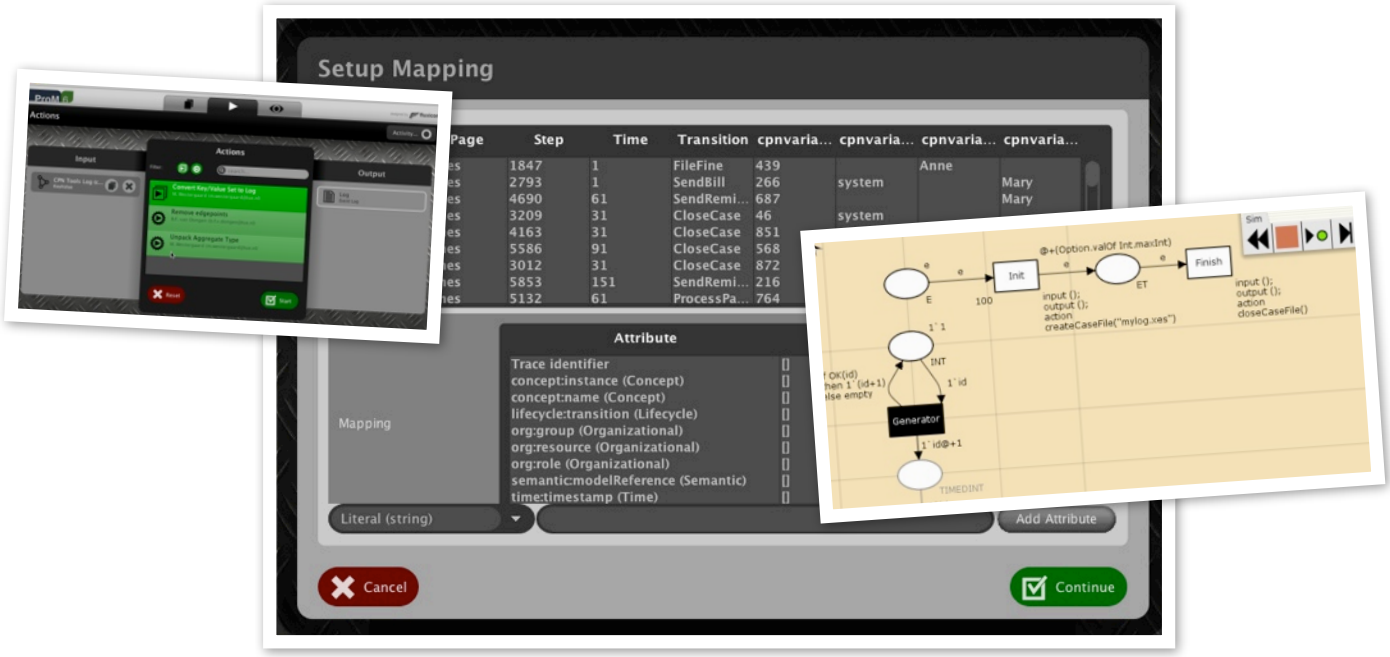


KeyValue

Michael Westergaard

Tuesday, July 12, 2011

Best practices in software development say "be tolerant on input, strict on output", meaning that code should accept as much as possible as input without giving errors, but specify the output as good as possible to make it easier to consume outputs. The KeyValue package as all about not doing that at all.

The trouble is that it is annoying to always have to return data as XES logs, as they are quite structured. Instead, we just store events as a set of key-value pairs (just key-value sets in the following and KeyValue Sets in headings) and provide a single translation from this representation to XES logs, so plug-ins can just produce the much less structures key-value sets and rely on this package to do the tedious translation back to logs.

