Tutorial Process Mining: Beyond Business Intelligence

prof.dr.ir. Wil van der Aalst

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Thanks!

- Wil van der Aalst
- Peter van den Brand
- Boudewijn van Dongen
- Christian Günther
- Eric Verbeek
- Ana Karla Alves de Medeiros
- Anne Rozinat
- Minseok Song
- Ton Weijters
- Remco Dijkman
- Gianluigi Greco
- Antonella Guzzo
- Kristian Bisgaard Lassen
- Ronny Mans
- Jan Mendling
- Vladimir Rubin
- Kenny van Uden
- Irene Vanderfeesten
- Barbara Weber
- Lijie Wen
- Mercy Amiyo
- Carmen Bratosin
- Toon Calders
- Jorge Cardoso
- Ronald Crooy
- Florian Gottschalk
- Monique Jansen-Vullers
- Peter Khisa Wakholi
- Nicolas Knaak
- Sven Lambrechts
- Joyce Nakatumba
- Mariska Netjes
- Mykola Pechenizkiy
- Maja Pesic
- Hajo Reijers
- Stefanie Rinderle
- Domenico Saccà
- Helen Schonenberg
- Marc Voorhoeve
- Jianmin Wang
- Jan Martijn van der Werf
- Martin van Wingerden
- Jianhong Ye
- Huub de Beer
- Elena Casares
- Alina Chipaila
- Walid Gaaloul
- Martijn van Giessel
- Shaifali Gupta
- Thomas Hoffmann
- Peter Hornix
- René Kerstjens
- Ralf Kramer
- Wouter Kunst
- Laura Maruster
- Andriy Nikolov
- Adarsh Ramesh
- Jo Theunissen
- ...

Outline

1. Overview of Process Mining
2. Process Discovery: The Basics
3. Other Process Discovery Techniques
4. Balancing Between Overfitting and Underfitting
5. Discovering Other Perspectives
6. Conformance Checking and Extension
7. ProM Tool
8. Conclusion

Overview of Process Mining

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Role of models

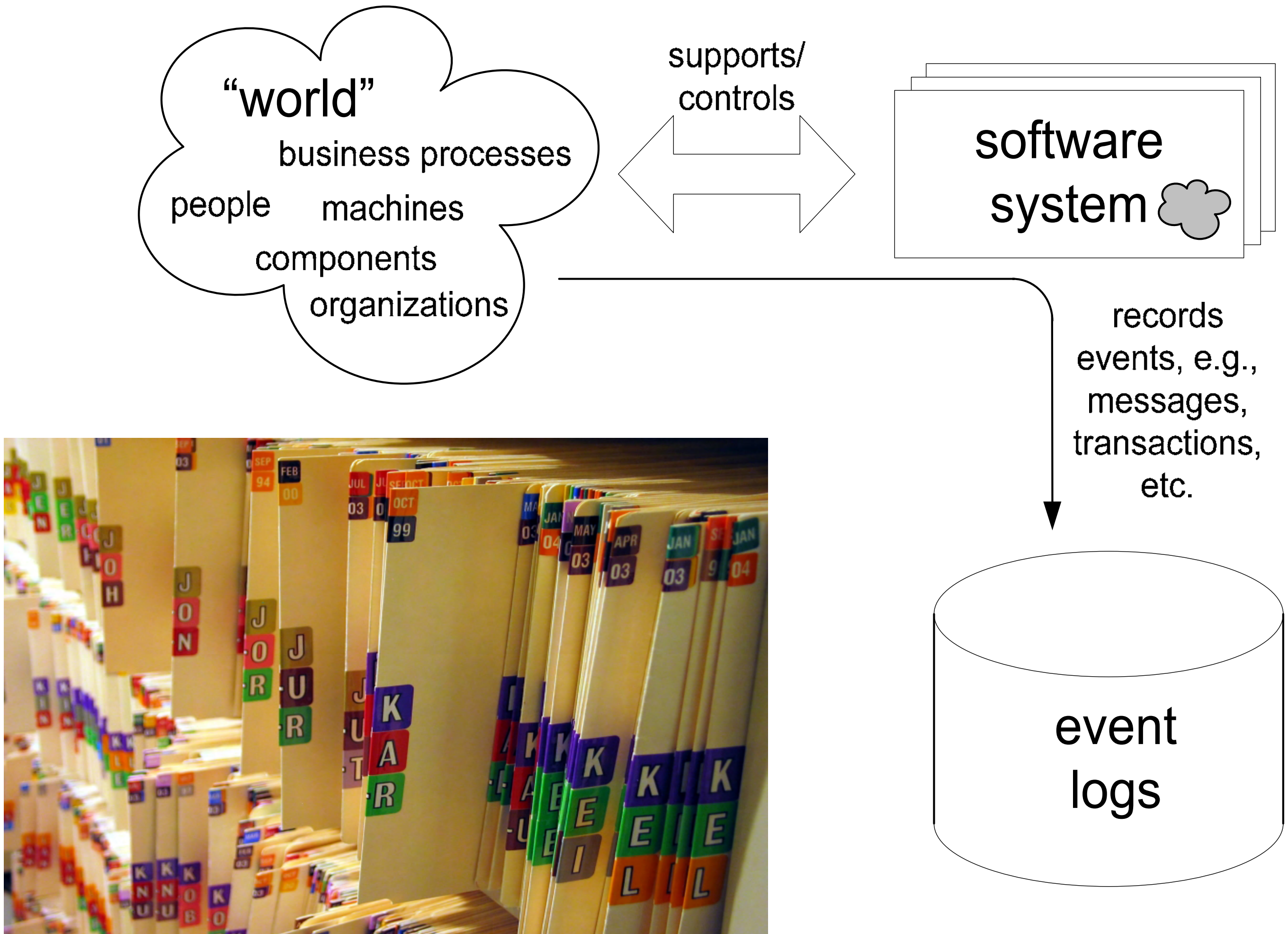
"world"
business process
people
machines
components
organizations

verification

real world

powerpoint reality

Event logs are a reflection of reality



Examples:

IBM

staffware. THE POWER OF PROCESS

Pallas Athena PROCESS MANAGEMENT

FILENET An IBM Company

océ

SAP

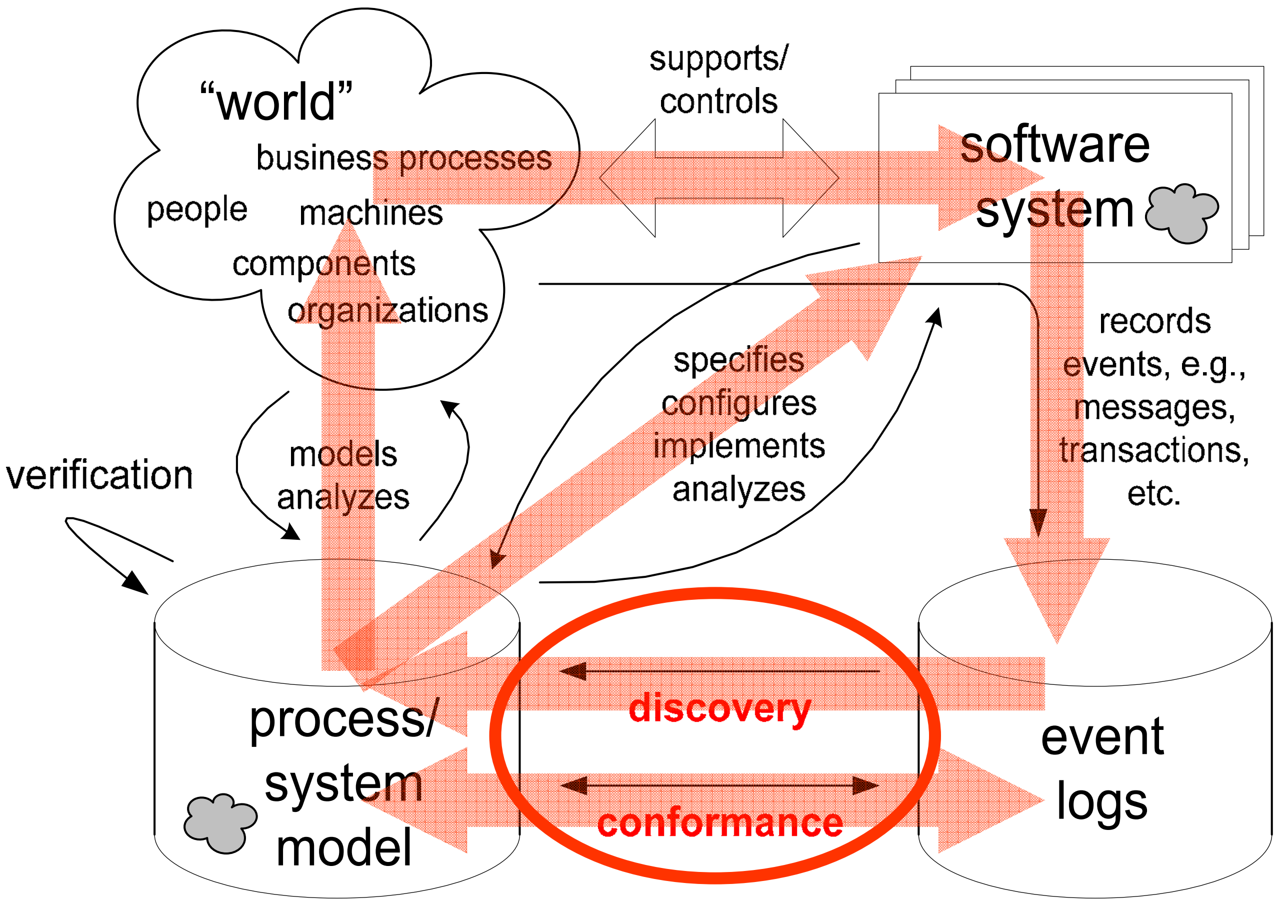
PHILIPS sense and simplicity

ORACLE

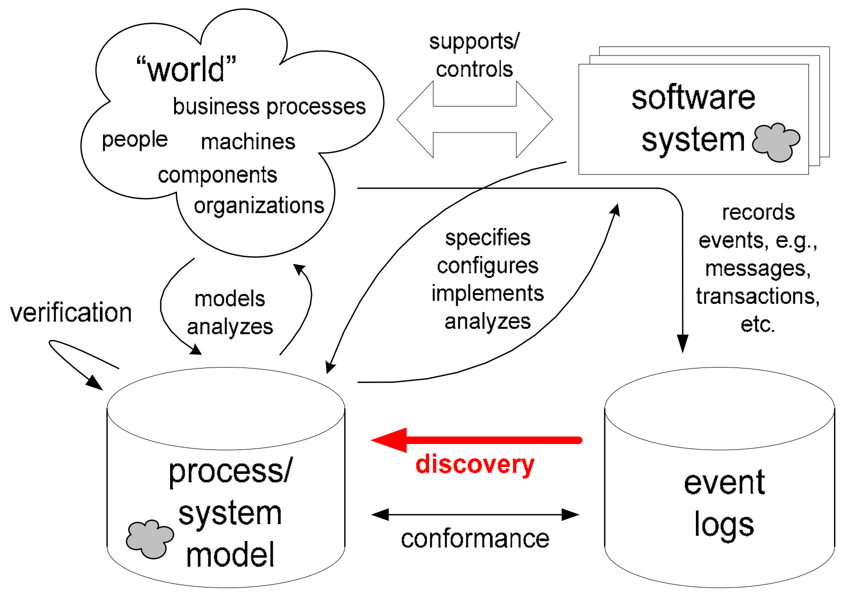
ASML

WebSphere. software

Process mining: Linking events to models

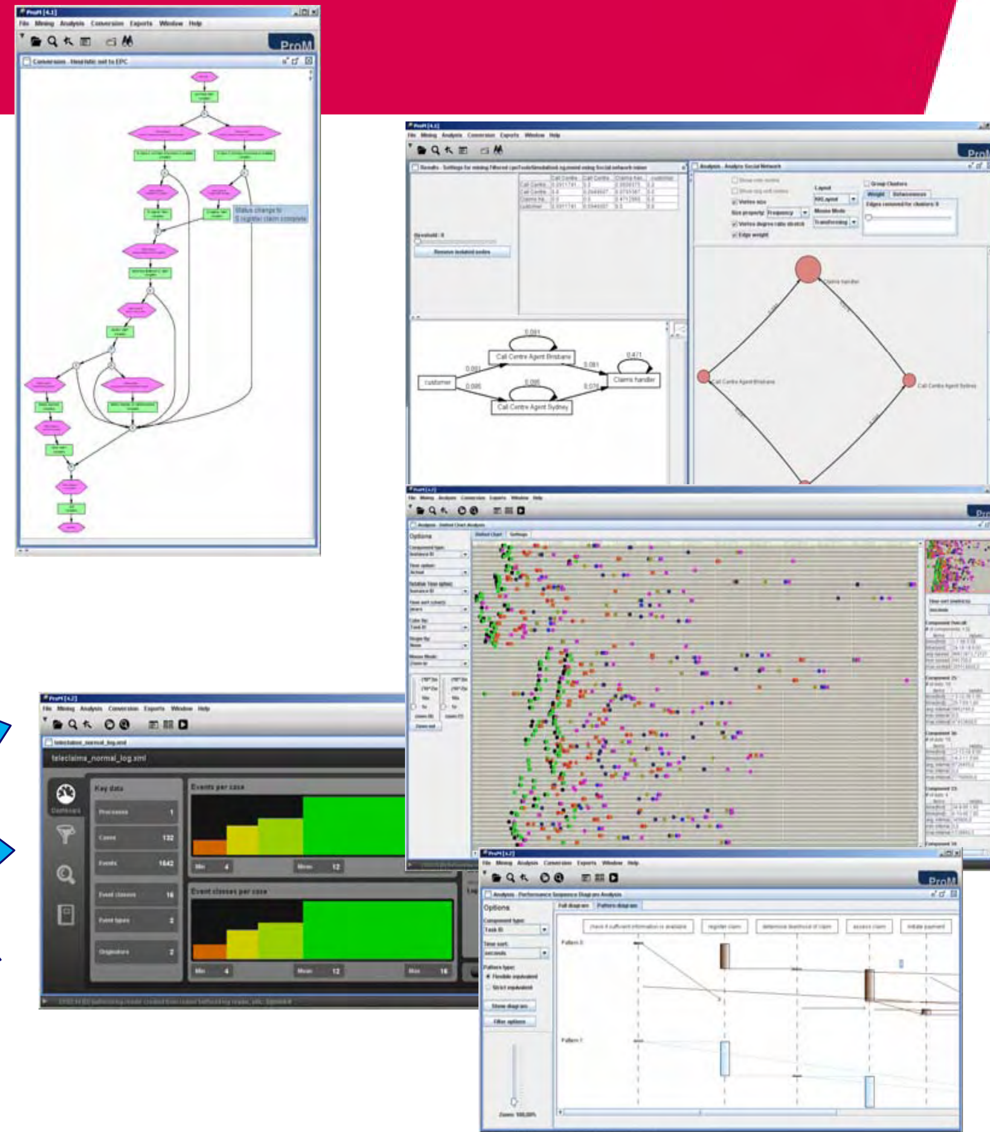


Discovery



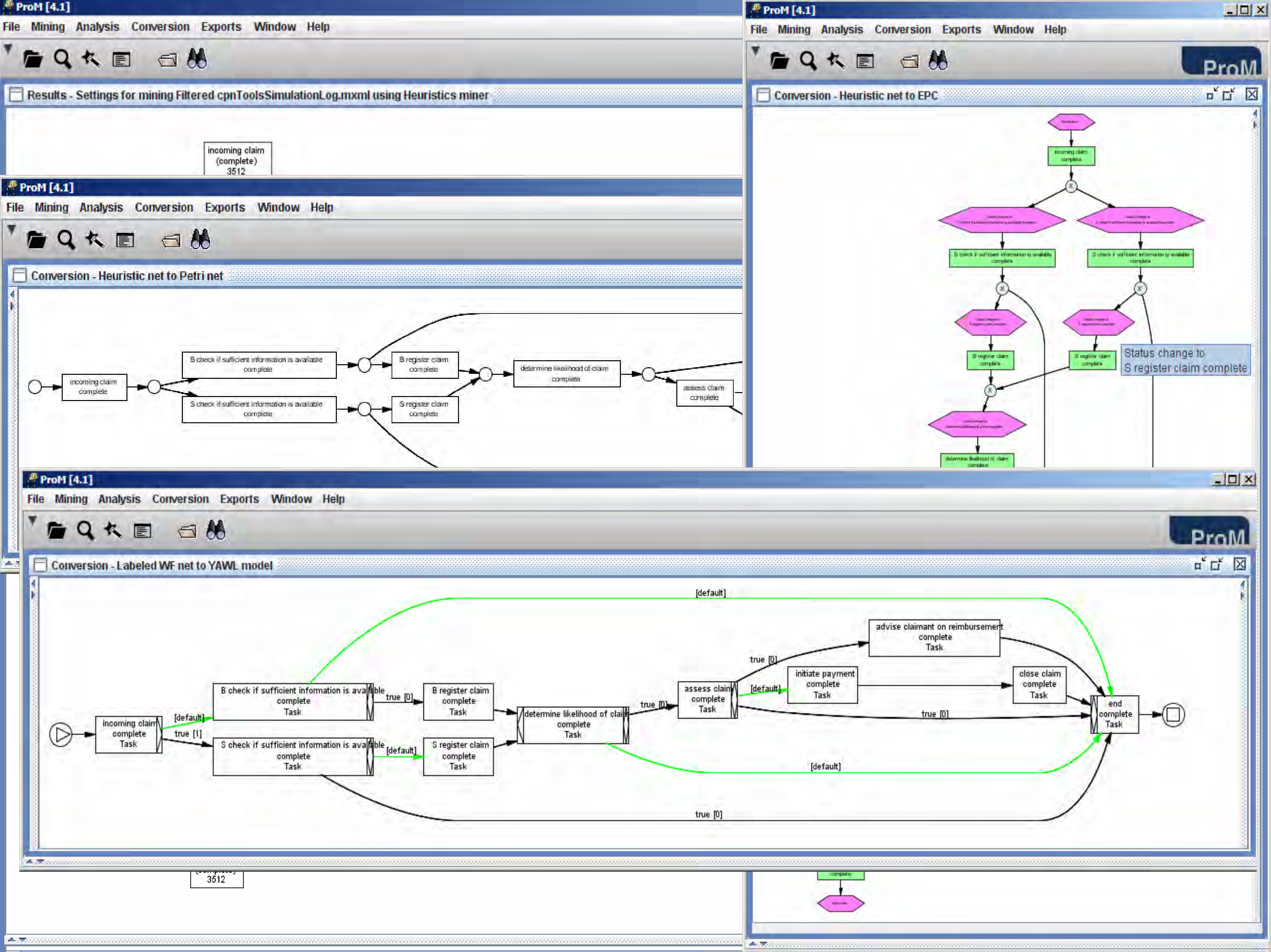
"world"
business processes
people
machines
components
organizations
supports/
controls
software
system
records
events, e.g.,
messages,
transactions,
etc.
specifies
configures
implements
analyzes
models
analyzes
verification
process/
system
model
discovery
conformance
event
logs

MXML Log
- **instances:** 3512
- **audit trail entries:** 46138

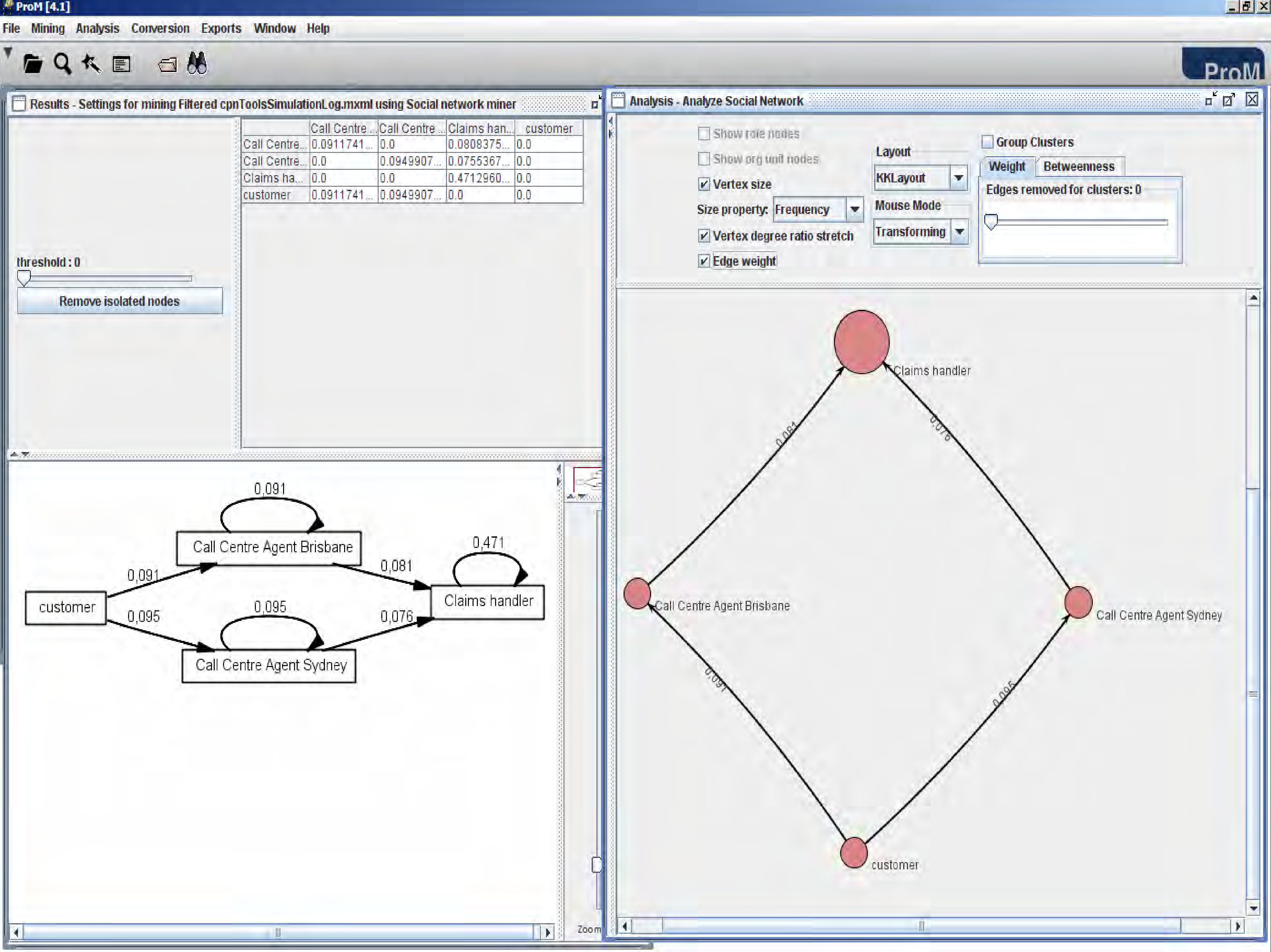


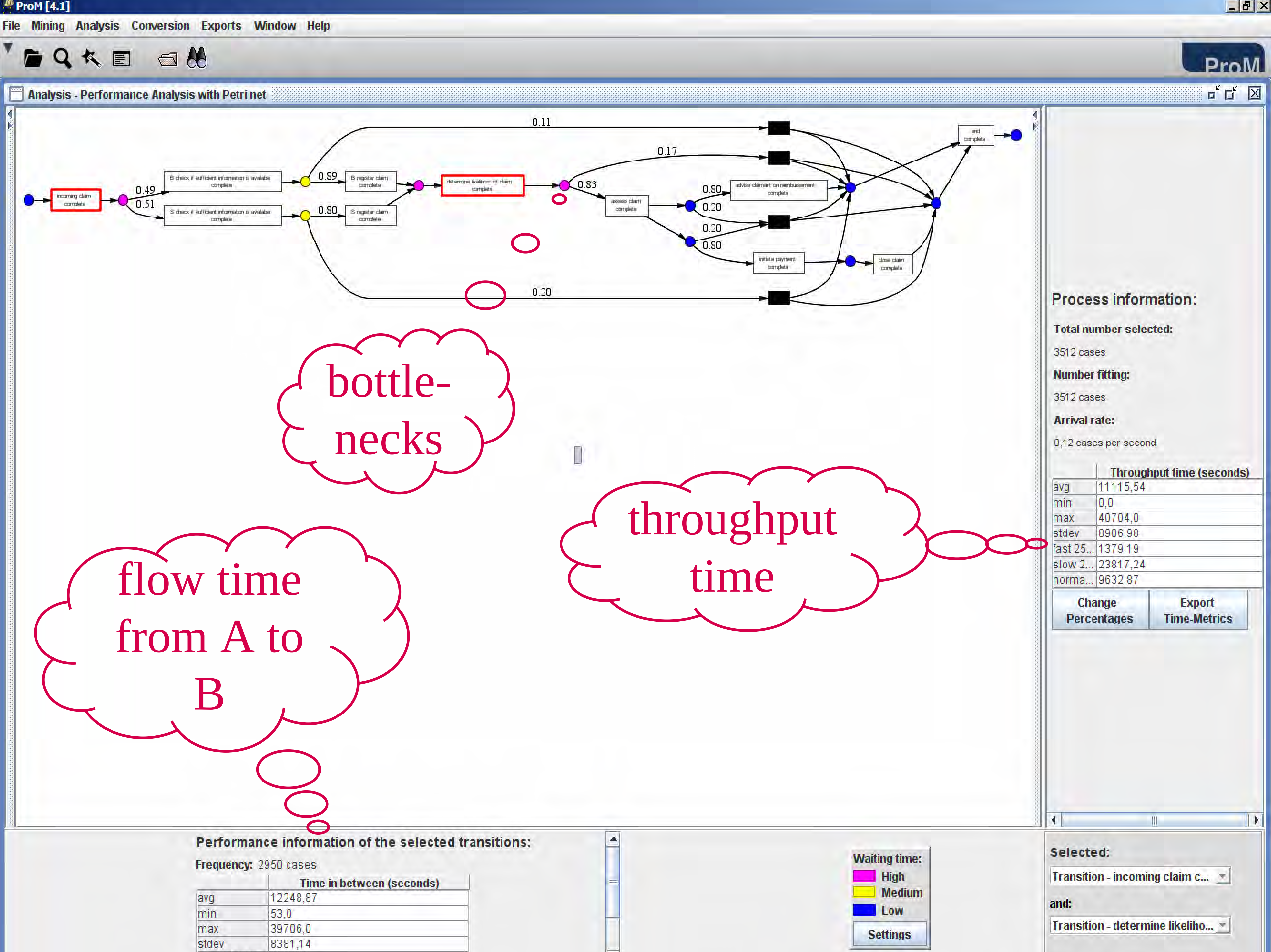
ProM supports +40 types of model discovery!


ProM [4.2]
File Mining Analysis Conversion Exports Window Help
ProM
Analysis - Dotted Chart Analysis
Options
Component type:
Instance ID
Time option:
Relative(Time)
Relative Time option:
Instance ID
Time sort (chart):
years
Color By:
Task ID
Shape By:
None
Mouse Mode:
Zoom in
(10^3)x
(10^2)x
10x
1x
zoom (X)
zoom (Y)
Zoom out
Dotted Chart
Settings
0days:0:0:0
365days:0:0:0
730days:0:0:0
1095days:0:0:0
1460days:0:0:0
1825days:0:0:0
2190days:0:0:0
2555days:0:0:0
2920days:0:0:0
3285days:0:0:0
3650days:0:0:0
4015days:0:0:0
49 | initiate payment | Claims handler | Tue Jun 25 01:00:00 CEST 2013 | complete | Position:2.033856E8
49 | close claim | Claims handler | Tue Jun 25 01:00:00 CEST 2013 | start | Position:2.033856E8
49 | close claim | Claims handler | Wed Jul 03 01:00:00 CEST 2013 | complete | Position:2.040768E8
49 | end | Claims handler | Wed Jul 03 01:00:00 CEST 2013 | start | Position:2.040768E8
49 | end | Claims handler | Wed Jul 03 01:00:00 CEST 2013 | complete | Position:2.040768E8
Time sort (metrics):
seconds
Component Overall:
# of components: 132
items values
time(first) 0days:0:0:0
time(end) 4462days:0:0:0
avg spread 96612872,72727
min spread 691200,0
max spread 385516800,0
Component 35:
# of dots: 16
items values
time(first) 0days:0:0:0
time(end) 1016days:0:0:0
avg. interval 5852160,0
min interval 0,0
max interval 47433600,0
Component 36:
# of dots: 16
items values
time(first) 0days:0:0:0
time(end) 1515days:0:0:0
avg. interval 8726400,0
min interval 0,0
max interval 77760000,0
Component 33:
# of dots: 4
items values
time(first) 0days:0:0:0
time(end) 12days:0:0:0
avg. interval 345600,0
min interval 0,0
max interval 1036800,0
Component 34:
# of dots: 16
items values
time(first) 0days:0:0:0
time(end) 2482days:0:0:0
avg. interval 14296320,0
min interval 0,0
max interval 146361600,0
Component 39:
# of dots: 16
items values
time(first) 0days:0:0:0
time(end) 537days:0:0:0
21:59:19 [D] Buffered log reader created from reader buffered log reader, pitk.: [l@b8dfdf



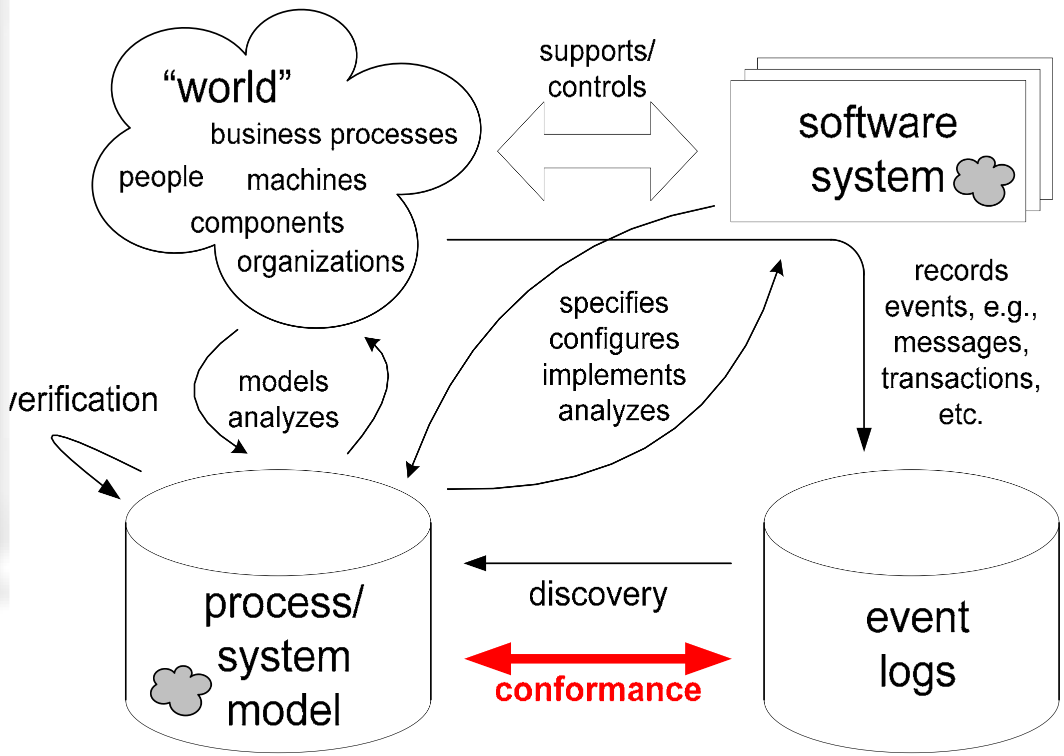
ProM [4.1]
File Mining Analysis Conversion Exports Window Help
Results - Settings for mining Filtered cpnToolsSimulationLog.mxml using Heuristics miner
incoming claim (complete) 3512
Conversion - Heuristic net to Petri net
incoming claim complete
B check if sufficient information is available complete
S check if sufficient information is available complete
B register claim complete
S register claim complete
determine likelihood of claim complete
assess claim complete
Conversion - Heuristic net to EPC
Status change to S register claim complete
Conversion - Labeled WF net to YAWL model
[default]
advise claimant on reimbursement complete Task
true [0]
initiate payment complete Task
close claim complete Task
incoming claim complete Task
B check if sufficient information is available complete Task
S check if sufficient information is available complete Task
B register claim complete Task
S register claim complete Task
determine likelihood of claim complete Task
assess claim complete Task
end complete Task
true [1]
3512

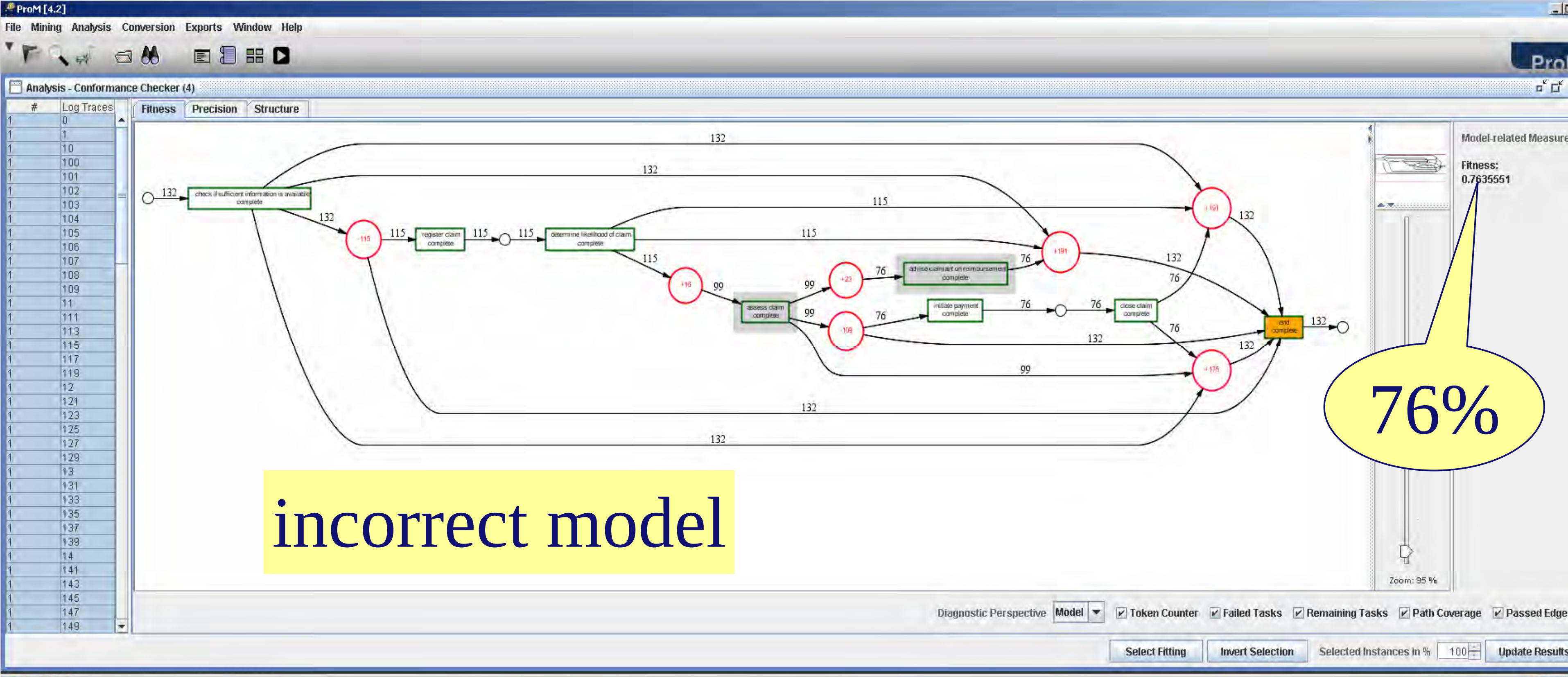

ProM [4.1]
File Mining Analysis Conversion Exports Window Help
ProM
Results - Settings for mining Filtered cpnToolsSimulationLog.mxml using Social network miner
threshold : 0
Remove isolated nodes
Call Centre... Call Centre... Claims han... customer
Call Centre... 0.0911741... 0.0 0.0808375... 0.0
Call Centre... 0.0 0.0949907... 0.0755367... 0.0
Claims ha... 0.0 0.0 0.4712960... 0.0
customer 0.0911741... 0.0949907... 0.0 0.0
customer
Call Centre Agent Brisbane
Call Centre Agent Sydney
Claims handler
0,091
0,095
0,081
0,076
0,471
Zoom
Analysis - Analyze Social Network
Show role nodes
Show org unit nodes
Vertex size
Size property: Frequency
Vertex degree ratio stretch
Edge weight
Layout
KKLayout
Mouse Mode
Transforming
Group Clusters
Weight
Betweenness
Edges removed for clusters: 0
Claims handler
Call Centre Agent Brisbane
Call Centre Agent Sydney
customer

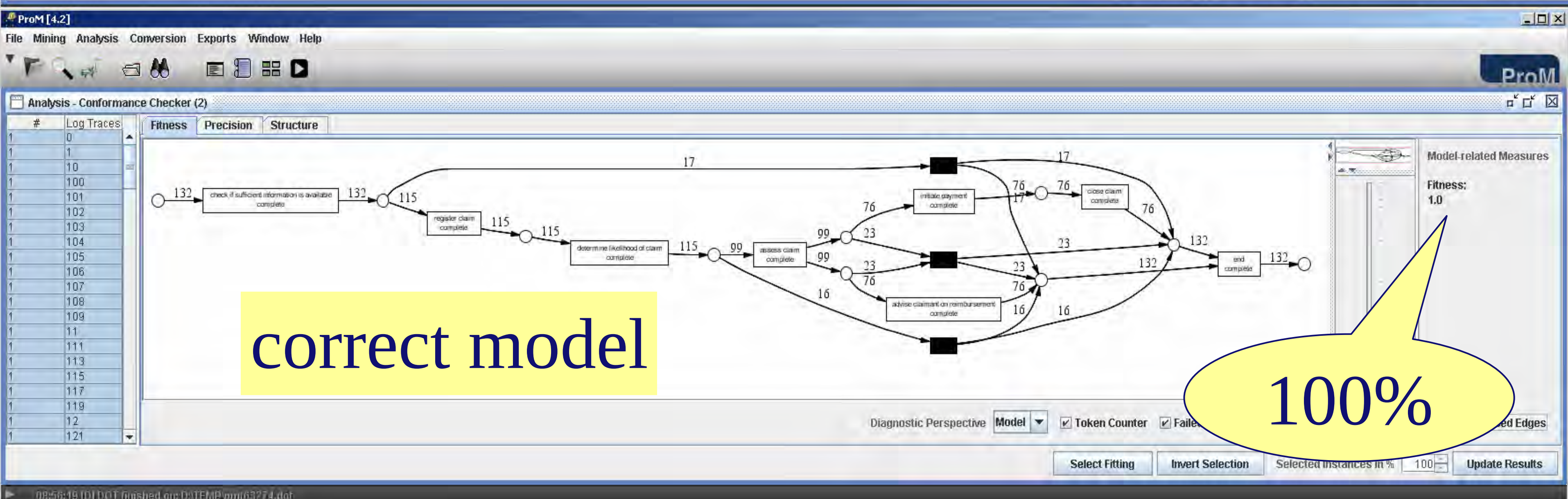


ProM [4.1]
File Mining Analysis Conversion Exports Window Help
Analysis - Performance Analysis with Petri net
0.11
0.17
0.49
0.51
0.89
0.80
0.83
0.80
0.20
0.20
0.80
0.20
bottle-necks
flow time from A to B
throughput time
Process information:
Total number selected:
3512 cases
Number fitting:
3512 cases
Arrival rate:
0,12 cases per second
Throughput time (seconds)
avg 11115,54
min 0,0
max 40704,0
stdev 8906,98
fast 25... 1379,19
slow 2... 23817,24
norma... 9632,87
Change Percentages
Export Time-Metrics
Performance information of the selected transitions:
Frequency: 2950 cases
Time in between (seconds)
avg 12248,87
min 53,0
max 39706,0
stdev 8381,14
Waiting time:
High
Medium
Low
Settings
Selected:
Transition - incoming claim c...
and:
Transition - determine likeliho...

