Discovering Petri Nets: Evidence-Based Business Process Management

Wil van der Aalst

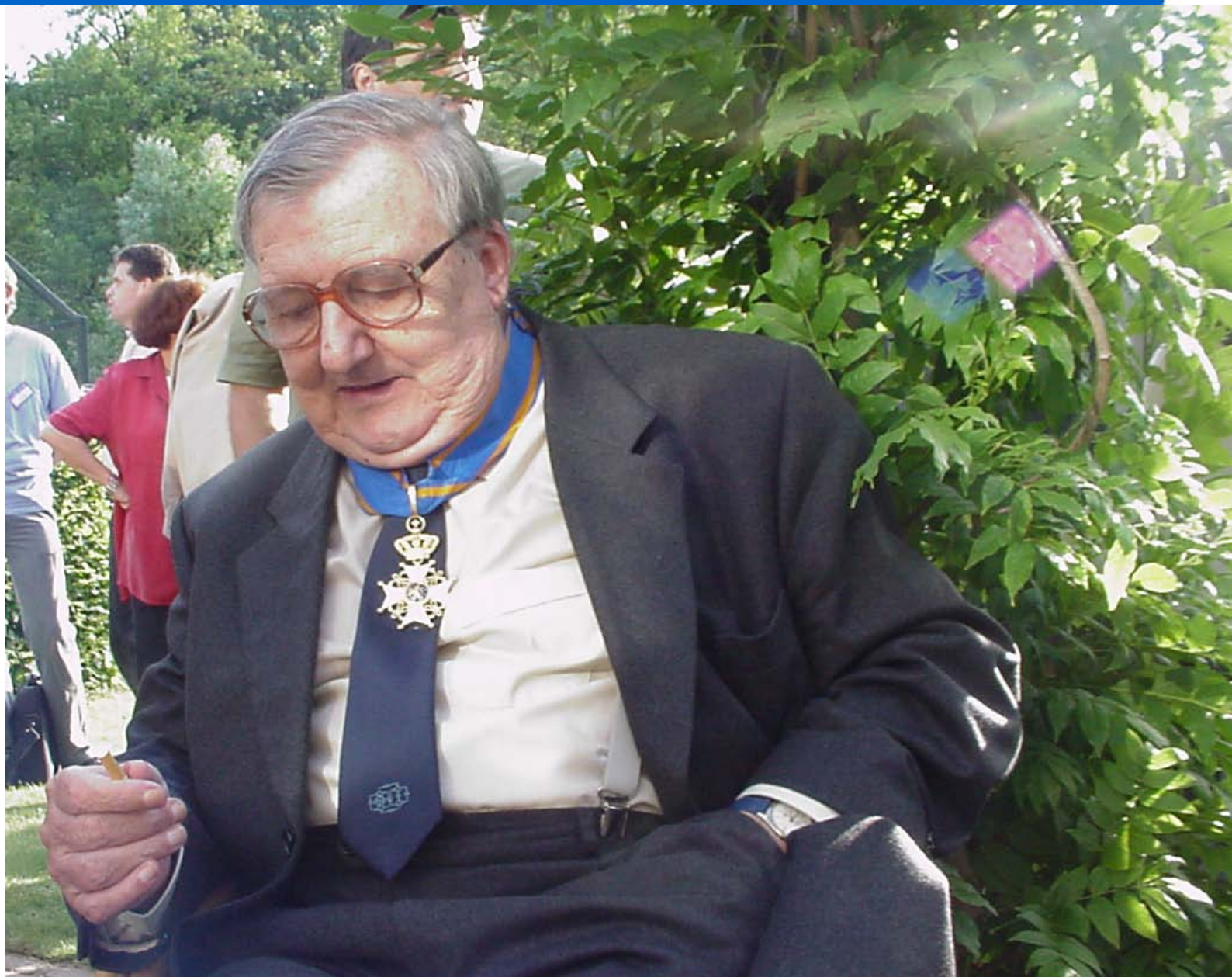
Carl Adam Petri Memorial Symposium
Humboldt-Universität zu Berlin
February 4, 2011

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Commandeur in de Orde van de Nederlandse Leeuw (Petri net conference 2003 in Eindhoven)

About 50 people are Commandeur in de Orde van de Nederlandse Leeuw, e.g., Wim Duisenberg, Gerard 't Hooft, Harry Mulisch, Gerard Reve, Joop den Uyl

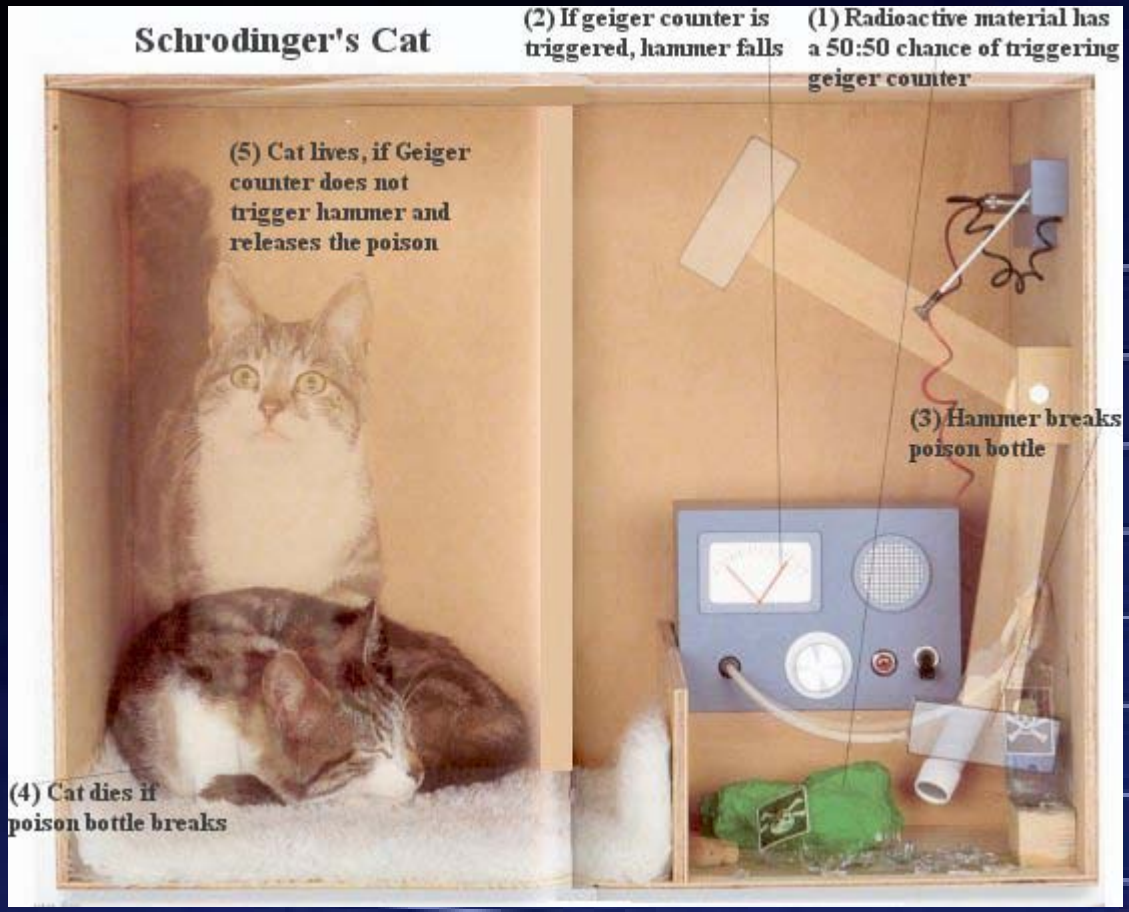


Carl Adam Petri (12 July 1926 – 2 July 2010)

- **Started a new subfield of computer science.**

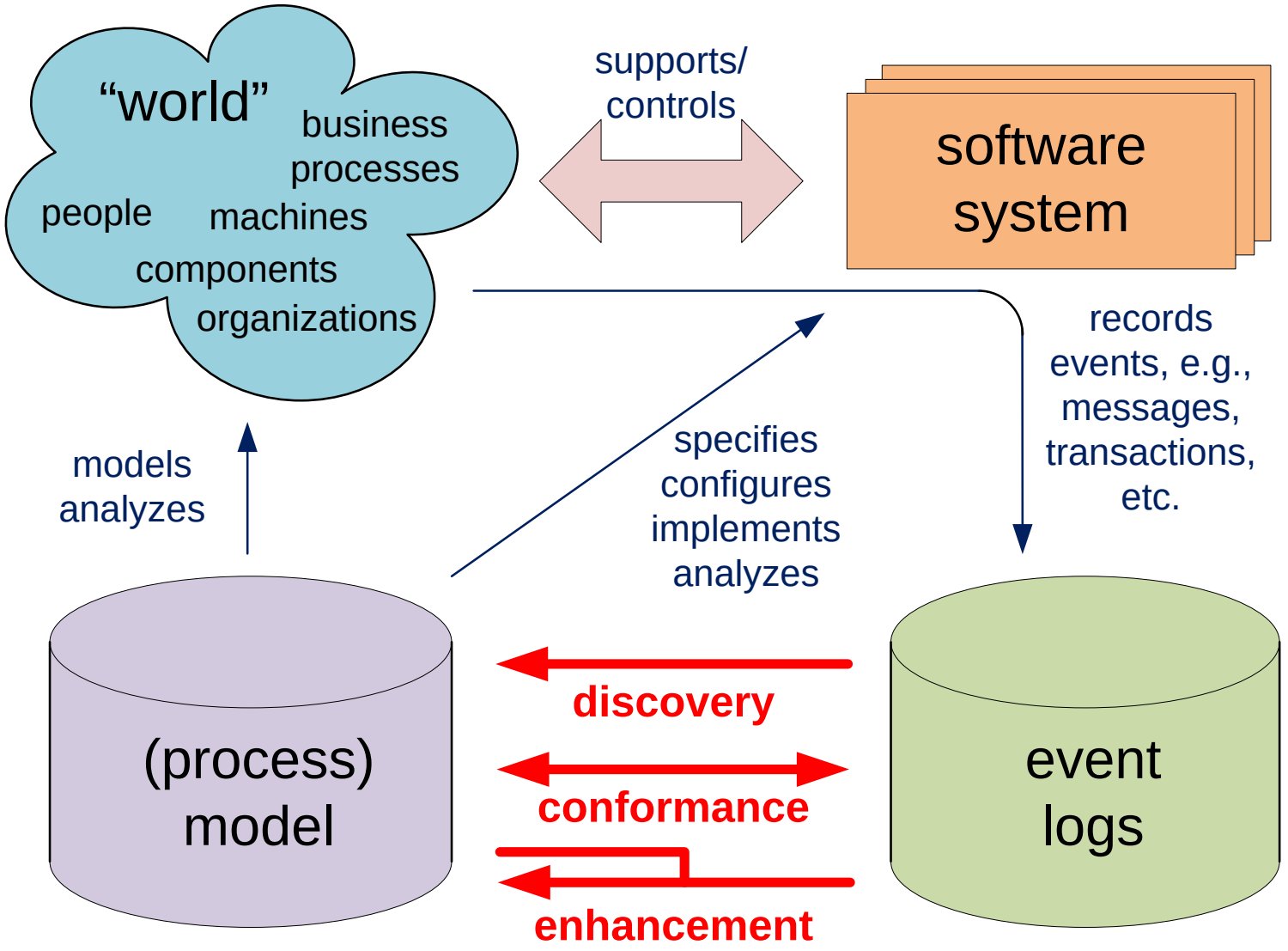
  "Much of what I have been saying was already well understood in the sixties by Carl Adam Petri, who pioneered the scientific modeling of discrete concurrent systems. Petri's work has a secure place at the root of concurrency theory." (Robin Milner 1991)

