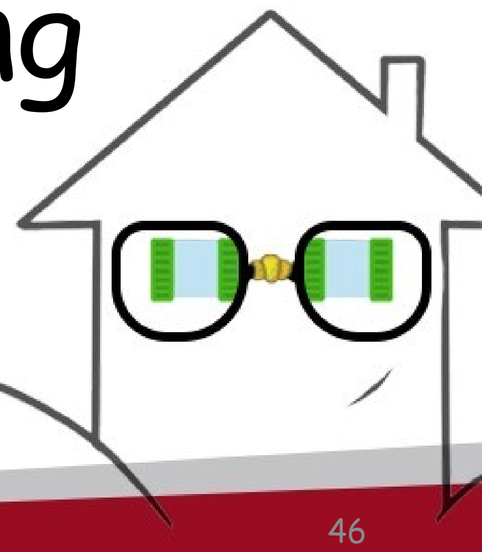
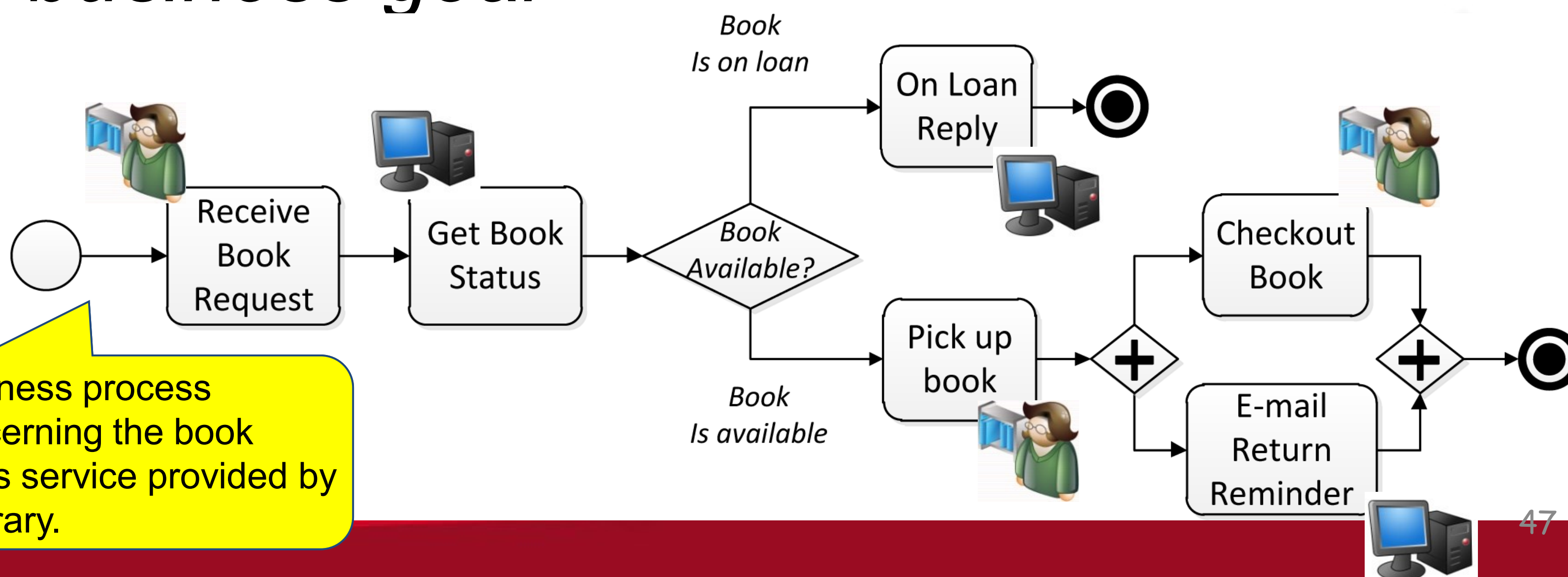
SAPIENZA
UNIVERSITÀ DI ROMA

Business Process Management and Process Mining



Business Processes

A business process consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal



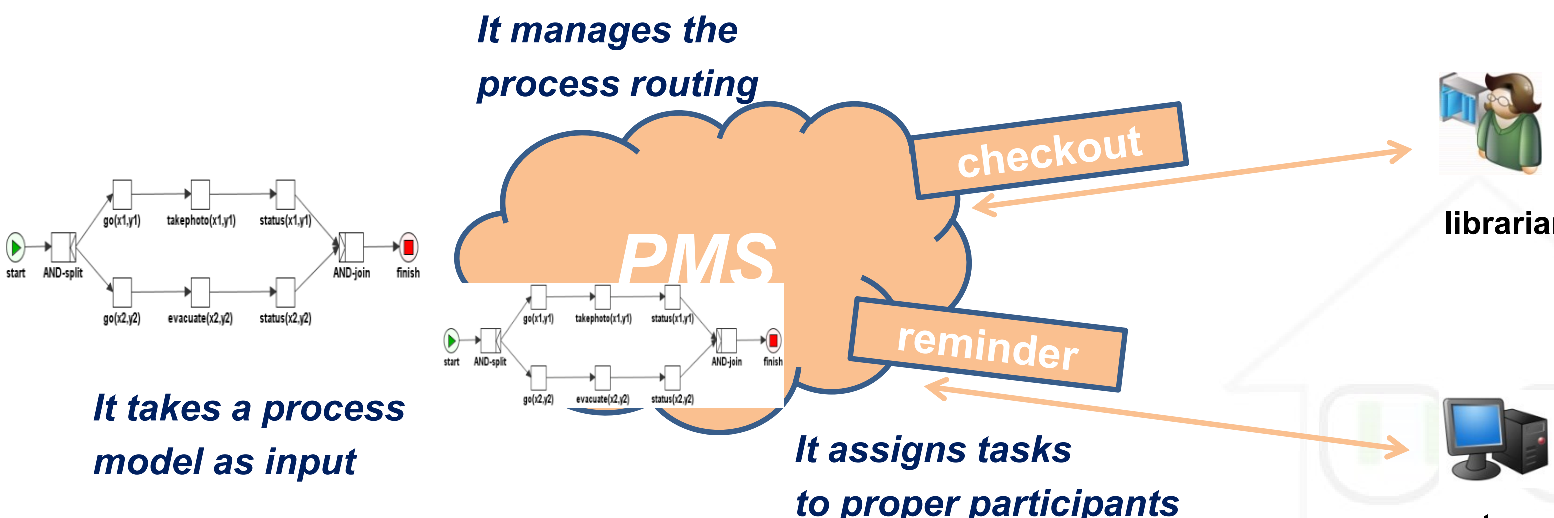
Business process concerning the book loans service provided by a library.

Business Processes and Information Systems

- Currently, business processes are the core of most information systems
  - production line of a car manufacturer
  - procedures for buying tickets on-line
- This requires that organizations specify their **flow of work** (their business processes) for the **orchestration** of participants, information and technology for the realization of products and services
- An information system that supports a business process is called ***Process Management System*** (or ***Process Aware Information System***)

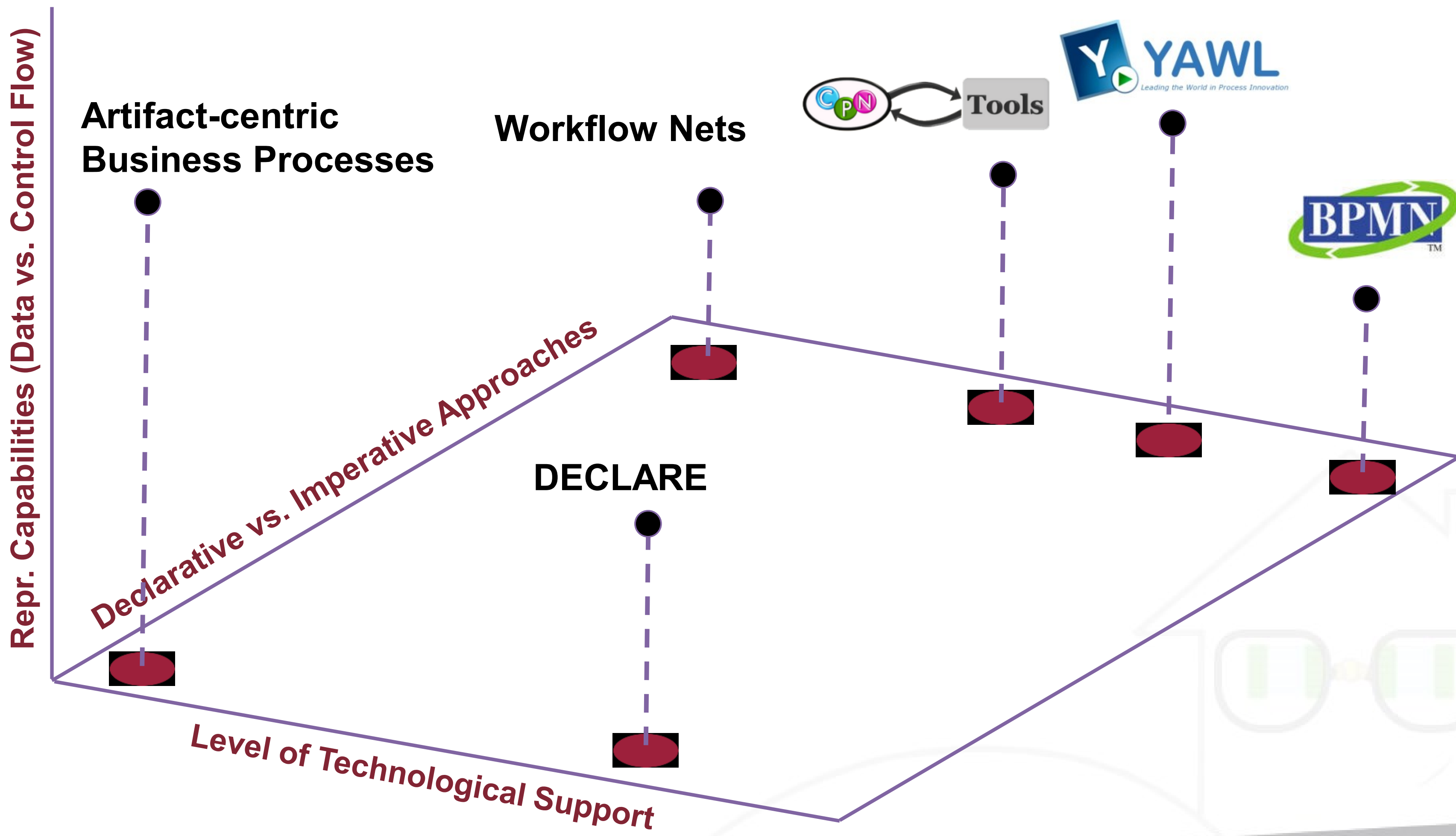
Process Management Systems

A **Process Management System (PMS)** *is a generic software system that is driven by explicit process representations to coordinate the enactment of business processes*



– A PMS is driven by a specific business process model…

Modeling Languages for Business Processes



Classifying Business P

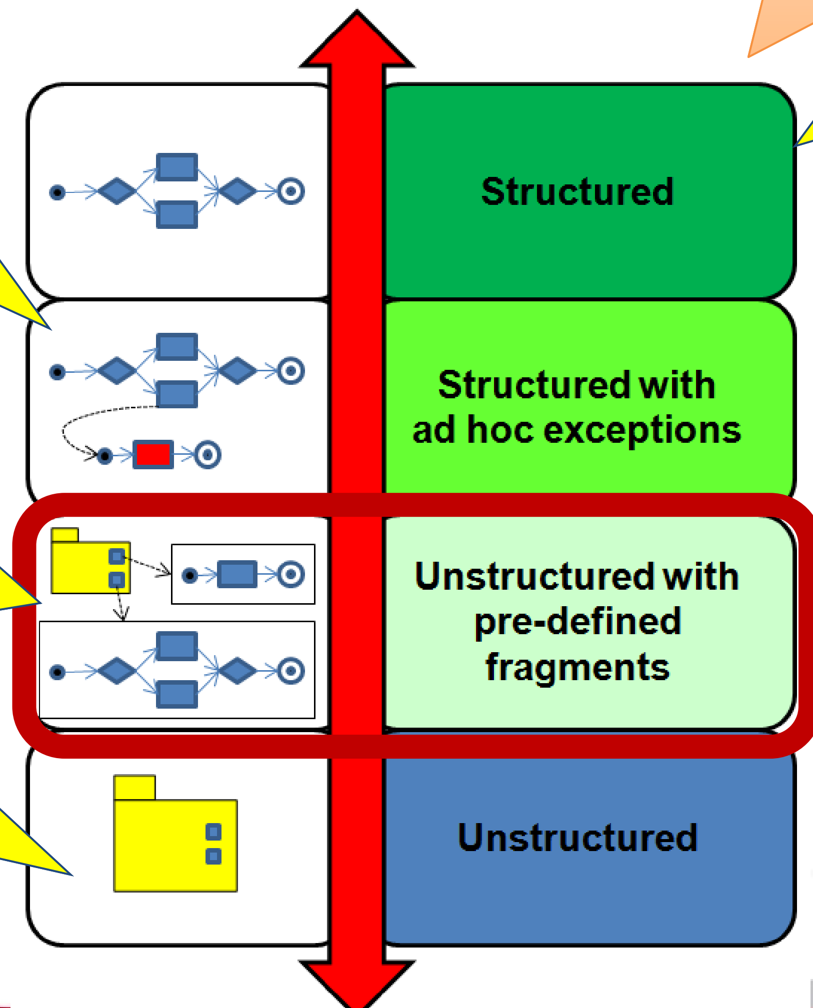
S. Kemsley. *The changing nature of work: From structured to unstructured, from controlled to social*. BPM, 2011

These processes are **completely predictable** and all possible paths are **well-understood.**

Such processes require to be **adapted** according to **changing circumstances during the execution.**

Class of processes where **process modeling could not be completed before the execution.**

It is **impossible to define a priori the exact steps to be taken** in order to complete an assignement



Process mining .. a bit of history

- The term "Process Mining" emerged in the 1998 in the software engineering field with Cook and Wolf, specifically in the work: "*Discovering models of software processes from event-based data*".
- Applying process mining to workflows has been proposed for the first time in the work of Agrawal and Leymann: "*Mining Process Models from Workflow Logs*" (1998).
- However, its roots date back about half a century….
  - For example, in 1958, Anil Nerode presented an approach to synthesize finite-state machines from example traces, in the research work: "*Linear Automaton Transformations*".
- The first survey of process mining was published in 2003 by van der Aalst et al.
  - After that, the progresses of process mining have been spectacular….

Great Idea in ICT?

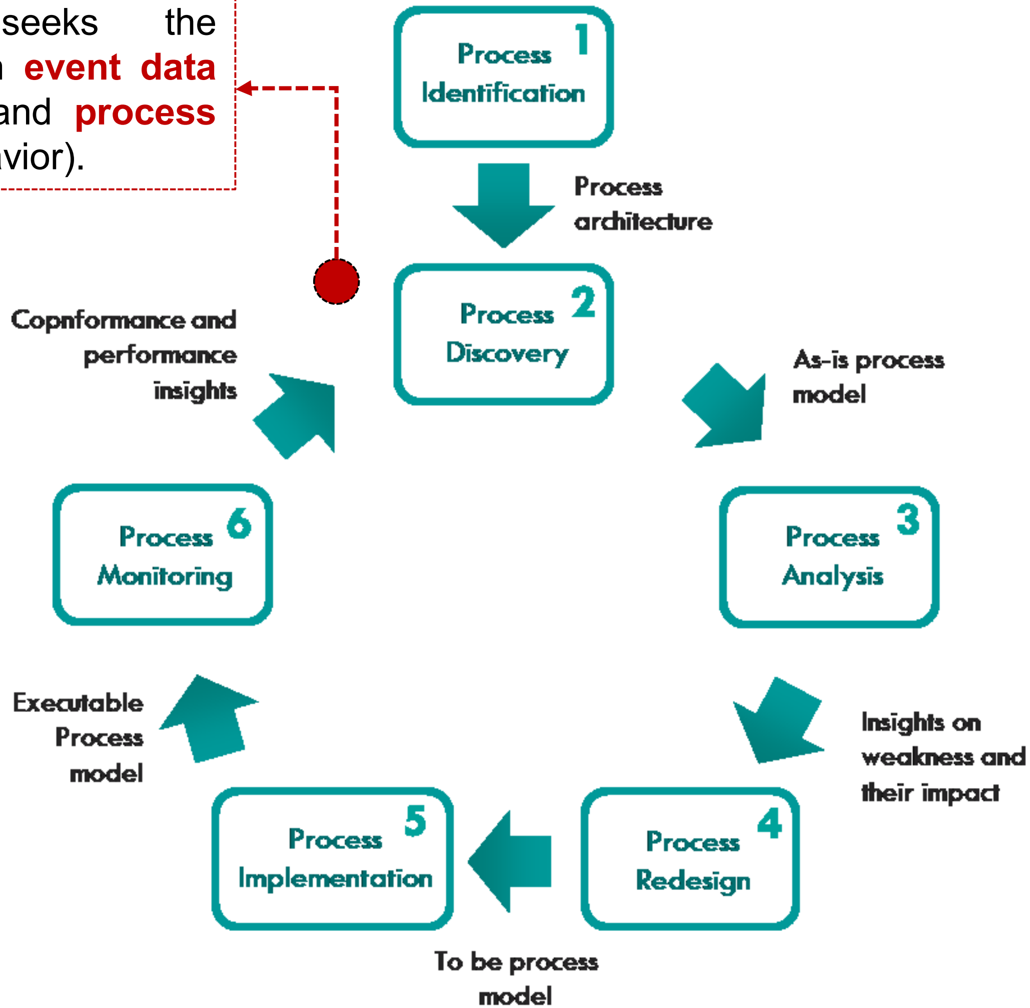
- Over the last decade, event data started to become **readily available** and process mining algorithms have been implemented in various academic and commercial systems.
- Today, there is an active group of researchers working on process mining, and it has become one of the **hot topics** in ICT research.
  - **ICPM 2019** - 1st International Conference on Process Mining
- Moreover, there is a rapidly growing **interest from industry** in process mining. More and more software vendors started adding process mining functionality to their tools.

Great Idea?

W.M.P. van der Aalst received in 2017 an Alexander von Humboldt Professorship, the highest German award for academics, with a value of **five million euros** for opening a research center in data science and process mining!

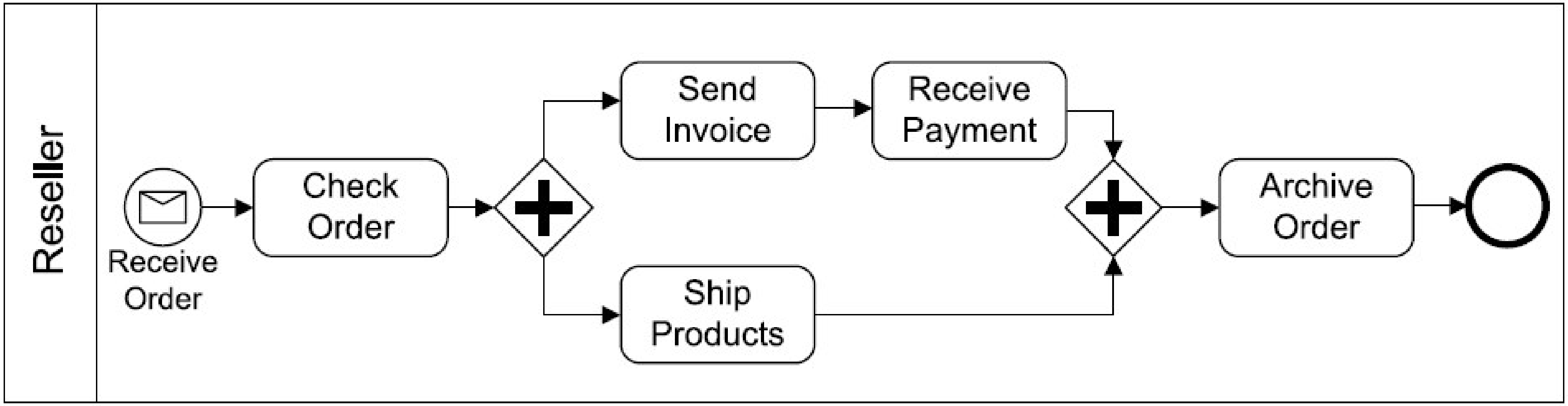
Process Mining in the BPM life cycle

Process Mining seeks the **confrontation** between **event data** (observed behaviour) and **process models** (expected behavior).

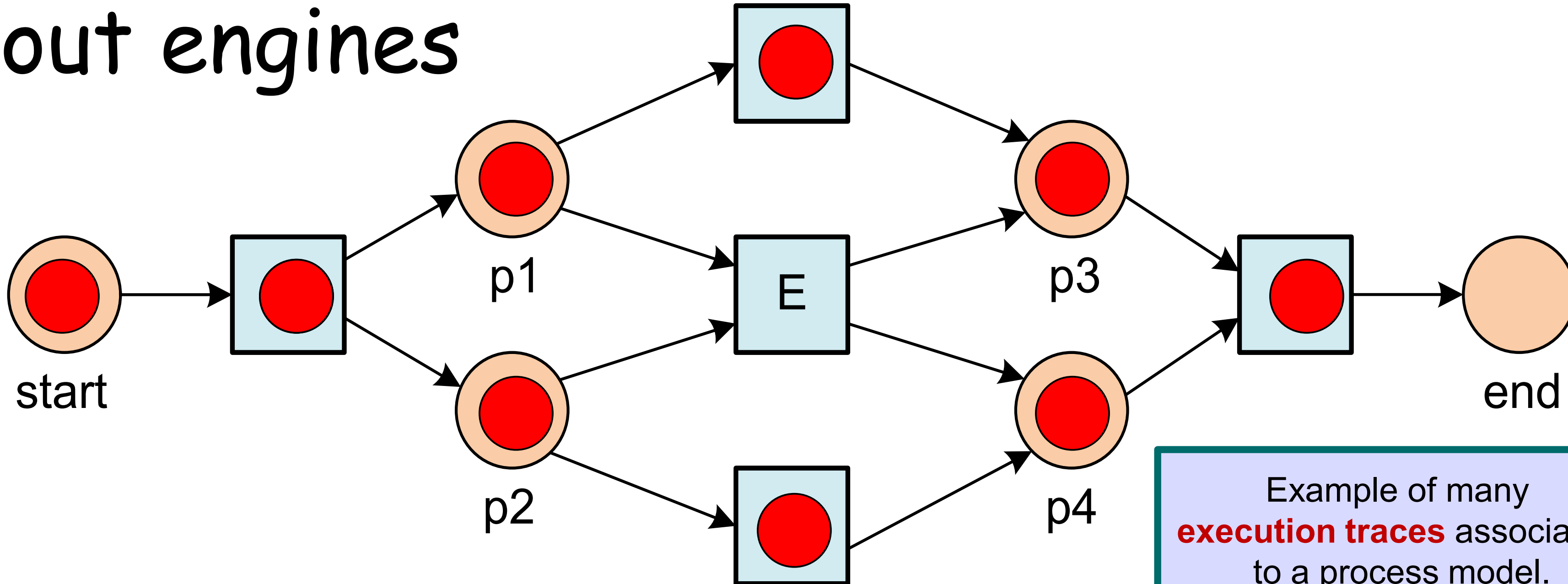


Process Models

- A **process model** is a **process representation** that consists of a set of activities and execution constraints between them, criteria to indicate the start and termination of the process, and information about participants, associated IT applications and data, etc.
- Process models focus on the **process structure** rather than on technical aspects of their realization.



PMSs as play-out engines



Example of many **execution traces** associated to a process model.

A B C D

A E D

A E D

A B C D

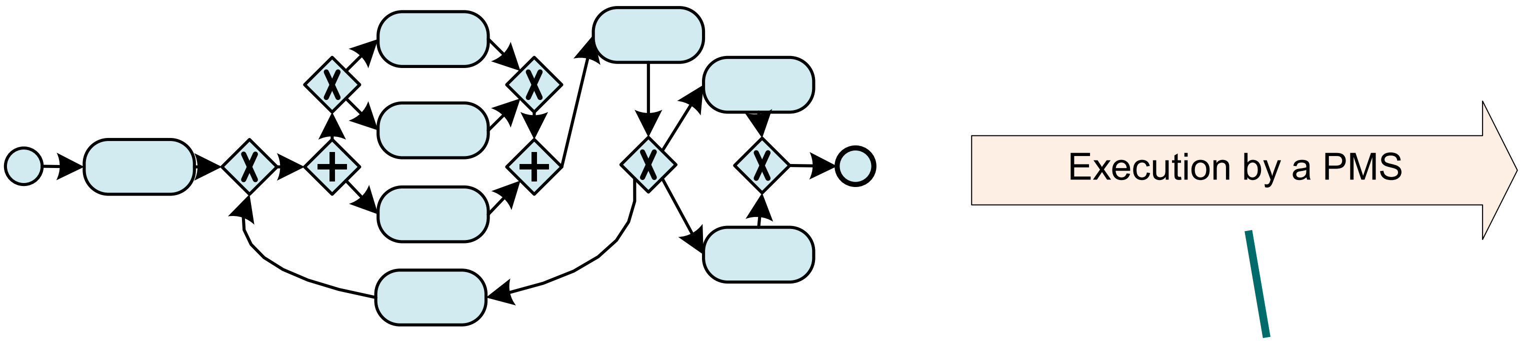
A C B D

A C B D

A E D

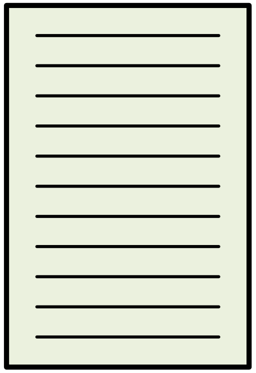
A C B D

Generation of Event Logs



process model

Any execution of a process model produces a new **execution trace** (i.e., a **process instance**) recorded in an **event log**.



event log

Goals of process mining

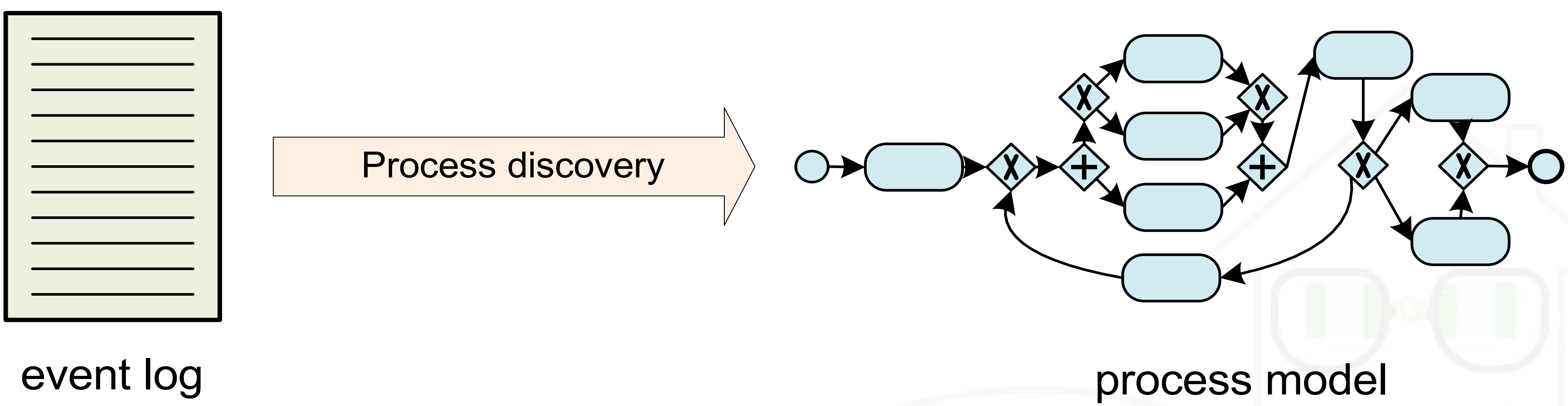
- Process Mining seeks the **confrontation** between **event logs** (i.e., observed behaviour) and **process models** (expected behavior).

- Process mining aims at answering the following questions:
  - What really happened in the past?
  - Why did it happen?
  - What is likely to happen in the future?
  - When and why do organizations and people deviate?
  - How to control a process better?
  - How to redesign a process to improve its performance?

- Two strategies to relate models and logs: **Play-In** and **Replay**.

