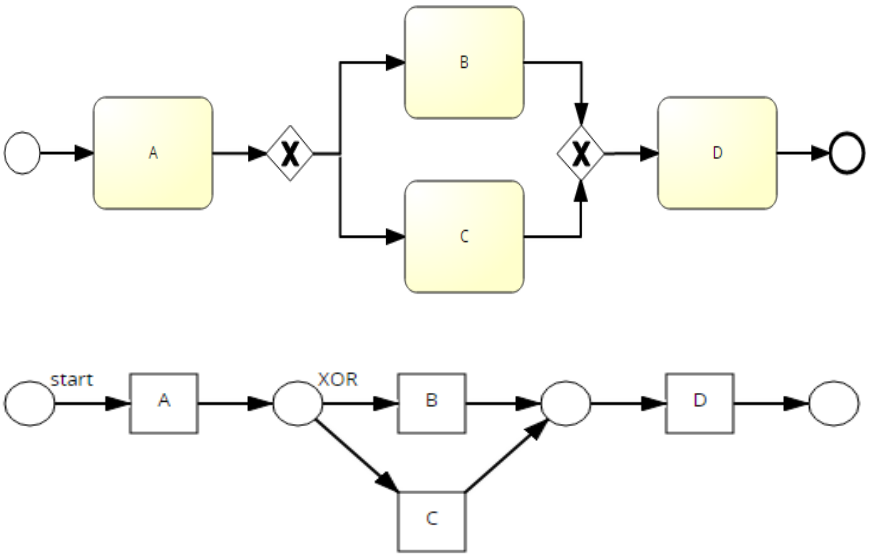
Lab 11: Process Mining

Process Discovery

Event Log

- Event logs contain the executed activities of process model that related to process instance.
- Format (case id , task\ activity name)

Case id	Activity
1	A
1	B
2	A
2	C
1	D
2	D



Alpha Algorithm

Definitions

Let T be a set of tasks - activities and T^* be the set of all sequnces or arbitrary lengths over T, it shall:

- *$\sigma \in T^*$ is called an execution sequence, if all activities in σ belong to the same process instance*
- *$W \subseteq T^*$ is called workflow log*

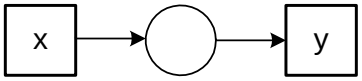
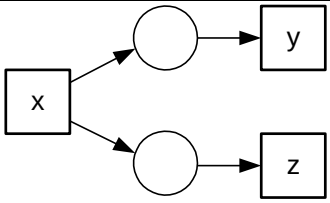
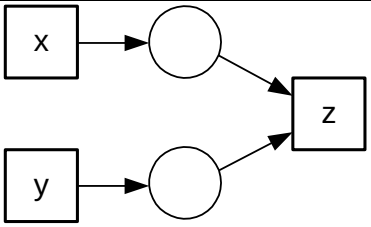
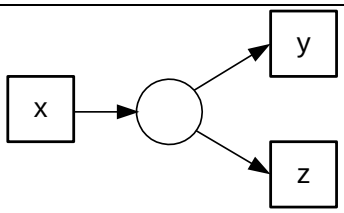
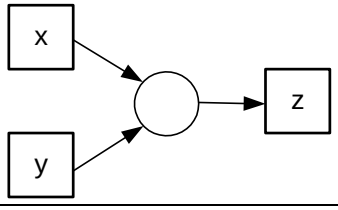
Assumptions

- In a process model each activity occurs at most once
- Any direct neighborhood relation between activities is observed at least once

Relations

- Direct follower : $a >_w b$ are in execution seqeunce iff b directly follows a.
- **Causality** : $a \rightarrow_w b$ iff $a >_w b$ but not $b >_w a$
- **Parallelism** : $a \parallel_w b$ iff $a >_w b$ and $b >_w a$
- **Exclusiveness** : $a \#_w b$ iff not $a >_w b$ and not $b >_w a$

Representation of relation in petri net

Sequence $x \rightarrow y$	
And split $x \rightarrow y$, $x \rightarrow z$ and $y \mid\mid z$	
And join $x \rightarrow z$, $y \rightarrow z$ and $x \mid\mid y$	
Xor split $x \rightarrow y$, $x \rightarrow z$ and $y \# z$	
Xor join $x \rightarrow z$, $y \rightarrow z$ and $x \# y$	