- **Guiding principles:**
  - **Concurrency as a starting point rather than an afterthought (locality of actions).**
  - **Formalism should be consistent with the laws of physics.**
- **Researchers in the growing Business Process Management (BPM) community use these principles.**
  - **Modeling languages have Petri-net like semantics.**
  - **More and more emphasis on empirical validation.**

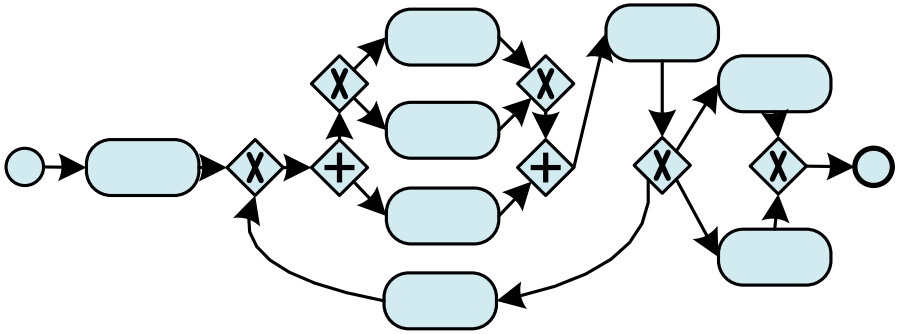


Source: UC Berkeley Research Project, How Much Information?, 2003; IDC, Disk Storage System Quarterly Tracker (as of 2006)

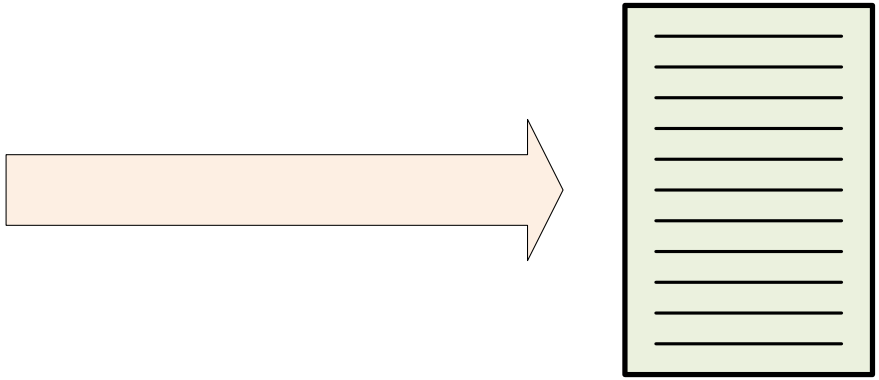
Process Mining



Play-Out

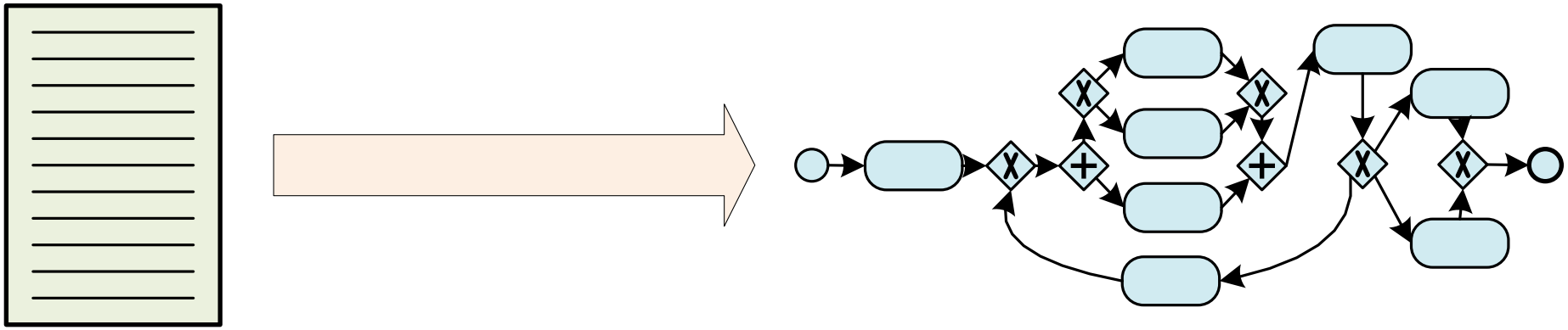


process model



event log

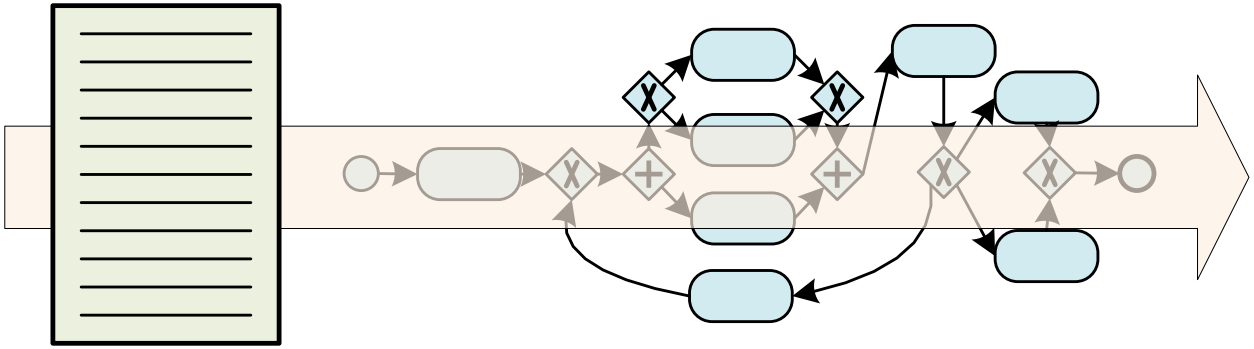
Play-In



event log

process model

Replay



event log

process model

- extended model showing times, frequencies, etc.
- diagnostics
- predictions
- recommendations

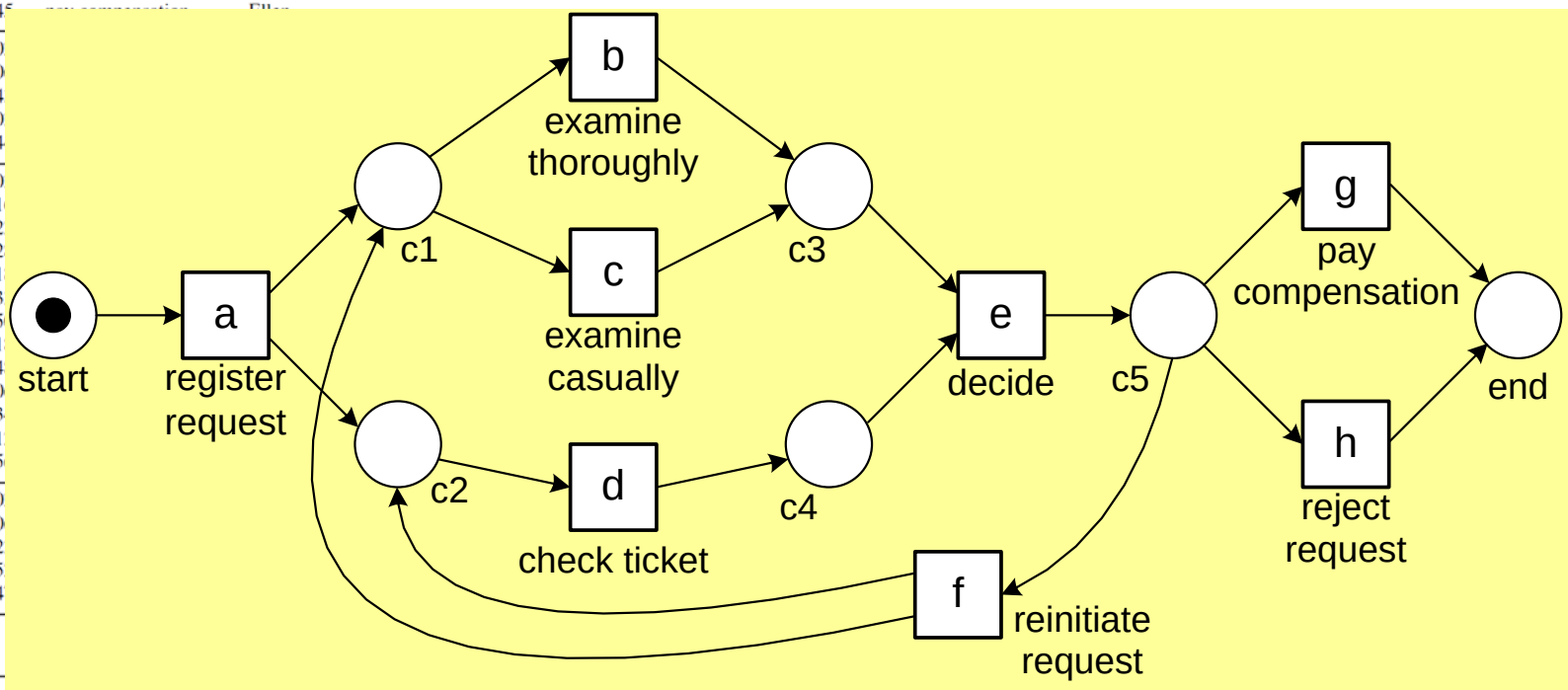
Simple example log

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	...