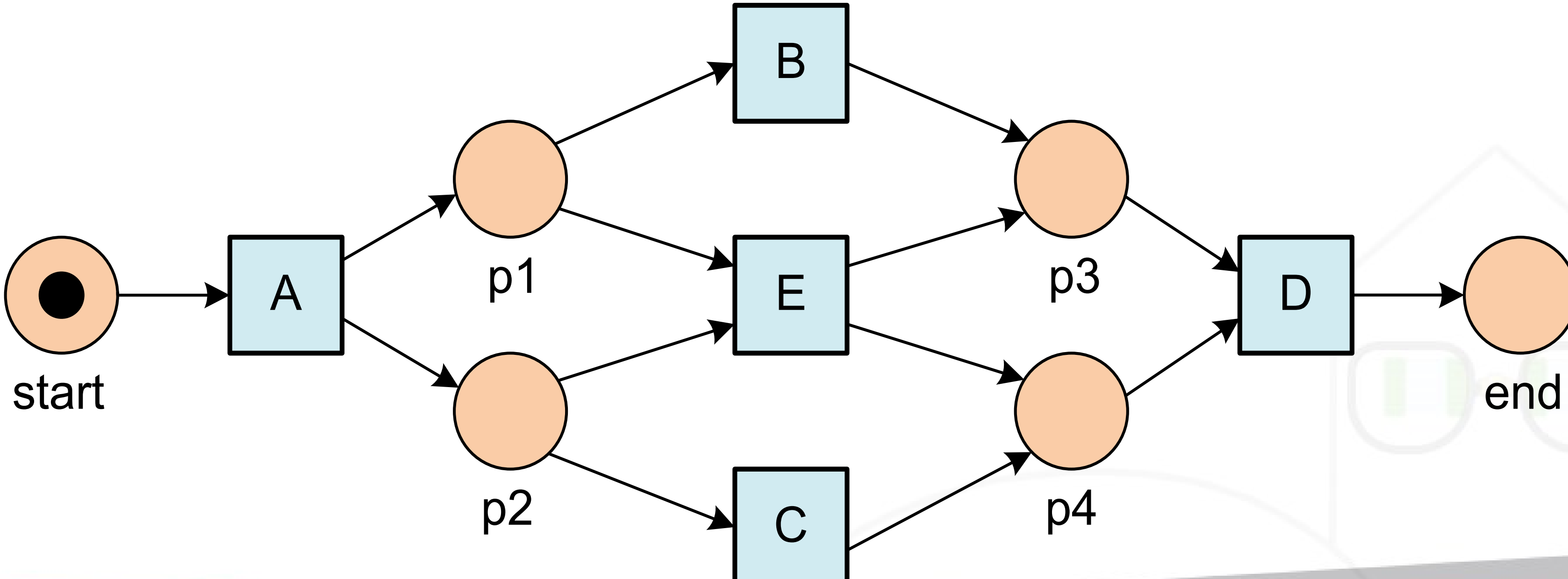
Play-In

- **Play-In** is the opposite of Play-Out: several execution traces are taken in input and the goal is to construct a process model.
- In the context of process mining, Play-In techniques are often referred to as **process discovery**.



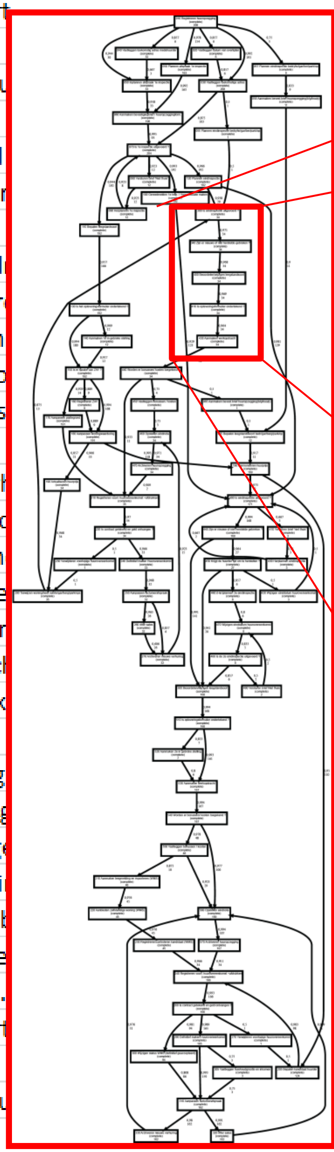
Play-In

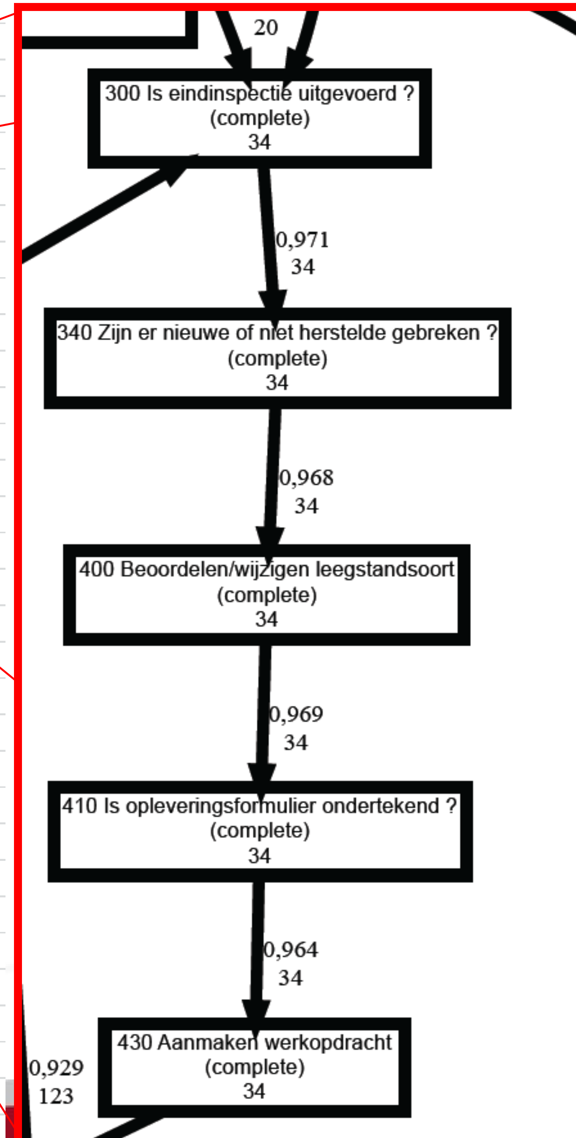
A B C D
A E D
A E D
A B C D
A C B D
A C B D
A E D
A C B D



Example of Process Discovery

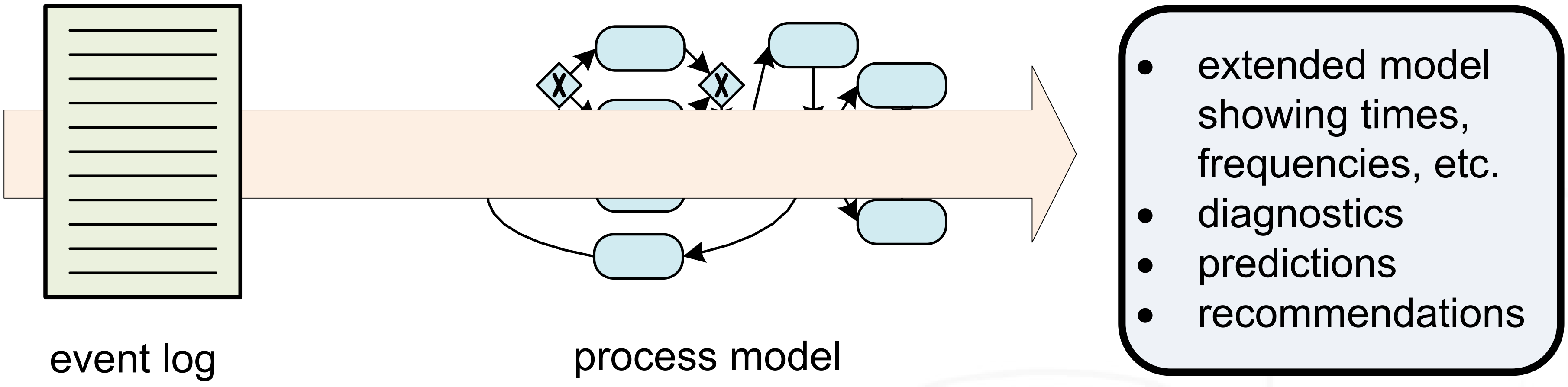
117315	110 Bepalen leegstandsoort	16.05.2007 14:06:23
117315	120 Plannen eindinspectie	16.05.2007 14:36:01
117315	130 Is het opleveringsformu	23.05.2007 09:41:40
117315	150 Is er sprake van ZAV ?	23.05.2007 09:41:51
117315	170 Aanpassen plattegrond	23.05.2007 11:57:18
117315	180 Aanpassen woningwaar	23.05.2007 09:42:37
117315	190 Actualiseren huurprijs	23.05.2007 09:48:23
117315	200 Toewijzen woning/bed	23.05.2007 09:48:29
117315	210 Registreren voorl. huur	10.09.2007 16:24:36
117315	220 Is contract getekend en	11.09.2007 14:56:18
117315	240 Definitief maken Huurc	31.03.2008 16:17:12
117315	250 Aanpassen factureerafs	09.09.2008 15:39:59
117315	260 After sales	09.09.2008 16:51:24
117315	270 Archiveren nieuwe verh	10.09.2008 07:52:08
117315	300 Is eindinspectie uitgevo	07.06.2007 14:47:04
117315	340 Zijn er nieuwe of niet h	07.06.2007 14:47:06
117315	400 Beoordelen/wijzigen le	07.06.2007 14:51:16
117315	410 Is opleveringsformulier	07.06.2007 14:51:26
117315	430 Aanmaken werkopdrac	11.06.2007 09:21:39
117315	440 Worden er bonussen/ k	11.06.2007 09:21:49
117315	460 Opstellen eindnota	08.08.2007 16:18:26
117315	470 Archiveren huuropzegg	09.08.2007 14:42:23
119763	010 Registreren huuropzegg	09.05.2007 11:19:14
119763	030 Vastleggen toekomstige	09.05.2007 12:25:01
119763	050 Inplannen afspraak 1e i	09.05.2007 11:59:52
119763	060 Aanmaken bevestiging	09.05.2007 12:31:57
119763	070 Is 1e inspectie uitgevoe	16.05.2007 13:04:26
119763	100 Gereedmelden 1e insp.	16.05.2007 13:43:39
119763	110 Bepalen leegstandsoort	16.05.2007 13:43:28
119763	120 Plannen eindinspectie	16.05.2007 13:42:58
119763	130 Is het opleveringsformu	16.05.2007 13:34:49
119763	150 Is er sprake van ZAV ?	16.05.2007 13:34:56





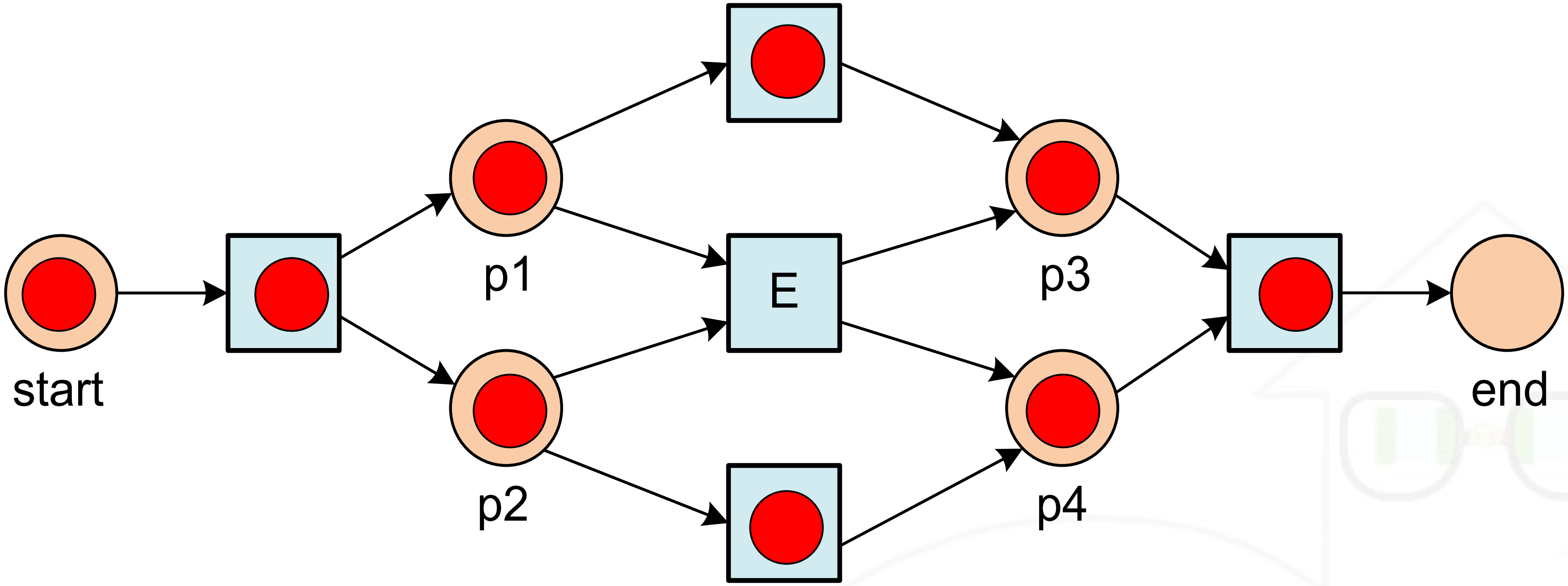
Replay

- **Replay** uses an event log and a process model as input. The event log is *replayed* on top of the process model.
- In this way, *discrepancies* between the log and the model can be *detected* and *quantified* (**conformance checking**).



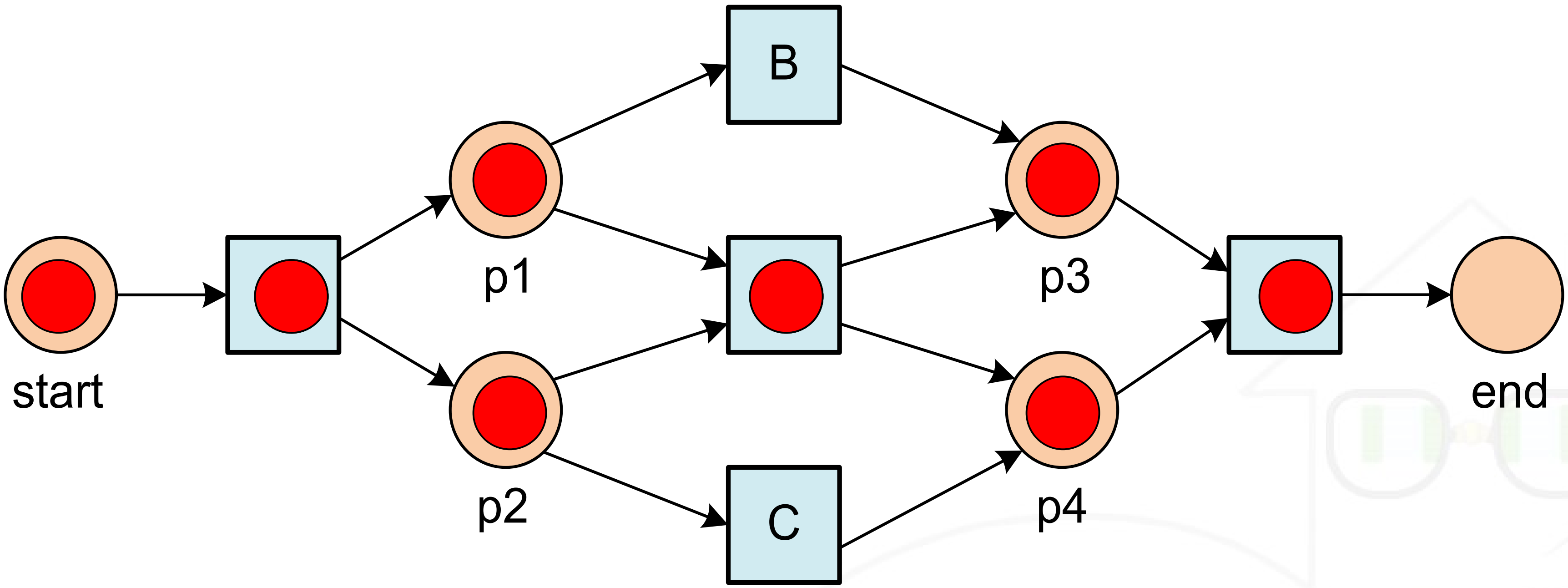
Replay

A B C D



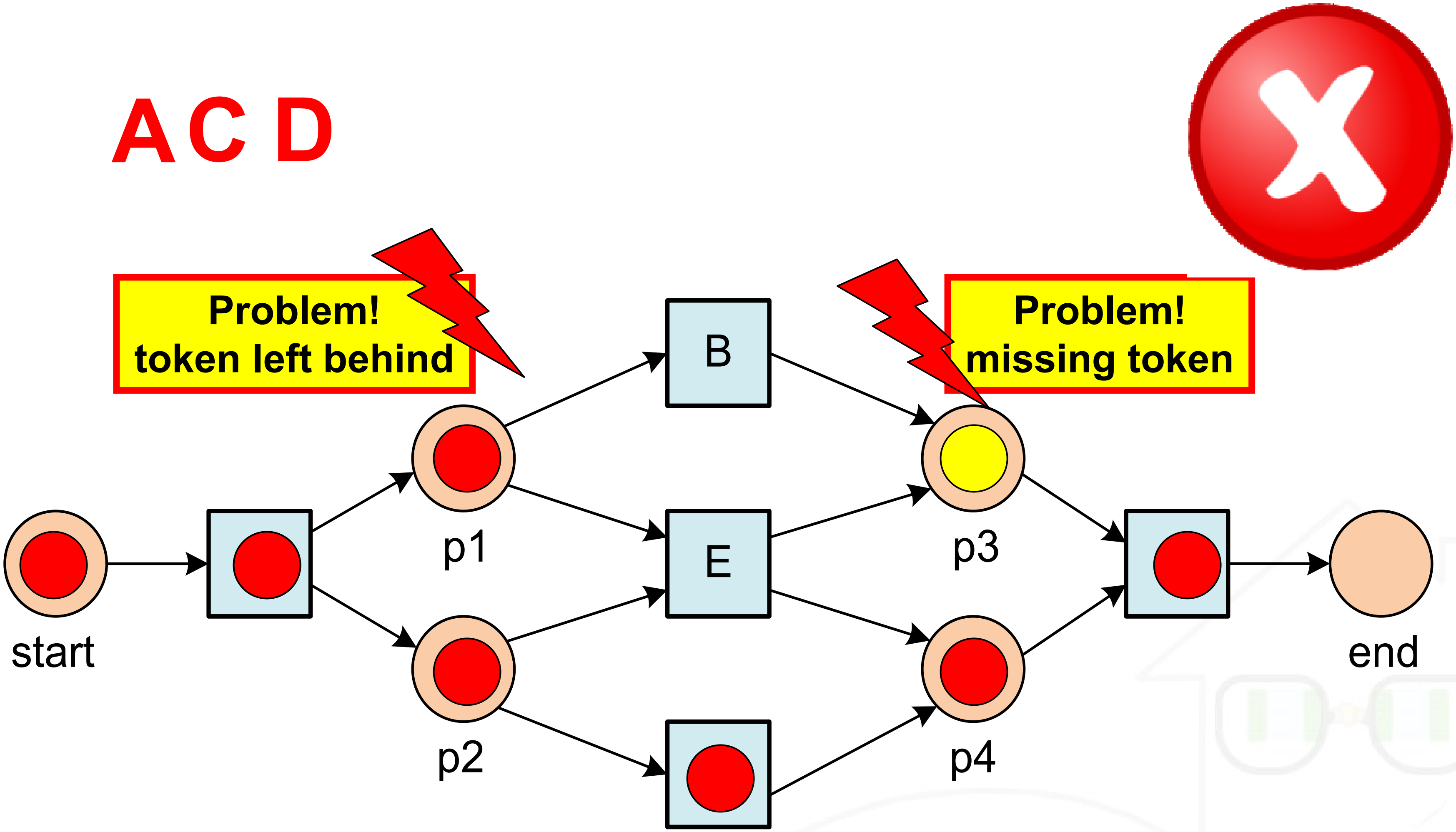
Replay

A E D

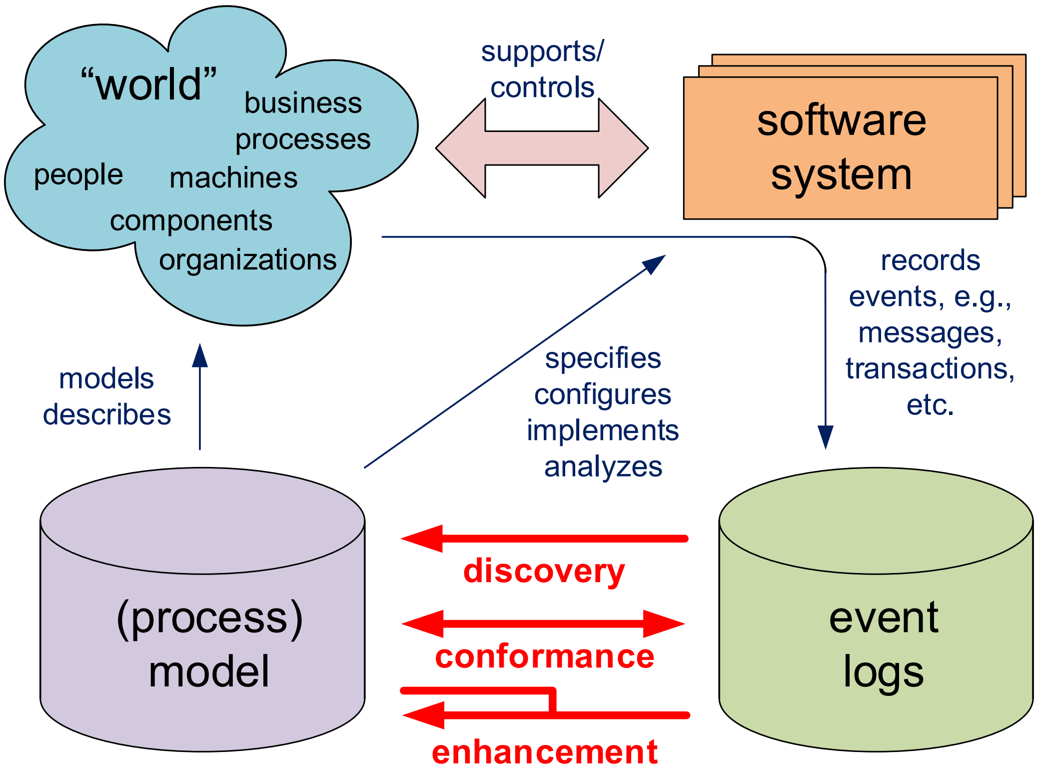


Replay

A C D


Problem!
token left behind
Problem!
missing token
B
E
start
p1
p2
p3
p4
end

Process mining techniques



- **Process discovery**
  - "What is really happening?"
- **Conformance checking:**
  - "Do we do what was agreed upon?"
- **Other techniques:**
  - **Performance analysis:**
    - "Where are the bottlenecks?"
  - **Process prediction:**
    - "Will this process instance be late?"
  - **Process enhancement:**
    - "How to redesign and refine this process?"

❖ **Process mining techniques have become mature over the years and are nowadays supported by various academic/commercial tools.**

Process Mining tools

- ProM
- Apromore
- Disco (Fluxicon)
- Perceptive Process Mining
- Celonis Discovery
- ARIS Process Performance Manager
- QPR ProcessAnalyzer
- Interstage Process Discovery (Fujitsu)
- Discovery Analyst (StereoLOGIC)
- XMAnalyzer (XMPro)
- …

ProM: Academic Process Mining Tool

- Download and install **ProM 6.9** from http://www.promtools.org/

600+ plug-ins available covering the whole process mining spectrum

(Rough) structure of an Event Log

- A single execution of a process is recorded into a **trace** (or a **case**).
- A trace consists of **events** such that each event relates to one trace.
- Events within a case are ordered (through timestamps) and can have **attributes**.
  - Examples of typical attribute names are activity, time, costs, and resource.
- **Minimal requirement**: ordered events referring to:
  - an activity name
  - a case id

case id	event id	properties				
		timestamp	activity	resource	cost	...
1	35654423	30-12-2010:11.02	register request	Pete	50	...
	35654424	31-12-2010:10.06	examine thoroughly	Sue	400	...
	35654425	05-01-2011:15.12	check ticket	Mike	100	...
	35654426	06-01-2011:11.18	decide	Sara	200	...
	35654427	07-01-2011:14.24	reject request	Pete	200	...
2	35654483	30-12-2010:11.32	register request	Mike	50	...
	35654485	30-12-2010:12.12	check ticket	Mike	100	...
	35654487	30-12-2010:14.16	examine casually	Pete	400	...
	35654488	05-01-2011:11.22	decide	Sara	200	...
	35654489	08-01-2011:12.05	pay compensation	Ellen	200	...
3	35654521	30-12-2010:14.32	register request	Pete	50	...
	35654522	30-12-2010:15.06	examine casually	Mike	400	...
	35654524	30-12-2010:16.34	check ticket	Ellen	100	...
	35654525	06-01-2011:09.18	decide	Sara	200	...
	35654526	06-01-2011:12.18	reinitiate request	Sara	200	...
	35654527	06-01-2011:13.06	examine thoroughly	Sean	400	...
	35654530	08-01-2011:11.43	check ticket	Pete	100	...
	35654531	09-01-2011:09.55	decide	Sara	200	...
	35654533	15-01-2011:10.45	pay compensation	Ellen	200	...
4	35654641	06-01-2011:15.02	register request	Pete	50	...
	35654643	07-01-2011:12.06	check ticket	Mike	100	...
	35654644	08-01-2011:14.43	examine thoroughly	Sean	400	...
	35654645	09-01-2011:12.02	decide	Sara	200	...
	35654647	12-01-2011:15.44	reject request	Ellen	200	...

Event logs as multi-set of traces

- An event log can be seen as a multi-set of traces.

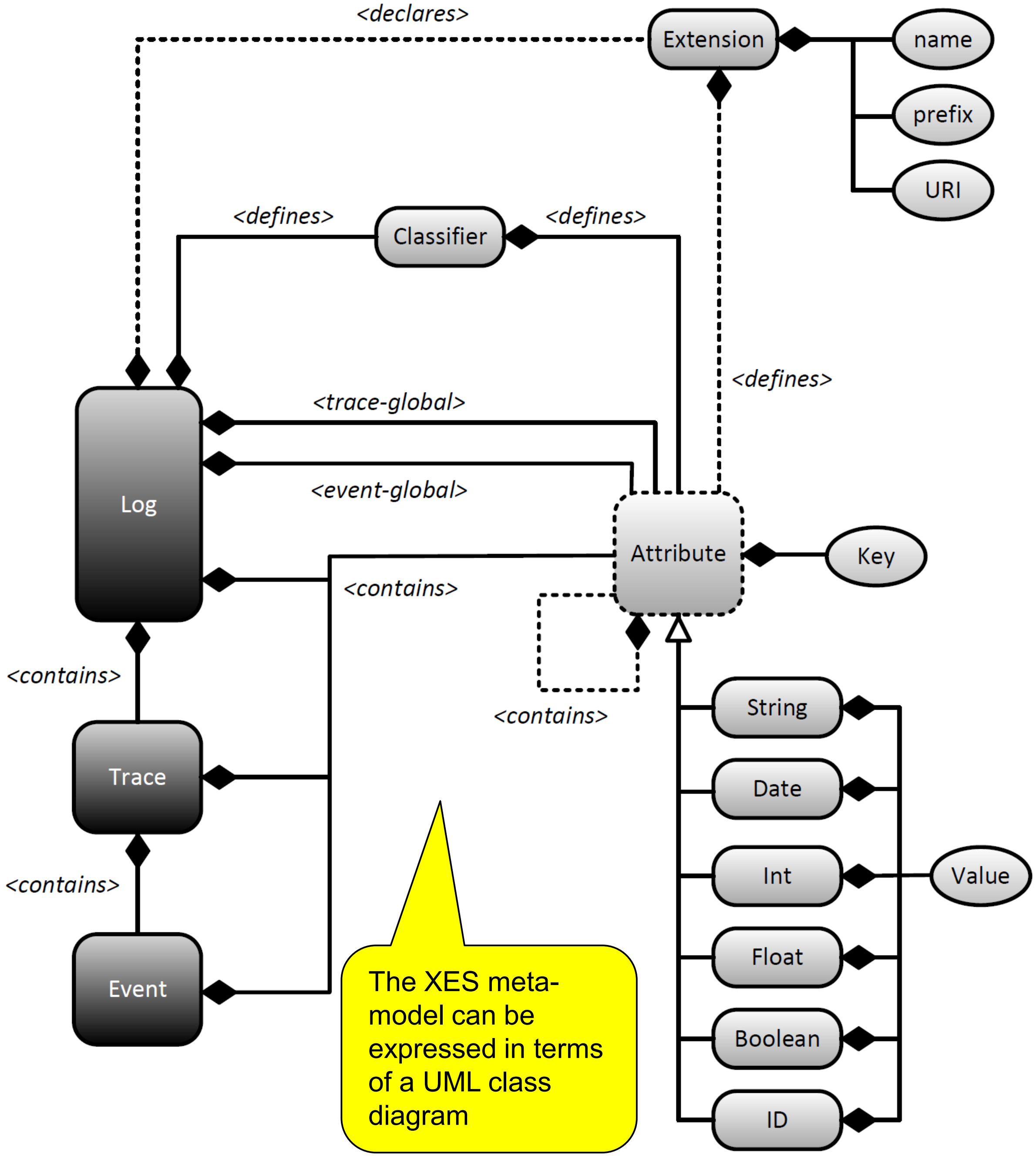
$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

  - Three traces <a,b,c,d>
  - Two traces <a,c,b,d>
  - One trace <e,e,d>

XES (eXtensible Event Stream)

- De-facto standard for storing, representing and exchanging event logs.
- See www.xes-standard.org.
- Adopted by the IEEE Task Force on Process Mining.
- Predecessor: MXML (2010).
- The format is supported by the majority of process mining tools.