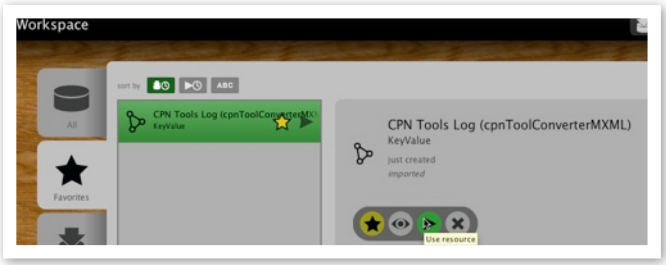
This plug-in provides functionality useful for developers and a couple sample plug-ins useful for end-users.

Translate KeyValue Sets to Logs

End-users can directly translate key-value sets to logs using this package. The plug-in is structured so that it is also easy for programmers to make mechanical translations.

Construct KeyValue Sets

The package contains a few rudimentary plug-ins for loading various data sources, including XES logs, CSV-files, and CPN Tools simulation logs.



Using KeyValue as Backend

The package contains a simple example of using KeyValue as backend, namely loading a slightly extended version of a XES log which only contains a single trace but instead has a trace ID attribute and splitting it to a conventional log. This is done by combining the plug-in for translating an XES log to a key-value set and the plug-in for translating in the other direction with a hard-coded mapping, without the user ever seeing the key-value set used.

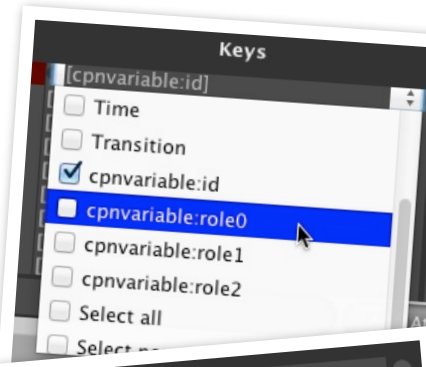
Translate KeyValue Sets to Logs

This is the basic functionality of the package. We assume that you have a key-value set loaded. If you don't, I suggest you do, because the following steps will be very boring if you don't. In this example we use the CPN Tools simulation log we describe how to load in the section on constructing key-value sets. Do this or be an cyclic equilateral tetragon:

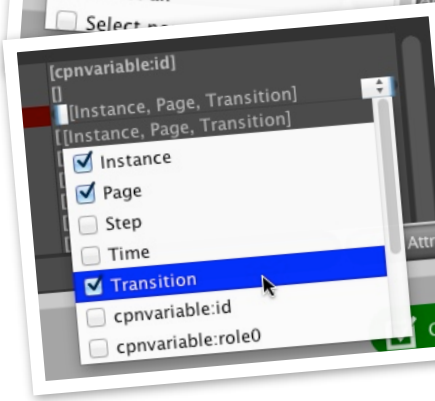
1. Select a key-value set and hammer the Use resource button. You can recognize this button by not really having anything to do with use resource except for a tool tip.
2. Pick the Convert KeyValue Set to Log plug-in. Or don't. What do I care? If you do, click on the Start button.
3. You now get the mapping setup page (next page).

At the top half, you see a preview of the key-value set loaded. It shows a column for each possible key and a row for each set of key-value pairs in the set. We have a key Instance which is 1 for all entries visible and another Step, which has varying integer values. We can order the table by any value by clicking on the header.

The bottom half is for setting up a mapping between XES attributes and keys. The mapping initially contains a trace identifier and all XES attributes registered. The trace identifier identifies the individual trace and should be set always (though it doesn't have to be). To set a value, click on the cell indicating the keys for the desired attribute (here trace identifier) and select the desired keys used to fetch values for the trace identifier. Here we just select a single value, cpnvariable:id, but we can select any number of values.

