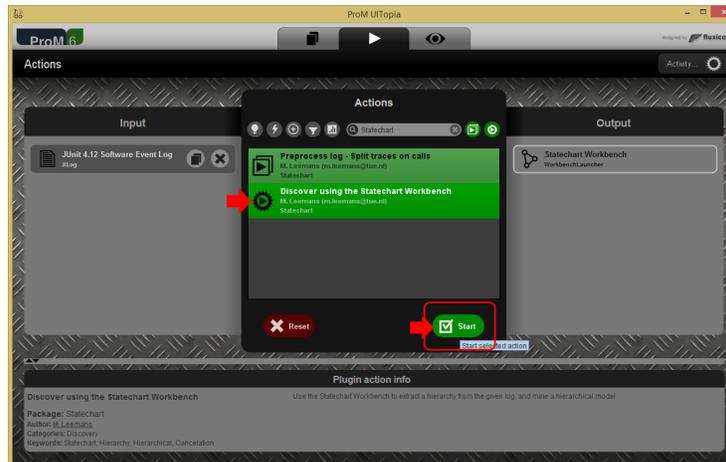


Figure 2: Launching the *Discover using the Statechart Workbench* plugin action.

### 3.2 Step 2: Setup Key Heuristics

The Statechart Workbench implements a simple workflow. First, you setup some key heuristics: *Hierarchy* and *Cancellation*. After that, the discovered model is shown in the the *Discovery & Analysis screen* screen, allowing you to explore and and perform analyses. In the top left corner of the Statechart Workbench (see Figure 3), the workflow and the current step is shown in three big chevrons.

Hierarchy and Cancellation are two key heuristics in the Statechart Workbench that require some setup. Before you can start discovering a model, you need to select a strategy for these heuristics. In most cases, you can choose one of the two quick presets, see also Figure 3:

- **Start Discovery – Normal Log** This preset covers most normal logs: no special hierarchy or cancellation is setup.
- **Start Discovery – Software Log** This preset covers most software logs: the default software hierarchy and cancellation heuristics are used.

For the JUnit example log, we can just select the **Start Discovery – Software Log** preset. If you choose a preset, you immediately jump to the *Discovery & Analysis screen* screen, see Section 3.3. If you choose to configure the heuristics manually, you get to option to select a hierarchy heuristic, and afterwards a cancellation heuristic.

**Note: Classifier Selection** The activity classifier used for discovery is set in the hierarchy heuristic.

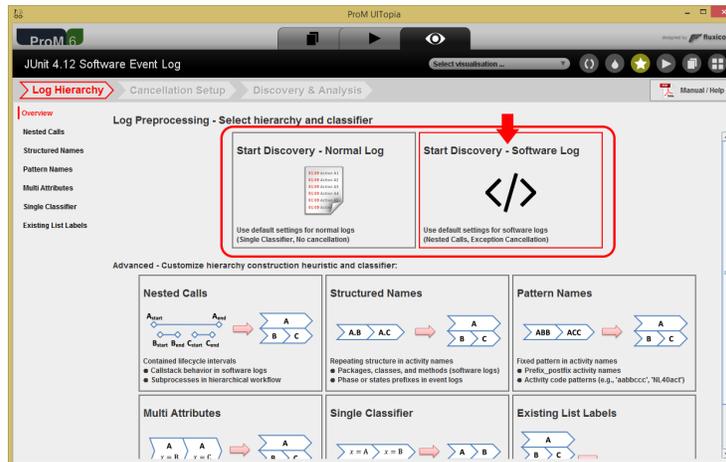


Figure 3: Initial screen when launching the *Statechart Workbench*: the setup of the key heuristics. In red are the two quick preset options.

### 3.2.1 Log Hierarchy

There are several heuristics options for creating a log hierarchy. In short, a hierarchy heuristics tries to create a multi-level (hierarchical) structure over the event log, such that the discovery algorithm can discover hierarchy, subprocesses and recursions. The available heuristics options are shown on the home screen in blocks, and listed on the sidebar to the left, see also Figure 3. Each heuristics option is explained in the tool with a visual cue and a description.