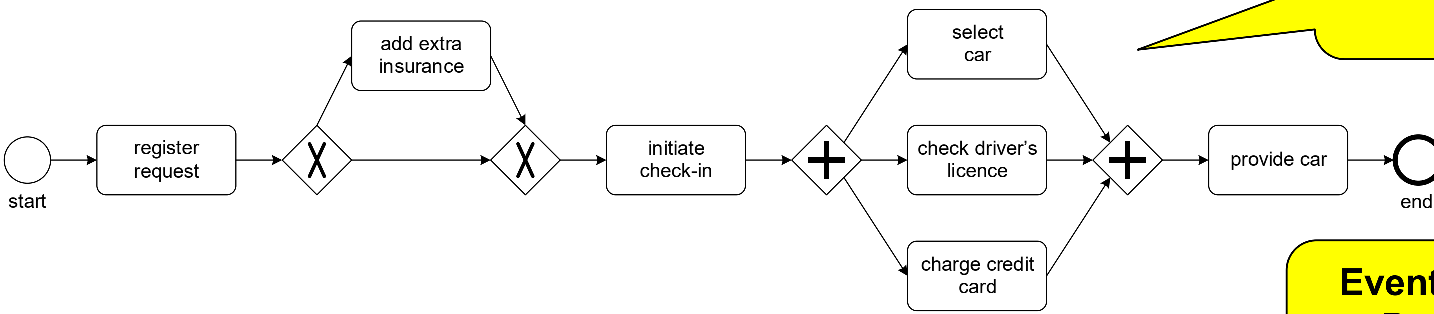
XES
Extensible Event Stream



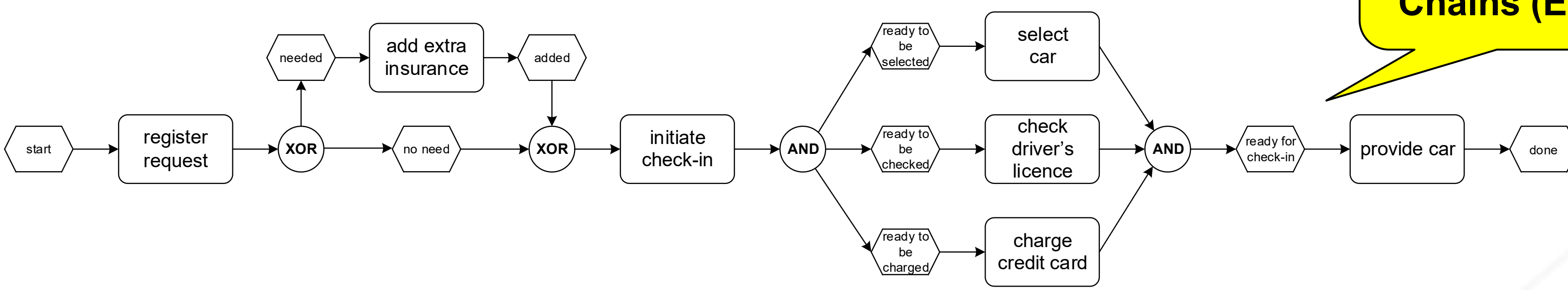
- A log contains traces and each trace contains events.
- Logs, traces, and events have attributes.
- Extensions may define new attributes and a log should declare the extensions used in it.
- Global attributes are attributes that are declared to be mandatory. Such attributes reside at the trace or event level.
- Attributes may be nested.
- Event classifiers are defined for the log and assign a “label” (e.g., activity name) to each event. There may be multiple classifiers

Different Modeling Notations

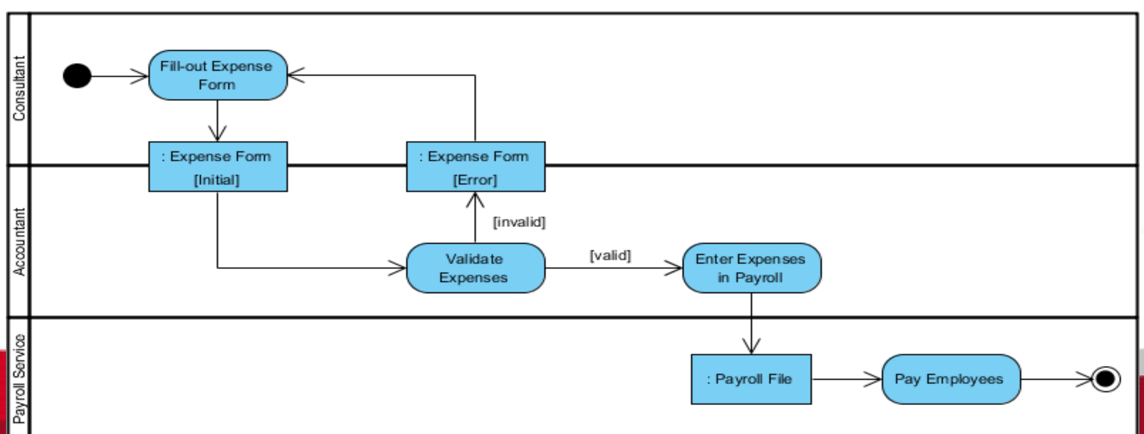
Business Process Modeling Notation (BPMN)



Event-Driven Process Chains (EPCs)



UML Activity Diagrams

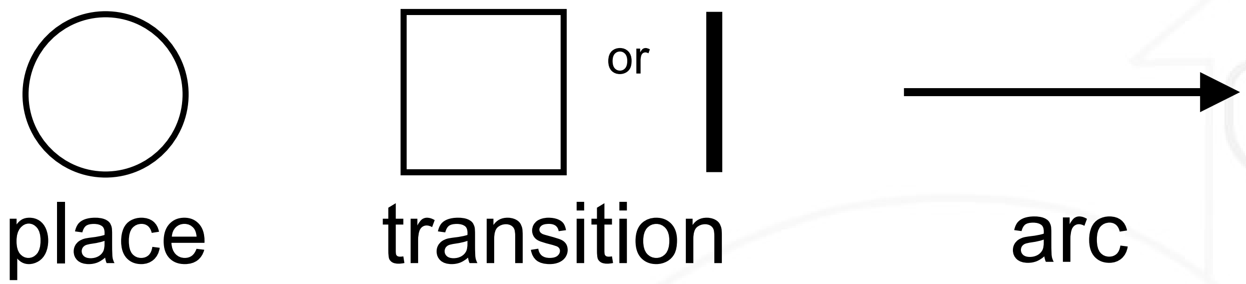


A formal notation for process modeling

- One of the frequent criticisms of modeling notations is that they are often **imprecise** and, as a consequence, they may be subject to **varying interpretations**.
- Describing both the **syntax** and **semantics** of a modeling notation in terms of a **formal well-founded technique** is an effective means of **minimizing the potential for ambiguity**.
- **Petri nets** is a formal technique that is proven to be suitable for modeling the *static* and *dynamic* aspects of business processes.
- Petri nets provide three specific advantages:
  - Formal semantics despite the graphical nature.
  - Modeling of concurrency.
  - Abundance of analysis techniques.
- **Process mining algorithms work with Petri net based models**

Petri nets

- A Petri net takes the form of a **directed bipartite graph** where the nodes are either *places* or *transitions*.
- **Places** represent **intermediate states** that may exist during the operation of a process.
  - Places are represented by circles.
- Places can be input/output of **transitions**. Transitions correspond to the **activities** or **events** of which the process is made up.
  - Transitions are represented by rectangles or thick bars.
- **Arcs** connect places and transitions in a way that <u>places can only be connected to transitions and vice-versa</u>.



Petri nets for modelling business processes

- In the research article:

  Van der Aalst, Wil MP. "The application of Petri nets to workflow management." Journal of circuits, systems, and computers 8.01 (1998): 21-66.

  it was proposed to explicitly use Petri nets for **business process modelling**.

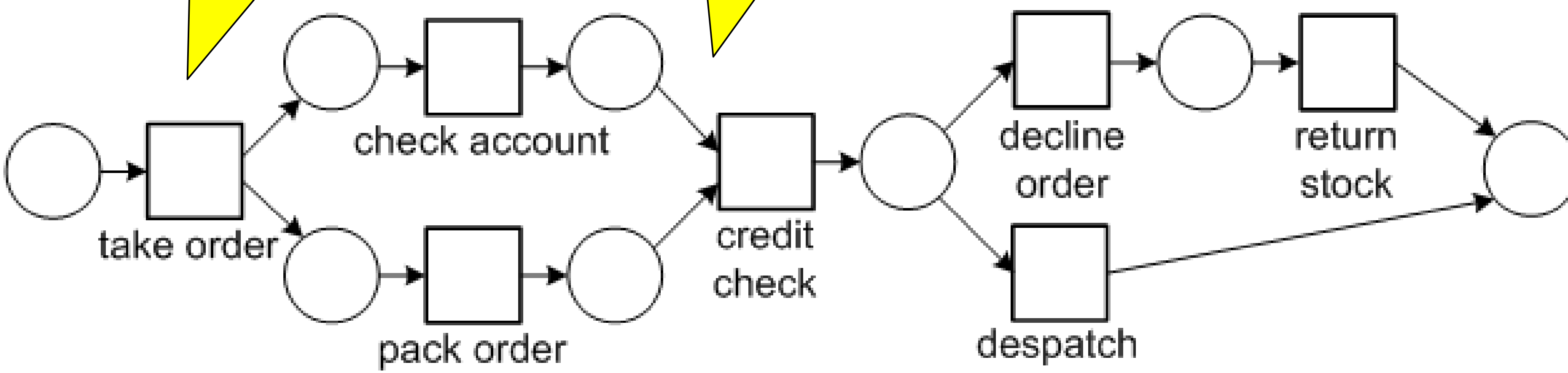
- **Intuition**: transitions represent the **activities** included in a business process and places represent the **conditions** preceding and following the activities.

Order Fulfillment Example

First, a ***take order*** task is executed.

Then, ***pack order*** and ***check account*** tasks are executed in parallel.

When ***pack order*** and ***check account*** tasks have been both completed, the ***credit check*** task is executed.



If the customer has sufficient credit remaining, the order is ***despatched***.

If the customer has not sufficient credit the ***decline order*** runs and, finally, the ***return stock*** task ensures that the items from the order are returned to the warehouse.

Process Discovery

- It is one of the **most challenging** process mining tasks.
- Based on an event log, a **process model is constructed** thus capturing the behavior seen in the log.

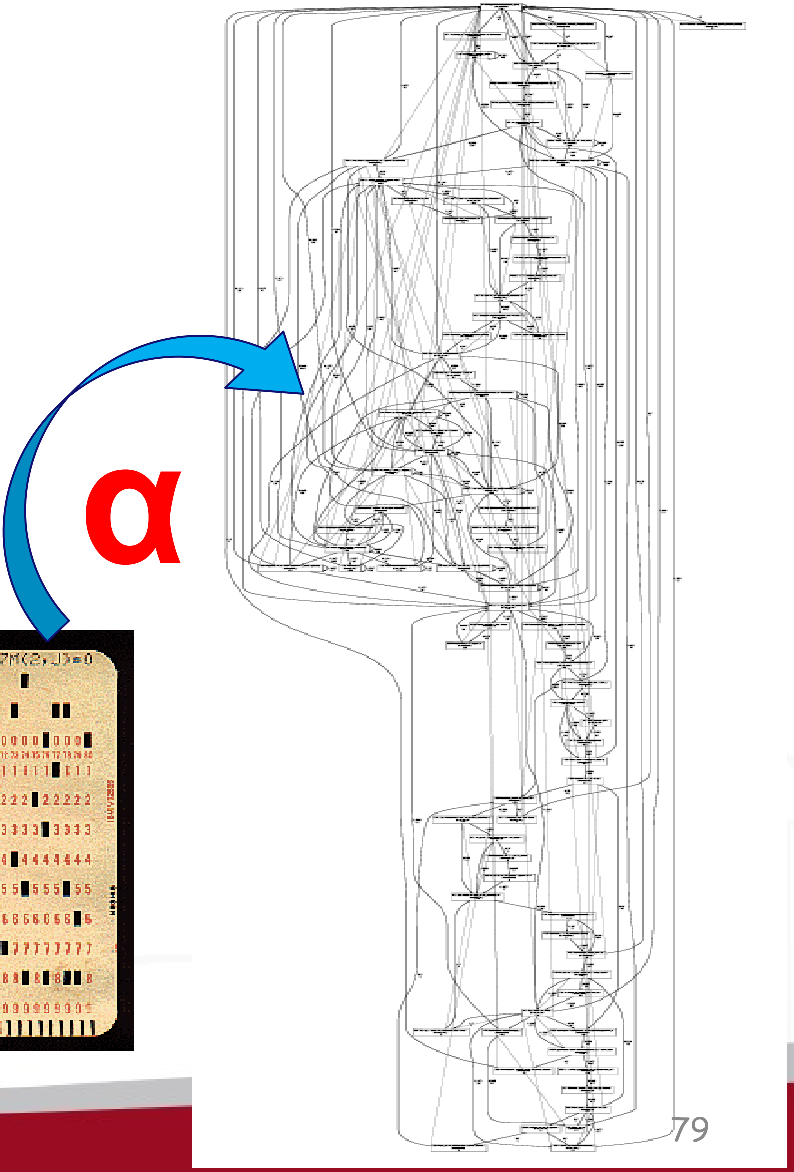
- **General Process Discovery Problem**
  - *Let L be a simple event log, i.e., a multi-set of traces over a set of activities. A process discovery algorithm is a function that maps L onto a process model, such that the model is "representative" for the behavior seen in L.*

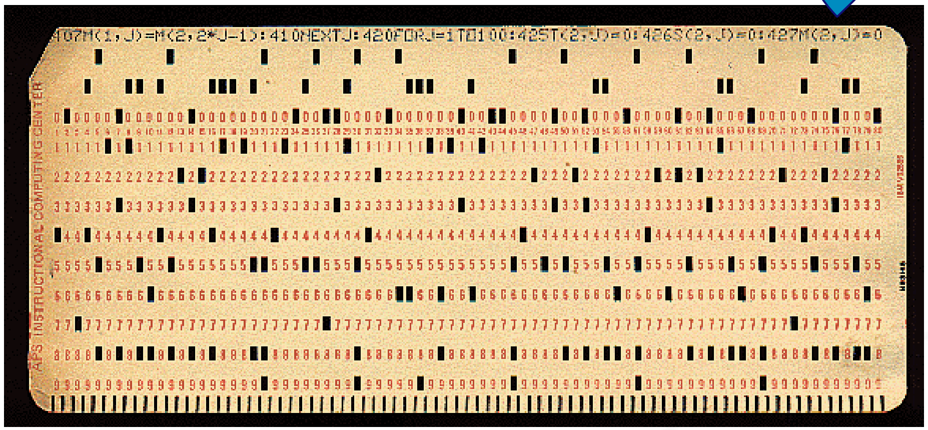
➢ The definition does not specify what kind of process model should be generated (e.g., BPMN, Petri Net, etc.).

➢ The concept of "representative" is unclear (we will discuss it later in detail).

Process Discovery: the α-algorithm

- The α-algorithm is one of the first process discovery algorithms that is able to deal with **concurrency**. It allows to discover **WF-Nets**.
- The α-algorithm is simple and many of its ideas have been used as baseline in other more robust techniques.





Basic Idea

- The α-algorithm scans the event log for particular patterns starting from some log-based ***ordering relations***.

- **Log-based ordering relations**: Let L be an event log over A (which is a set of activities). Let a, b ∈ A.
  - **Direct succession**: **a>b** if and only if - in some trace - a is directly followed by b.
  - **Causality**: **a→b** if and only if a>b and NOT b>a.
  - **Parallel**: **a||b** if and only if a>b and b>a
  - **Choice**: **a#b** iff NOT a>b and NOT b>a.

Identifying ordering relations

$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

Direct succession

a>b
a>c
a>e
b>c
b>d
c>b
c>d
e>d

Causality

a→b
a→c
a→e
b→d
c→d
e→d

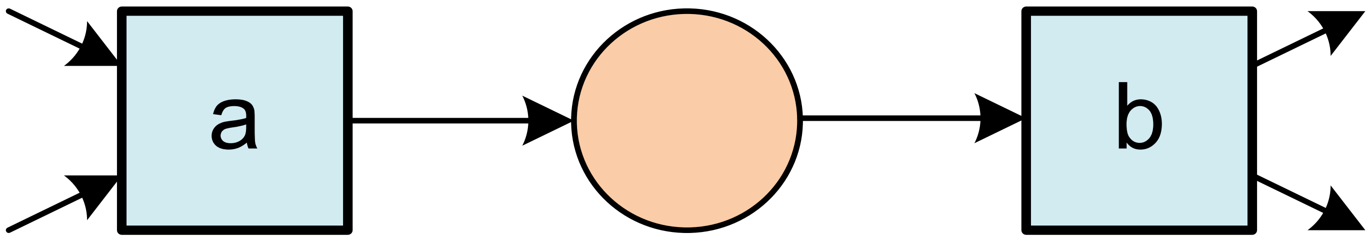
Parallel

b||c
c||b

Choice

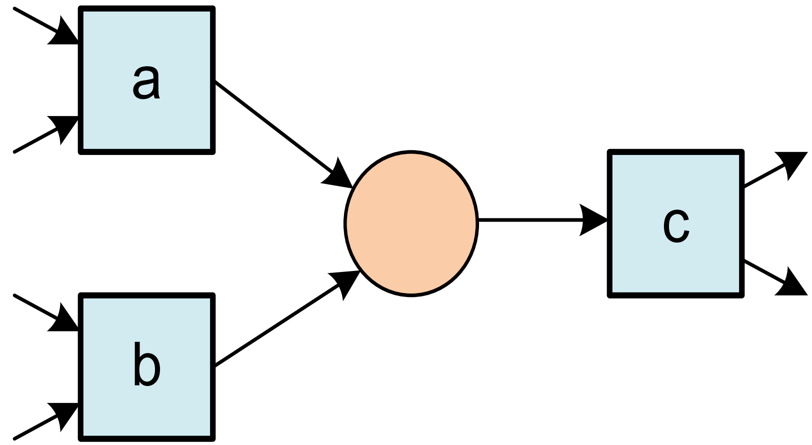
b#e
e#b
c#e
e#c
a#d
d#a

Discovery patterns

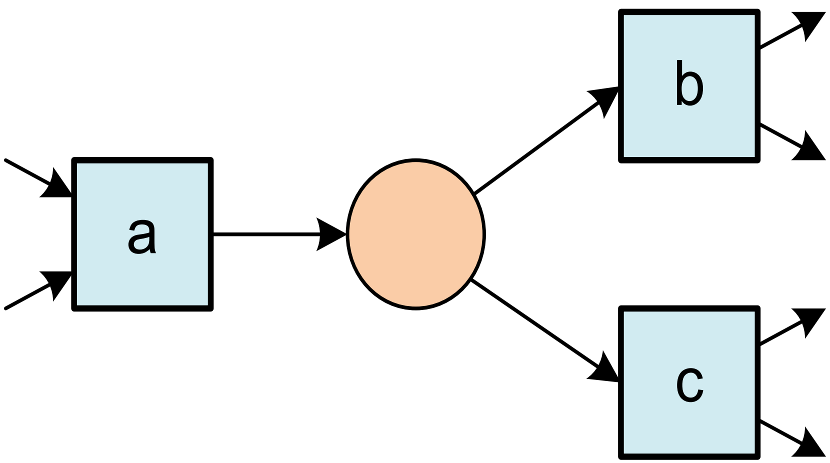


(a) sequence pattern: a→b

Discovery patterns

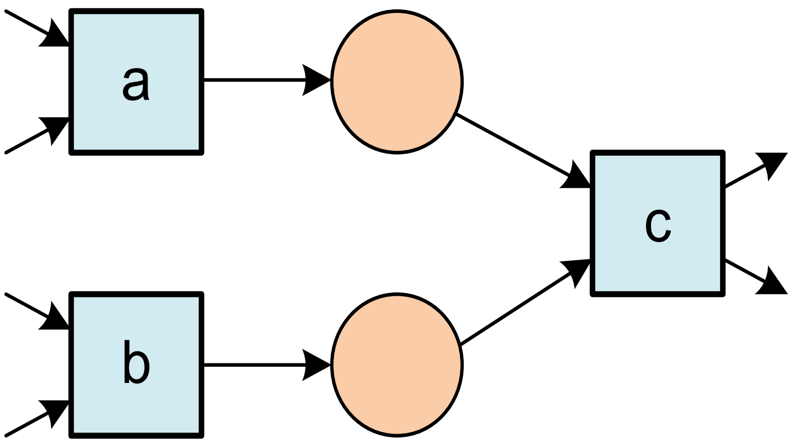


(c) XOR-join pattern:
a→c, b→c, and a#b

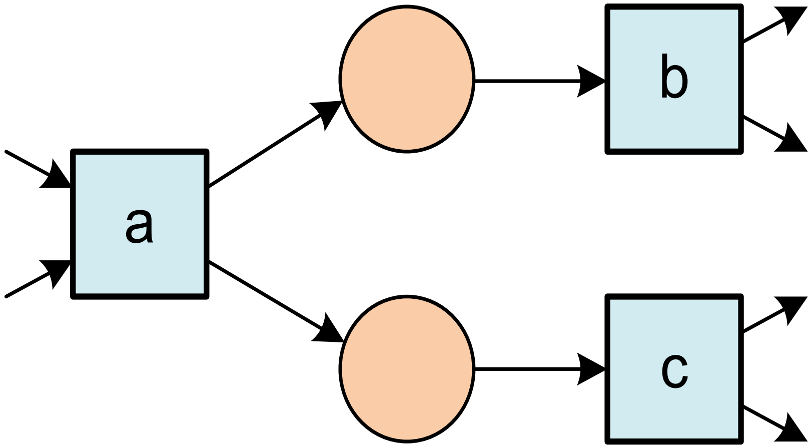


(b) XOR-split pattern:
a→b, a→c, and b#c

Discovery patterns

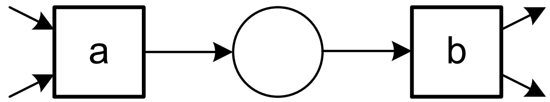


(e) AND-join pattern:
a→c, b→c, and a||b

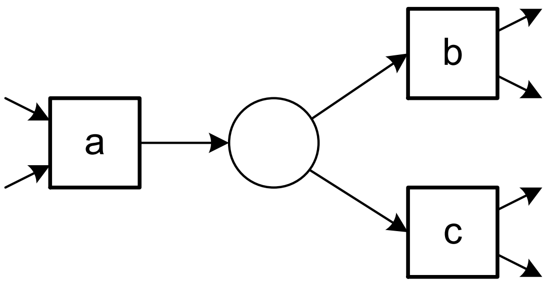


(d) AND-split pattern:
a→b, a→c, and b||c

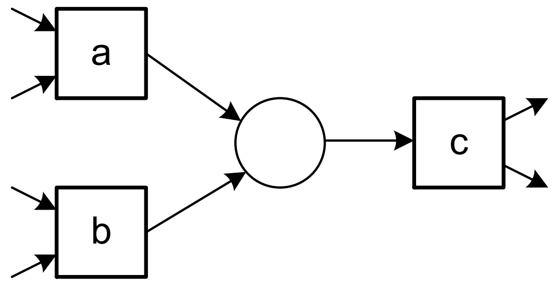
Simple patterns



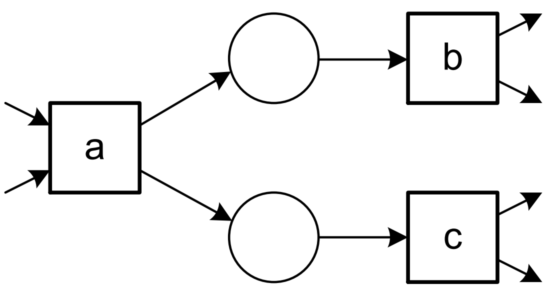
(a) sequence pattern: a→b



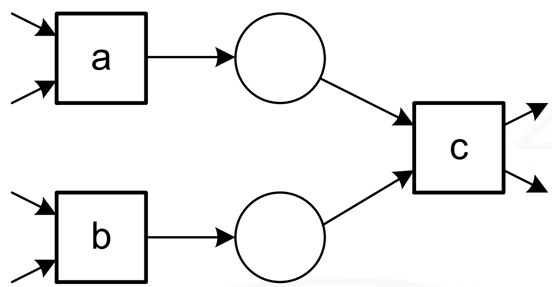
(b) XOR-split pattern:
a→b, a→c, and b#c



(c) XOR-join pattern:
a→c, b→c, and a#b



(d) AND-split pattern:
a→b, a→c, and b||c



(e) AND-join pattern:
a→c, b→c, and a||b

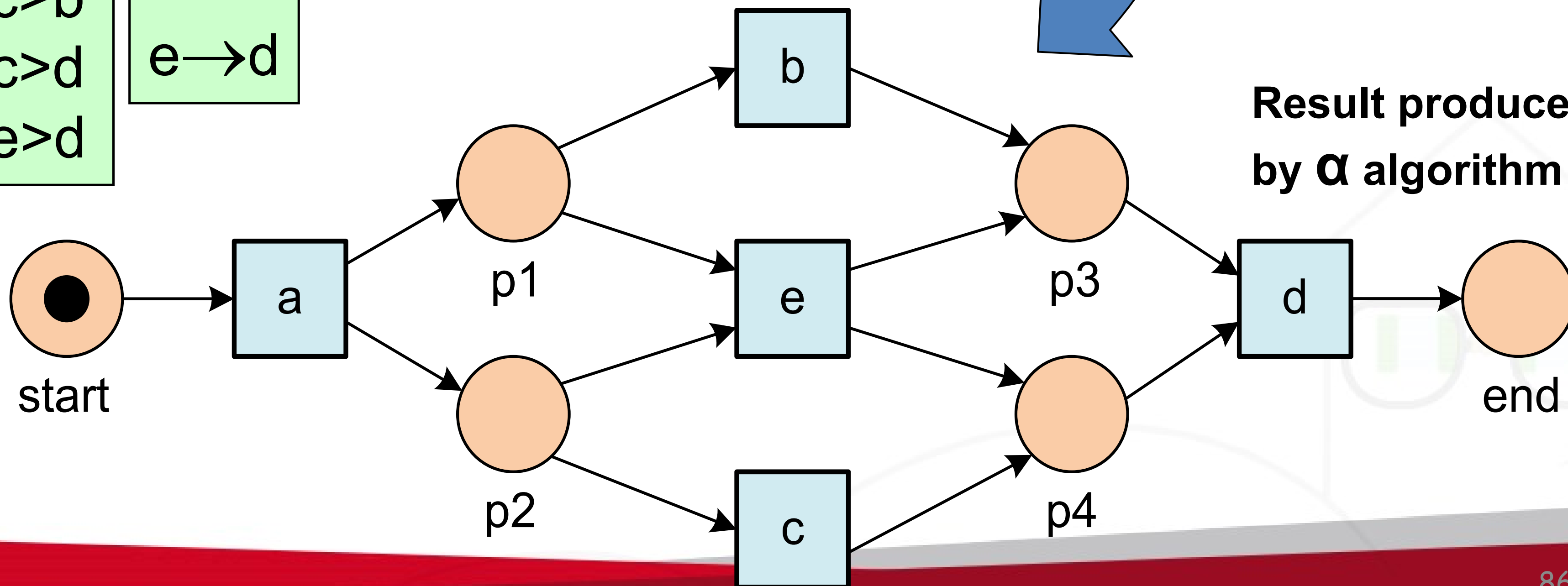
From discovery patterns to process model

$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

a>b	a→b	b||c	b#e
a>c	a→c	c||b	e#b
a>e	a→e		c#e
b>c	b→d		a#d
b>d	c→d		…
c>b	e→d		
c>d			
e>d			

The algorithm uses discovery patterns to build a process model

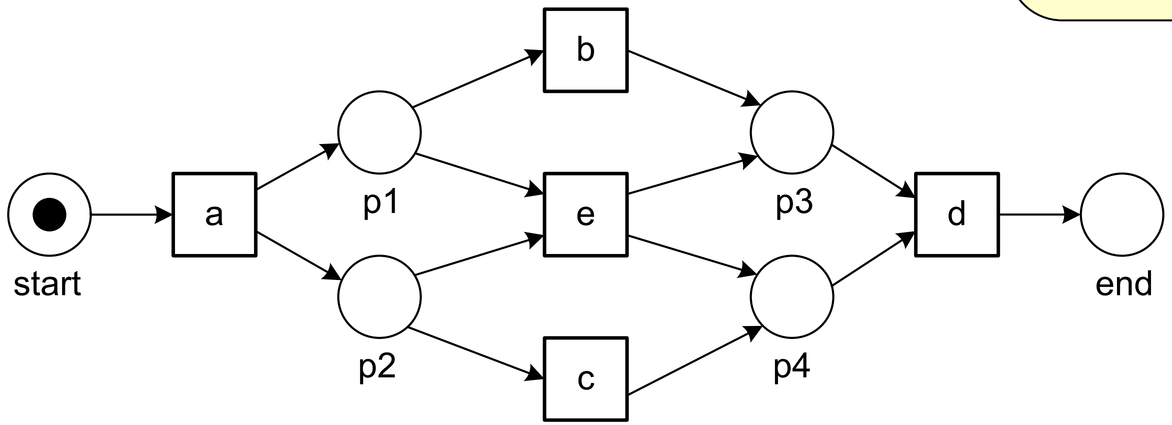
Result produced by α algorithm



Footprint of L1

$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

For any log it is possible to capture its footprint in a matrix. One of the following:
$\rightarrow, \leftarrow$ #, $\|$



	a	b	c	d	e
a	$\#_{L_1}$	$\rightarrow_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$
b	$\leftarrow_{L_1}$	$\#_{L_1}$	$\|_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
c	$\leftarrow_{L_1}$	$\|_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
d	$\#_{L_1}$	$\leftarrow_{L_1}$	$\leftarrow_{L_1}$	$\#_{L_1}$	$\leftarrow_{L_1}$
e	$\leftarrow_{L_1}$	$\#_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$

The complete algorithm

Let L be an event log over $T \subseteq A$. $\alpha(L)$ is defined as follows.