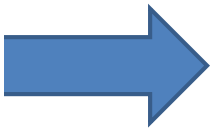
α-Algorithm

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\ t \in \sigma\}$- Set of distinct activities in W
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\ t = first(\sigma) \}$ - Set of start activities, first element in each trace in W
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\ t = last(\sigma) \}$ - Set of end activities, last element in each trace in W
4. $X_W = X_L = \{ (A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B}\ a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\ a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\ b_1 \#_W b_2 \}$ - set of (A,B) where $a \in A$ and $b \in B$ are in causality relation, all activities in A are independent relation and same for B
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$ - Delete (A,B) from X_W that are not maximal
6. $P_W \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$ - set of places
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, set of arcs
8. $a(W) = (P_W, T_W, F_W)$. Finally construct petri net

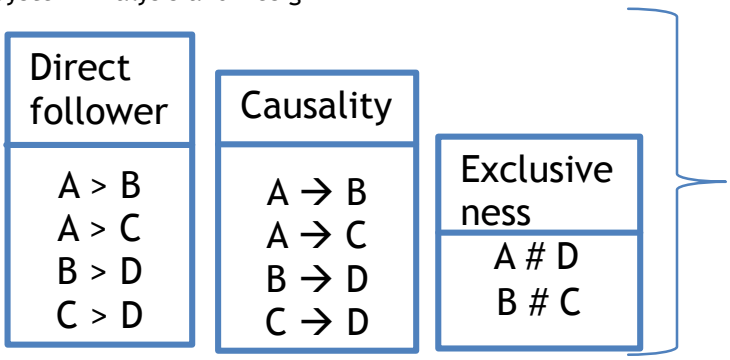
Examples

Example 1

Case id	Activity
1	A
1	B
2	A
2	C
1	D
2	D



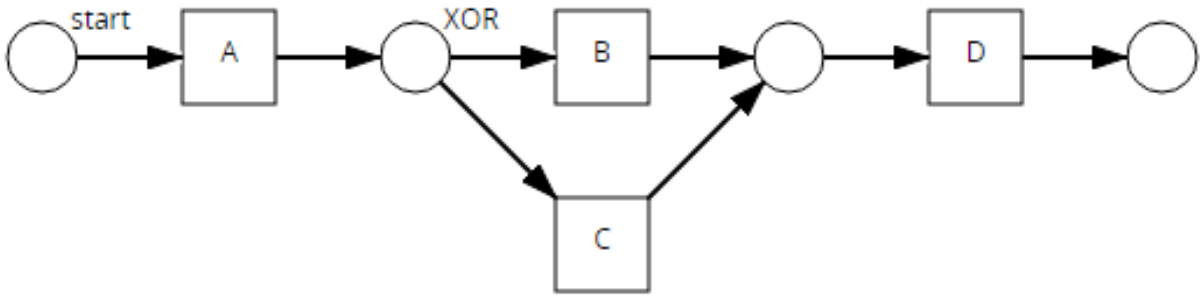
W= {ABD,ACD}
T = set of Tasks = {A,B,C,D}



	A	B	C	D
A	#	→	→	#
B	←	#	#	→
C	←	#	#	→
D	#	←	←	#

Applying alpha steps

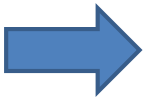
1. T_w={A,B,C,D}
2. T_I = {A}
3. T_o= {D}
4. X_w : { ({A},{B}), ({A},{C}),({A},{B,C}), ({B},{D}), ({C},{D}) , ({B,C},{D}) }
5. Y_w: { ({A},{B,C}), ({B,C},{D}) }
6. P_w: {$P_{(\{A\},\{B,C\})}$, $P_{(\{B,C\},\{D\})}$, i_w , o_w }
7. F_w: {(i_w , A), (A, $P_{(\{A\},\{B,C\})}$), ($P_{(\{A\},\{B,C\})}$,B), ($P_{(\{A\},\{B,C\})}$,C),(B,$P_{(\{B,C\},\{D\})}$), (C, $P_{(\{B,C\},\{D\})}$), ($P_{(\{B,C\},\{D\})}$,D), (D , o_w)}
8. a(W) = (P_W,T_W,F_W).



