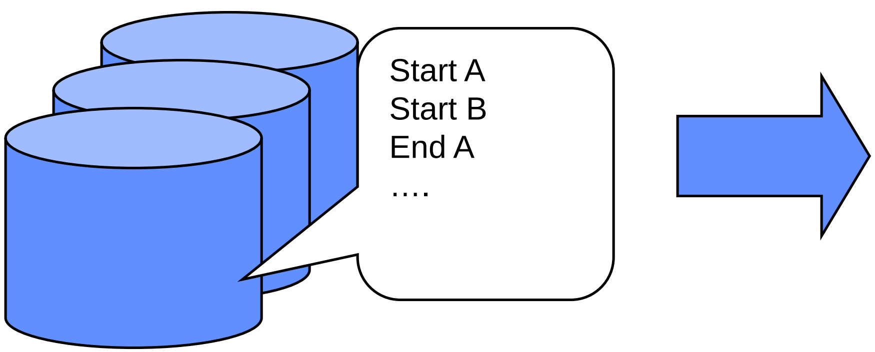
Process Discovery and Conformance Checking

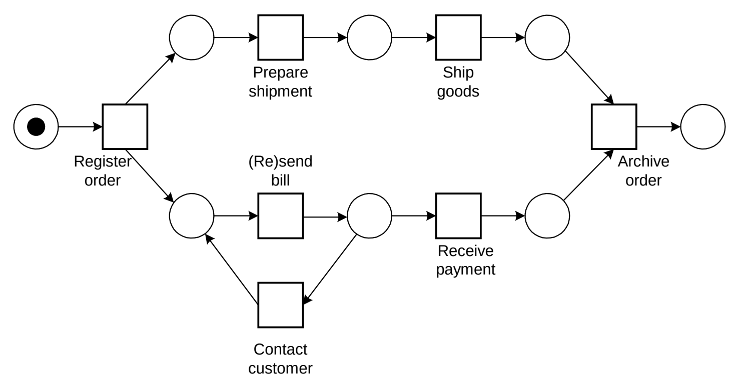
Univ. Prof. Dr. Stefanie Rinderle-Ma
Workflow Systems and Technology Group
University of Vienna
stefanie.rinderle-ma@univie.ac.at

Contents

1 Motivation

- Particularly exploration („finding“) process models is often a cumbersome and errorneous task
- Are there alternatives?
- Observation: Processes are often implicitly executed (maybe distributed over different systems)
- Prerequisite: Log data of processes available
- Process / Workflow mining offers techniques to automatically derive process / workflow models from such log data





1 Motivation

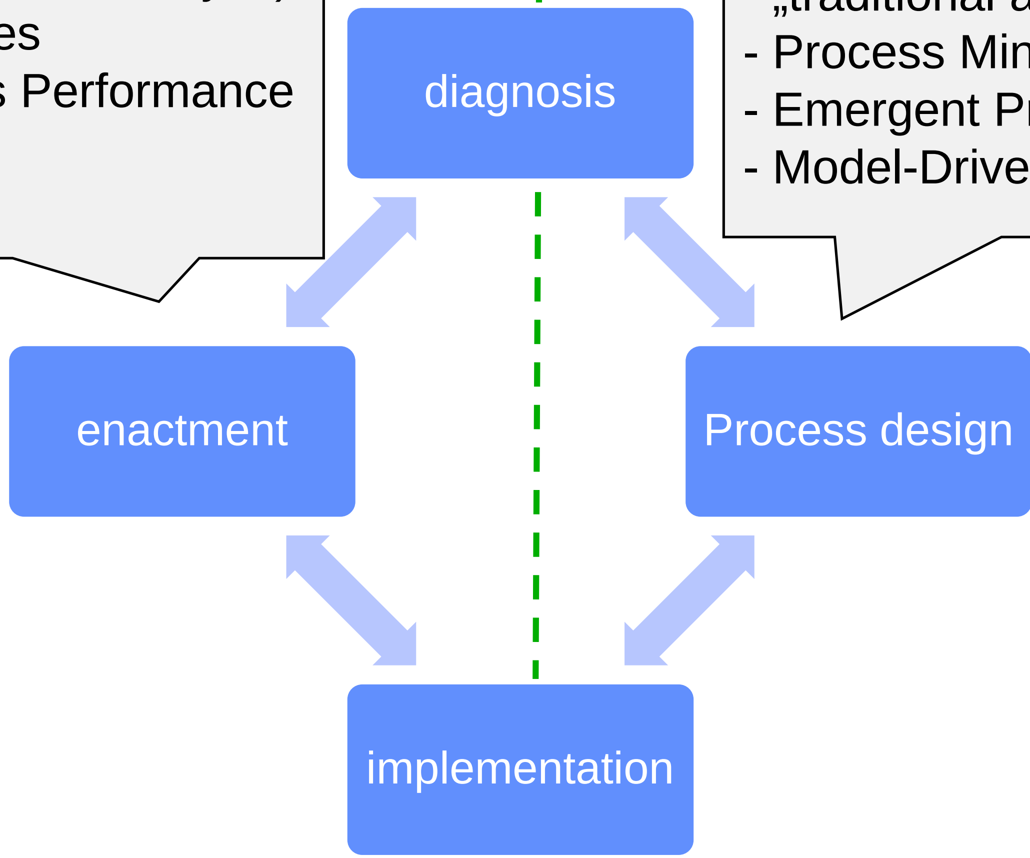
Business process life cycle

Runtime

- Process Mining (Delta Analyse)
- Adaptive processes
- Business Process Performance Management

Design time

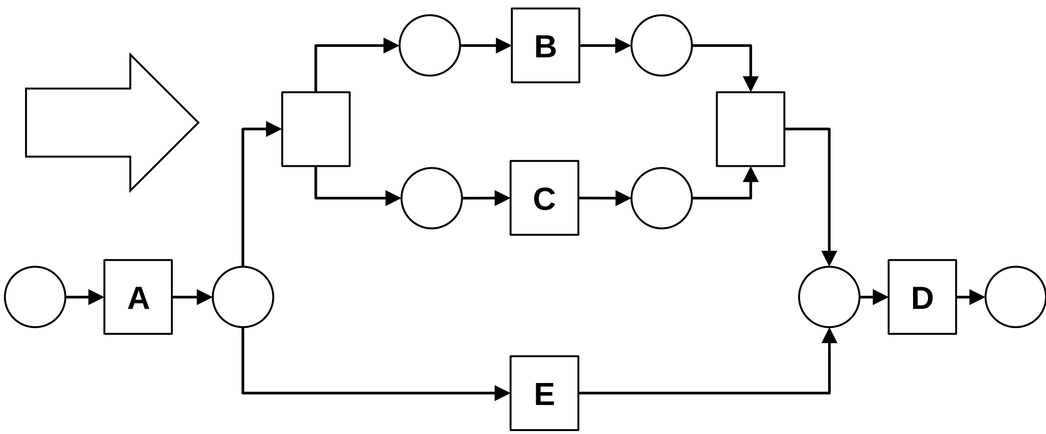
- „traditional approaches“
- Process Mining
- Emergent Processes
- Model-Driven Development



1 Motivation

Workflow Instance	Workflow Task
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

Execution log (e.g., Staffware) [ADHM03]



Teaching Objectives

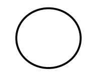
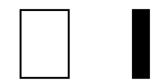
After discussing this chapter it should be clear

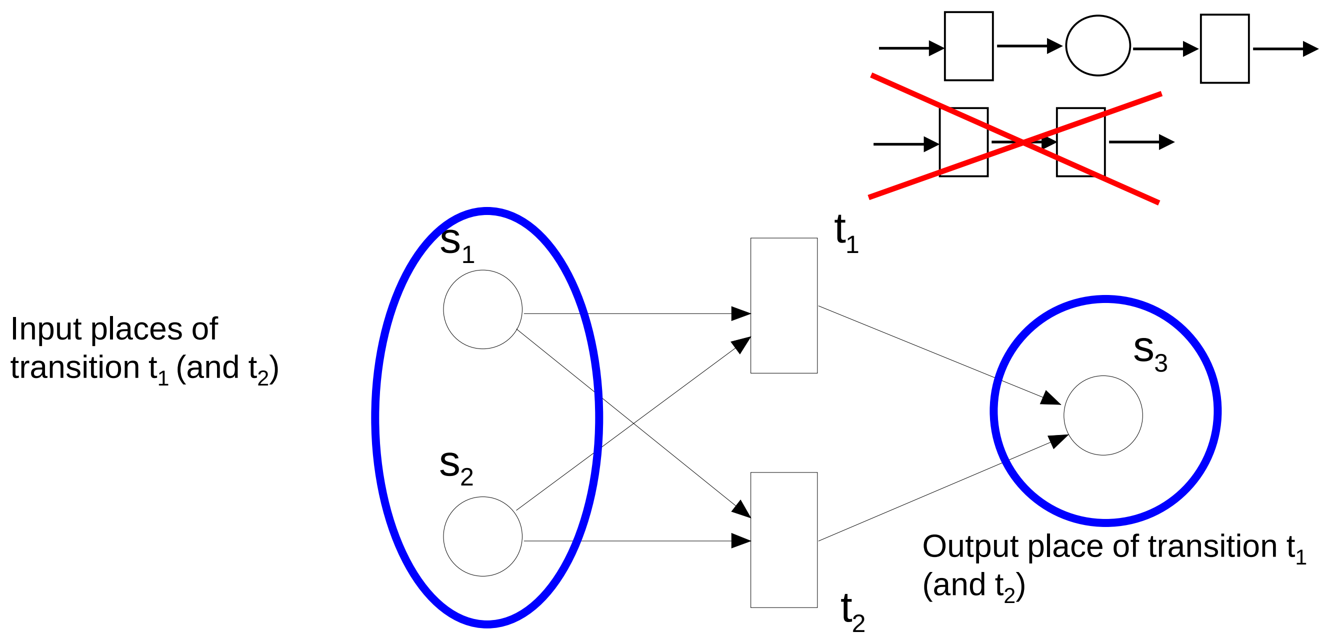
- what process discovery means
- what is the basic idea behind process discovery algorithms
- what are typical problems arising with process discovery algorithms
- how these problems can be tackled
- which algorithms exist and how they differ
- discovery is one of the process mining tasks (i.e., there are more tasks)

Contents

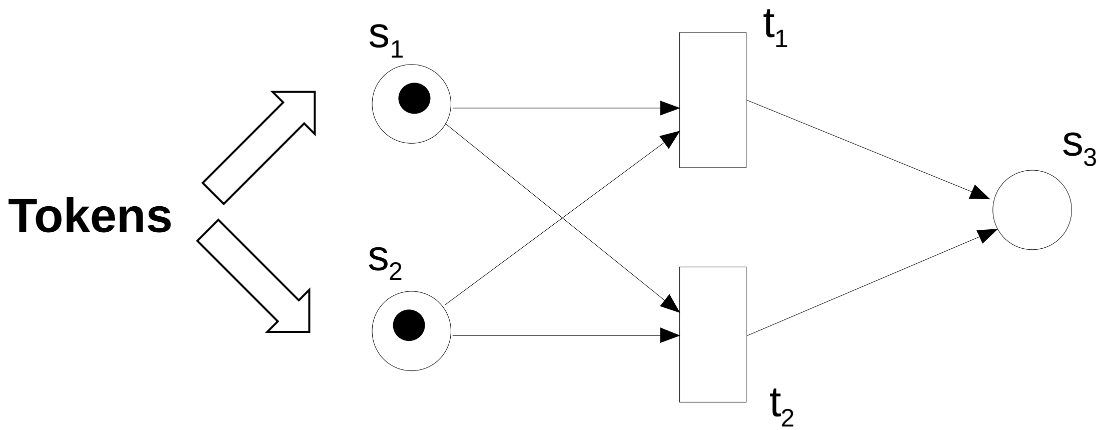
2 Petri Nets in a Nutshell

A Petri Net is a bipartite graph and contains the following (static) elements:

- Places → Preconditions / States 
- Transitions → Actions / Activities 
- Edges that connect places with transitions and transitions with places
- Example:



- Dynamic behavior of a Petri Net is represented by tokens and markings respectively.
- Example:

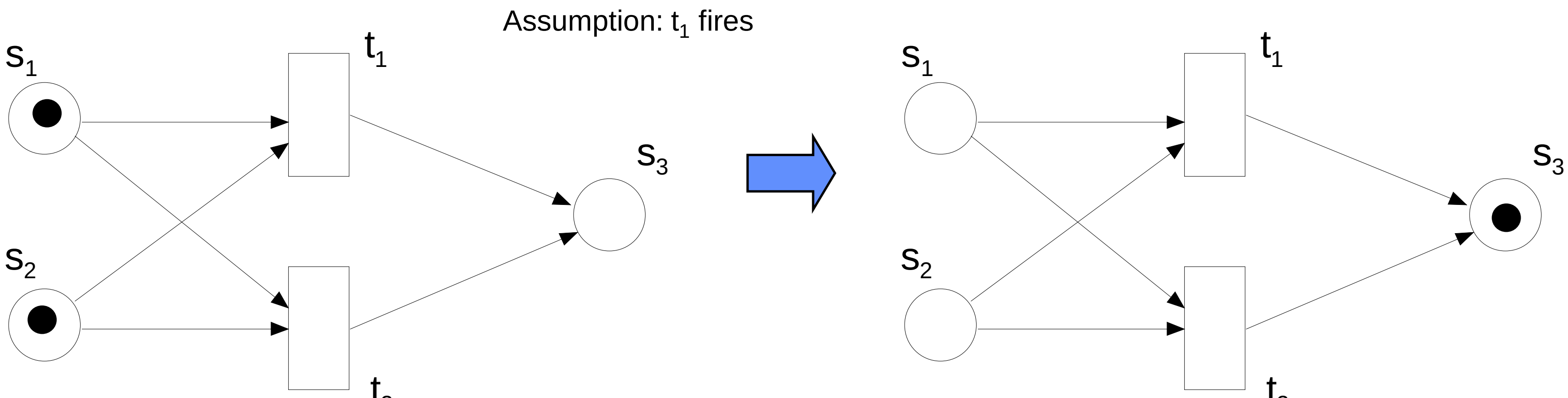


Interpretation:

Transitions t_1 and t_2 are activated → only one of them can fire

Dynamic behavior (ctd):

- Firing rules determine the execution of the Petri Net
- Example:



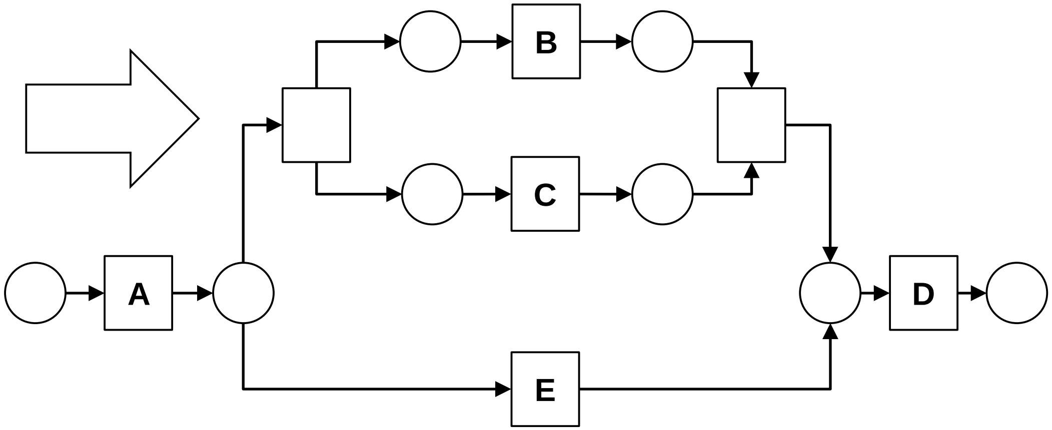
- Result: all tokens from the input places of t1 are removed and all output places of t2 are marked with tokens

Contents

3 The alpha-Algorithm

Workflow Instance	Workflow Task
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

Execution log (e.g., Staffware) [ADHM03]



In general: Basic workflow mining algorithms analyze frequency and order of events within the logs of the different workflow instances.

Workflow Instance	Workflow Task
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

1st Step:

Determine traces σ within log W:

Case 1: <A, B, C, D>

Case 2: <A, C, B, D>

Case 3: <A, B, C, D>

Case 4: <A, C, B, D>

Case 5: <A, E, D>

α-algorithm [ADHM03 AWM04, MDAW04]

Traces σ in log W:

Case 1: <A, B, C, D>

Case 2: <A, C, B, D>

Case 3: <A, B, C, D>

Case 4: <A, C, B, D>

Case 5: <A, E, D>

2nd step:

Analyzing order relations between tasks per each trace:

→ $a >_W b \Leftrightarrow \exists$ trace $\sigma = t_1 t_2 t_3 \ldots t_{n-1}$, such that: $\sigma \in W$ and $t_i = a$ and $t_{i+1} = b$

Result of Analysis in 2nd step:

Case 1: $A >_W B$, $B >_W C$, $C >_W D$

Case 2: $A >_W C$, $C >_W B$, $B >_W D$

Case 3: $A >_W B$, $B >_W C$, $C >_W D$

Case 4: $A >_W C$, $C >_W B$, $B >_W D$

Case 5: $A >_W E$, $E >_W D$

α-algorithm [ADHM03 AWM04, MDAW04]

Case 1: A $>_W$ B, B $>_W$ C, C $>_W$ D

Case 2: A $>_W$ C, C $>_W$ B, B $>_W$ D

Case 3: A $>_W$ B, B $>_W$ C, C $>_W$ D

Case 4: A $>_W$ C, C $>_W$ B, B $>_W$ D

Case 5: A $>_W$ E, E $>_W$ D

3rd step:

Analysis of order relations over entire workflow log W:

- **a $\rightarrow_W$ b $\Leftrightarrow$ a $>_W$ b and $\neg$ (b $>_W$ a)**
- **a$\#_W$b $\Leftrightarrow$ $\neg$ (a $>_W$ b or b $>_W$ a)**
- **a$||_W$b $\Leftrightarrow$ a $>_W$ b and b $>_W$ a**

Result of analysis step 3:

A $\rightarrow_W$ B, A $\rightarrow_W$ C, A $\rightarrow_W$ E

B $||_W$ C

B $\rightarrow_W$ D, C $\rightarrow_W$ D, E $\rightarrow_W$ D

E $\#_W$ B, E$\#_W$ C

α-algorithm [ADHM03 AWM04, MDAW04]

A $\rightarrow_W$ B, A $\rightarrow_W$ C, A $\rightarrow_W$ E

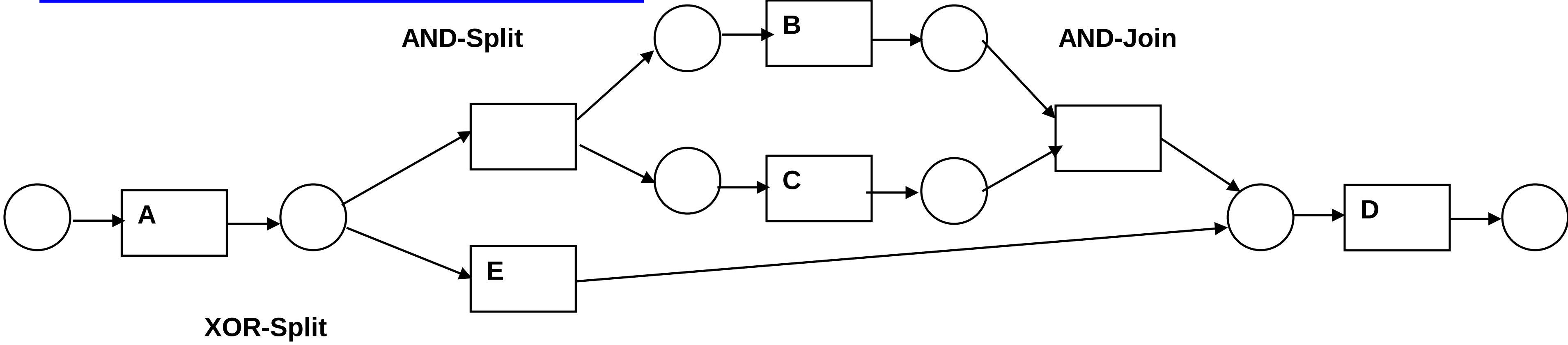
B $||_W$ C

B $\rightarrow_W$ D, C $\rightarrow_W$ D, E $\rightarrow_W$ D

E $\#_W$ B, E$\#_W$ C

4th step:

Deriving associated Petri net



EXERCISE: For the following log, apply the α-algorithm and derive the corresponding Petri Net:

```
- Case1: <A, C, D, E, F, G>
- Case2: <A, B, D, E, F, G>
- Case3: <A, C, D, F, E, G>
- Case4: <A, B, D, F, E, G>
- Case5: <A, B, D, E, F, G>
```

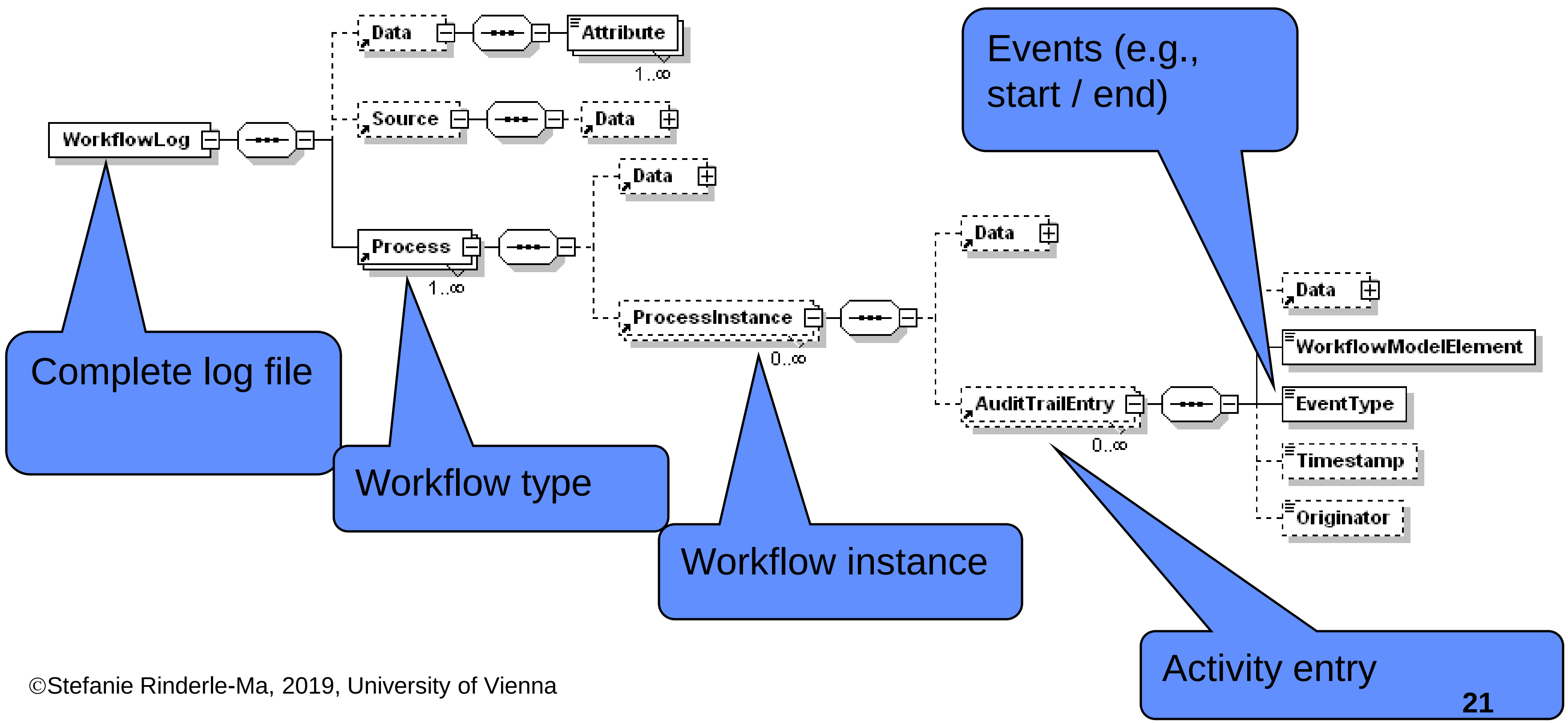
Result:

ProM system:

- Alternative approaches for process exploration are highly relevant since „traditional“ approaches (e.g., interviews, questionnaires) are often costly
- Process mining offers means to automatically derive process models from log data
- However, existence of log data and its quality are preconditions for applicability of process mining techniques

- Currently: ProM is the most comprehensive mining toolkit
- open source: http://www.processmining.org/prom/start
- Developed at Technical University of Eindhoven (The Netherlands)
- ProM comprises:
  - ProM framework: process analysis, filtering, and mining
  - ProM*import* framework: imports and converts log data from different systems into ProM specific format
  - ProM CPN library: simulation environment based on colored Petri Nets
- Two log formats: MXML and XES (eXtensible Event Stream) → see also chapter on data provisioning

- In the following examples: execution logs are generated in ADEPT and transformed into MXML (ProMimport contains converter from ADEPT to ProM)
- MXML: XML-based data format for ProM [GuAa06]:



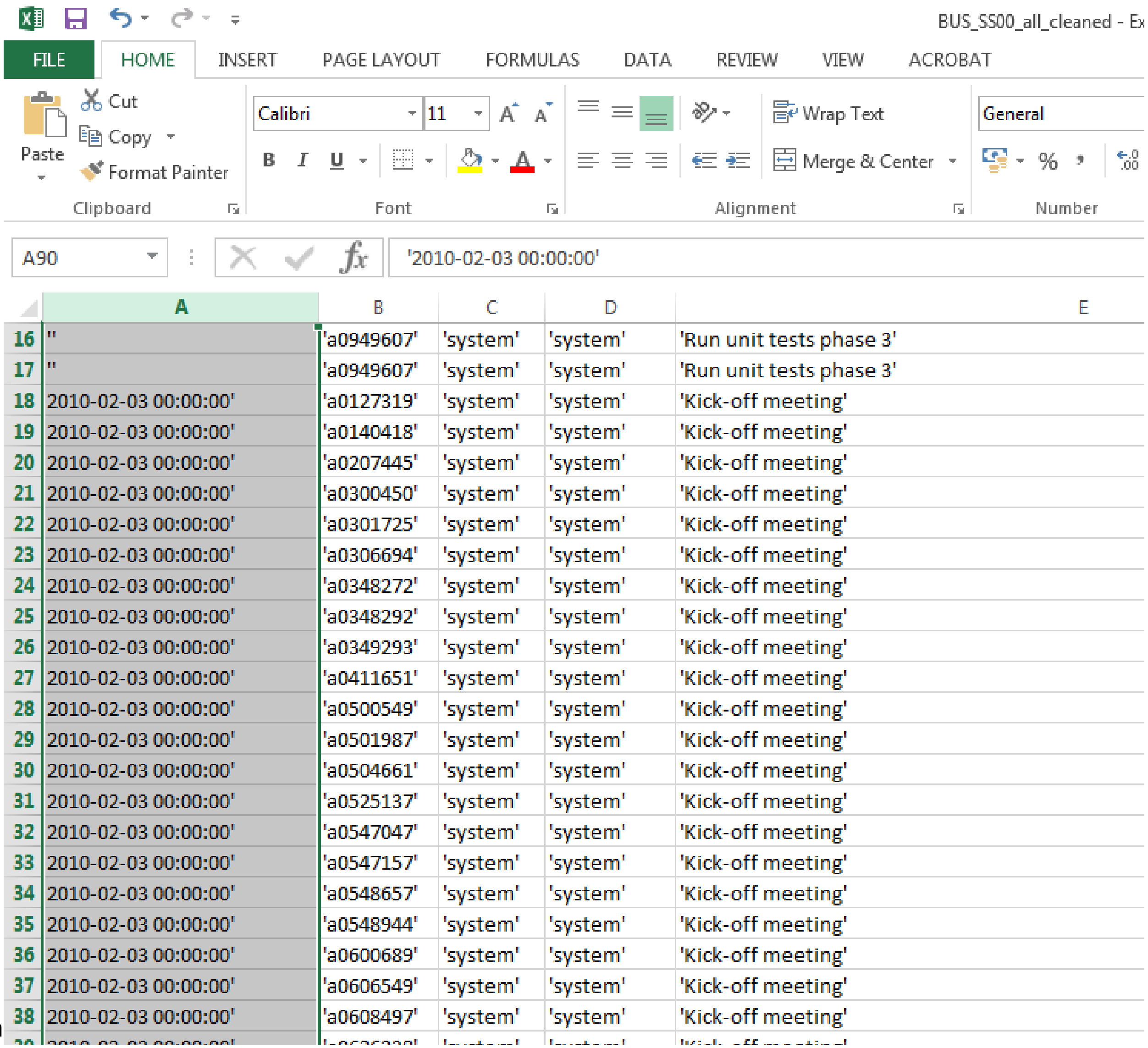
MXML-file for example workflow instance:

```
<ProcessInstance id="EXECLOG_OP-Vorbereitung_39">
        <AuditTrailEntry>
                <WorkflowModelElement>0 Start</WorkflowModelElement>
                <EventType>start</EventType>
                <Timestamp>2006-04-19T14:58:54Z</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>0 Start</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2006-04-19T14:58:54Z</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>2 Patient aufnehmen</WorkflowModelElement>
                <EventType>start</EventType>
                <Timestamp>2006-04-19T14:58:54Z</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>2 Patient aufnehmen</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2006-04-19T14:58:54Z</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>3 Blutentnahme</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2006-04-19T14:58:54Z</Timestamp>
        </AuditTrailEntry>
...
```

3 The alpha-Algorithm

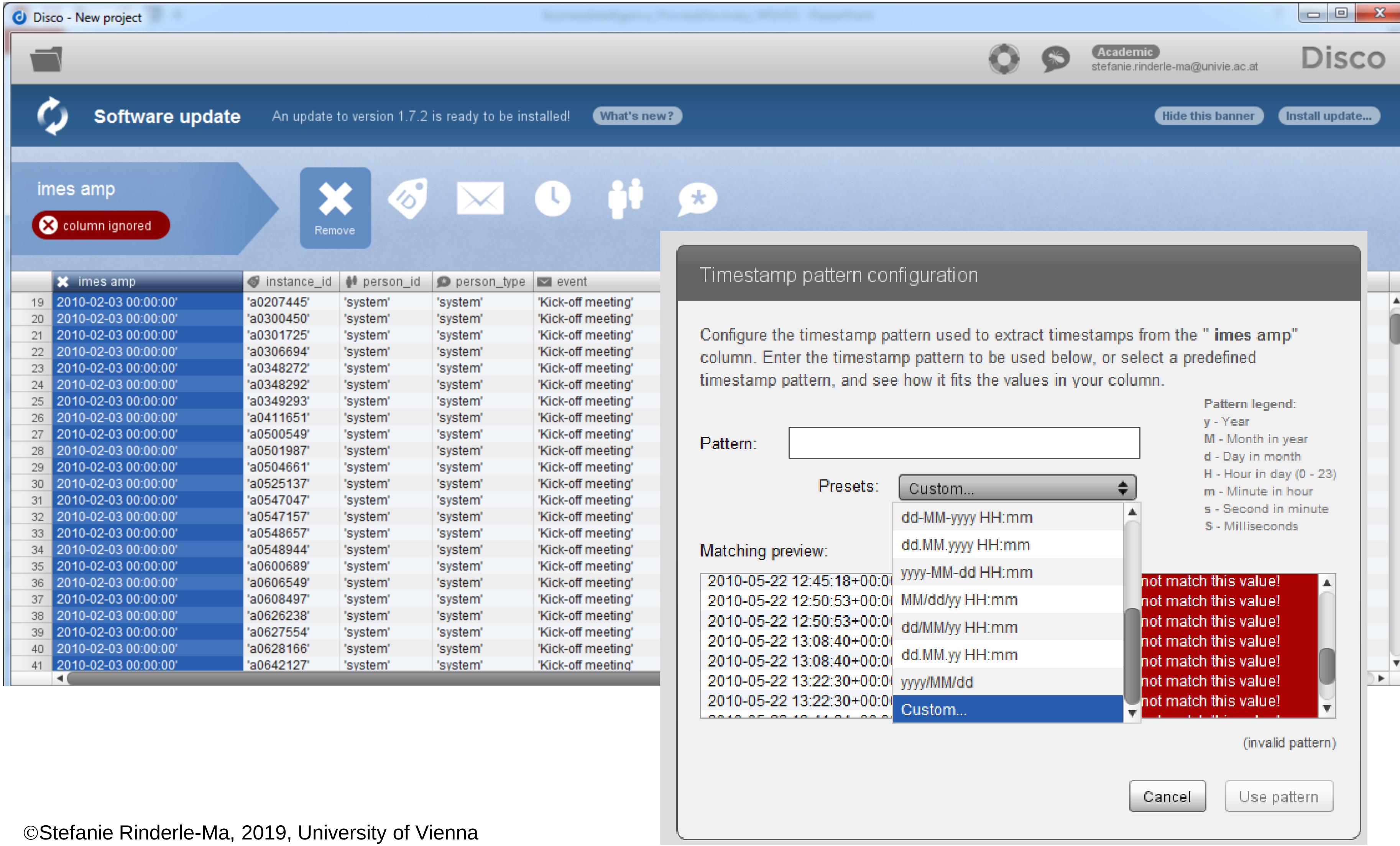
- How to obtain logs?
- Challenges:
  - distributed sources
  - different format
- → information integration problem
- Minimum requirements
  - case ID
  - events (START / END); order relevant (time stamps or ordered log)
- Additionally:
  - Performers
  - General data
- Useful tools:
  - ProM Import
  - Commercial: Disco (direct import of csv data)

3 The alpha-Algorithm

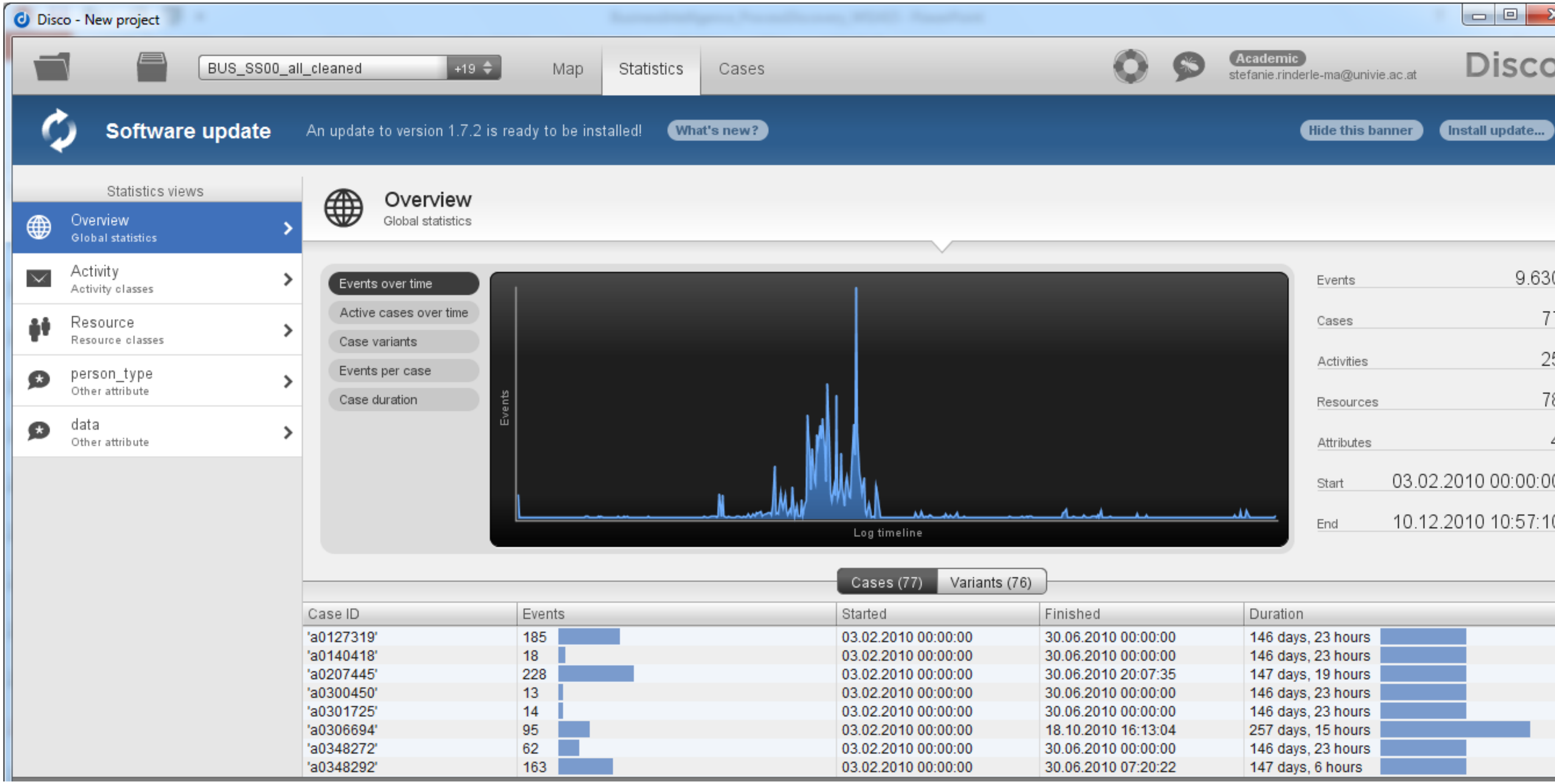


	A	B	C	D	E
16	''	'a0949607'	'system'	'system'	'Run unit tests phase 3'
17	''	'a0949607'	'system'	'system'	'Run unit tests phase 3'
18	2010-02-03 00:00:00'	'a0127319'	'system'	'system'	'Kick-off meeting'
19	2010-02-03 00:00:00'	'a0140418'	'system'	'system'	'Kick-off meeting'
20	2010-02-03 00:00:00'	'a0207445'	'system'	'system'	'Kick-off meeting'
21	2010-02-03 00:00:00'	'a0300450'	'system'	'system'	'Kick-off meeting'
22	2010-02-03 00:00:00'	'a0301725'	'system'	'system'	'Kick-off meeting'
23	2010-02-03 00:00:00'	'a0306694'	'system'	'system'	'Kick-off meeting'
24	2010-02-03 00:00:00'	'a0348272'	'system'	'system'	'Kick-off meeting'
25	2010-02-03 00:00:00'	'a0348292'	'system'	'system'	'Kick-off meeting'
26	2010-02-03 00:00:00'	'a0349293'	'system'	'system'	'Kick-off meeting'
27	2010-02-03 00:00:00'	'a0411651'	'system'	'system'	'Kick-off meeting'
28	2010-02-03 00:00:00'	'a0500549'	'system'	'system'	'Kick-off meeting'
29	2010-02-03 00:00:00'	'a0501987'	'system'	'system'	'Kick-off meeting'
30	2010-02-03 00:00:00'	'a0504661'	'system'	'system'	'Kick-off meeting'
31	2010-02-03 00:00:00'	'a0525137'	'system'	'system'	'Kick-off meeting'
32	2010-02-03 00:00:00'	'a0547047'	'system'	'system'	'Kick-off meeting'
33	2010-02-03 00:00:00'	'a0547157'	'system'	'system'	'Kick-off meeting'
34	2010-02-03 00:00:00'	'a0548657'	'system'	'system'	'Kick-off meeting'
35	2010-02-03 00:00:00'	'a0548944'	'system'	'system'	'Kick-off meeting'
36	2010-02-03 00:00:00'	'a0600689'	'system'	'system'	'Kick-off meeting'
37	2010-02-03 00:00:00'	'a0606549'	'system'	'system'	'Kick-off meeting'
38	2010-02-03 00:00:00'	'a0608497'	'system'	'system'	'Kick-off meeting'