Conformance Checking




ProM [4.2]
File Mining Analysis Conversion Exports Window Help
Analysis - Conformance Checker (4)
Fitness Precision Structure
Log Traces
Model-related Measure
Fitness:
0.7635551
Zoom: 95 %
Diagnostic Perspective Model Token Counter Failed Tasks Remaining Tasks Path Coverage Passed Edge
Select Fitting Invert Selection Selected Instances in % 100 Update Results
incorrect model
76%


ProM [4.2]
File Mining Analysis Conversion Exports Window Help
Analysis - Conformance Checker (2)
Fitness Precision Structure
Log Traces
Model-related Measures
Fitness:
1.0
Diagnostic Perspective Model Token Counter
Select Fitting Invert Selection Update Results
correct model
100%

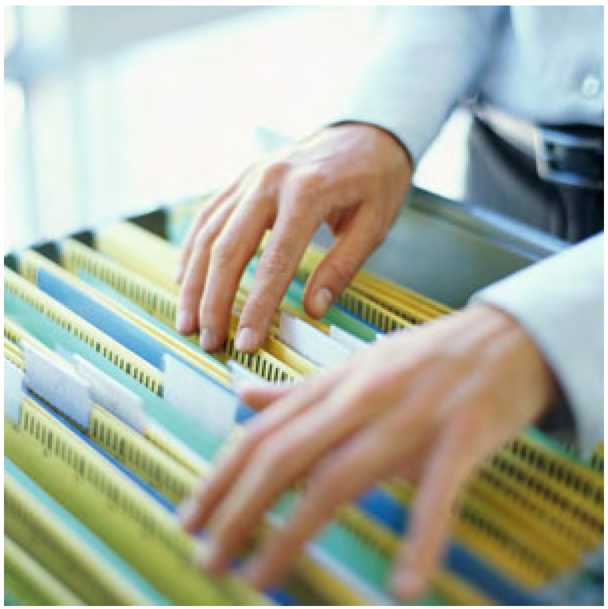

ProM [4.2]
Exposé ...select the frame you want to bring forward
Results - Fuzzy Miner on Selected EPC
Fuzzy Miner
Graph
Unary metrics
Binary metrics
Concurrency filter
Edge filter
Node filter
Significance cutoff
Results - Multi-phase Macro Plugin on Selected Petri net
ProM
Zoom: 50 %
+/- 250 plug-ins
reviewslog_with_fewer_errors.xml
reviewslog_with_fewer_errors.xml
Results - Alpha algorithm plugin on Raw reviewslog_with_fewer_errors.xml (unfiltered)
Zoom: 66 %
Edit log relations
Dashboard
Key data
Processes 1
Cases 100
Events 2297
Event classes 20
Event types 2
Originators 10
Events per case
Min 11
Mean 22
Max 50
Event classes per case
Min 11
Mean 14
Max 15
Log info
Source
CPN Tools
Source program
CPN Tools
Start date
2006-01-01 00:00:00
End date
2008-05-05 01:00:00
Description
Log extracted from CPN Tools
start analyzing this log

Process Discovery: The Basics

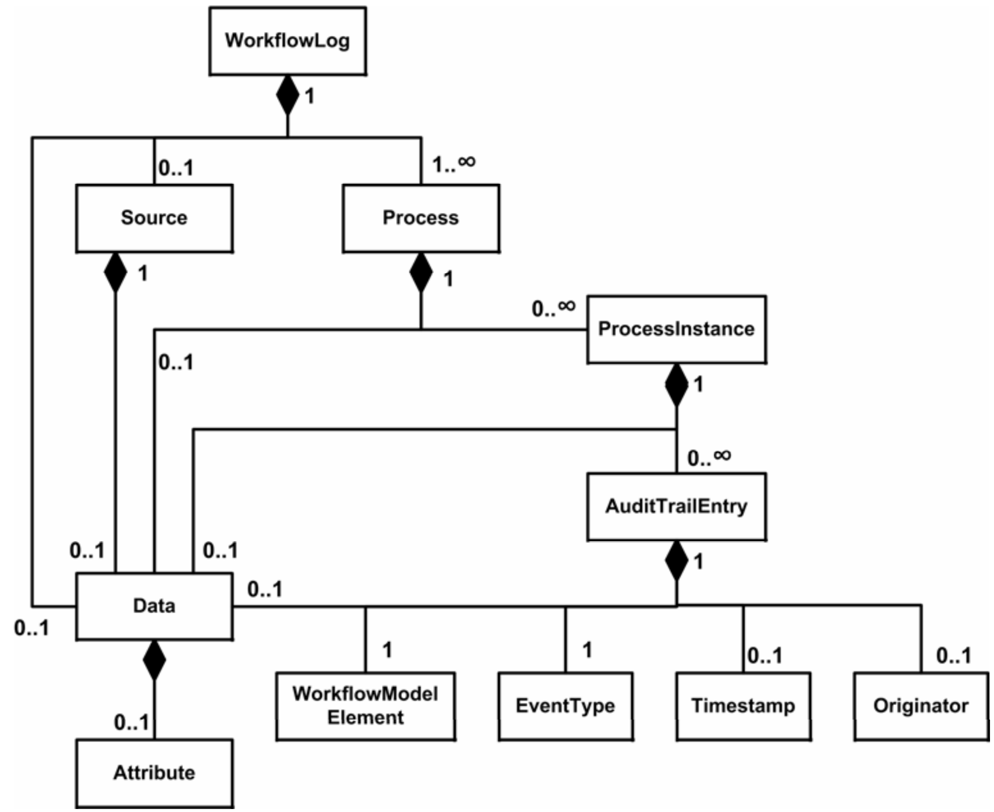
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Starting point: event logs



event logs, audit trails, databases, message logs, etc.



unified event log (MXML)

MXML

Event log:

- processes
  - process instances
    - events

Per event:

- activity name
- (event type)
- (originator)
- (timestamp)
- (data)

</ProcessInstance>
- <ProcessInstance id="51" description="">
- <AuditTrailEntry>
<WorkflowModelElement>invite reviewers</WorkflowModelElen
<EventType>start</EventType>
<Timestamp>2006-08-28T00:00:00.000+01:00</Timestamp>
<Originator>Mike</Originator>
</AuditTrailEntry>
- <AuditTrailEntry>
<WorkflowModelElement>invite reviewers</WorkflowModelElen
<EventType>complete</EventType>
<Timestamp>2006-08-31T00:00:00.000+01:00</Timestamp
<Originator>Mike</Originator>
</AuditTrailEntry>
- <AuditTrailEntry>
- <Data>
<Attribute name="result">reject</Attribute>
</Data>
<WorkflowModelElement>get review 3</WorkflowModelEleme
<EventType>complete</EventType>
<Timestamp>2006-09-02T00:00:00.000+01:00</Timestam
<Originator>Mary</Originator>
</AuditTrailEntry>
- <AuditTrailEntry>
<WorkflowModelElement>time-out 1</WorkflowModelElement>
<EventType>complete</EventType>
<Timestamp>2006-09-03T00:00:00.000+01:00</Timestamp>
<Originator />
</AuditTrailEntry>
- <AuditTrailEntry>

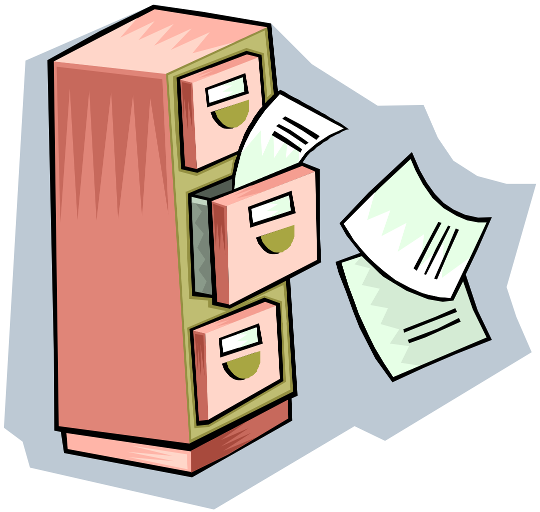
attributes of an event

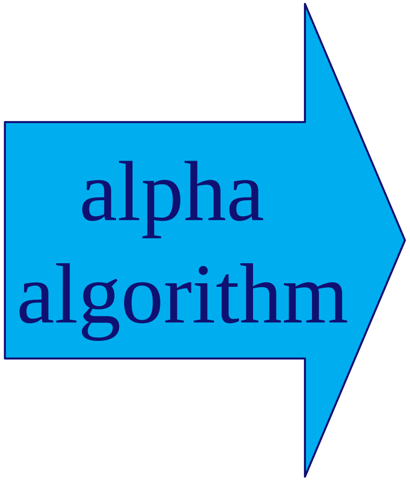
end of activity

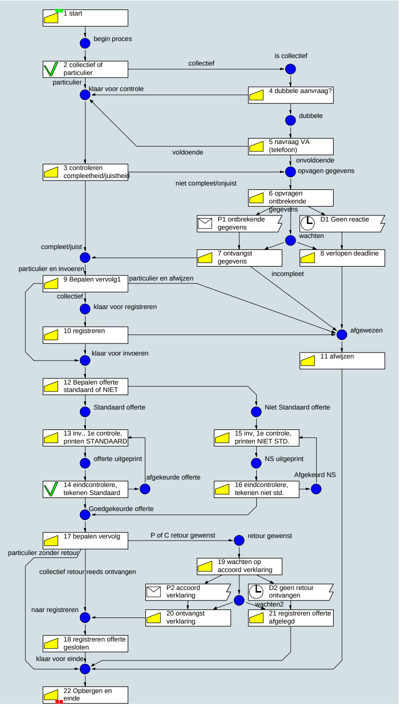
activity

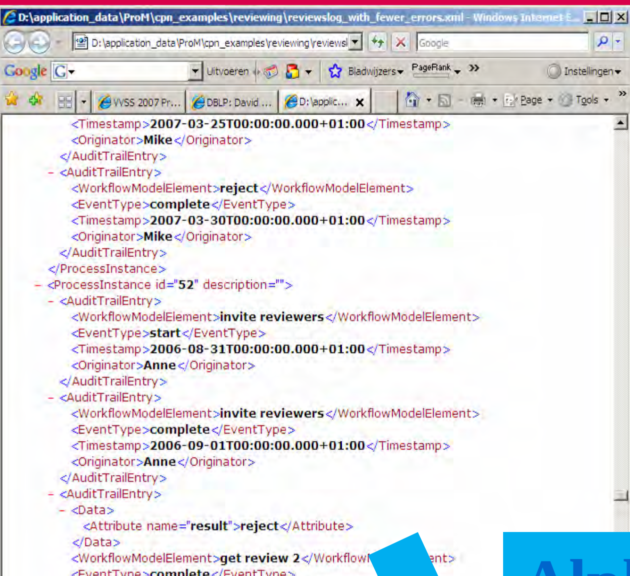
instance

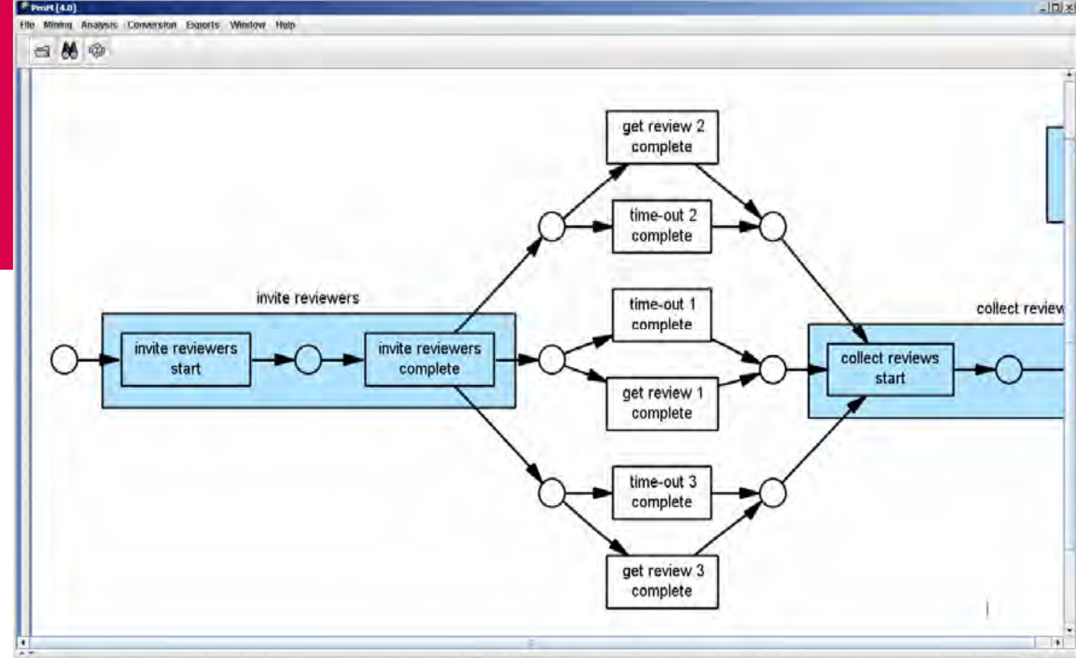
Process Mining: The alpha algorithm



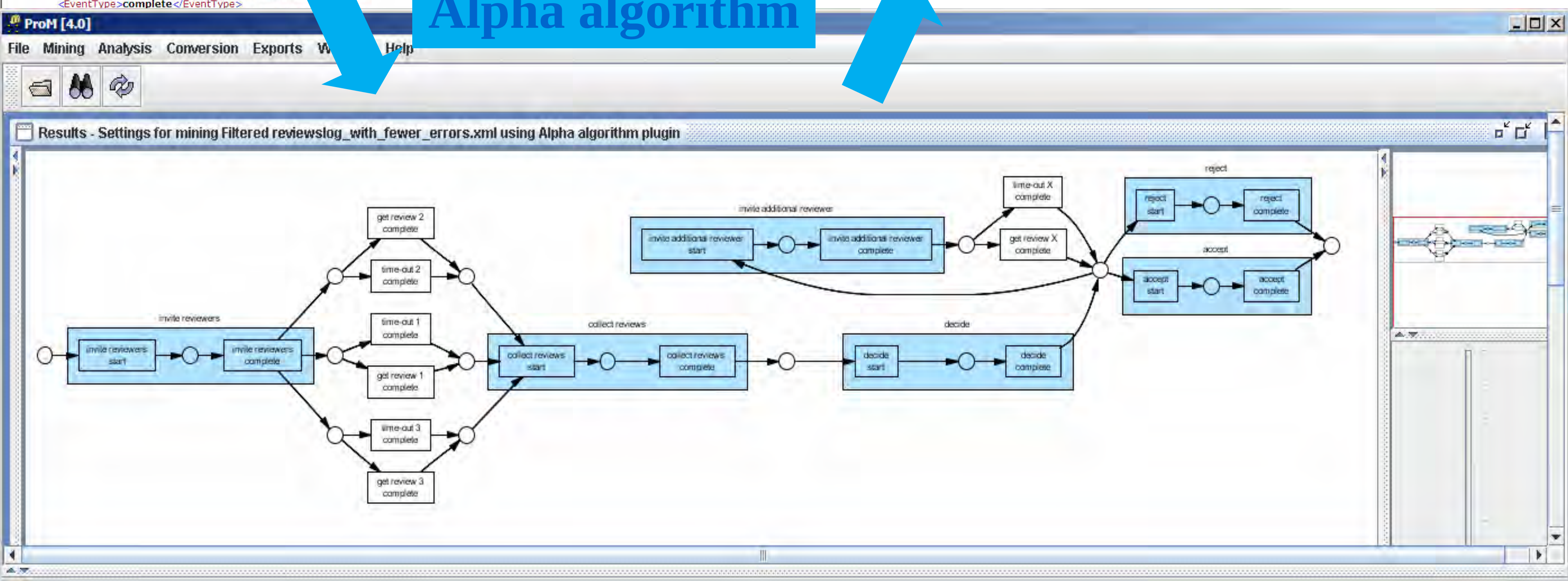




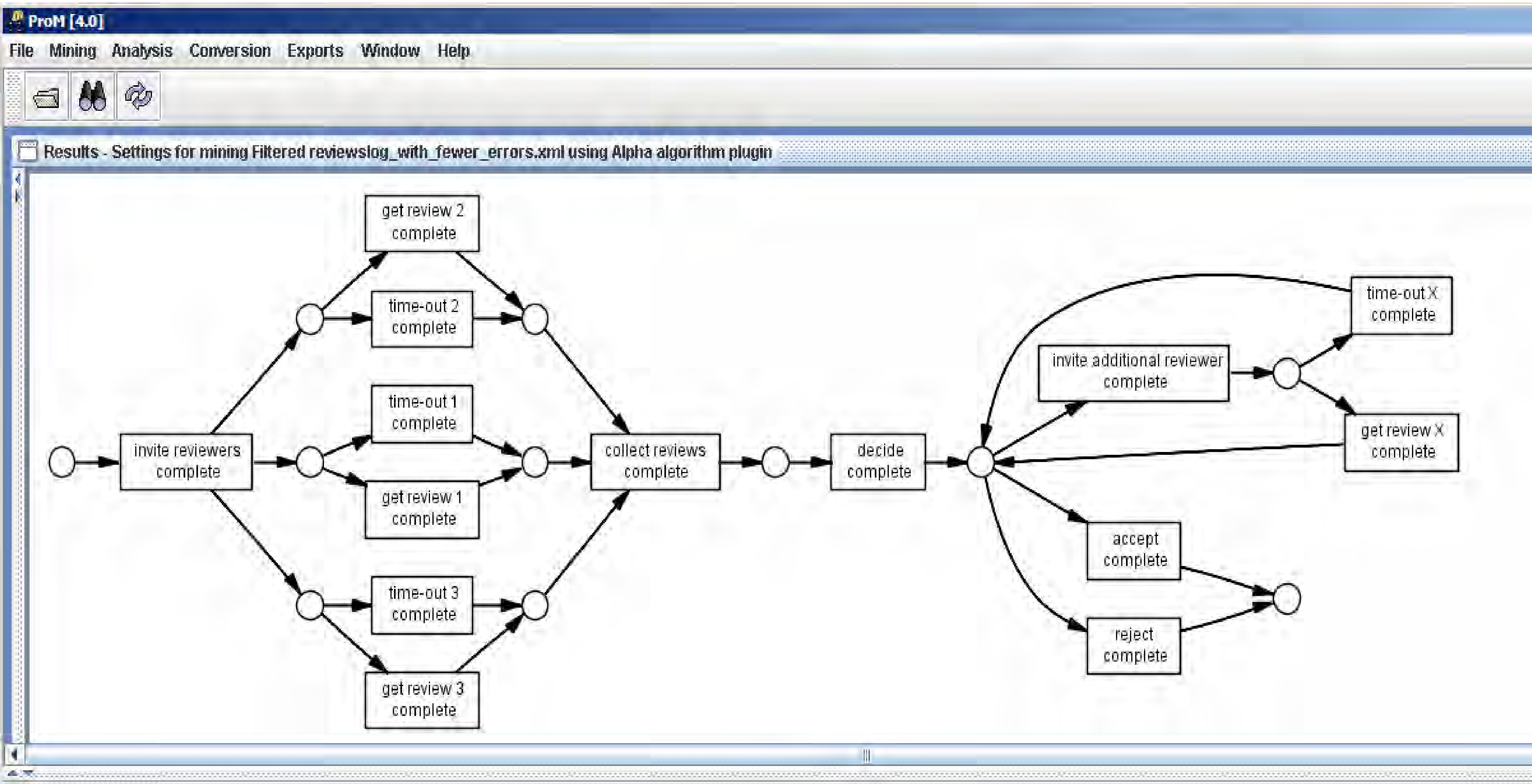




Alpha algorithm



Without transactional information (just completes)



Example log

- Minimal information in log: case id's and task id's.
- Additional information: event type, time, resources, and data.
- Sequences:
  - 1: ABCD
  - 2: ACBD
  - 3: ABCD
  - 4: ACBD
  - 5: EF
- So this log there are three possible sequences:
  - ABCD
  - ACBD
  - EF

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

>,→,||,# relations

- Direct succession: **x>y iff for some case x is directly followed by y.**
- Causality: **x→y iff x>y and not y>x.**
- Parallel: **x||y iff x>y and y>x**
- Choice: **x#y iff not x>y and not y>x.**

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

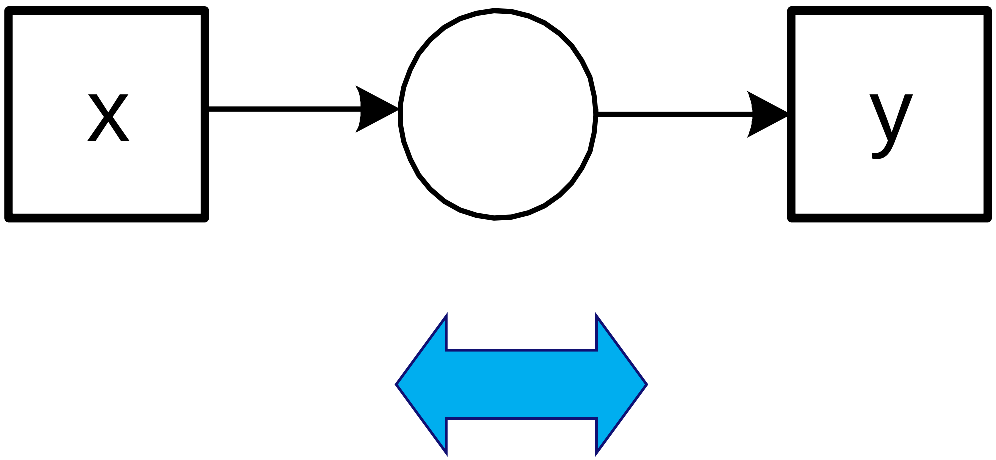
ABCD
ACBD
EF

A>B
A>C
B>C
B>D
C>B
C>D
E>F

A→B
A→C
B→D
C→D
E→F

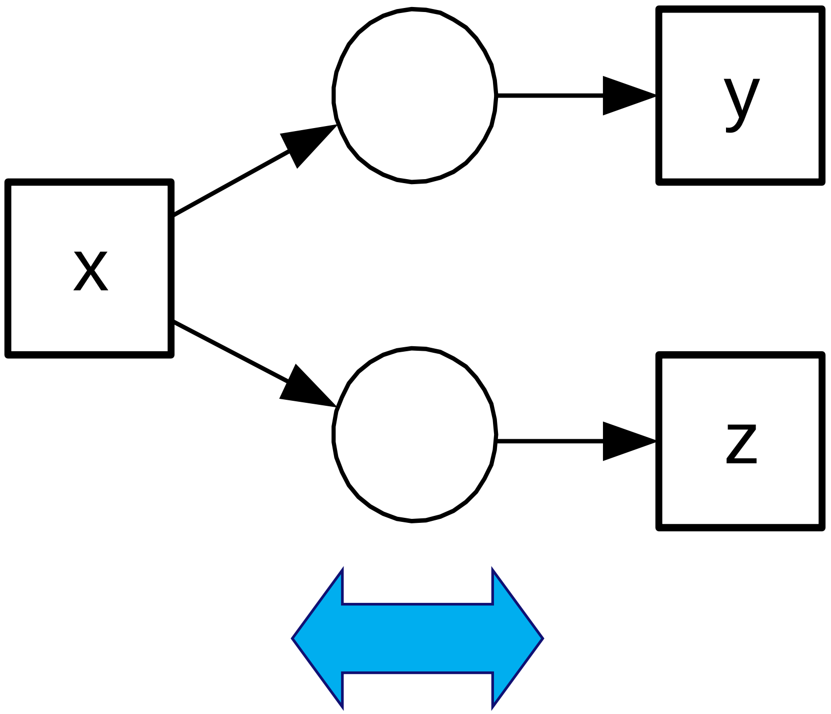
B||C
C||B

Basic idea (1)



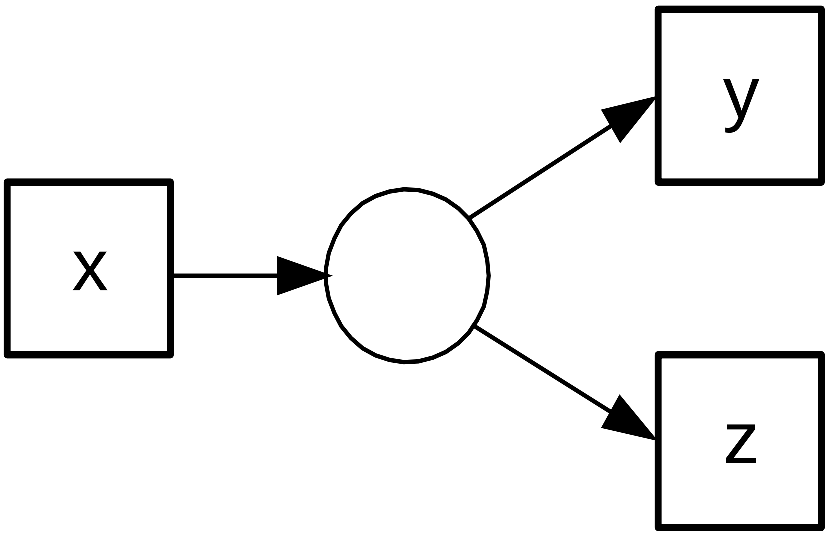
$$x \rightarrow y$$

Basic idea (2)



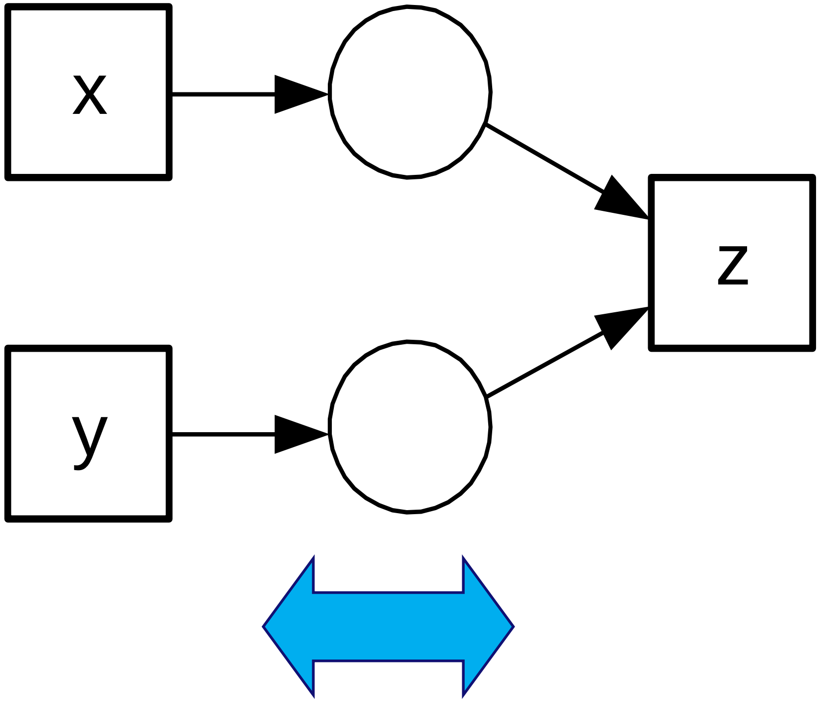
$x \rightarrow y$, $x \rightarrow z$, and $y || z$

Basic idea (3)



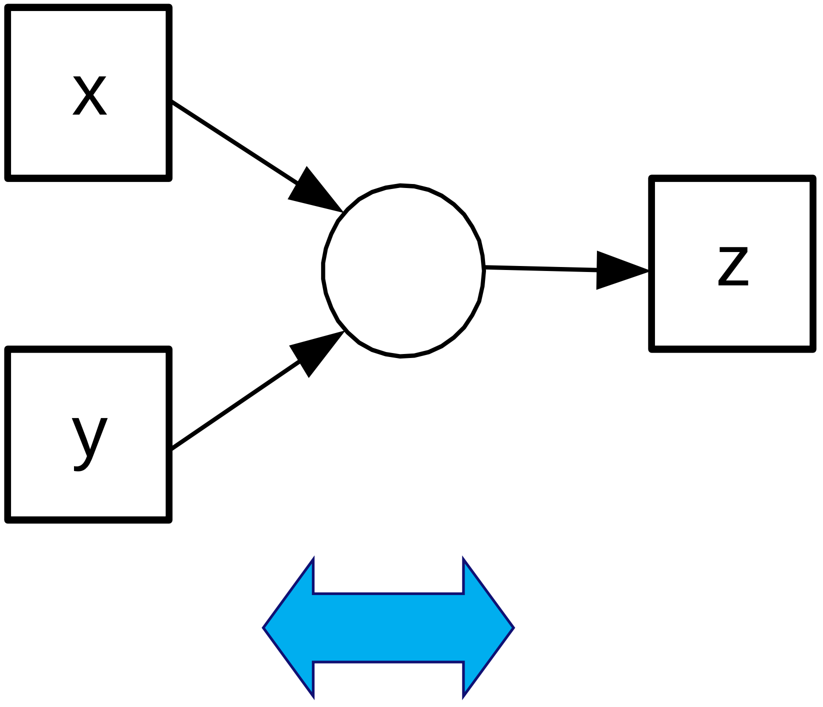
$x \rightarrow y$, $x \rightarrow z$, and $y \# z$

Basic idea (4)



$x \rightarrow z$, $y \rightarrow z$, and $x || y$

Basic idea (5)



x→z, y→z, and x#y

It is not that simple: Basic alpha algorithm

Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \;\wedge\; \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \;\wedge\; \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

Example revisited

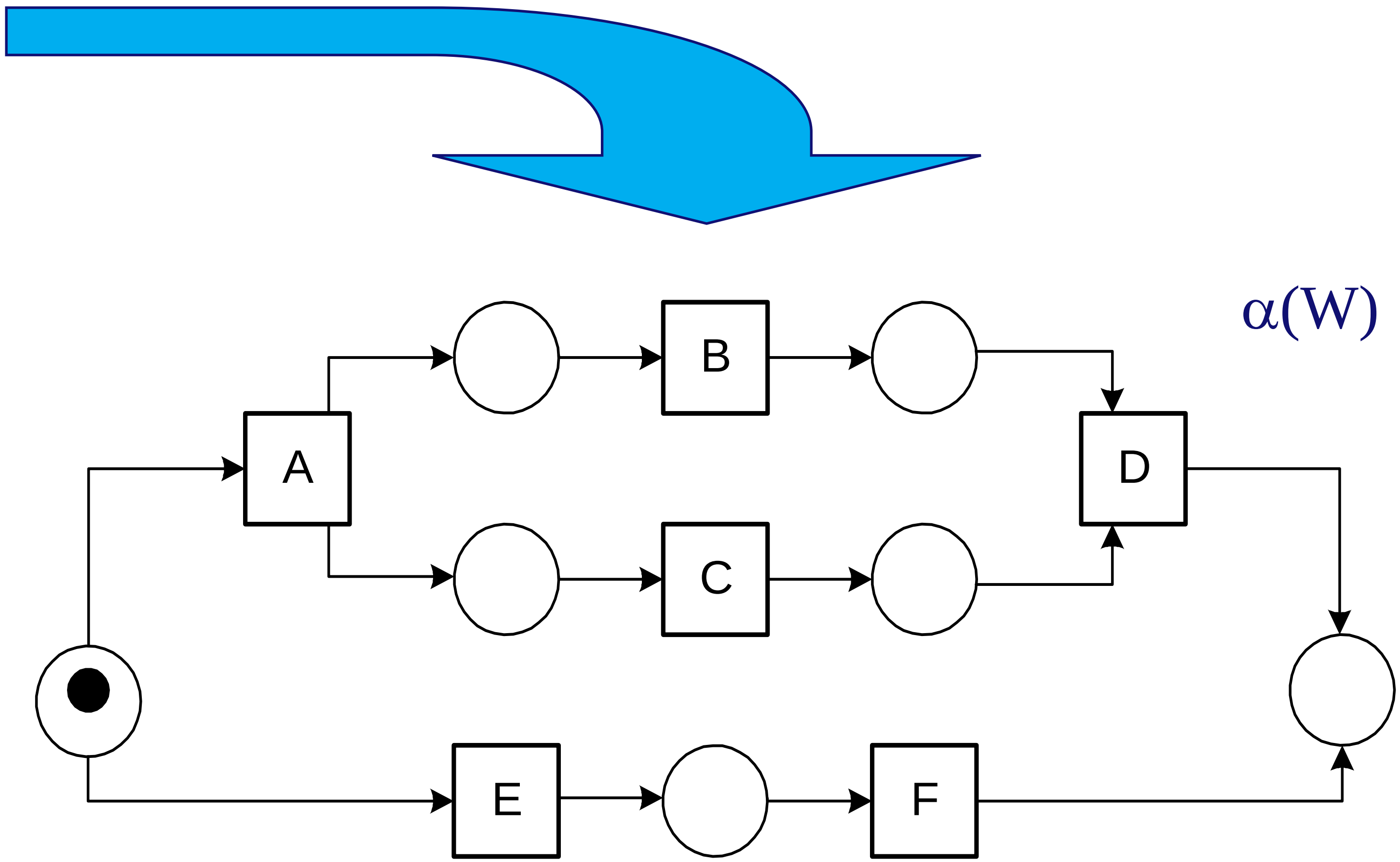
W:

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

A>B
A>C
B>C
B>D
C>B
C>D
E>F

A→B
A→C
B→D
C→D
E→F

B||C
C||B



Another example taken step-by-step ...