Example 2

Case id	Activity
0	A
0	B
0	C
1	A
1	D
1	E
0	E
0	F
0	G
0	H
1	G
1	F
1	H
2	A
2	D
2	E
2	F
2	G
2	H

Case id	Activity
3	A
3	B
3	C
4	A
4	D
4	E
4	F
4	G
4	H
3	E
3	G
3	F
3	H
5	A
5	B
5	C
5	E
5	G
5	F
5	H



W = {ABCEFGH, ABCEGFH, ADEGFH, ADEFGH}

Direct succession

A > B
A > D
B > C
C > E
D > E
E > F
E > G
F > G
F > H
G > F
G > H

Causality

A → B
A → D
B → C
C → E
D → E
E → F
E → G
F → H
G → H

Parallel

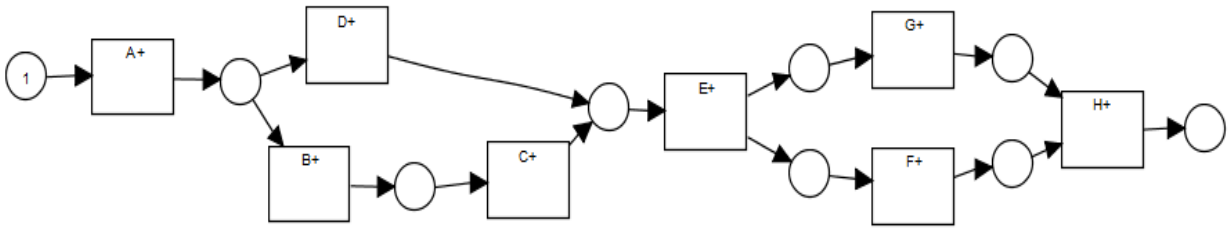
F || G

Choice	Choice
A # C	C # D
A # E	C # F
A # F	C # G
A # G	C # H
A # H	D # B
B # D	D # C
B # E	D # F
B # F	D # G
B # G	D # H
B # H	E # H

	A	B	C	D	E	F	G	H
A	#	→	#	→	#	#	#	#
B	←	#	→	#	#	#	#	#
C	#	←	#	#	→	#	#	#
D	←	#	#	#	→	#	#	#
E	#	#	←	←	#	→	→	#
F	#	#	#	#	←	#	||	→
G	#	#	#	#	←	||	#	→
H	#	#	#	#	#	←	←	#

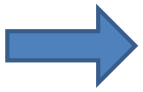
Applying alpha steps

1. T_w={A,B,C,D,E,F,G,H}
2. T_I = {A}
3. T_o= {H}
4. X_w = {({A},{B}),({A},{D}),({A},{B,D}),({B},{C}),({C},{E}),({D},{E}),({C,D},{E}), ({E},{F}), ({E},{G}), ({F},{H}), ({G},{H})}
5. Y_w ={({A},{B,D}),({B},{C}), ({C,D},{E}),({E},{F}), ({E},{G}), ({F},{H}), ({G},{H})}
6. P_w ={$P_{(\{A\},\{B,D\})}$, $P_{(\{B\},\{C\})}$, $P_{(\{C,D\},\{E\})}$, $P_{(\{E\},\{F\})}$, $P_{(\{E\},\{G\})}$, $P_{(\{F\},\{H\})}$, $P_{(\{G\},\{H\})}$, i_w , o_w }
7. F_w={(i_w , A), (A, $P_{(\{A\},\{B,D\})}$) , ($P_{(\{A\},\{B,D\})}$,B),($P_{(\{A\},\{B,D\})}$,D), (B, $P_{(\{B\},\{C\})}$), ($P_{(\{B\},\{C\})}$,C),(C, $P_{(\{C,D\},\{E\})}$) ,(D, $P_{(\{C,D\},\{E\})}$),($P_{(\{C,D\},\{E\})}$,E), (E, $P_{(\{E\},\{F\})}$), ($P_{(\{E\},\{F\})}$,F) , (E, $P_{(\{E\},\{G\})}$), ($P_{(\{E\},\{G\})}$,G) , (F, $P_{(\{F\},\{H\})}$), ($P_{(\{F\},\{H\})}$,H) , (G, $P_{(G\},\{H\})}$), ($P_{(\{G\},\{H\})}$,H), (H , o_w)}
8. a(W) = (P_W,T_W,F_W).