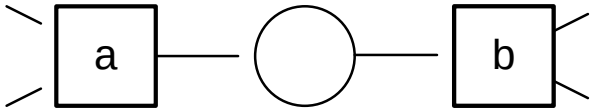
Simple control-flow discovery (Play-In)

case id	event id	properties		
		timestamp	activity	resource
1	35654423	30-12-2010:11.02	register request	Pete
	35654424	31-12-2010:10.06	examine thoroughly	Sue
	35654425	05-01-2011:15.12	check ticket	Mike
	35654426	06-01-2011:11.18	decide	Sara
	35654427	07-01-2011:14.24	reject request	Pete
2	35654483	30-12-2010:11.32	register request	Mike
	35654485	30-12-2010:12.12	check ticket	Mike
	35654487	30-12-2010:14.16	examine casually	Pete
	35654488	05-01-2011:11.22	decide	Sara
	35654489	08-01-2011:12.05	pay compensation	Ellen
3	35654521	30-12-2010:14.32	register request	Pete
	35654522	30-12-2010:15.06	examine casually	Mike
	35654524	30-12-2010:16.34	check ticket	Ellen
	35654525	06-01-2011:09.18	decide	Sara
	35654526	06-01-2011:12.18	reinitiate request	Sara
	35654527	06-01-2011:13.06	examine thoroughly	Sean
	35654530	08-01-2011:11.43	check ticket	Pete
	35654531	09-01-2011:09.55	decide	Sara
	35654533	15-01-2011:10.45	pay compensation	Ellen
4	35654641	06-01-2011:15.0		
	35654643	07-01-2011:12.0		
	35654644	08-01-2011:14.4		
	35654645	09-01-2011:12.0		
	35654647	12-01-2011:15.4		
5	35654711	06-01-2011:09.0		
	35654712	07-01-2011:10.1		
	35654714	08-01-2011:11.2		
	35654715	10-01-2011:13.2		
	35654716	11-01-2011:16.1		
	35654718	14-01-2011:14.3		
	35654719	16-01-2011:15.5		
	35654720	19-01-2011:11.1		
	35654721	20-01-2011:12.4		
	35654722	21-01-2011:09.0		
	35654724	21-01-2011:11.3		
	35654725	23-01-2011:13.1		
	35654726	24-01-2011:14.5		
6	35654871	06-01-2011:15.0		
	35654873	06-01-2011:16.0		
	35654874	07-01-2011:16.2		
	35654875	07-01-2011:16.5		
	35654877	16-01-2011:11.4		
...	...	...		

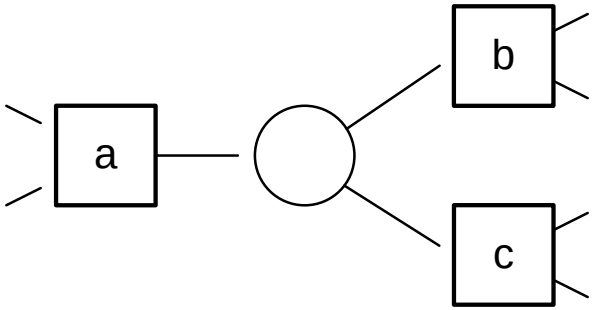
case id	trace
1	$\langle a,b,d,e,h \rangle$
2	$\langle a,d,c,e,g \rangle$
3	$\langle a,c,d,e,f,b,d,e,g \rangle$
4	$\langle a,d,b,e,h \rangle$
5	$\langle a,c,d,e,f,d,c,e,f,b,d,e,h \rangle$
6	$\langle a,c,d,e,g \rangle$



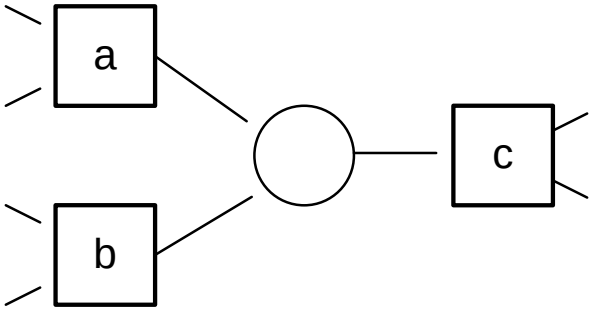
Alpha algorithm



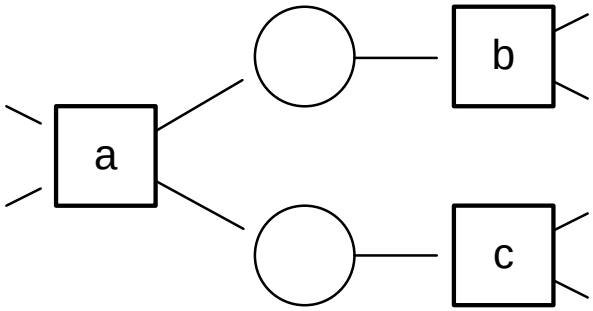
(a) sequence pattern: a→b



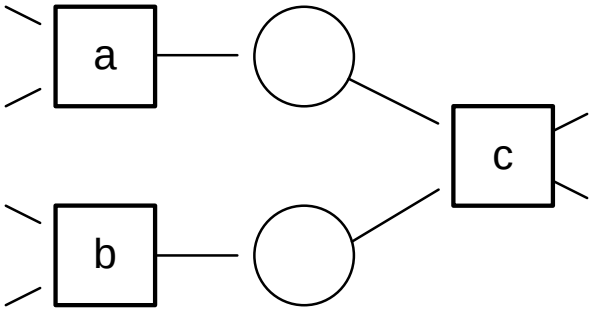
(b) XOR-split pattern:
a→b, a→c, and b#c



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

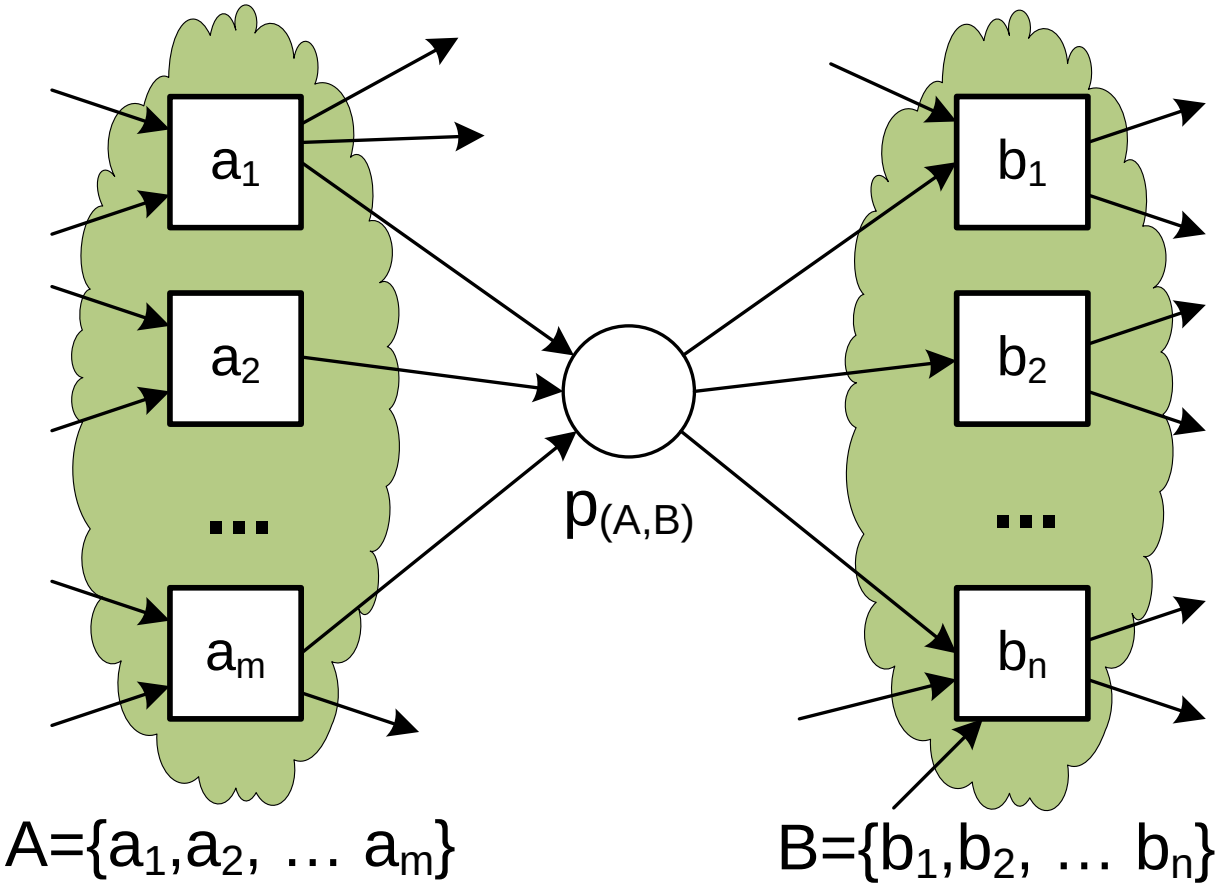


(d) AND-split pattern:
a→b, a→c, and b||c



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

Discovering places



$$X_L = \{(A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\ a \rightarrow_L b \wedge \forall_{a_1,a_2 \in A}\ a_1 \#_L a_2 \wedge \forall_{b_1,b_2 \in B}\ b_1 \#_L b_2\},$$

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

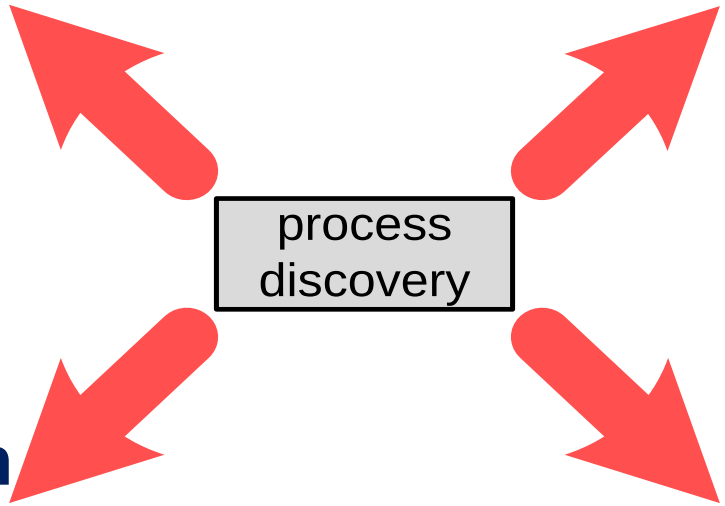
Challenge

“able to replay event log”

fitness

“Occam’s razor”

simplicity



generalization

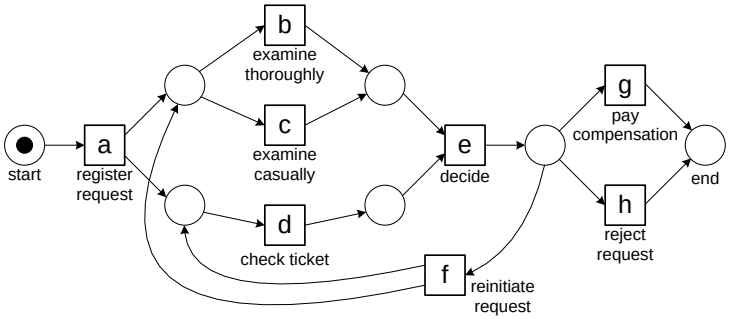
“not overfitting the log”