Definition 3.1 (Workflow trace, Workflow log). *Let T be a set of tasks. $\sigma \in T^*$ is a* workflow trace *and $W \in \mathcal{P}(T^*)$ is a* workflow log.[2]

The workflow trace of case 1 in Table 1 is $ABCD$. The workflow log corresponding to Table 1 is

$$\{ABCD, ACBD, AED\}.$$

ase identifier	task identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

Definition 3.2 (Log-based ordering relations). *Let W be a workflow log over T, i.e., $W \in \mathcal{P}(T^*)$. Let $a, b \in T$:*

- $a >_W b$ *iff there is a trace* $\sigma = t_1 t_2 t_3 \ldots t_{n-1}$ *and* $i \in \{1, \ldots, n-2\}$ *such that* $\sigma \in W$ *and* $t_i = a$ *and* $t_{i+1} = b$,
- $a \rightarrow_W b$ *iff* $a >_W b$ *and* $b \not>_W a$,
- $a \#_W b$ *iff* $a \not>_W b$ *and* $b \not>_W a$, *and*
- $a \|_W b$ *iff* $a >_W b$ *and* $b >_W a$.

Consider the workflow log $W = \{ABCD, ACBD, AED\}$ (i.e., the log shown in Table 1). Relation $>_W$ describes which tasks appeared in sequence (one directly following the other). Clearly, $A >_W B$, $A >_W C$, $A >_W E$, $B >_W C$, $B >_W D$, $C >_W B$, $C >_W D$, and $E >_W D$. Relation $\rightarrow_W$ can be computed from $>_W$ and is referred to as the *(direct) causal relation* derived from workflow log W. $A \rightarrow_W B$, $A \rightarrow_W C$, $A \rightarrow_W E$, $B \rightarrow_W D$, $C \rightarrow_W D$, and $E \rightarrow_W D$. Note that $B \not\rightarrow_W C$ because $C >_W B$. Relation $\|_W$ suggests potential parallelism. For log W, tasks B and C seem to be in parallel, i.e., $B \|_W C$ and $C \|_W B$. If two tasks can follow each other directly in any order, then all possible interleavings are present and,

A>B
A>C
A>E
B>C
D>D
C>B
C>D
E>D

A→B
A→C
A→E
B→D
C→D
E→D

B‖C
C‖B

Definition 4.3 (Mining algorithm α). *Let W be a workflow log over T. $\alpha(W)$ is defined as follows:*

$W = \{ABCD, ACBD, AED\}$

1. $T_W = \{t \in T \mid \exists_{\sigma \in W} t \in \sigma\}$,
2. $T_I = \{t \in T \mid \exists_{\sigma \in W} t = first(\sigma)\}$,
3. $T_O = \{t \in T \mid \exists_{\sigma \in W} t = last(\sigma)\}$,
4. $X_W = \{(A, B)$
   $\wedge \ \forall_{a \in A} \forall$
   $\wedge \ \forall_{b_1, b_2 \in}$
5. $Y_W = \{(A, B$
   $\wedge B \subseteq B' \Longrightarrow (A, B) = (A', B')\}$,
6. $P_W = \{p_{(A,B)} \mid (A, B) \in Y_W\} \cup \{i_W, o_W\}$,
7. $$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \ \wedge \ a \in A\} \cup \{(p_{(A,B)}, b) \mid (A, B) \in Y_W \ \wedge \ b \in B\} \cup \{(i_W, t) \mid t \in T_I\} \cup \{(t, o_W) \mid t \in T_O\},$$

   and
8. $\alpha(W) = (P_W, T_W, F_W)$.

$$1. \quad T_W = \{A, B, C, D, E\},$$

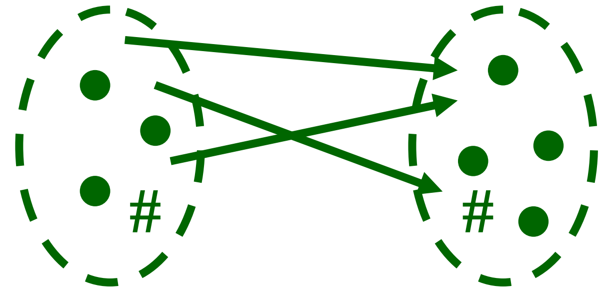
$$2. \quad T_I = \{A\},$$

$$3. \quad T_O = \{D\},$$

Definition 4.3 (Mining algorithm α). *Let W be a workflow log over T. $\alpha(W)$ is defined as follows:*

1. $T_W = \{t \in T \mid \exists_{\sigma \in W} t \in \sigma\}$,
2. $T_I = \{t \in T \mid \exists_{\sigma \in W} t = first(\sigma)\}$,
3. $T_O = \{t \in T \mid \exists_{\sigma \in W} t = last(\sigma)\}$,
4. $$X_W = \{(A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow_W b \wedge \forall_{a_1,a_2 \in A} a_1 \#_W a_2 \wedge \forall_{b_1,b_2 \in B} b_1 \#_W b_2\},$$
5.

$$W = \{ABCD, ACBD, AED\}$$



A and B need to be non-empty.

4.

$$X_W = \{(\{A\},\{B\}),(\{A\},\{C\}),(\{A\},\{E\}), (\{B\},\{D\}),(\{C\},\{D\}),(\{E\},\{D\}), (\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\{D\}), (\{C,E\},\{D\})\},$$

and

8. $\alpha(W) = (P_W, T_W, F_W)$.

4.

$$X_W = \{(\{A\},\{B\}),(\{A\},\{C\}),(\{A\},\{E\}),\\ (\{B\},\{D\}),(\{C\},\{D\}),(\{E\},\{D\}),\\ (\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\{D\}),\\ (\{C,E\},\{D\})\},$$

5.

$$Y_W = \{(A,B) \in X_W \mid \forall_{(A',B')\in X_W} A \subseteq A' \\ \wedge B \subseteq B' \Longrightarrow (A,B) = (A',B')\},$$

$$W = \{ABCD, ACBD, AED\}$$

5.

$$Y_W = \{(\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\\ \{D\}),(\{C,E\},\{D\})\},$$

and

8. $\alpha(W) = (P_W, T_W, F_W)$.

5.

$$Y_W = \{(\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\{D\}),(\{C,E\},\{D\})\},$$

$\wedge\ \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b\ \wedge\ \forall_{a_1,a_2 \in A}\, a_1 \#_W a_2$
$\wedge\ \forall_{b_1,b_2 \in B}\, b_1 \#_W b_2\},$

$$W = \{ABCD, ACBD, AED\}$$

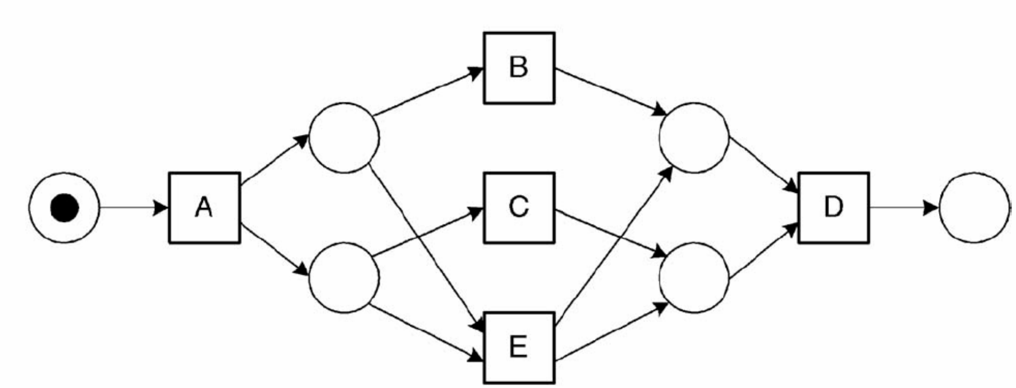
$$P_W = \{i_W, o_W, p_{(\{A\},\{B,E\})}, p_{\{A\},\{C,E\})}, p_{(\{B,E\},\{D\})}, p_{(\{C,E\},\{D\})}\},$$

$$F_W = \{(i_W, A), (A, p_{(\{A\},\{B,E\})}), (p_{(\{A\},\{B,E\})}, B), \ldots, (D, o_W)\},$$

✓ 6. $P_W = \{p_{(A,B)} \mid (A,B) \in Y_W\} \cup$

✓ 7.

$F_W = \{(a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge$
$\cup\ \{(p_{(A,B)}, b) \mid (A,B) \in Y_W$
$\cup \{(i_W, t) \mid t \in T_I\}\ \cup \{(t, o$

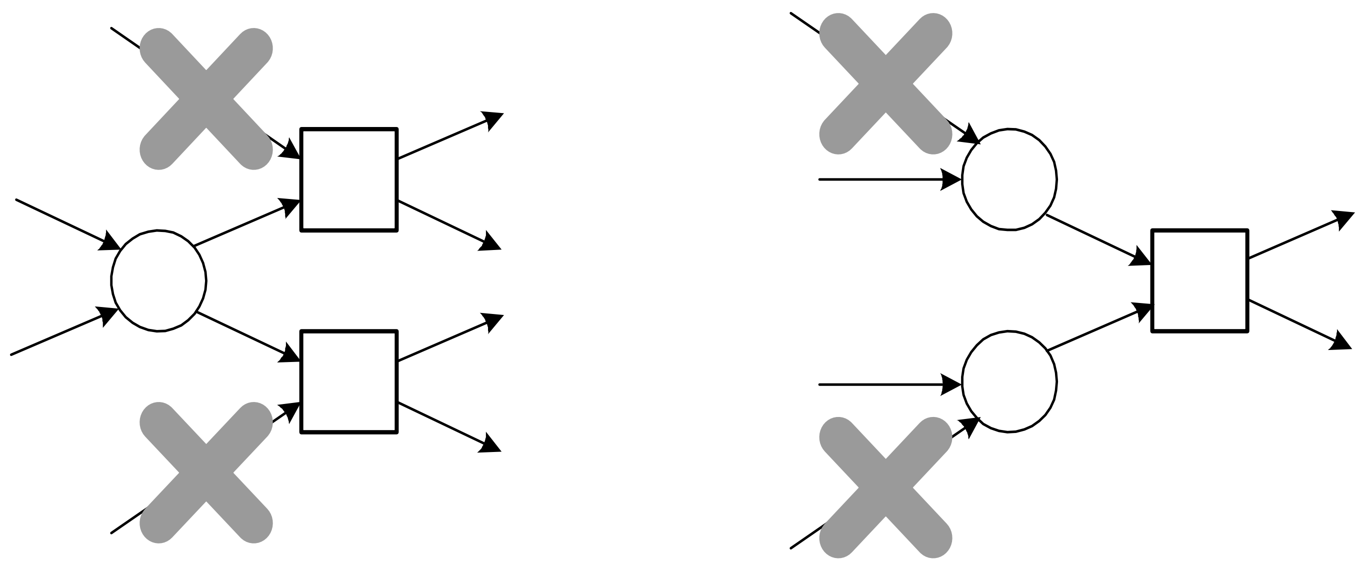


and

✓ 8. $\alpha(W) = (P_W, T_W, F_W)$.

Properties of the Alpha algorithm

- **If log is complete with respect to relation >, it can be used to mine any SWF-net!**
- ***Structured Workflow Nets* (SWF-nets) have no implicit places and the following two constructs cannot be used:**

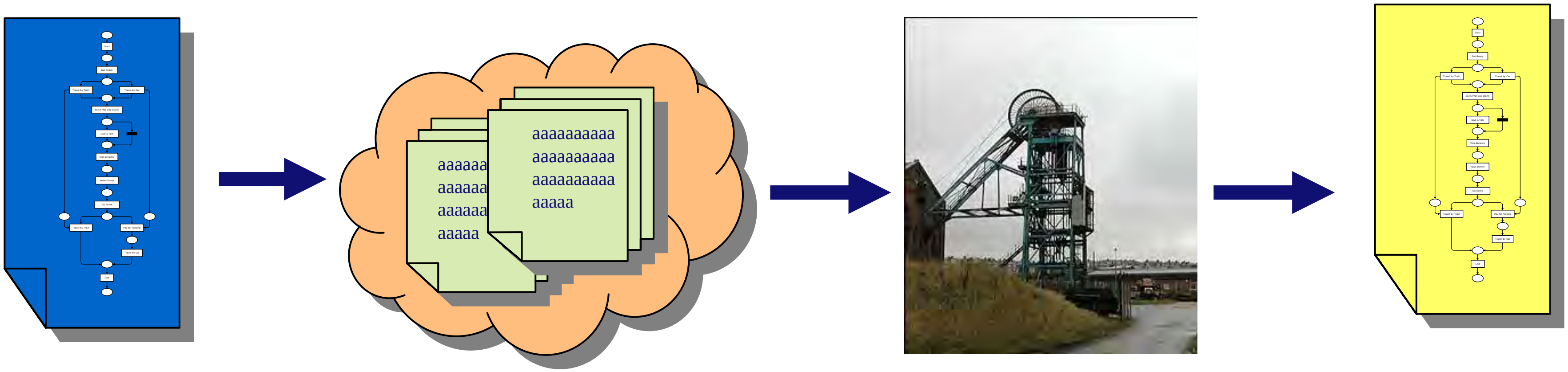


(Short loops require some refinement but not a problem.)

Alpha algorithm

- Mainly of theoretical interest!
- Too simple to be applicable to real-life logs.
- Does not address issues such as noise, etc.
- Should NOT be taken as a benchmark.
- However, the algorithm reveals:
  - basic process mining ideas and concepts in 8 lines,
  - theoretical limits of process mining.

Basic test for any mining algorithm: Rediscovery



Original Process

Logs

Mining algortithm

Mined Process

Can the mined process generate all the behavior in the log?

How close is the behavior of the mined process to the original one?

Controlled choices cannot be rediscovered (and in may cases this is good!)

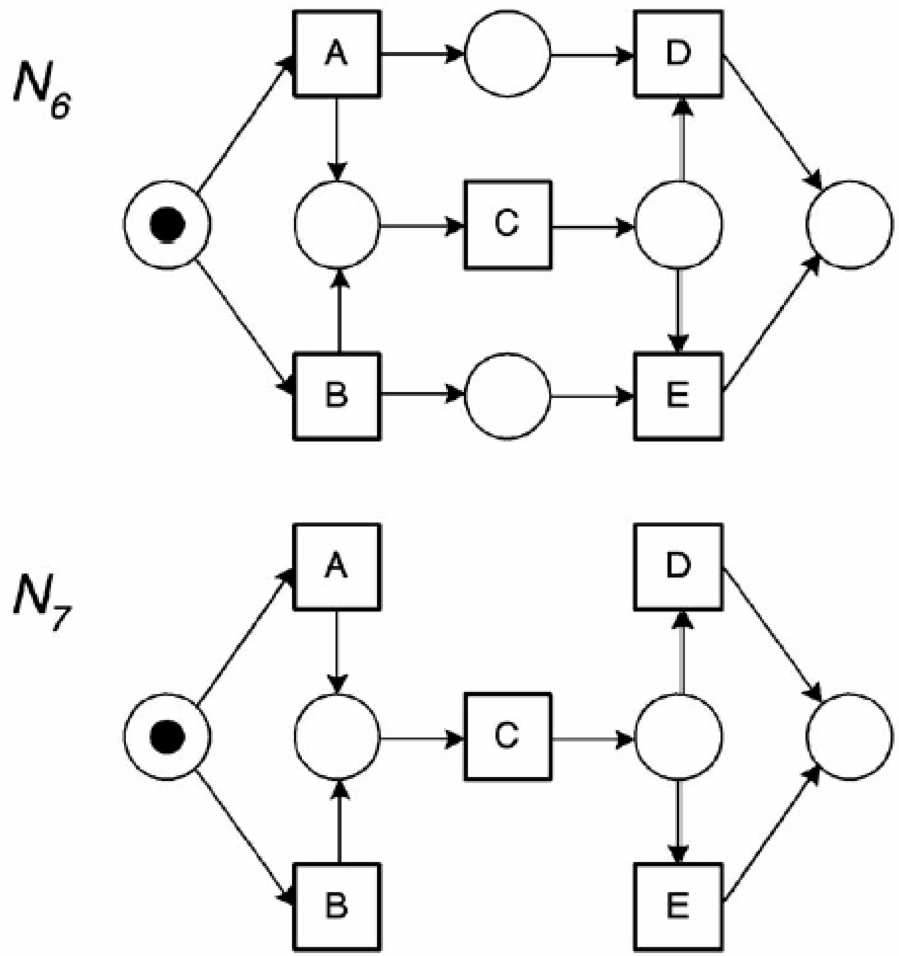


Fig. 7. The nonfree-choice WF-net N_6 cannot be rediscovered by the α algorithm.

Log only contains information about behavior and not structure

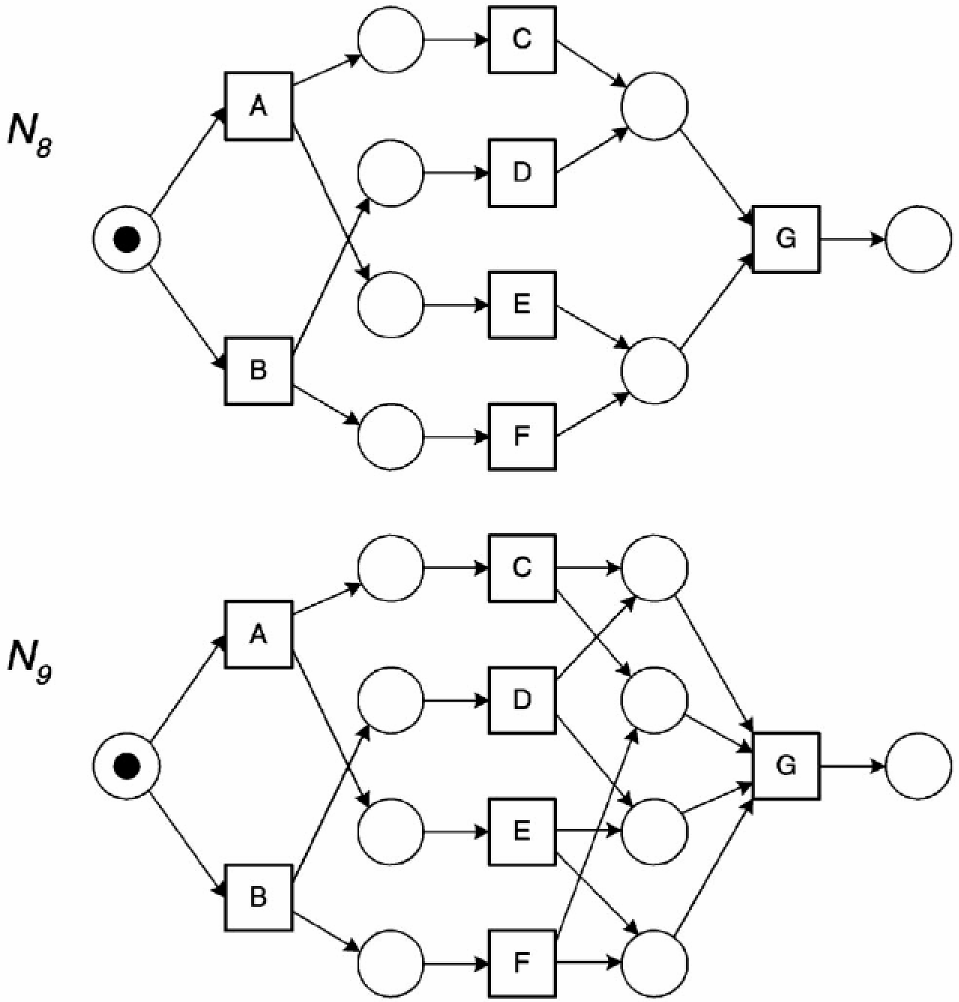


Fig. 8. WF-net N_8 cannot be rediscovered by the α algorithm. Nevertheless, α returns a WF-net which is behavioral equivalent.

Completeness notion may be too crude in some cases

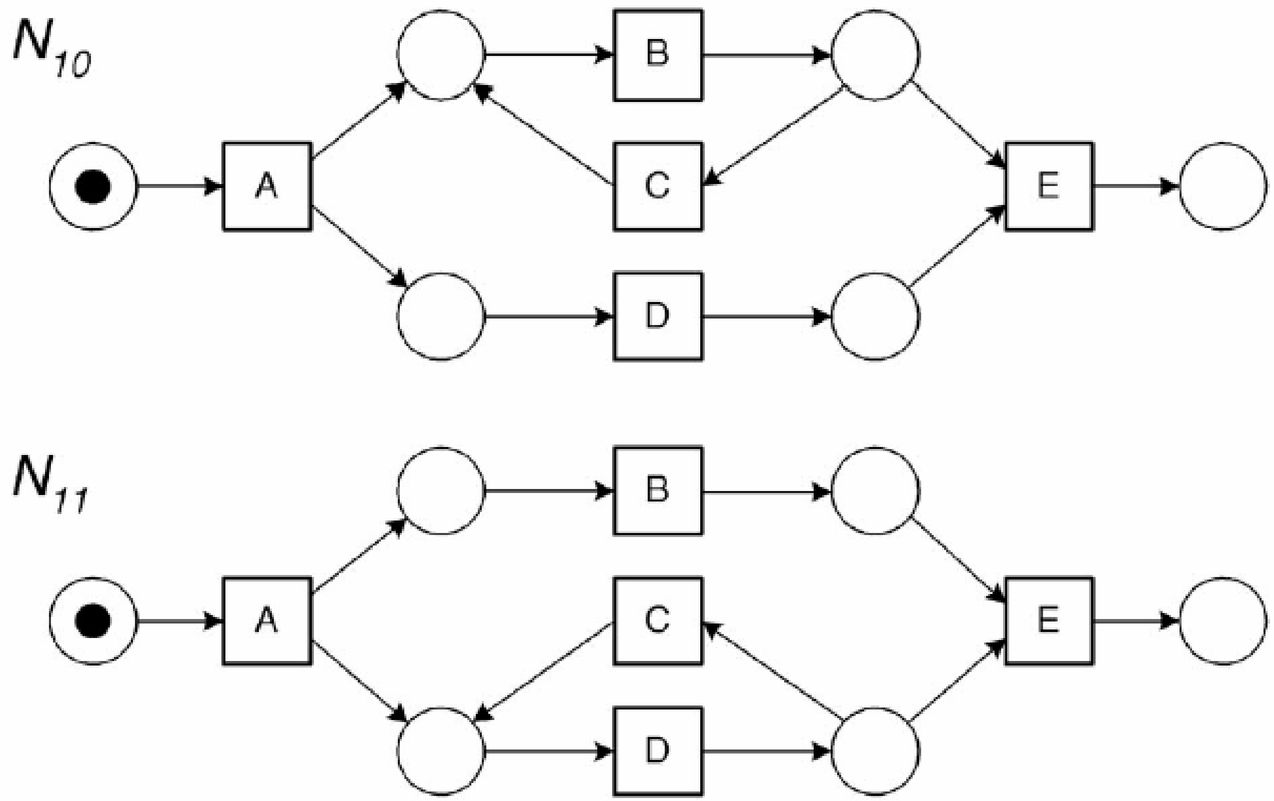


Fig. 9. Although both WF-nets are not behavioral equivalent they are identical with respect to $>$.

Another example of behaviorally equivalent SWF-nets

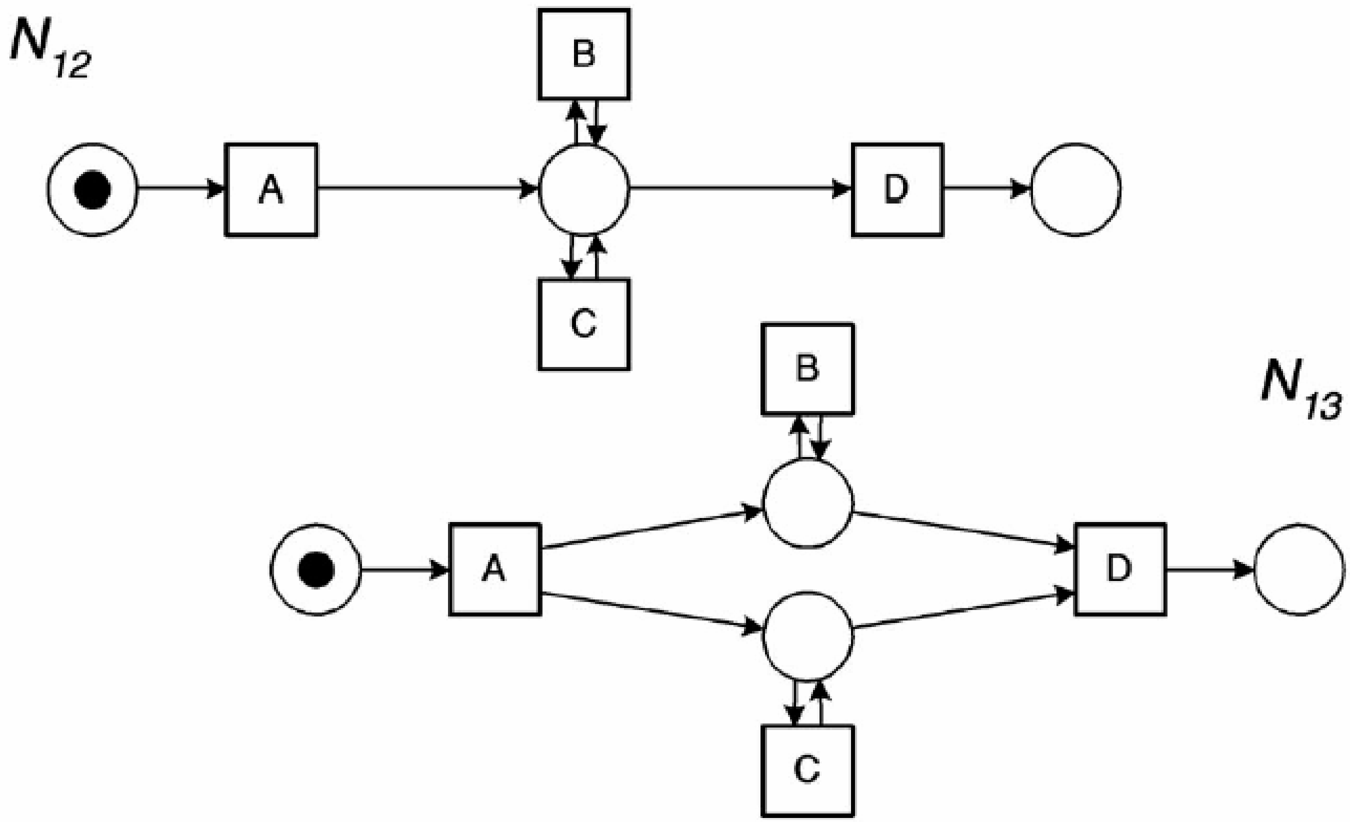
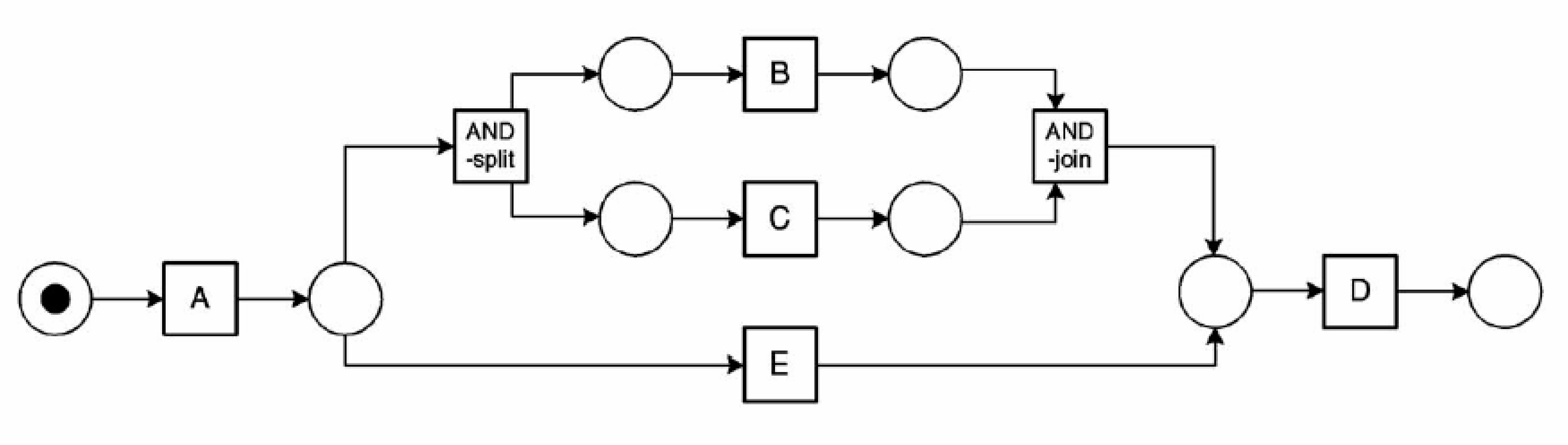
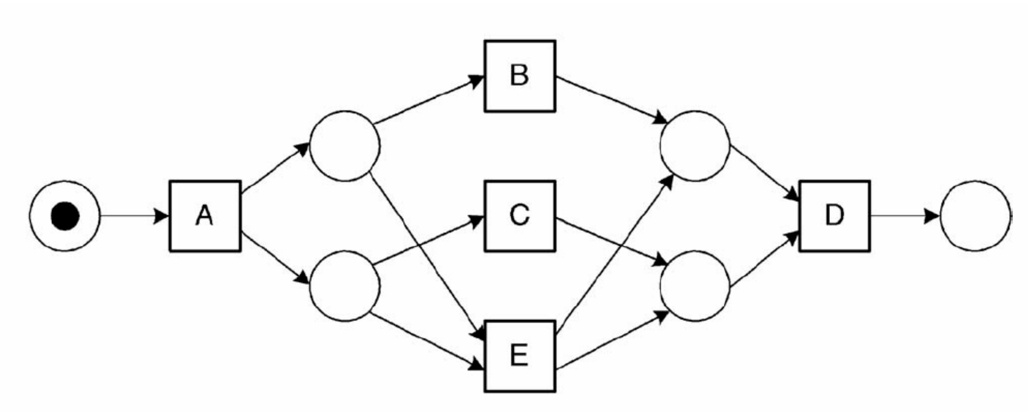


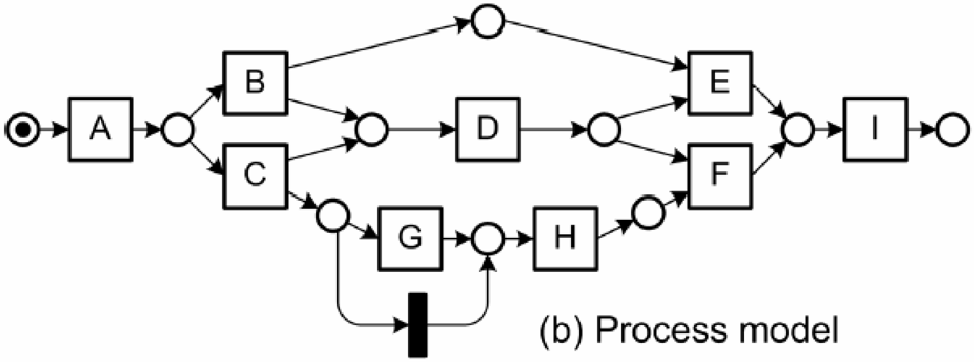
Fig. 10. Both SWF-nets are behavioral equivalent and, therefore, any algorithm will be unable to distinguish N_{12} from N_{13} (assuming a notion of completeness based on $>$).

Silent steps (and duplicate steps) cannot be discovered

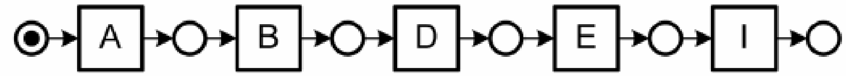




will be revisited later



(b) Process model



(c) Process model

fitness +
precision +
generalization +
structure +

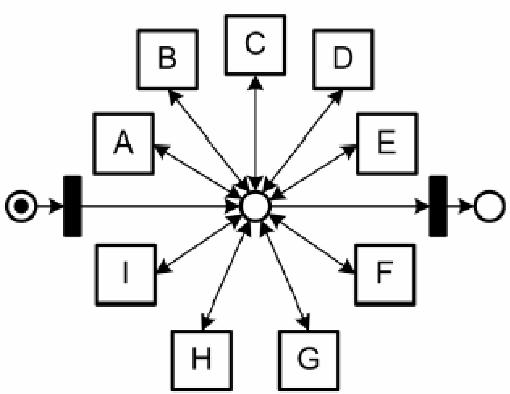
No. of Instances	Log Traces
1207	ABDEI
145	ACDGHFI
56	ACGDHFI
23	ACHDFI
28	ACDHFI

(a) Event Log

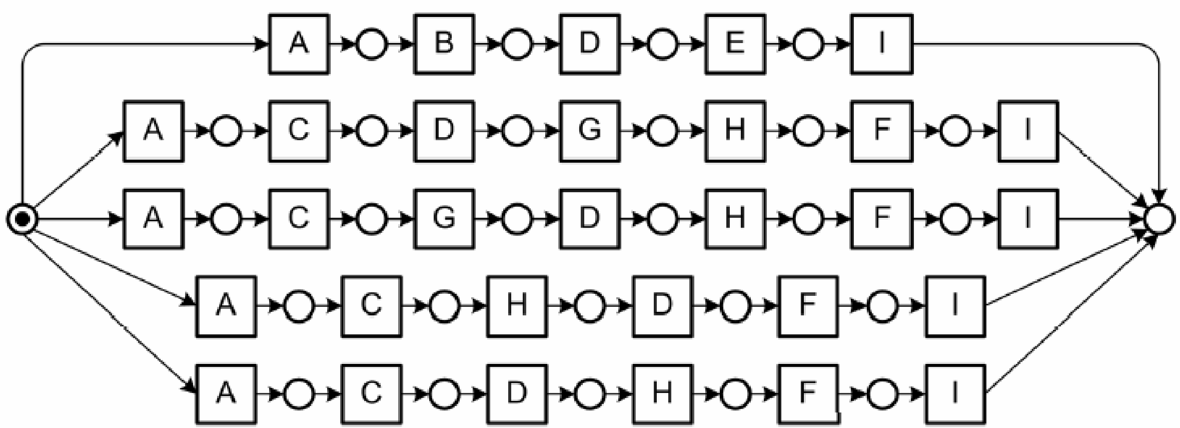
fitness -
precision +
generalization -
structure +

fitness +
precision -
generalization +
structure +

fitness +
precision +
generalization -
structure -



(d) Process model



(e) Process model

Simple process mining algorithms tend to:

- Have problems with complex control-flow constructs. **For example, many process mining algorithms are unable to deal with non-free-choice constructs and complex nested loops.**
- Not allow for duplicates. **In the event log it is not possible to distinguish between activities that are logged in a similar way, i.e., there are multiple activities that have the same “footprint” in the log. As a result, most algorithms map these different activities onto a simple activity thus making the model incorrect or counter-intuitive.**
- Silent steps. **Things that are not recorded cannot be discovered.**
- Underfit (i.e., overgeneralize) or overfit. **Many algorithms have a tendency to overgeneralize, i.e., the discovered model allows for much more behavior than actually recorded in the log. In some circumstances this may be desirable. However, there seems to be a need to flexibly balance between “overfitting” and “underfitting”.**
- Yield inconsistent models. **For more complicated processes many algorithms have a tendency to produce models that may have deadlocks and/or livelocks. It seems vital that the generated models satisfy some soundness requirements (e.g., the soundness property).**

Other Process Discovery Techniques

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Overview of process discovery techniques

- Classical techniques (e.g., learning state machines and the theory of regions): cannot handle concurrency and/or do not generalize (i.e., if it did not happen, it cannot happen).
- Algorithmic techniques
  - Alpha miner
  - Alpha+, Alpha++, Alpha#
  - Heuristic miner
  - Multi phase miner
  - ...
- Genetic process mining
- Region-based process mining
  - State-based regions
  - Language based regions

Multi-Phase Miner
(Boudewijn van Dongen et al.)