### 3.2.2 Cancellation Setup

After setting up a hierarchy, there are several heuristics options for creating a cancellation oracle. In short, a cancellation oracle tries to create a reference map of activities that are potentially involved in cancellation patterns. The cancellation oracle aids the discovery algorithm in finding these cancellation regions. The available heuristics options are shown on the *Cancellation Setup* screen in blocks, and listed on the sidebar to the left. Each heuristics option is explained in the tool with a visual cue and a description.

## 3.3 Step 3: Discovery and Analysis

After choosing a preset or setting up the heuristics, you end up on the *Discovery & Analysis* screen, see Figure 4. In the center of this screen, the discovered model is shown. On the left, a list of all the activities is shown. And on the right, the options and settings are shown. Using the three big chevrons at the top, different hierarchy cancellation heuristics can be set.

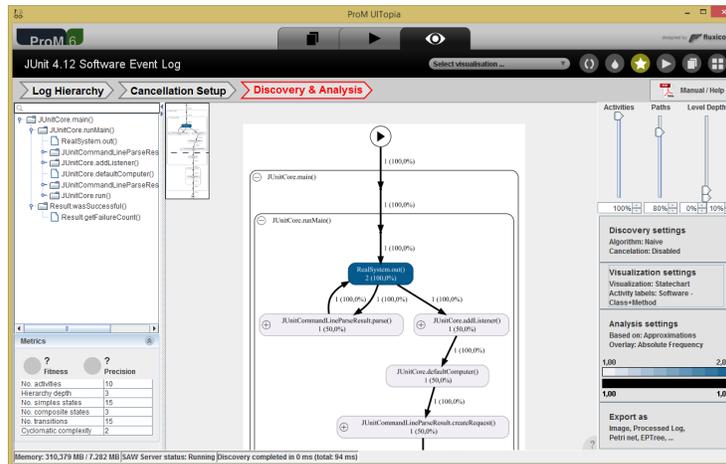


Figure 4: Discovery & Analysis screen screen of the Statechart Workbench.

### 3.3.1 Center panel: Discovered Model

The center panel shows the discovered model. By dragging the screen (left-click and hold on the model), you can move through the model. By using the scroll wheel, you can zoom in and out. By left clicking on a node, you can open up a detailed popup with more information (Figure 5).

In the case of hierarchical models, the plus and minus signs allow you to expand and collapse a subprocess / hierarchy level.

**Source Code Link** If you're using the SAW Eclipse plugin available via <https://svn.win.tue.nl/repos/prom/XPort/>, you can link the shown software model back to the source code. When a model is discovered, you can connect to the SAW server in the current Statechart Workbench, see Figure 6. The status of the SAW server is shown at the bottom of the screen. Once connected, a double click on a model node will trigger the SAW Eclipse plugin to open the corresponding source code line.

### 3.3.2 Left panel: Activity overview and selection

The panel on the left shows all the activities. The search bar at the top allows one to lookup specific activities: it filters the list. By selecting activities (left click), the corresponding parts in the center model will be outlined in red. To select multiple activities, use control click. To deselect, control click on selected activities.

In the case of hierarchical models, one can expand and collapse a subprocess / hierarchy level by clicking on the tree icons left of the activity names. The activity list on the left and model in the center are linked: expanding or collapsing in one view, will update the other.