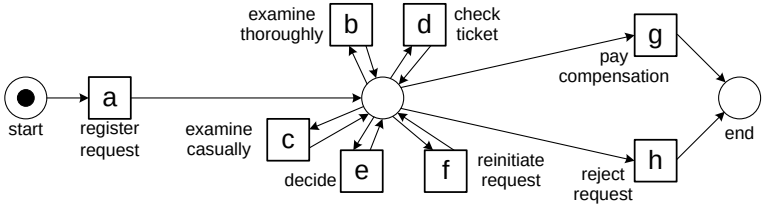
precision

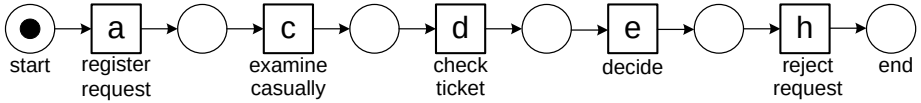
“not underfitting the log”

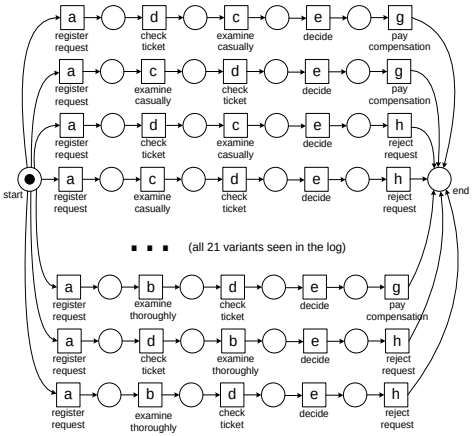
Example

#	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	



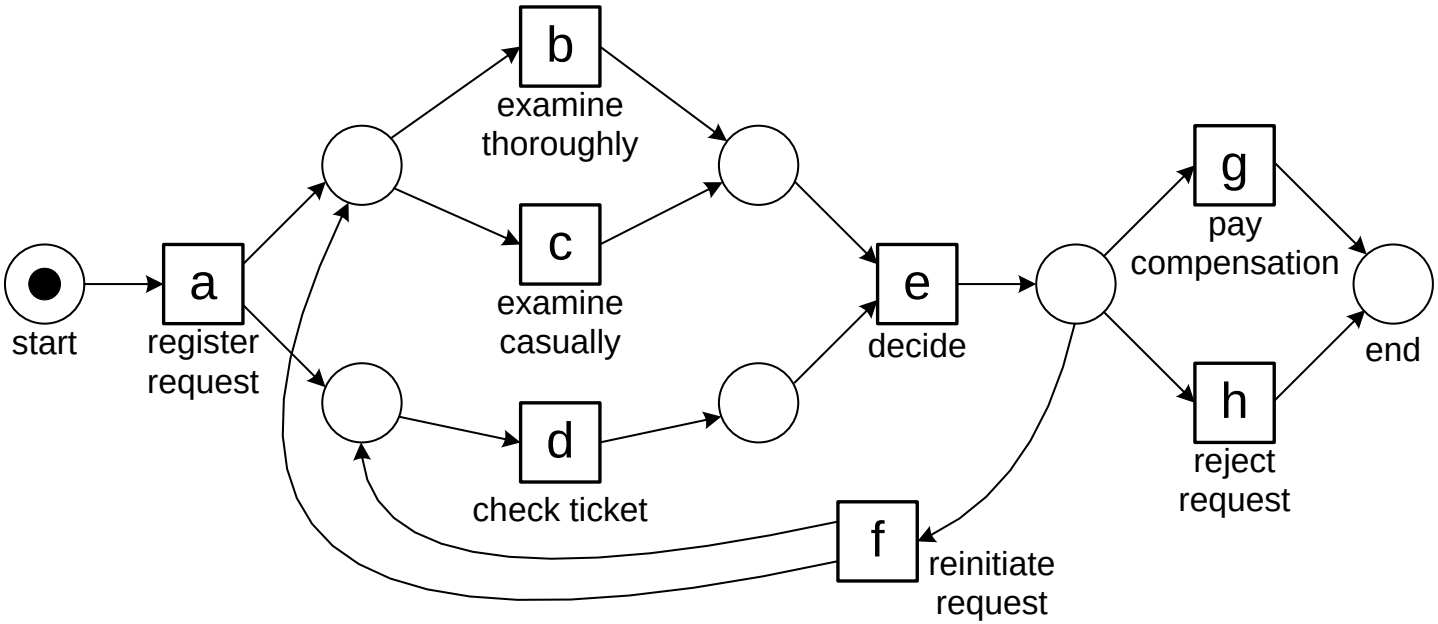






Model 1: Seems right

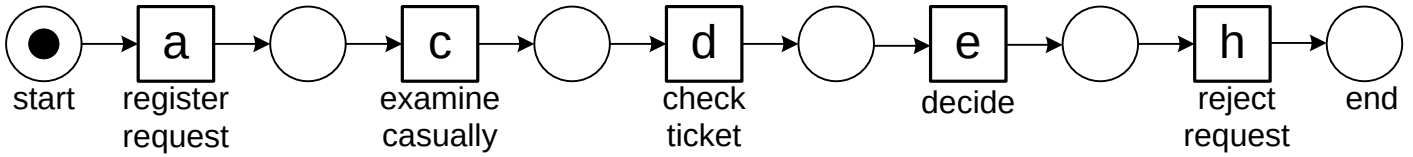
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	



fitness = +
precision = +
generalization = +
simplicity = +

Model 2: Simple and precise, but …

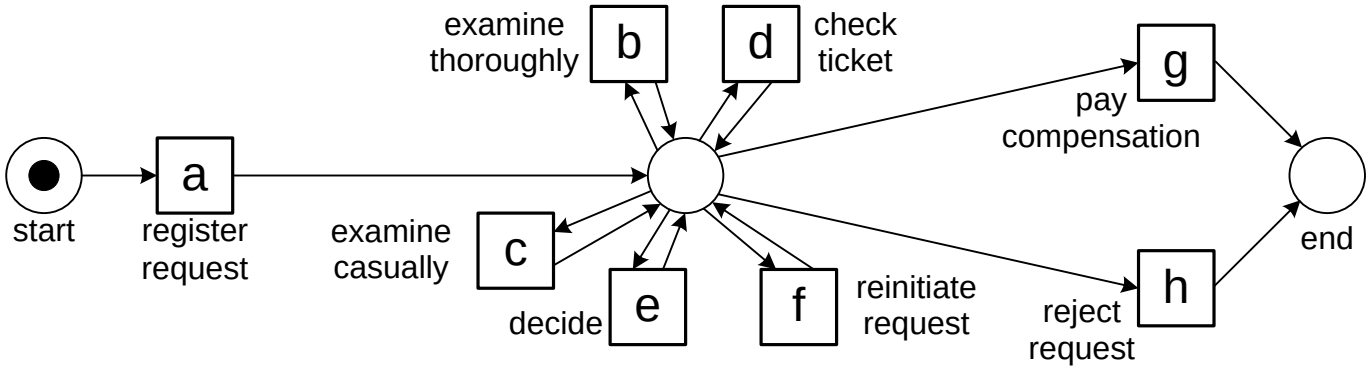
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	



fitness = -
precision = +
generalization = -
simplicity = +

Model 3: Simple but underfitting

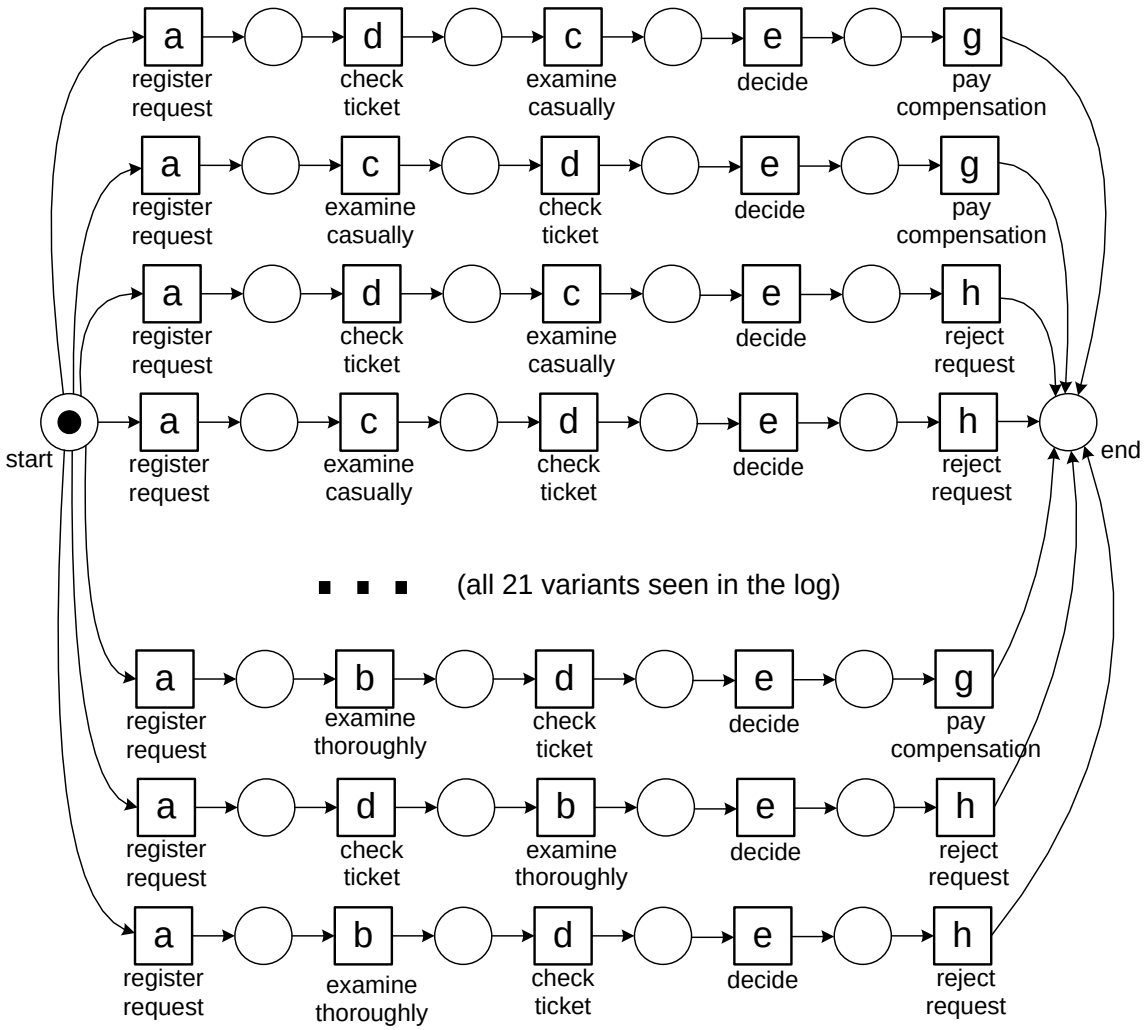
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	



fitness = +
precision = -
generalization = +
simplicity = +

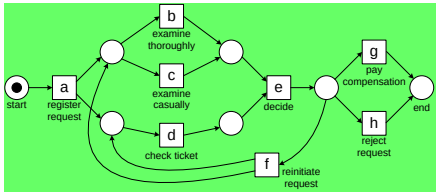
Model 4: Complex and overfitting

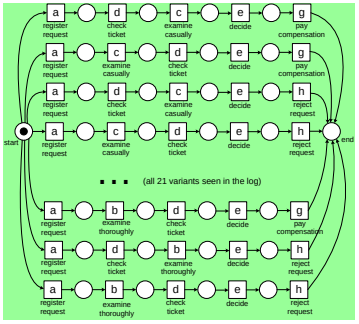
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