Two phases:

1) Create a visual description of each instance, without choices and loops (cf. runs or occurrence nets).
   - Comprehensive representation
   - Ideal for performance analysis (cf. ARIS PPM)
2) Aggregate multiple instances to one process model.
   - Only causal relations between tasks are required

Properties:

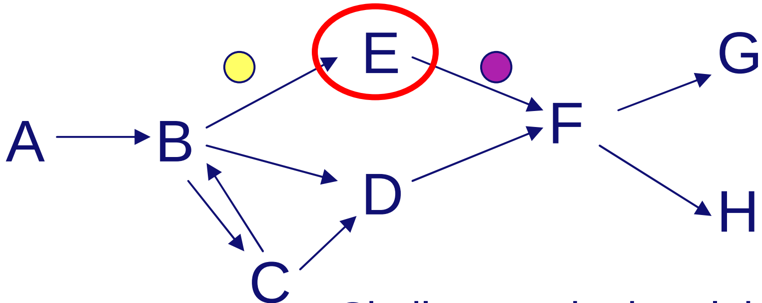
- More robust and multi-lingual (cf. EPCs).
- Possibility of inspect instances

Step 1: Create instance graphs

Log file:

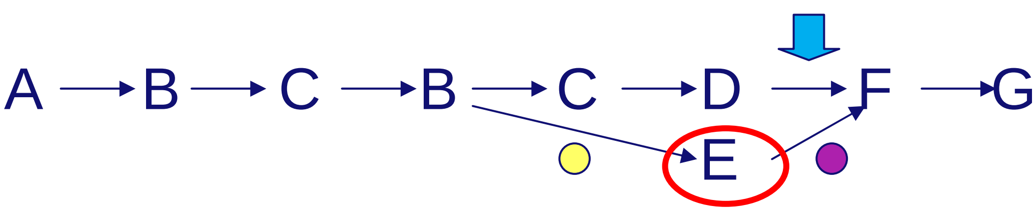
- **A, B, C, B, C, D, E, F, G**
- **A, B, E, D, F, G**
- **A, B, D, E, F, H**

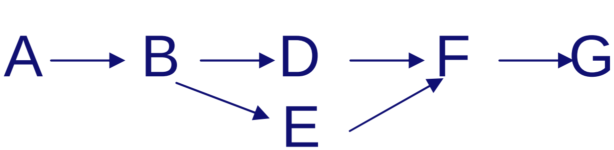
Causal relations:

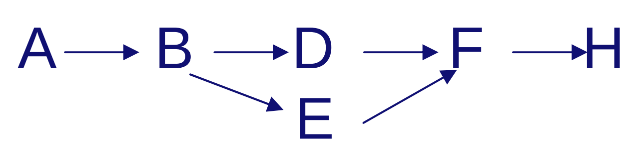


Similar to →in the alpha algorithm but refined further.

For each entry in every instance, find the closest causal predecessor and successor, and build instance graphs

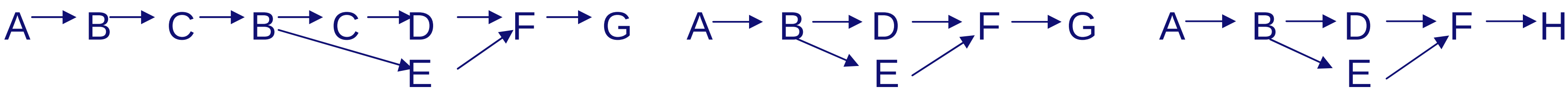




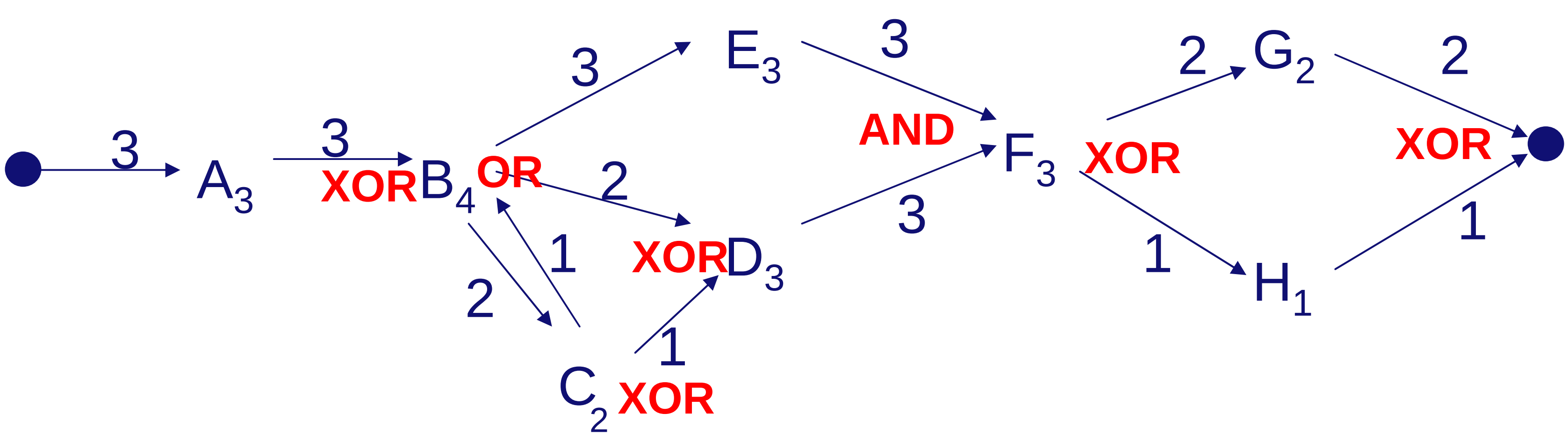


Step 2: Aggregate instance graphs

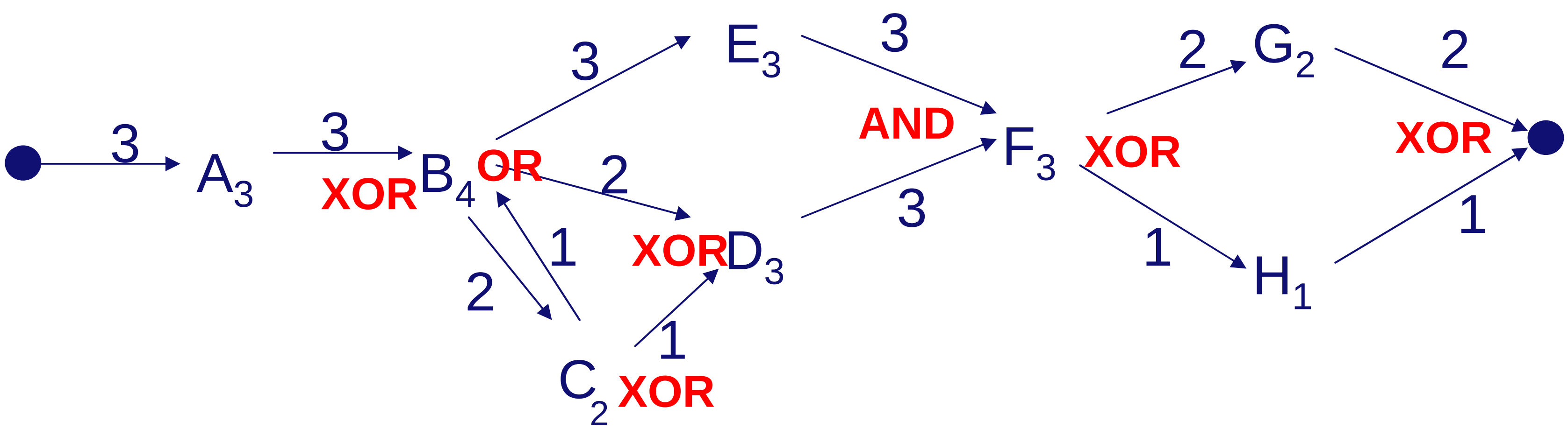
Three instance graphs:

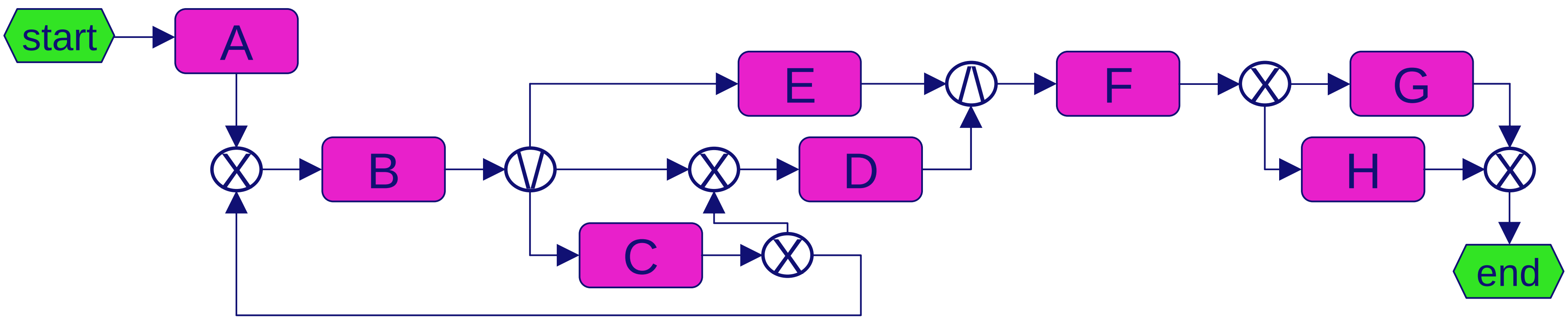


Aggregated instances:



Representation as an EPC





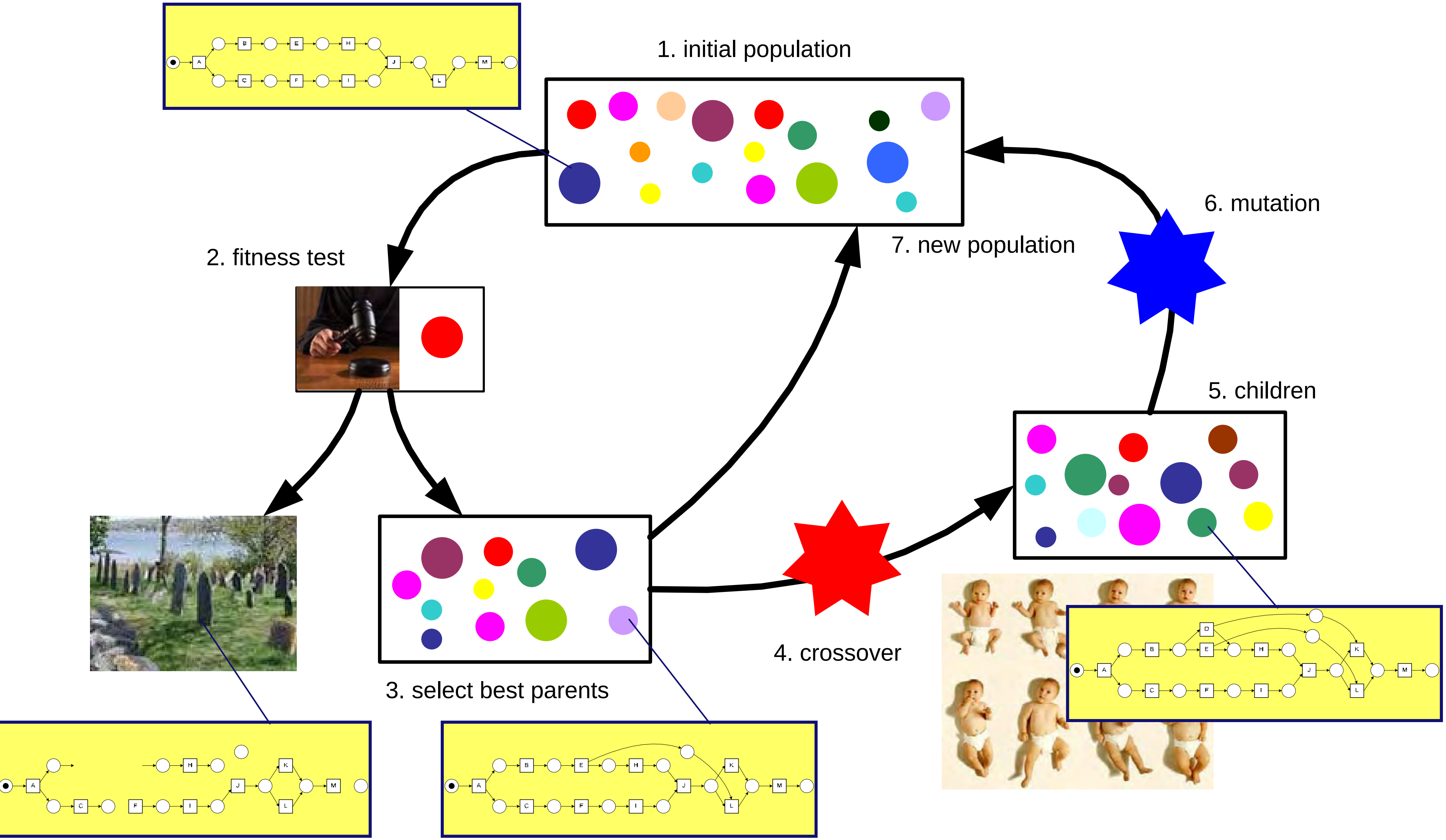
Any notation with OR-splits and OR-joins can be used, e.g., YAWL, BPMN, etc.

Properties of Multi-phase miner

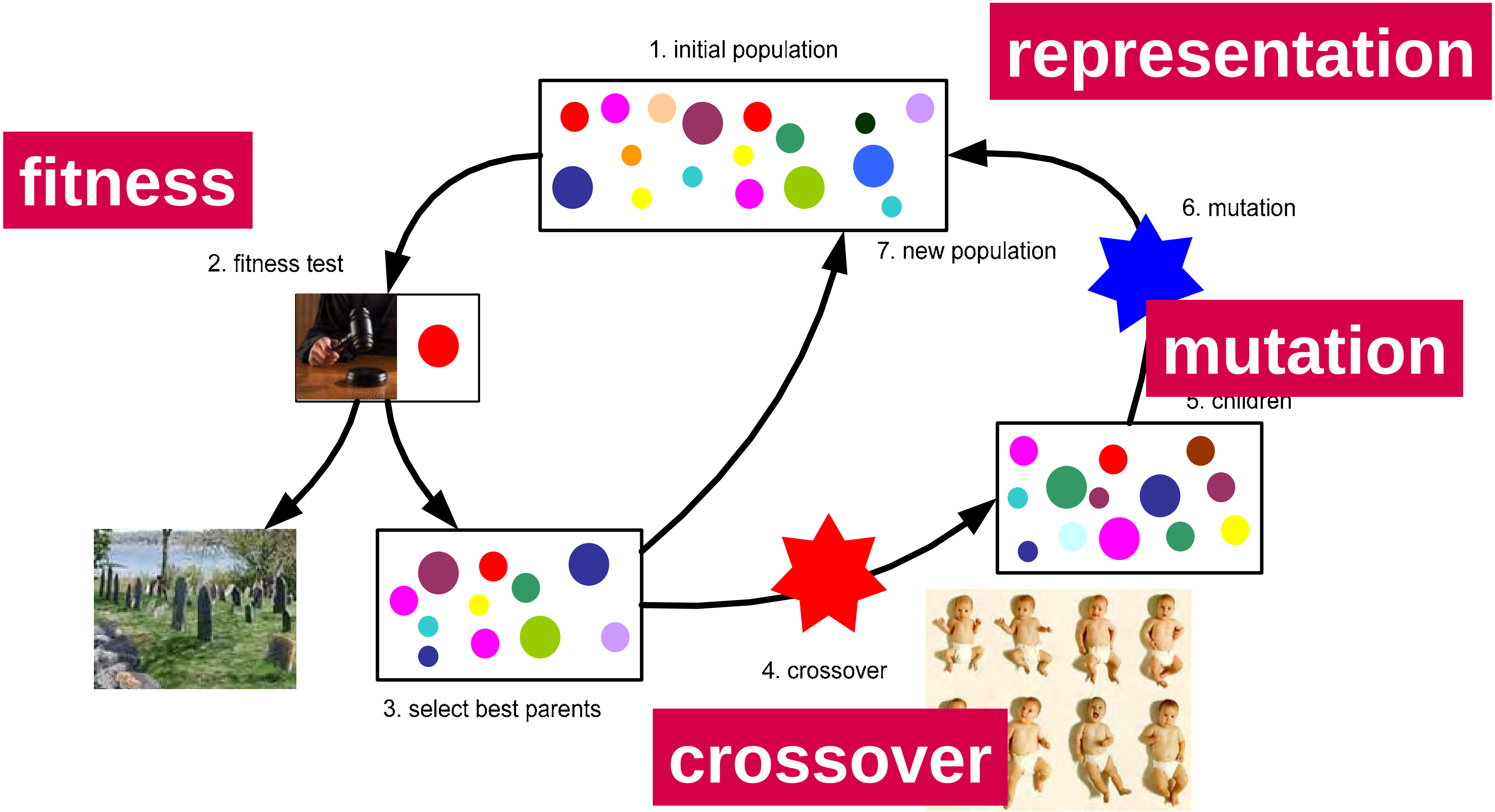
- **Always a fitness of 1, i.e., all traces in the logs may be reproduced (both potentially also many more).**
- **Very robust and fast, but tends to overgeneralize.**
- **Any subset of traces produces a meaningful result (event a single instance) that can be used for visualization purposes.**
- **No special provisions for noise or infrequent behavior.**

Genetic Mining

(Ana Karla Alves de Medeiros et al.)

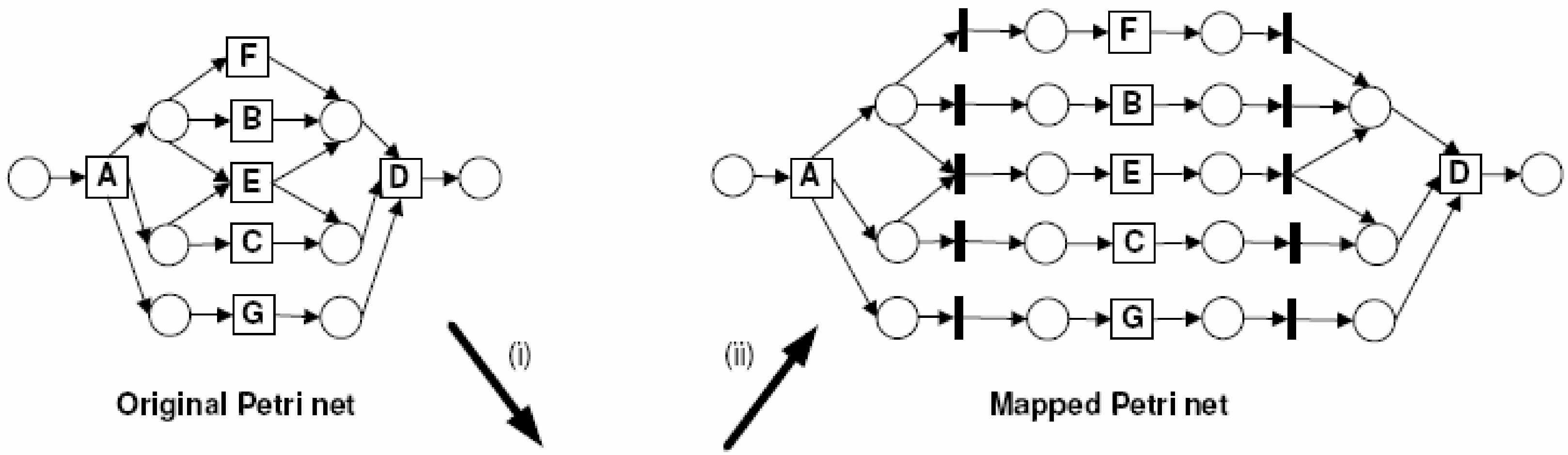


Design choices



Example representation

(Ana Karla Alves de Medeiros et al.)



Original Petri net

Mapped Petri net

ACTIVITY	*I(ACTIVITY)*	*O(ACTIVITY)*
A	{}	{{F,B,E},{E,C},{G}}
B	{{A}}	{{D}}
C	{{A}}	{{D}}
D	{{F,B,E},{E,C},{G}}	{}
E	{{A}}	{{D}}
F	{{A}}	{{D}}
G	{{A}}	{{D}}

Compact representation of the causal matrix

also allows for duplicates

Many fitness notions possible
(see also conformance checking techniques)

- Replay the token game and punish when:
  - tokens are missing
  - tokens are left
  - arcs are not taken
  - "too many" transitions are enabled
  - etc.
- Additional measures:
  - punish "complicated" models
  - insert artificial negative events
  - etc.

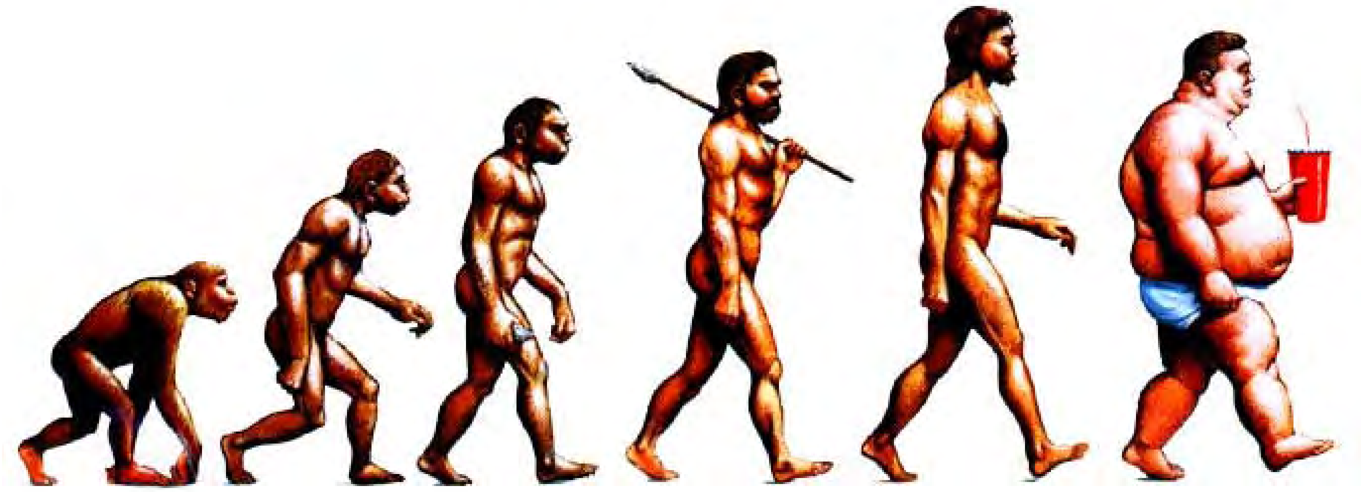
Crossover and Mutation
(Many possibilities depending on representation).

ACTIVITY	I(ACTIVITY)	O(ACTIVITY)
A	{}	{{F,B,E},{E,C},{G}}
B	{{A}}	{{D}}
C	{{A}}	{{D}}
D	{{F,B,E},{E,C},{G}}	{}
E	{{A}}	{{D}}
F	{{A}}	{{D}}
G	{{A}}	{{D}}

- **Crossover:**
  - **Child inherits some activities of one parent and the rest from another parent.**
- **Mutation:**
  - **Randomly activities or relations are inserted or removed.**

Properties of Genetic Mining

- Requires a lot of computing power.
- Can deal with noise, infrequent behavior, duplicate tasks, invisible tasks, etc.
- Allows for incremental improvement and combinations with other approaches (heuristics post-optimization, etc.).

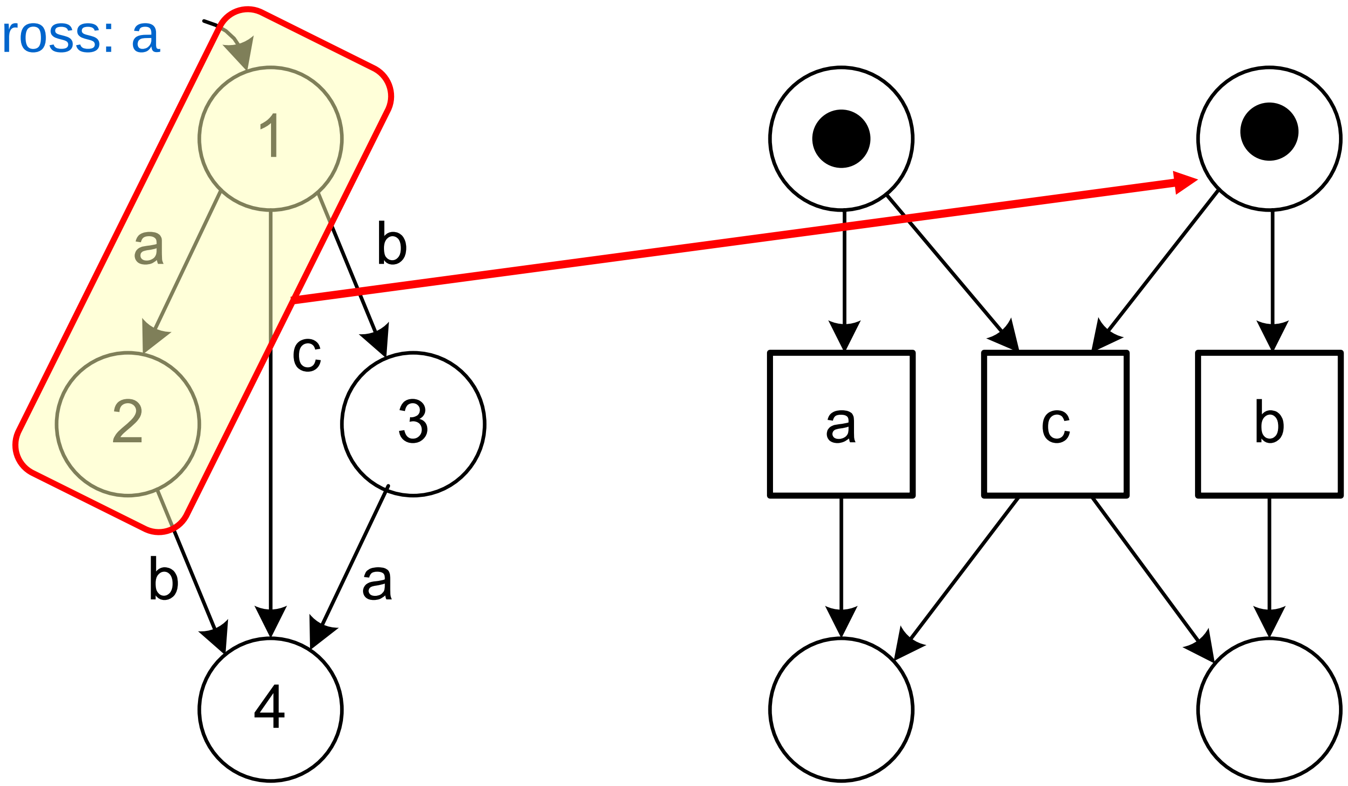


Region-Based Approaches

- Classical synthesis approaches to translate "behavior" into "models":
  - State-based regions
  - Language-based regions
- Synthesis ≠ process mining!
- Common issues:
  - Translating logs into transition systems (for state-based regions).
  - Overfitting.
  - Performance of algorithms and complexity of result.