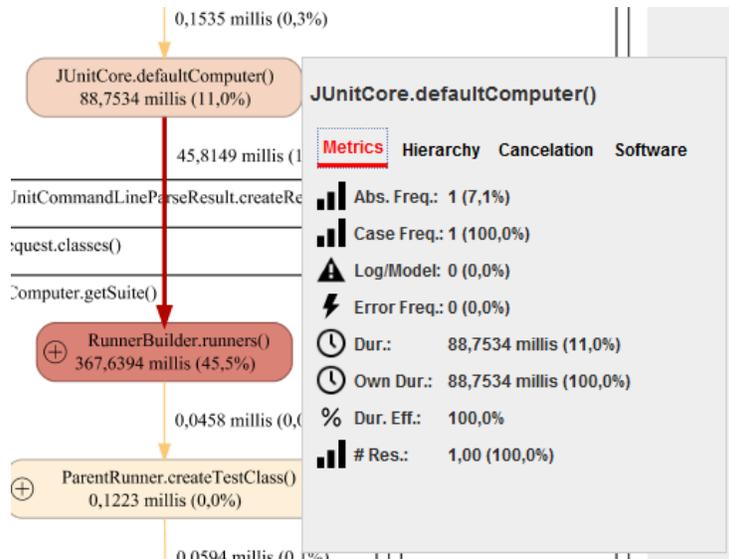


Figure 5: Clicking on a model element opens up an info popup with several tabs of detailed information.

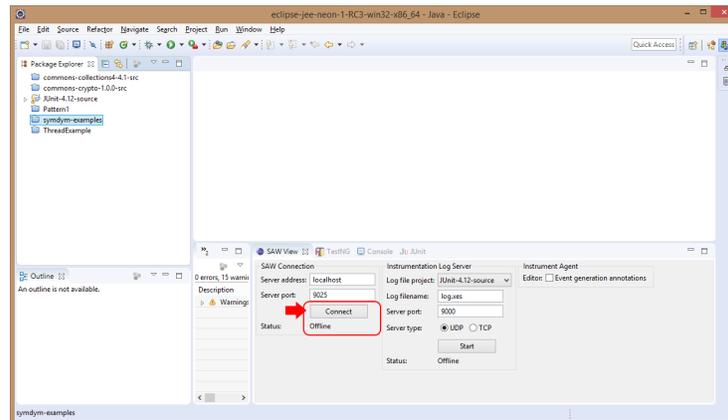


Figure 6: Connected the SAW Eclipse plugin after a model was discovered.

### 3.3.3 Right panel: Settings

The right panel shows all the options to manipulate the discovery settings and analysis overlays.

The three big sliders on top manipulate the activity, path, and level depth filters for discovery. The activities slider controls the fraction of activities that is included in log. The paths slider controls the amount of noise filtering applied. The level depth slider controls the amount of hierarchy levels shown / collapsed

in the model. When exploring hierarchical models, the level depth slider allows for a coarse grained exploration, while the plus and minus signs on the model allow for a fine grained exploration of the hierarchy.

Below the sliders are several setting buttons. These buttons show the current settings for discovery and analysis. When you click on these buttons, a popout allows you to set detailed settings. There are four such settings buttons:

**Discovery settings** Setup the recursion and cancelation detection, as well as toggling basic tree rewrite simplification.

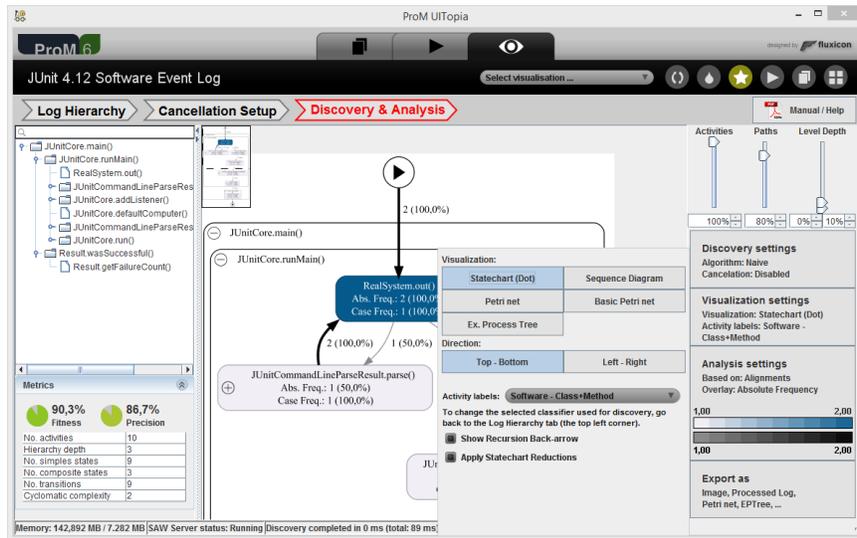
**Visualization settings** Choose the type of model shown, be it a statechart (Figure 7(a)), sequence diagram, Petri net or process tree (Figure 7(b)). In addition, one can choose the model layout orientation. For software logs, several activity label heuristics are available to reduce the activity names without changing the discovered model (e.g., removing the package name from the labels). Figure 7(a) shows the Visualization settings panel.