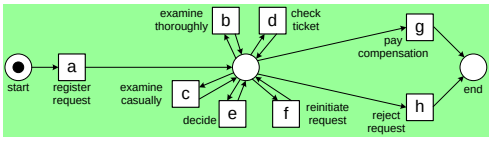


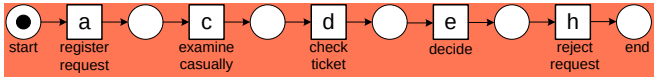
fitness = +, precision = +,
generalization = -, simplicity = -

Overview

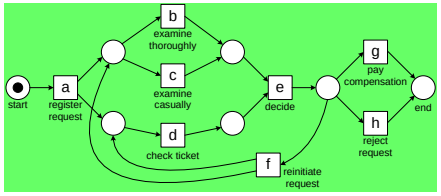


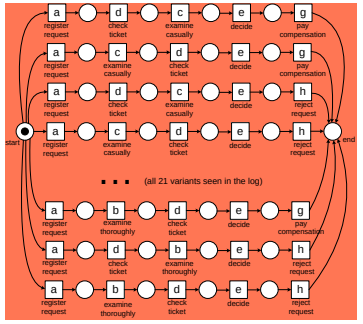


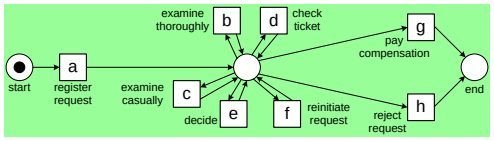


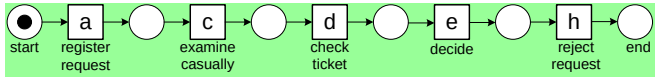


"able to replay event log"









"Occam's razor"

fitness

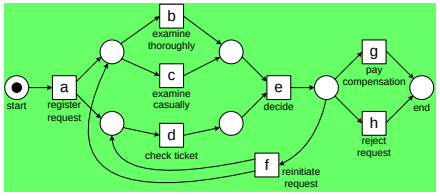
simplicity

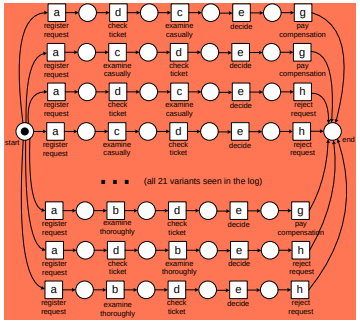
process discovery

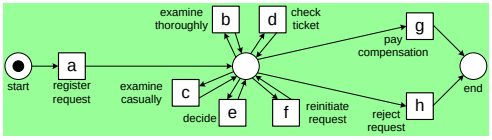
generalization

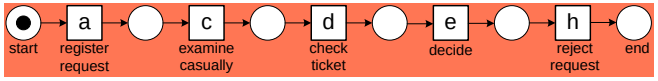
precision

"not overfitting the log"

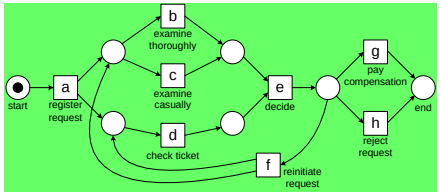


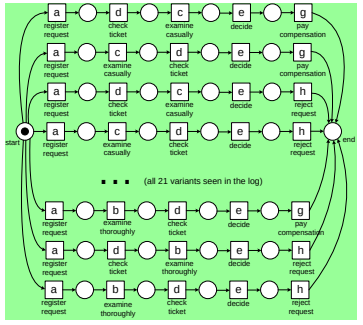


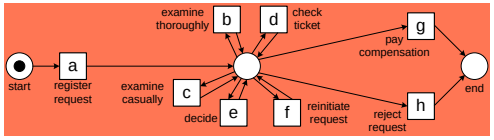


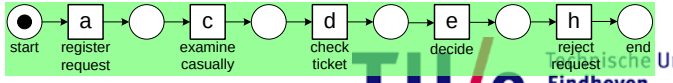


"not underfitting the log"