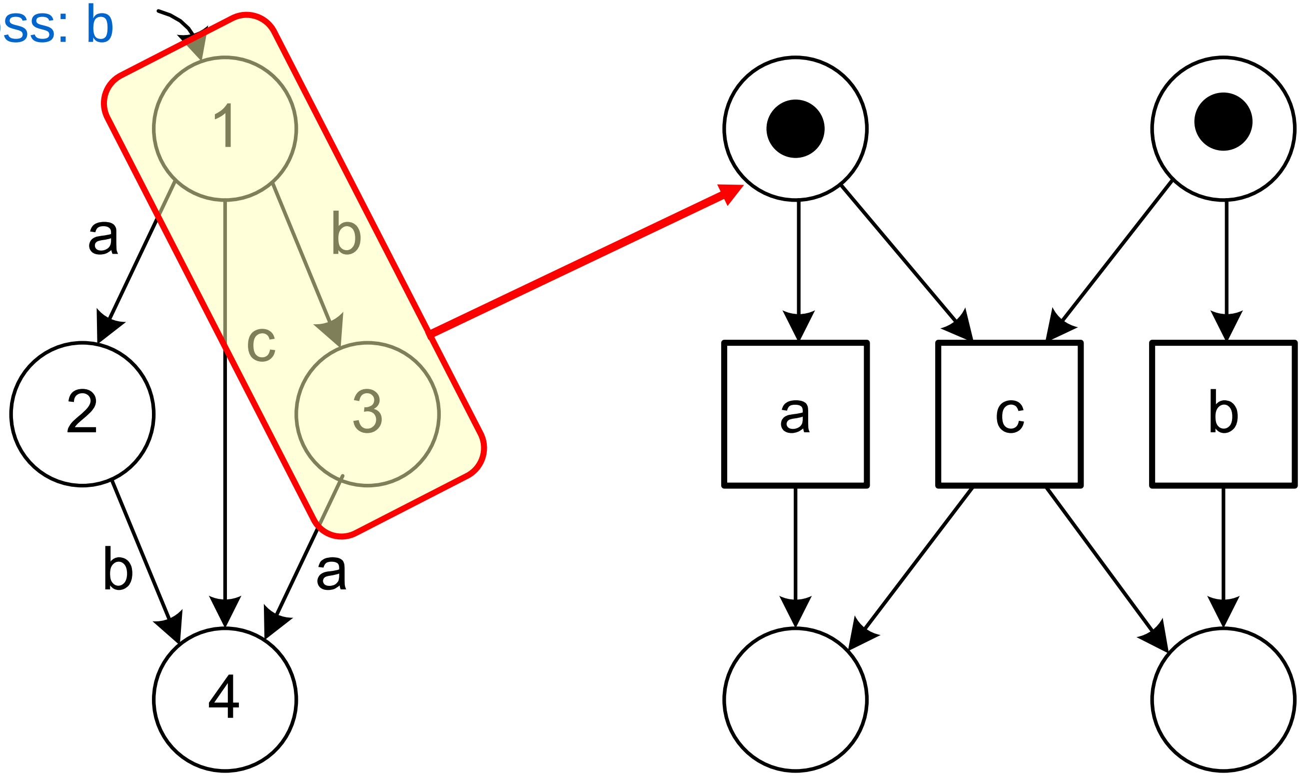
State-based regions

enter: -
exit: b,c
do not cross: a



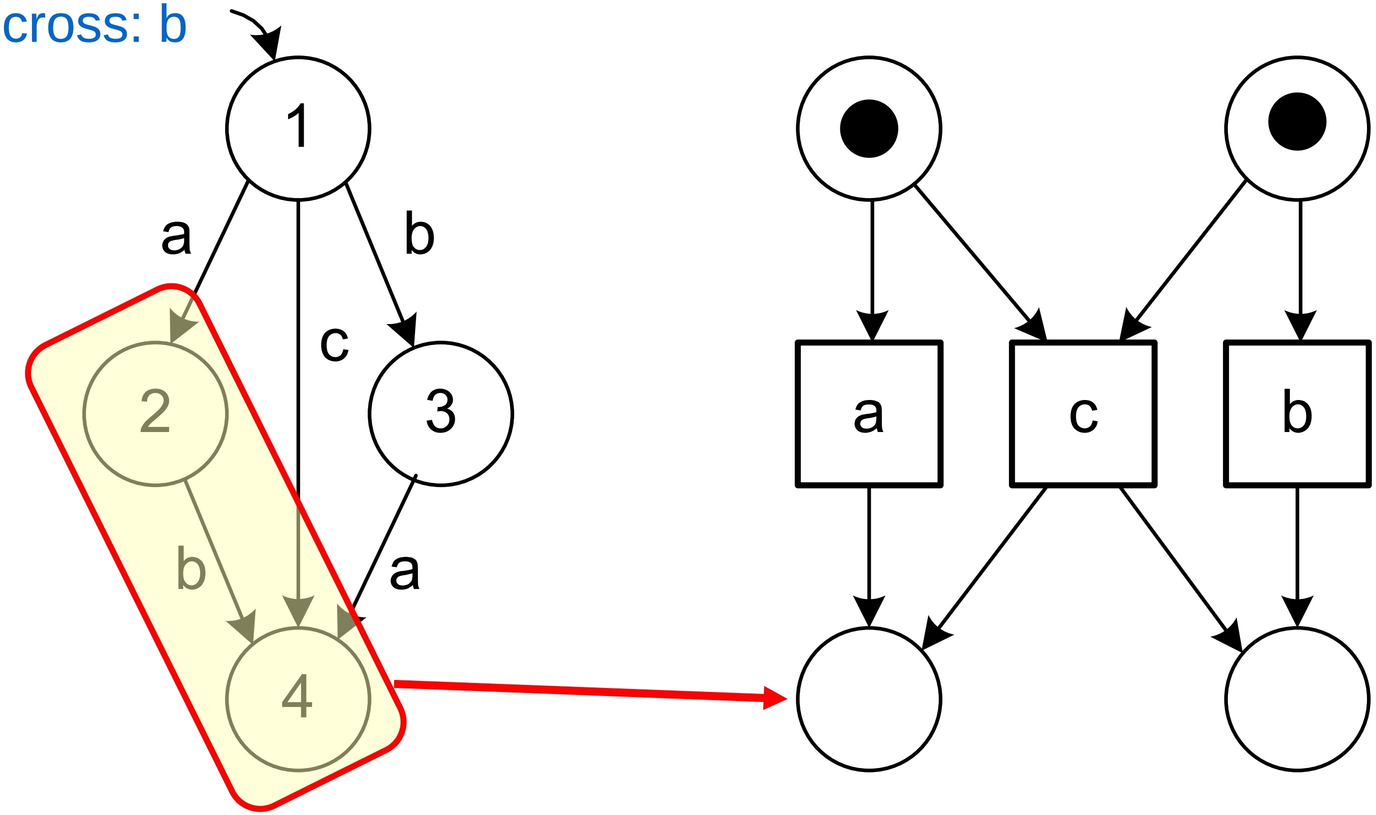
Second region

enter: -
exit: a,c
do not cross: b



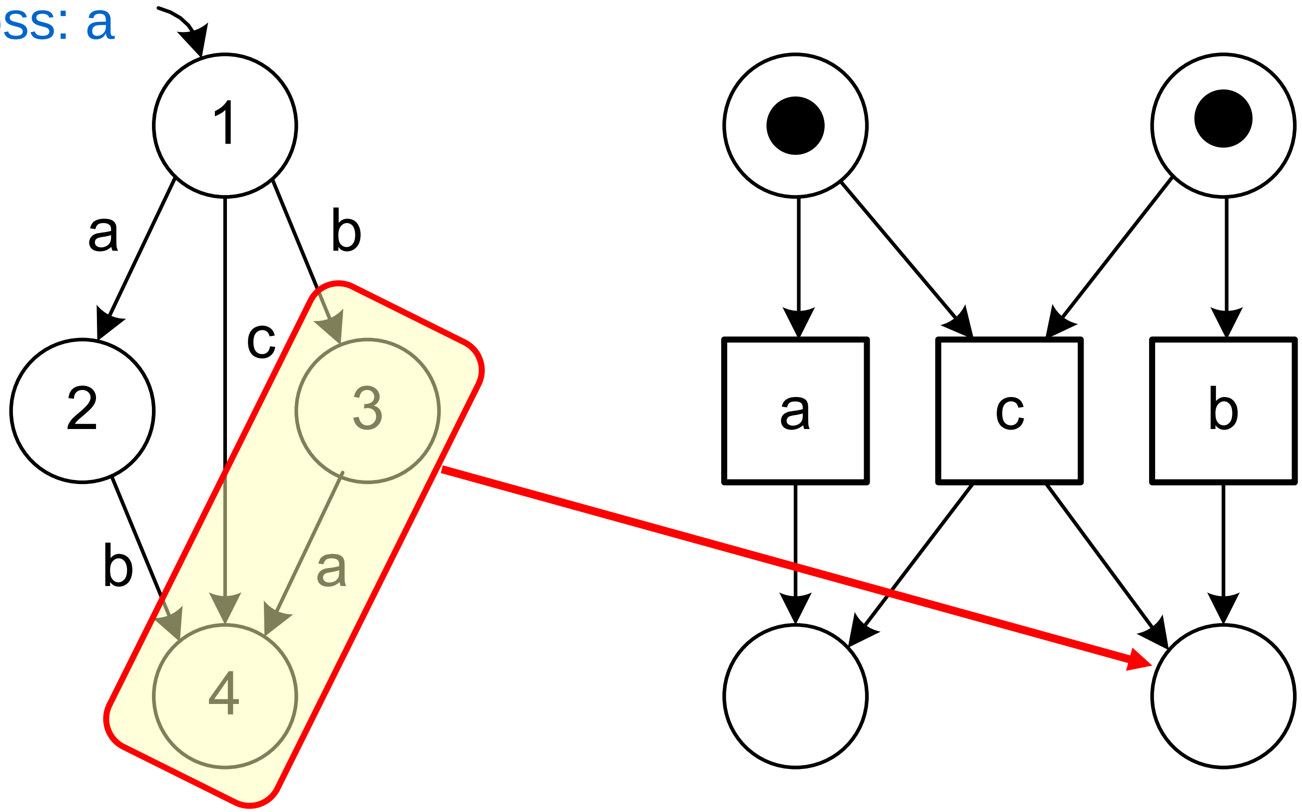
Third region

enter: a,c
exit: -
do not cross: b

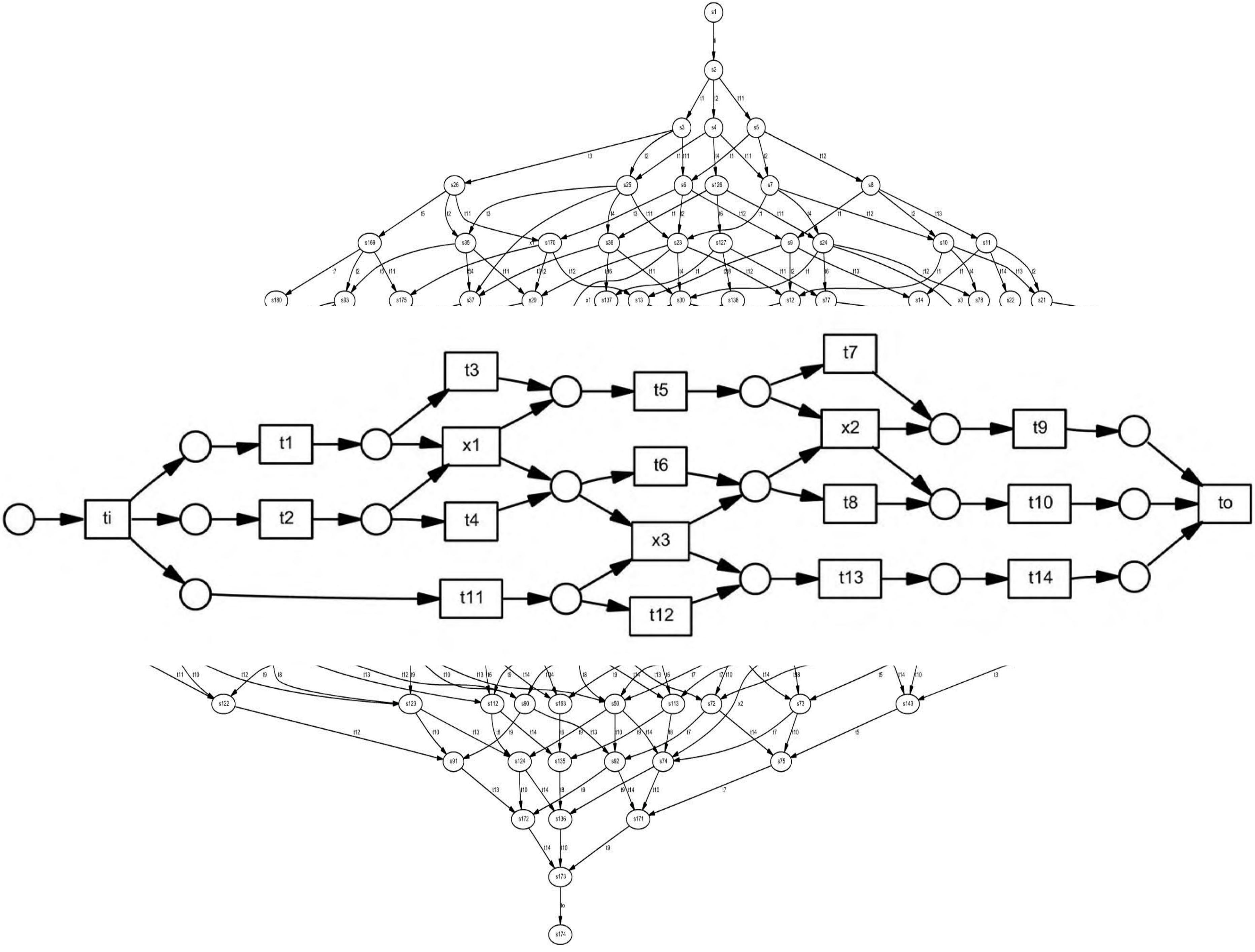


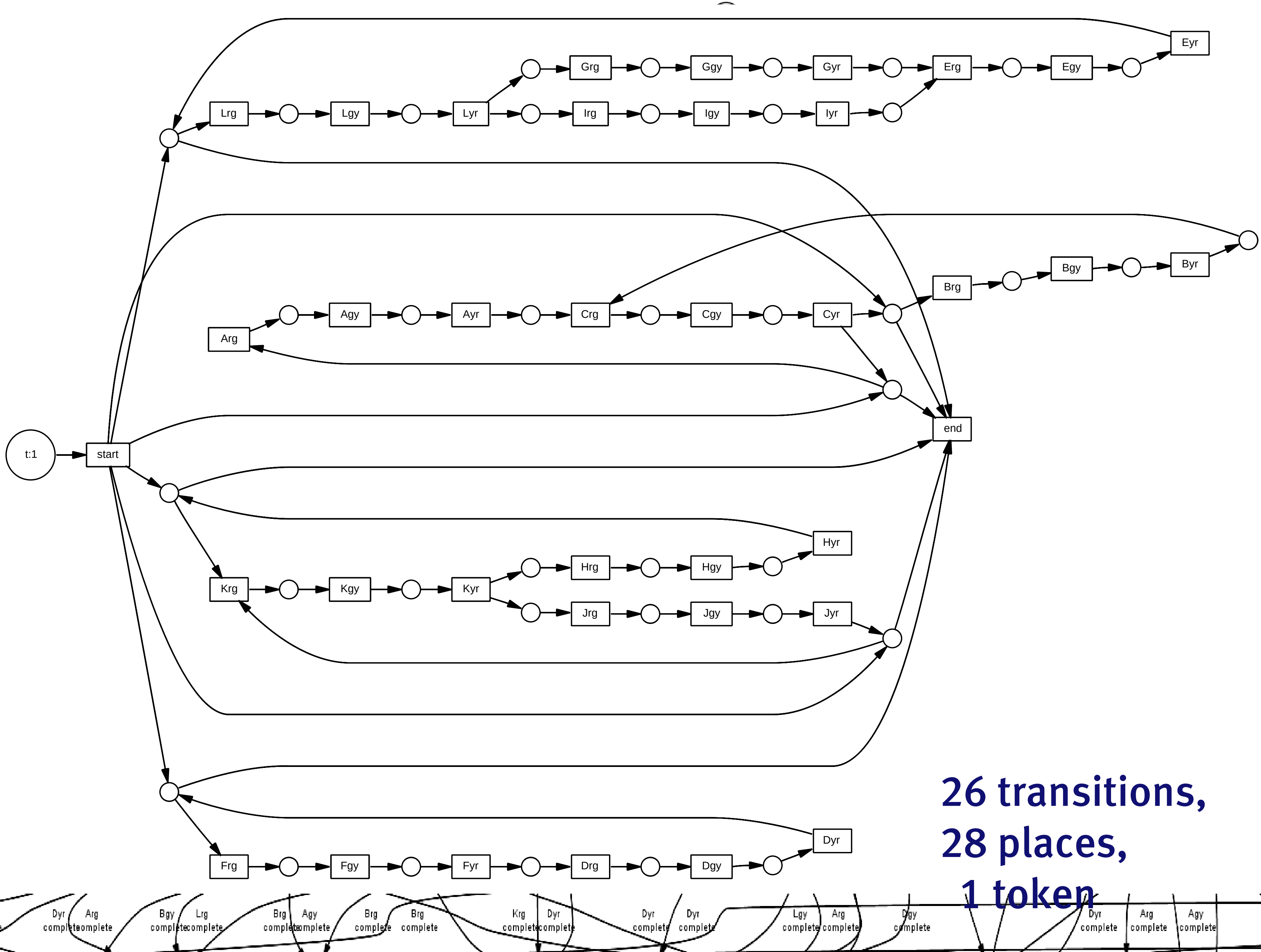
Fourth region

enter: b,c
exit: -
do not cross: a



With some extensions (see work Cortadella et al.) any transition system can be converted into a bisimilar Petri net.

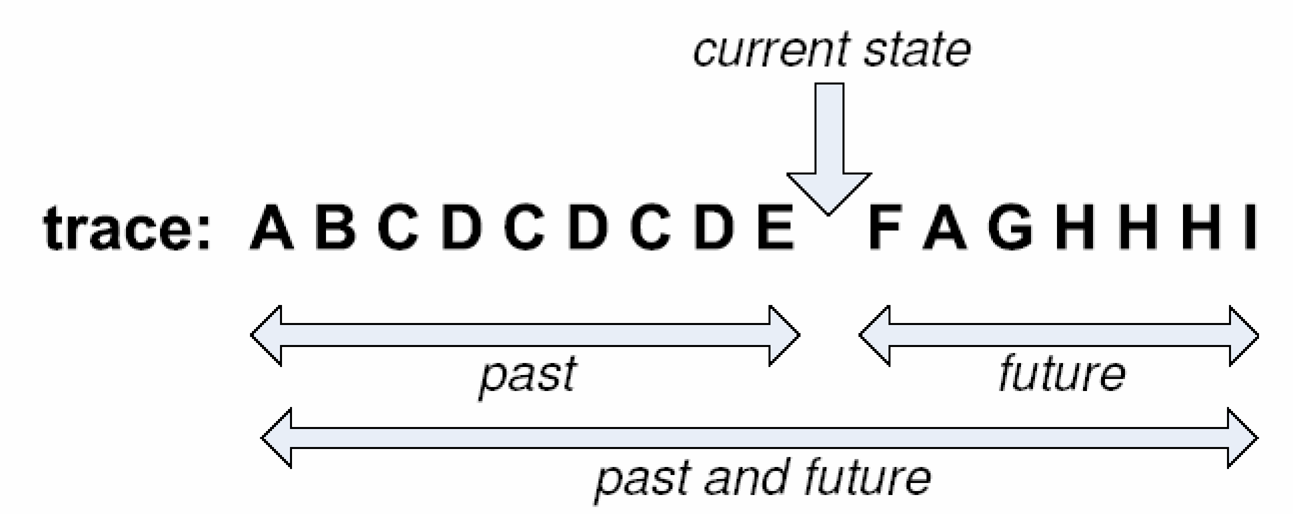

ti
t1
t2
t3
t4
t5
t6
t7
t8
t9
t10
t11
t12
t13
t14
x1
x2
x3
to



26 transitions,
28 places,
1 token

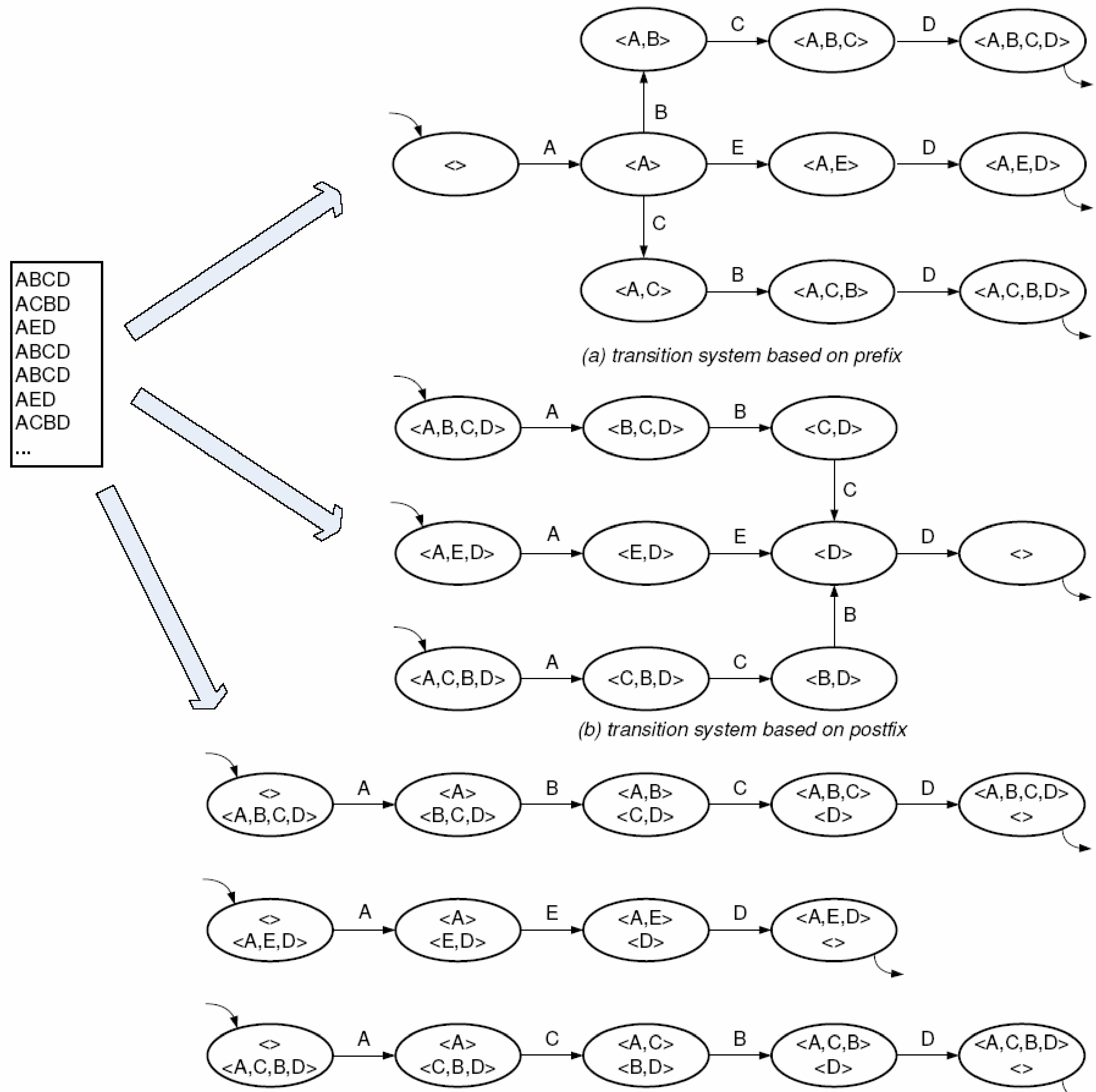
From event logs to transition systems

- **How to determine the current state?**
- **Determine scope:**



- **Determine abstraction:**
  - *sequence*, i.e., the order of activities is recorded in the state,
  - *multi-set of activities*, i.e., the number of times each activity is executed ignoring their order, or
  - *set of activities*, i.e., the mere presence of activities.

Example



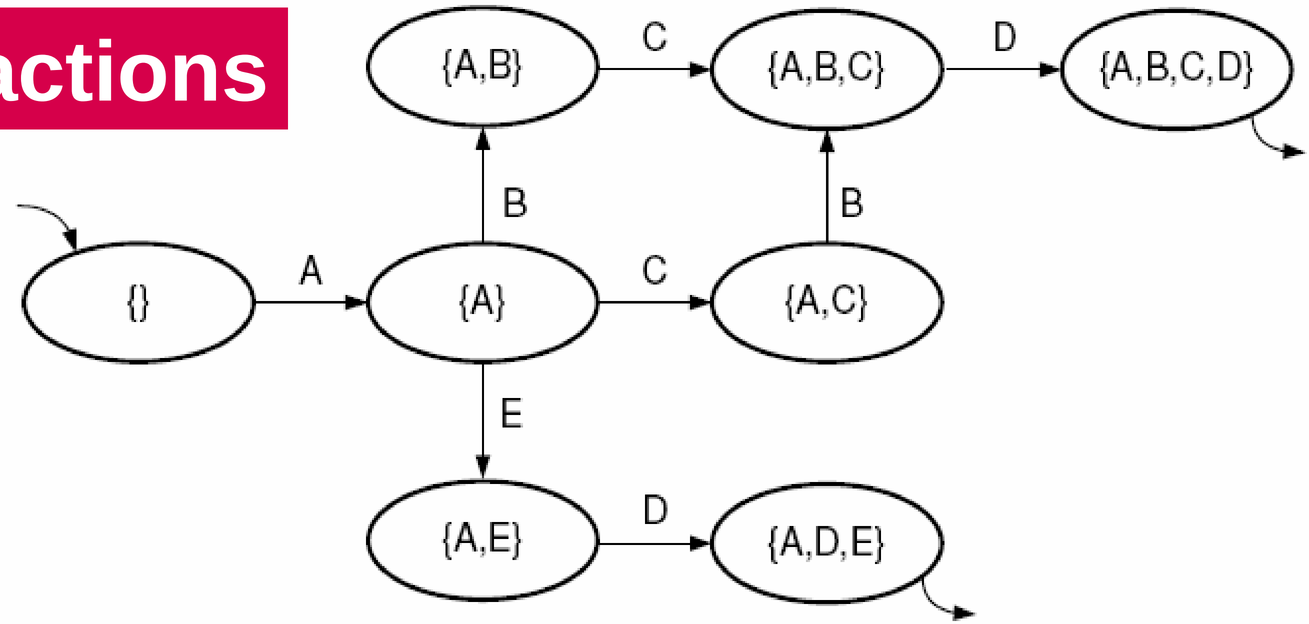
(a) transition system based on prefix

(b) transition system based on postfix

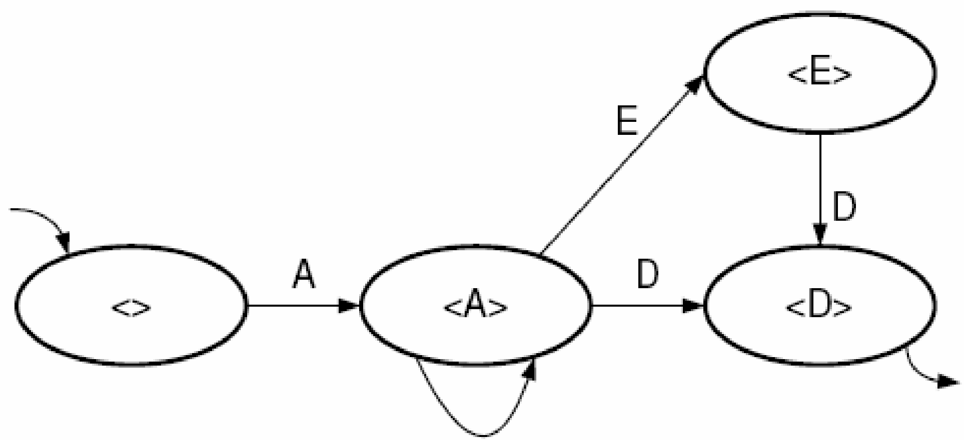
(c) transition system based on prefix and postfix

Other abstractions

ABCD
ACBD
AED
ABCD
ABCD
AED
ACBD
...

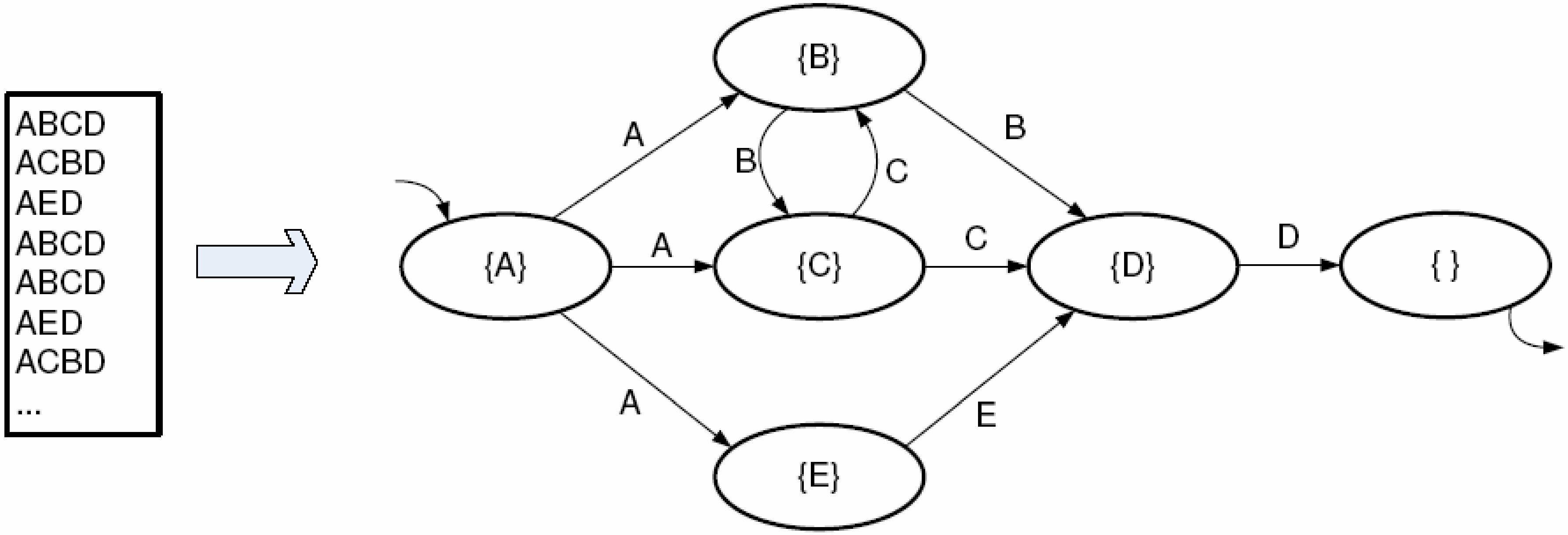


(a) transition system based on sets



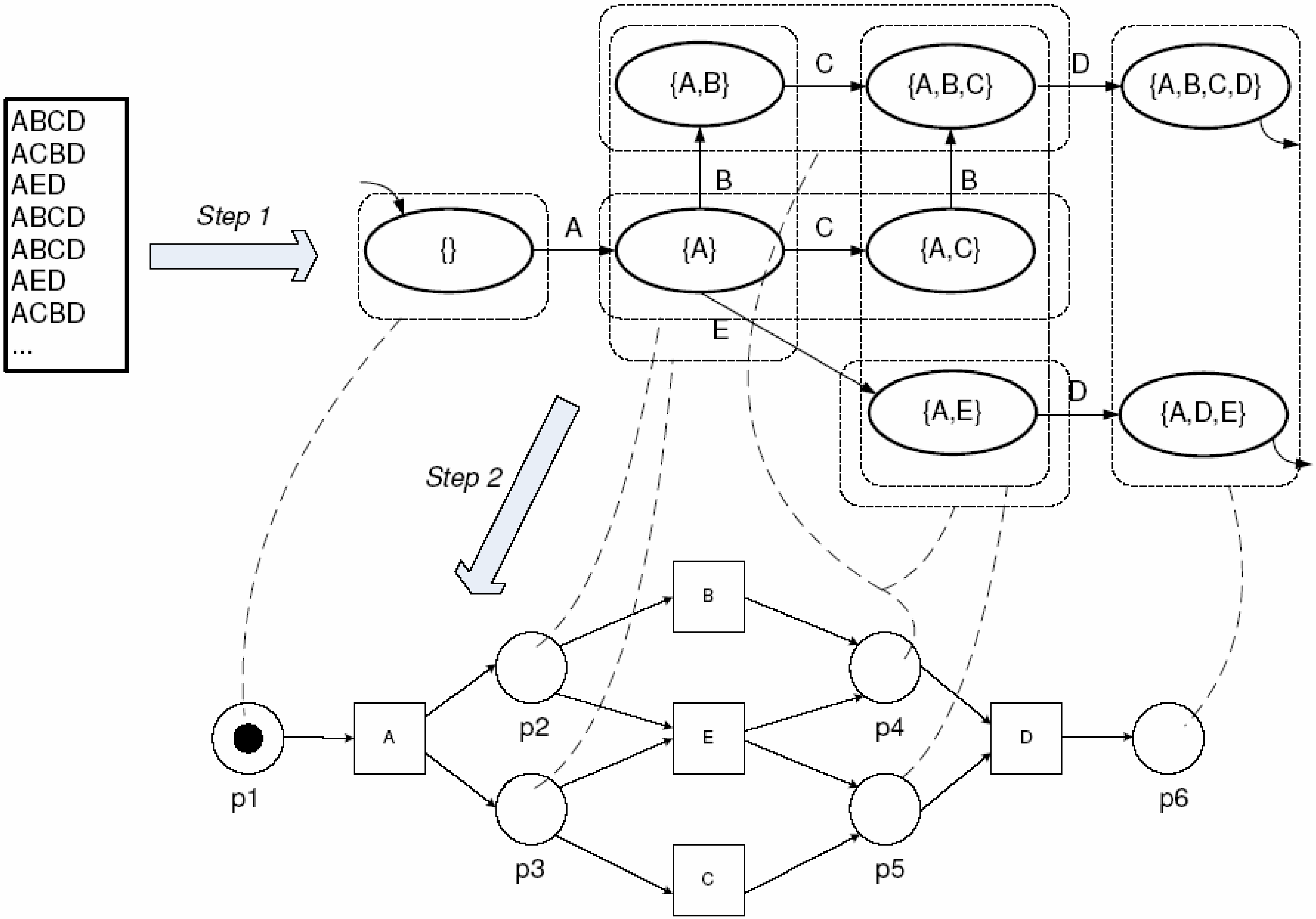
(b) transition system abstracting from B and C

Set abstraction based on future (only one step)



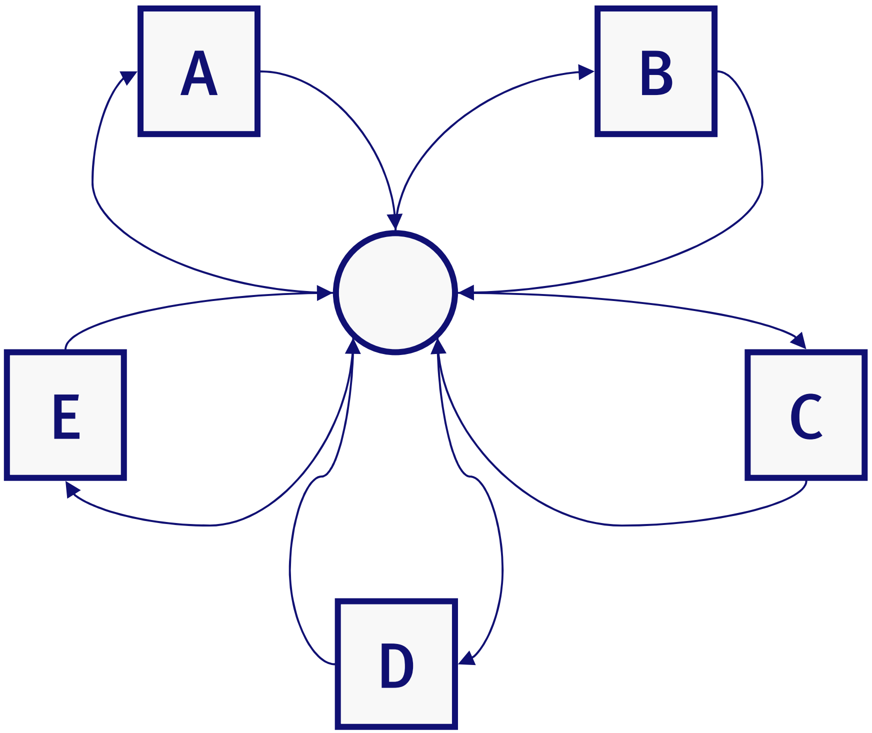
For larger processes with incomplete logs, limiting the horizon helps generalizing!

Combination: A two step approach based on controlled abstraction and regions



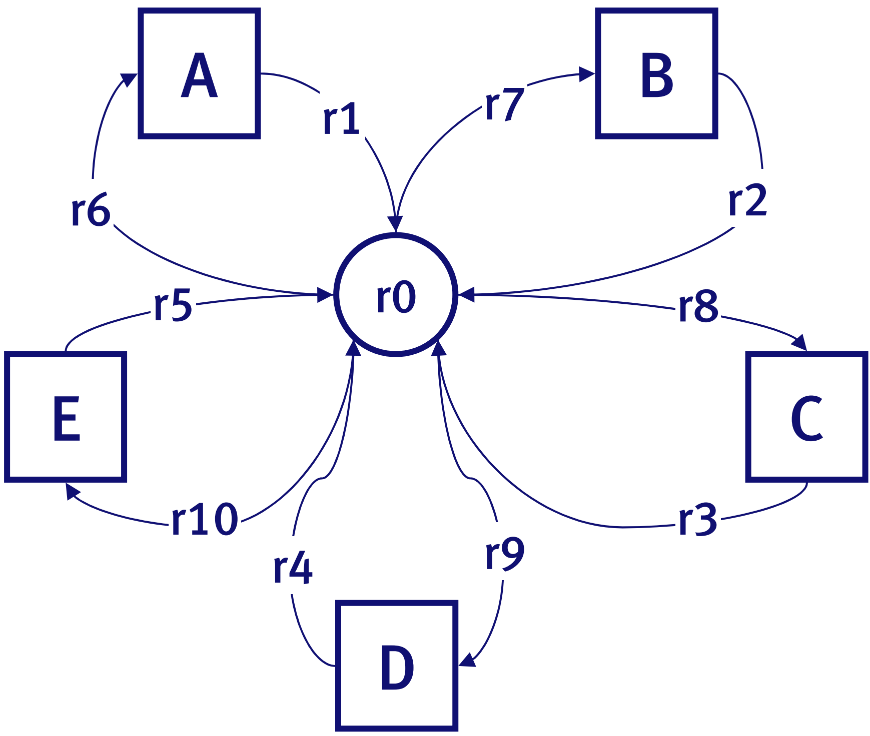
Language-Based Regions
(Van Dongen, Van der Werf, Lorenz, Desel, et al.)

- **Consider the following language:** ***abbe, acde, adce***
- **Which prefix-closed is:** ***a, ab, abb, abbe, ac, acd, acde, ad, adc, adce***
- **Idea: What are all the places I can add without making these prefixes impossible?**



Translated into a linear programming problem *A.r >= 0*

a	r0−r6	>= 0
ab	r0+r1−r6−r7	>= 0
abb	r0+r1+r2−r6−2 r7	>= 0
abbe	r0+r1+2 r2−r6−2 r7−r10	>= 0
ac	r0+r1−r6−r8	>= 0
acd	r0+r1+r3−r6−r8−r9	>= 0
acde	r0+r1+r3+r4−r6−r8−r9−r10	>= 0
ad	r0+r1−r6−r9	>= 0
adc	r0+r1+r4−r6−r9−r8	>= 0
adce	r0+r1+r4+r3−r6−r9−r8−r10	>= 0

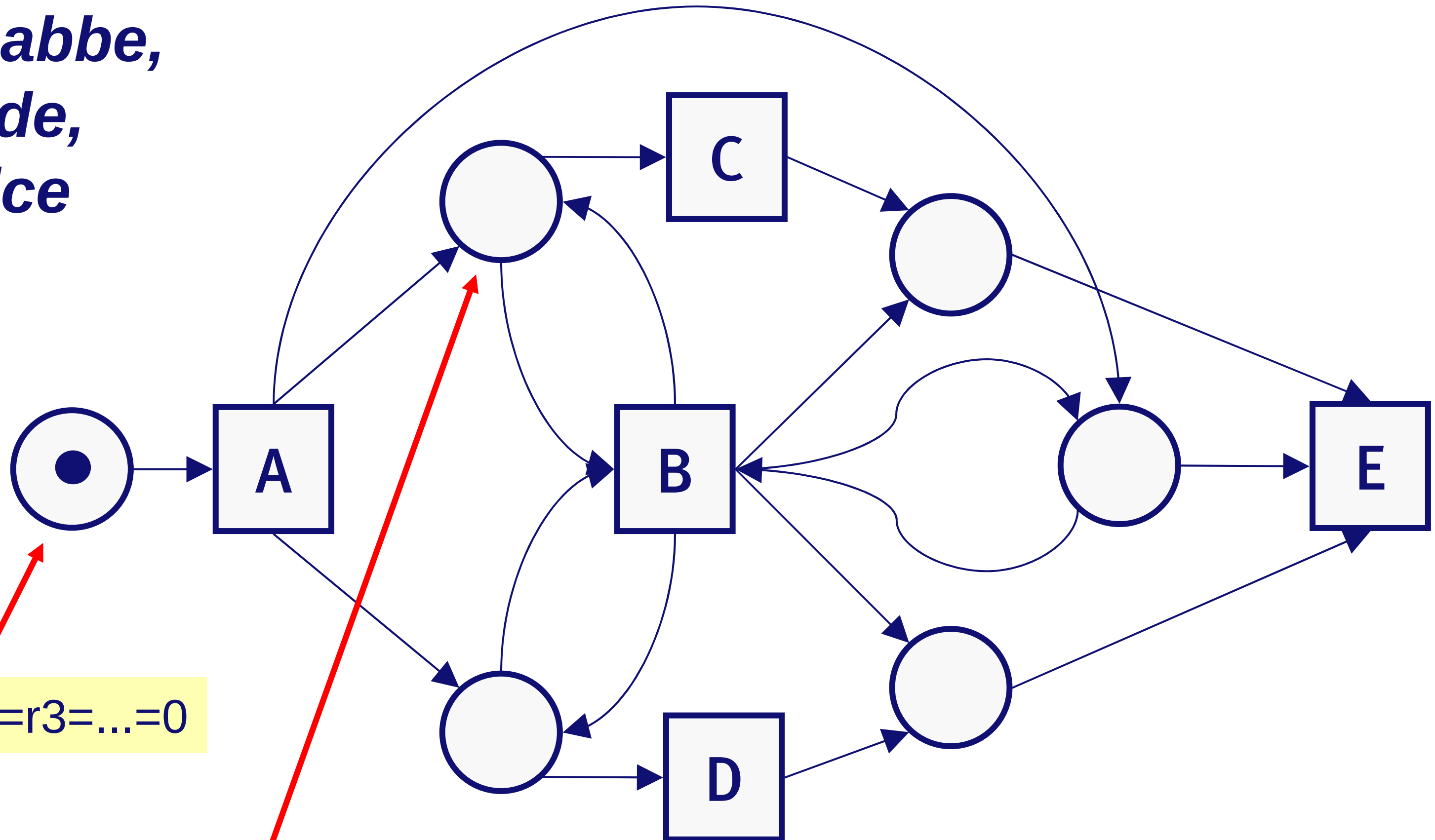


r0=r6=1 and r1=r2=r3=...=0 is an example solution and hence a possible place.

r1=r2=r8=r7=1 and r0=r3=r4=...=0 is an example solution and hence a possible place.