1. $T_L = \{ t \in T \mid \exists_{\sigma \in L}\ t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in L}\ t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in L}\ t = \mathit{last}(\sigma) \}$,
4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\ a \rightarrow_L b \wedge \forall_{a1,a2 \in A}\ a_1 \#_L a_2 \wedge \forall_{b1,b2 \in B}\ b_1 \#_L b_2 \}$,
5. $Y_L = \{ (A,B) \in X_L \mid \forall_{(A',B') \in X_L}\ A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\}$,
7. $F_L = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \{ (i_L, t) \mid t \in T_I \} \cup \{ (t, o_L) \mid t \in T_O \}$, and
8. $\alpha(L) = (P_L, T_L, F_L)$.

Another event log L3

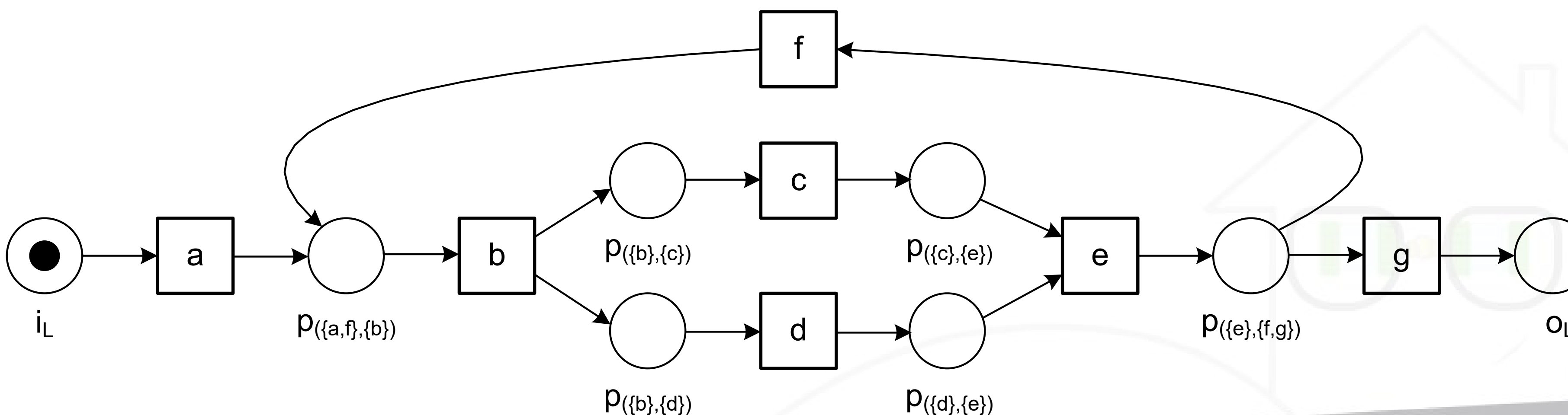
$$L_3 = [\langle a,b,c,d,e,f,b,d,c,e,g\rangle,$$
$$\langle a,b,d,c,e,g\rangle^2,$$
$$\langle a,b,c,d,e,f,b,c,d,e,f,b,d,c,e,g\rangle]$$

	a	b	c	d	e	f	g
a	#	$\rightarrow$	#	#	#	#	#
b	$\leftarrow$	#	$\rightarrow$	$\rightarrow$	#	$\leftarrow$	#
c	#	$\leftarrow$	#	$\parallel$	$\rightarrow$	#	#
d	#	$\leftarrow$	$\parallel$	#	$\rightarrow$	#	#
e	#	#	$\leftarrow$	$\leftarrow$	#	$\rightarrow$	$\rightarrow$
f	#	$\rightarrow$	#	#	$\leftarrow$	#	#
g	#	#	#	#	$\leftarrow$	#	#

Model for L3

	a	b	c	d	e	f	g
a	#	$\rightarrow$	#	#	#	#	#
b	$\leftarrow$	#	$\rightarrow$	$\rightarrow$	#	$\leftarrow$	#
c	#	$\leftarrow$	#	$\parallel$	$\rightarrow$	#	#
d	#	$\leftarrow$	$\parallel$	#	$\rightarrow$	#	#
e	#	#	$\leftarrow$	$\leftarrow$	#	$\rightarrow$	$\rightarrow$
f	#	$\rightarrow$	#	#	$\leftarrow$	#	#
g	#	#	#	#	$\leftarrow$	#	#

$$
\begin{aligned}
L_3 = [&\langle a,b,c,d,e,f,b,d,c,e,g\rangle, \\
&\langle a,b,d,c,e,g\rangle^2, \\
&\langle a,b,c,d,e,f,b,c,d,e,f,b,d,c,e,g\rangle]
\end{aligned}
$$



Limitations of the α-algorithm

- The α-algorithm guarantees to produce correct a process model provided that the underlying process can be described by a WF-net that:
  - does not contain duplicate activities (two transitions with the same activity label)
  - does not contain invisible transitions (activities that are not explicitly recorded in the event log)
  - does not contain some specific complex constructs (see later)

- The α-algorithm nicely illustrates some of the main ideas behind process discovery, but it has several limitations.

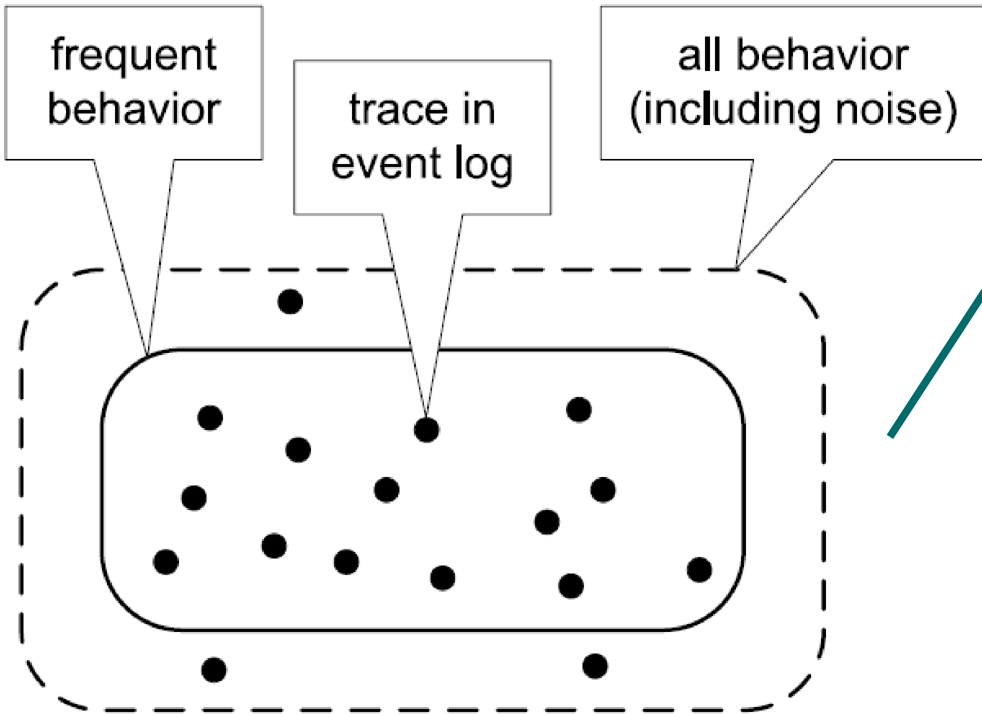
Limitations of the α-algorithm

- There are several extensions of the α-algorithm that overcome its weakness. The main ones are:
- Heuristic miners (Fuzzy Miner)
  - Extract footprints from the event logs (like the α-algorithm) and take *frequencies* into account to deal with noise and incompleteness.
- Region-based miners
  - they use a *2-steps approach* where: (1) a low-level model is built (e.g., transition systems or Markov models) and (2) is then converted in a high level-model (e.g., BPMN) that can express concurrency and more advanced control-flow patterns.

Noise and Incompleteness

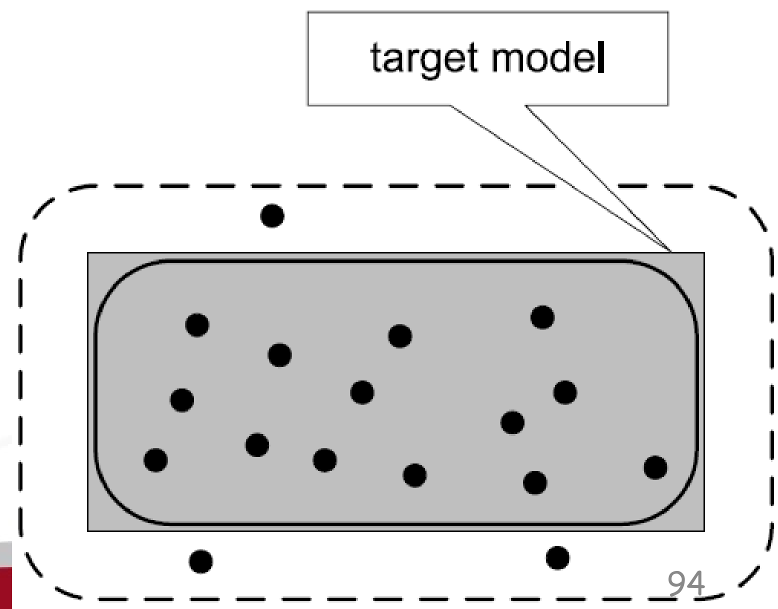
- An event log typically contains **only a fraction** of the possible process behavior.
- To successfully apply process mining in practice, one needs to deal with **noise** and **incompleteness**.
  - **Noise**: the event log contains *rare and exceptional behavior* not representative for the typical behavior of the process.
    - One is typically interested in frequent behavior and not in all possible ones.
  - **Incompleteness**: the event log contains *too few events* to be able to discover some of the underlying control-flow structures.
    - Many discovery algorithms make the *strong completeness* assumption (assuming that the log contains all possible behaviors).

Noise and Incompleteness



The **ideal process model** allows for the behavior coinciding with the frequent behavior seen when the process would be observed ad infinitum while being in steady state.

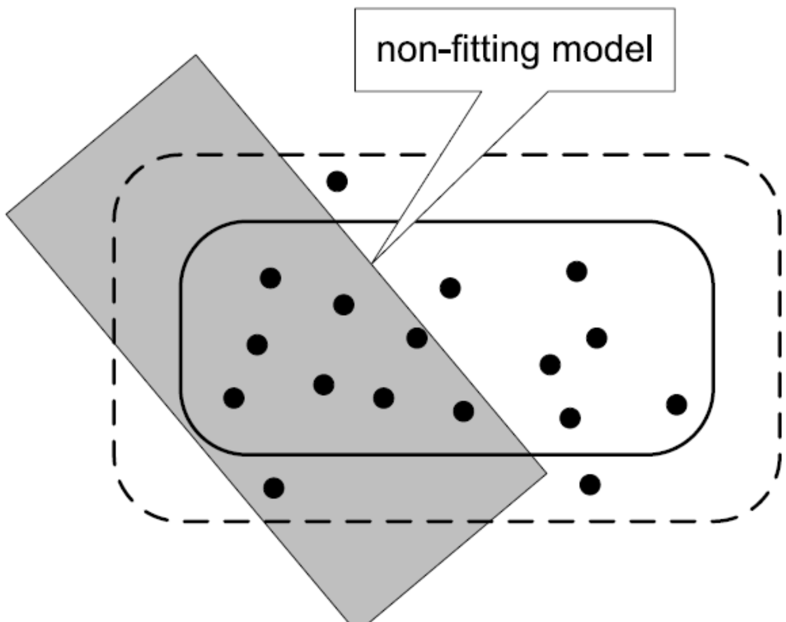
Mature process mining algorithms allow to **abstract** from infrequent behavior.



Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

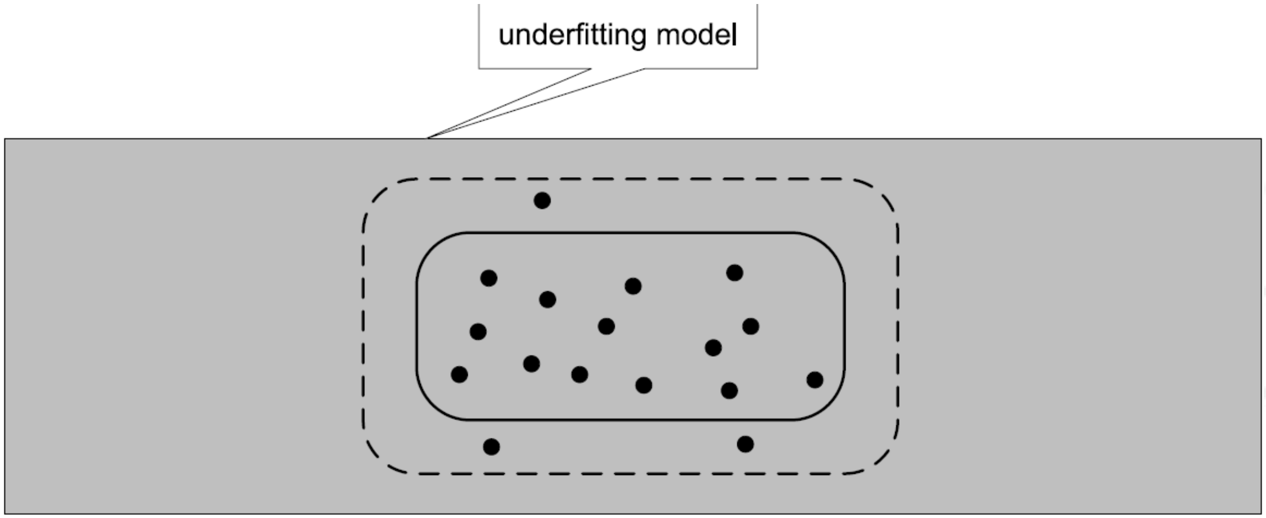
1. **Fitness**: the discovered model should allow for the behavior seen in the event log.
   - A model has a *perfect fitness* if all traces in the log can be replayed from the beginning to the end.



Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

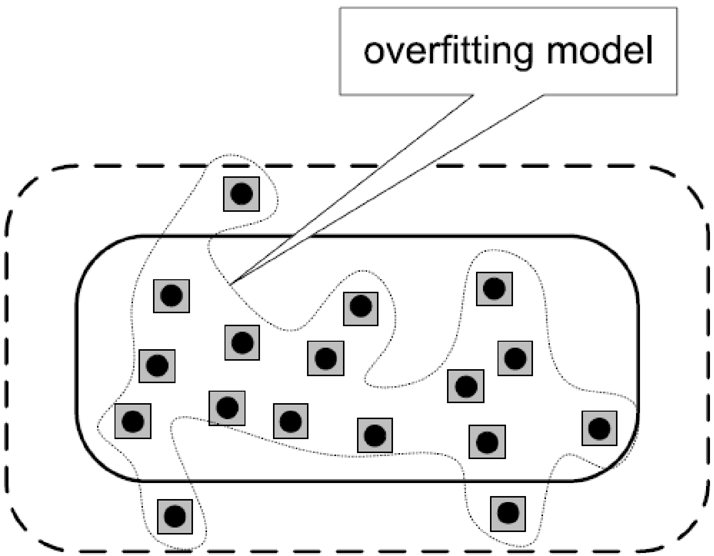
1. **Fitness**
2. **Precision** (*avoid underfitting*): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.



Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**:
2. **Precision** (avoid underfitting)
3. **Generalization** (avoid overfitting): the discovered model should generalize the example behavior seen in the event log.



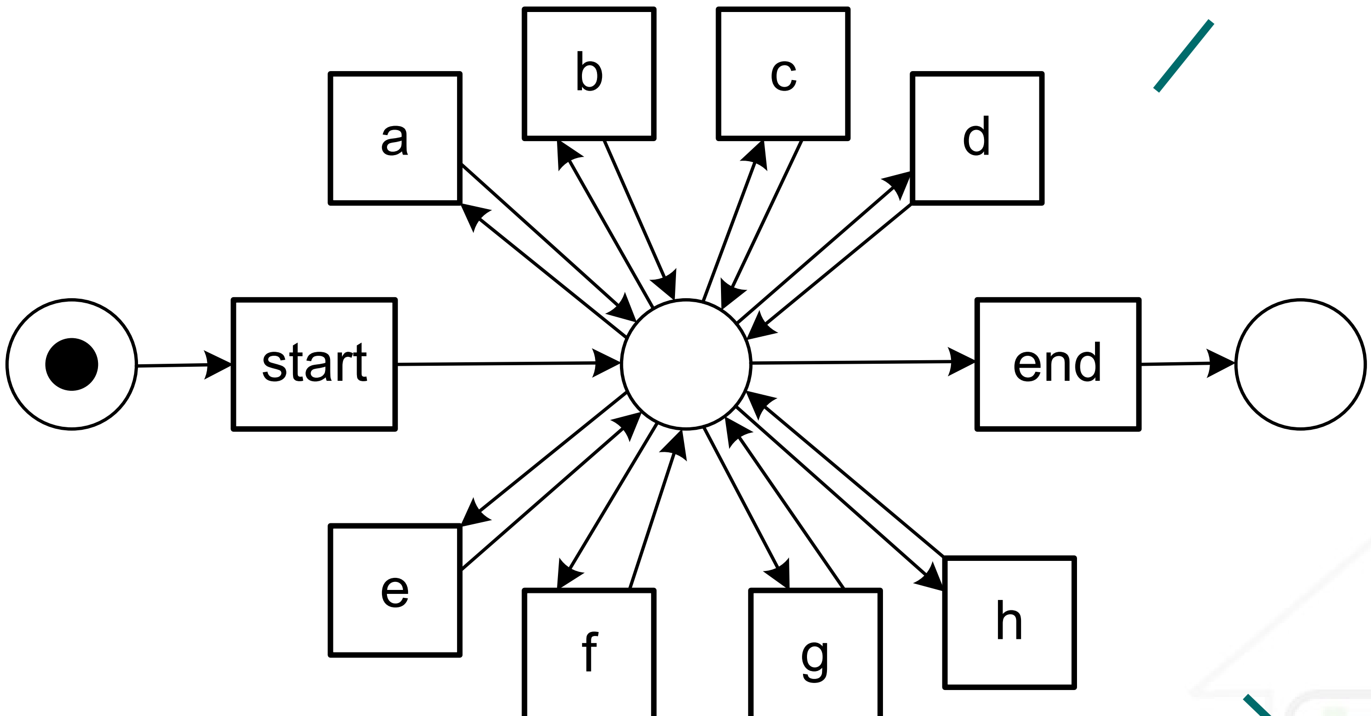
Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**
2. **Precision** (avoid underfitting)
3. **Generalization** (avoid overfitting)
4. **Simplicity**: the discovered model should be as simple as possible.
   - Occam's Razor: The simplest model that can explain the behavior seen in the log is the best model.
   - Metrics to quantify the **complexity** and **understandability** of a process model:
     - size of the model (e.g., the number of nodes and/or arcs),
     - "structuredness" or "homogeneity" of the model.

The "flower" model

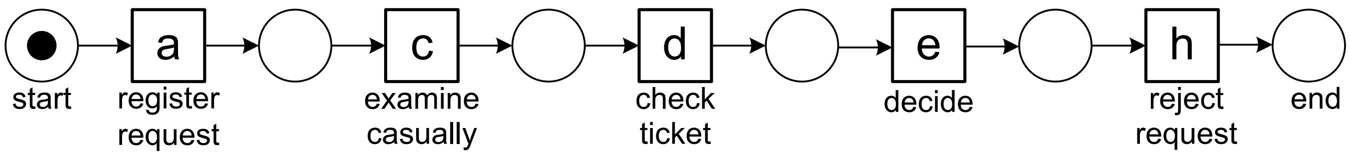
Fitness and simplicity alone **are not adequate**. The flower net allows for any sequence starting in start and ending in end. Basically, it can be constructed on the occurrences of activities only.



The flower net has **perfect fitness** and is **simple**, but it is **useless**. It does not contain any knowledge other than the activities in the event log.

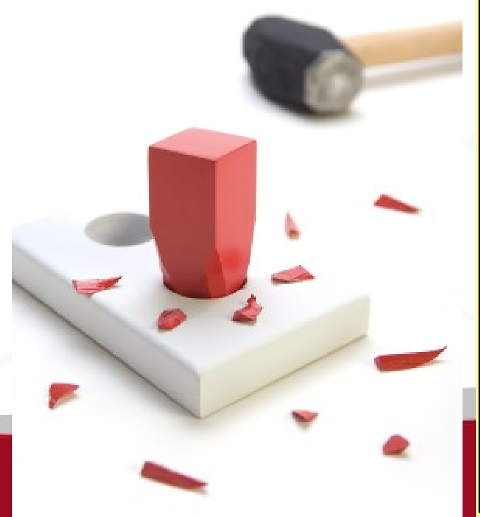
Model N_1

It models just the most frequent trace! Hence, it is very precise! None of the other traces is recognized. Hence, it also **does not generalize**.



N_2 : *fitness = -, precision = +, generalization = -, simplicity = +*

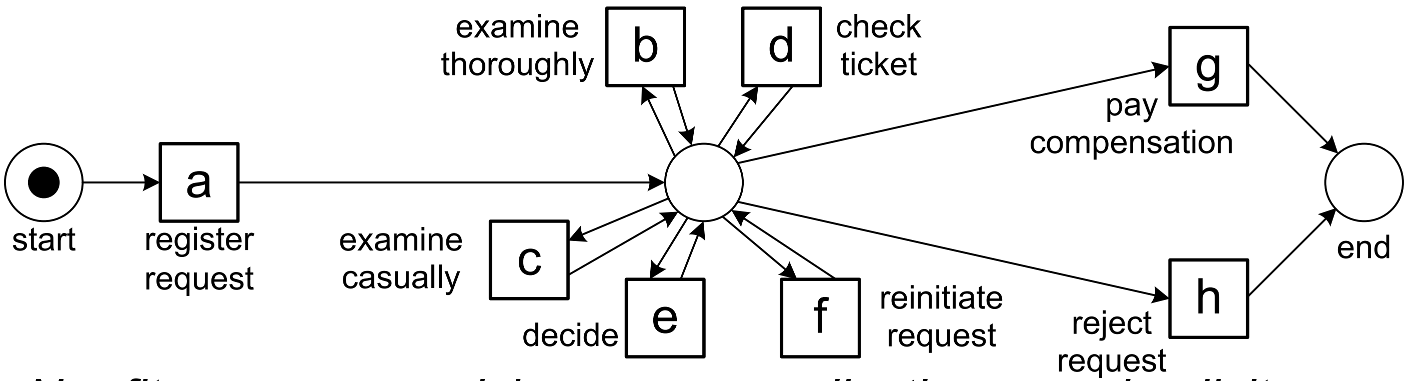
non-fitting



#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Model N_2

It **lacks precision**, since it allows for traces very different from what seen in the log. For example, <a,b,b,b,b,b,b,f,f,f,f,f,f,g> is possible.