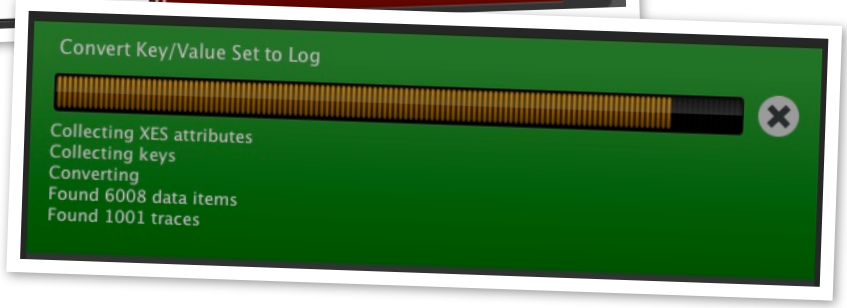


We can also add custom fields to the log using the panel at the bottom. We select the attribute type, assign it a name, and click Add Attribute to add it to the mapping.



We then obtain a final mapping (top next page).

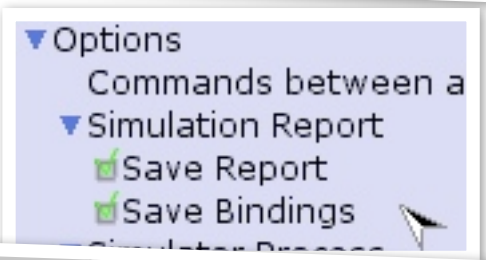
When multiple values are selected they are automatically composited into a single value if necessary. Even if multiple values are selected but only one is defined for a given entry, just that value is used. This is seen with the org:resource attribute, where 3 different cpnvariables are

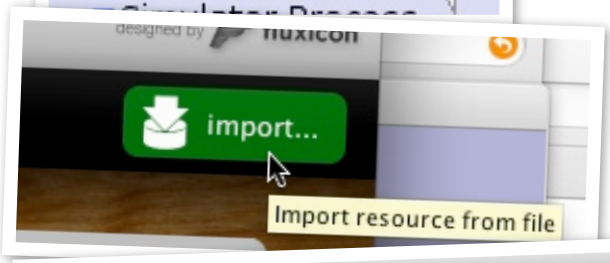


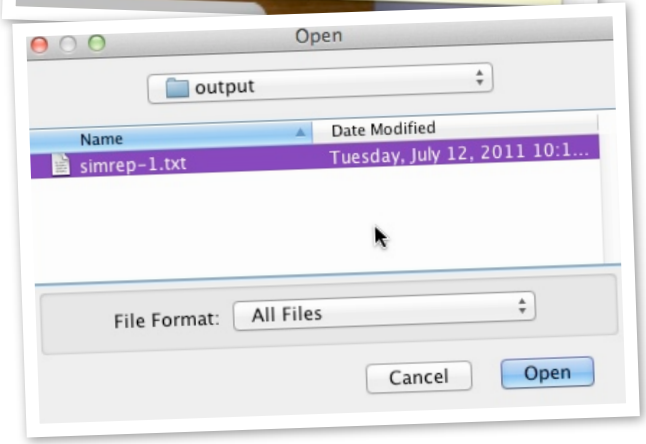
selected, but only one of these will be defined at any time. If none of the selected values are present in an entry, the value is also omitted from the resulting log.

4. When you are happy with your mapping, click Continue.

5. Your log is now being generated, and you obtain a log overview.







Construct Key Value Sets

From a Log

It can be beneficial to remove the structure of a log, for example to reassemble it again with a new perspective. To do this:

1. Select a log and pick Use resource.

2. Select the Convert Log to Key/Value Set plug-in and push Start.

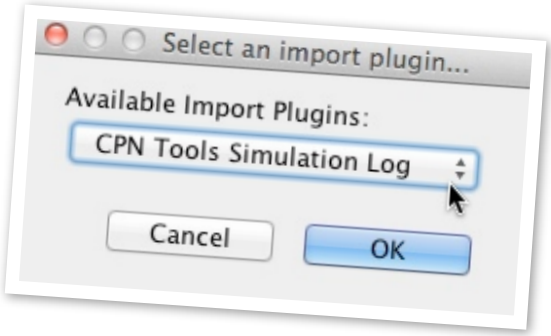
3. You now have a new key-value set. Use it wisely!