Result of Integer Linear Programming
(multiple formulations possible)

a, ab, abb, abbe,
ac, acd, acde,
ad, adc, adce



r0=r6=1 and r1=r2=r3=...=0

r1=r2=r8=r7=1 and r0=r3=r4=...=0

Customizable and tunable

- There are infinitely many places, but the selection of places to be added can be controlled.
- The ILP formulation can be used to search for subclasses (marked graph, state machine, free-choice, etc.) or to avoid showing "complex" places.
- The ILP formulation can be used to take frequencies into account.

Summary

- Alpha miner
- Multi phase miner
- Genetic process mining
- State-based region mining
- Language based region mining

Many more:

- Fuzzy miner
- Heuristics miner
- Alpha+, Alpha++, Alpha #, etc.
- ...

Balancing Between Overfitting and Underfitting

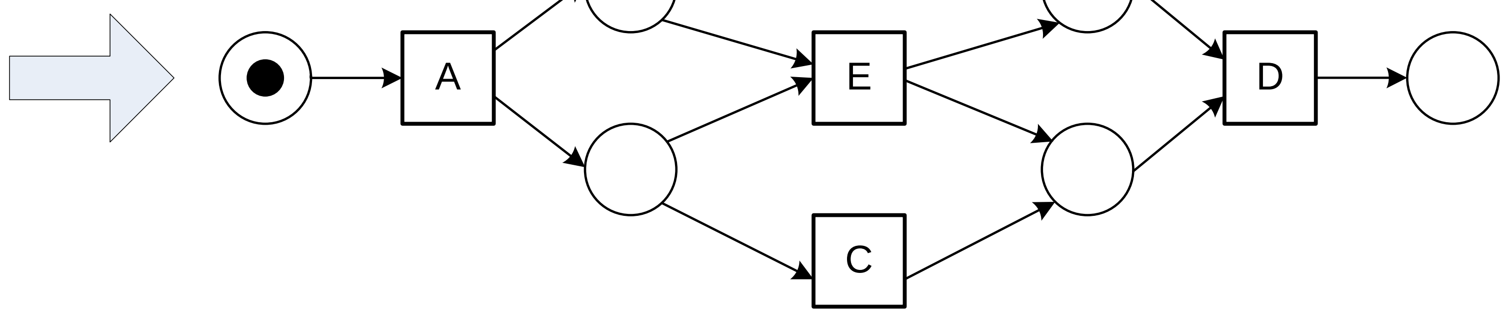
Technische Universiteit Eindhoven
University of Technology

Where innovation starts

Challenge: Balancing Between Underfitting and Overfitting

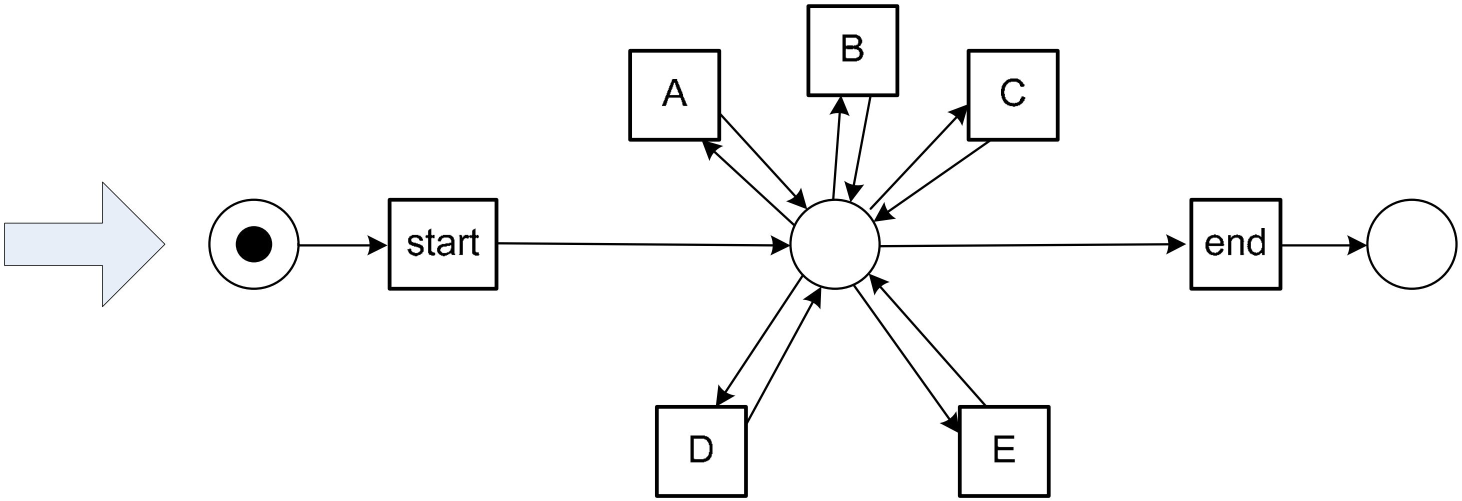
The essence

ABCD
ACBD
AED
ABCD
ABCD
AED
ACBD
...

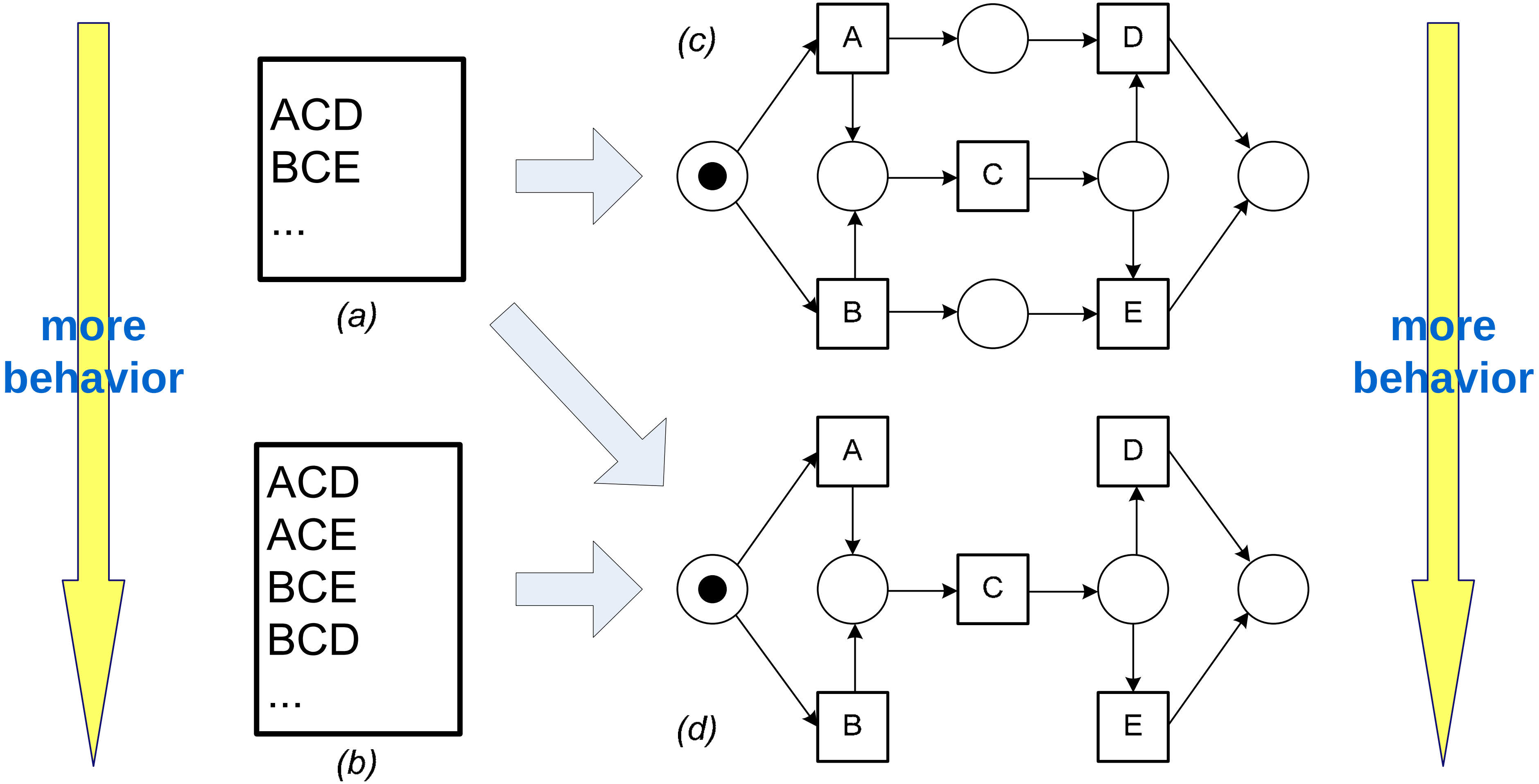


But ...

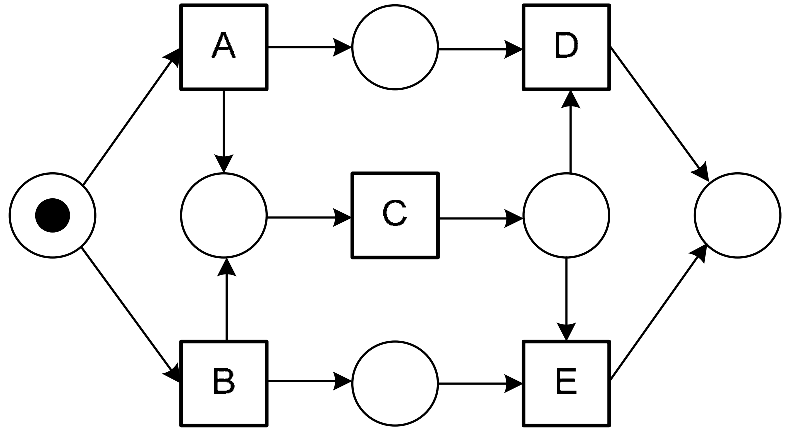
Any log containg activities A, B, C, D, and E.

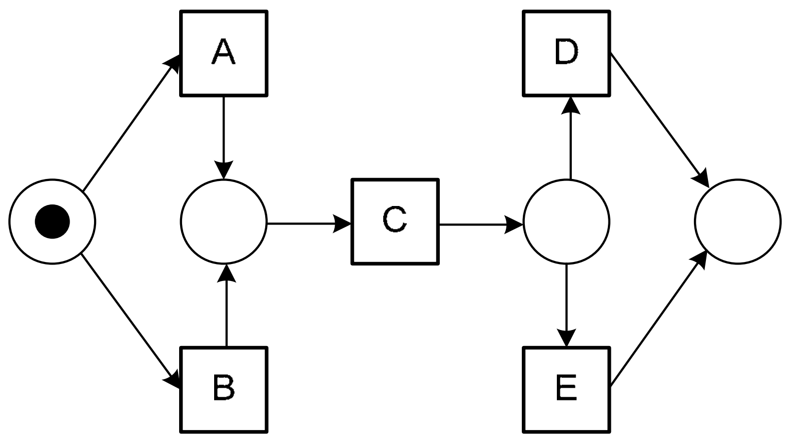


Finding a balance

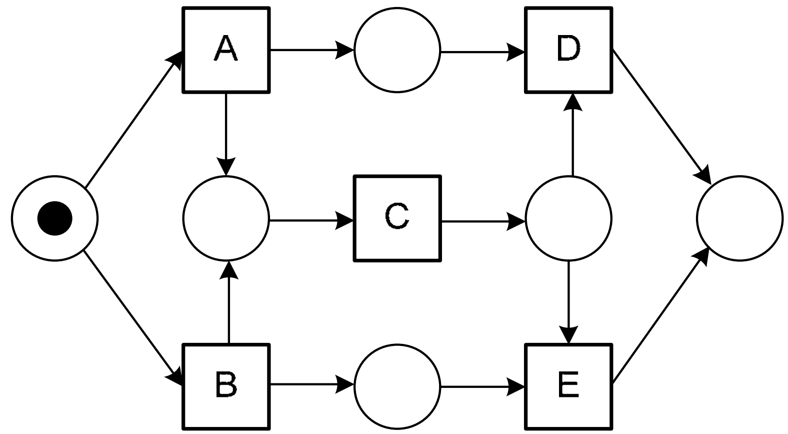


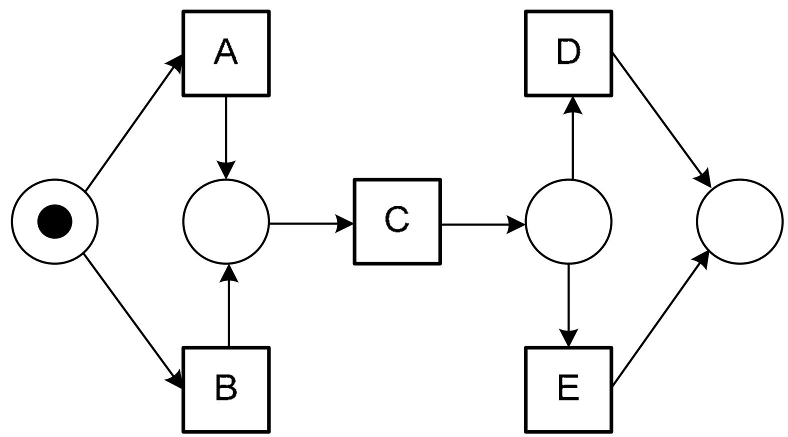
ACD	99
ACE	0
BCE	85
BCD	0



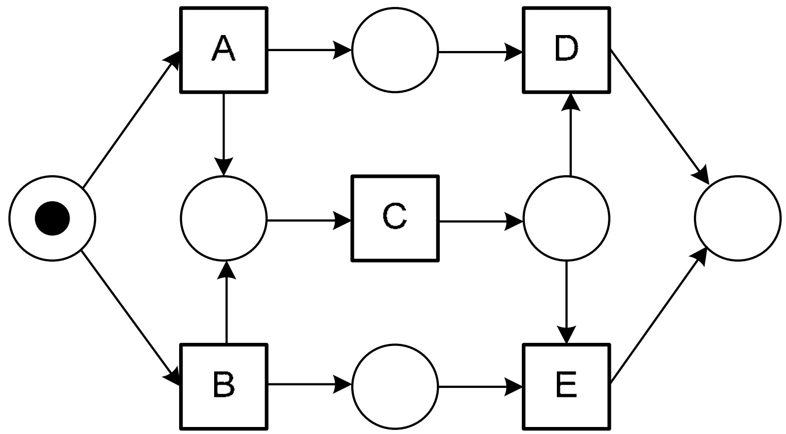


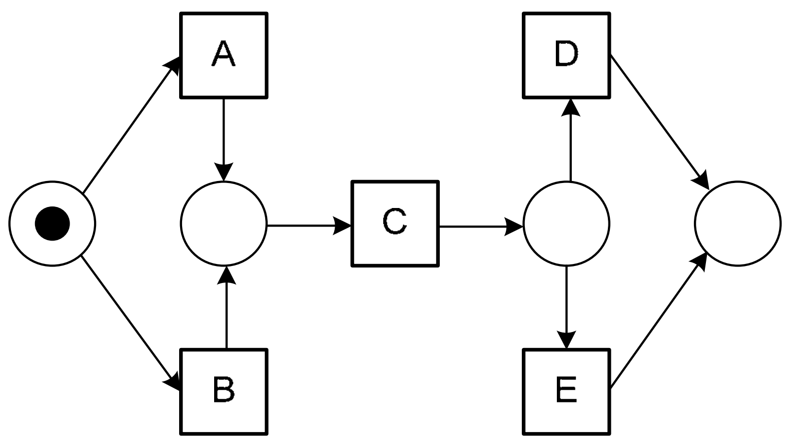
ACD 99
ACE 88
BCE 85
BCD 78





ACD	**99**
ACE	**2**
BCE	**85**
BCD	**3**

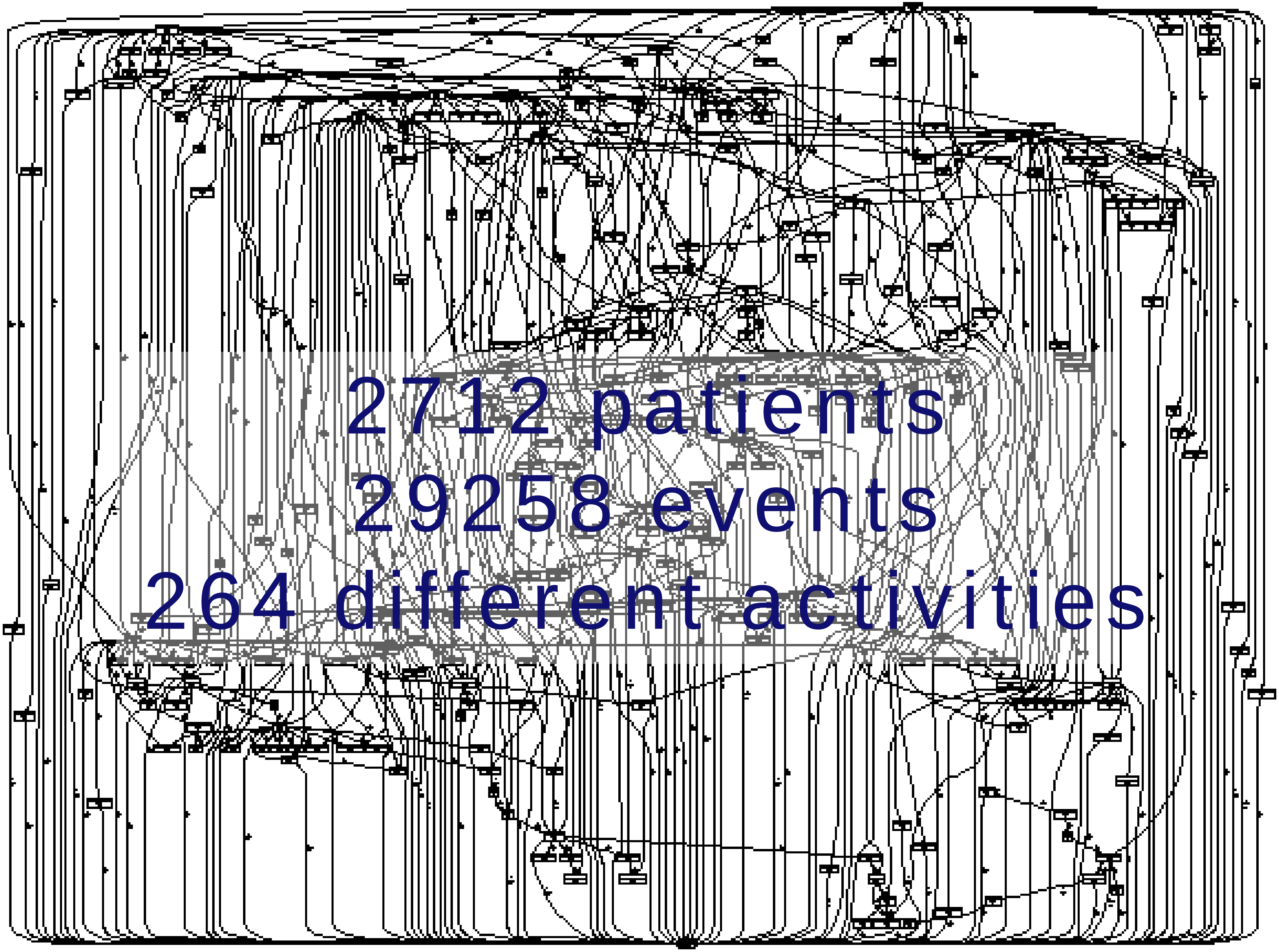


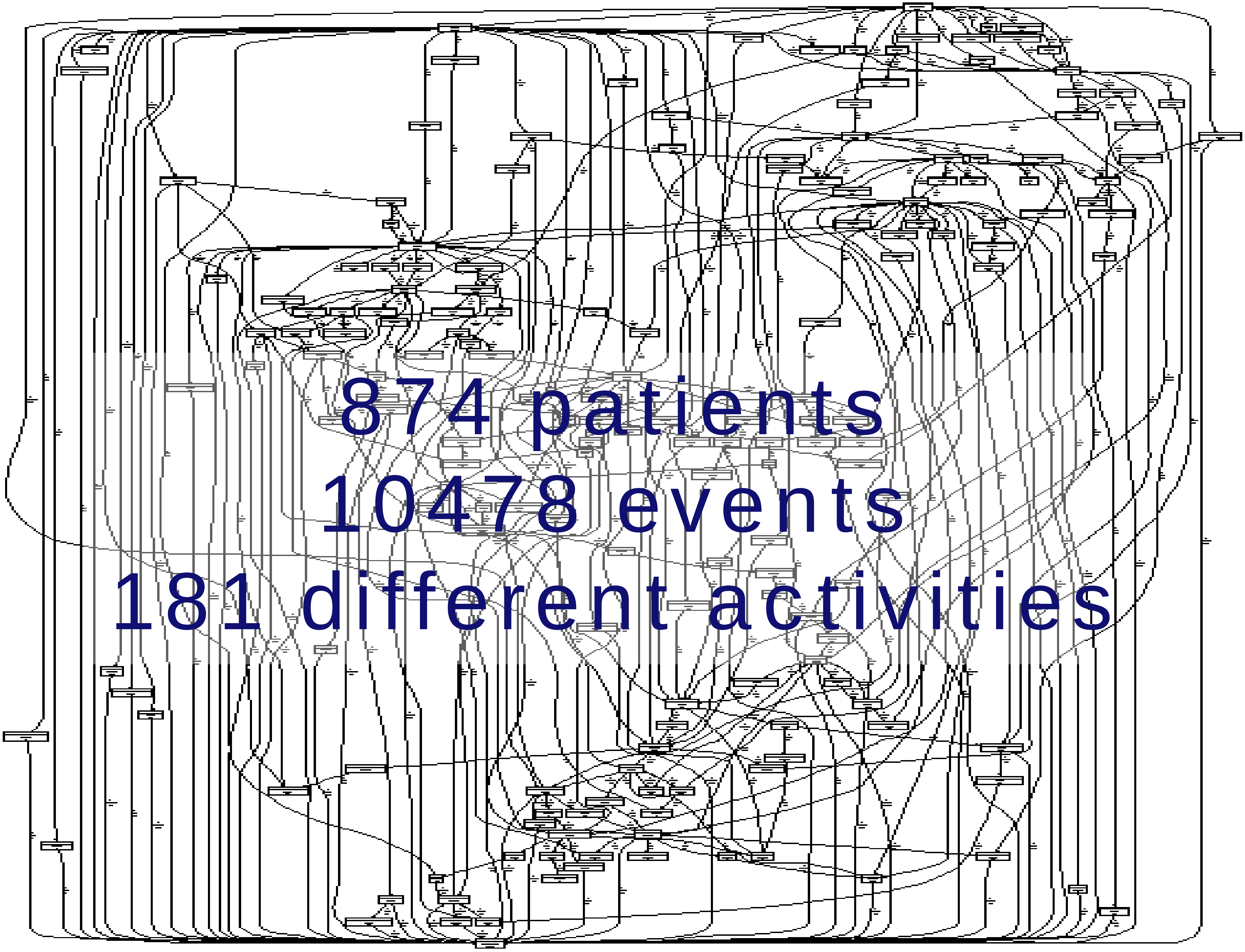


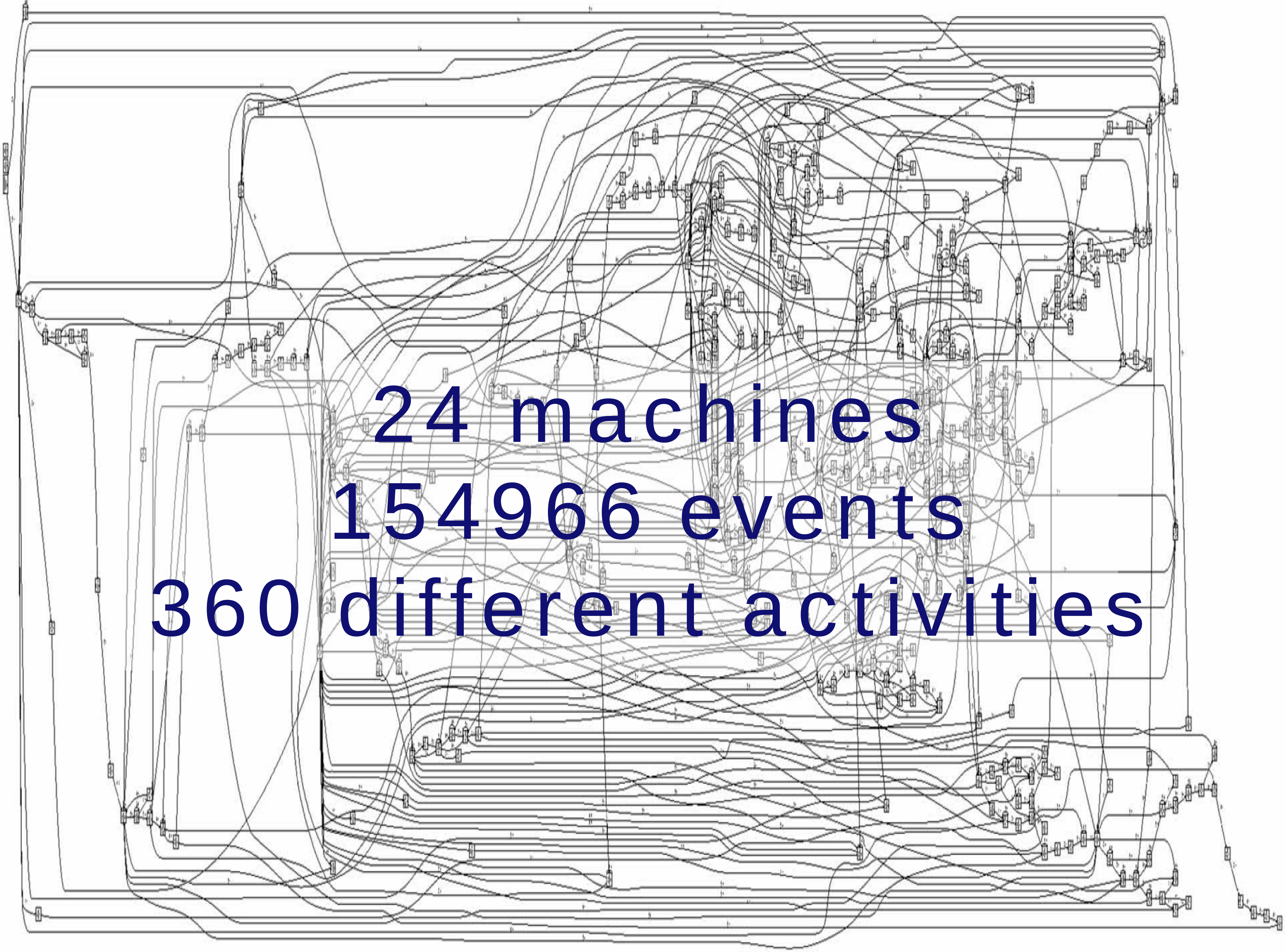
Important observations

- **Frequencies matter!**
- **Adding a place equals restricting behavior!**
- **"The model" does not exist!**

Relevance


2712 patients
29258 events
264 different activities


874 patients
10478 events
181 different activities



24 machines

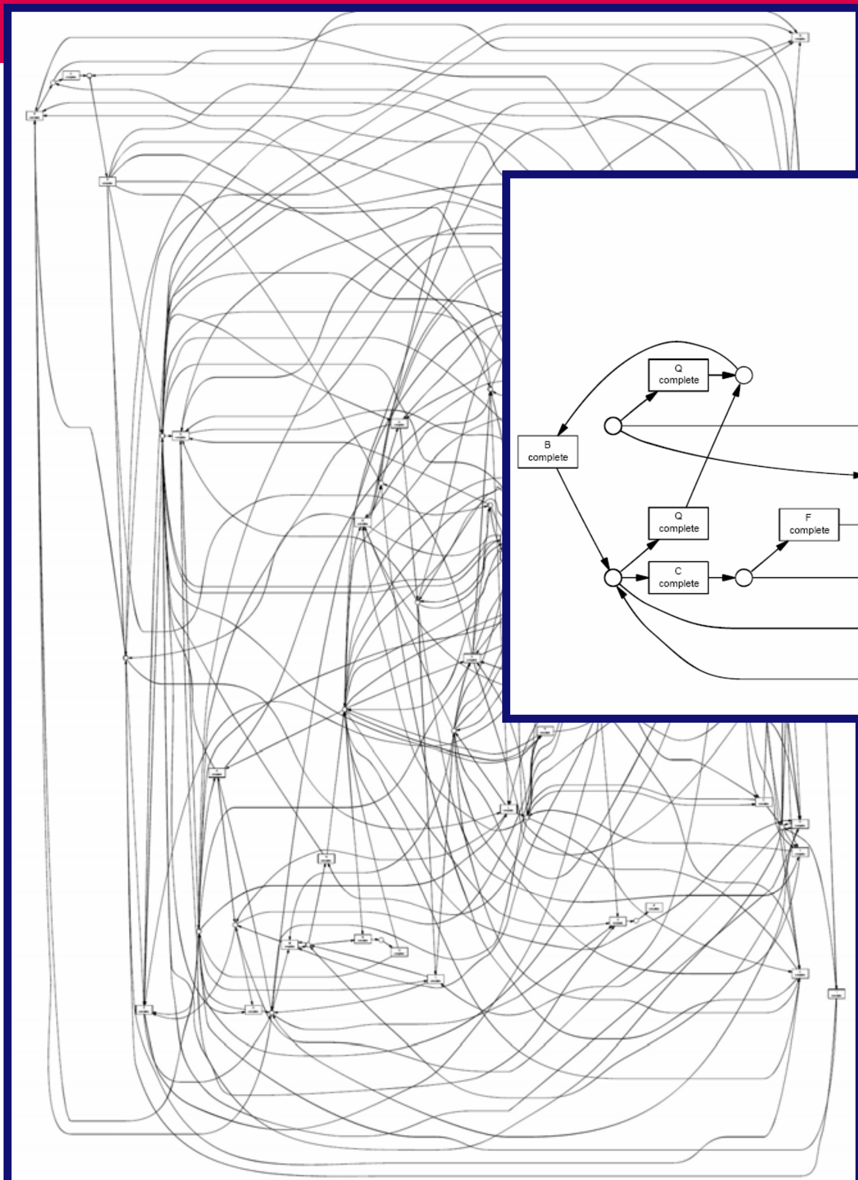
154966 events

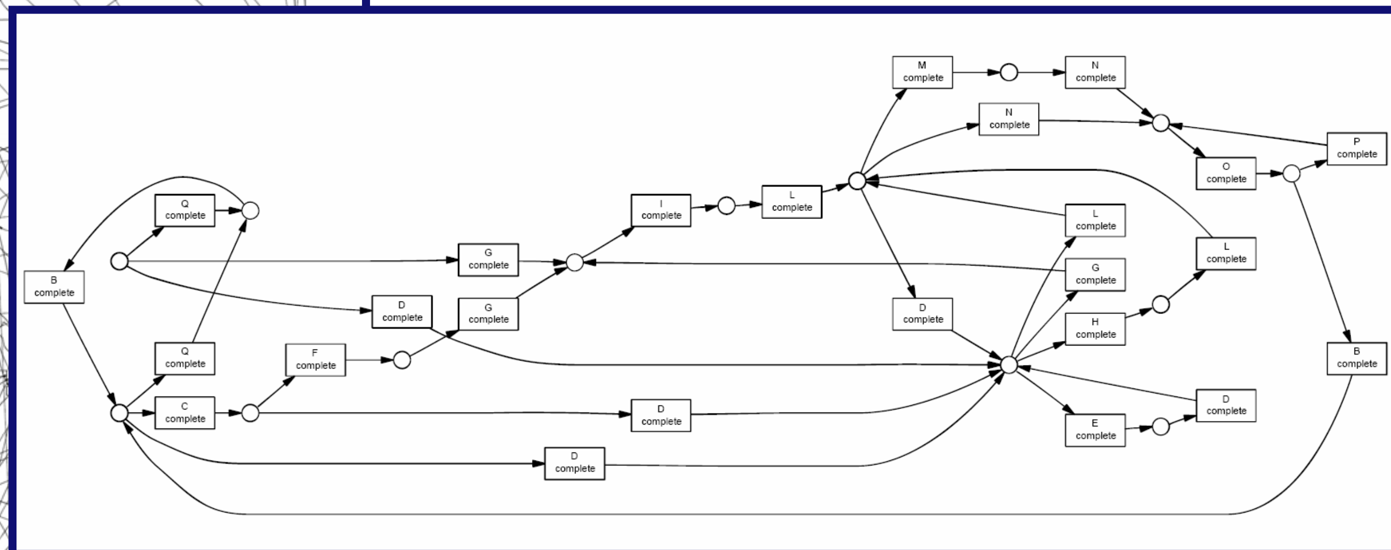
360 different activities

Problems

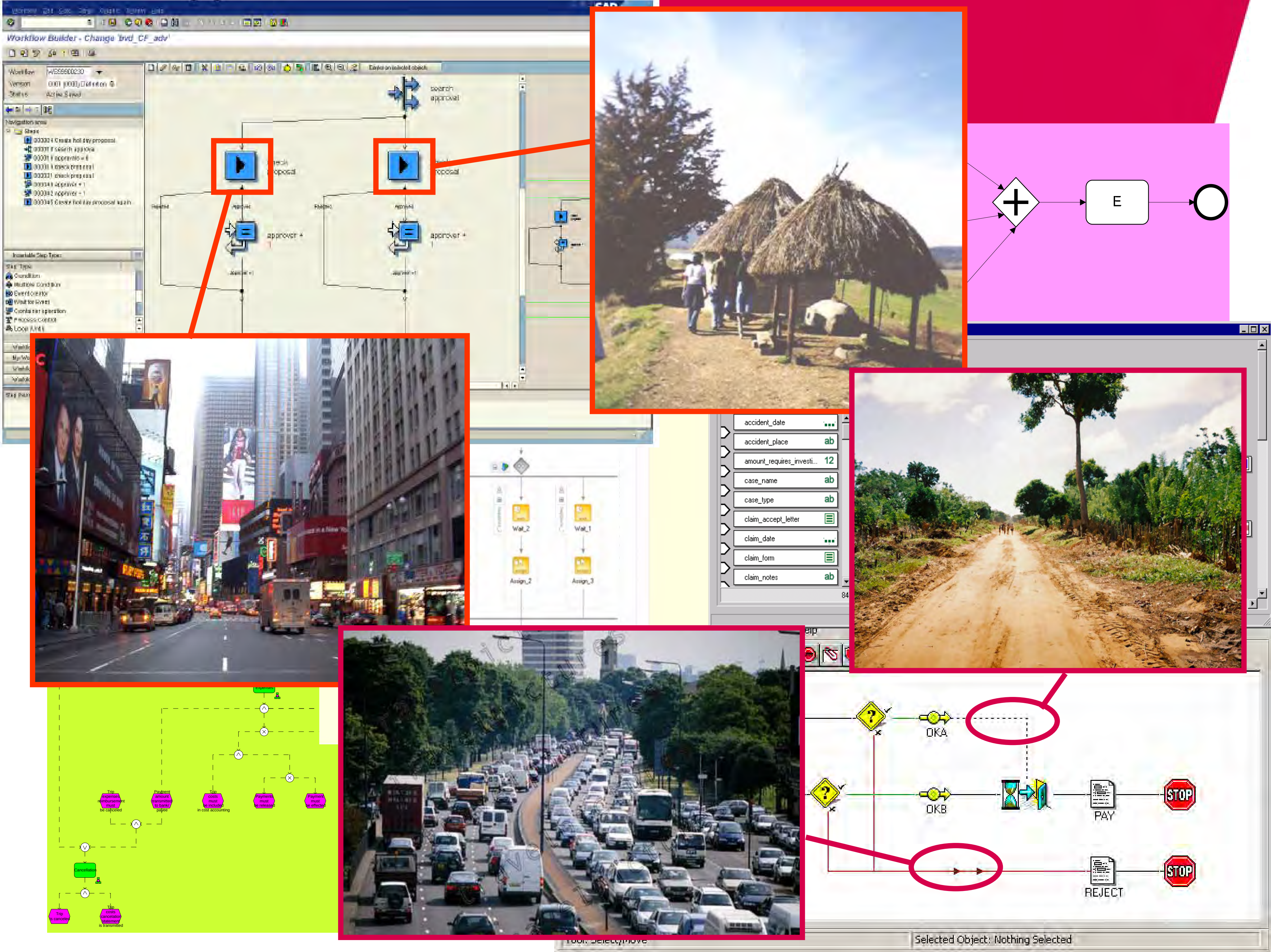
- Representational bias (i.e., generalization is driven by representation rather than log or preferences).
- Inability of dealing with or detecting noise.
- Wrong abstraction level.
- Limitation of current process modeling (visualization) techniques.