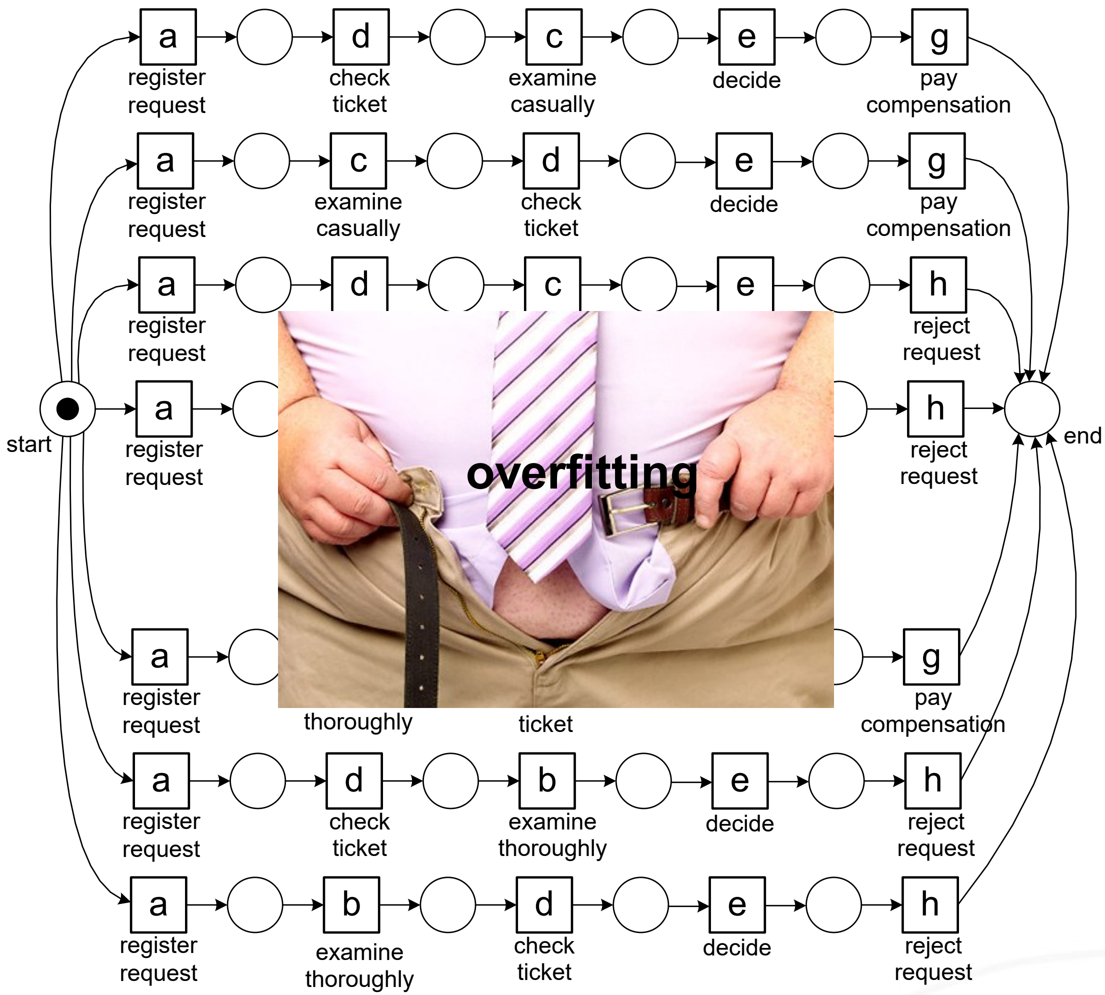
N_3 : fitness = +, precision = -, generalization = +, simplicity = +

underfitting

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Model N_3

Enumerating model. It recognizes all traces in the logs, but it **does not generalize!**

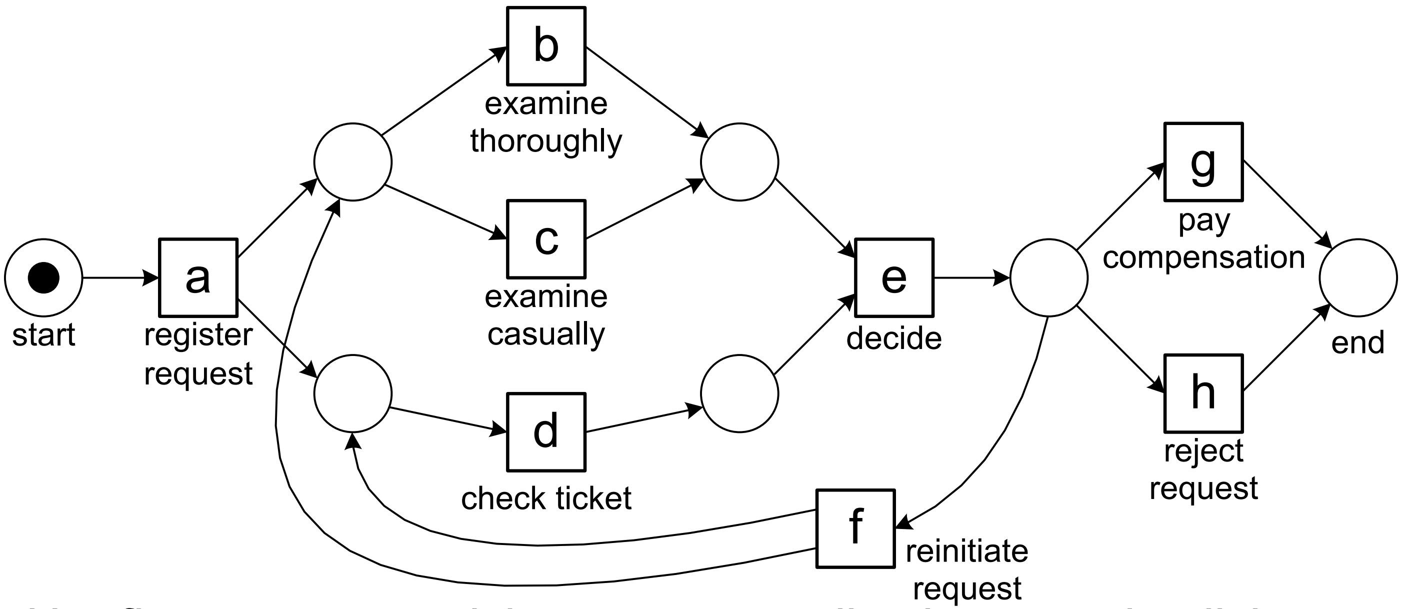


N_4 : fitness = +, precision = +, generalization = -, simplicity = -

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Model N_4

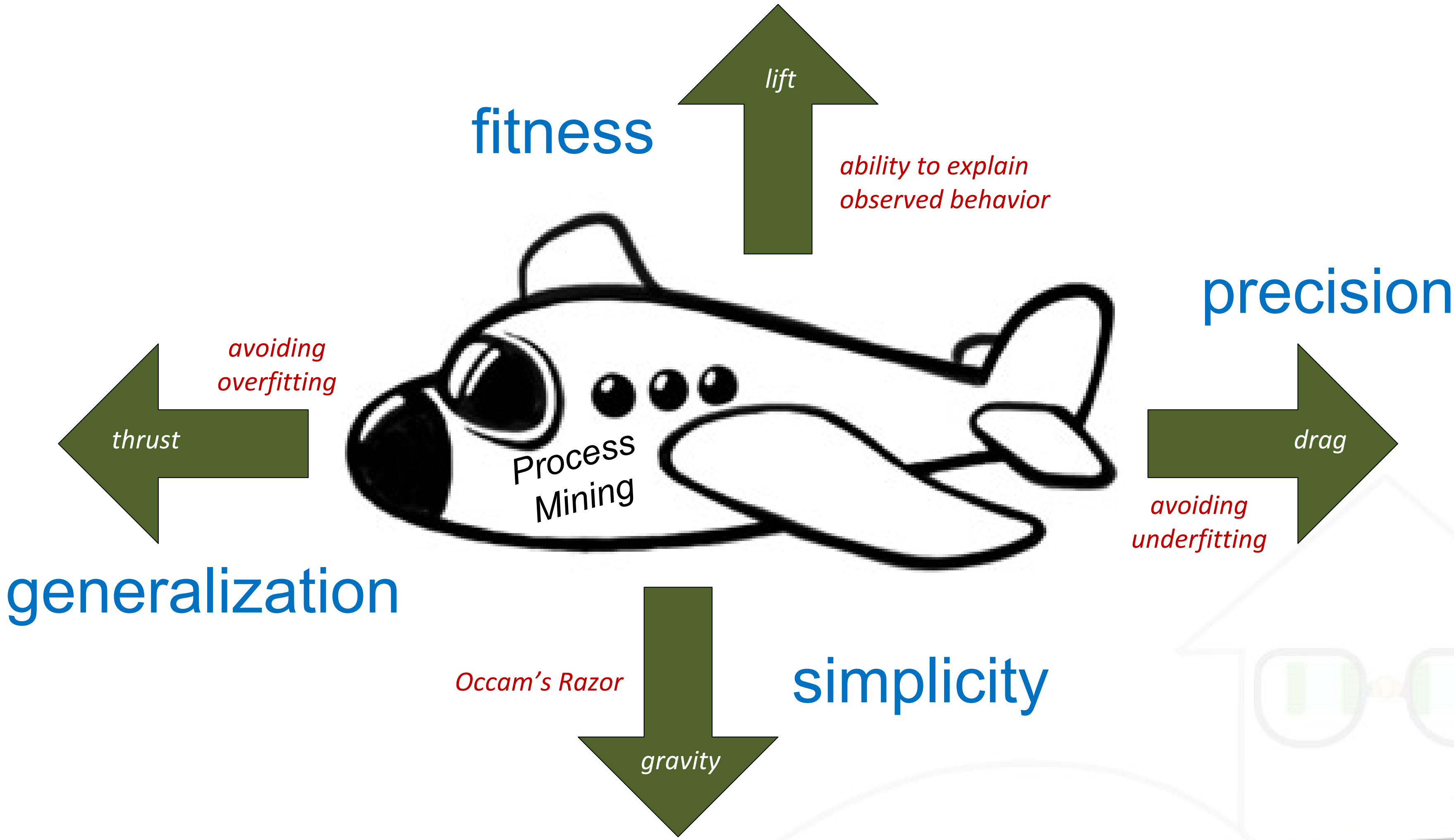
Good model that balances between precision and generalization!



N_1 : fitness = +, precision = +, generalization = +, simplicity = +

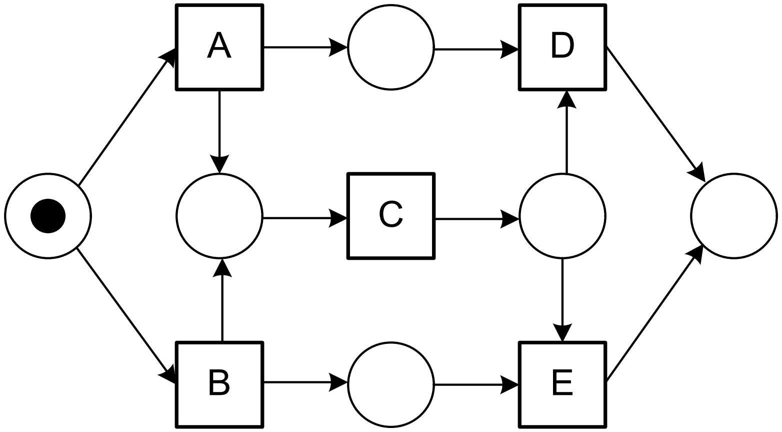
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

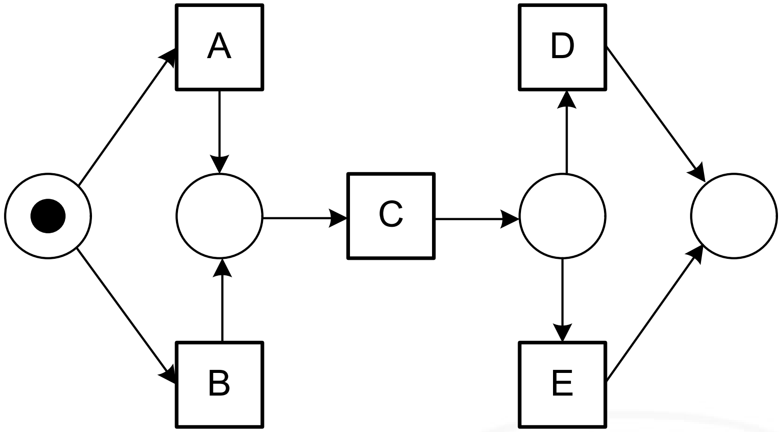
Challenge: find the right trade-off



What is the best model?

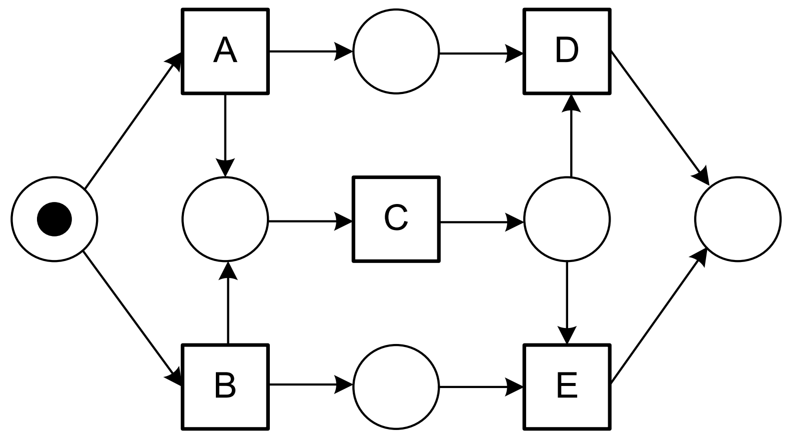
ACD **99**
BCE **85**

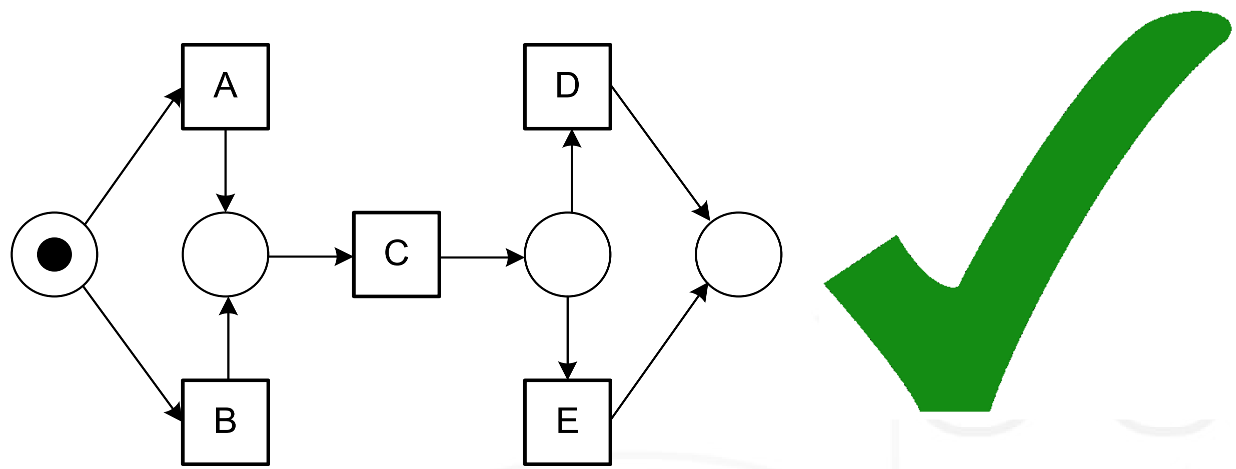




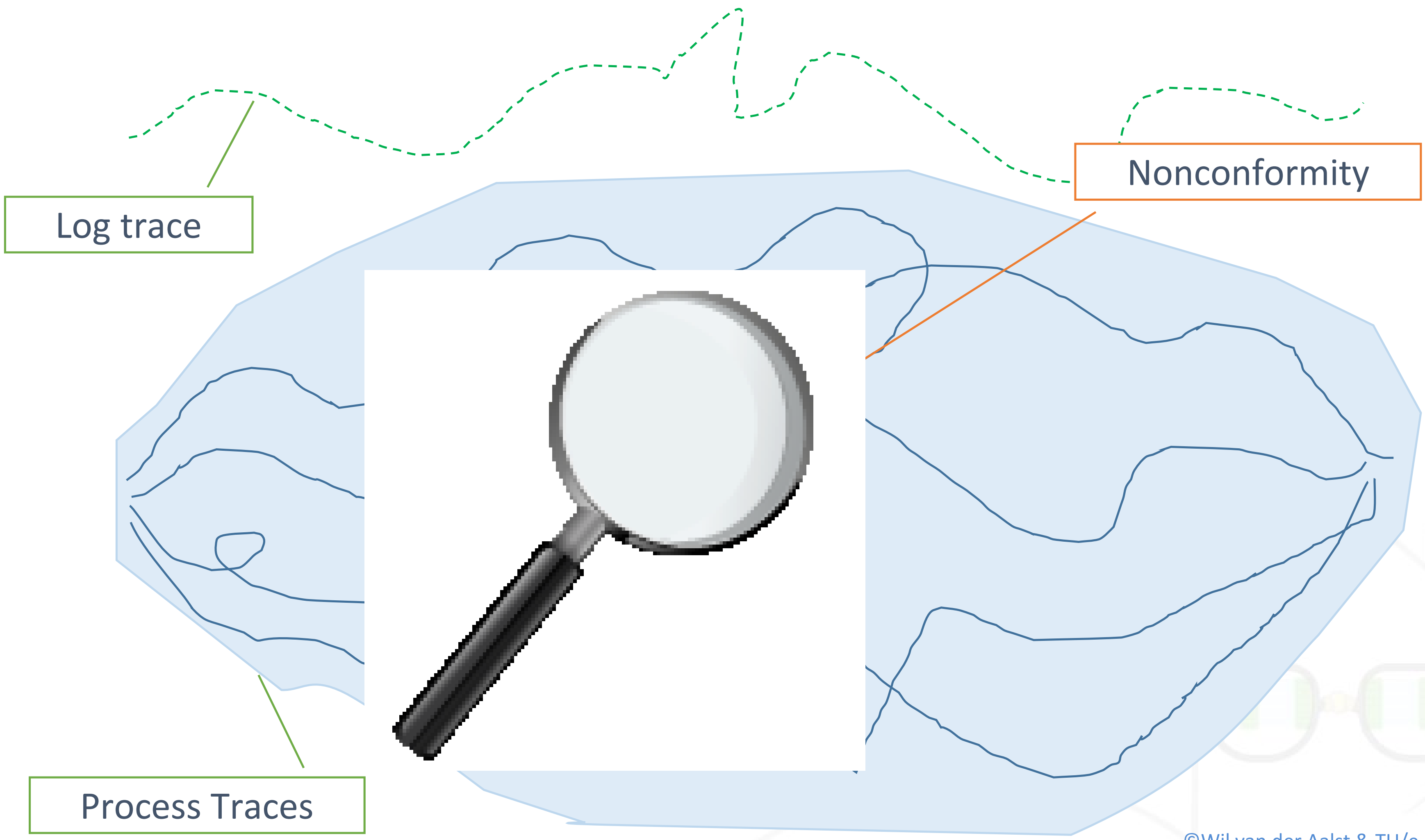
What is the best model?

ACD	99
ACE	88
BCE	85
BCD	78

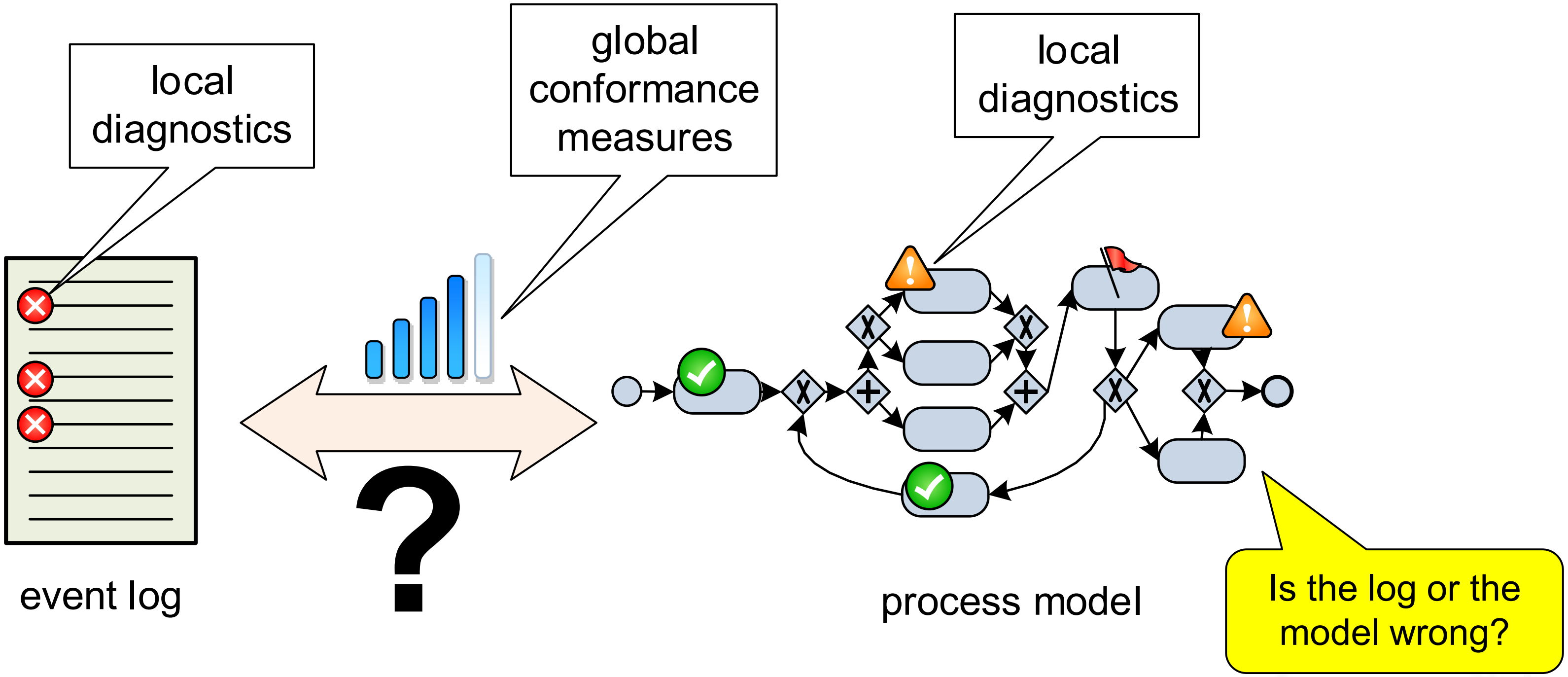




General Idea



Diagnostics



- **Global conformance measures** quantify the overall conformance of the model and log (e.g., 85% of the log traces can be replayed by the model).
- **Local diagnostics** are given by identifying the points in the model and in the log where model and log disagree (e.g., activity x was executed 15 times although this was not allowed according to the model).

Interpretation of non-conformance

- The **interpretation of non-conformance** depends on the purpose of the process model investigated.
- If the model is intended to be **descriptive**, or if it has been **discovered** from an event log of small size, then discrepancies between model and log indicate that the ***model needs to be improved*** to capture reality better.
- If the model is **normative**, discrepancies may be interpreted as ***undesirable*** or ***desirable deviations***.
  - *undesirable deviations* signal the need for a better control of the process.
  - *desirable deviations* happen to handle circumstances not foreseen by the process model, e.g., to serve a customer better.
- Conformance checking is (especially) relevant for **business alignment** and **auditing**.

Business Alignment

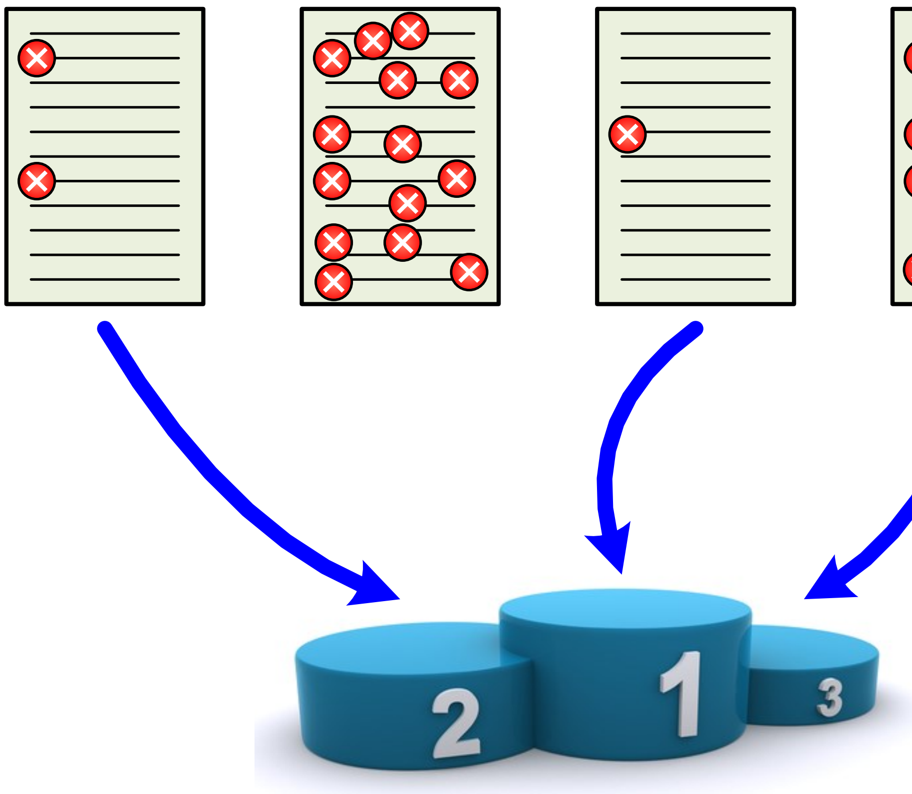
- **ISO 9001:2008** requires organizations to model their processes.
  - There is often a **mismatch** between the information systems on the one hand and the actual processes and needs of workers on the other hand.
  - First of all, many organizations use **product software**, i.e., generic software that was not developed for a specific organization (e.g., SAP).
    - Although such systems are configurable, the particular needs of an organization may be different from what was envisioned by the product software developer.
  - Second, **processes may change faster than the information system.**
  - Finally, there may be different stakeholders in the organization having **conflicting requirements**.
    - A manager may want to enforce a fixed working procedure whereas an experienced worker prefers to have more flexibility to serve customers better.
- **Business alignment** makes sure that the information systems and the real business processes are well aligned.
  - Conformance Checking can be **successfully employed** for this task.

Auditing

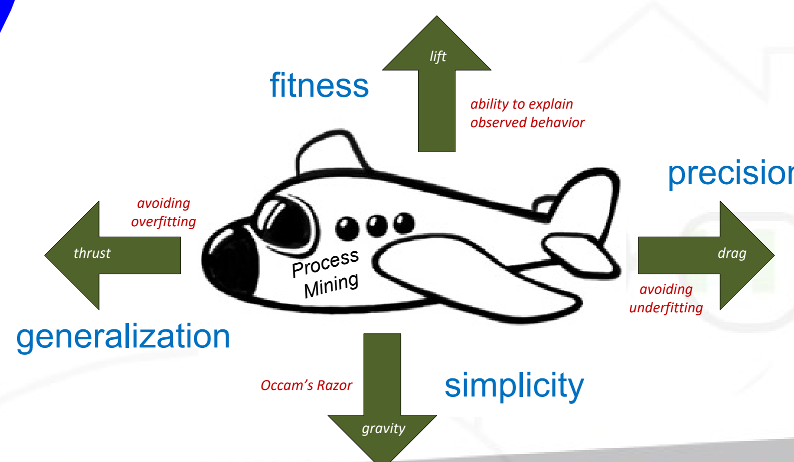
- Audits are performed to check if business processes are executed ***within certain rules*** enforced by managers and governments.
  - An auditor should check whether these rules are followed or not.
- Traditionally, auditors can only provide ***reasonable assurance*** that processes are executed within the given set of rules.
  - When these controls are not in place hey typically ***only check samples of factual data off-line***, often in the "paper world".
- The availability of logs and conformance checking techniques allows **new forms of auditing** that automatically detect violations of these rules indicating fraud, risks, and inefficiencies.
  - All events in a business process can be evaluated and this can also be done while the process is still running.

Another important use case
Evaluation of process mining algorithms

Model 1 produced by algorithm A	Model 2 produced by algorithm B	Model 3 produced by algorithm C	Model 4 produced by algorithm D



Evaluation based on the **quality criteria**.



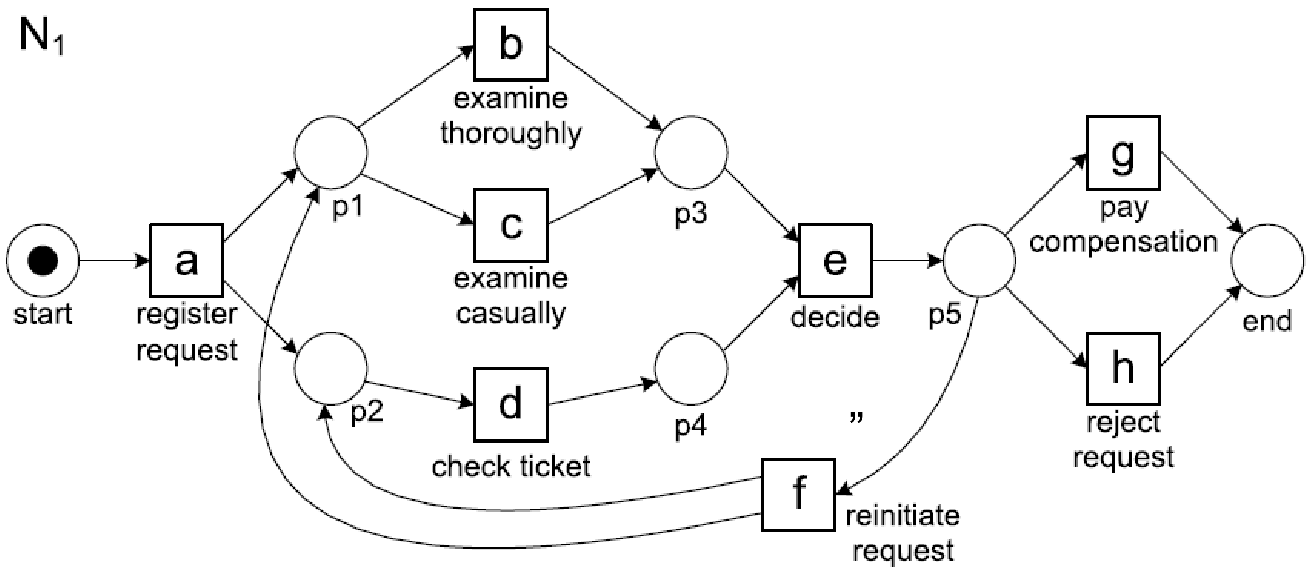
Fitness for Conformance Checking

- **Fitness** measures "*the proportion of behavior in the event log possible according to the process model*".
  - It can vary from 0 to 1 (**perfect fitness**)
- Of the four quality criteria, fitness is most related to conformance checking.

➢ But…**how to compute the fitness value?**

- A "simple approach" to compute fitness is to **count the fraction of cases** that can be "parsed completely" (i.e., the proportion of cases corresponding to firing sequences leading from [start] to [end]).