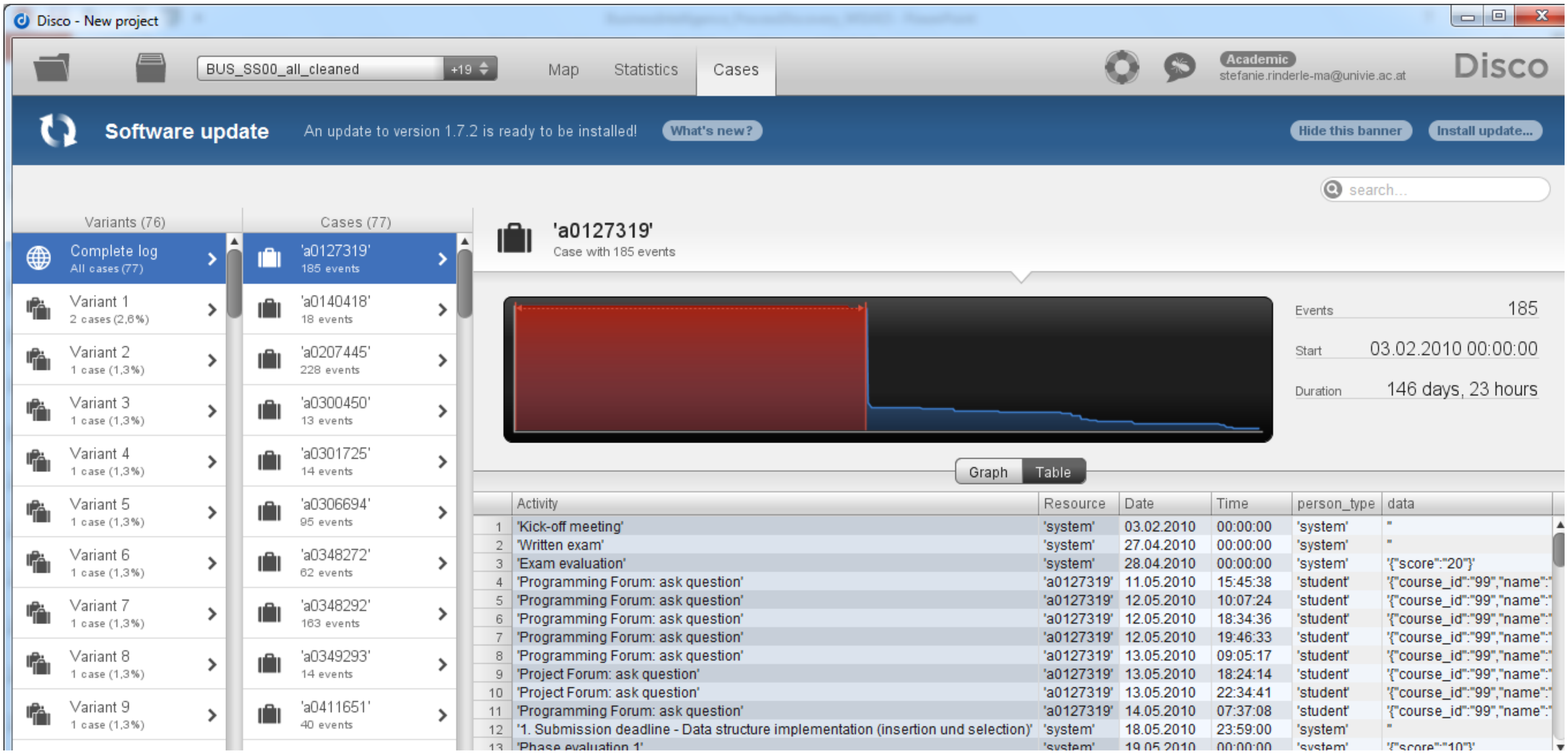
3 The alpha-Algorithm



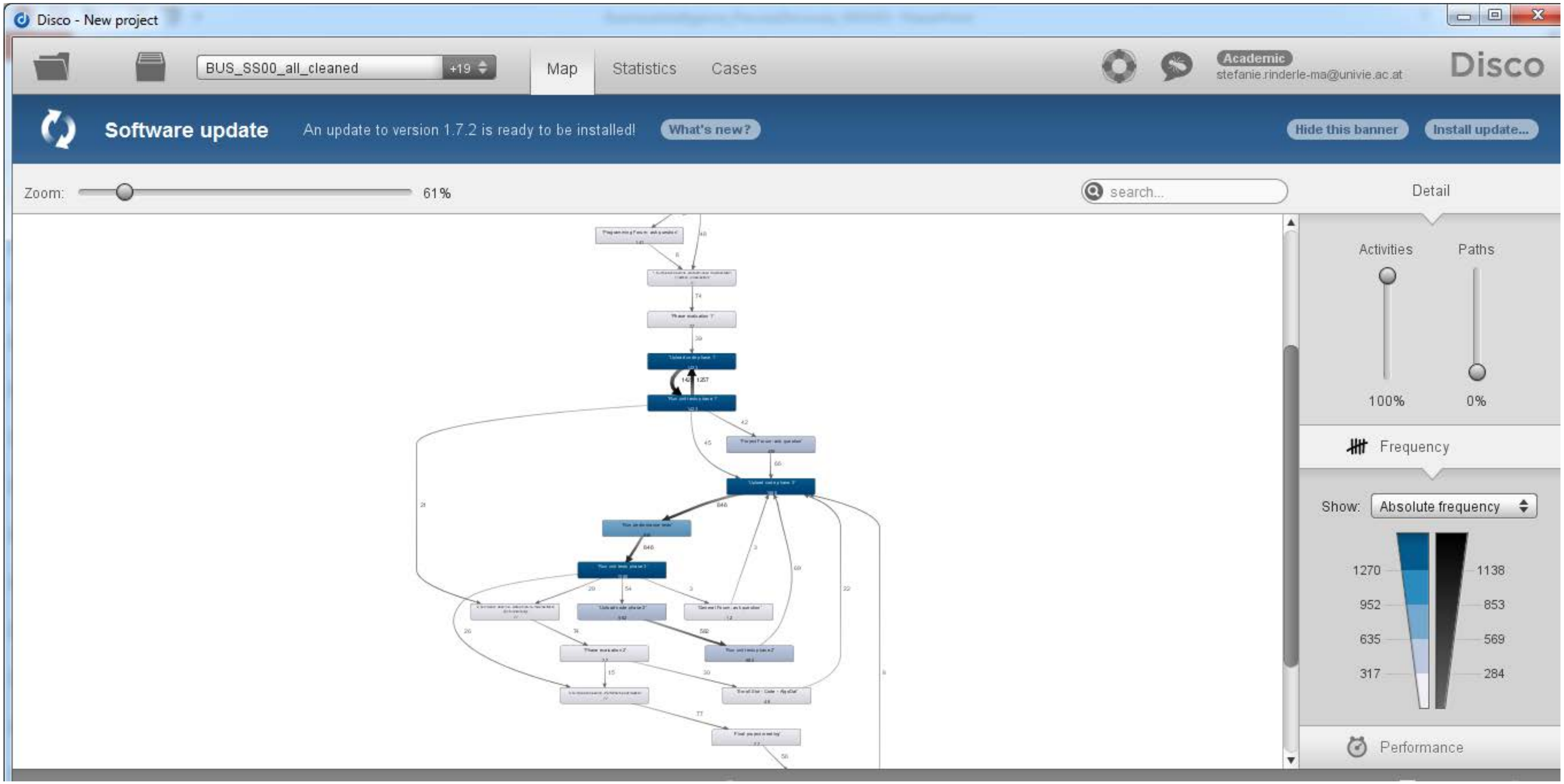
3 The alpha-Algorithm



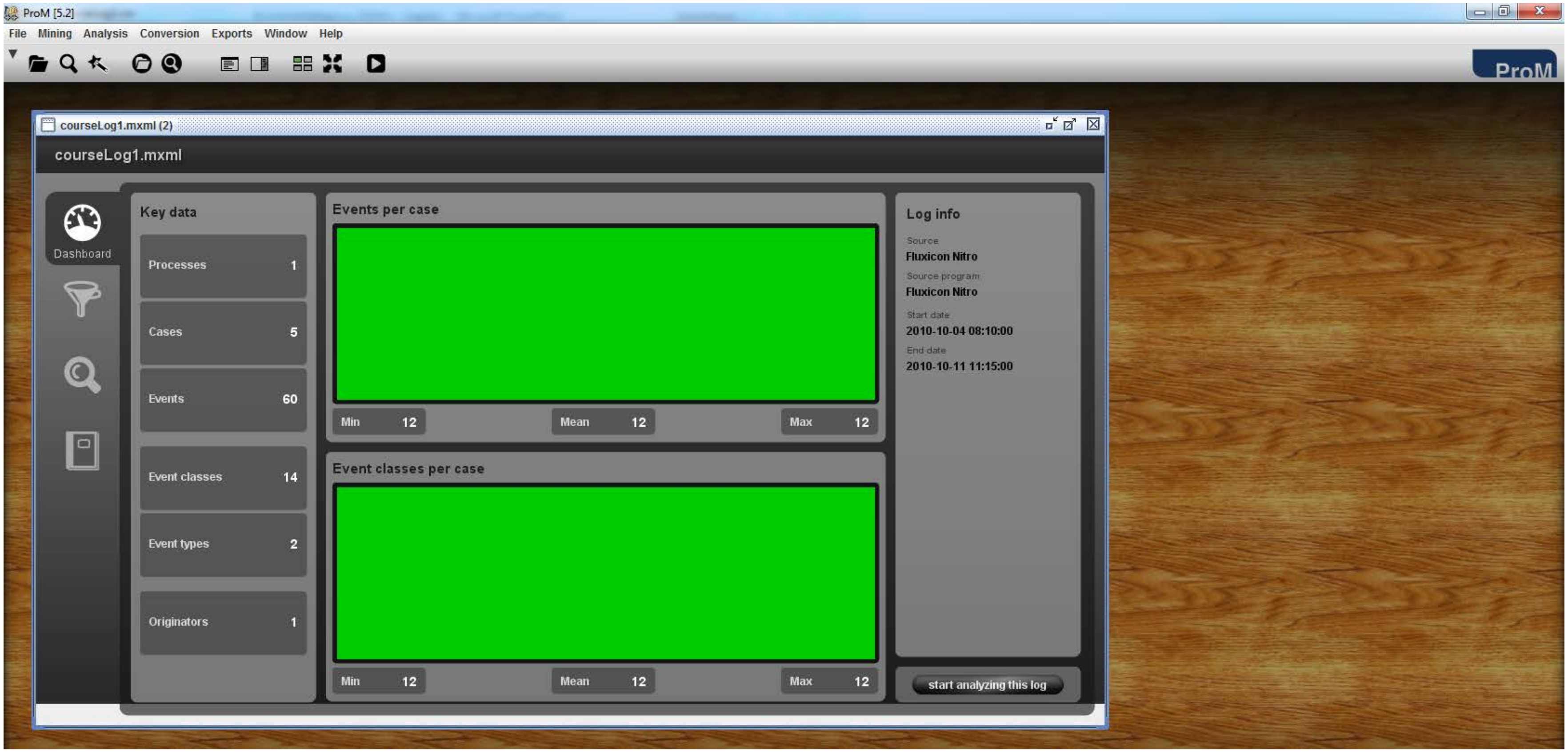
3 The alpha-Algorithm



3 The alpha-Algorithm

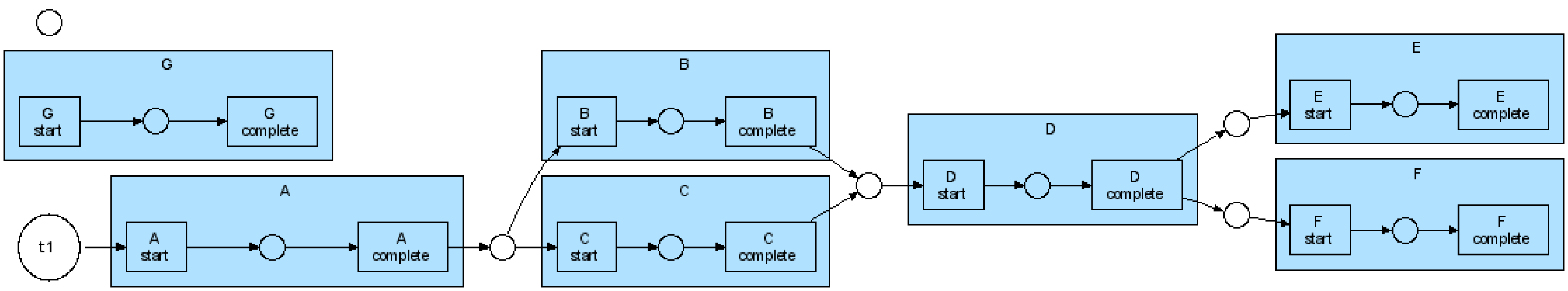


3 The alpha-Algorithm



After import to ProM → some first statistics (of course not that interesting for this small example)

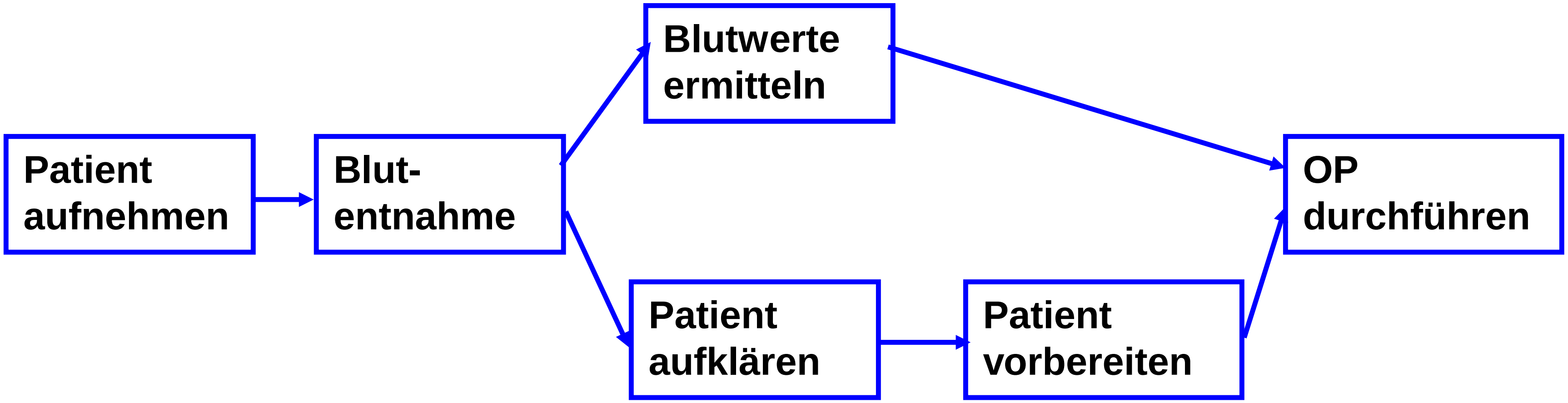
Our exercise example after applying α-algorithm

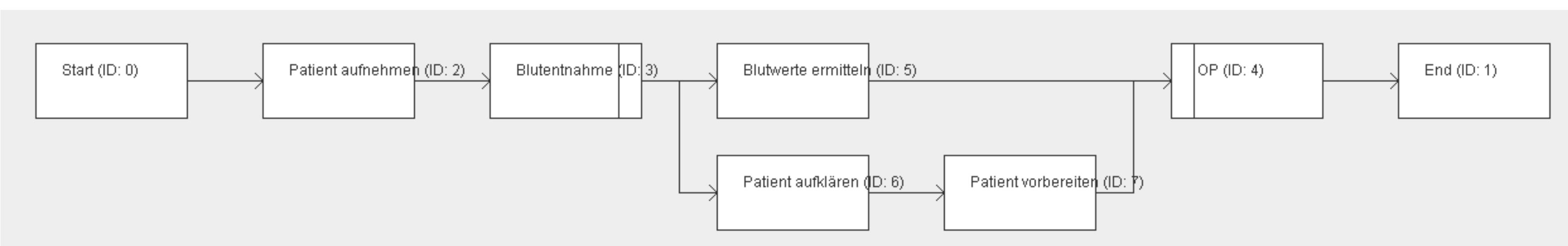


Why does it look different form our hands-on result?

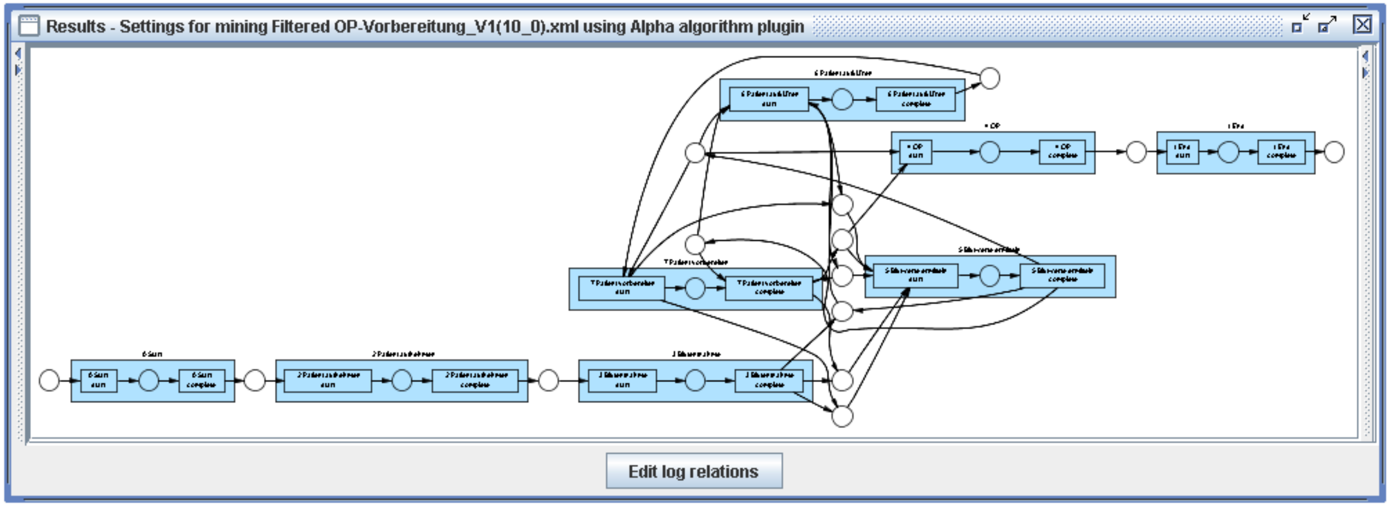
3 The alpha-Algorithm

Example process (designed and executed within ADEPT Demonstrator)

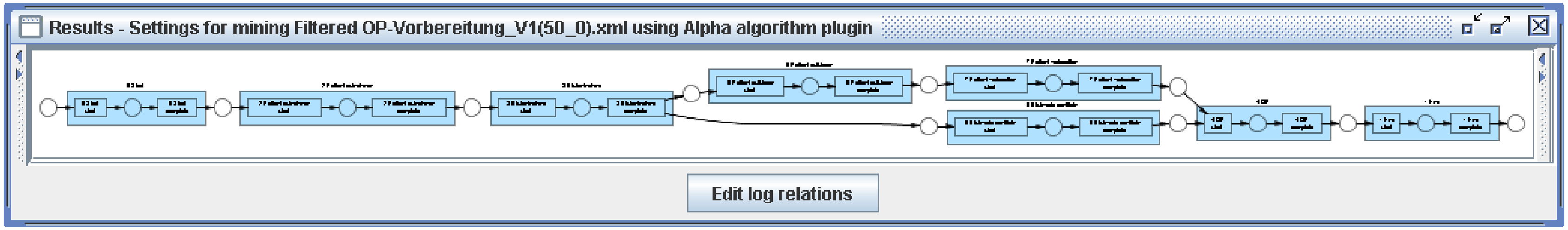




Example 1: α-algorithm for 10 instances

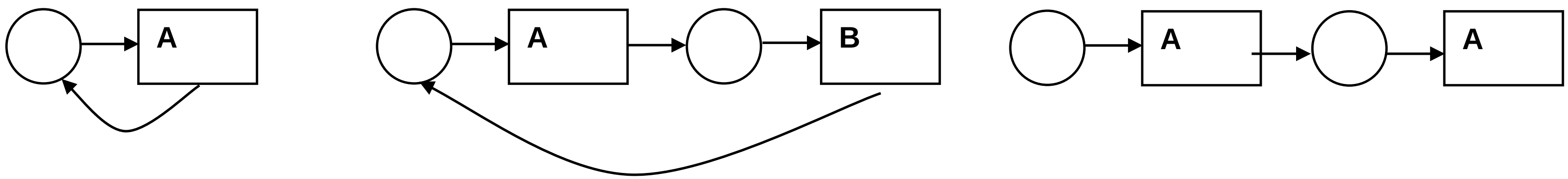


Example 2: α-algorithm for 50 instances

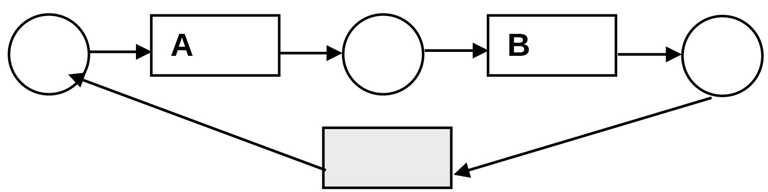


Structural problems (e.g., α-algorithm [ADHM03])

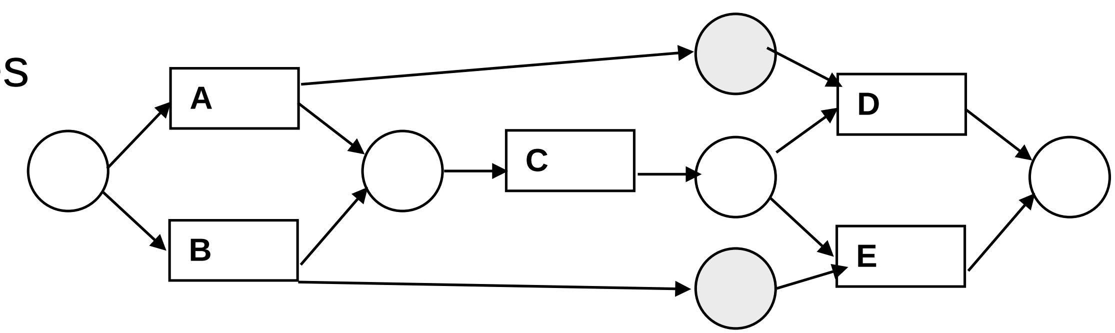
- Short loops and multiple occurrences of activities



- Hidden activities (e.g. for structuring purposes)

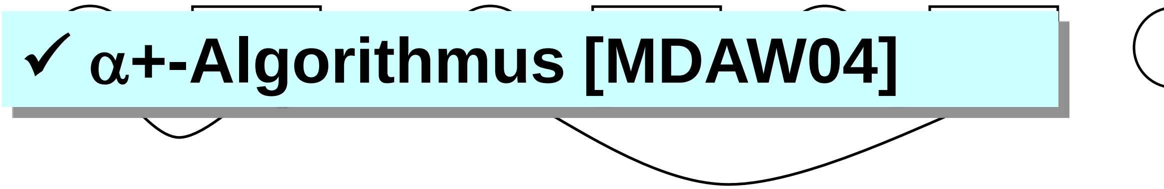


- Implicit dependencies



Structural problems (e.g., α-algorithm [ADHM03])

- Short loops and multiple occurrences of activities



✓ α+-Algorithmus [MDAW04]

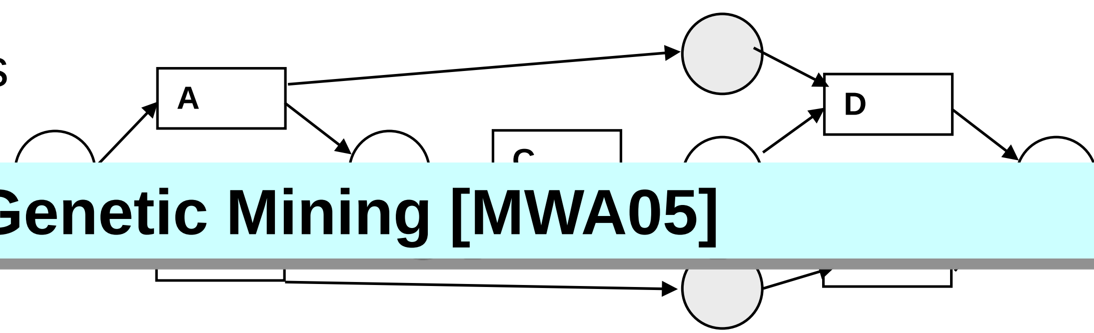
Still open (WHY?)

- Hidden activities (e.g. for structuring purposes)

✓ Heuristics Miner

✓ Genetic Mining [MWA05]

- Implicit dependencies



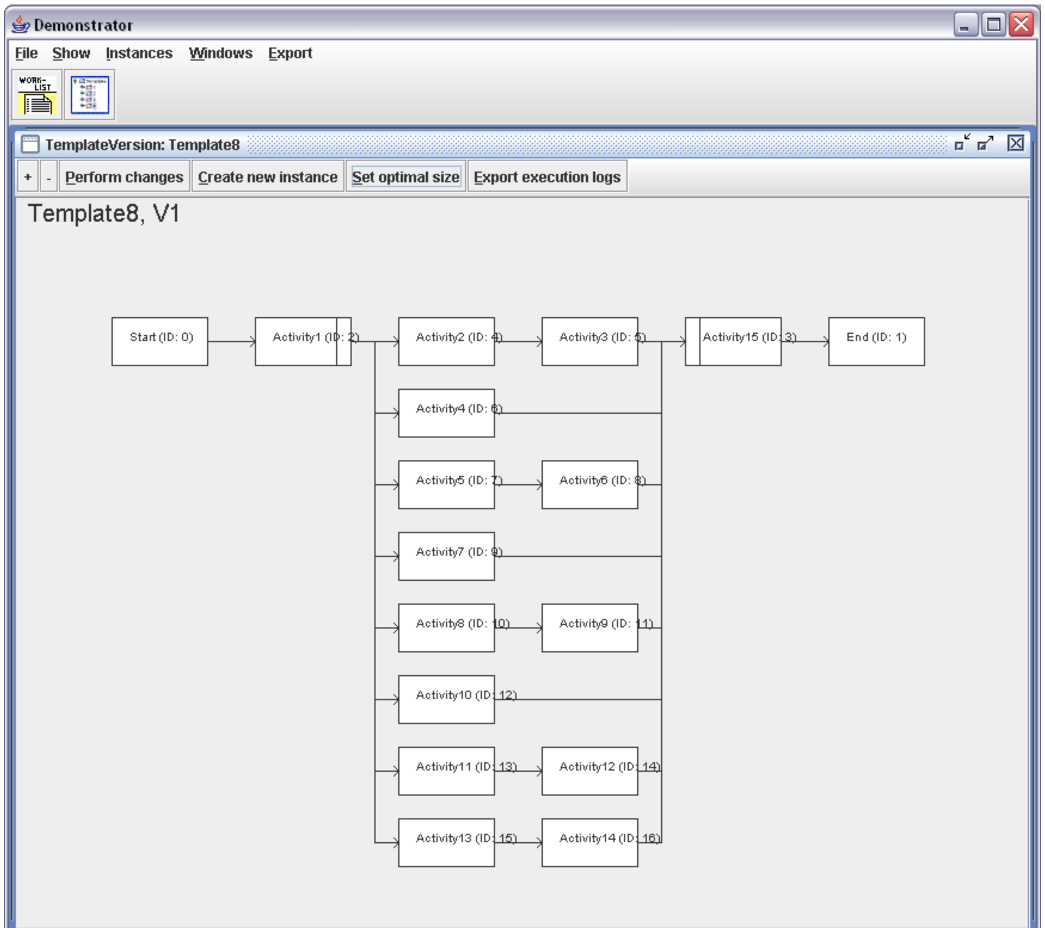
✓ Genetic Mining [MWA05]

- Further problems:
  - Noise (missing or erroneous log data)
  - small/ big number of log data
  - Parallel branchings with high number of branches
- Algorithms:
  - Multi-Phase-Mining [DoAa05]:
    - For each case a separate graph is generated (Petri Net, EPC)
    - The case graphs are aggregated afterwards
    - Robust when dealing with noise
  - Genetic algorithm [MWA05]:
    - Robust when dealing with noisy log data
    - Not very efficient