Example: Heusden

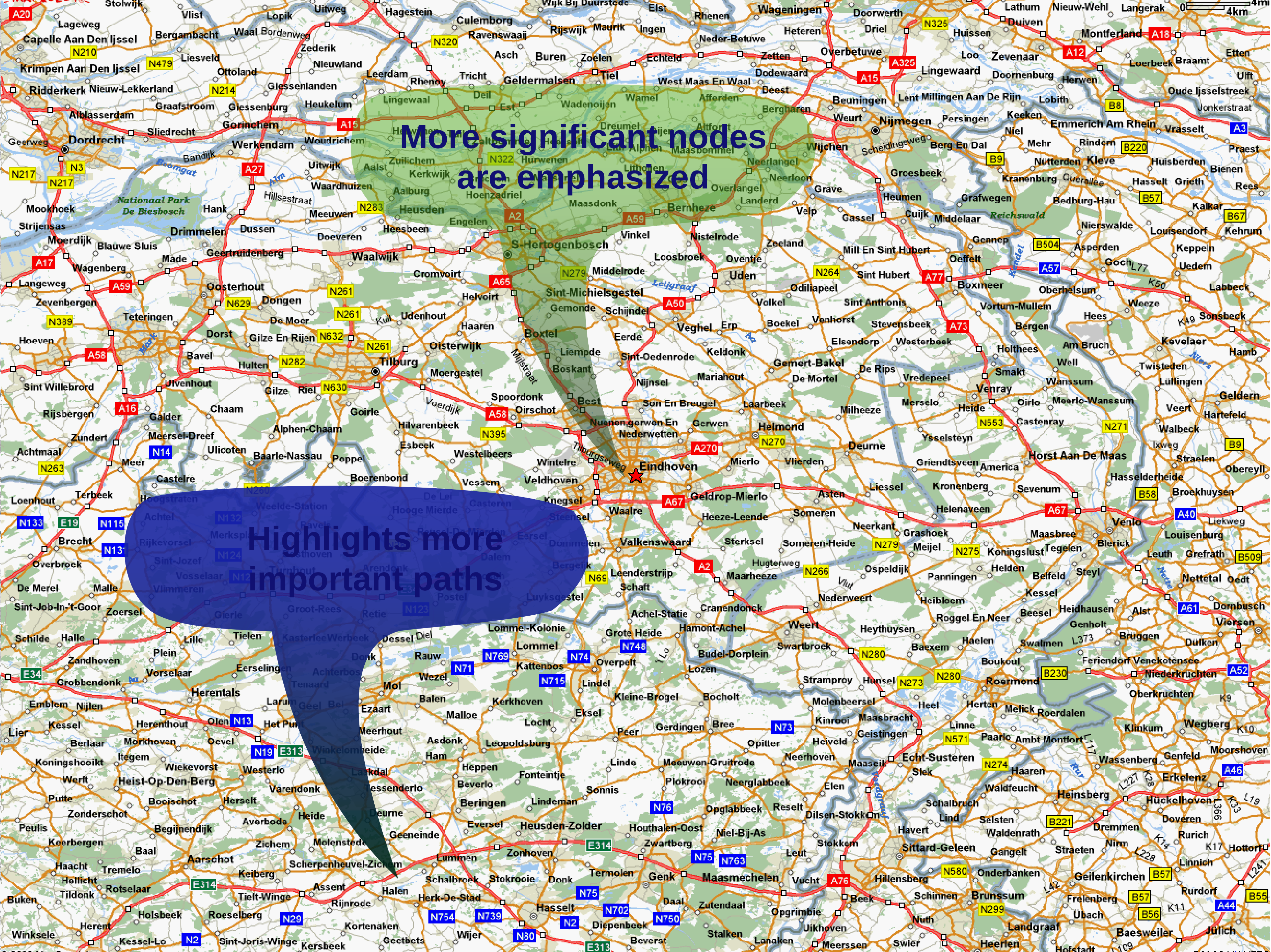




State horizon set to last activity.



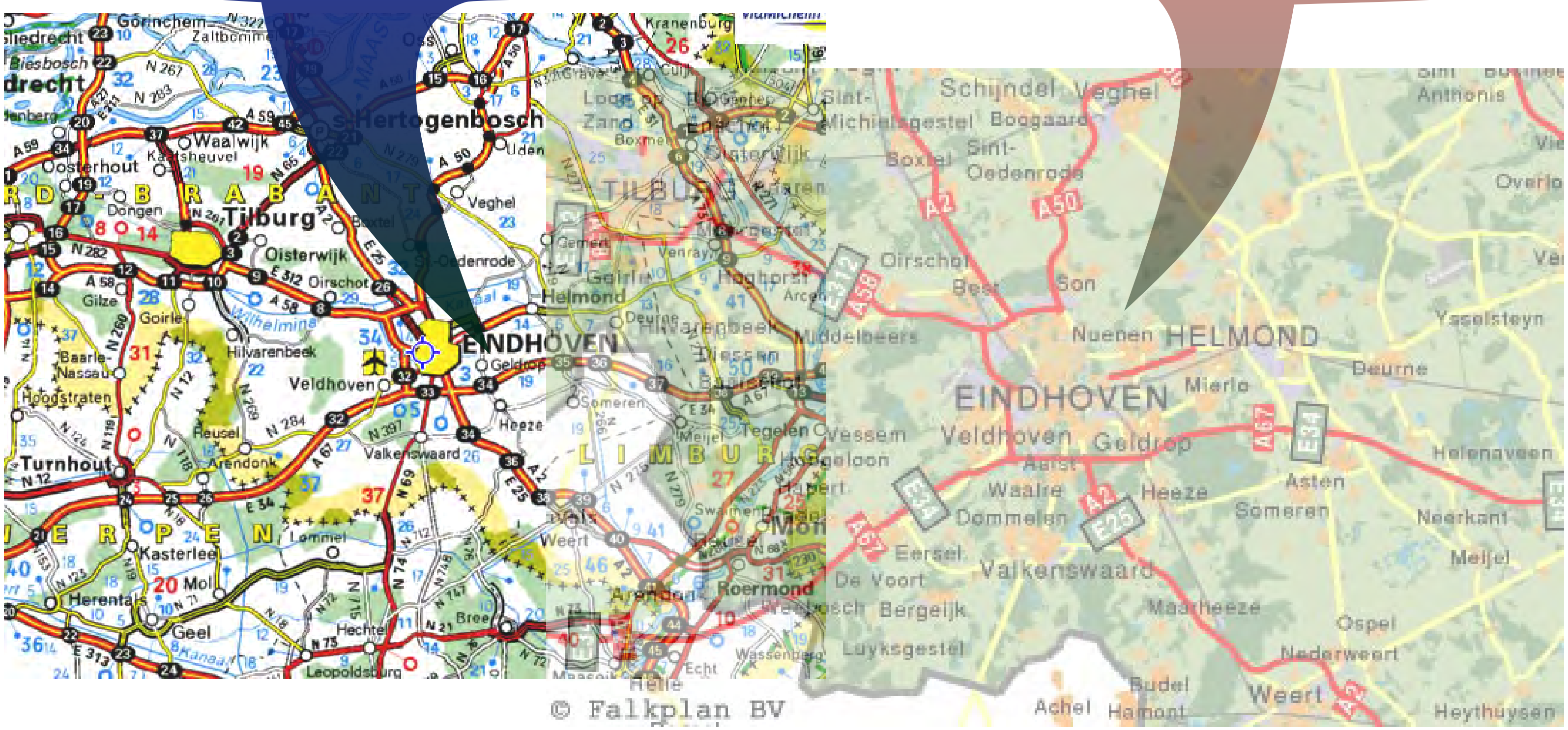
Workflow Builder - Change 'bvd_CF_adv'
search approval
approver +
E
Wait_2
Wait_1
Assign_2
Assign_3
accident_date
accident_place
amount_requires_investi...
case_name
case_type
claim_accept_letter
claim_date
claim_form
claim_notes
OKA
OKB
PAY
STOP
REJECT
Selected Object: Nothing Selected


More significant nodes are emphasized
Highlights more important paths

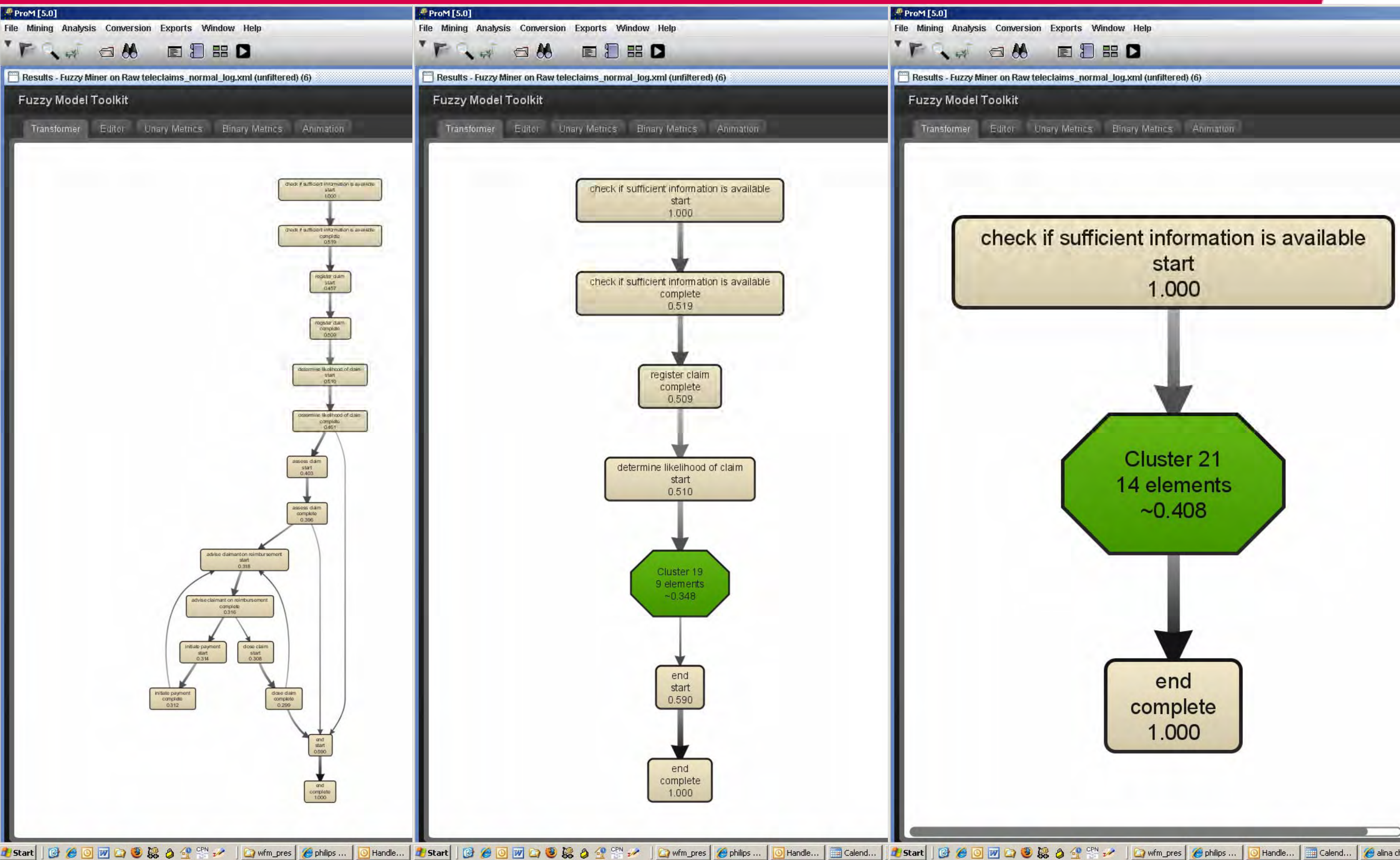
More to learn from maps...

Aggregation
Clustering of coherent, less significant structures

Abstraction
Removing isolated, less significant structures

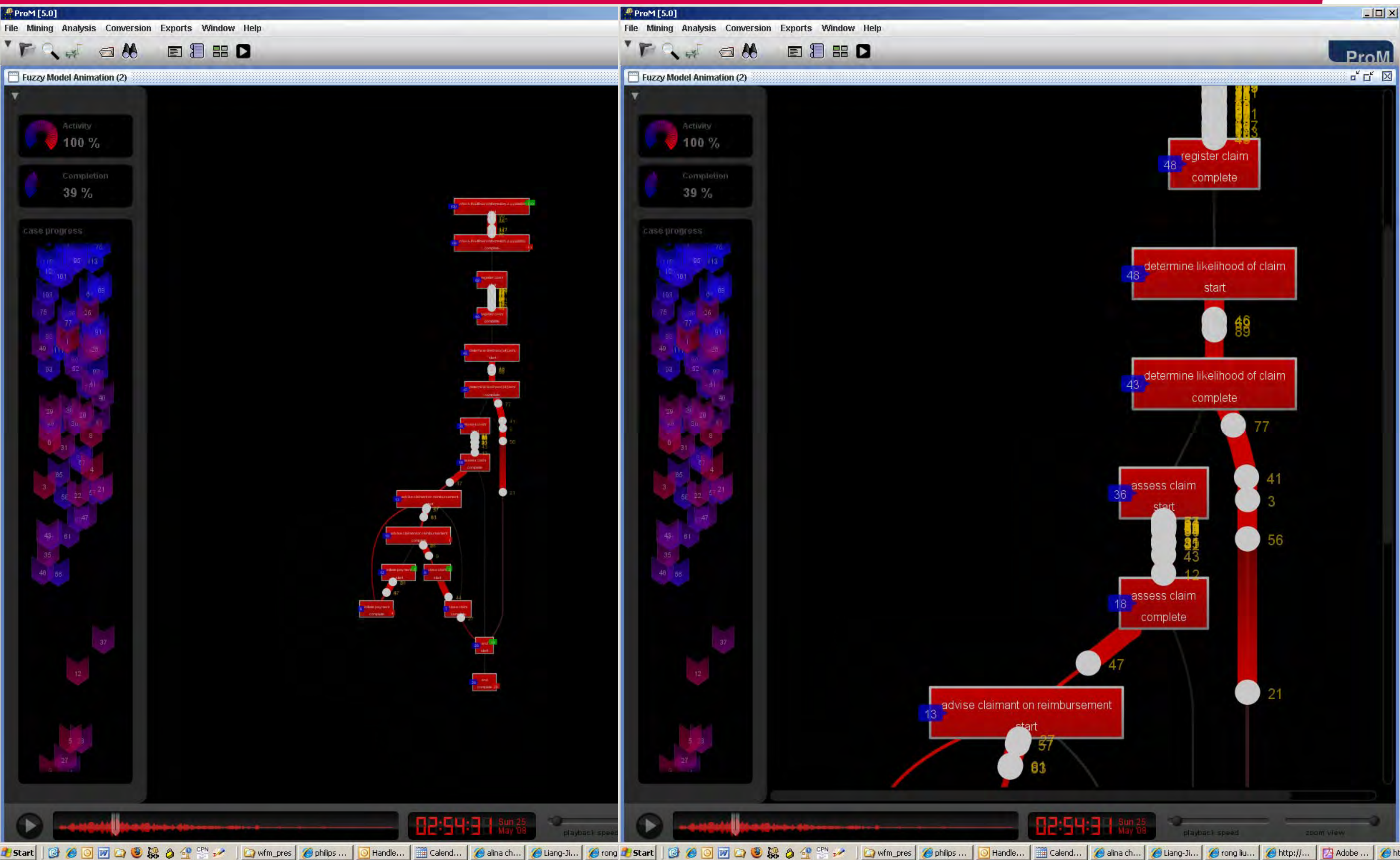


Fuzzy miner



ProM [5.0]
File Mining Analysis Conversion Exports Window Help
Results - Fuzzy Miner on Raw teleclaims_normal_log.xml (unfiltered) (6)
Fuzzy Model Toolkit
Transformer Editor Unary Metrics Binary Metrics Animation
check if sufficient information is available start 1.000
check if sufficient information is available complete 0.519
register claim complete 0.509
determine likelihood of claim start 0.510
Cluster 19 9 elements ~0.348
end start 0.590
end complete 1.000
Cluster 21 14 elements ~0.408
Start wfm_pres philips ... Handle... Calend...

Showing reality



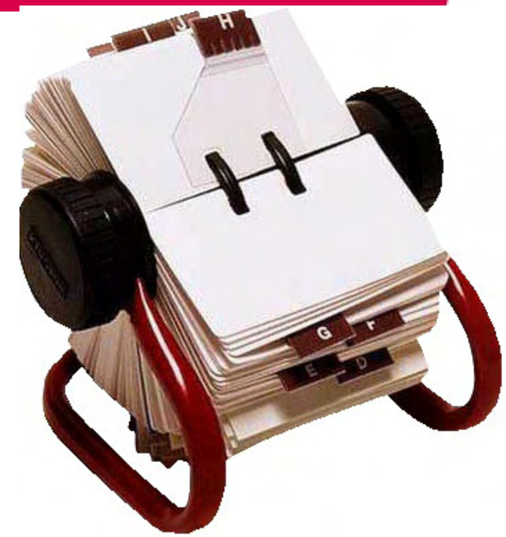
Discovering Other Perspectives

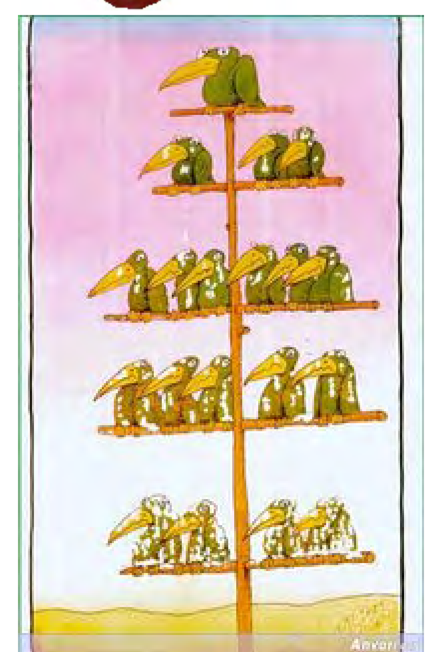
TU/e Technische Universiteit **Eindhoven** University of Technology

Where innovation starts

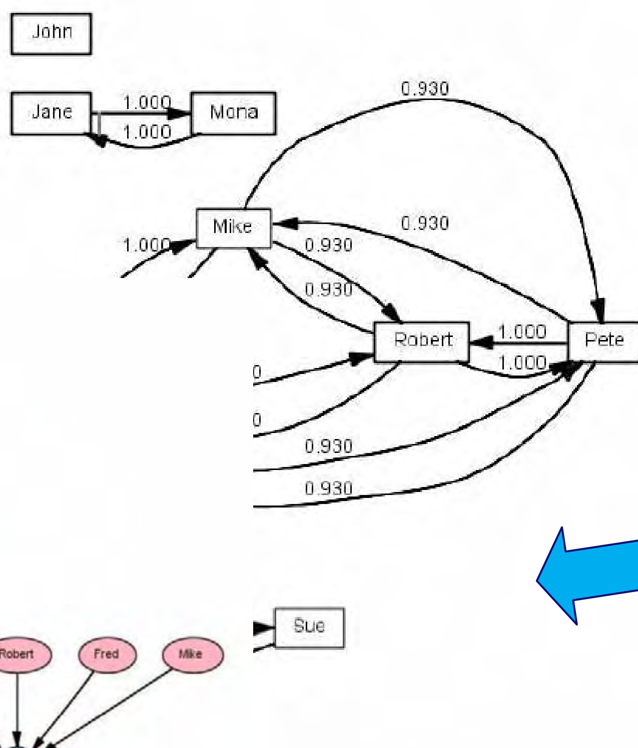
Perspectives

- Control-flow perspective
  - As before ...
- Data perspective
  - How does data flow from one task to another?
  - What data is influencing decisions?
  - What are the (data-driven) business rules?
- Organizational perspective
  - Who is doing what?
  - Who is working with who?
  - What are the (real) roles in an organization?
- ...

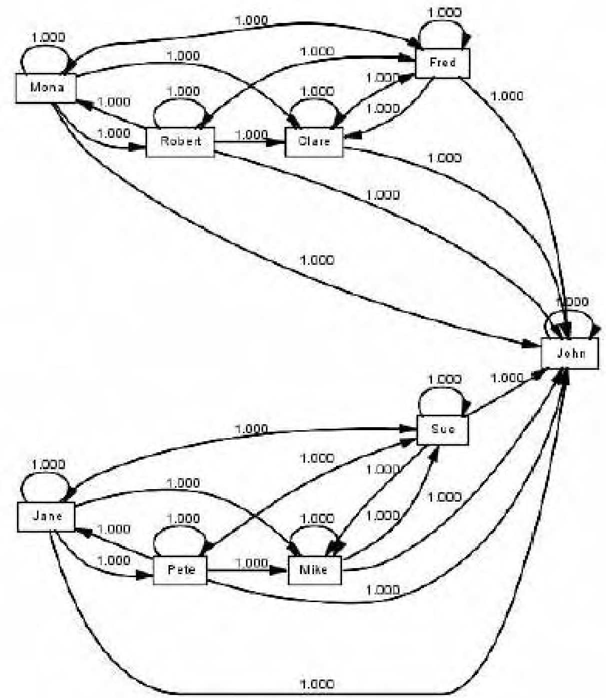




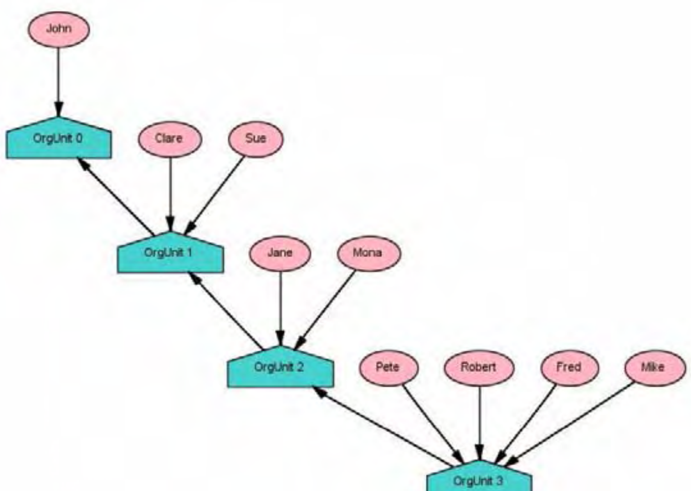
Examples of social network mining
(Minseok Song et al.)



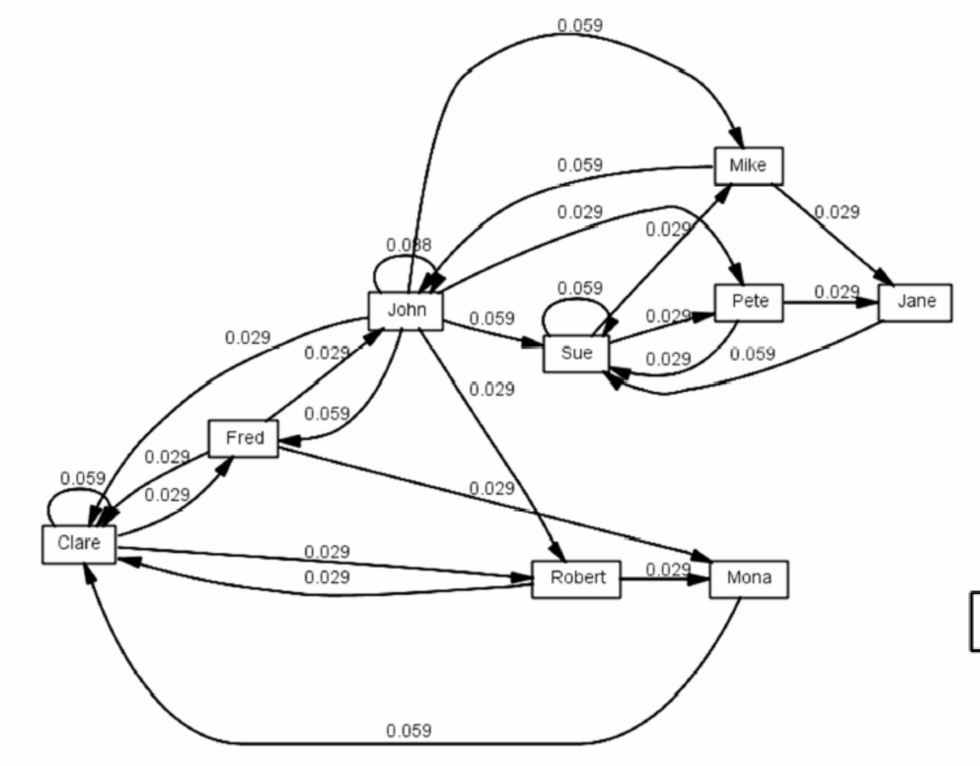
oing similar task

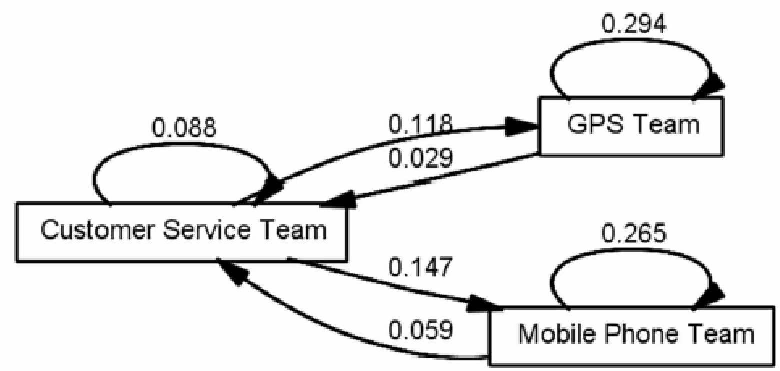


(b) working together

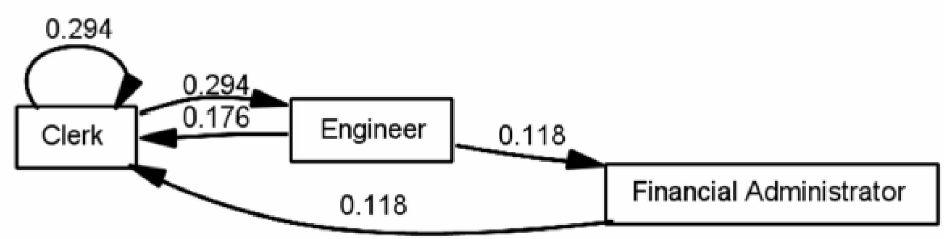


Social networks based on hand-over of work



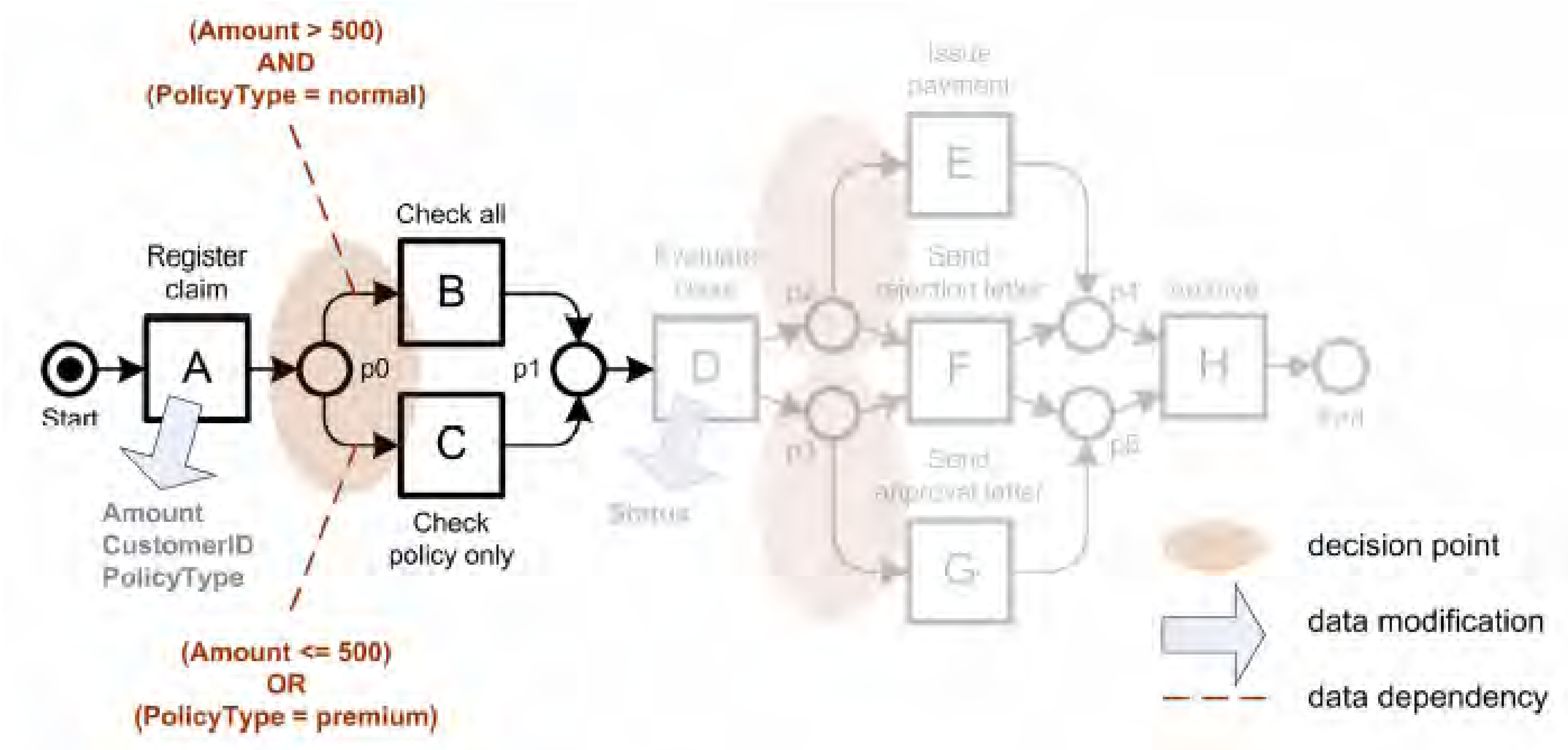


(a) social network for organizational units



(b) social network for roles

Decision mining
(Anne Rozinat et al.)

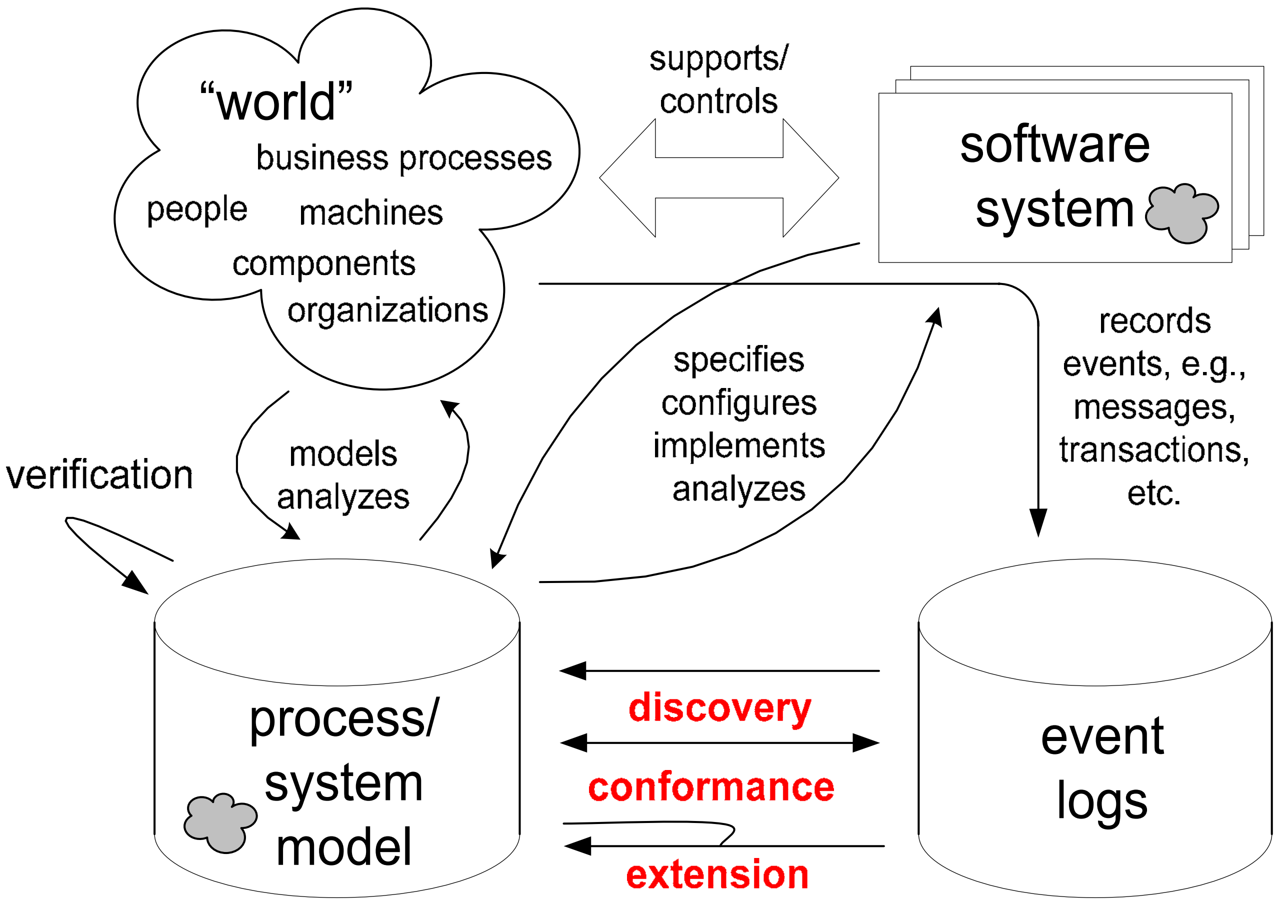


Conformance Checking and Extension

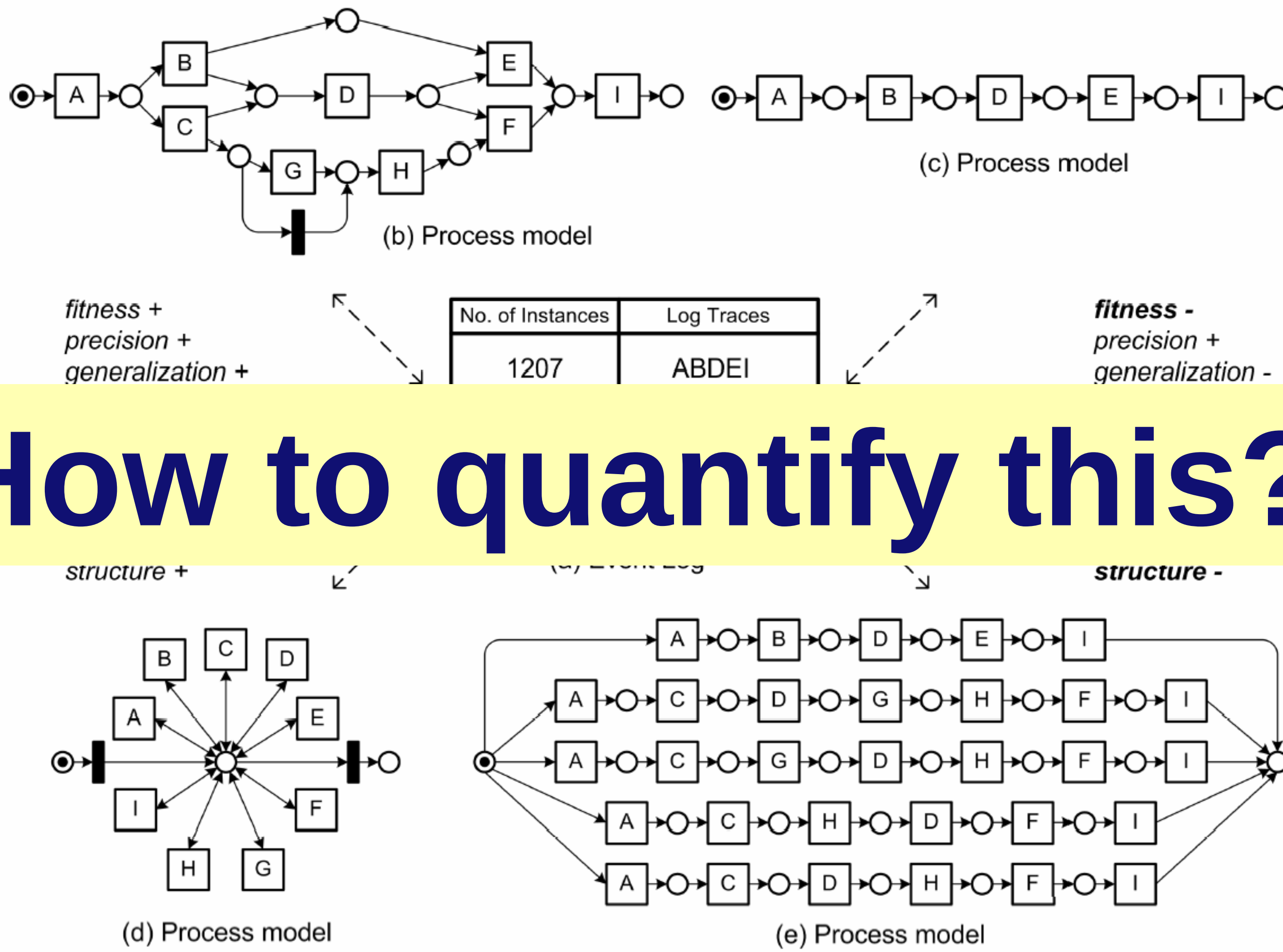
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Conformance and Extension



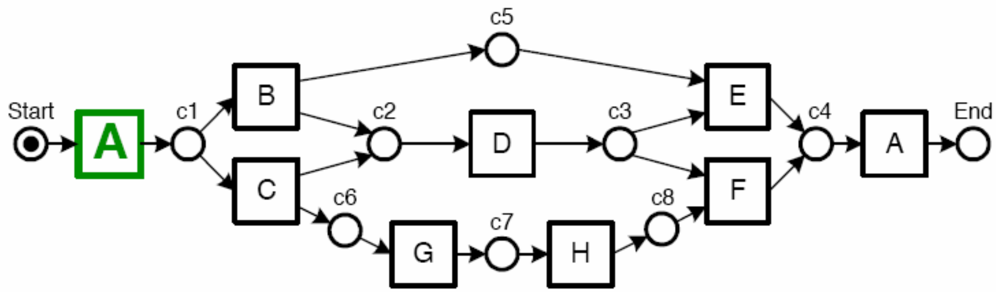
Conformance checker
(Anne Rozinat et al.)