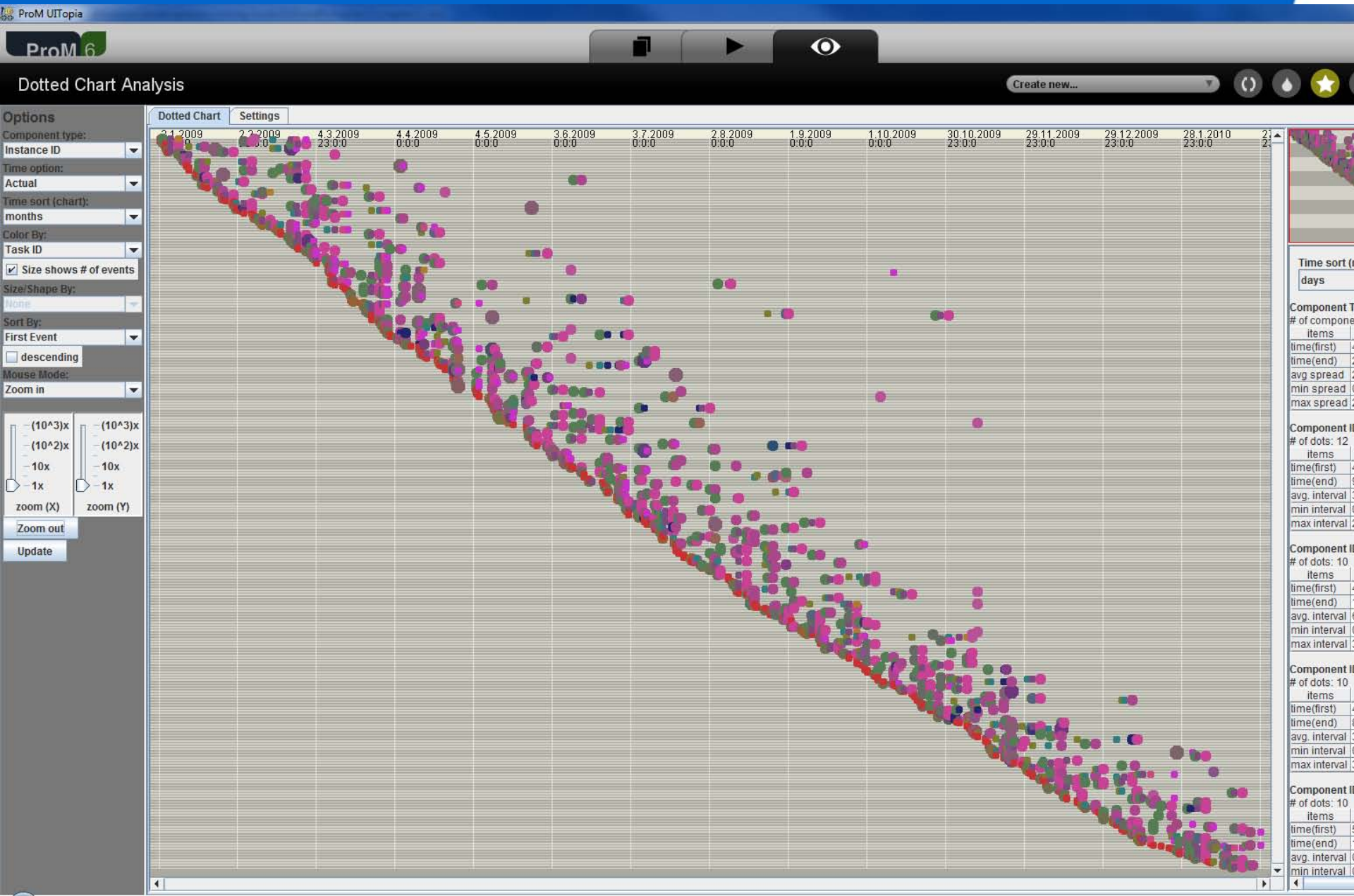




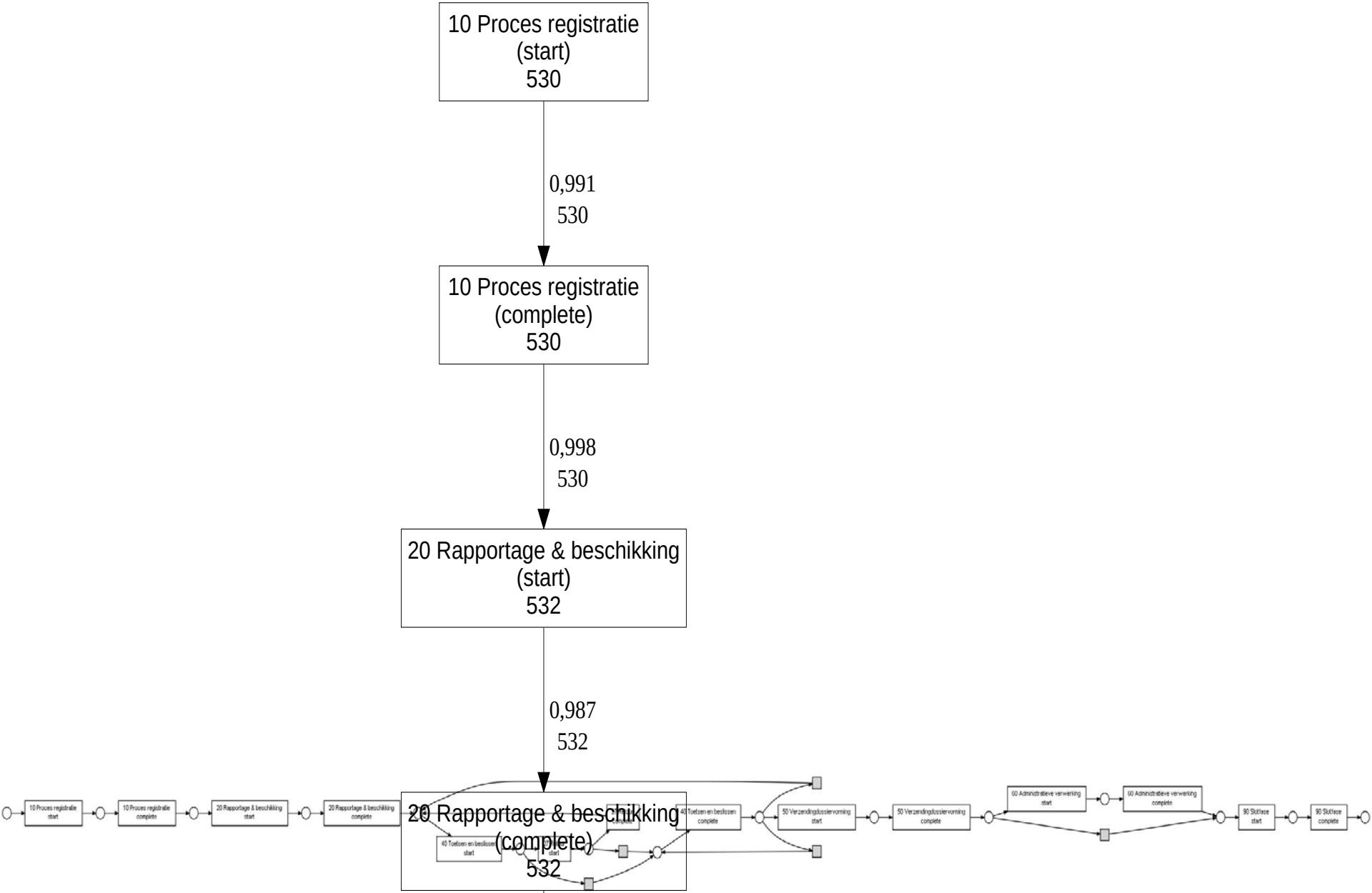




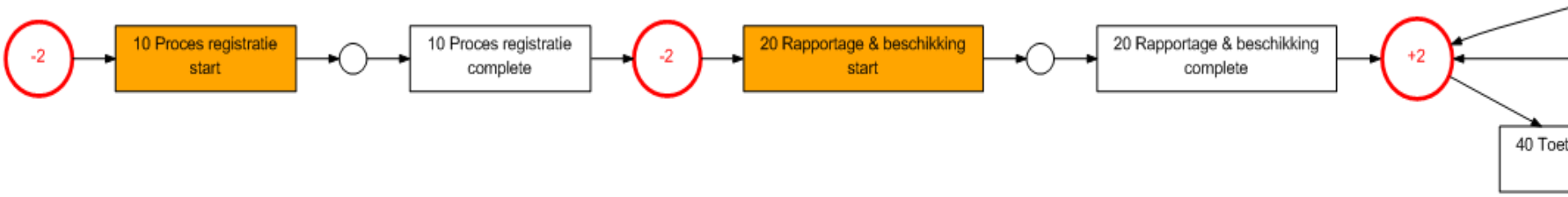
WMO process Harderwijk



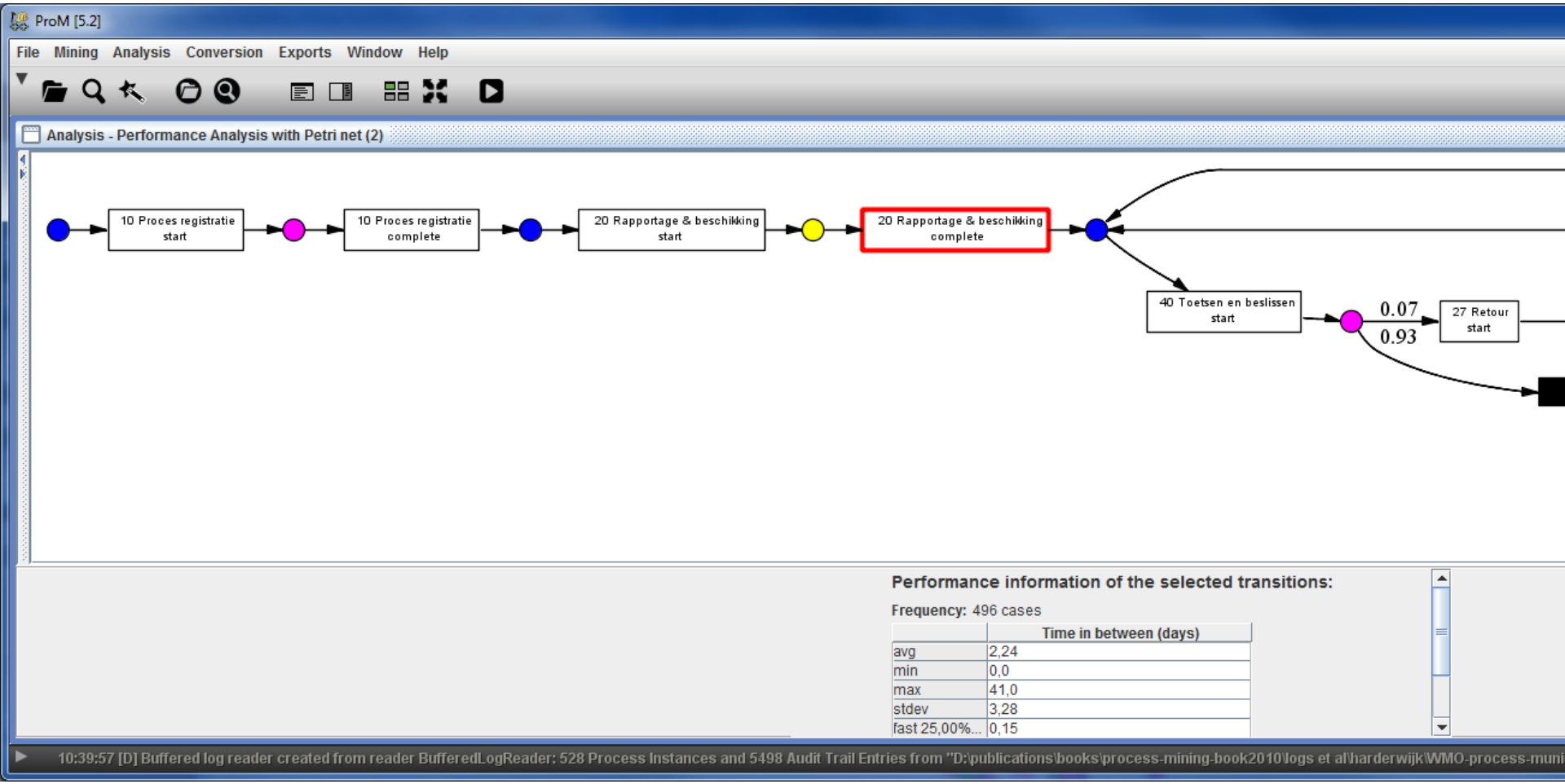
Process model discovered by HM



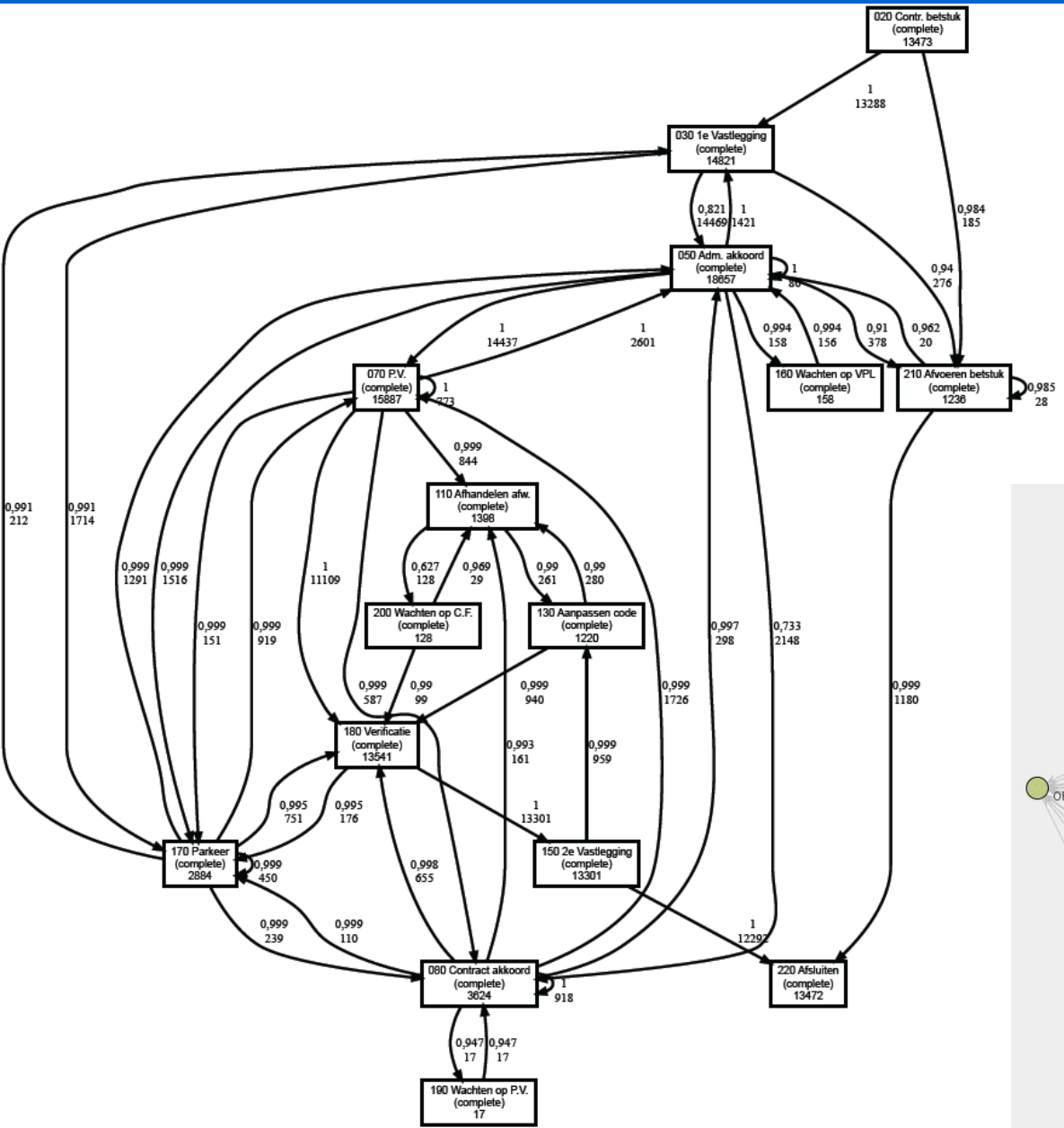
Conformance checking of discovered model