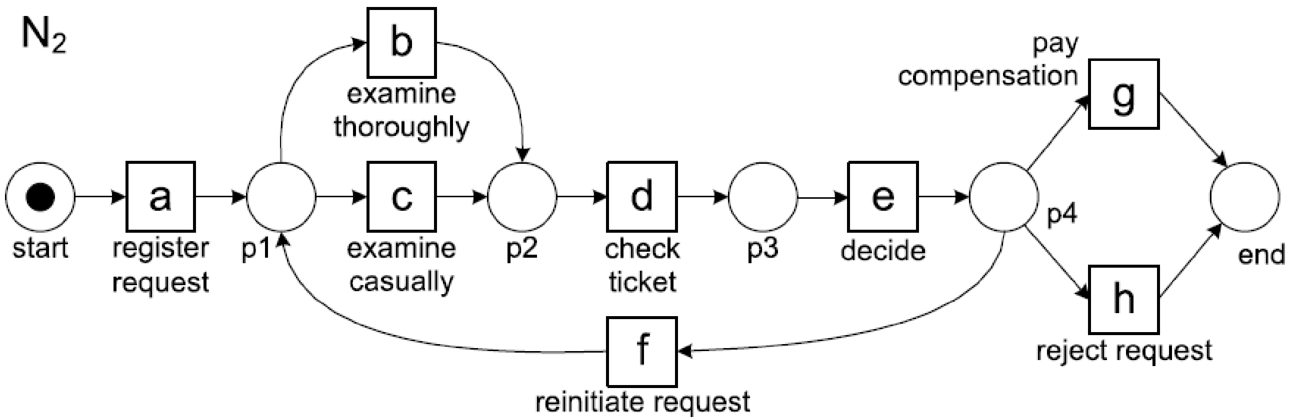
Example for Model N_1



- WF-net N_1 is the process model discovered when applying the α-algorithm to the entire log.
- Using the "simple approach", fitness of N_1 is $\frac{1391}{1391} = 1$
  - All 1391 cases in the log correspond to a firing sequence of N_1 (they can be <u>completely replayed</u>).

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

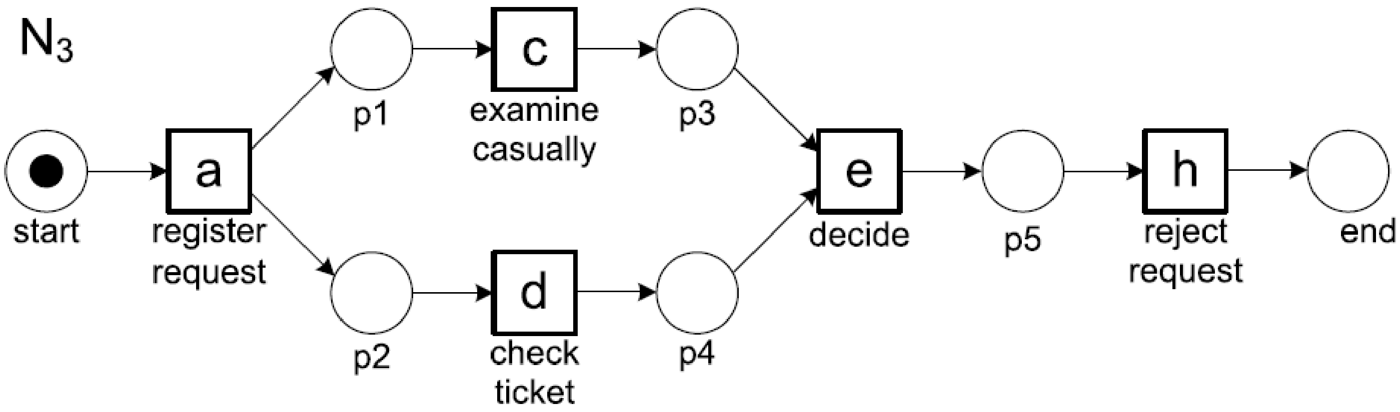
Example for Model N_2



#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

- WF-net N_2 does not allow for replaying all log traces.
  - For example, trace ‹*adceg*› can not be replayed.
- Using the "simple approach", fitness of N_2 is $\frac{948}{1391} = 0{,}6815$
  - 948 cases can be replayed correctly whereas 443 cases do not correspond to a firing sequence of N_2

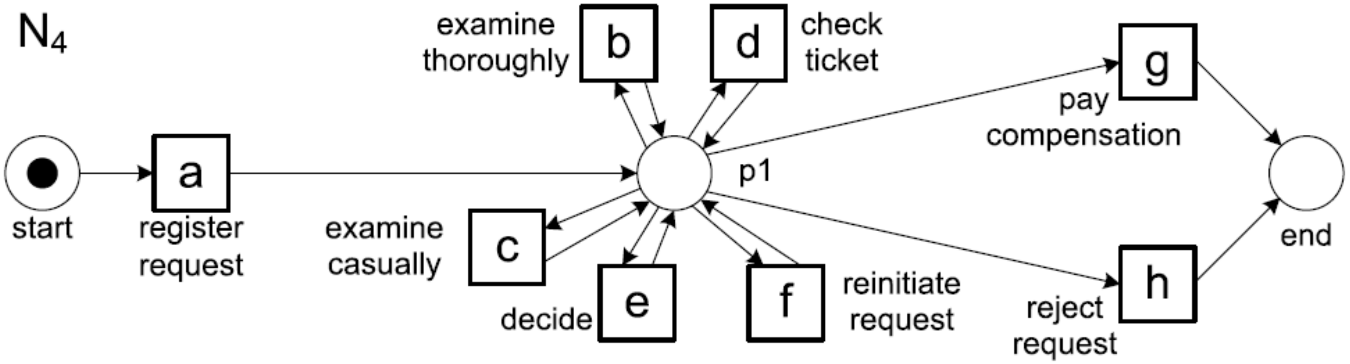
Example for Model N_3



- WF-net N_3 has no choices, e.g., the request is always rejected.
  - Many traces in the log cannot be replayed by this model, for example, σ_2 = <a,b,d,e,g> is not possible according to N_3
- The fitness of N_3 is $\frac{632}{1391} = 0{,}4543$
  - Only 632 cases have a trace corresponding to a firing sequence of N_3.

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Example for Model N_4



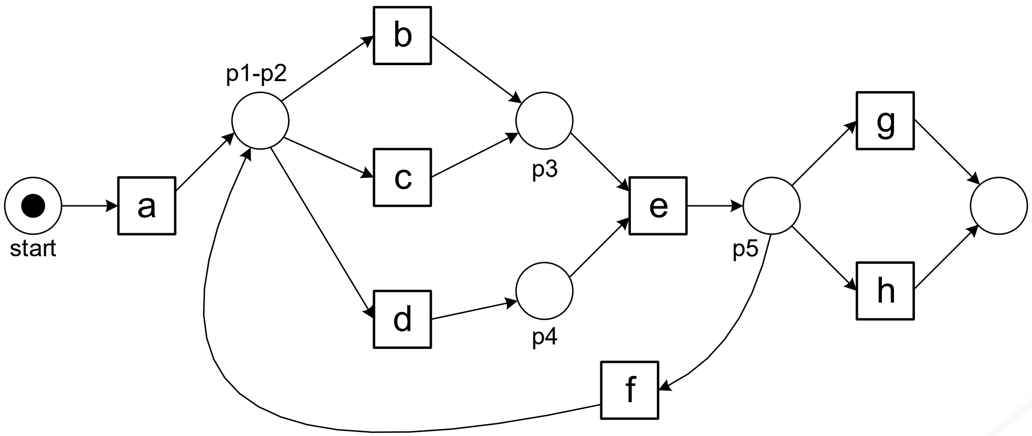
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

- WF-net N_4 is a variant of the "flower model".
  - The only requirement is that traces need to start with *a* and end with *g* or *h*.

- The fitness of N_4 is $\frac{1391}{1391} = 1$, because the model is able to replay all traces in the log.

Limitation of the simple fitness metric

- The simple fitness metric is not really suitable for more realistic processes.
  - Consider, for example, a variant of WF-net N_1 in which places p1 and p2 are merged into a single place.



  - This model variant has a fitness of $\frac{0}{1391} = 0$, because none of the traces can be replayed.
  - This fitness notion is **too strict** as most of the model seems to be consistent with the event log.

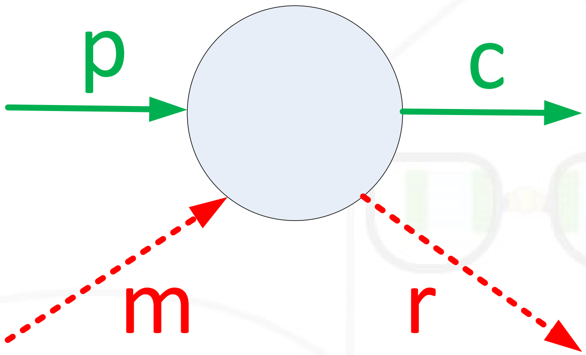
Limitation of the simple fitness metric

- This is especially the case for larger process models. Consider, for example, a trace $\sigma = \langle a_1, a_2, ..., a_{100} \rangle$ in some log L.
- Now consider a model that cannot replay σ , but that can replay 99 of the 100 events in σ (i.e., *the trace is "almost" fitting*).
- Also consider another model that can only replay 10 of the 100 events in σ (i.e., *the trace is not fitting at all*).
- Using the simple fitness metric, the trace would simply be classified as non-fitting for both models without acknowledging that σ was almost fitting in one model and in complete disagreement with the other model.

➢ We need to define a fitness notion defined at the **level of events** rather than full traces.

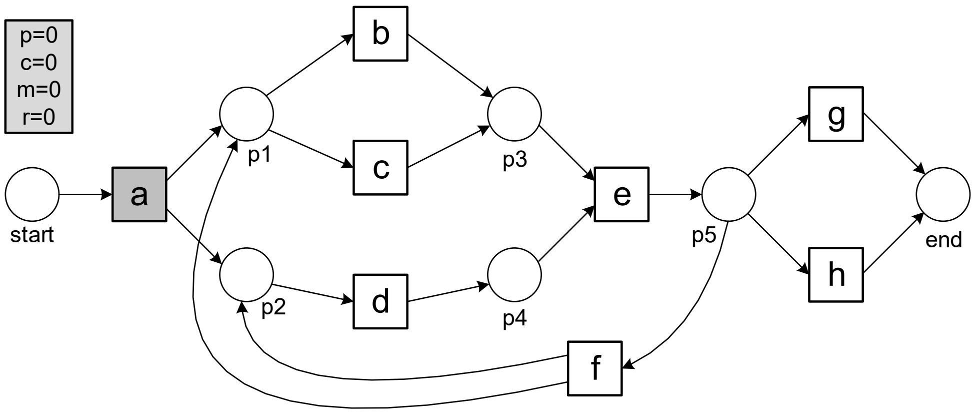
Event-based approach for fitness

- In the simple fitness computation, **we stopped replaying a trace** once we encounter a problem and mark it as **non-fitting**.
- An event-based approach to calculate fitness consists of just continue replaying the trace on the model and:
  - record all situations where a transition is forced to fire without being enabled, i.e., **we count all missing tokens**.
  - record the **tokens that remain at the end**.
- Use of four counters:
  - p = produced tokens
  - c = consumed tokens
  - m = missing tokens
  - r = remaining tokens

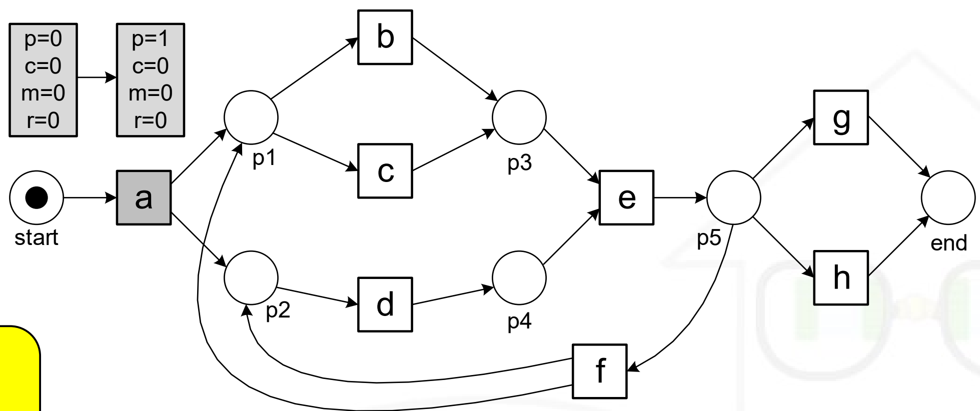


Replaying σ_1 on N_1 (1/4)

$$\sigma_1 = \langle a, c, d, e, h \rangle$$



Initially, **p = c = 0** and all places are empty.

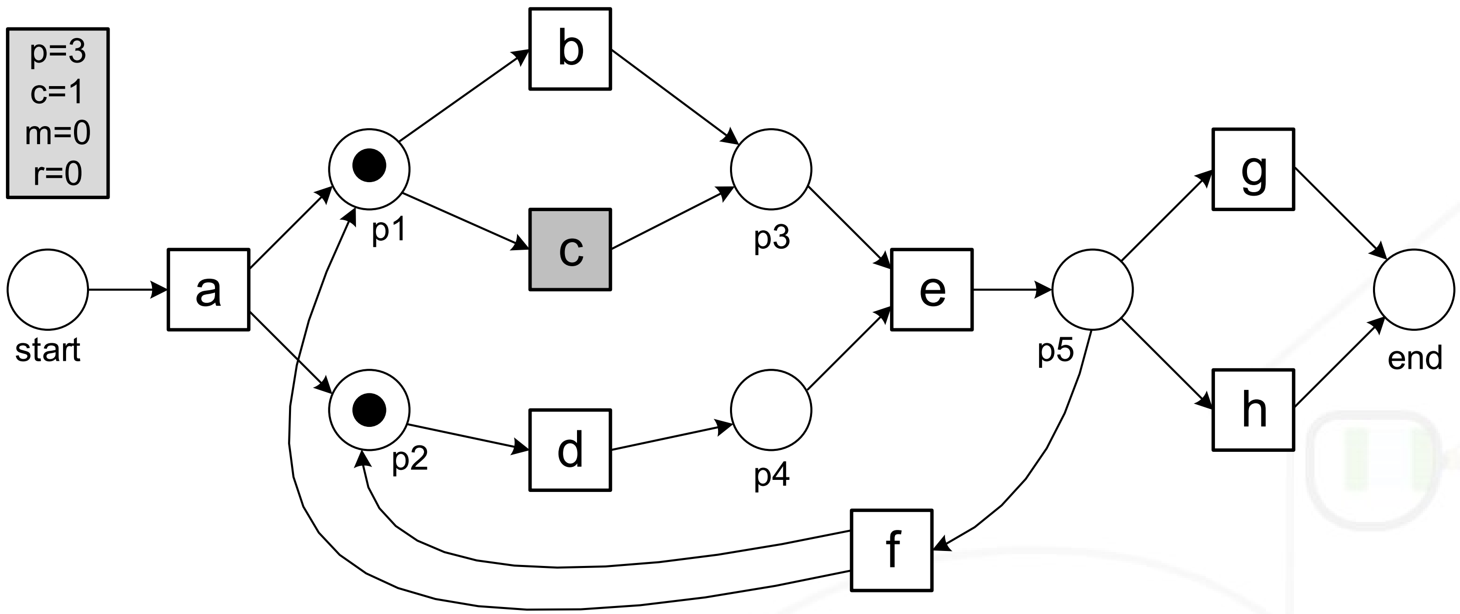


At the beginning the environment produces a token for place start. Therefore, the **p** counter is incremented: **p = 1**.

Replaying σ_1 on N_1 (2/4)

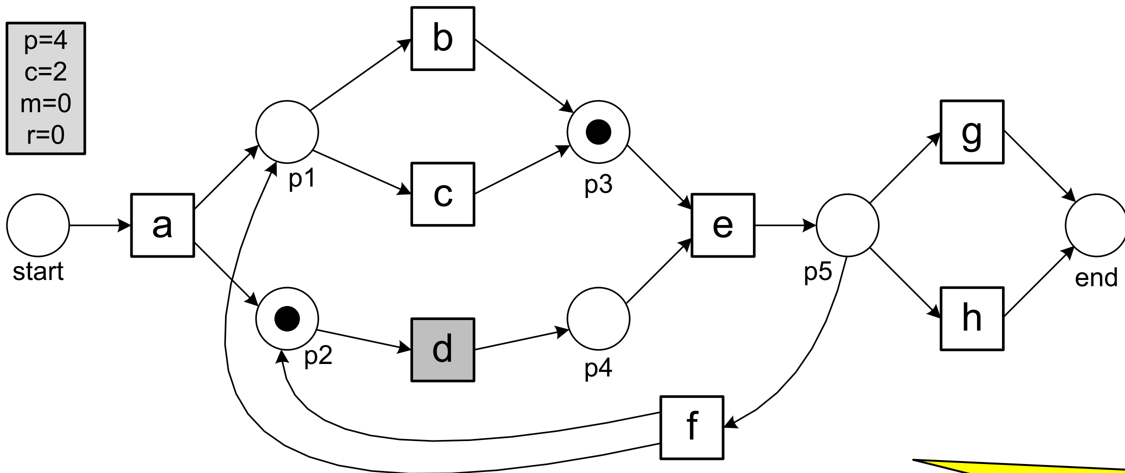
$$\sigma_1 = \langle a, c, d, e, h \rangle$$

We first fire transition *a*. Since *a* consumes one token and produces two tokens, the **c** counter is incremented by 1 and the **p** counter is incremented by 2. Therefore, **p = 3** and **c = 1.**

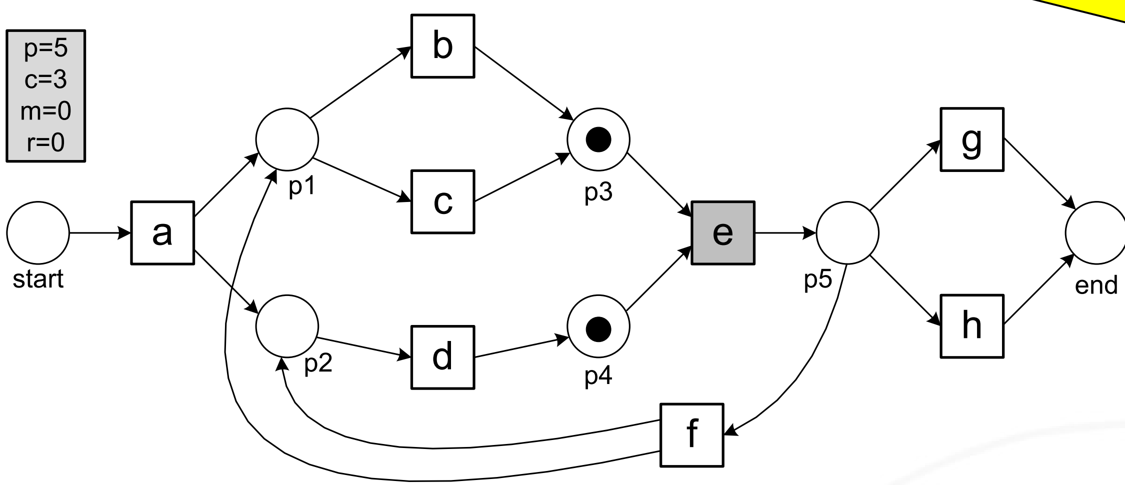


Replaying σ_1 on N_1 (3/4)

$$\sigma_1 = \langle a, c, d, e, h \rangle$$



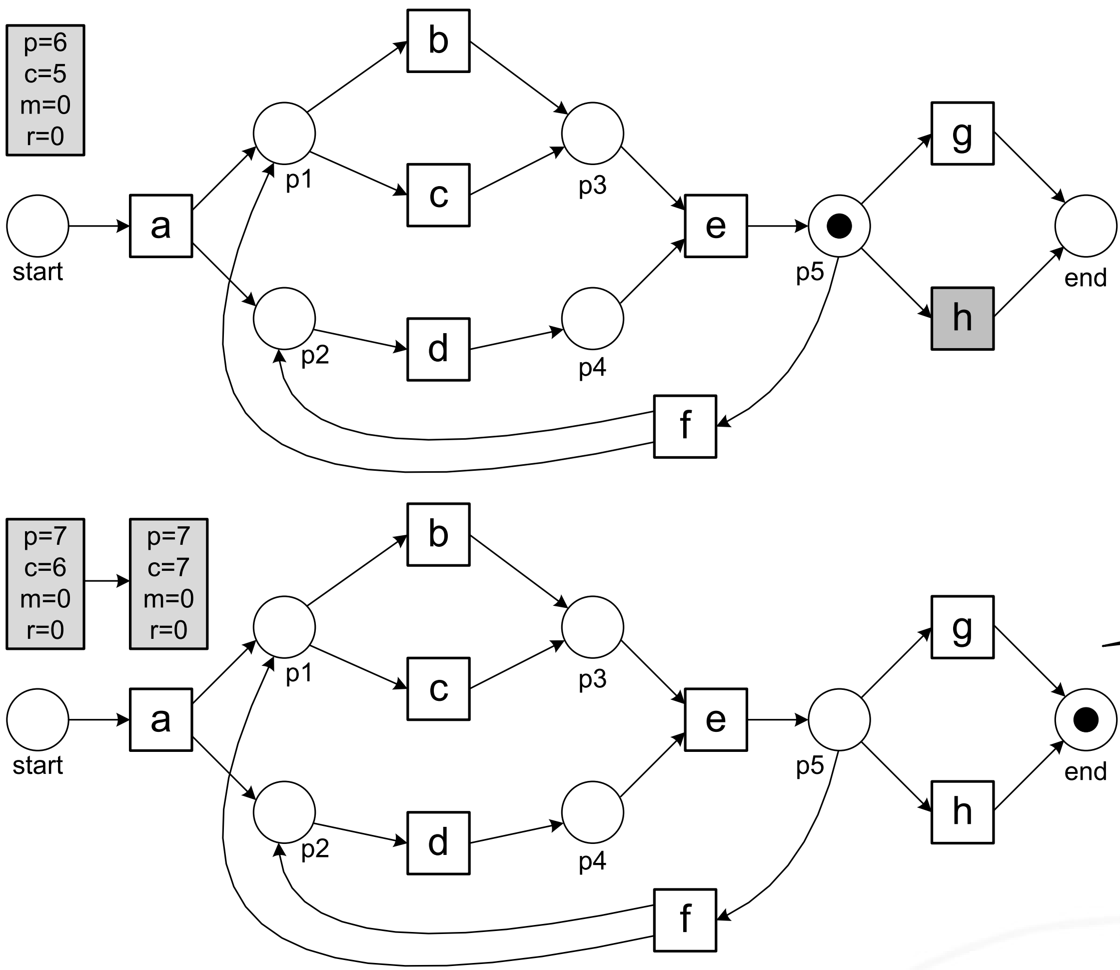
Then we replay the second event *c* and the third event *d*.
p = 5 and **c = 3**



$$\sigma_1 = \langle a, c, d, e, h \rangle$$

Replaying σ_1 on N_1 (4/4)

$$\sigma_1 = \langle a, c, d, e, h \rangle$$



At the end, the environment consumes one token from place end. Hence, **p = c = 7**

No missing or remaining token
m=r=0

No problems found!

Event-based fitness of N_1

- The fitness of a case with trace σ on WF-net N is defined as follows:

$$fitness(\sigma, N) = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right)$$

We pay a **penalty** when there are missing or remaining tokens.

The first part computes the fraction of missing tokens relative to the number of consumed tokens

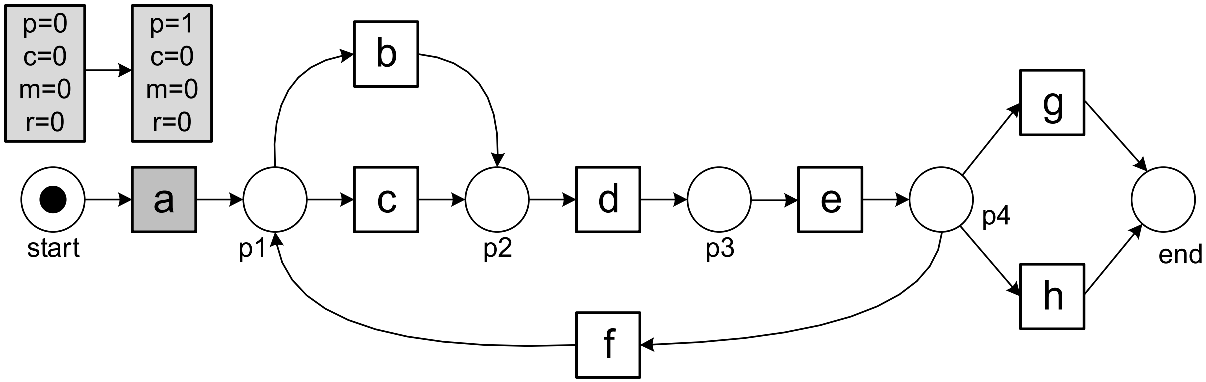
The second part computes the fraction of remaining tokens relative to the number of produced tokens

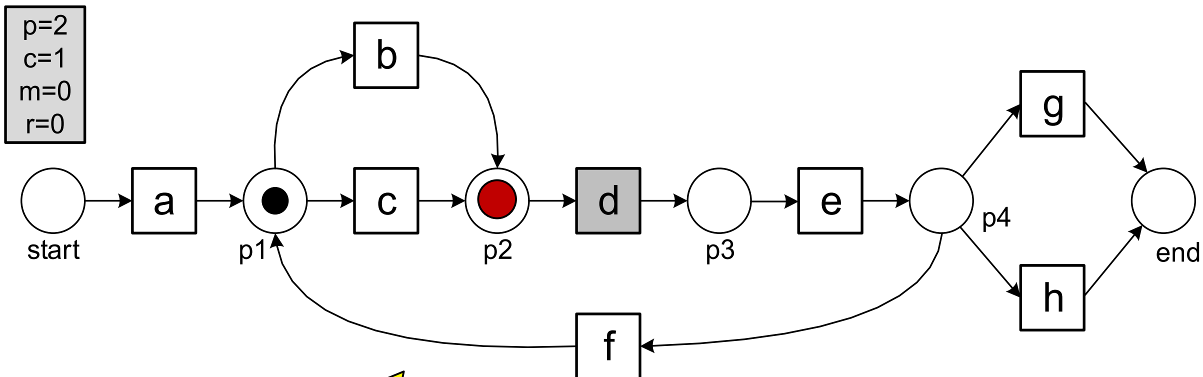
$$fitness(\sigma_1, N_1) = \tfrac{1}{2}(1 - \tfrac{0}{7}) + \tfrac{1}{2}(1 - \tfrac{0}{7}) = 1$$

Replaying σ_3 on N_2 (1/3)

$$\sigma_3 = \langle a, d, c, e, h \rangle$$

STOP





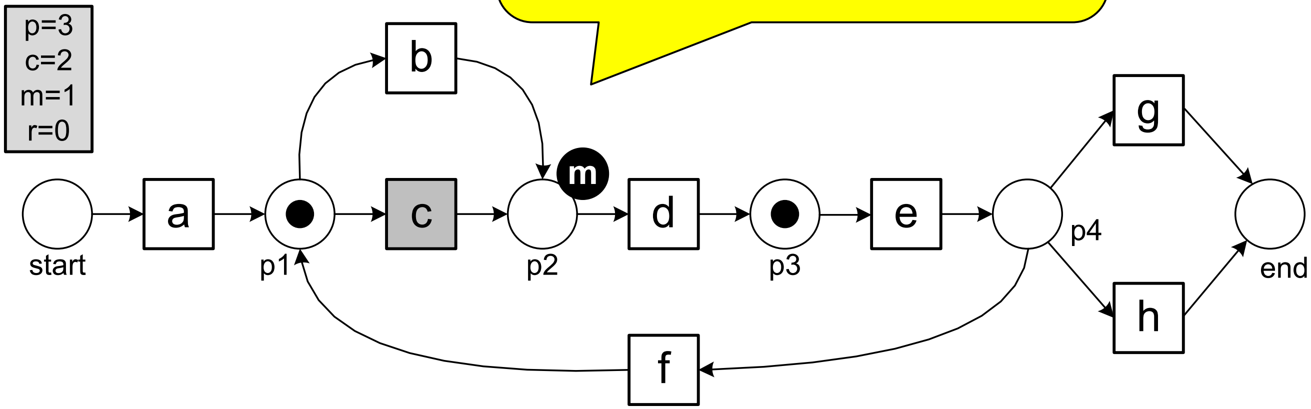
We would like to fire d…but it this not possible, because d **is not enabled**.

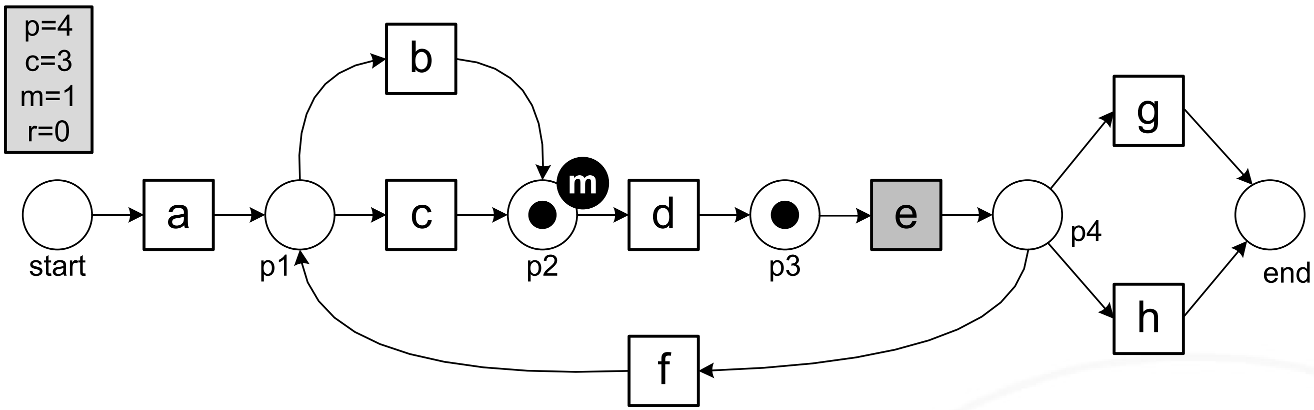
We need to add a **new token** that enables to fire activity d.