How to quantify this?

Fitness by replay

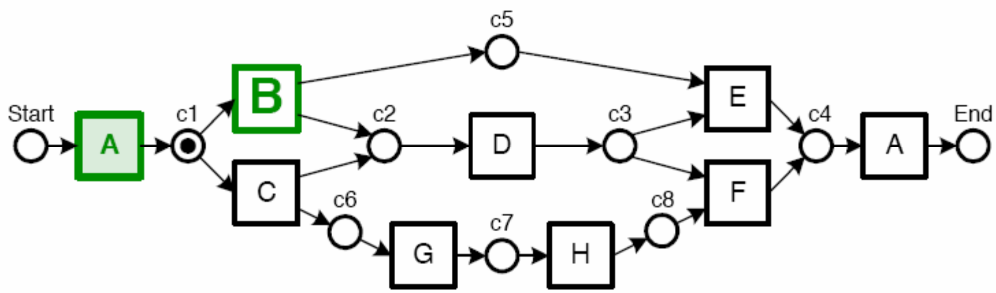
(a)



m = 0
r = 0
c = 0
p = 1

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

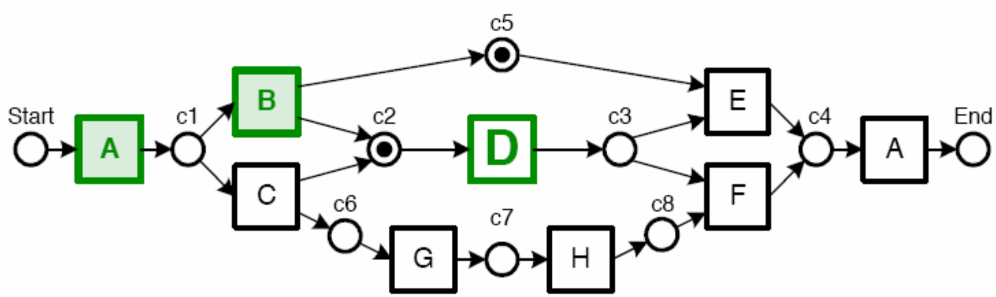
(b)



m = 0
r = 0
c = 1
p = 2

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

(c)



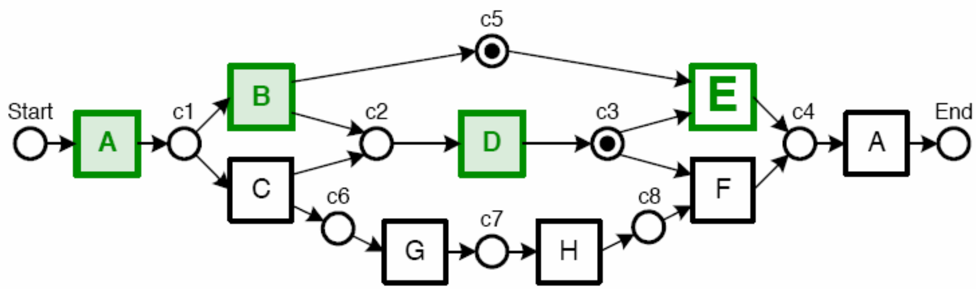
m = 0
r = 0
c = 2
p = 4

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

m=missing,r=remaining,c=consumed,p=produced

No problem (m=0, r=0)

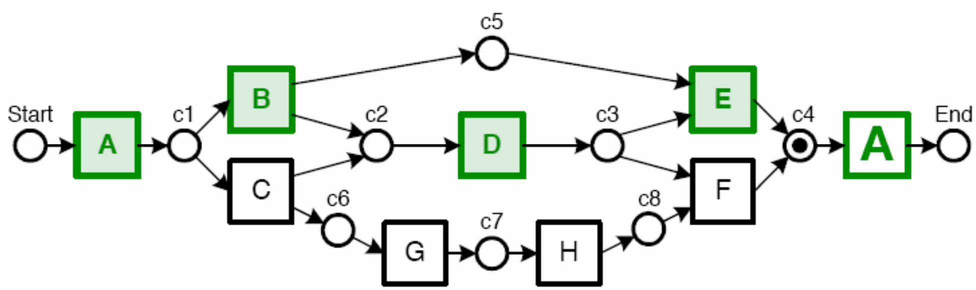
(d)



m = 0
r = 0
c = 3
p = 5

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

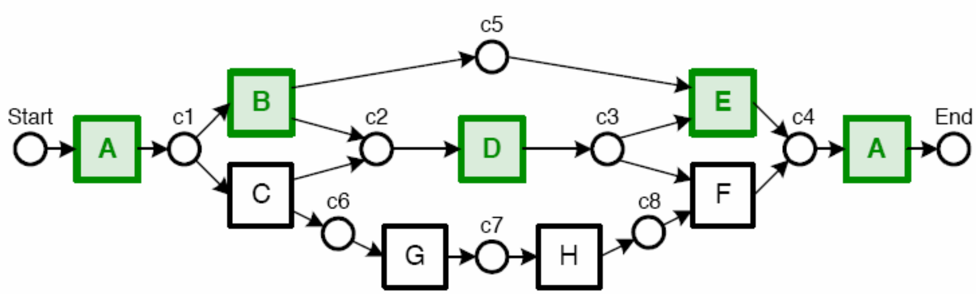
(e)



m = 0
r = 0
c = 5
p = 6

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

(f)

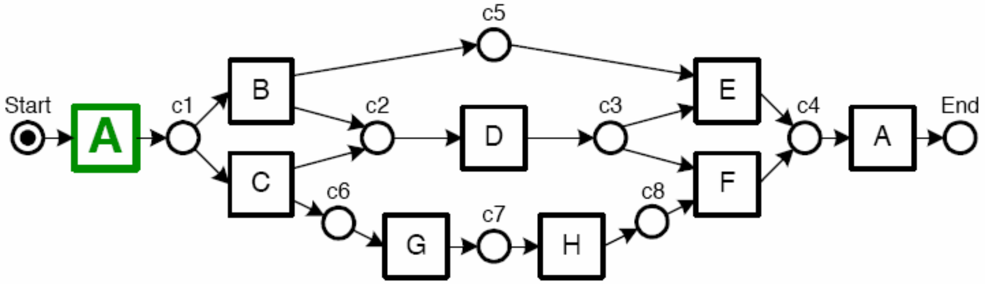


m = 0
r = 0
c = 7
p = 7

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

Another (impossible) trace

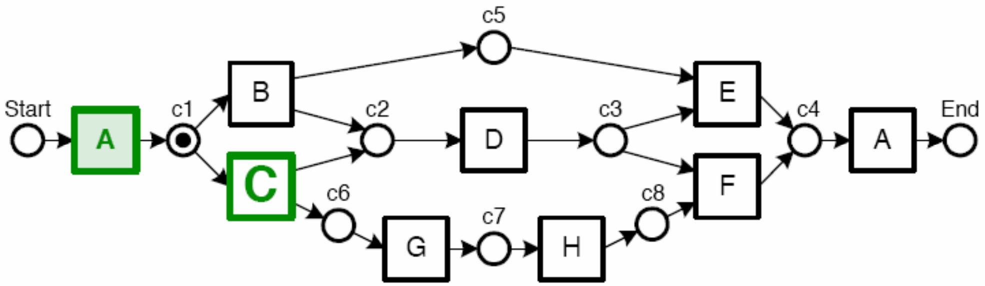
(a)



m = 0
r = 0
c = 0
p = 1

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

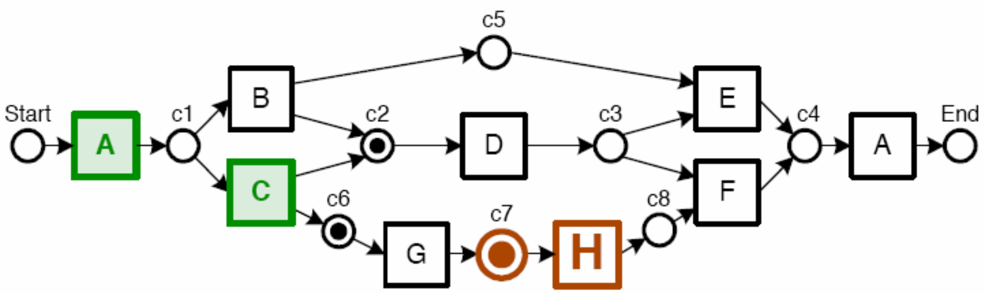
(b)



m = 0
r = 0
c = 1
p = 2

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

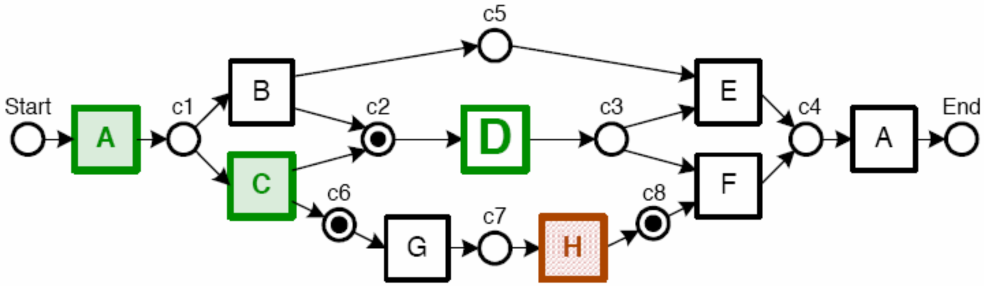
(c)



m = 1
r = 0
c = 2
p = 4

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

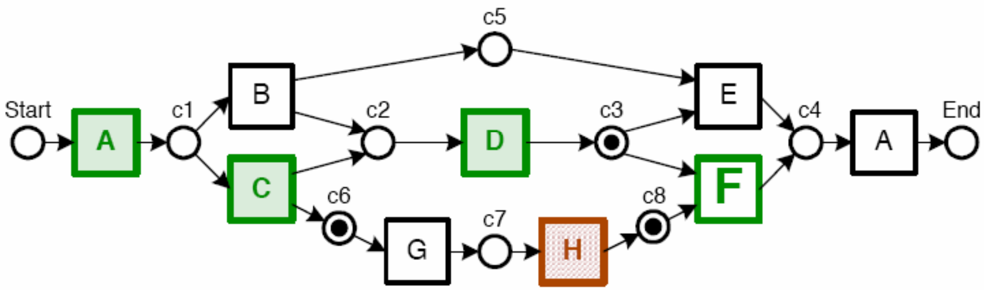
(d)



m = 1
r = 0
c = 3
p = 5

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

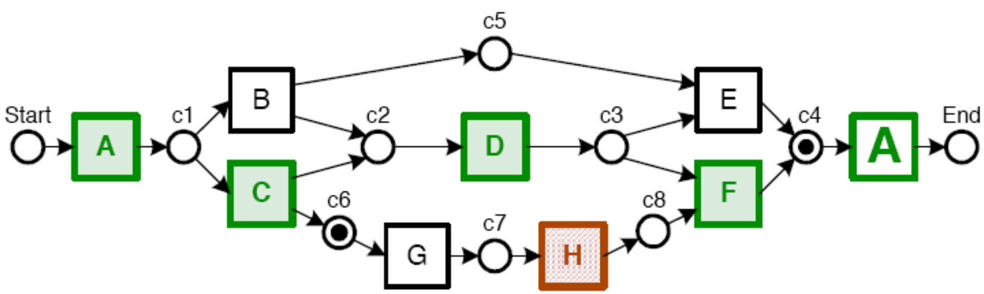
(e)



m = 1
r = 0
c = 4
p = 6

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

(f)

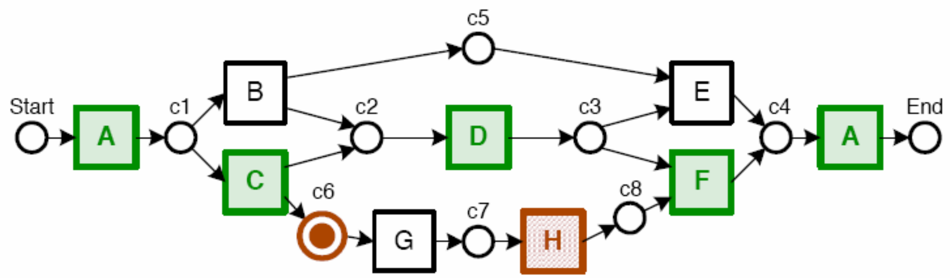


m = 1
r = 0
c = 6
p = 7

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

Fitness calculation

(g)



m = 1
r = 1
c = 8
p = 8

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	**ACHDFA**
28	ACDHFA

$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

$$f(M1, L2) = \frac{1}{2}\left(1 - \frac{51}{10666}\right) + \frac{1}{2}\left(1 - \frac{51}{10666}\right) \approx 0.995$$

Examples

No. of Instances	Log Traces
4070	ABDEA
245	ACDGHFA
56	ACGDHFA

(a) Event Log L1

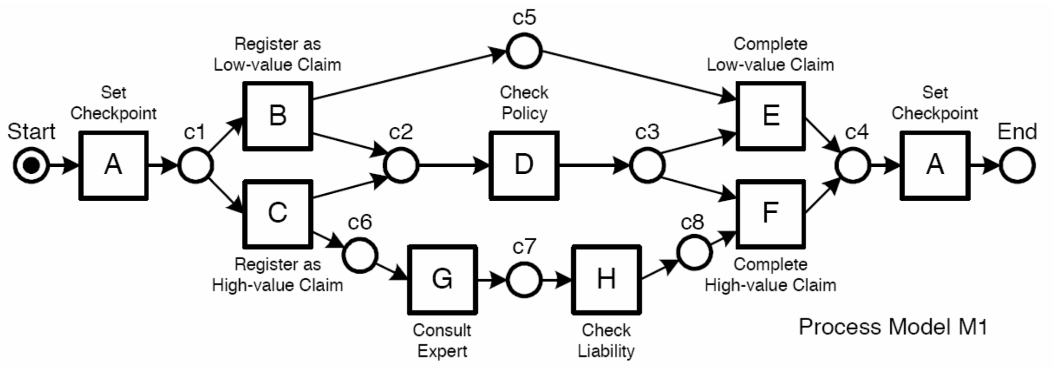
No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

(b) Event Log L2

No. of Instances	Log Traces
24	BDE
7	AABHF
15	CHF
6	ADBE
1	ACBGDFAA
8	ABEDA

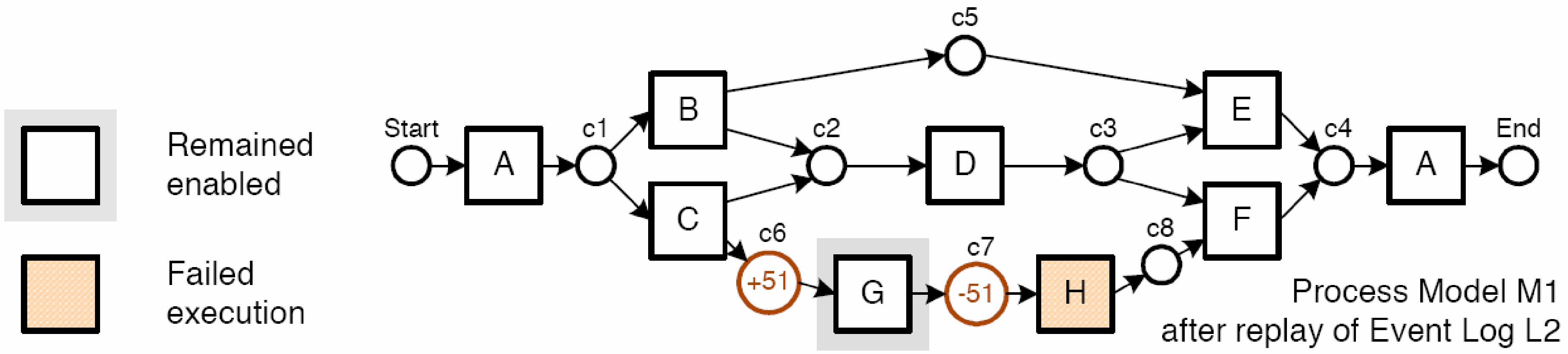
(c) Event Log L3

f=1.000 **f=0.995** **f=0.540**



Process Model M1

Diagnostics

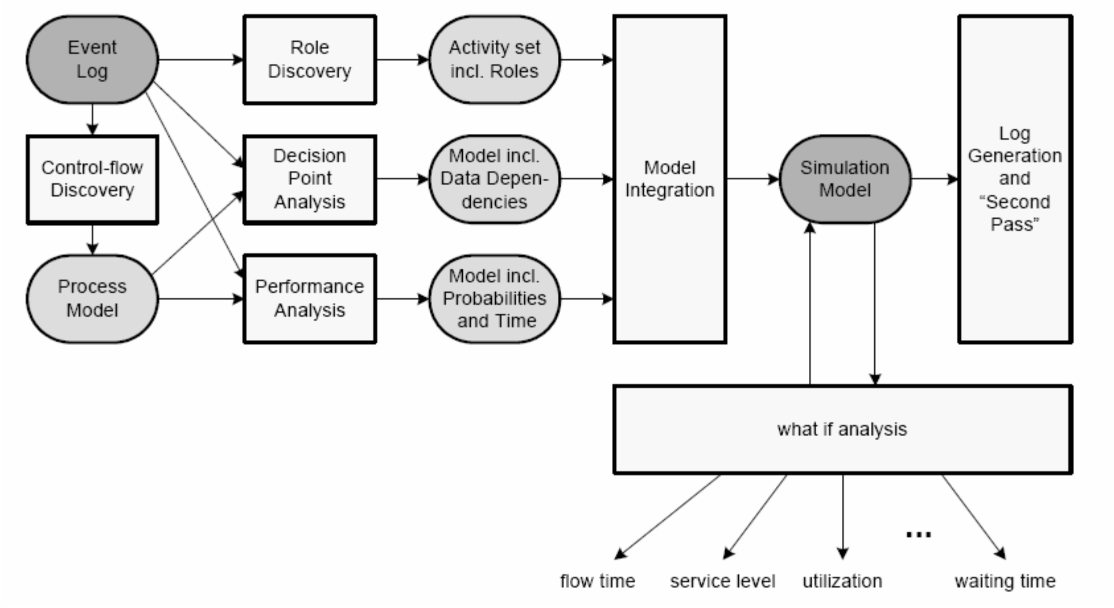


Other Metrics

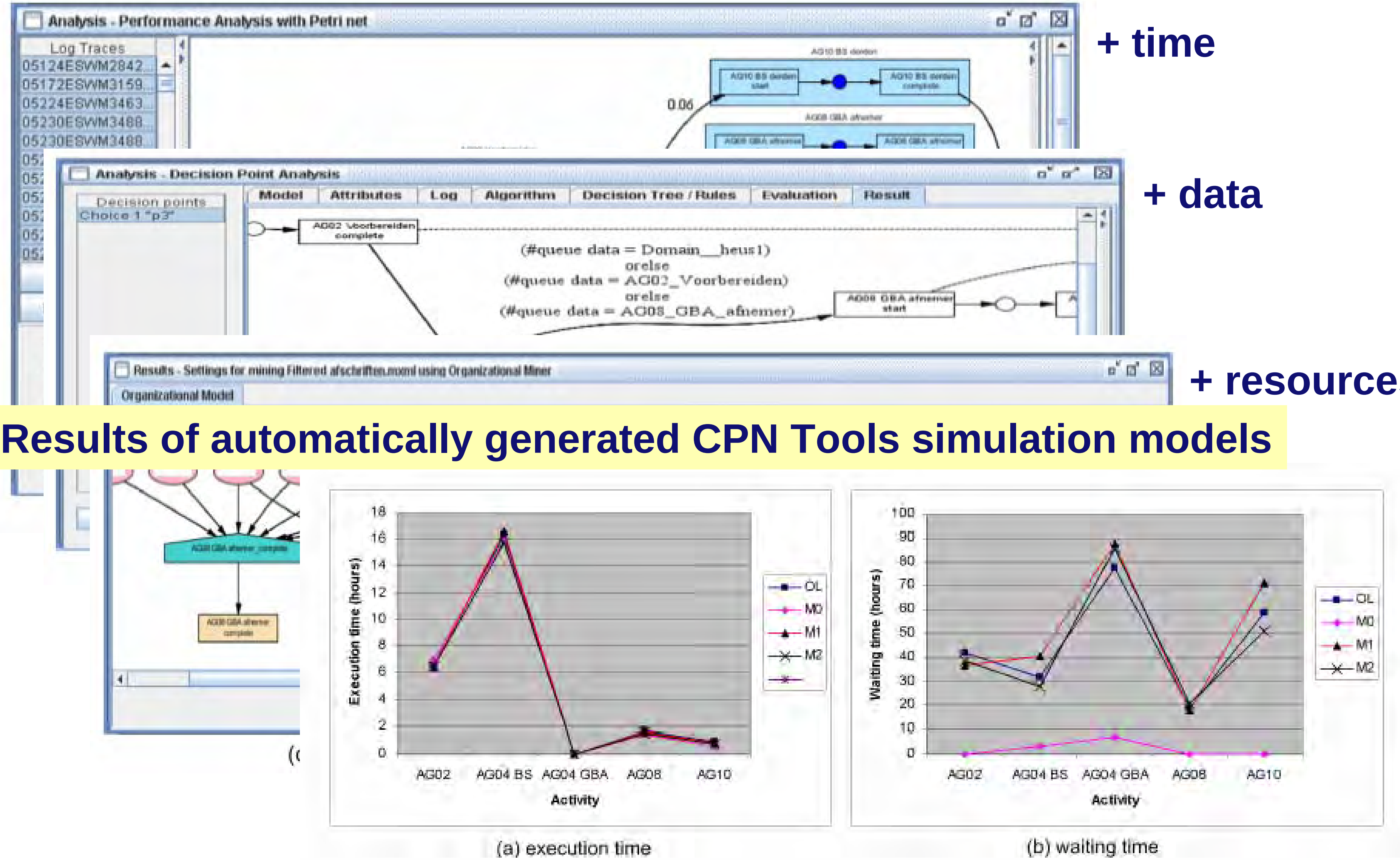
- Fitness is not sufficient: hence other metrics are needed such as behavioral and structural appropriateness, etc.
- These metrics cover aspects such as:
  - Punishing for "too much" behavior.
  - Punishing for "overly complex" models.

Extension

- Existing models can be enriched by logs analysis (e.g., indicating bottlenecks, etc.).
- Process mining results can be combined.
- Can be used to create comprehensive simulation models and export them to e.g. CPN Tools:



Example: Log from Dutch municipality



+ time

+ data

+ resources

Results of automatically generated CPN Tools simulation models

ProM Tool

ProM

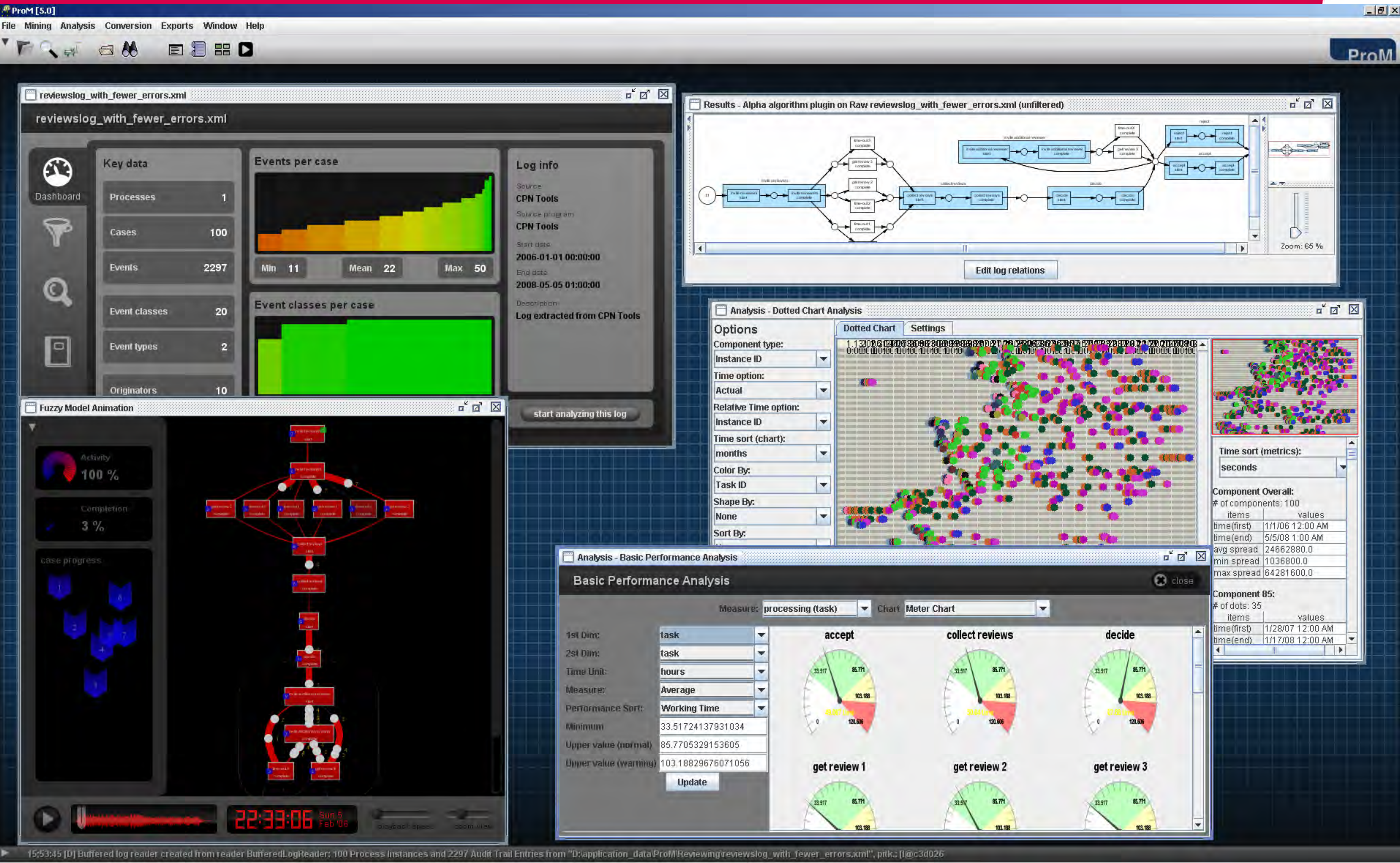
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

ProM

- Open source initiative started in 2003 after several early prototypes.
- Common Public License (CPL).
- Current version: 5.0.
- ProMimport: to extract MXML from all kinds of applications
- Plug-in architecture.
- About 250 plug-ins available:
  - mining plug-ins: 38 (all mining algorithms presented and many more)
  - analysis plug-ins: 71 (e.g., verification, SNA, LTL, conformance checking, etc.)
  - import: 21 (for loading EPCs, Petri nets, YAWL, BPMN, etc.)
  - export: 44 (for storing EPCs, Petri nets, YAWL, BPMN, BPEL, etc.)
  - conversion: 45 (e.g., translating EPCs or BPMN into Petri nets)
  - filter: 24 (e.g., removing infrequent activities)

Screenshot of ProM 5.0



Demo

Conclusion

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Conclusion

- The existence of event data enables a wide variety of process mining techniques ranging from process discovery to conformance checking.
- ProM supports this through +/- 250 plug-ins.
- A reality check for people that are involved in process modeling.
- Interesting challenges for both researchers and practitioners.
- Please join us! (www.processmining.org)

References

Introduction to Process Mining and ProM

- W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems 32(1), 713-732.*
- W.M.P. van der Aalst, et al. ProM 4.0: Comprehensive Support for Real Process Analysis. In J. Kleijn, A. Yakovlev, Petri Nets 2007, Lecture Notes in Computer Science, Vol. 4546, pp. 484-494. Berlin: Springer, 2007.
- W.M.P. van der Aalst and A.J.M.M. Weijters. Process Mining. *In M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede, editors, Process-Aware Information Systems: Bridging People and Software through Process Technology, pages 235-255. Wiley & Sons, 2005*
- W.M.P. van der Aalst , B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering, 47(2):237-267, 2003.*

Note that these references are far from complete and not intended to provide a comprehensive overview. See www.processmining.org for a good overview of (at least) all ProM-related publications.

References

Algorithmic Approaches

- W.M.P. van der Aalst , A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering (TKDE), volume 16(9), pages 1128-1142, 2004.
- L. Wen, W.M.P. van der Aalst, J. Wang, and J. Sun. Mining process models with non-free-choice constructs. Data Mining and Knowledge Discovery, 15(2):145-180, 2007.
- W.M.P. van der Aalst and B.F. van Dongen. Discovering Workflow Performance Models from Timed Logs. In Y. Han, S. Tai, and D. Wikarski, editors, International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002), volume 2480 of Lecture Notes in Computer Science, pages 45-63. Springer-Verlag, Berlin, 2002.
- B.F. van Dongen and W.M.P. van der Aalst. Multi-phase Process mining: Building Instance Graphs. Conceptual Modeling - ER 2004, LNCS 3288, pages 362-376, 2004.
- A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integral Computer-Aided Engineering, 10(2):151-162, 2003.
- C.W. Günther and W.M.P. van der Aalst. Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In International Conference on Business Process Management (BPM 2007), volume 4714 of Lecture Notes in Computer Science, pages 328-343. Springer-Verlag, Berlin, 2007.

References

Genetic Mining

- A.K. Alves de Medeiros, A.J.M.M. Weijters and W.M.P. van der Aalst. Genetic Process Mining: An Experimental Evaluation. Data Mining and Knowledge Discovery, volume 14, issue 2, pages 245-304, 2007.
- W.M.P. van der Aalst, A.K. Alves de Medeiros and A.J.M.M. Weijters. Genetic Process Mining. 26th International Conference on Applications and Theory of Petri Nets (ICATPN 2005), G. Ciardo and P. Darondeau, LNCS 3536, pages 48-69, 2005.

References

Mining Based on Regions

- J.M.E.M. van der Werf, B.F. van Dongen, C.A.J. Hurkens, and A. Serebrenik. Process Discovery using Integer Linear Programming. In K. van Hee and R. Valk, editors, *Proceedings of the 29th International Conference on Applications and Theory of Petri Nets (Petri Nets 2008)*, volume 5062 of *Lecture Notes in Computer Science*, pages 368-387. Springer-Verlag, Berlin, 2008.
- W.M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. Günther. Process Mining: A Two-Step Approach using Transition Systems and Regions. BPM Center Report BPM-06-30, BPMcenter.org, 2006.
- R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser. Process Mining Based on Regions of Languages. In G. Alonso, P. Dadam, and M. Rosemann, editors, *International Conference on Business Process Management (BPM 2007)*, volume 4714 of *Lecture Notes in Computer Science*, pages 375-383. Springer-Verlag, Berlin, 2007.

References

Mining Other Perspectives

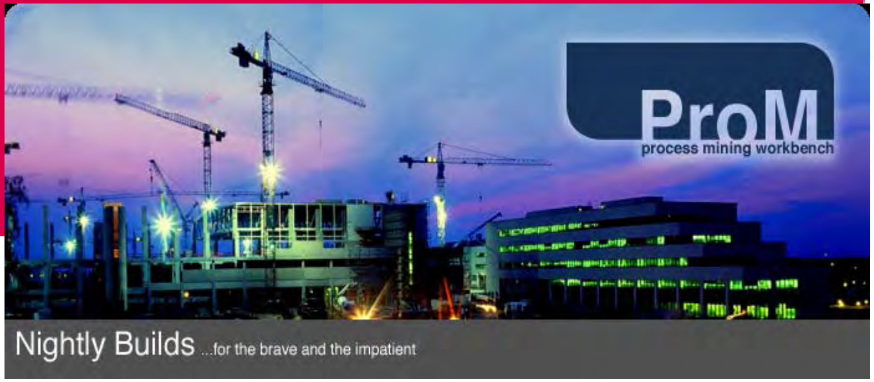
- A. Rozinat and W.M.P. van der Aalst. Decision Mining in ProM. In S. Dustdar, J.L. Faideiro, and A. Sheth, editors, *International Conference on Business Proces Management (BPM 2006)*, volume 4102 of *Lecture Notes in Computer Science*, pages 420-425. Springer-Verlag, Berlin, 2006.
- Wil M. P. van der Aalst, Hajo A. Reijers. Minseok Song: Discovering Social Networks from Event Logs. Computer Supported Cooperative Work 14(6): 549-593, 2005.
- Anne Rozinat, R. S. Mans, Minseok Song, Wil M. P. van der Aalst: Discovering colored Petri nets from event logs. STTT 10(1): 57-74, 2008.

References

Conformance Checking and Extension

- A. Rozinat and W.M.P. van der Aalst. Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems, Volume 33, Issue 1, Pages 64-95, 2008.
- W.M.P. van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, and H.M.W. Verbeek. Conformance Checking of Service Behavior. ACM Transactions on Internet Technology (TOIT), Volume 8, Issue 3, 2008.
- A. Rozinat, M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede and C.J. Fidge. Workflow Simulation for Operational Decision Support Using Design, Historic and State Information. In M. Dumas, M. Reichert, and M.-C. Shan (Eds.): BPM 2008, LNCS 5240, pp. 196–211, Springer-Verlag Berlin.
- Anne Rozinat, Ana Karla Alves de Medeiros, Christian W. Günther, A. J. M. M. Weijters, Wil M. P. van der Aalst: The Need for a Process Mining Evaluation Framework in Research and Practice. Business Process Management Workshops 2007: 84-89.

Relevant WWW sites



- http://www.processmining.org
- **http:// promimport.sourceforge.net**
- **http://prom.sourceforge.net**
- **http://www.workflowpatterns.com**
- **http://www.workflowcourse.com**
- **http://www.vdaalst.com**