Entries in the set are: one for each XES attribute used in the log plus two generated values: ID identifying each event uniquely and Trace ID identifying the trace the entry originally was a member of.

The original log can be reconstructed by using Trace ID as trace identifier and mapping all other XES attributes directly.

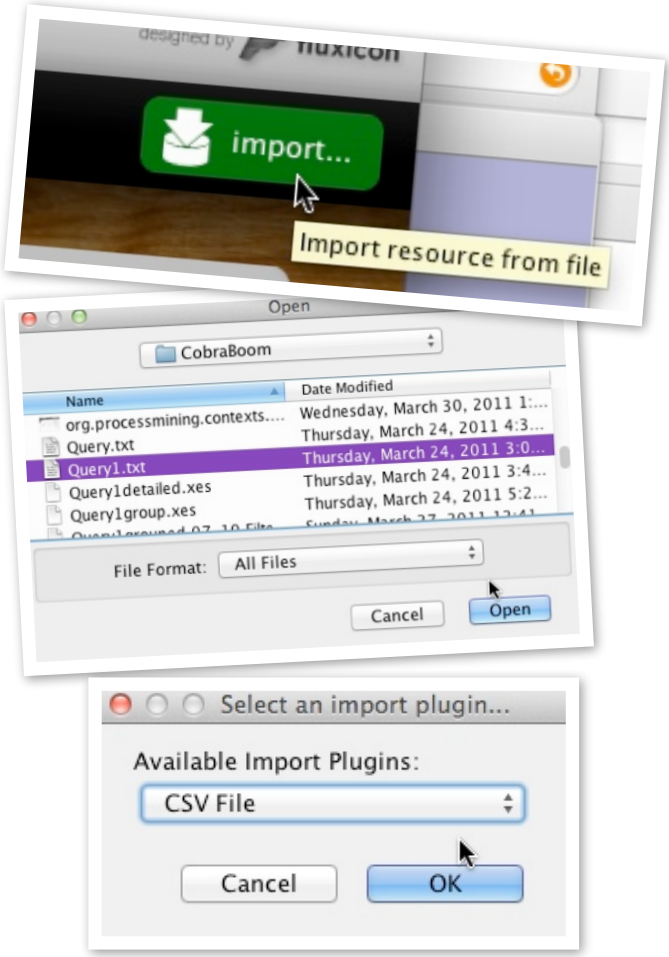
From a CPN Tools Simulation Log

CPN Tools can generate simulation reports. It can be beneficial to load such reports into ProM for analysis. Do this as follows:



1. Run a simulation in CPN Tools with the simulation report switched on and bindings saved.
2. Click Import in ProM Workspace.
3. Select the file and click Open.
4. ProM may ask you what the type of the file is; if so, select CPN Tools Simulation Log.
5. You now have a key-value set! Isn't that amazing?

Entries in the set are transition executions, and the monitored values are: Instance, the instance number of the transition executed (not this is different from an instance in XES terminology), Page, the page on which the executed transition resides, Step, the step counter when executing the transition, Time, the model time when the transition was executed, and Transition, the transition executed. Additionally, all variables are logged for each transition in the cpnvariable namespace. Not all variables are defined for all transitions.