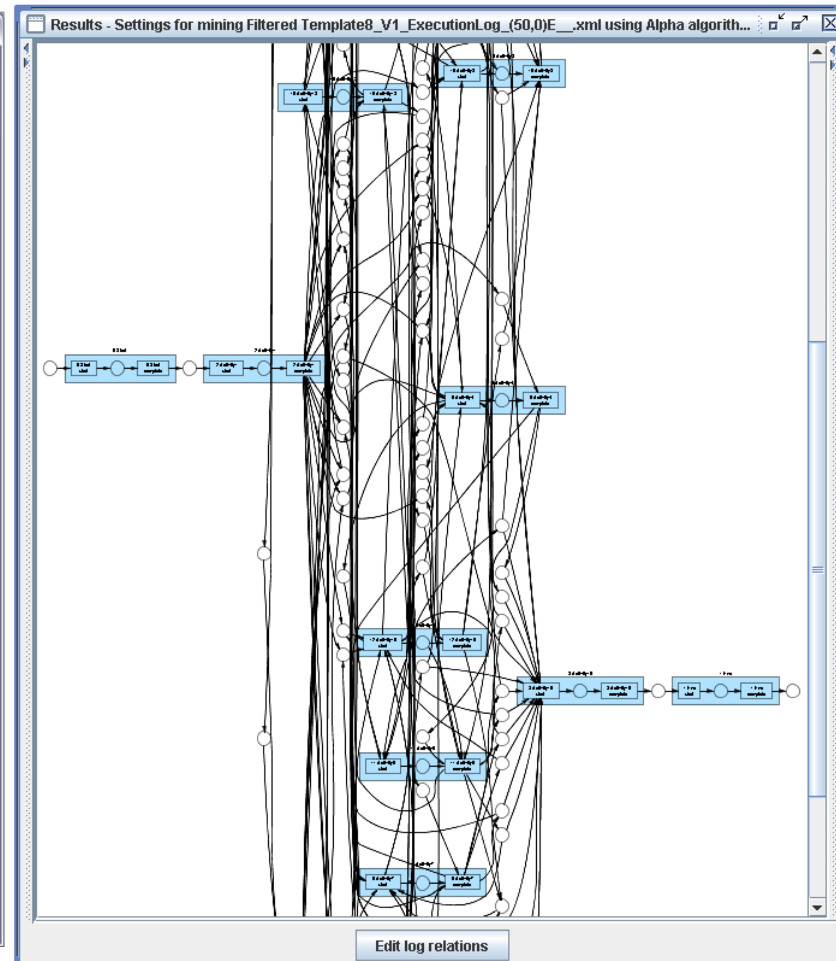
3 The alpha-Algorithm

- Problem 1: Parallel branches

Start process

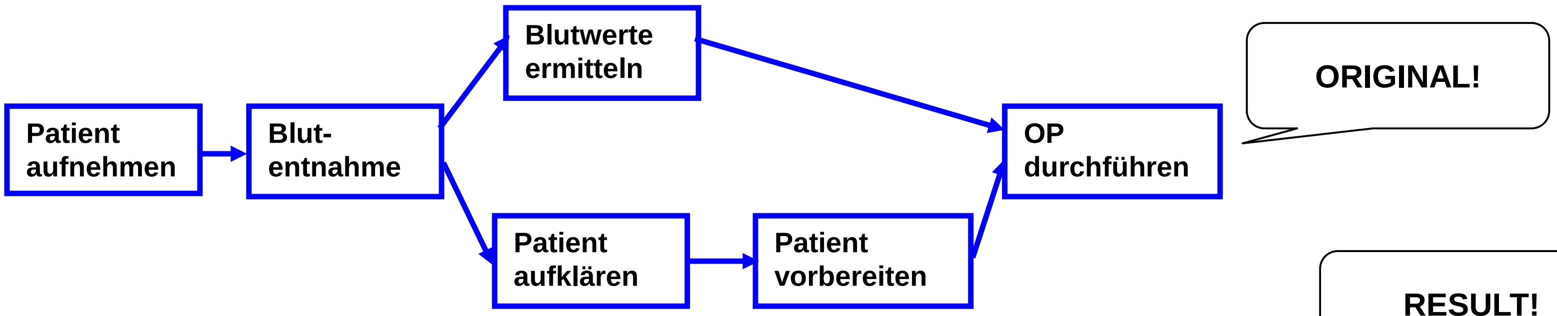


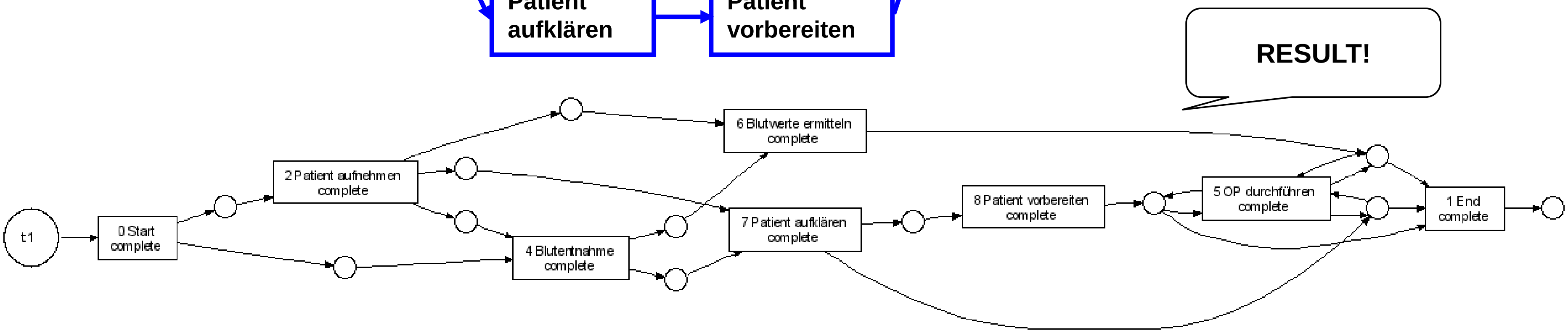
Result α-algorithm



3 The alpha-Algorithm

•Problem 2: missing or wrong log data (noise)





α-algorithm, 100 instances, 5% noise

- Further problems:
  - Noise (missing or errone
  - small/ big number of log
  - Parallel branchings with high number of branches

✓ **Heuristics Miner**

✓ **Genetic Mining [MWA05]**

Contents

Heuristics Miner

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations **based on their frequencies**
4. Build the net based on inferred relations
5. Output the net

Weijters, AJMM (Ton), van der WMP (Wil) Aalst, und de AKA (Ana Karla) Medeiros. 2006. *Process mining with the HeuristicsMiner algorithm*. Technische Universiteit Eindhoven. http://repository.tue.nl/615595.

Heuristics Miner:

Let W be an event log over T, and $a, b \in T$:

- $|a >_W b|$ is the number of times $a >_W b$ occurs in W,
- $a \Rightarrow_W b = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

Insight:

The more frequently a task A directly follows another task B, and the less frequently the opposite occurs, the higher the probability that A causally follows B!

Source: T. Weijters, A. K. de Medeiros: Process Mining course, TU Eindhoven, 2009 (http://prom.win.tue.nl/research/wiki/courses/processmining)

Traces σ in log W:

Case 1: <A, B, C, D>

Case 2: <A, C, B, D>

Case 3: <A, B, C, D>

Case 4: <A, C, B, D>

Case 5: <A, E, D>

Case 1: $A >_W B$, $B >_W C$, $C >_W D$

Case 2: $A >_W C$, $C >_W B$, $B >_W D$

Case 3: $A >_W B$, $B >_W C$, $C >_W D$

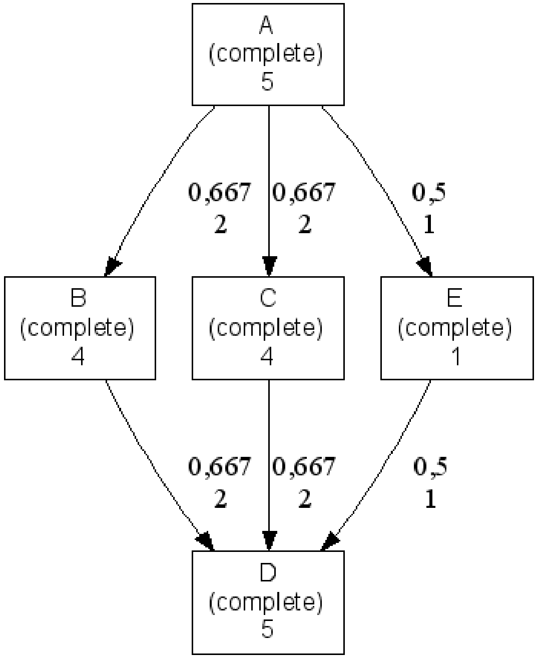
Case 4: $A >_W C$, $C >_W B$, $B >_W D$

Case 5: $A >_W E$, $E >_W D$

Let W be an event log over T, and $a, b \in T$:

- $|a >_W b|$ is the number of times $a >_W b$ occurs in W,
- $a \Rightarrow_W b = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

Heuristics Miner



4 Heuristic and Genetic Miner

Traces σ in log W:

Case 1: <A, B, C, D>

Case 2: <A, C, B, D>

Case 3: <A, B, C, D>

Case 4: <A, C, B, D>

Case 5: <A, E, D>

Case 1: $A >_W B$, $B >_W C$, $C >_W D$

Case 2: $A >_W C$, $C >_W B$, $B >_W D$

Case 3: $A >_W B$, $B >_W C$, $C >_W D$

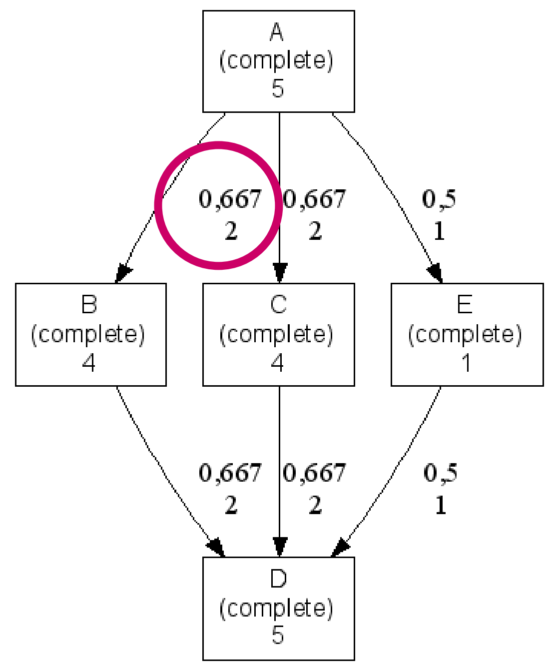
Case 4: $A >_W C$, $C >_W B$, $B >_W D$

Case 5: $A >_W E$, $E >_W D$

Let W be an event log over T, and $a, b \in T$:

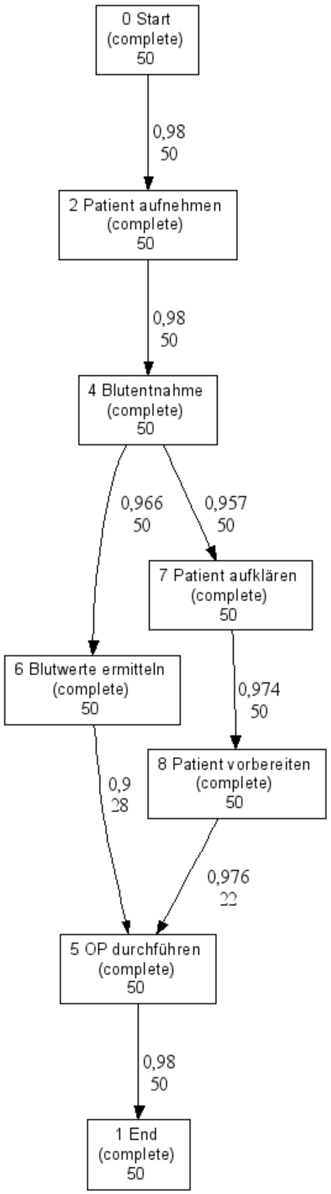
- $|a >_W b|$ is the number of times $a >_W b$ occurs in W,
- $a \Rightarrow_W b = \left(\dfrac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

Heuristics Miner



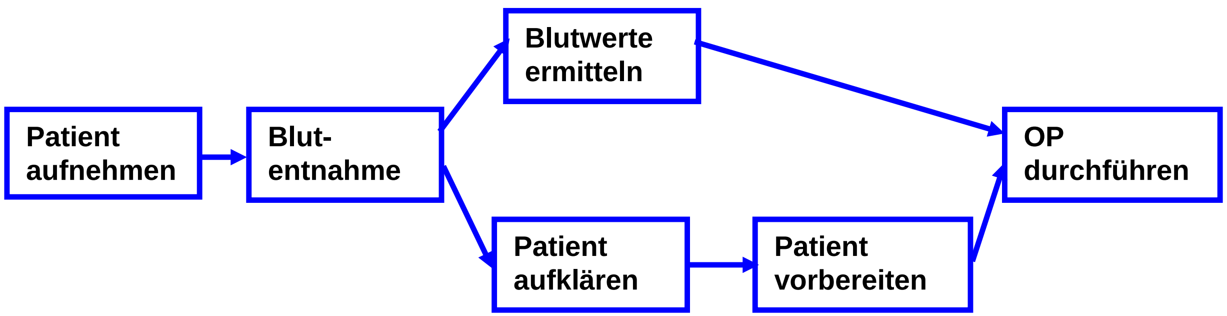
Heuristics Miner

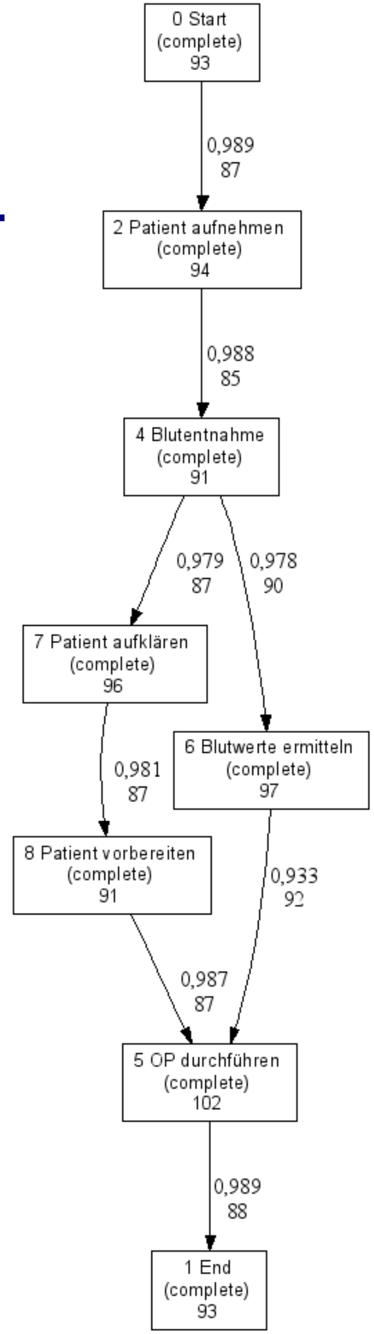
- Let us have a look at our healthcare example
- Results for 50 workflow logs
- We learned that Heuristics Miner can better deal with noise than the α-algorithm
- Let us try…



Heuristics Miner

- Healthcare example
- 1000 workflow logs, 5% noise





4 Heuristic and Genetic Miner

Workflow Instance 1
- Preparation
- Travel by train
- Conference
- Visit brewery
- Dinner
- Travel by train
- Travel refund

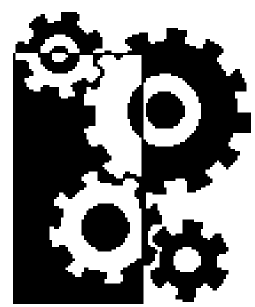
Workflow Instance 2
- Preparation
- Travel by train
- Conference
- Presentation
- Visit brewery
- Dinner
- Travel by train
- Travel refund

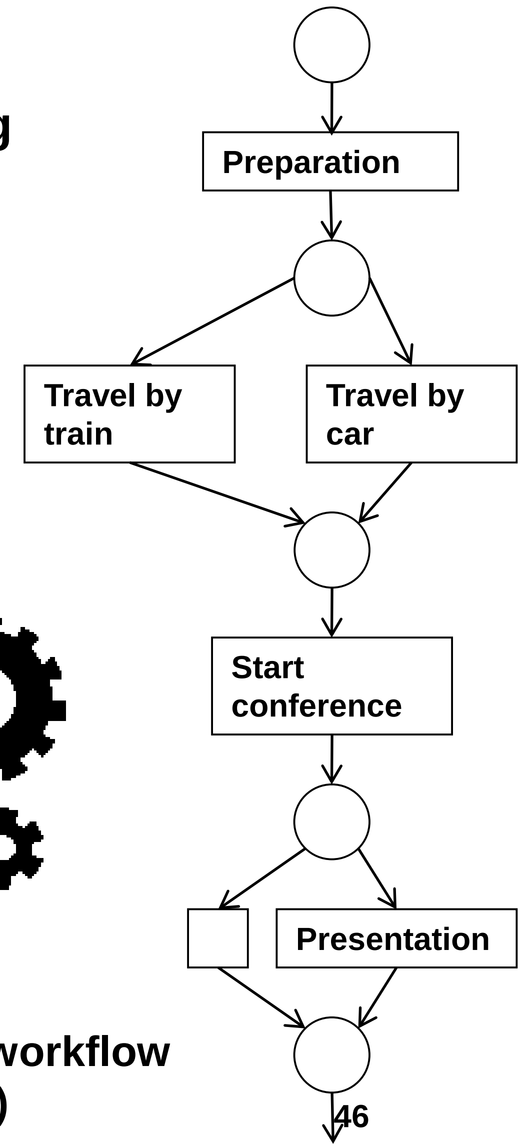
Workflow Instance 3
- Preparation
- Travel by car
- Conference
- Visit brewery
- Dinner
- Pay parking
- Travel by car
- Travel refund

Workflow Instance 4
- Preparation
- Travel by train
- Conference
- Presentation
- Visit brewery
- MISSING
- Travel by train
- Travel refund

Execution logs for a business process

Genetic Mining





Corresponding workflow model (Petri Net)

4 Heuristic and Genetic Miner

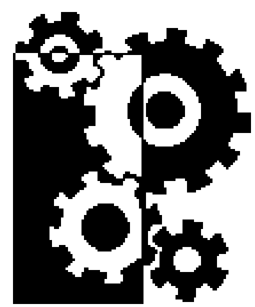
•Strengths of genetic algorithms:

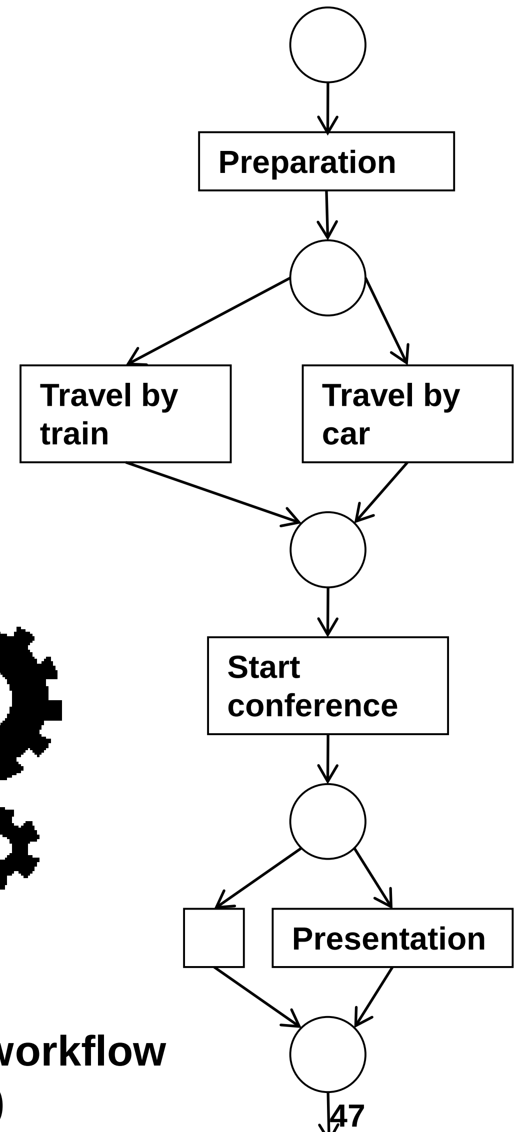
- Highly complex problems
- Noisy logs

Workflow Instance 3
Preparation
Travel by car
Conference
Visit brewery
Dinner
Pay parking
Travel by car
Travel refund

Workflow Instance 4
Preparation
Travel by train
Conference
Presentation
Visit brewery
MISSING
Travel by train
Travel refund

Execution logs for a business process



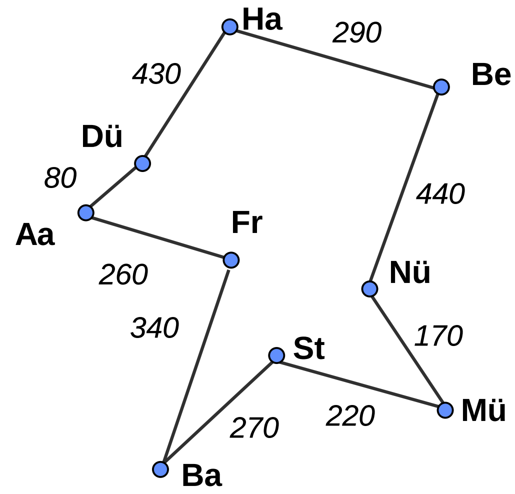


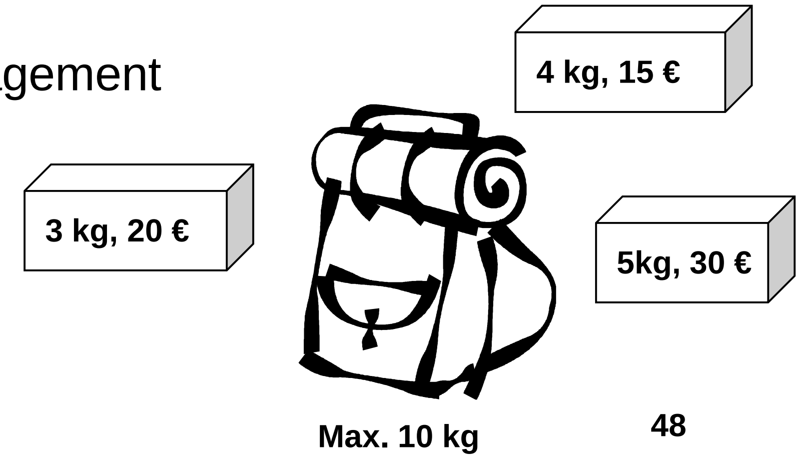
Corresponding workflow model (Petri Net)

Many „traditional“ Computer Science problems

Example taken from [Jung94]

- Traveling Salesman Problem
  - Combinatoric optimization
  - Used, for example, in logistics
- Knapsack Problem
  - Combinatoric optimization
  - Used, for example, for portfolio management





The following introductory slides follow Eiben, Smith: Introduction to Evolutionary Computing, Springer 2003