Example 3

Case id	Activity
0	A
0	B
0	D
1	A
1	C
1	G
1	H
0	E
0	F
0	H
2	A
2	C
2	G
3	A
3	C
2	H
3	G
3	H
4	A
4	B
4	E
4	D
4	F
4	H



W = {ABDEFH, ACGH, ABEDFH}

Direct follower
A > B
A > C
B > D
B > E
C > G
D > E
D > F
E > D
E > F
F > H
G > H

Causality
A → B
A → C
B → D
B → E
C → G
D → F
E → F
F → H
G → H

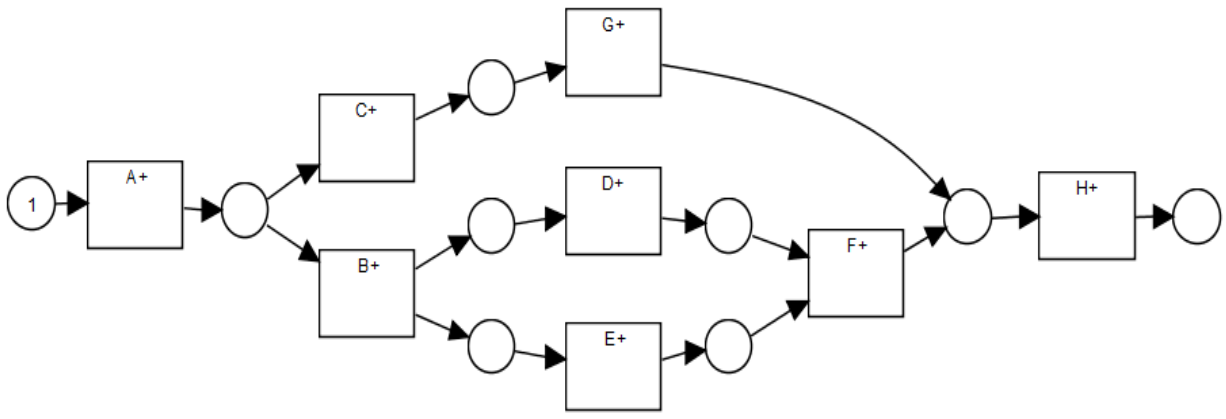
Parallel
D || E

Exclusive ness	Exclusive ness
A # D	C # D
A # E	C # E
A # F	C # F
A # G	C # H
A # H	D # C
B # C	D # G
B # F	D # H
B # G	E # G
B # H	E # H
	F # G

	A	B	C	D	E	F	G	H
A	#	→	→	#	#	#	#	#
B	←	#	#	→	→	#	#	#
C	←	#	#	#	#	#	→	#
D	#	←	#	#	||	→	#	#
E	#	←	#	||	#	→	#	#
F	#	#	#	←	←	#	#	→
G	#	#	←	#	#	#	#	→
H	#	#	#	#	#	←	←	#