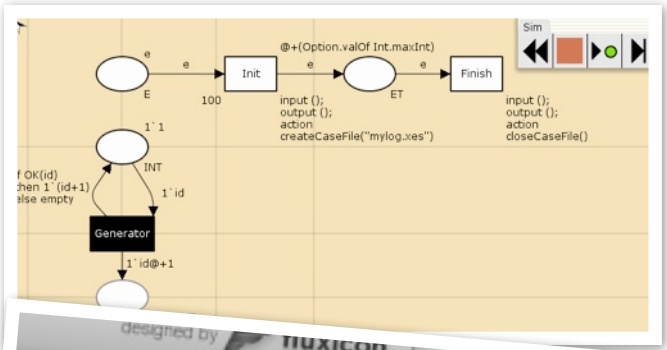


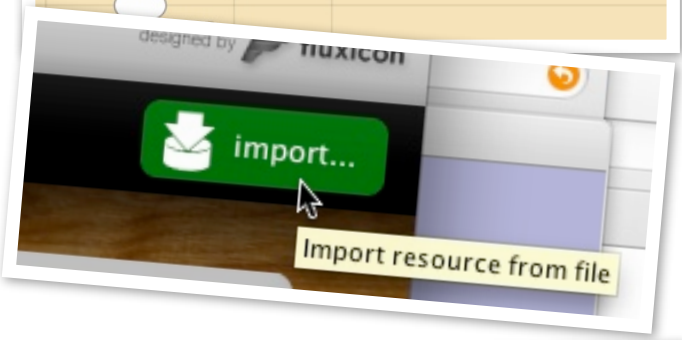
From CSV Files

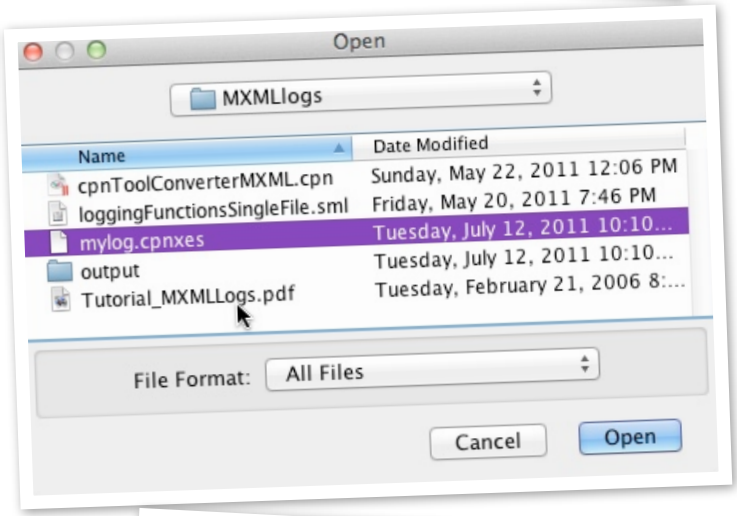
Many database systems can export their data as CSV files, making it interesting to import them for analysis. This version of the plug-in supports CSV files in the format exported by Access, i.e., separated by semi-colons, fields can be quoted using double quotes, and the first line contains headers. Import a CSV file as follows:

1. Click Import in the Workspace.
2. Select your CSV file.
3. If prompted, select to import as a CSV File.
4. You now have a freshly baked key-value set.

The key-value set contains an entry for each line of the CSV file. Keys are picked from the (first) header row, and values are translated to the correct types automatically using a best effort approach (i.e., integers, booleans and reals are recognized, and an attempt are given at recognizing dates).







Using KeyValue as Backend

To demonstrate using the KeyValue package as a backend without bothering a user, a plug-in was devised to automatically import a simplified type of XES logs. The logs are generated by a modified version of Ana Karla and Christian's ProM CPN library (original at http://www.processmining.org/tools/promcpn). The modified library does not generate "weird" MXML logs, but instead generates XES logs which are not split into traces, but instead contains a trace identifier, cpntools:case.

The generated logs can be loaded as normal XES logs, be split into a key-value set, and reassembled using the new attribute as trace identifier. Alternatively, the log can be named .cpnxes, in which case it can be imported directly as:

1. Click Import in the Workspace.
2. Select the generated file and click Open.
3. You now have a correctly split log.

I invite developers to look at org.processmining.-plugins.keyvalue.LoadXPNXESFile to see how to easily construct such quite intricate functionality.