Replaying σ_3 on N_2 (2/3)

$$\sigma_3 = \langle a, d, c, e, h \rangle$$

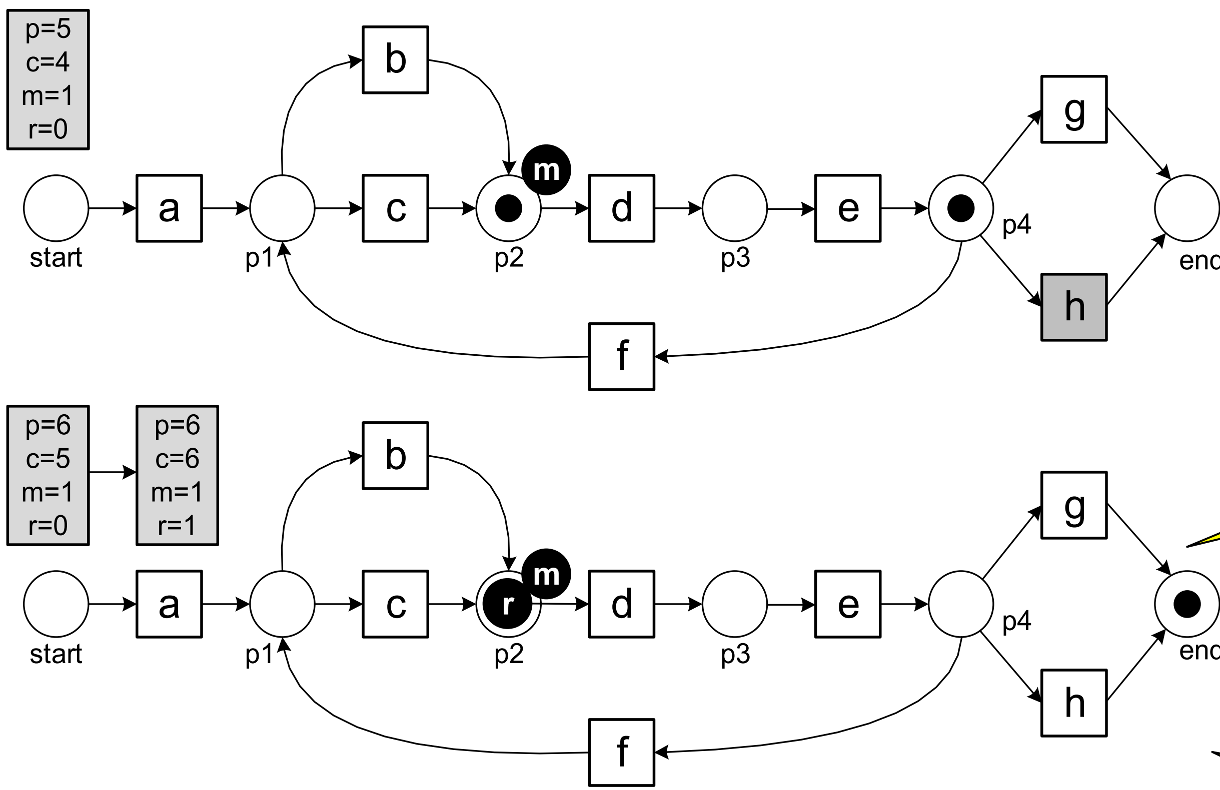
To fire d, we need to add a token in place p2 and **record the missing token** with a **m-tag**, i.e., the **m counter is incremented.**





Replaying σ_3 on N_2 (3/3)

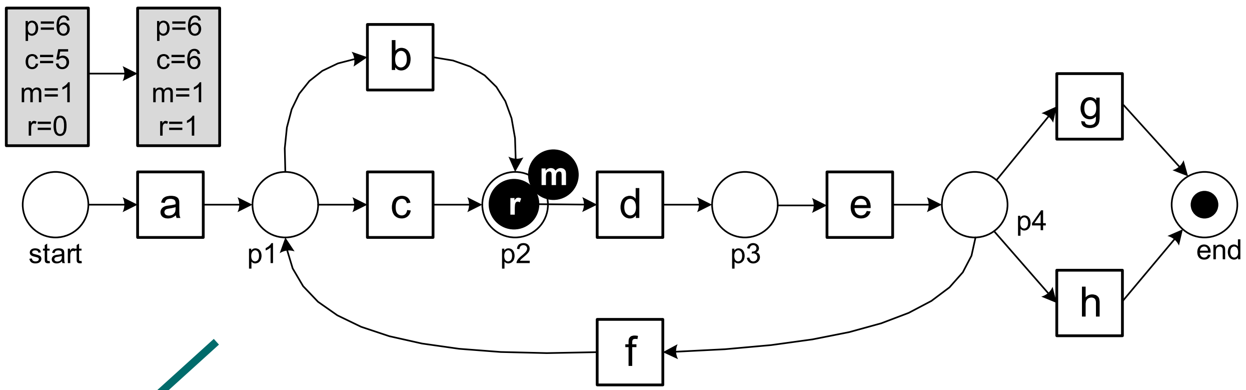
$$\sigma_3 = \langle a, d, c, e, h \rangle$$



A token remains in place p2. Therefore, place p2 is tagged with a **r-tag** and the **r counter is incremented.**

The **r-tag** and **m-tag** highlight the *places where the trace and the model diverge*.

Missing and remaining tokens



How to interpret missing and remainining tokens?
There was a situation in which **d occurred but could not happen** according to the model (**m-tag**) and there was a situation in which **d was supposed to happen** but **did not occur** according to the log (**r-tag**).

$$fitness(\sigma_3, N_2) = \frac{1}{2}\left(1 - \frac{1}{6}\right) + \frac{1}{2}\left(1 - \frac{1}{6}\right) = 0.8333$$

Computing fitness at the log level

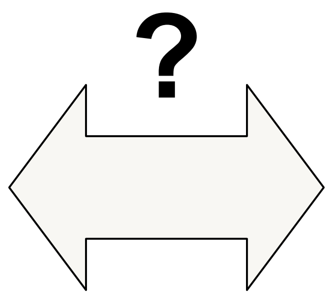
Number of occurrences of a specific trace in the log (e.g., if a trace σ appears 200 times in the log, L(σ) will be equal to 200)

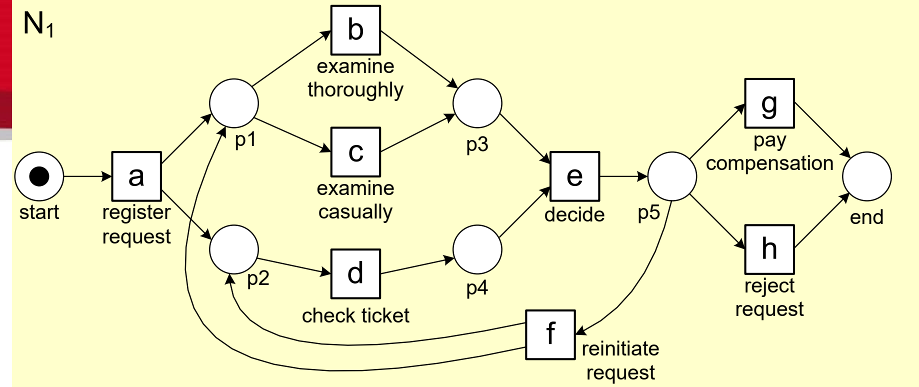
$$fitness(L,N) = \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}}\right) +$$

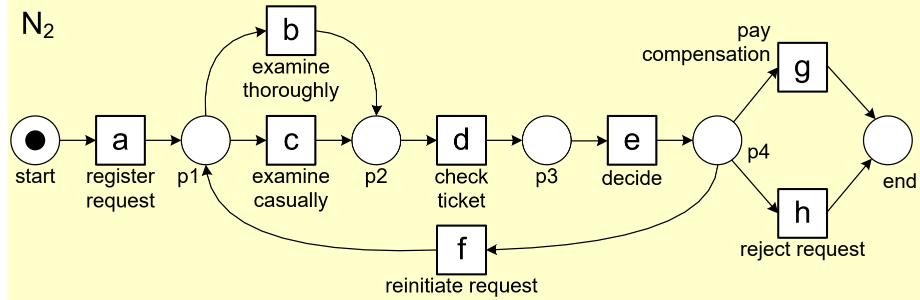
$$\frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}}\right)$$

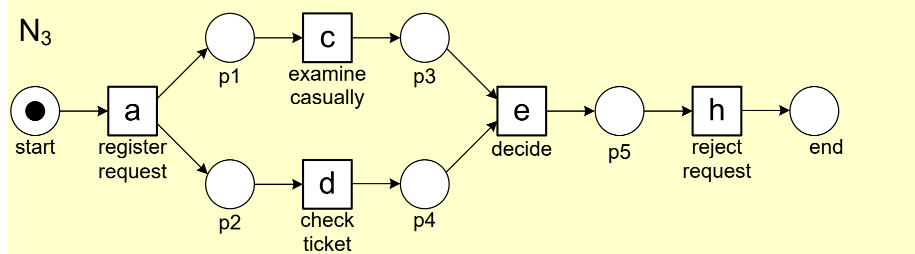
Compute fitness

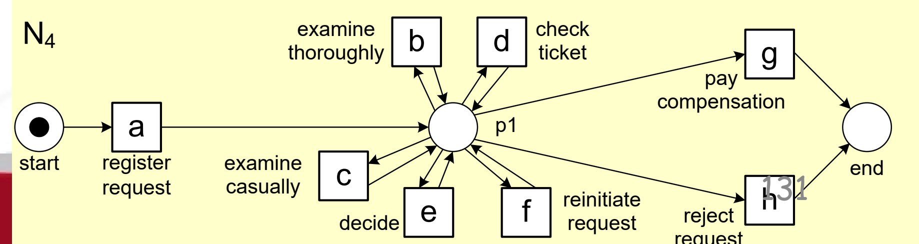
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	





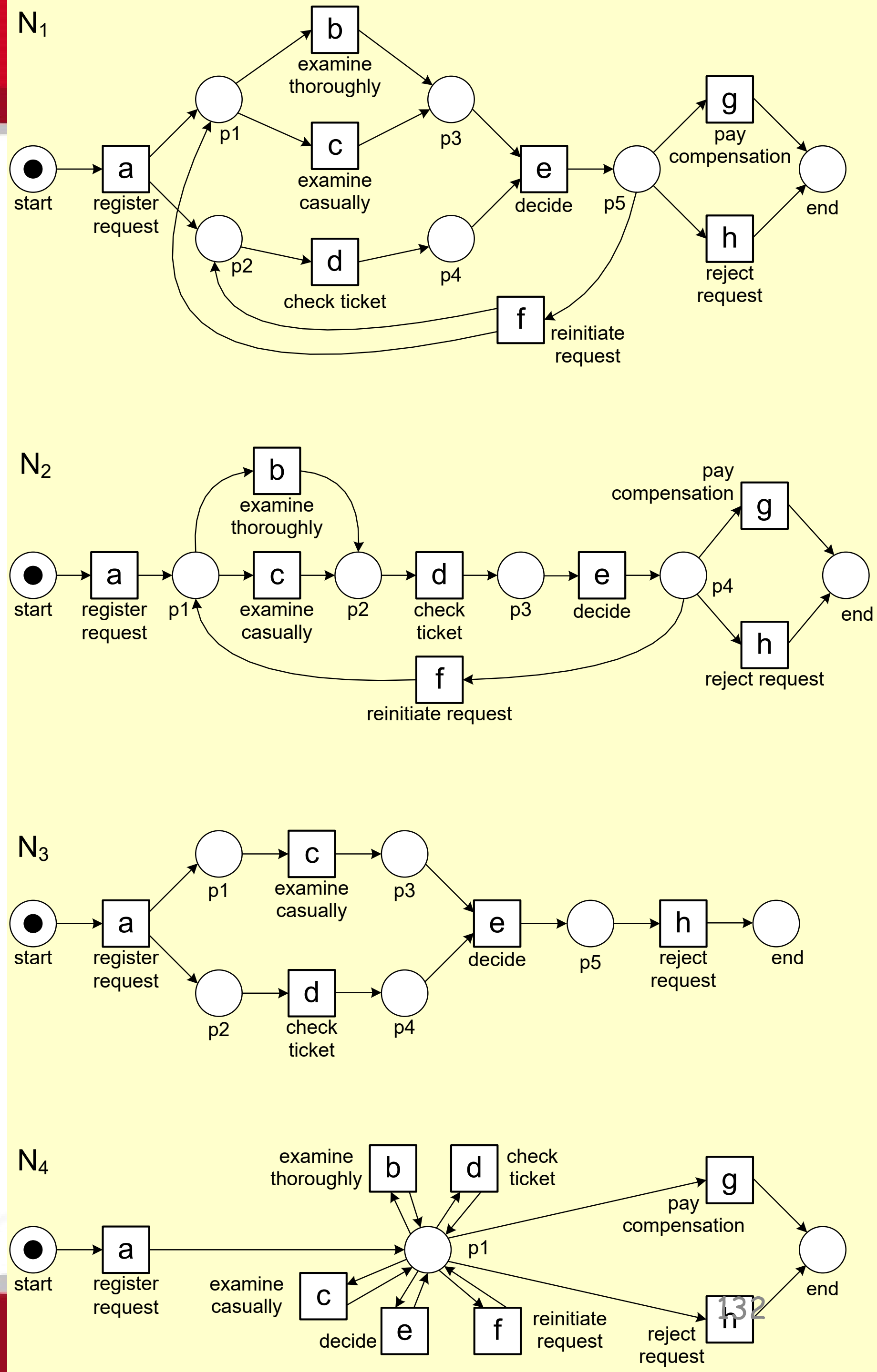






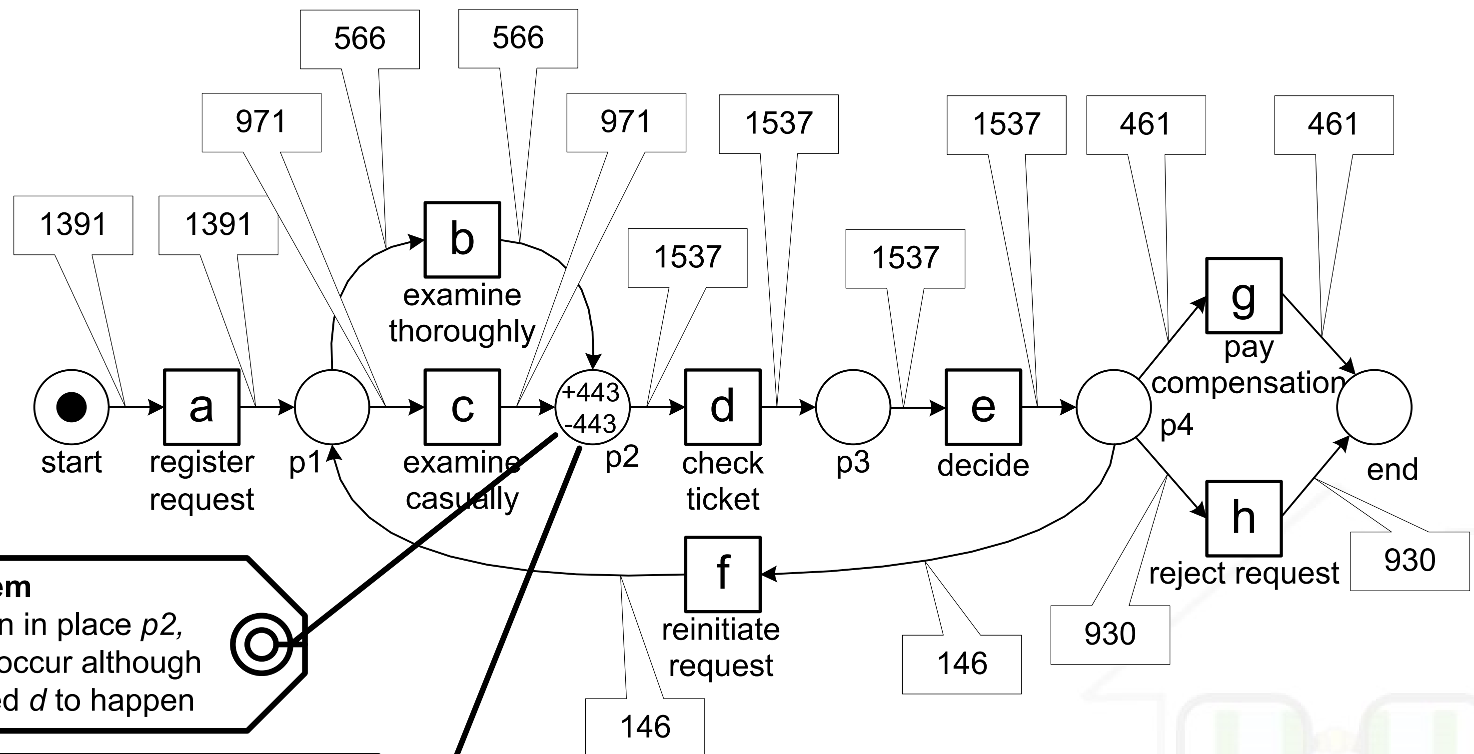
$$fitness(L,N) = \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}}\right) + \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}}\right)$$

$$fitness(L_{full}, N_1) = 1$$
$$fitness(L_{full}, N_2) = 0.9504$$
$$fitness(L_{full}, N_3) = 0.8797$$
$$fitness(L_{full}, N_4) = 1$$



Diagnostics for N_2

$$(fitness(L_{full}, N_2) = 0.9504)$$

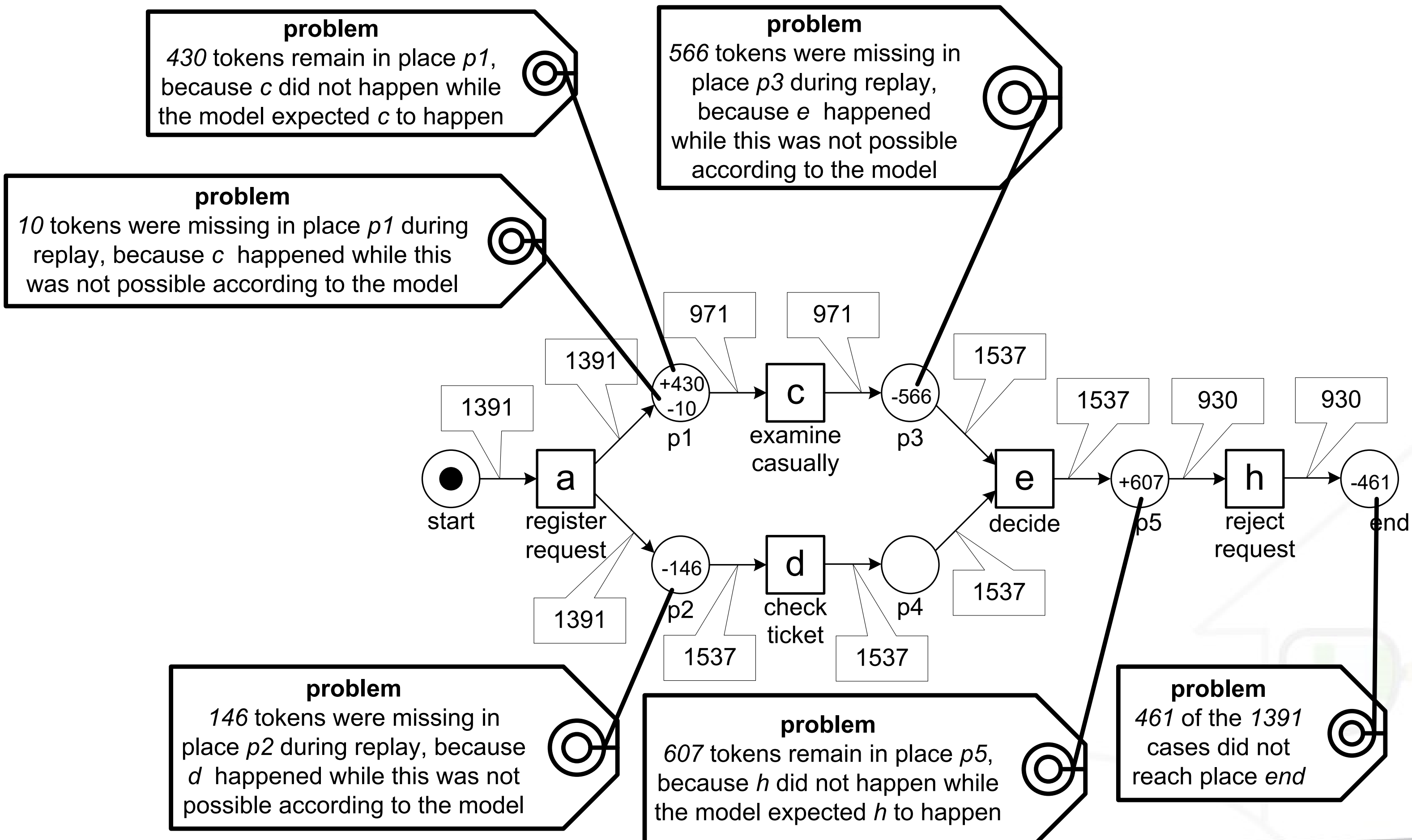


problem
443 tokens remain in place *p2,* because *d* did not occur although the model expected *d* to happen

problem
443 tokens were missing in place *p2* during replay, because *d* happened even though this was not possible according to the model

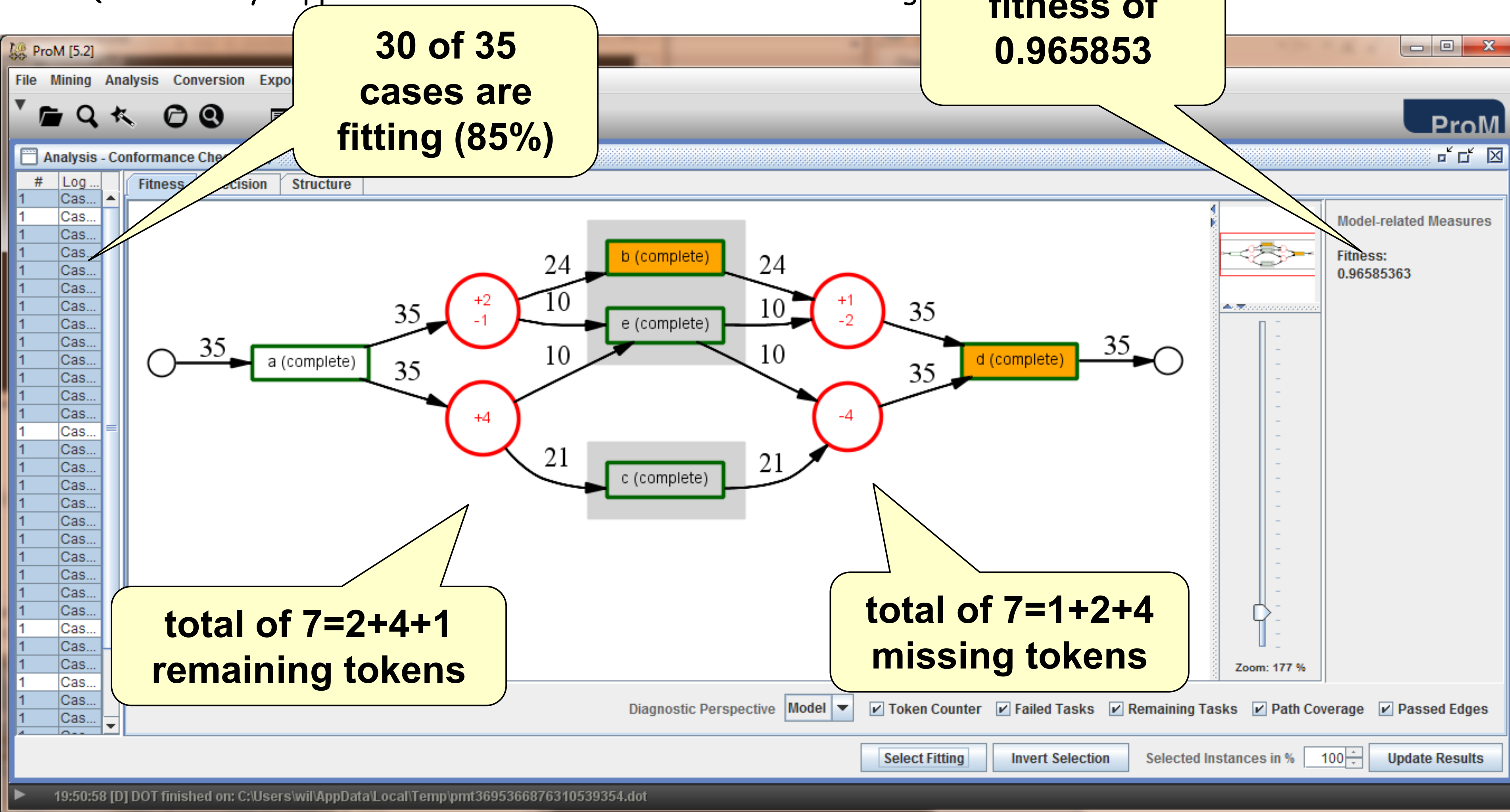
Diagnostics for N_3

$$(fitness(L_{full}, N_3) = 0.8797)$$



ProM 5.2 output

(ProM 6 only supports more advanced conformance checking



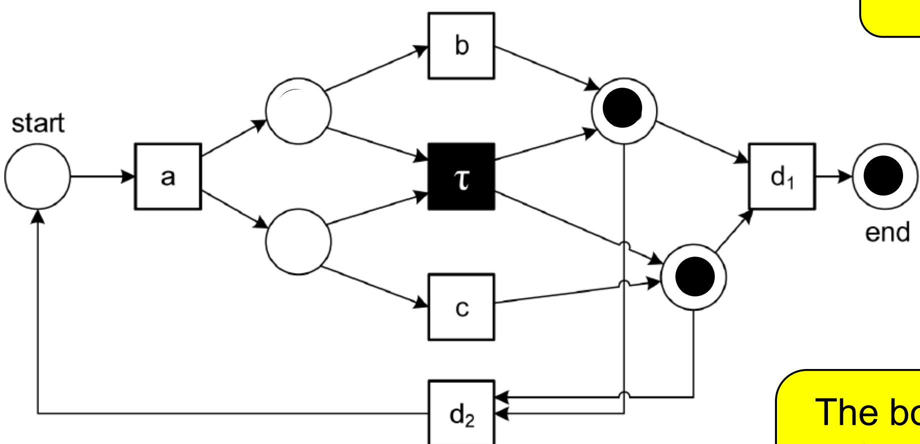
From token-based replay to Trace Alignment

- Using token-based replay we can differentiate between fitting and non-fitting cases.
- However, the token-based approach also has some **drawbacks**.
  - Fitness values tend to be **too high** for extremely problematic event logs.
  - Moreover, if a case does not fit, the approach **does not create a corresponding path** through the model.
  - The approach becomes more complicated when there are **duplicates** or **silent** activities (i.e., activities with a τ label).
- In order to provide better diagnostics, it is required to **relate also non-fitting cases** to the model.
- **Trace Alignment** has been introduced to overcome these limitations.

Trace alignment

- Investigate relations between **moves in the log** and **moves in the model** to establish an alignment between a model N and a trace σ.

Alignment of σ = **<adbc>**



The top row corresponds to σ.

Move in the model only

$$\gamma_1 = \begin{array}{|c|c|c|c|c|} a & d & b & c & \gg \\ \hline a & \gg & b & c & d_1 \end{array}$$

The bottom row corresponds to a path from the initial marking to the final marking.