Applying alpha steps

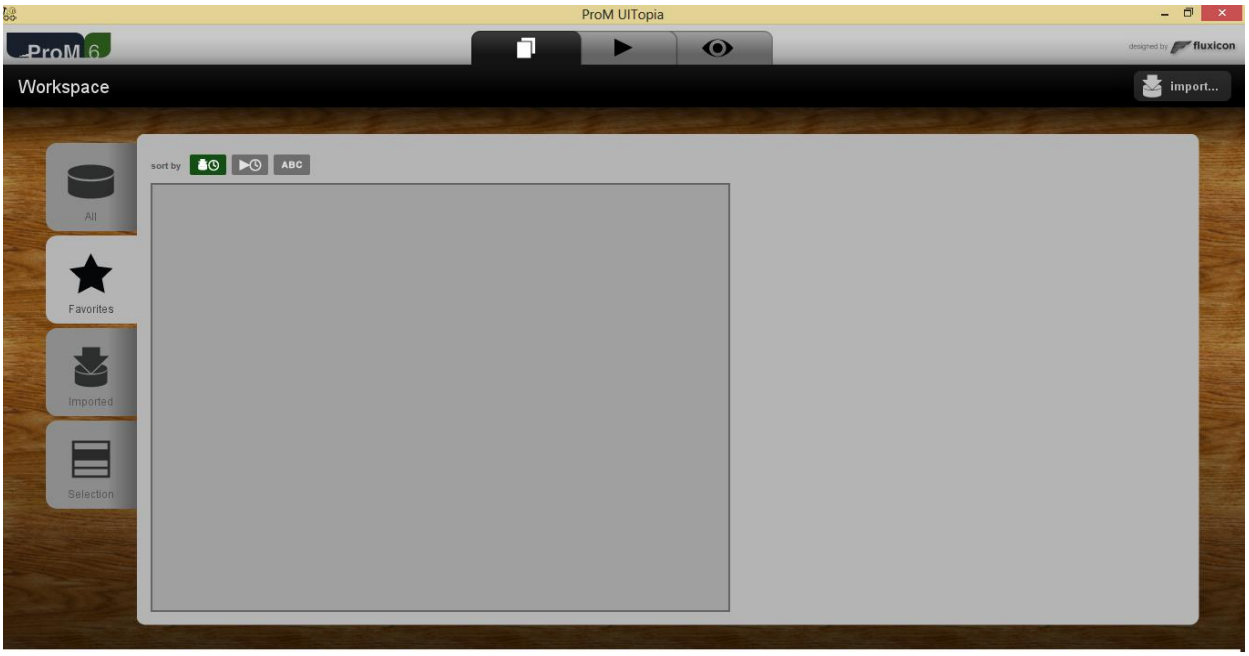
1. T_w={A,B,C,D,E,F,G,H}
2. T_I = {A}
3. T_o= {H}
4. X_w = {({A},{B}),({A},{C}),({A},{B,C}),({B},{D}),({B},{E}),({C},{G}),({D},{F}), ({E},{F}), ({F},{H}), ({G},{H}),({F,G},{H})}
5. Y_w ={({A},{B,C}),({B},{D}),({B},{E}),({C},{G}),({D},{F}), ({E},{F}),({F,G},{H})}
6. P_w =($P_{({A},{B,C})}$,$P_{({B},{D})}$,$P_{({B},{E})}$,$P_{({C},{G})}$,$P_{({D},{F})}$, $P_{({E},{F})}$,$P_{({F,G},{H})}$}
7. F_w={(i_w , A), (A, $P_{({A},{B,C})}$) , ($P_{({A},{B,C})}$,B),($P_{({A},{B,C})}$,C),(C, $P_{({C},{G})}$),($P_{({C},{G})}$,G), (B, $P_{({B},{D})}$), ($P_{({B},{D})}$,D),(D, $P_{({D},{F})}$) ,($P_{({D},{F})}$, F),(B,$P_{({B},{E})}$),($P_{({B},{E})}$,E), (E, $P_{({E},{F})}$), ($P_{({E},{F})}$,F) , (F, $P_{({F,G},{H})}$), (G, $P_{({F,G},{H})}$), ($P_{({F,G},{H})}$,H) , (H , o_w)}
8. a(W) = (P_W,T_W,F_W).