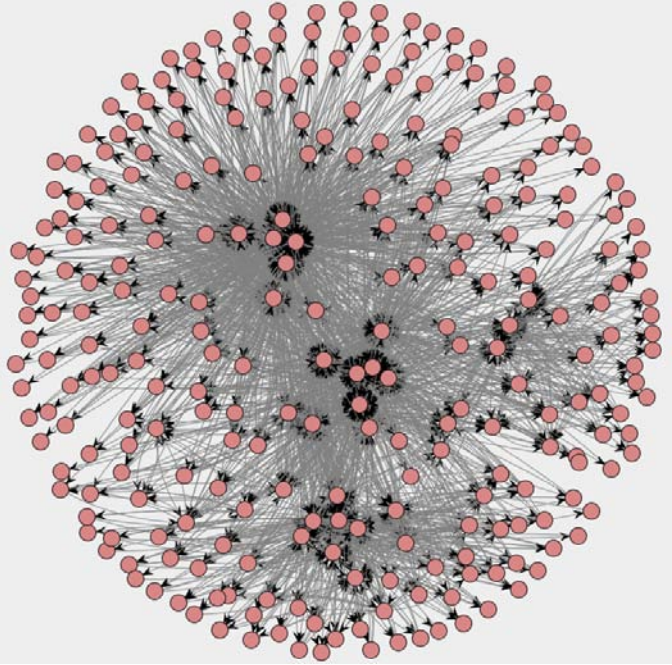


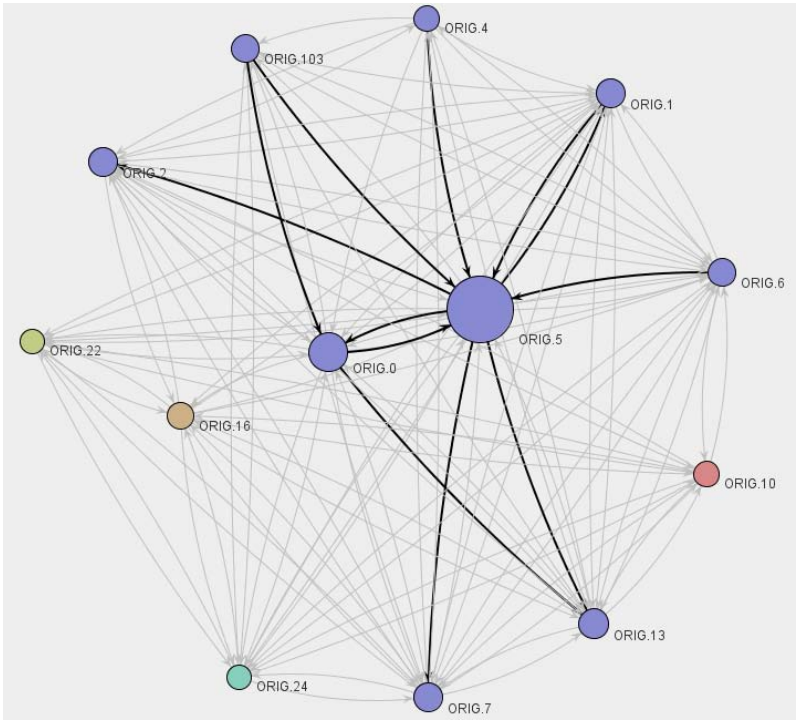
Performance Analysis



Rijkswaterstaat

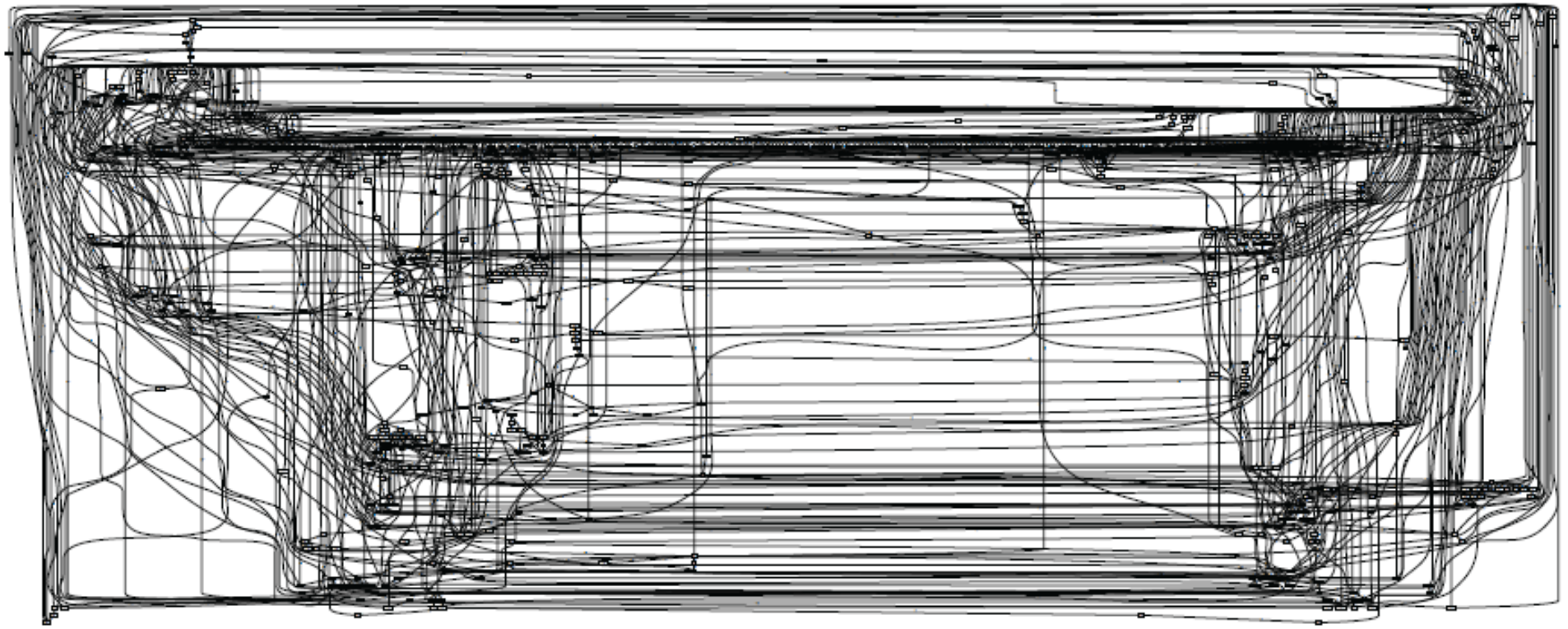






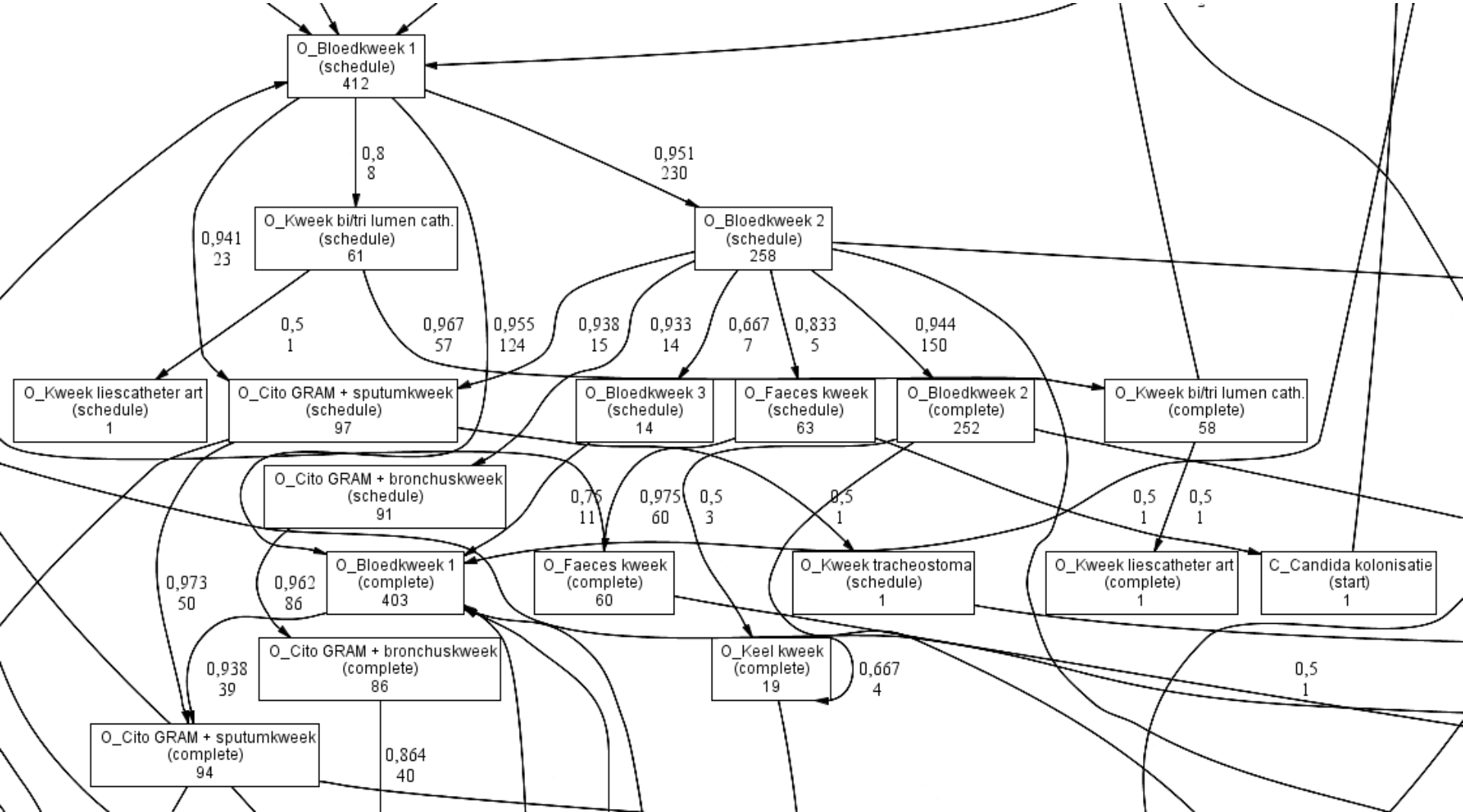
Catharina hospital (Eindhoven)

2765 patients, 114,592 events, 619 different activities (taking event types into account) executed by 266 different individuals (doctors, nurses, etc.)