Move in the log only

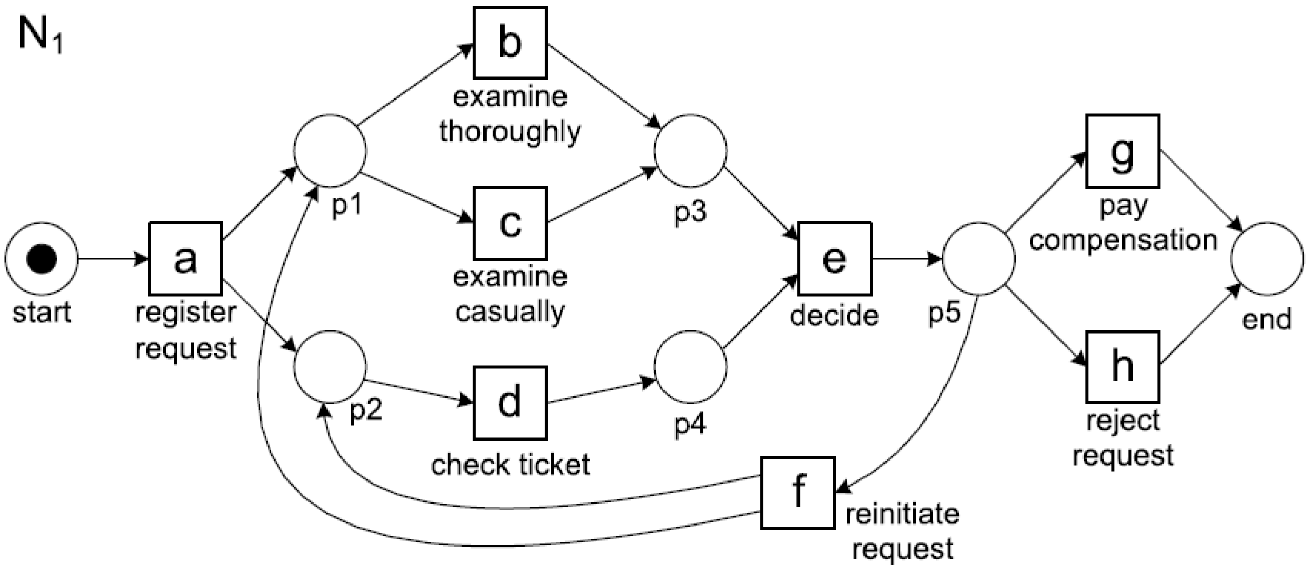
- If a move in the log cannot be mimicked by the model and vice-versa, such “no moves" are denoted by >> (and may have a **cost**).

An example of alignment

a	b	d	e	g
a	b	d	e	g

The trace perfectly fits the model

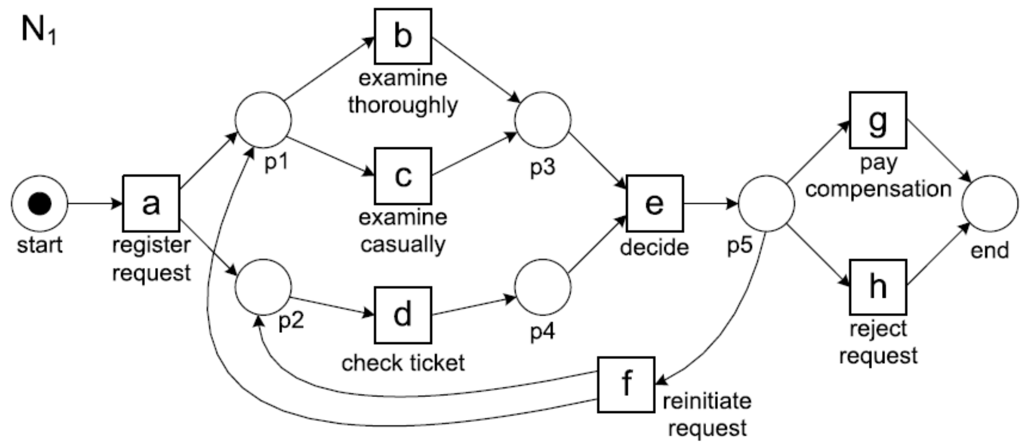
N_1



#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Several possible alignments

<abdeg>



a	b	d	e	g
a	b	d	e	g

a	b	»	d	e	g
a	»	c	d	e	g

a	b	d	e	g	»	»	»	»	»
»	»	»	»	»	a	c	d	e	g

Moves have costs

- Standard cost function:
  - c(x,») = 1
  - c(»,y) = 1
  - c(x,y) = 0, if x=y
  - c(x,y) = ∞, if x≠y

…	a	…
…	»	…

…	»	…
…	a	…

…	a	…
…	a	…

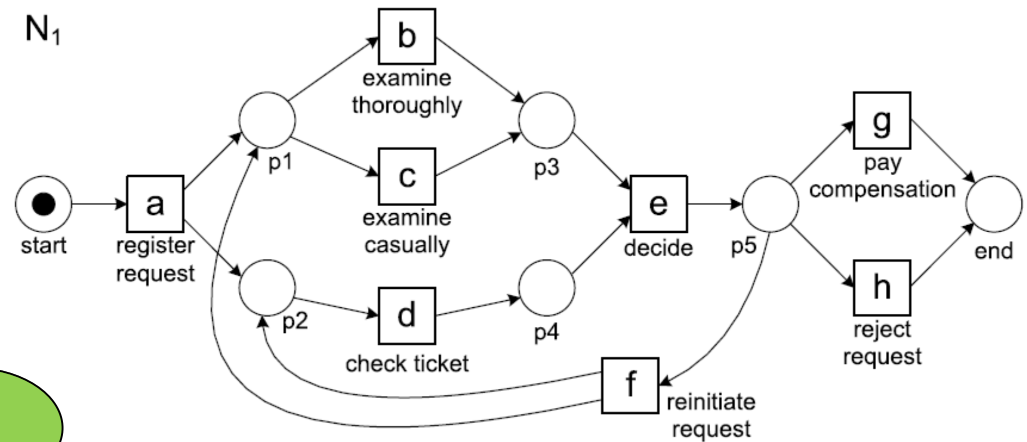
…	b	…
…	a	…

OPTIMAL ALIGNMENT
alignment with minimum deviation cost

Any cost structure is possible!

Optimal alignments

<abdeg>



optimal

a	b	d	e	g
a	b	d	e	g

0

a	b	»	d	e	g
a	»	c	d	e	g

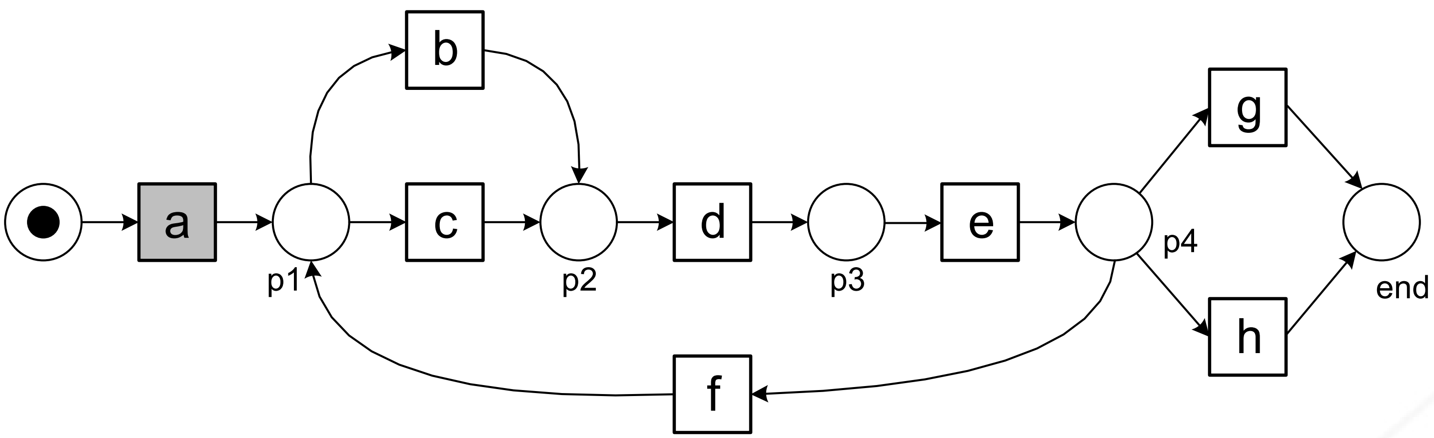
2

a	b	d	e	g	»	»	»	»	»
»	»	»	»	»	a	c	d	e	g

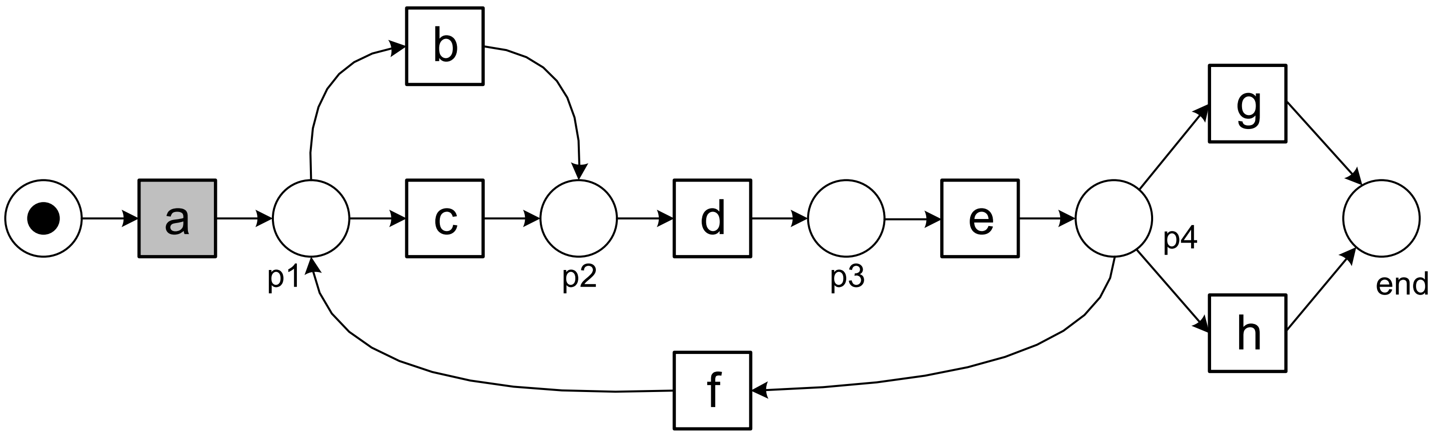
10

Exercise

- Find at least two optimal alignments between the trace <a,d,b,e,h> and the model N_2



Solution



a	»	d	b	e	h
a	b	d	»	e	h

a	d	b	»	e	h
a	»	b	d	e	h

Fitness based on alignments

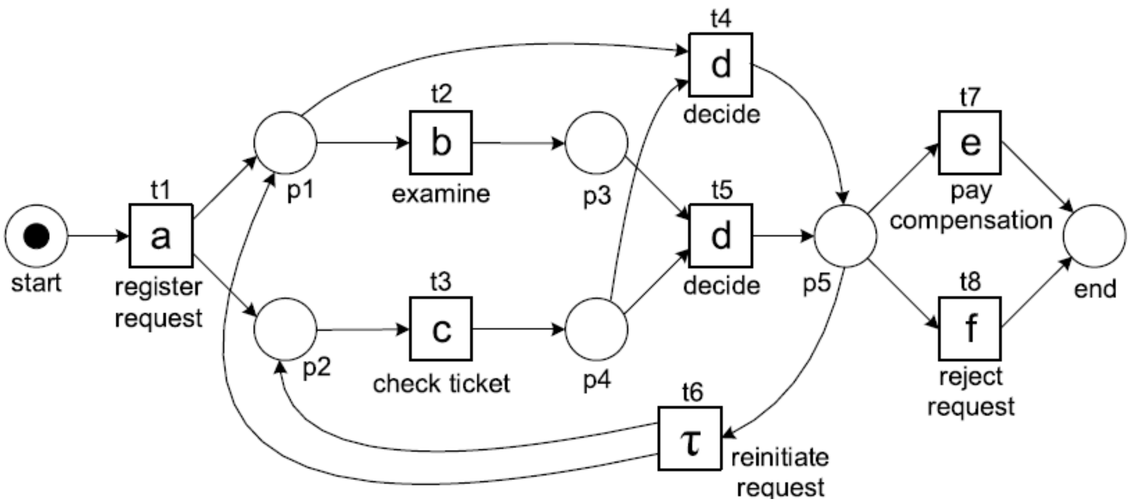
Cost of the **optimal alignment** of the trace σ

$$fitness(\sigma, N) = 1 - \frac{\delta(\lambda_{opt}^{N}(\sigma))}{\delta(\lambda_{worst}^{N}(\sigma))}$$

Cost of the **worst-case alignment** where there are no sinchronous moves and moves in model and log only.

In a worst-case alignment:
(i) all events in trace *σ* are converted to log moves and (ii) a shortest path from an initial state to a final state of the model is added as a sequence of model moves

Example



$$\gamma_{5,2a} = \begin{array}{|c|c|c|c|c|} a & b & \gg & d & f \\ \hline a & b & c & d & f \\ t1 & t2 & t3 & t5 & t8 \end{array}$$

Cost of optimal alignment: 1

Cost of worst-case alignment: 8

$$\gamma_{5,2w} = \begin{array}{|c|c|c|c|c|c|c|c|} a & b & d & f & \gg & \gg & \gg & \gg \\ \hline \gg & \gg & \gg & \gg & a & c & d & f \\ & & & & t1 & t3 & t4 & t8 \end{array}$$

$$fitness(\sigma_2, N_5) = 1 - \frac{1}{8} = 0.875$$

Fitness for the entire log

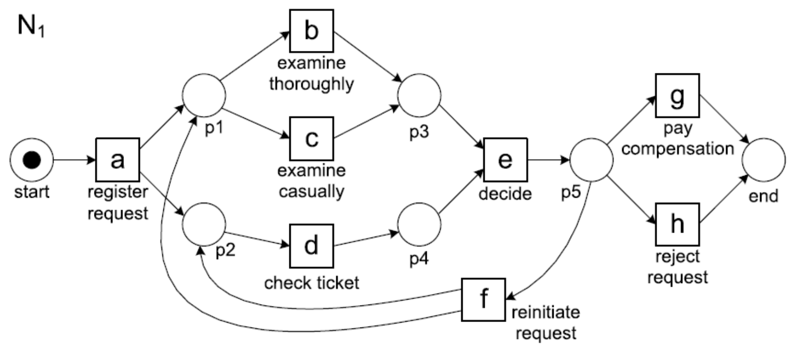
This is the sum of all costs when replaying the entire event log using optimal alignments

Number of occurrences of a specific trace in the log (e.g., if a trace σ appears 200 times in the log, L(σ) will be equal to 200)

$$fitness(L,N) = 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda_{opt}^{N}(\sigma))}{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda_{worst}^{N}(\sigma))}$$

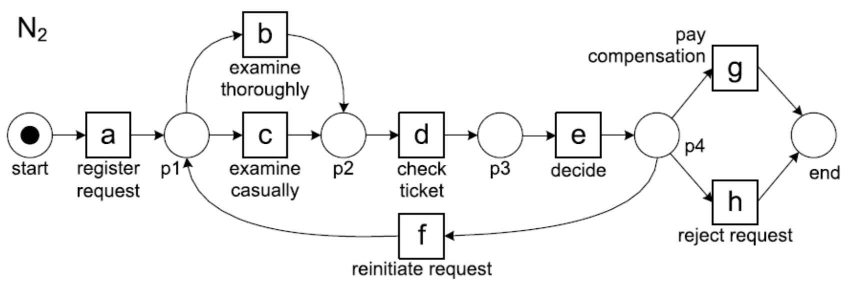
It is divided by the sum of the cost of all worst-case scenarios to obtain a normalized fitness value

Event-based vs Alignment-based fitness



$$fitness(L_{full}, N_1) = 1$$

Alignment-based fitness is **lower** because in the log there are several cases where d occurs multple times before b or c (within the same case)

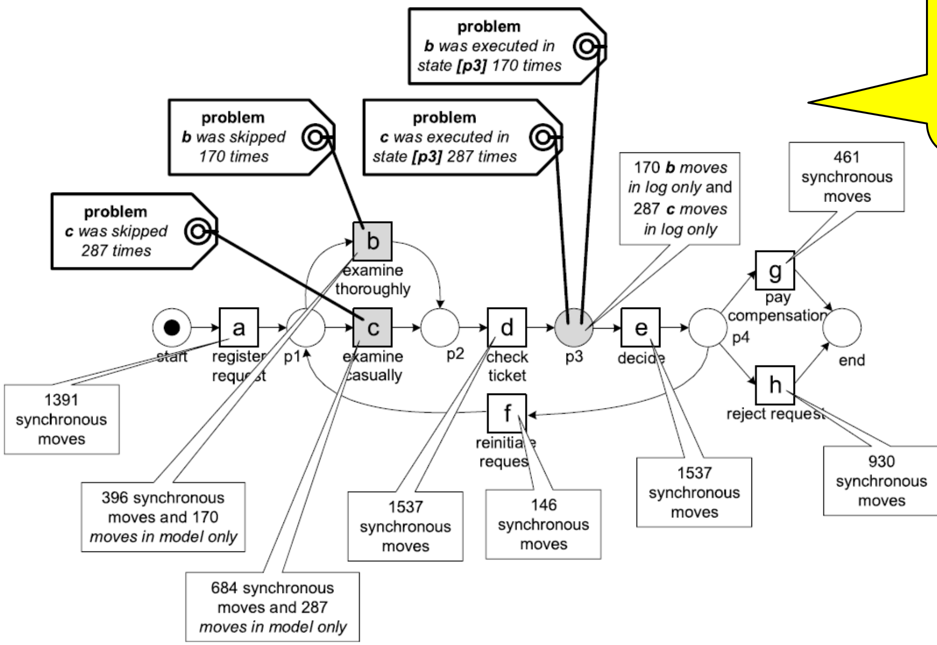


$$fitness(L_{full}, N_2) = 1 - \frac{914}{14494} = 0.936939$$

$$fitness(L_{full}, N_2) = 0.9504$$

Event-based fitness is less precise!
A second or third misalignment of d in the same case is not detected due to a token remaining from the first misalignment.

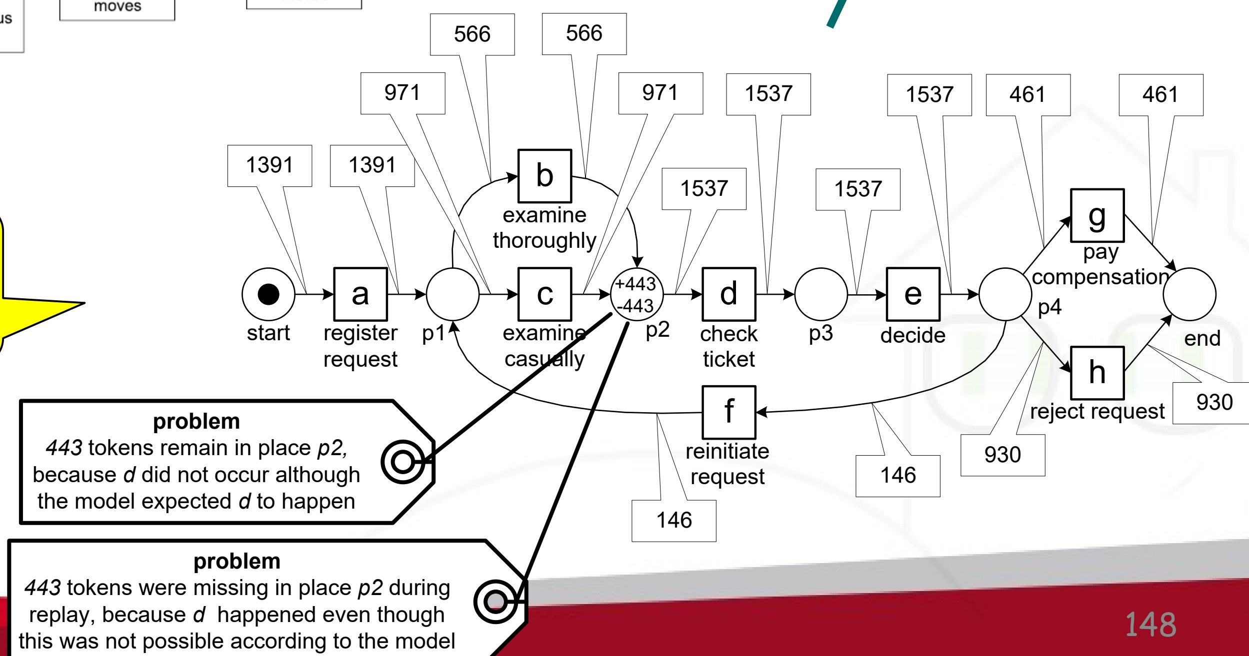
Diagnostic for N_2



In the alignment based diagnostics:
- b was executed 170 times after d
- c was executed 287 times after d
- d was executed 170 + 287 = **457 times before b or c**.

The difference between 443 and the correct 457 is caused by tokens remaining in place p2 after the first iteration.

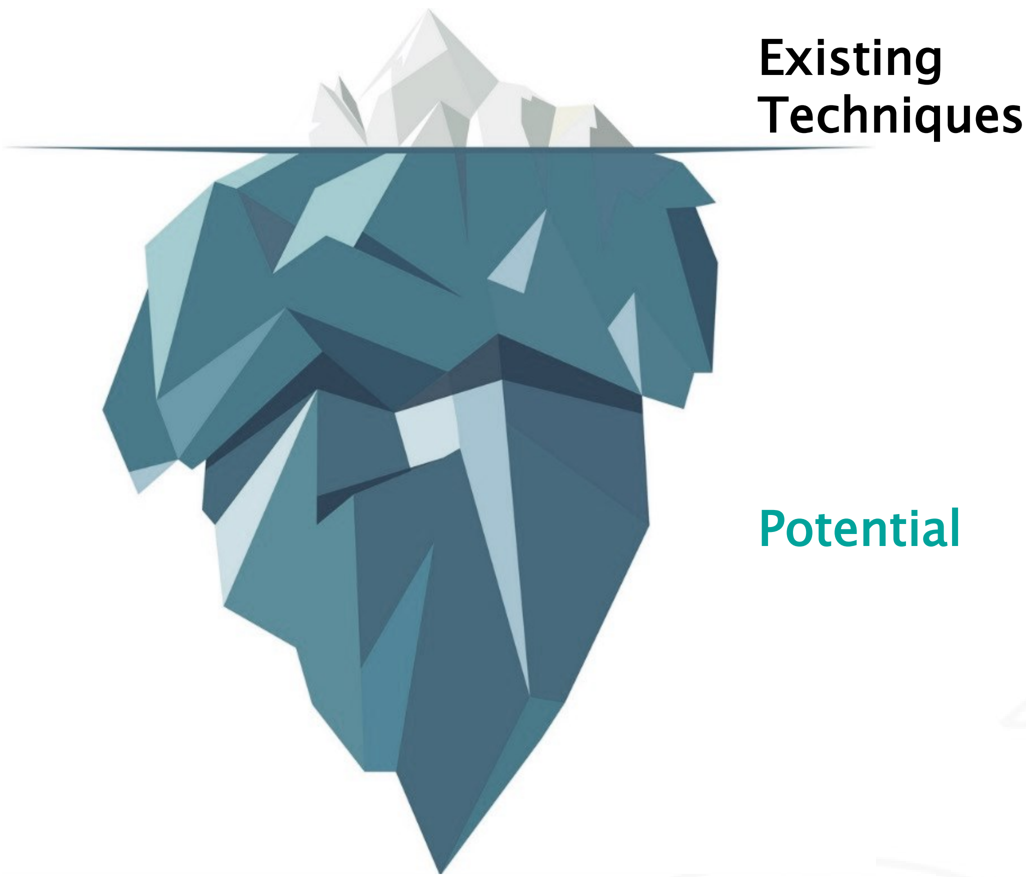
In the token replay diagnostics it is suggested that d was executed **443 times** before b or c.



Advantages of Trace Alignment

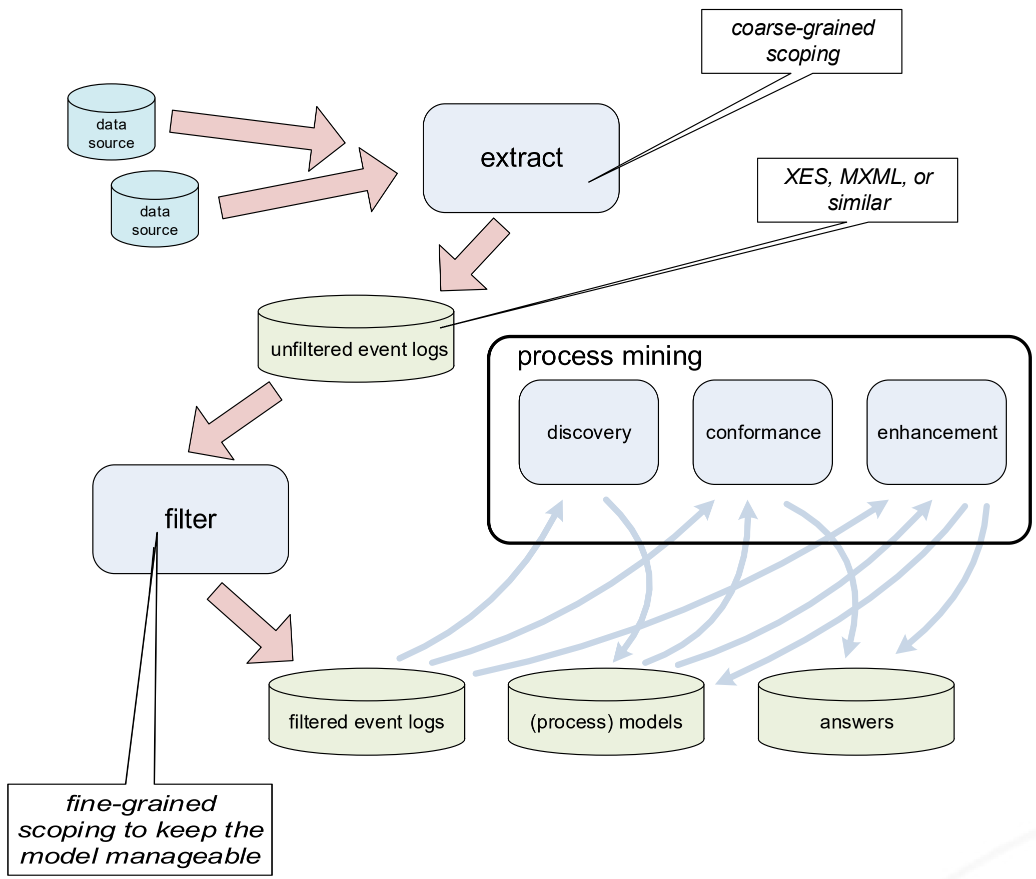
- Observed behavior is **directly related** to modeled behavior.
- **Highly flexible** (any cost structure).
- **Detailed diagnostics**: alignments explain where deviations occur and which deviations occur. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- **More accurate diagnostics**: Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations).
- Alignments:
  a) Are globally optimal
  b) Are robust to label duplication
  c) Are robust to routing transitions
  d) Provide a true execution of the model
- Existing implementation in ProM (try the plugin: "Replay a Log on Petri Net for Conformance Analysis")

Potential of process mining



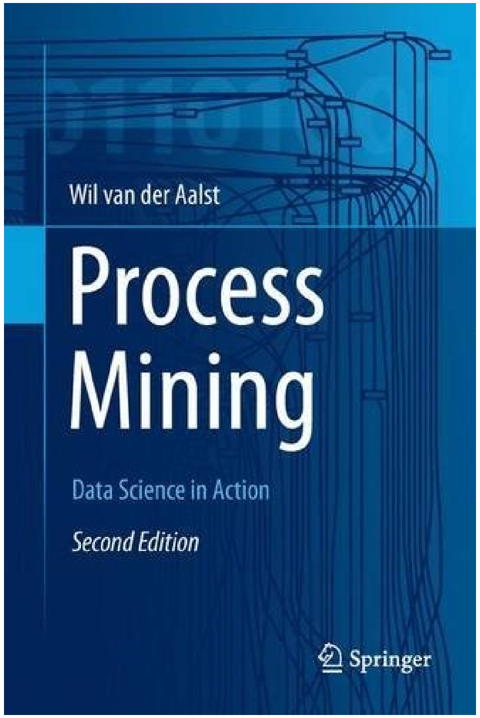
Main challenge - *Extracting event logs*

Starting point is the raw data hidden in all kinds of data sources. A data source may be a text file, an Excel spreadsheet, a transaction log, etc.



- Correlation: Events in an event log are grouped per trace. This simple requirement can be quite challenging as it requires **event correlation**, i.e., events need to be related to each other.
- Timestamps: Events need to be **ordered per trace**. Typical problems: only dates, different clocks, delayed logging.
- Snapshots. Traces may have a **lifetime extending beyond the recorded period**, e.g., a trace was started before the beginning of the event log.
- Scoping. How to decide **which events to incorporate?** An event log refers to one process consisting of many activities…
- Granularity: the events in the event log are at a **different level of granularity** than the activities relevant for end users.
- Life cycle: Activities may have a **life cycle** (assign, start, abort, complete, etc.). How to deal with it?

Main Reference



Book:
W.M.P. van der Aaalst
Process Mining. Data Science in Action
Springer, 2° edition, 2016

Slides:
http://www.processmining.org/

Online course:
https://www.coursera.org/learn/process-mining