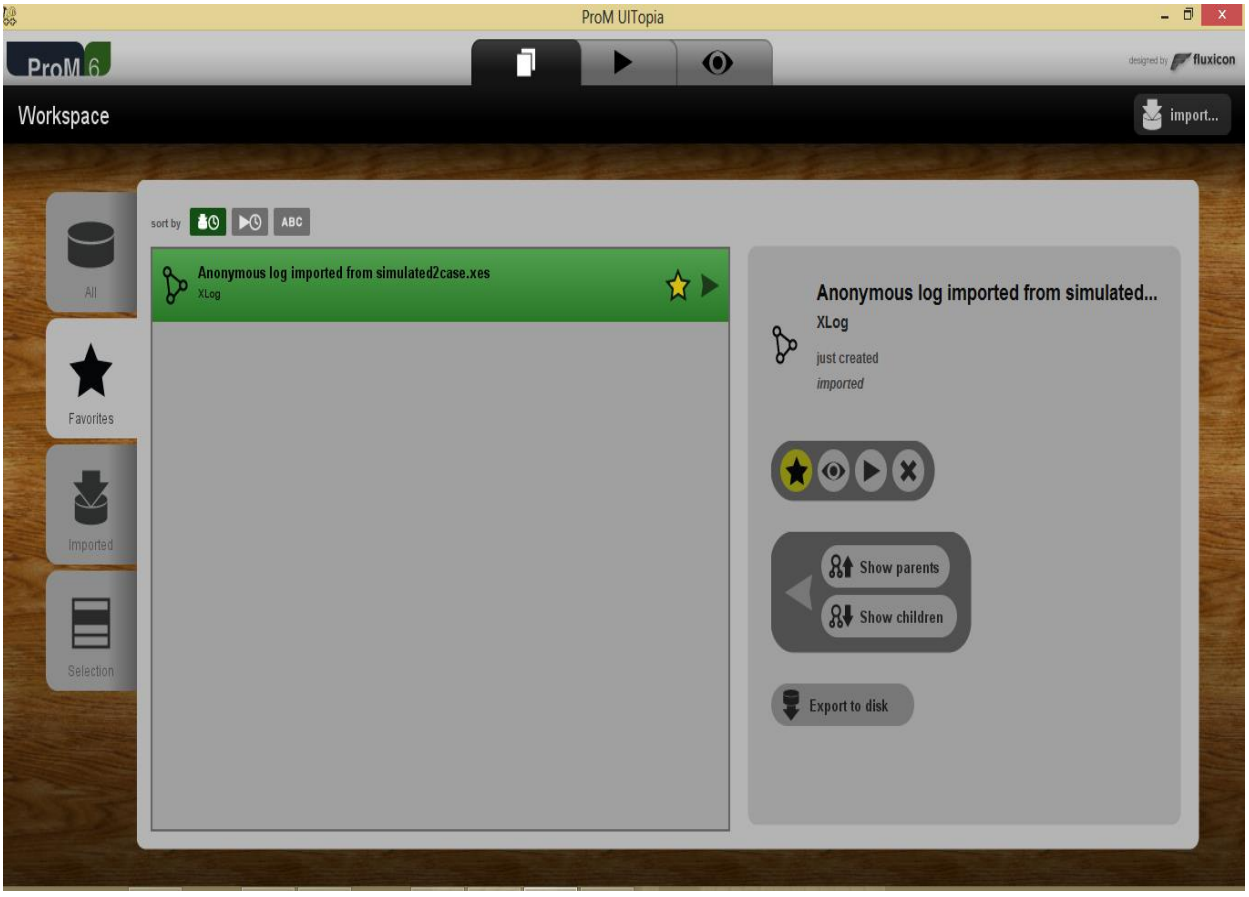


Prom

Prom is tool used for process mining techniques

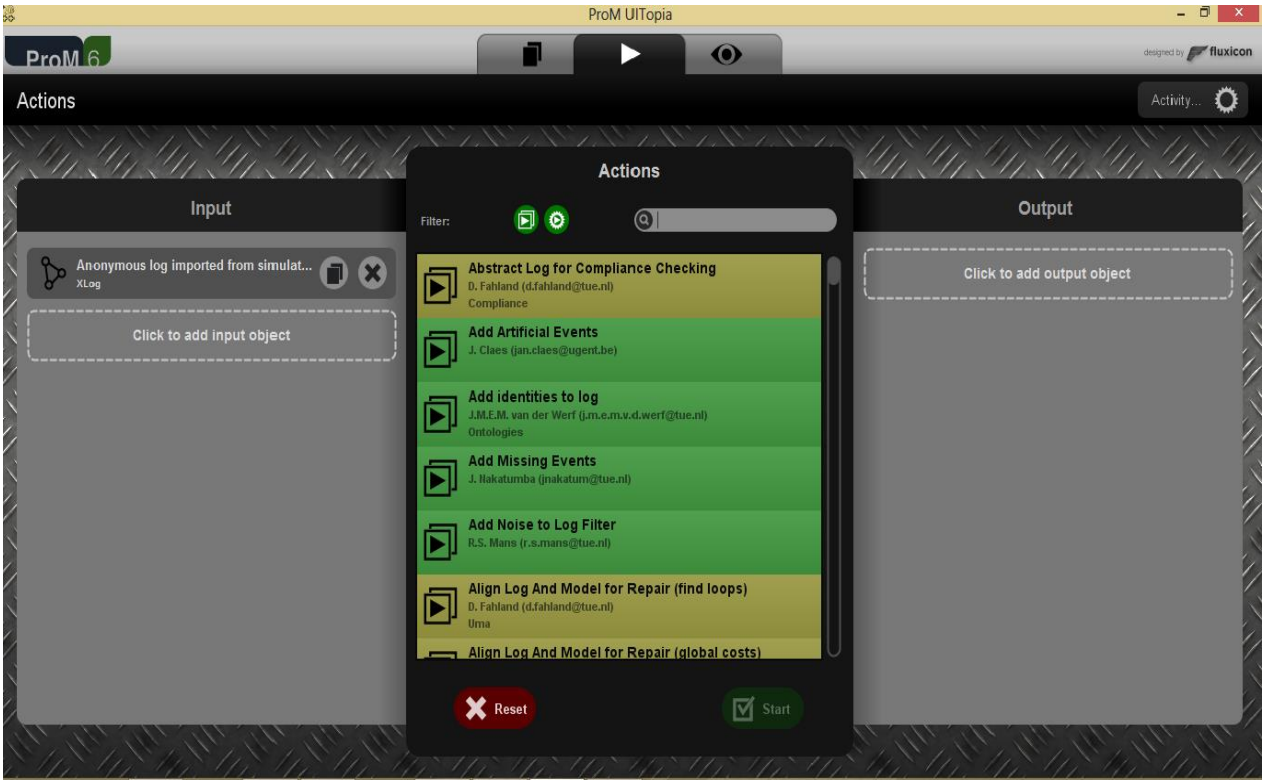
Alpha algorithm in prom

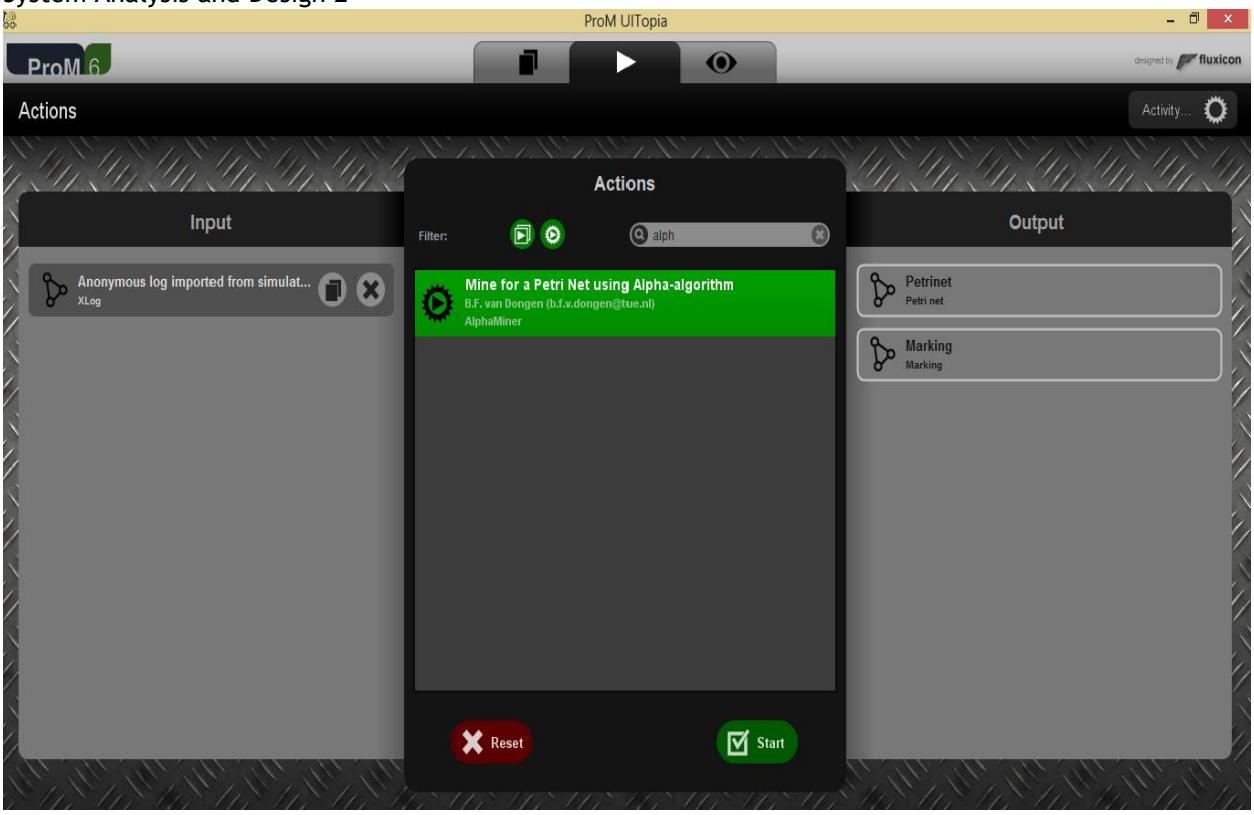


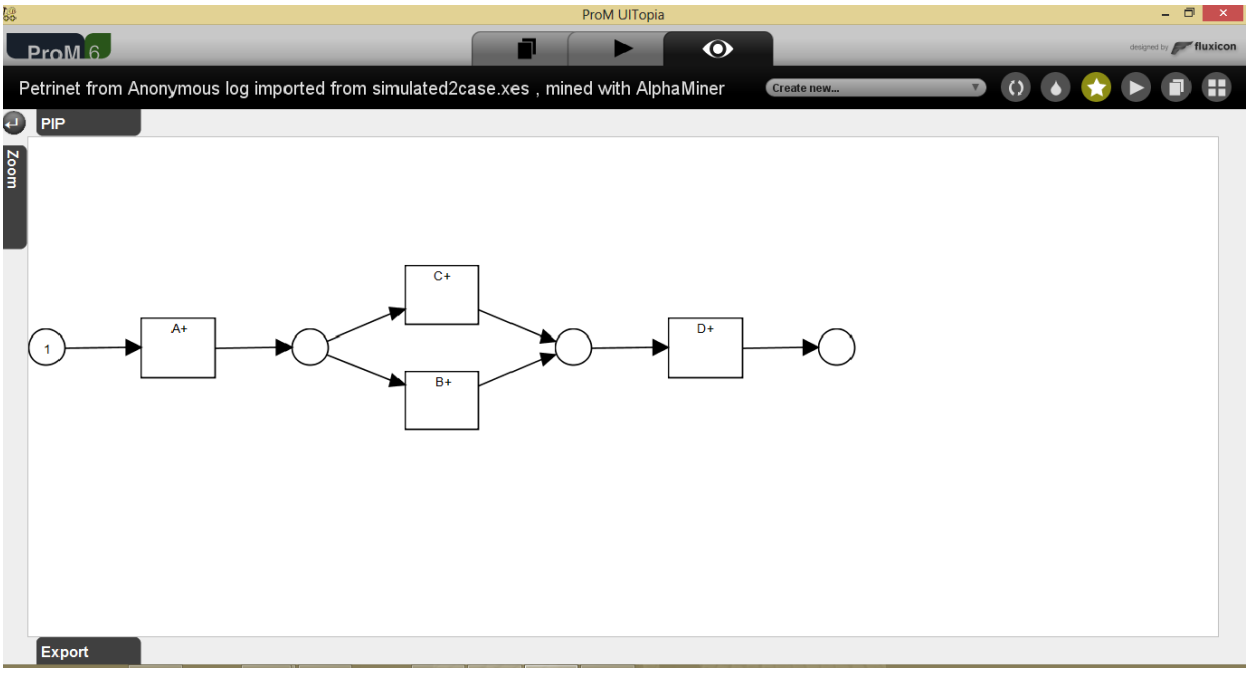


```xml
<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" >
  <trace>

    <event>

      <string key="concept:name" value="A"/>
    </event>
    <event>

      <string key="concept:name" value="B"/>
    </event>
    <event>

      <string key="concept:name" value="D"/>
    </event>
  </trace>
  <trace>
    <string key="concept:name" value="1"/>
    <event>

      <string key="concept:name" value="A"/>
    </event>
    <event>

      <string key="concept:name" value="C"/>
    </event>
    <event>
       <string key="concept:name" value="D"/>

    </event>
  </trace>
</log>
```

Applying alpha algorithm on imported event log




ProM UITopia
ProM 6
designed by fluxicon
Actions
Activity...
Actions
Input
Filter:
alph
Output
Anonymous log imported from simulat...
XLog
Mine for a Petri Net using Alpha-algorithm
B.F. van Dongen (b.f.v.dongen@tue.nl)
AlphaMiner
Petrinet
Petri net
Marking
Marking
Reset
Start


ProM UITopia
ProM 6
designed by fluxicon
Petrinet from Anonymous log imported from simulated2case.xes , mined with AlphaMiner
Create new...
PIP
Zoom
1
A+
C+
B+
D+
Export