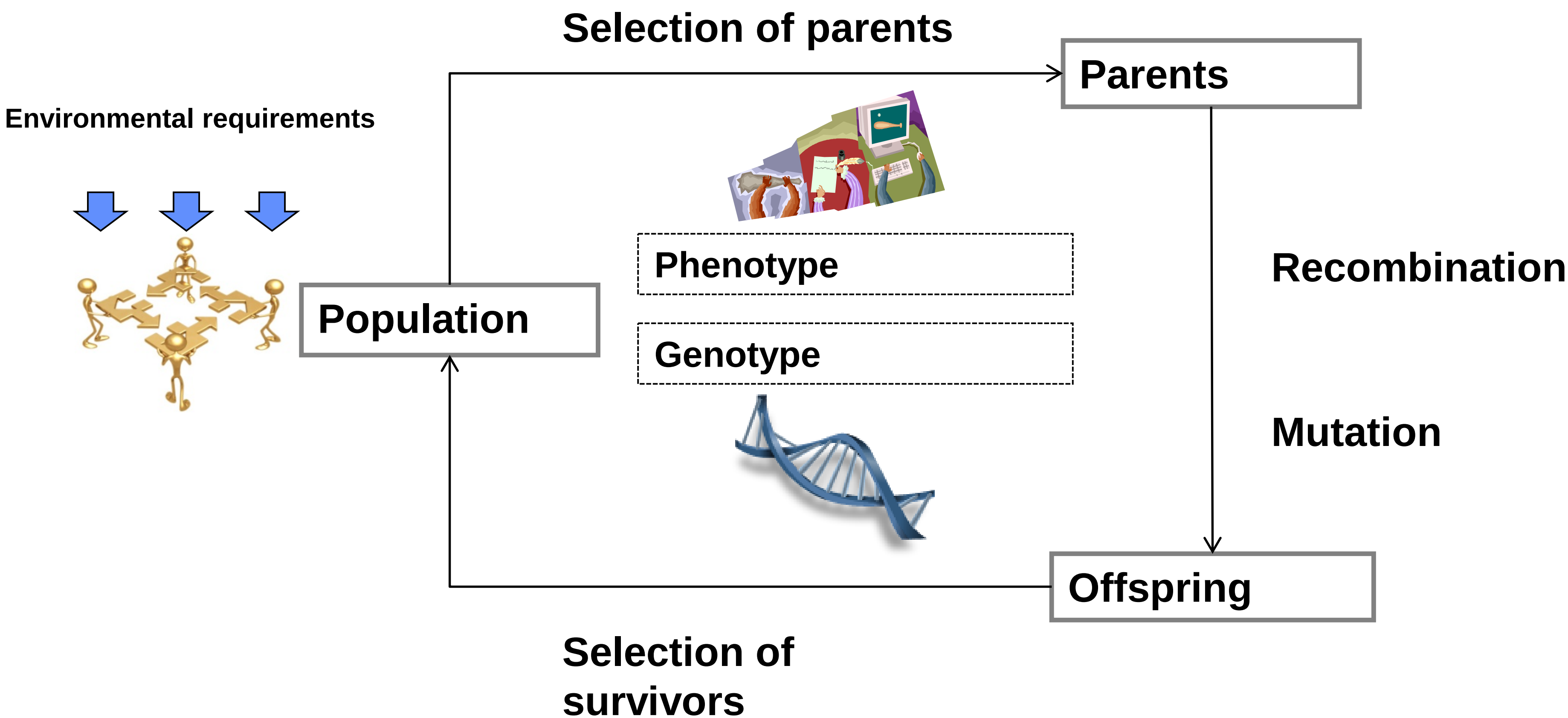
Observation of natural problem solvers

Evolutionary processes

Evolution ⟺	**Problem solving**
Environment ⟺	**Search space**
Individual ⟺	**Solution candidate**
Fitness ⟺	**Quality**

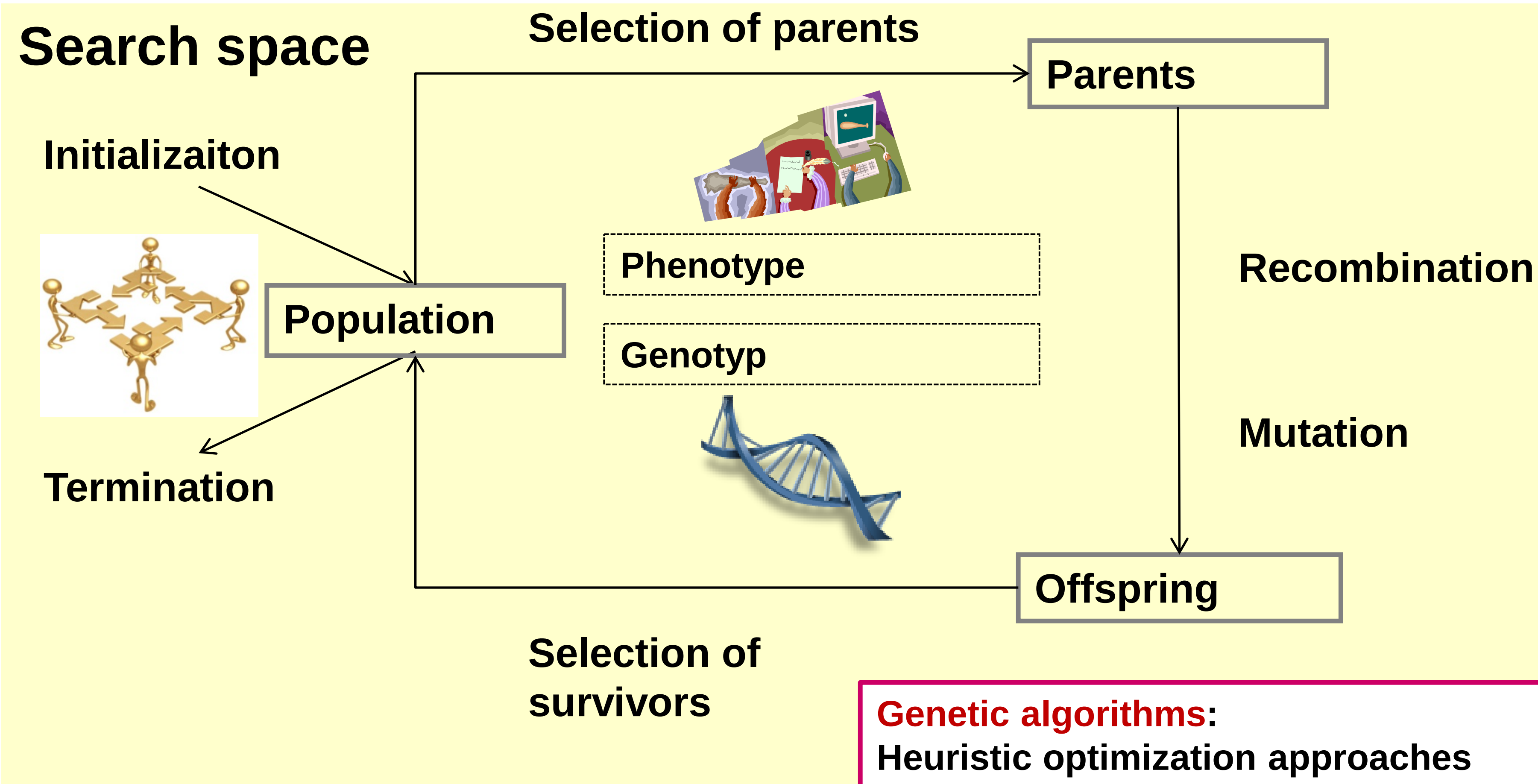
4 Excursus Genetic algorithms

Basic idea:



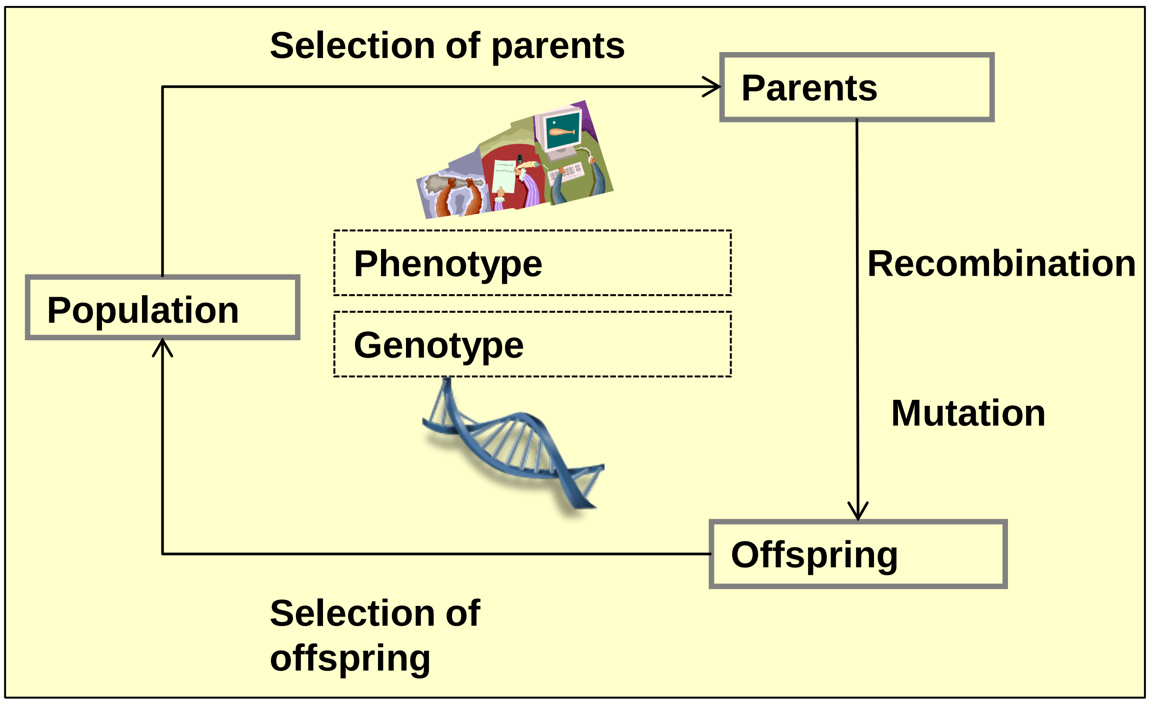
4 Excursus Genetic algorithms

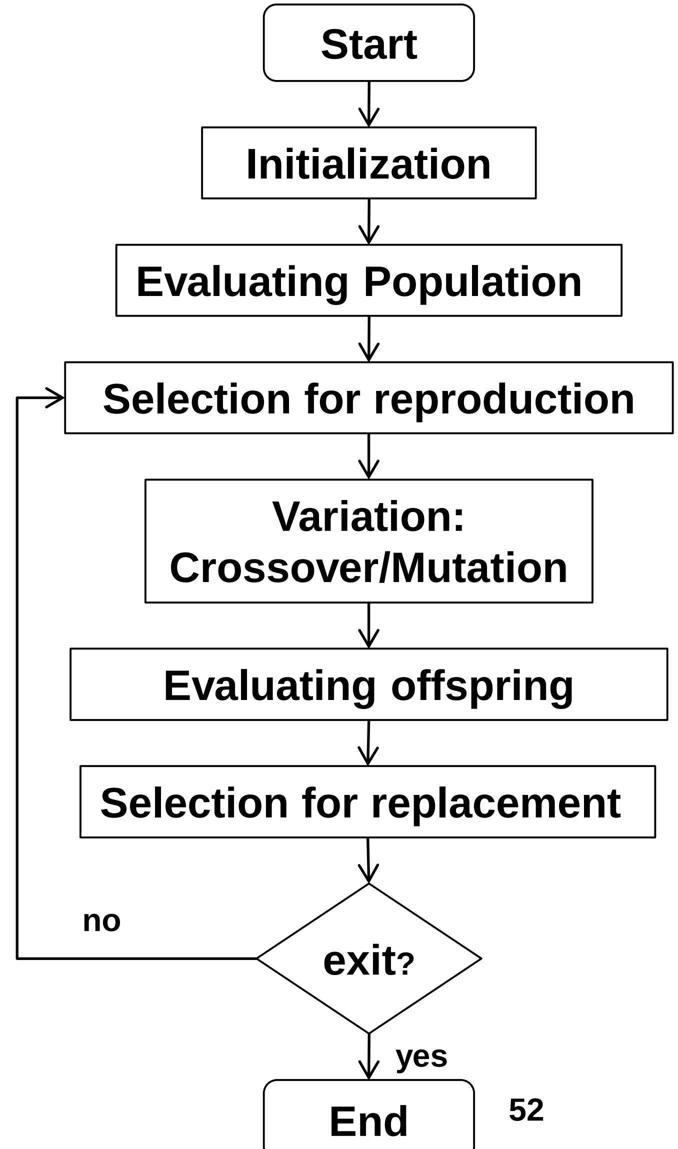
Basic idea:



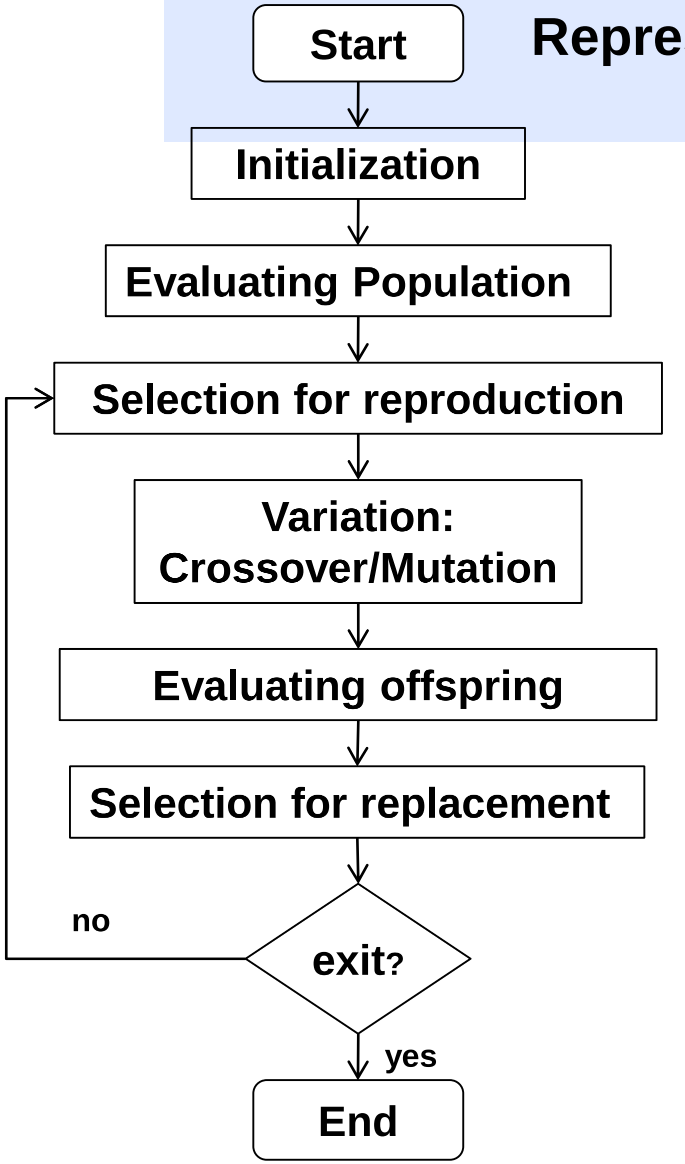
Genetic algorithms:
Heuristic optimization approaches based on evolution belong to class of evolutionary algorithms

Basic structure:



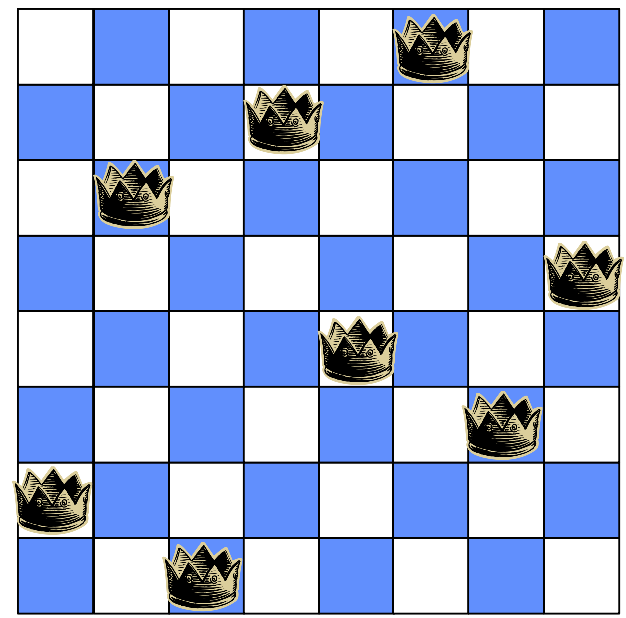


Representing individuals

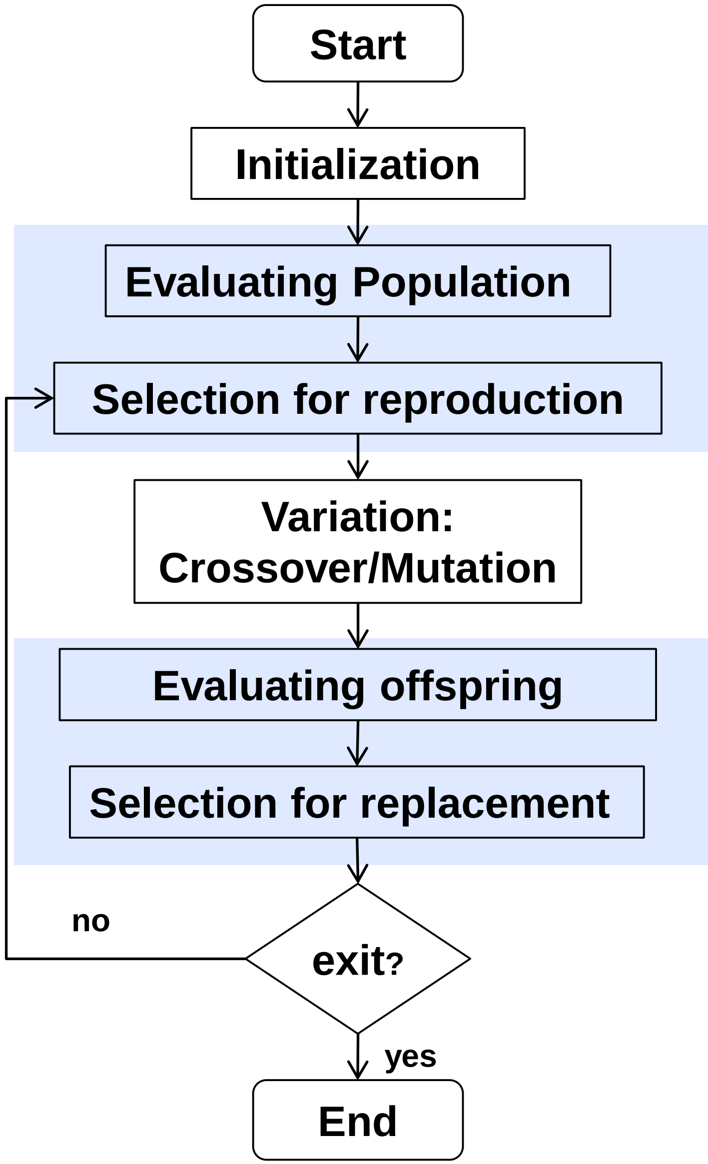


Representing individuals

- „good“ coding of individuals
- example: 8-Dames-Problem
  - Phenotype: one specific constellation
  - Coding genotype:
    - Matrix representation
    - Can we do it better?
    - $g = \langle i_1, \ldots, i_8 \rangle$:
    - n-th column, dame on position i_n
    - example: $g = \langle 2, 6, 1, 7, 4, 8, 3, 5 \rangle$



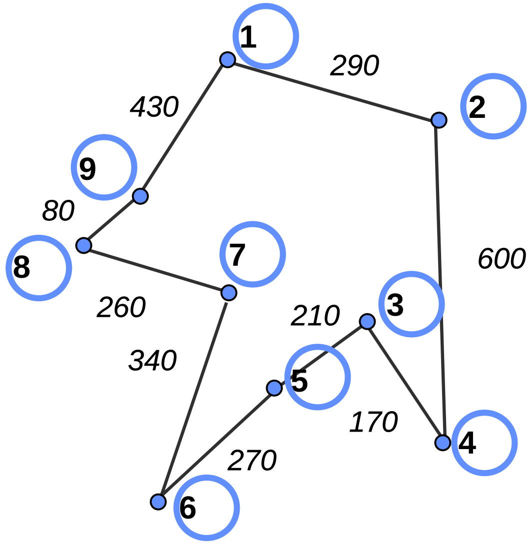
4 Excursus Genetic algorithms



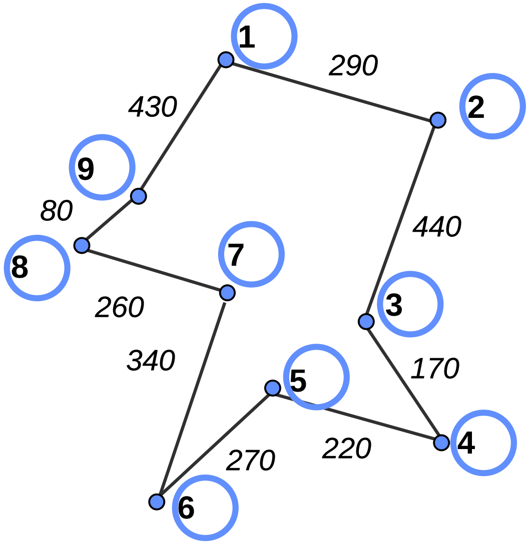
Selection criterion:

- Fitness function
- Determines quality of individuals
- Example: length of a trip with Traveling Salesman Problem

- ***Tour T1: Fitness f(T1) = 2650***



Tour T2: Fitness f(T2) = 2500



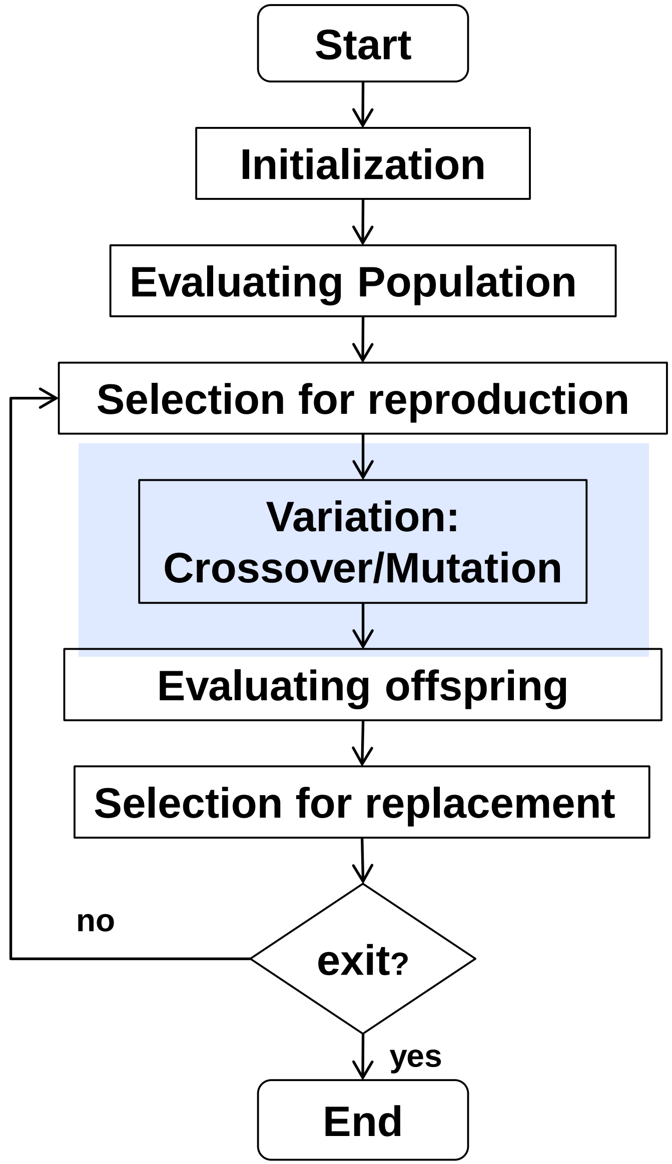
Selection of parents:

- Parents: creating offspring
- Goal: increased quality (fitness)
- „Better“ individuals are selected with higher priority
- However, also individuals with lower fitness values are selected with some probability → diversity

Selection of offspring:

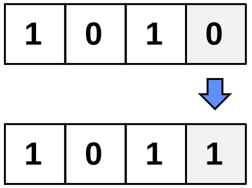
- Population size constant
- Selection based on **fitness**

4 Excursus Genetic algorithms

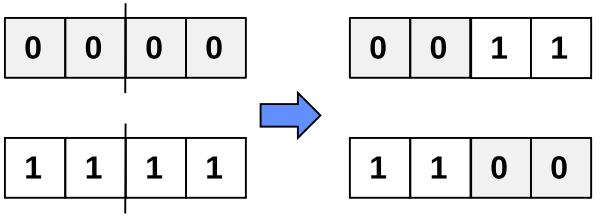


Variation operators:

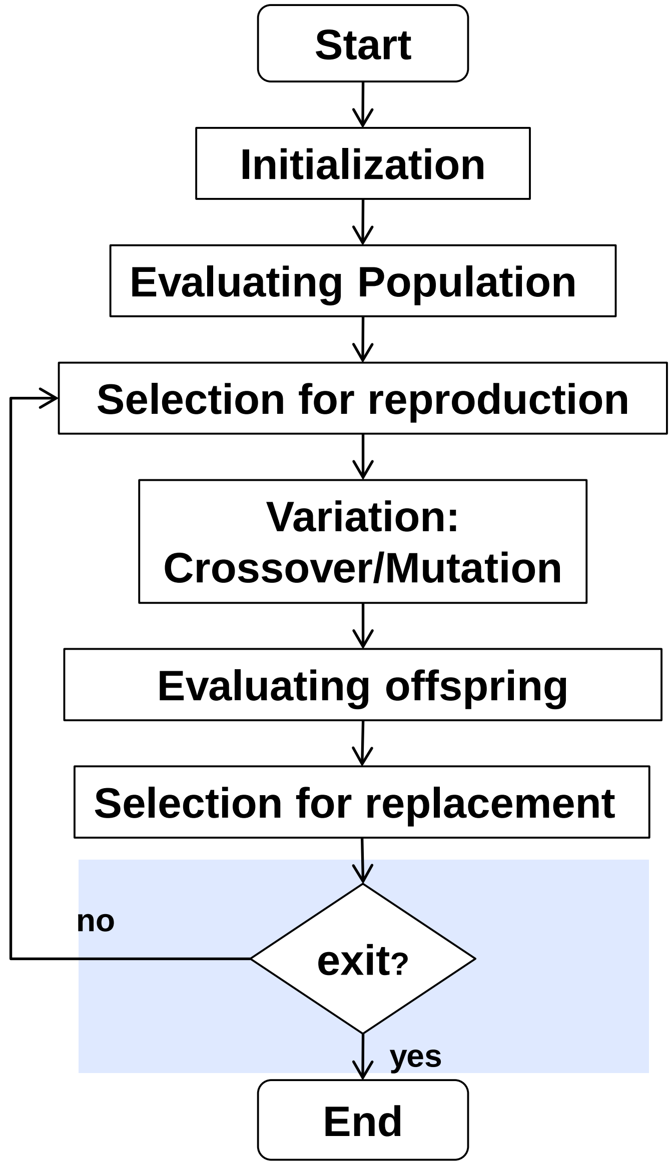
- Creating offspring
- **Mutation**:
  - Example: Sphynx cat
  - Unary operation



- **Recombination (Crossover)**:
  - Example: human being
  - In nature: binary operations



4 Excursus Genetic algorithms



Termination

- If optimal fitness value known → termination with optimal solution
- However not sufficient (**WHY??**):
  - Time limit
  - Upper bound for fitness evaluations

Representation of individuals:

- Finding an adequate genotype (coding)
- Often the most difficult part
- Depends on application
- Rules of thumb:
  - Data structure should corresponds as much as possible to natural representation
  - If possible: genotypes represent valid solutions
  - If possible: variation operators do not destroy validity

Possible representations:

- First possibility: **Bit-Strings $s \in \{0,1\}^n$**
- Example:
  - MAXONE: Fitness function f counts number of „1“ entries in bit string
  - $s_1 = 1111010101$ $f(s_1) = 7$
- Evaluation:
  - Well suited for analytical purposes
  - Representation of integer values:
    - Integer value corresponds to phenotype (e.g. x = 12)
    - Bit String corresponds to geno type (e.g., $s = 01100$, $s \in \{0,1\}^5$)

Variation operators for Bit Strings:

- **Mutation**: Bit Flip with probabilities p_m per Bit

1	0	1	0	1	0	1	0	→	1	0	1	1	1	0	1	1

- **Recombination**:
  - One-Point Crossover

1	0	1	0	1	0	1	0	→	1	0	1	1	1	0	1	0
1	0	1	1	1	0	1	1		1	0	1	0	1	0	1	1

  - N-Point Crossover (e.g. n = 2)

1	0	1	0	1	0	1	0	→	1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	1		1	0	1	1	1	0	1	1

4 Genetic Mining

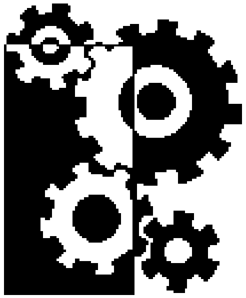
Workflow Instance 1
Preparation
Travel by train
Conference
Visit brewery
Dinner
Travel by train
Travel refund

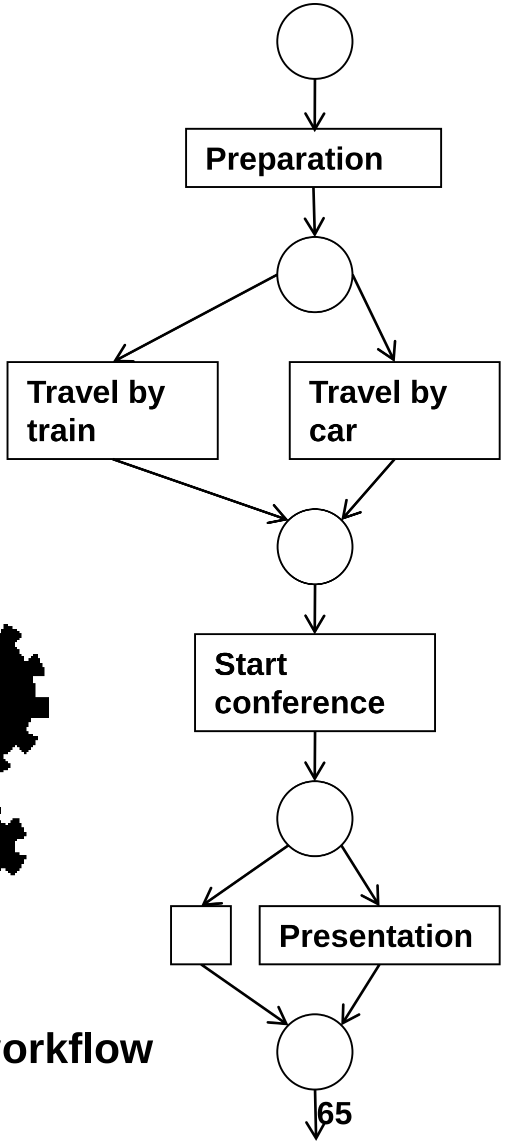
Workflow Instance 2
Preparation
Travel by train
Conference
Presentation
Visit brewery
Dinner
Travel by train
Travel refund

Workflow Instance 3
Preparation
Travel by car
Conference
Visit brewery
Dinner
Pay parking
Travel by car
Travel refund

Workflow Instance 4
Preparation
Travel by train
Conference
Presentation
Visit brewery
MISSING
Travel by train
Travel refund

Execution logs for a business process





Corresponding workflow model (Petri Net)

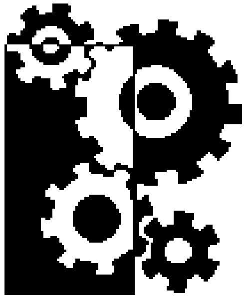
4 Heuristic and Genetic Miner

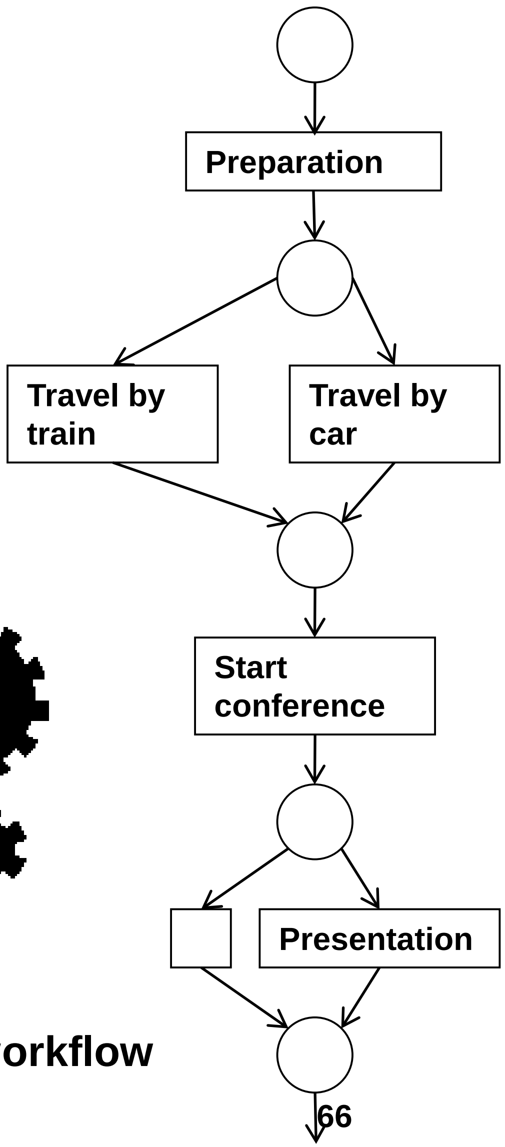
- **Strengths of genetic algorithms:**
  - Highly complex problems
  - Noisy logs

Workflow Instance 3
Preparation
Travel by car
Conference
Visit brewery
Dinner
Pay parking
Travel by car
Travel refund

Workflow Instance 4
Preparation
Travel by train
Conference
Presentation
Visit brewery
MISSING
Travel by train
Travel refund

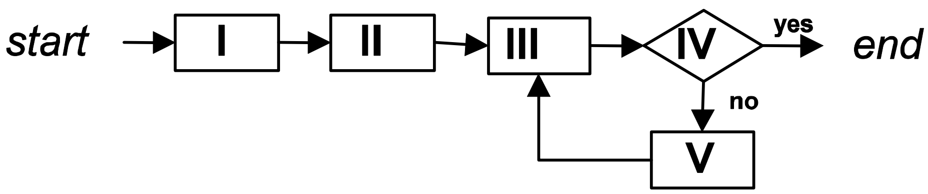
Execution logs for a business process





Corresponding workflow model (Petri Net)

Genetic Mining Algorithm:



Source: T. Weijters, A. K. de Medeiros: Process Mining course, TU Eindhoven, 2009 (http://prom.win.tue.nl/research/wiki/courses/processmining)

Internal Representation

Fitness Measure

Genetic Operators

Step	Description
I	*Read event log*
II	*Build the initial population* ●
III	*Calculate fitness of the individuals in the population* ● ●
IV	*Stop and return the fittest individuals?* ●
V	*Create next population – use elitism and genetic operators* ● ●

A.K.A. Medeiros, A.J.M.M. Weijters, und W.M.P. Aalst, “Genetic process mining: an experimental evaluation,” *Data Mining and Knowledge Discovery*, vol. 14, 2007, S. 245-304.

Internal Representation:

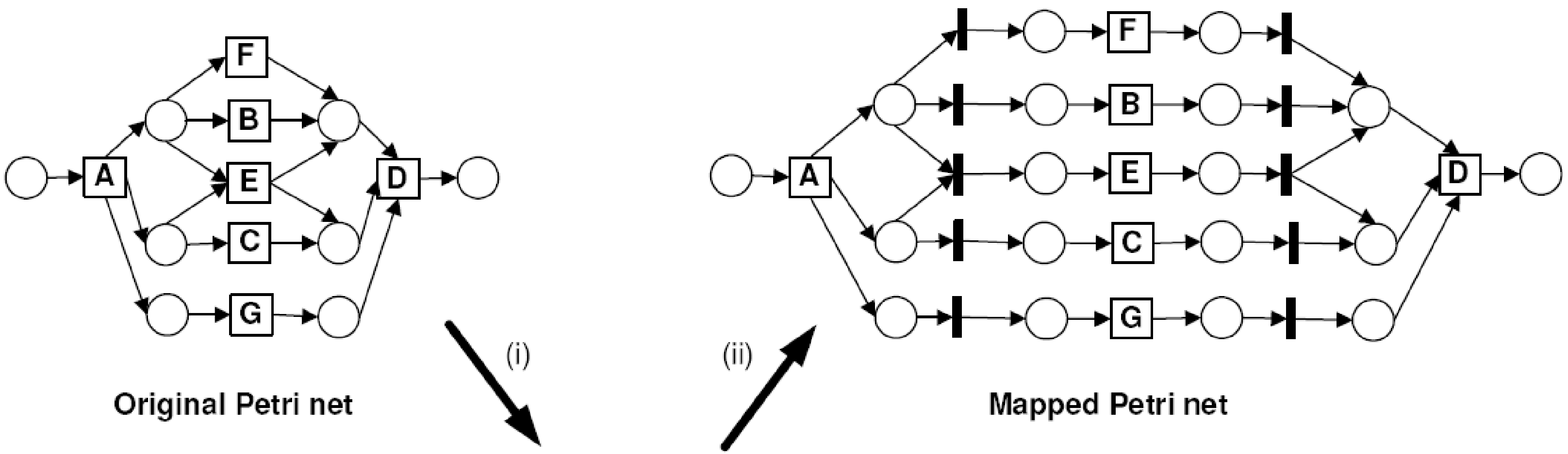
- Abstraction from Petri Nets → WHY?
- However, no information should be lost!
- Representation as causal matrix:

Genetic Mining

DEFINITION (CAUSAL MATRIX): A Causal Matrix is a tuple CM = (A, C, I, O), where
- A is a finite set of activities,
- $C \subseteq A \times A$ is the causality relation,
- $I: A \rightarrow \mathcal{P}(\mathcal{P}(A))$ is the input condition function
- $O: A \rightarrow \mathcal{P}(\mathcal{P}(A))$ is the output condition function,
such that
- $C = \{(a_1, a_2) \in A \times A \mid a_1 \in \cup\, I(a_2)\}$
- $C = \{(a_1, a_2) \in A \times A \mid a_2 \in \cup\, O(a_1)\}$
- $C \cup \{(a_o, a_i) \in A \times A \mid a_o \bullet = \emptyset \wedge \bullet a_i = \emptyset\}$ is a strongly connected graph,

Genetic Mining Example:

Source: A.K.A. Medeiros, A.J.M.M. Weijters, und W.M.P. Aalst, "Genetic process mining: an experimental evaluation," *Data Mining and Knowledge Discovery*, vol. 14, 2007, S. 245-304.



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}}

Causal Matrix

Mapping between Petri Net and Causal Matrix

Genetic Mining

Definition ($\Pi_{PN \rightarrow CM}$) *Let* $PN = (P, T, F)$ *be a Petri net. The mapping of* PN *is a tuple* $\Pi_{PN \rightarrow CM}(PN) = (A, C, I, O)$, *where*

- $A = T$,
- $C = \{(t_1, t_2) \in T \times T \mid t_1 \bullet \cap \bullet t_2 \neq \emptyset\}$,
- $I \in T \rightarrow \mathcal{P}(\mathcal{P}(T))$ *such that* $\forall_{t \in T}\ I(t) = \{\bullet p \mid p \in \bullet t\}$,
- $O \in T \rightarrow \mathcal{P}(\mathcal{P}(T))$ *such that* $\forall_{t \in T}\ O(t) = \{p \bullet \mid p \in t \bullet\}$.

Source: A.K.A. Medeiros, A.J.M.M. Weijters, und W.M.P. Aalst, "Genetic process mining: an experimental evaluation," ***Data Mining and Knowledge Discovery*****, vol. 14, 2007, S. 245-304.**

Genetic Mining - Initial Population:

- Individuals are causal matrices
- Given a log, all individuals in any population of the genetic algorithm have the same set of activities (or tasks) A. This set contains the tasks that appear in the log.
- The setting of the causality relation C can be done via a completely random approach or a heuristic one.
- The random approach uses 50% probability for establishing (or not) a causality relation between two task in A
- The heuristic approach uses the information in the log to determine the probability that two tasks are going to have a causality relation set:
- “The more often a task t_1 is directly followed by a task t_2 (i.e. the subtrace “t_1, t_2" appears in traces in the log), the higher the probability that individuals are built with a causality relation from t_1 to t_2 (i.e., $(t_1, t_2) \in C$)“

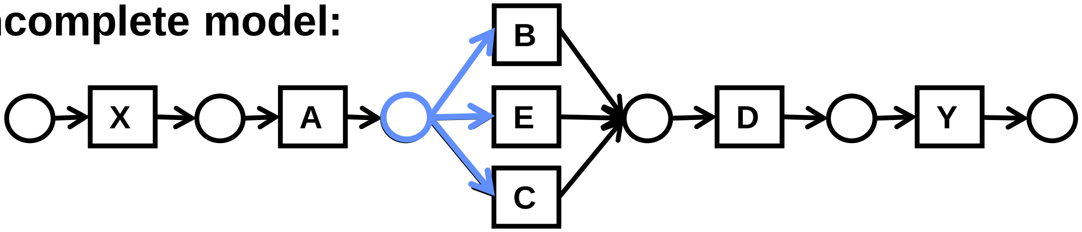
Source: A.K.A. Medeiros, A.J.M.M. Weijters, und W.M.P. Aalst, “Genetic process mining: an experimental evaluation,” *Data Mining and Knowledge Discovery*, vol. 14, 2007, S. 245-304.

Gentic Mining - Fitness function

Punish for the amount of non producable traces

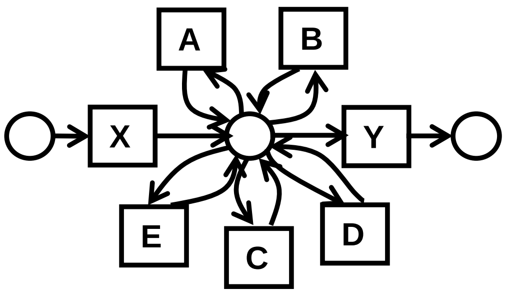
Process instance	Execution logs
1	X, A, C, D, Y
2	X, B, C, E, Y
3	X, A, C, D, Y
4	X, B, C, E, Y
5	X, B, C, E, Y
6	X, A, C, D, Y

Incomplete model:



Not producable
X, B, C, E, Y

Imprecise model:

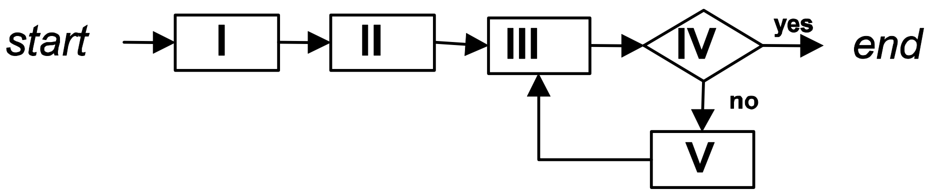


Additionally producable, e.g.,
X, A, A, A, B, Y

Punish for the amount of enabled tasks during the parsing!

Genetic Mining Algorithm:

Source: T. Weijters, A. K. de Medeiros: Process Mining course, TU Eindhoven, 2009 (http://prom.win.tue.nl/research/wiki/courses/processmining)



Internal Representation

Fitness Measure

Genetic Operators

Step	Description
I	*Read event log*
II	*Build the initial population*
III	*Calculate fitness of the individuals in the population*
IV	*Stop and return the fittest individuals?*
V	*Create next population – use elitism and genetic operators*

Genetic Operators:

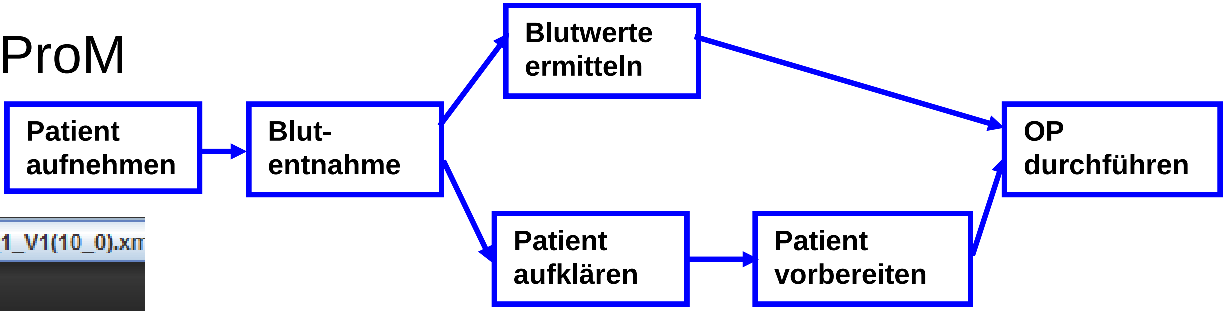
- *Elitism*: certain percentage of the best individuals is copied into next population
- *Crossover*:
  - two parents produce two offsprings
  - Selection of parents: tournament
- *Mutation*: Insertion of new material into current population
  - Randomly choose subset and add a task ($\in$ A) into subset
  - Randomly choose subset and remove task
  - Randomly redistribute elements in the subsets of I/O into new subsets
  - Example for Mutation: I(D) = {{F, B, E}, {E, C}, {G}}
    - Mutation by adding tasks: {{F, B, E}, {E, C}, {G, **D**}}
    - Mutation by removing tasks: {{F, B, E}, **{C}**, {G}}
    - Mutation by redistribution: {{F}, {E, C, B}, {G}, {E}}