**Analysis settings** Setup the metric overlays displayed on the model. Different metrics can be calculated using either a fast approximation algorithm or the accurate alignment-based algorithms. The following metrics are provided:

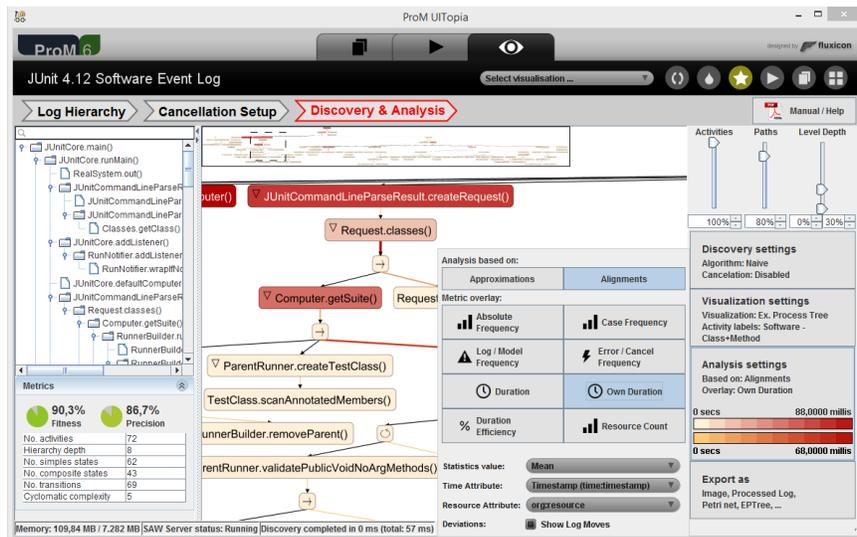
- **Absolute Frequency** How often did the method call / activity occur?
- **Case Frequency** In how many traces did the method call / activity occur?
- **Log / Model Frequency** How many deviations from the discovered model are present in the log?
- **Error / Cancel Frequency** How many times a cancel / exception trigger occurred?
- **Duration** What was the total duration of a method?
- **Own Duration** What was the duration of a method, minus the time spent in lower method calls?
- **Duration Efficiency** How much work was performed divided how much time? This can give an efficiency indication in case of multi-threaded code.
- **Resource Count** How many resources / threads were used to perform the method call / activity?

Figure 7(b) shows the Analysis settings panel.

**Export as** Export the model and/or processed log either as an image, or as a ProM object for further analysis.



(a) Shown is the *Statechart* model visualization with the *Frequency* metric overlay. In addition, the *Visualization settings* panel is opened.



(b) Shown is the *Process tree* model visualization with the *Own Duration* metric overlay. In addition, the *Analysis settings* panel is opened.

Figure 7: The user interface after the tool discovered and annotated a model. To the left, global metrics and a searchable alphabet summary is given. In the middle, an interactive, zoomable model with mini-map is given. The model and alphabet views are linked. To the right, the user can change various settings.