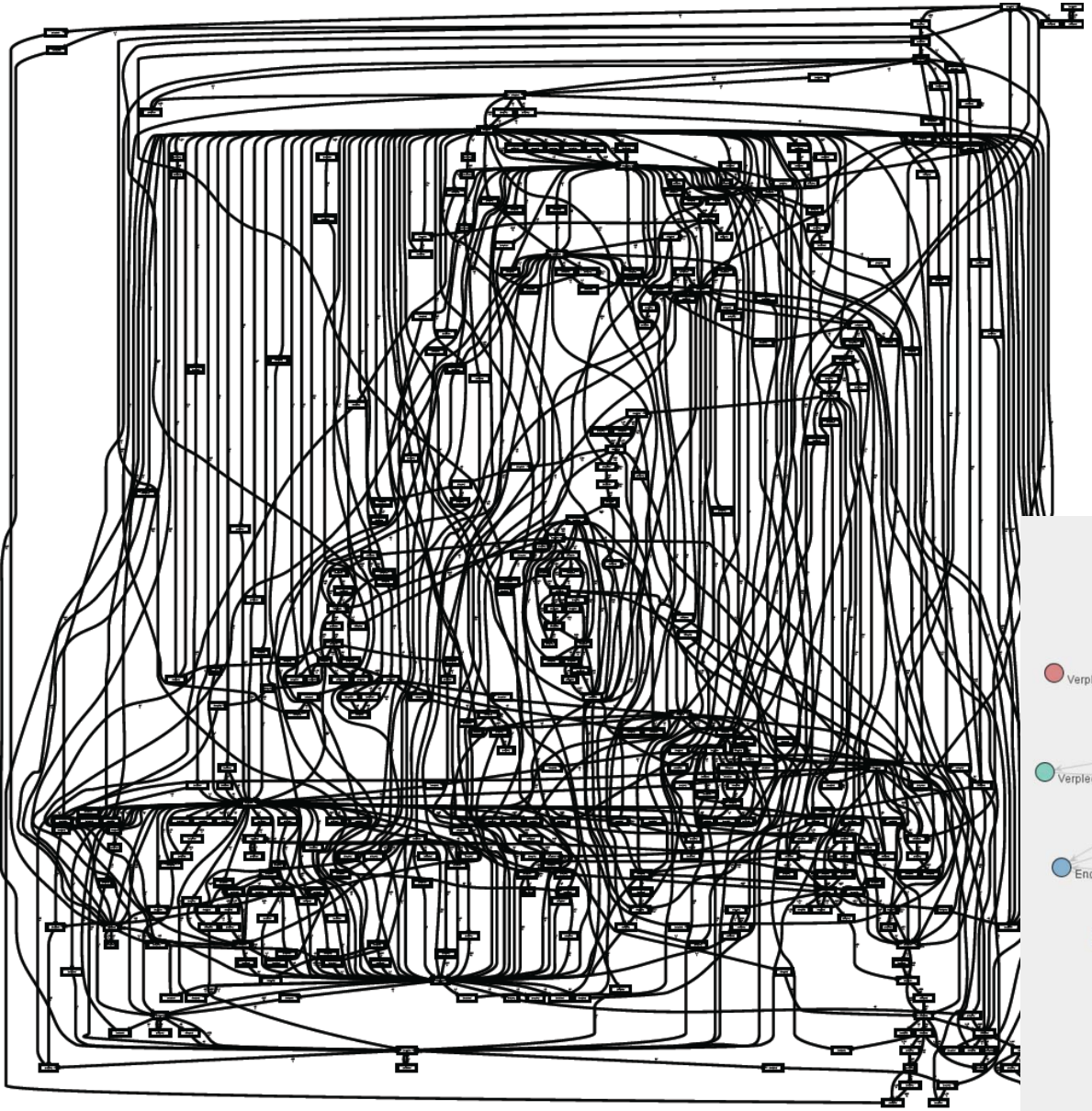


Fragment of process (2.9%)

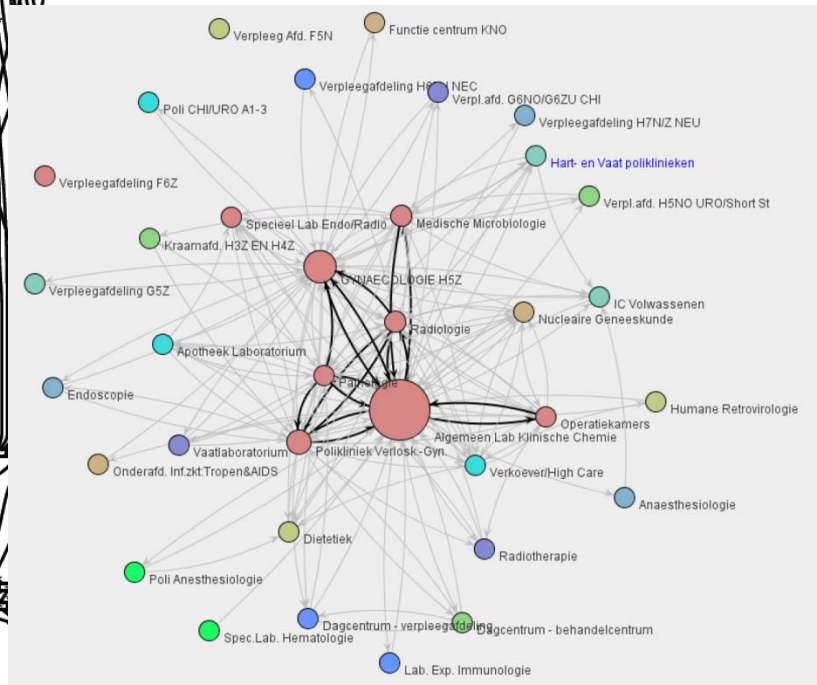
18 activities of the 619 activities



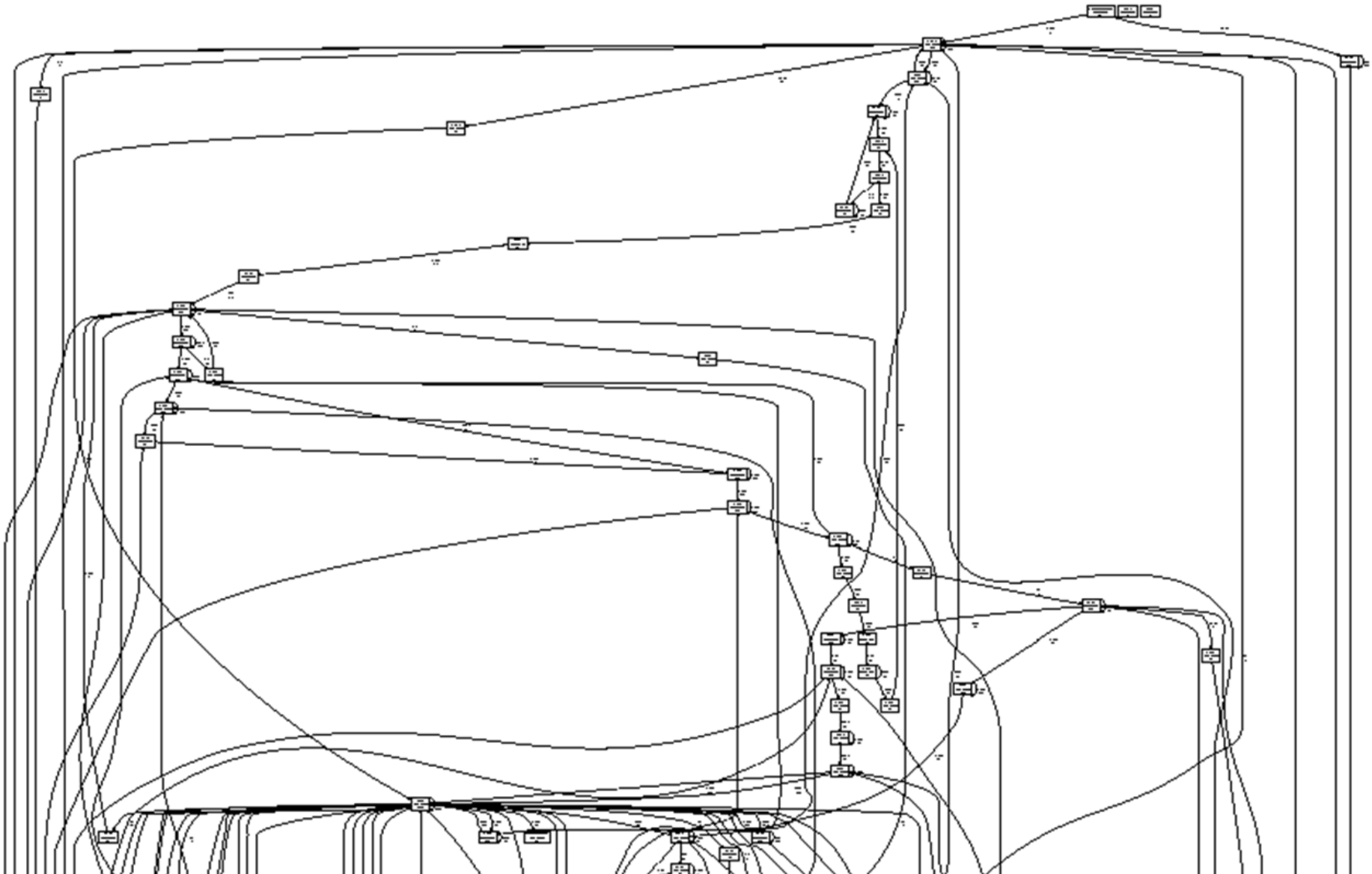
AMC hospital Amsterdam



627 gynecological oncology patients, 24331 events, 376 different activities



ASML testing process



Conclusion

Petri's guiding principles:

1) **Concurrency as a starting point rather than an afterthought (locality of actions).**
2) **Formalism should be consistent with the laws of physics.**

Process mining

- We can observe processes; they leave footprints in event logs that can be extracted from various data sources.
- We can learn process models.
- We can quantify the conformance of process models (standard equivalence notions are useless).
- Ignoring this source of information is like studying physics while ignoring experimental results.
- Evidence-based Business Process Management!

Wil M. P. van der Aalst

Process Mining

Discovery, Conformance and Enhancement of Business Processes

More and more information about business processes is recorded by information systems in the form of so-called "event logs". Despite the omnipresence of such data, most organizations diagnose problems based on fiction rather than facts. Process mining is an emerging discipline based on process model-driven approaches and data mining. It not only allows organizations to fully benefit from the information stored in their systems, but it can also be used to check the conformance of processes, detect bottlenecks, and predict execution problems.

Wil van der Aalst delivers the first book on process mining. It aims to be self-contained while covering the entire process mining spectrum from process discovery to operational support. In Part I, the author provides the basics of business process modeling and data mining necessary to understand the remainder of the book. Part II focuses on process discovery as the most important process mining task. Part III moves beyond discovering the control flow of processes and highlights conformance checking, and organizational and time perspectives. Part IV guides the reader in successfully applying process mining in practice, including an introduction to the widely used open-source tool ProM. Finally, Part V takes a step back, reflecting on the material presented and the key open challenges.

Overall, this book provides a comprehensive overview of the state of the art in process mining. It is intended for business process analysts, business consultants, process managers, graduate students, and BPM researchers.

Features and Benefits:

- First book on process mining, bridging the gap between business process modeling and business intelligence.
- Written by one of the most influential and most-cited computer scientists and the best-known BPM researcher.
- Self-contained and comprehensive overview for a broad audience in academia and industry.
- The reader can put process mining into practice immediately due to the applicability of the techniques and the availability of the open-source process mining software ProM.

Computer Science

ISBN 978-3-642-19344-6

springer.com

van der Aalst

Process Mining

Wil M. P. van der Aalst

Process Mining

Discovery, Conformance and Enhancement of Business Processes

More Information

IEEE Task Force on Process Mining

- ProM Software: prom.sourceforge.net
- Process mining: www.processmining.org
- ProM 5 series nightly builds: prom.win.tue.nl/tools/prom/nightly5/
- ProM 6 series nightly builds: prom.win.tue.nl/tools/prom/nightly/
- Converting logs (MXML-based) promimport.sourceforge.net
- XES: www.xes-standard.org and www.openxes.org
- Papers et al.: vdaalst.com
- IEEE Task Force on Process Mining: www.win.tue.nl/ieeetfpm/