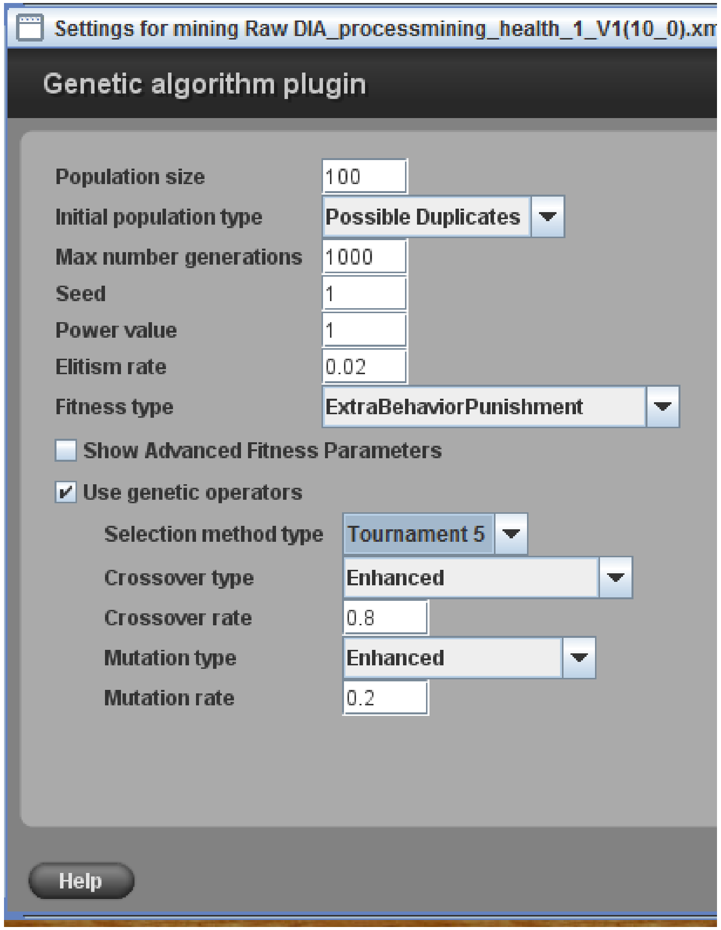
Genetic Mining - Stop Criteria:

- n generations allowed
- Fittest individual has not changed for n/2 generations in a row

Healthcare Example in ProM

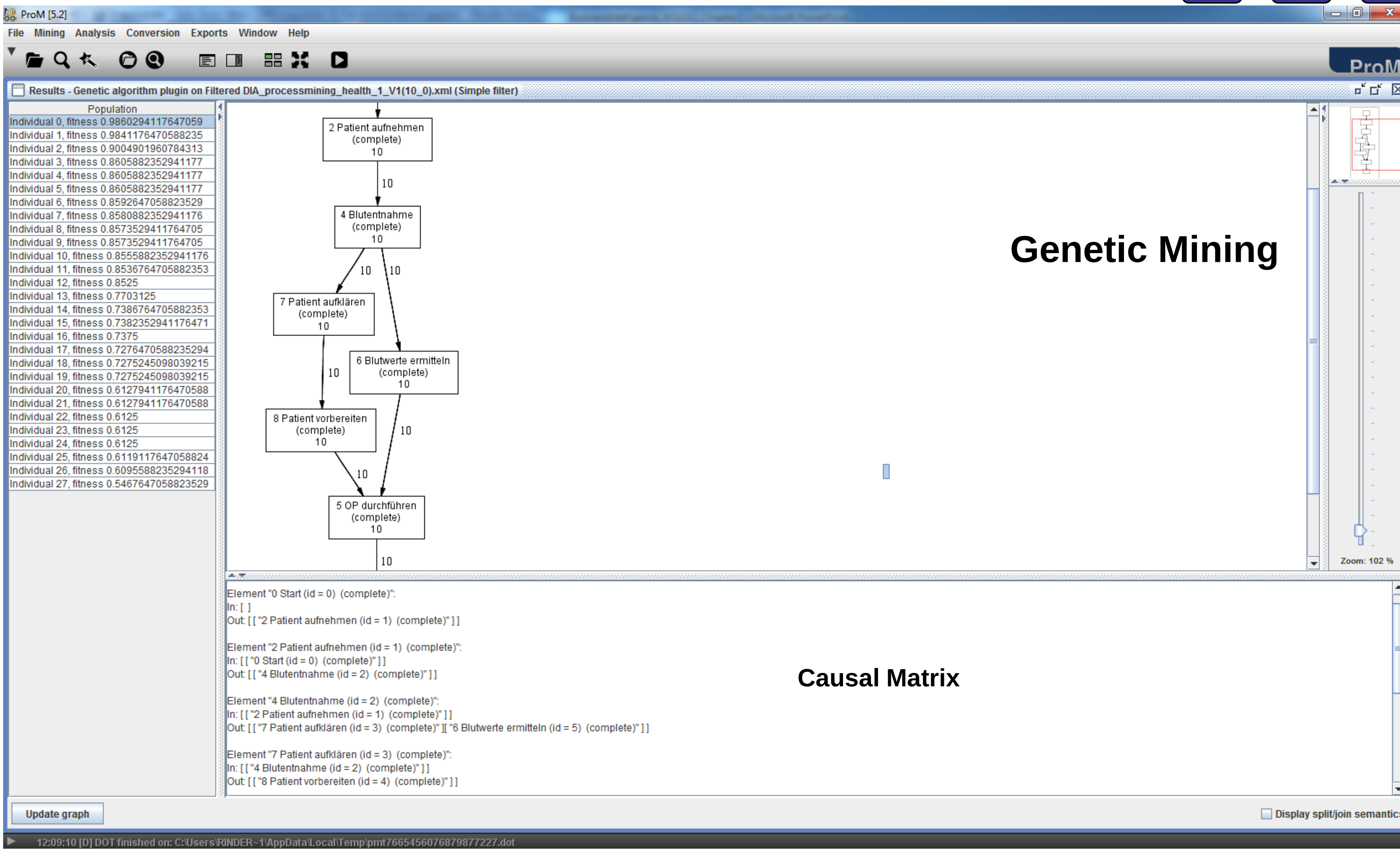
- Initial settings



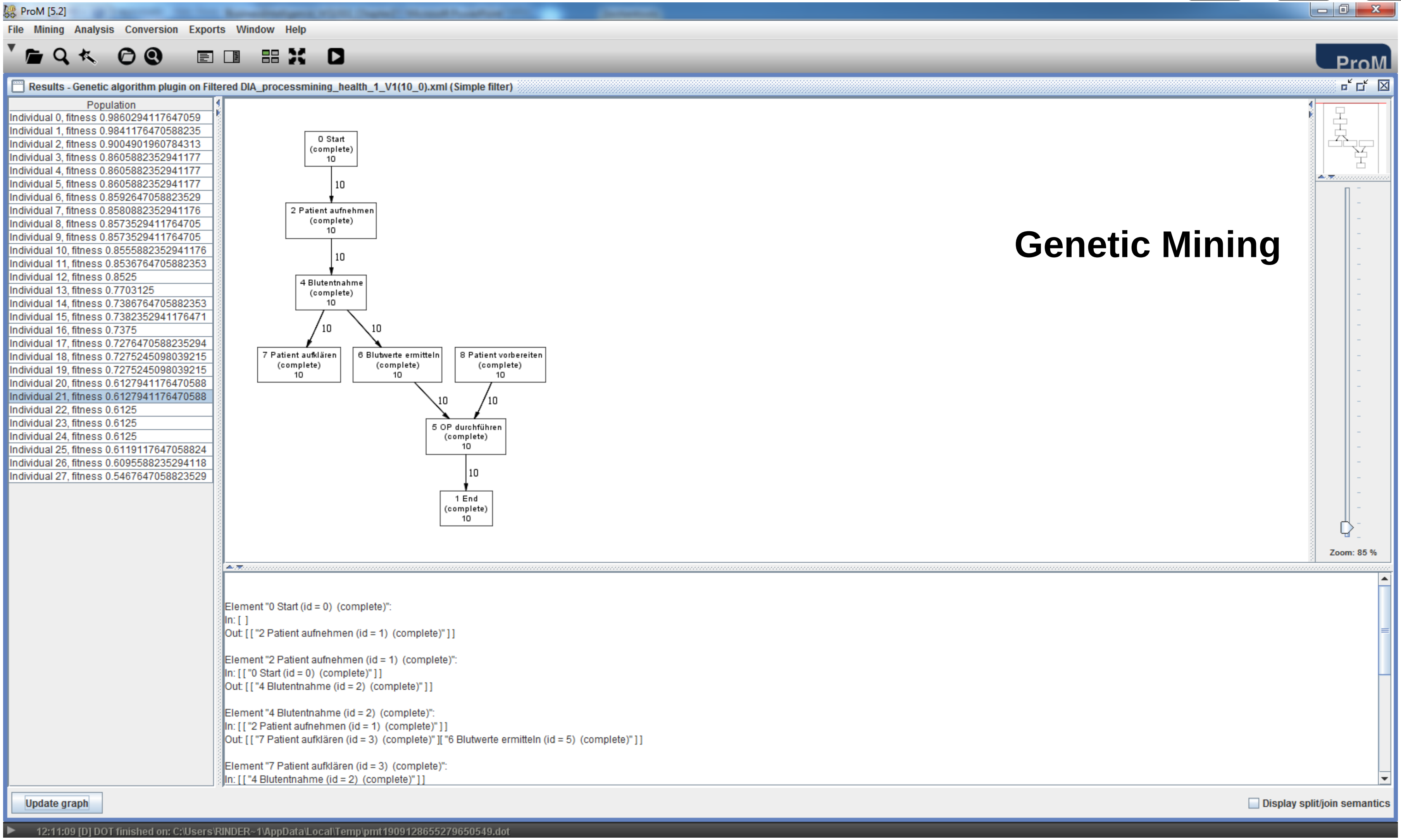


Genetic Mining

4 Heuristic and Genetic Miner



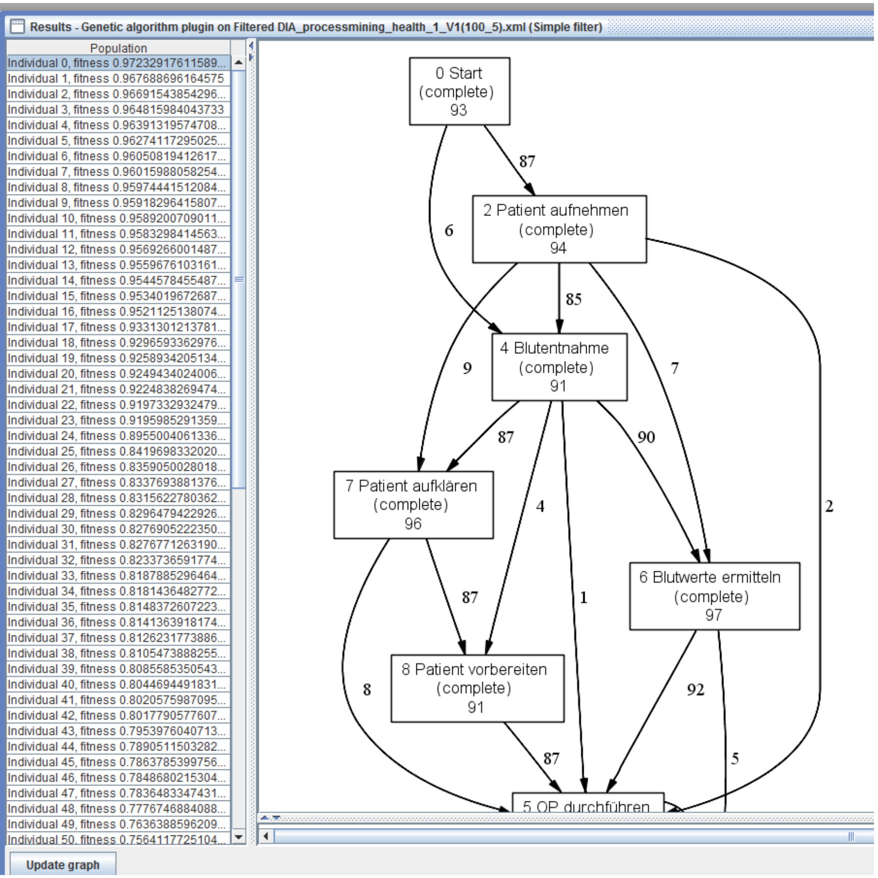
4 Heuristic and Genetic Miner

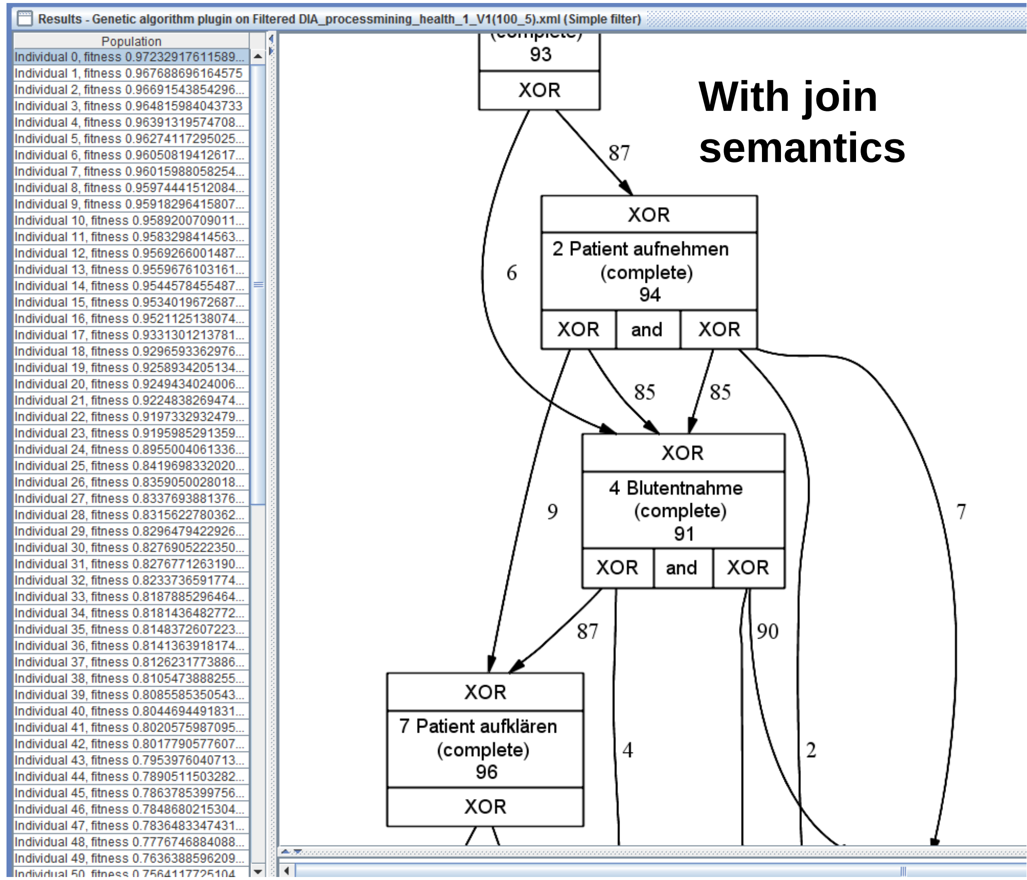


4 Heuristic and Genetic Miner

100 instances, 5% noise

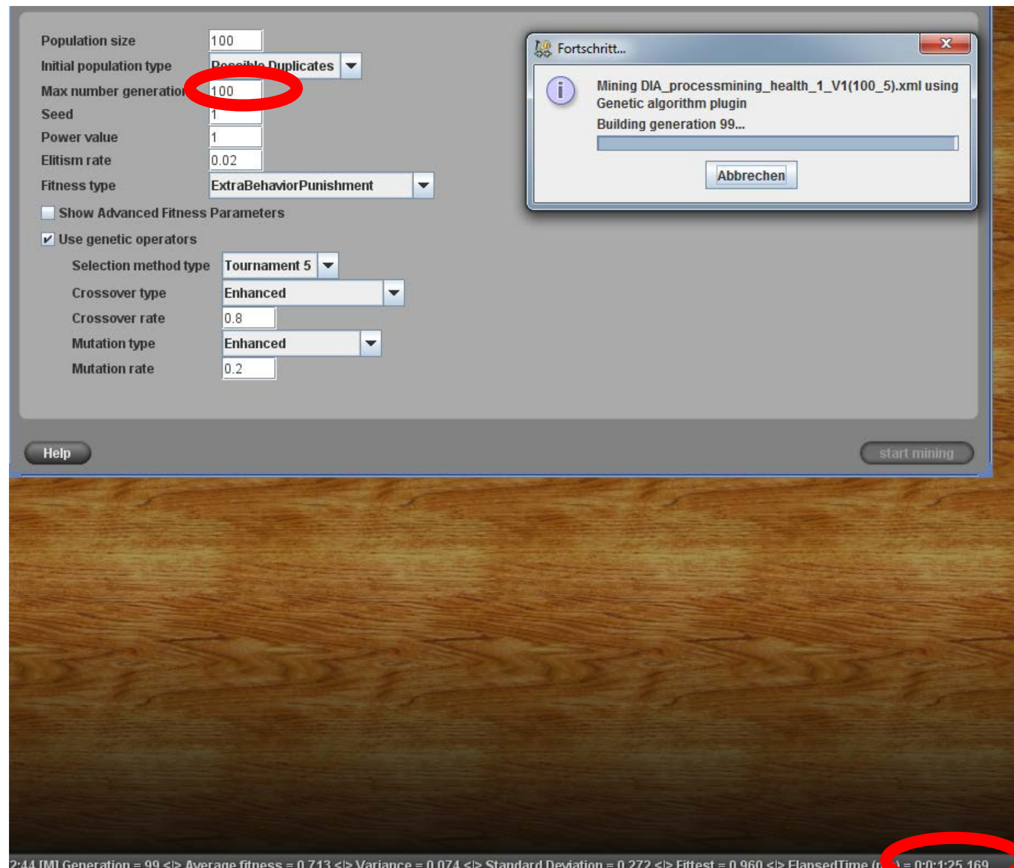
Genetic Mining





Genetic Mining - Discussion:

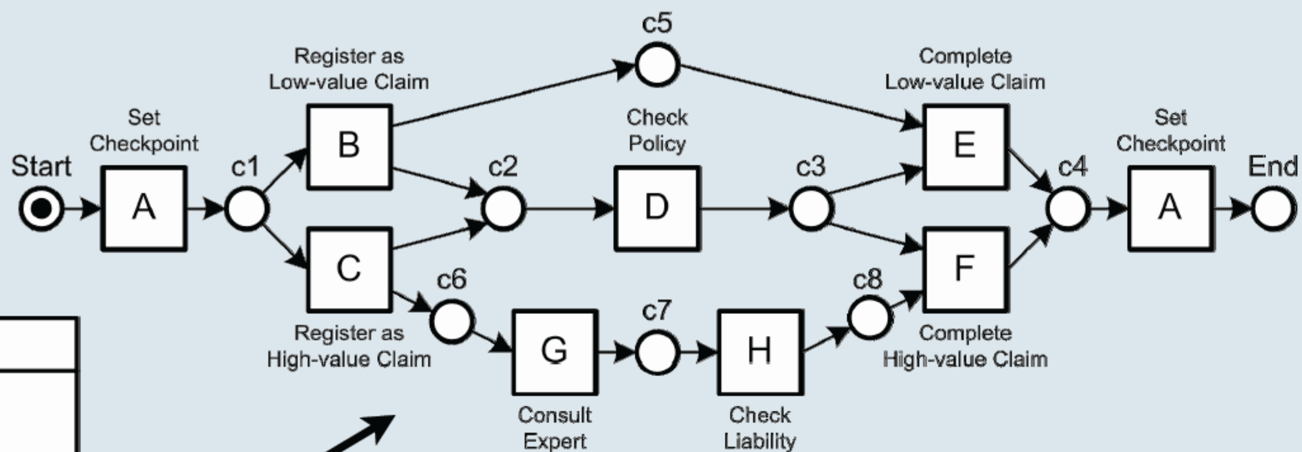
- Advantages
  - Tackles all common structural constructs
  - Robust to noise
- Disadvantages
  - Computational Time
  - Example:



Contents

Objectives:

- Compare process models and event data (e.g., for auditing)
- quantitatively measure conformance (i.e., metrics)
- locate deviations



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

Conformance?

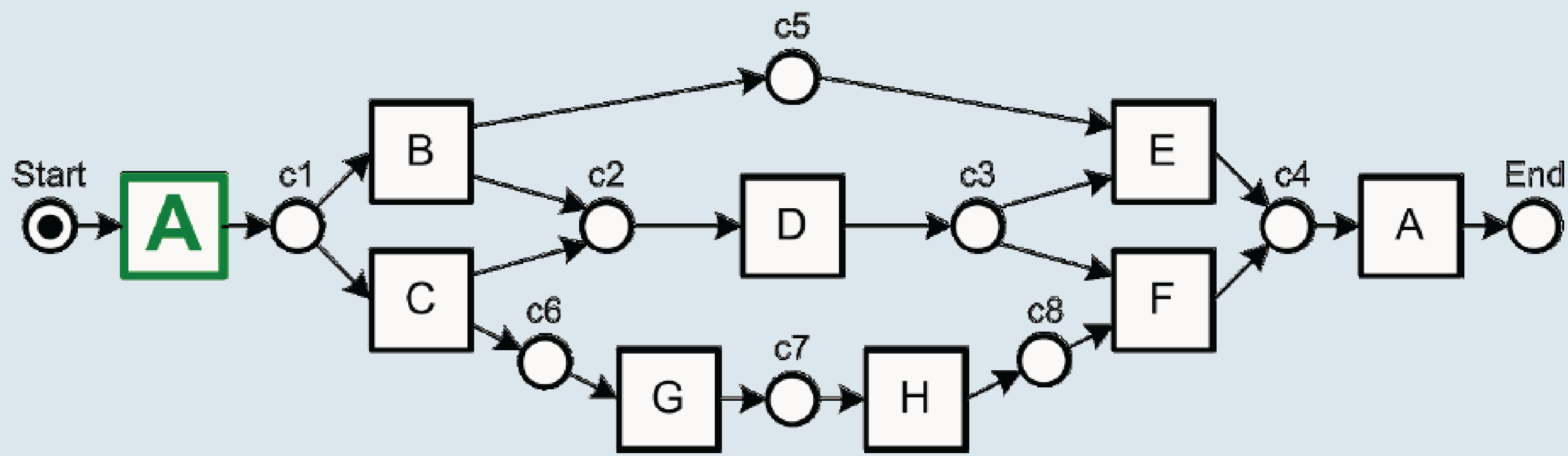
Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

Measuring fitness: Log replay analysis

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

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

$$\begin{array}{llll} \text{missing tokens} & = 0 & \text{consumed tokens} & = 0 \\ \text{remaining tokens} & = 0 & \text{produced tokens} & = 1 \end{array}$$



Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

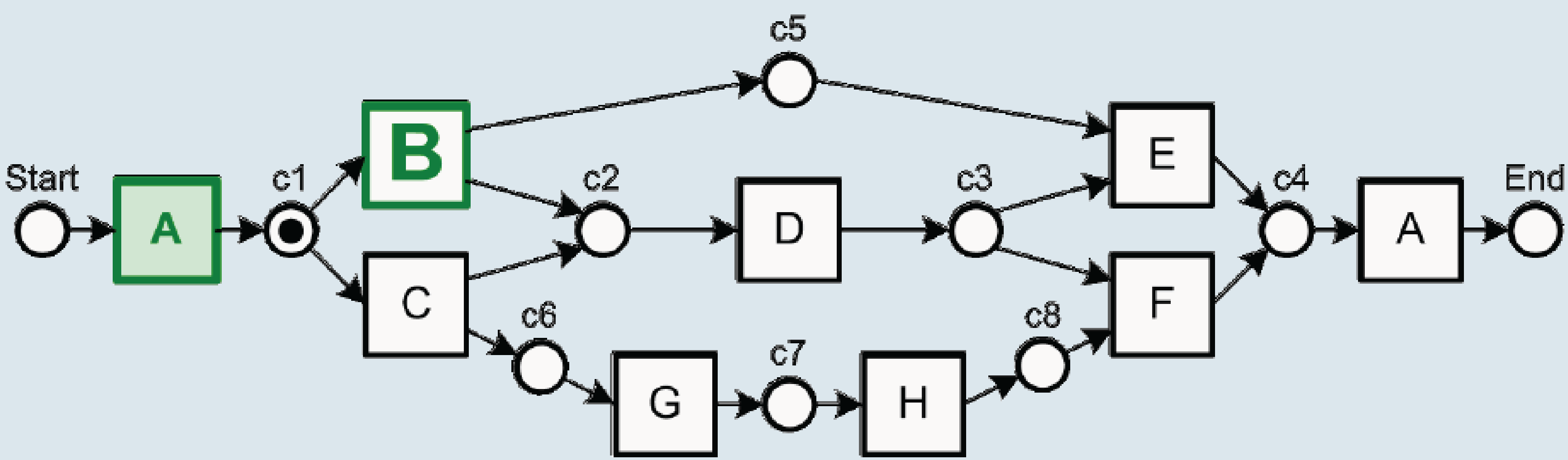
5 Conformance Checking

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

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

missing tokens $= 0$ consumed tokens $= 1$

remaining tokens $= 0$ produced tokens $= 2$



Source: Anne Rozinat: Conformance Checking of Service Behavior. ***BETA Conference, Eindhoven, 2007***

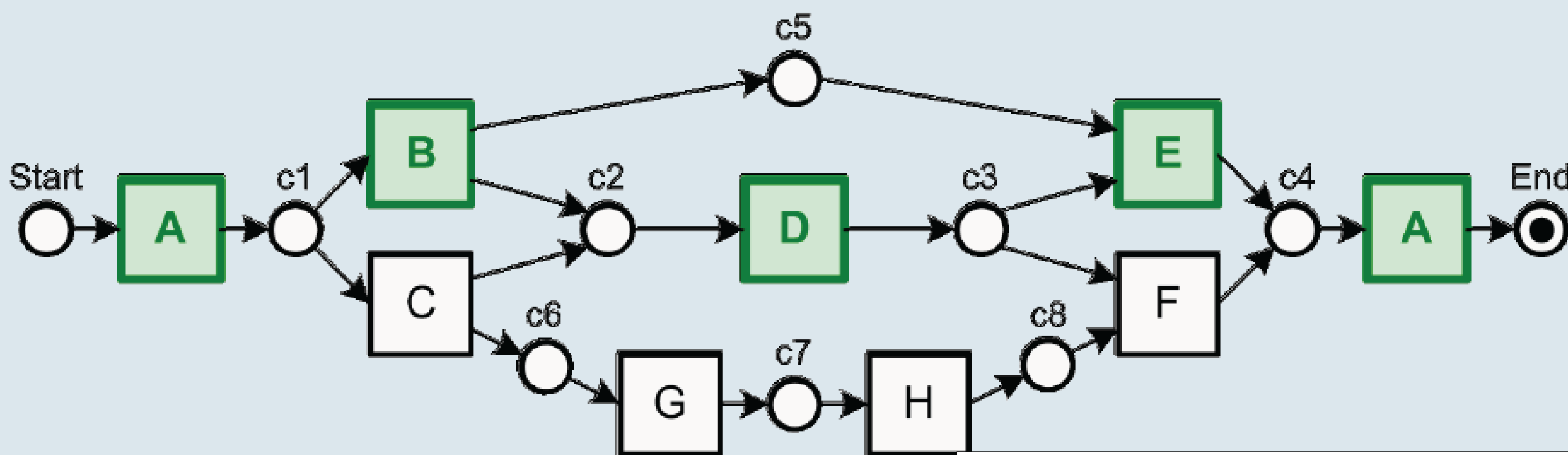
5 Conformance Checking

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

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

missing tokens $= 0$ consumed tokens $= 6$

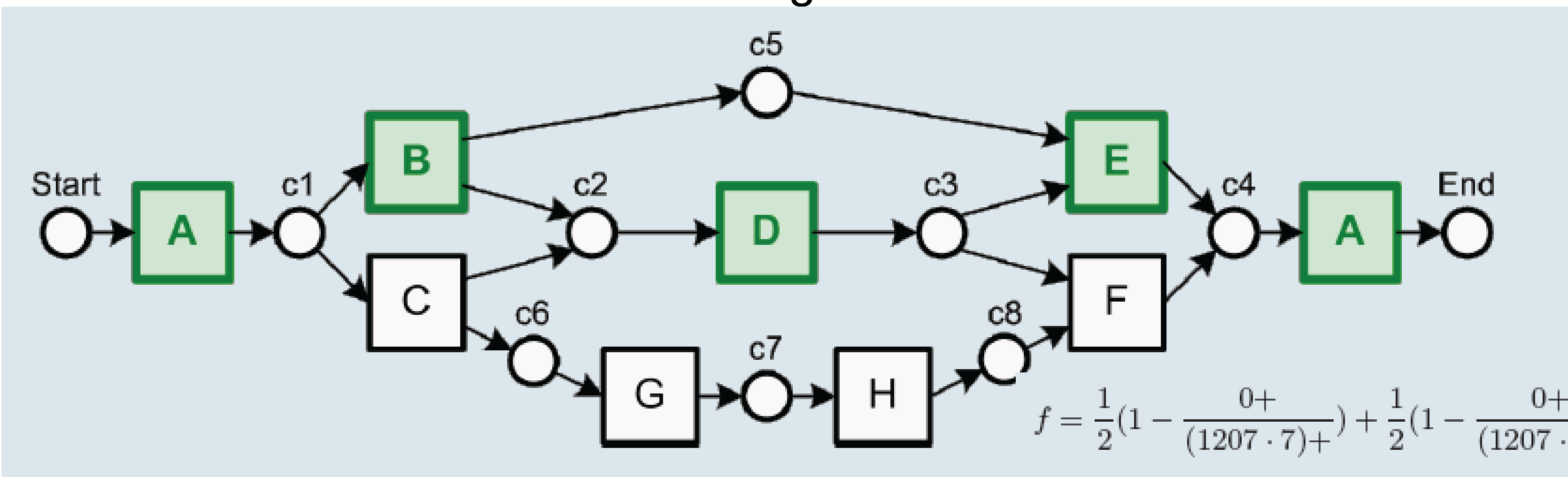
remaining tokens $= 0$ produced tokens $= 7$



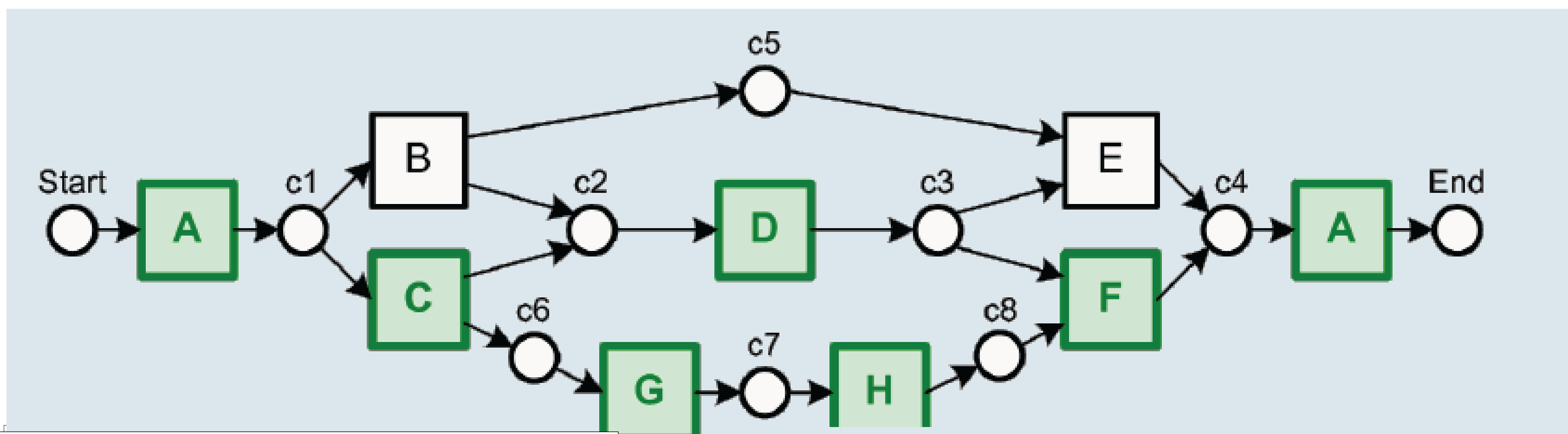
Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

5 Conformance Checking

Exercise: Calculate f for the following traces:



$$f = \frac{1}{2}(1 - \frac{0+}{(1207 \cdot 7)+}) + \frac{1}{2}(1 - \frac{0+}{(1207 \cdot 7)+})$$



$$f = \frac{1}{2}(1 - \frac{0+0+}{(1207 \cdot 7) + (145 \cdot 9)+}) + \frac{1}{2}(1 - \frac{0+0+}{(1207 \cdot 7) + (145 \cdot 9)+})$$

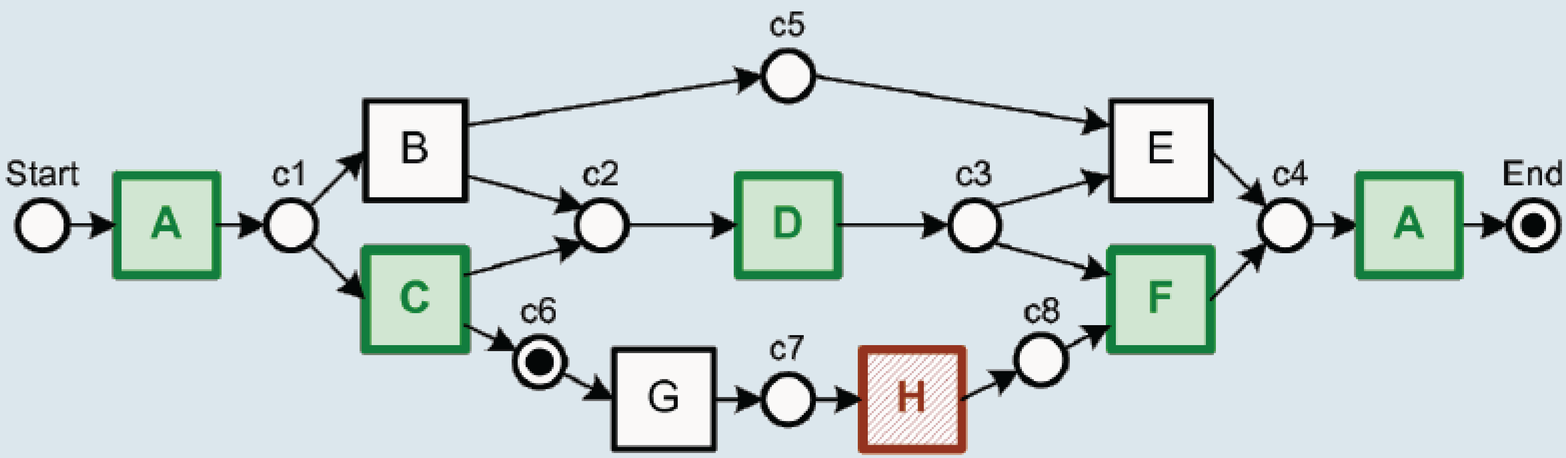
Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

5 Conformance Checking

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

$$f = \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}\right) + \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}\right)$$

$$\begin{array}{llll} \text{missing tokens} & = 1 & \text{consumed tokens} & = 7 \\ \text{remaining tokens} & = 0 & \text{produced tokens} & = 8 \end{array}$$



Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

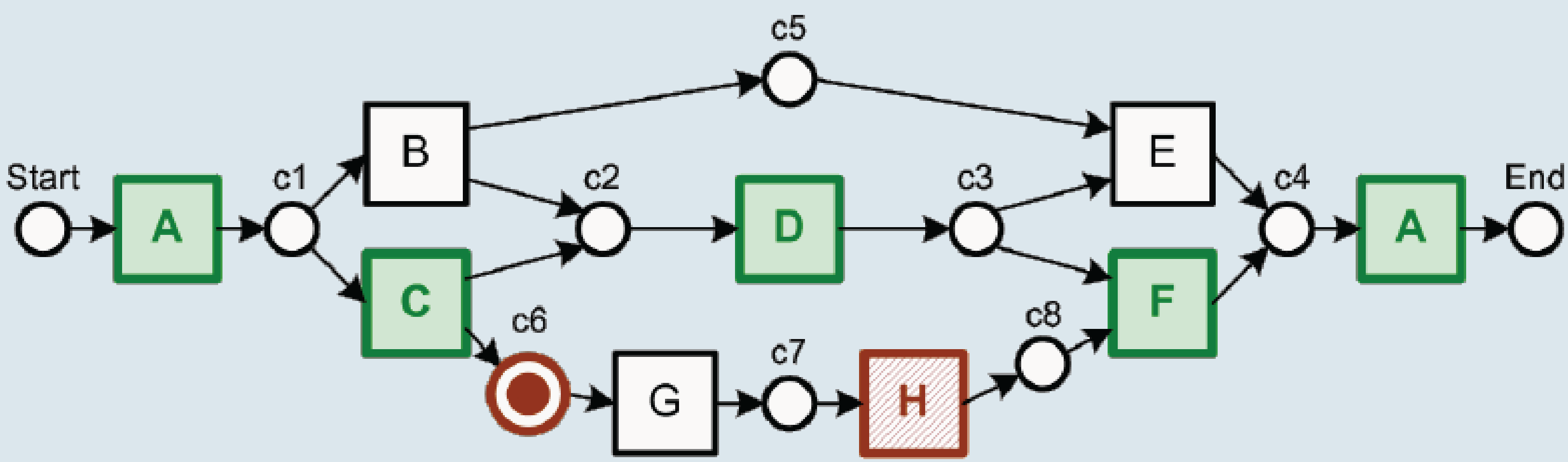
5 Conformance Checking

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

$$f = \frac{1}{2}\left(1 - \frac{0+0+0+(23\cdot 1)+}{(1207\cdot 7)+((145+56)\cdot 9)+(23\cdot 8)+}\right) + \frac{1}{2}\left(1 - \frac{0+0+0+(23\cdot 1)+}{(1207\cdot 7)+((145+56)\cdot 9)+(23\cdot 8)+}\right)$$

missing tokens = 1 consumed tokens = 8

remaining tokens = 1 produced tokens = 8



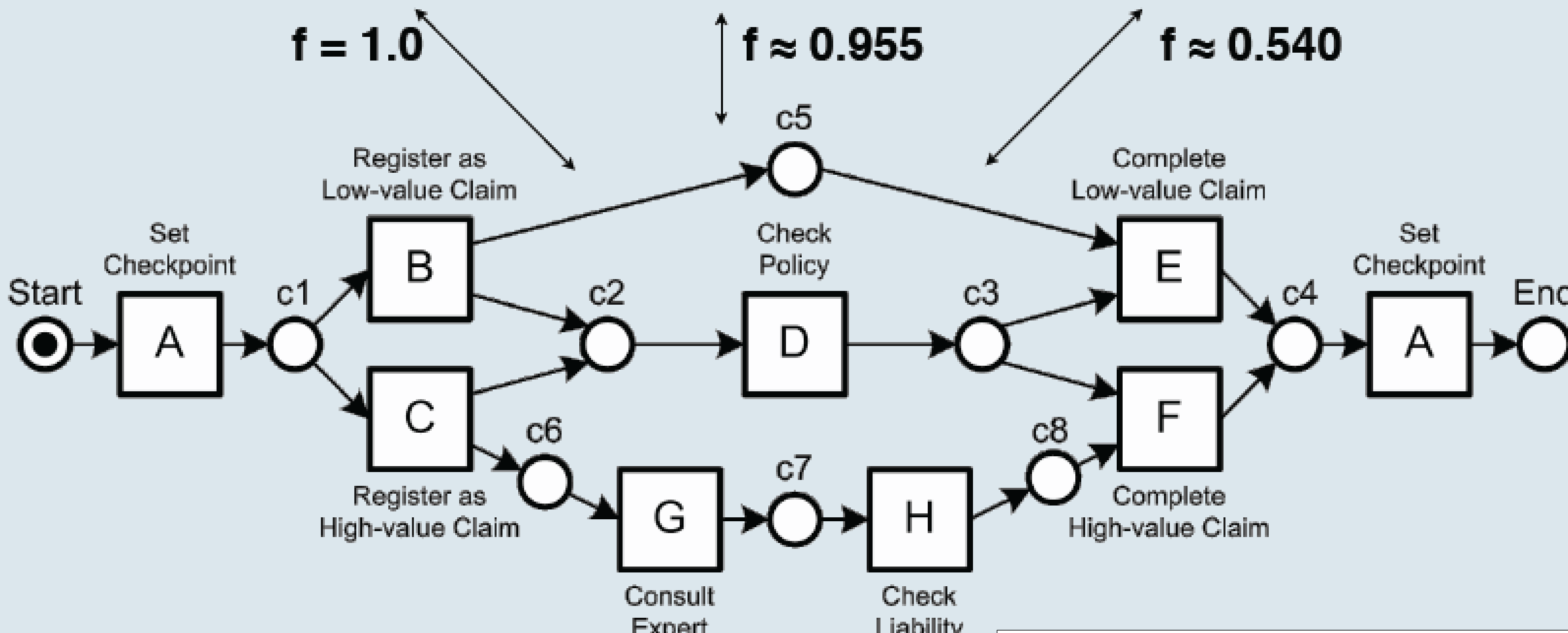
Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

5 Conformance Checking

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

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

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



Source: Anne Rozinat: Conformance Checking of Service Behavior. *BETA Conference, Eindhoven, 2007*

Contents

6 Summary

- So far only discovery of control flow aspects
- Development of approaches for discovery of other important aspects, for example:
  - *Organizational structures*:
    - Social Network Mining [AaSo04] (e.g., who is working with whom)?
    - Access rules / staff assignment rules [LRRD05] (e.g., step A is executed by an actor having role „physician“ and belonging to organizational unit „ward 1“)
  - **Decision Mining** [RoAa06]: Mining of transition conditions determining routing in alternative branchnings (e.g., age > 20) → combines process mining and data mining (decision trees9 → combined approaches

6 Summary

- Process disovery is one task of process mining
- A selection of further tasks is [Aalst11]:
  - *Enhance*: connect process models to event logs for, e.g., repair
  - Check: for auditing reasons, process models can be checked against log data
  - *Explore*: event data can be used on top of process models to explore the performance of processes during runtime
- Comparing event data and process models is based on → conformance checking

References

[AaSo0]	W.M.P. van der Aalst and M. Song: Mining Social Networks: Uncovering interaction patterns in business processes. International Conference on Business Process Management (BPM 2004), LNCS 3080, pages 244-260 (2004)
[ABD05]	W.M.P. van der Aalst, H.T. de Beer, and B.F. van Dongen. Process Mining and Verification of Properties: An Approach based on Temporal Logic. OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005, LNCS 3760, pages 130-147 (2005)
[ADHM03]	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)
[AWM04]	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) 16(9), pages 1128-1142 (2004)

References (ctd.)

[DoAA05]	B.F. van Dongen and W.M.P. van der Aalst: Multi-phase Process mining: Aggregating Instance Graphs into EPCs and Petri Nets. International Workshop on Applications of Petri Nets to Coordination, Worklflow and Business Process Management (PNCWB) at the ICATPN 2005 (2005)
[GuAa06]	C.W. Günther, W.M.P. van der Aalst: A Generic Import Framework for Process Event Logs. International Workshop Business Process Intelligence (BPI) in conjunction with BPM 2006, Vienna, pages 81 – 92 (2006)
[LRDR05]	T.L. Ly, S. Rinderle, P. Dadam, M. Reichert: Mining Staff Assignment Rules from Event-Based Data. Proc. Workshop on Business Process Intelligence (BPI), in conjunction with BPM 2005, Nancy, France (2005)
[RiAa07]	Rinderle-Ma, S. and van der Aalst, W.M.P.: Life-cycle support for staff assignment rules in process-aware information systems, Technical Report, TU Eindhoven (2007)

References (ctd.)

[MDAW04]	A.K.A de Medeiros, B.F. van Dongen, W.M.P. van der Aalst and A.J.M.M. Weijters: Process Mining: Extending the a-algorithm to Mine Short Loops. BETA Working Paper Series, WP 113, Eindhoven University of Technology, Eindhoven (2004)
[MWA05]	A.K. Alves de Medeiros, A.J.M.M.Weijters and W.M.P. van der Aalst: Genetic Process Mining: A Basic Approach and its Challenges . Workshop on Business Process Intelligence (BPI), Nancy (2005)
[RoAa05]	A. Rozinat and W.M.P. van der Aalst: Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. Workshop on Business Process Intelligence (BPI), Nancy (2005)
[RoAa06]	A. Rozinat and W.M.P. van der Aalst: Decision Mining. International Conference on Business Process Management (BPM 2006), Vienna, pages 420 – 425 (2006)

References

[AaSo0]	W.M.P. van der Aalst and M. Song: Mining Social Networks: Uncovering interaction patterns in business processes. International Conference on Business Process Management (BPM 2004), LNCS 3080, pages 244-260 (2004)
[ABD05]	W.M.P. van der Aalst, H.T. de Beer, and B.F. van Dongen. Process Mining and Verification of Properties: An Approach based on Temporal Logic. OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005, LNCS 3760, pages 130-147 (2005)
[ADHM03]	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)
[AWM04]	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) 16(9), pages 1128-1142 (2004)

References (ctd.)

[DoAA05]	B.F. van Dongen and W.M.P. van der Aalst: Multi-phase Process mining: Aggregating Instance Graphs into EPCs and Petri Nets. International Workshop on Applications of Petri Nets to Coordination, Worklflow and Business Process Management (PNCWB) at the ICATPN 2005 (2005)
[GuAa06]	C.W. Günther, W.M.P. van der Aalst: A Generic Import Framework for Process Event Logs. International Workshop Business Process Intelligence (BPI) in conjunction with BPM 2006, Vienna, pages 81 92 (2006)
[LRDR05]	T.L. Ly, S. Rinderle, P. Dadam, M. Reichert: Mining Staff Assignment Rules from Event-Based Data. Proc. Workshop on Business Process Intelligence (BPI), in conjunction with BPM 2005, Nancy, France (2005)
[Aalst11]	W.M.P. van der Aalst: Process Mining, Springer (2011)

References (ctd.)

[MDAW04]	A.K.A de Medeiros, B.F. van Dongen, W.M.P. van der Aalst and A.J.M.M. Weijters: Process Mining: Extending the a-algorithm to Mine Short Loops. BETA Working Paper Series, WP 113, Eindhoven University of Technology, Eindhoven (2004)
[MWA05]	A.K. Alves de Medeiros, A.J.M.M.Weijters and W.M.P. van der Aalst: Genetic Process Mining: A Basic Approach and its Challenges . Workshop on Business Process Intelligence (BPI), Nancy (2005)
[RoAa05]	A. Rozinat and W.M.P. van der Aalst: Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. Workshop on Business Process Intelligence (BPI), Nancy (2005)
[RoAa06]	A. Rozinat and W.M.P. van der Aalst: Decision Mining. International Conference on Business Process Management (BPM 2006), Vienna, pages 420 425 (2006)