

Object-Centric Process Mining

PADS Seminar 19-11-2019

Joint work with Alessandro Berti

prof.dr.ir. Wil van der Aalst
RWTH Aachen University
W: vdaalst.com T:@wvdaalst

Remember the initial assumption

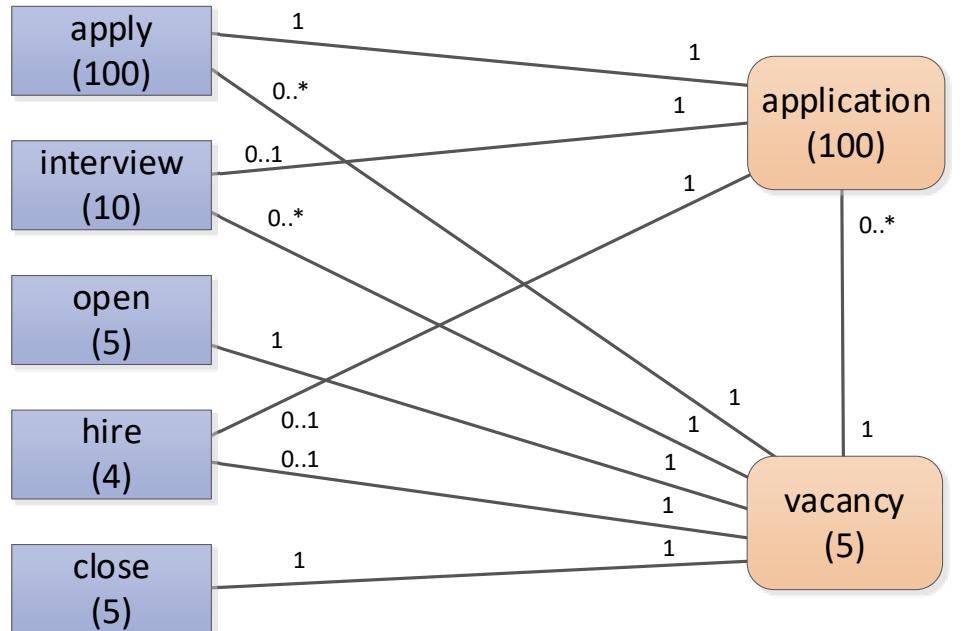
Case ID	Activity	Resource	Timestamp	product	prod-price	quantity	address
...
6350	place order	Aiden	2018/02/13 14:29:45.000	APPLE iPhone 6 16 GB	639,00 €	5	NL-7751DG-21
6283	pay	Lily	2018/02/13 14:39:25.000	SAMSUNG Galaxy S6 32 GB	542,99	3	NL-7828AM-11a
6253	prepare delivery	Sophia	2018/02/13 15:01:33.000	APPLE iPhone 6s Plus 64 GB	639,00 €	3	NL-7887AC-13
6257	prepare delivery	Aiden	2018/02/13 15:03:43.000	SAMSUNG Galaxy S6 32 GB	543,99	1	NL-9521KI-34
6185	confirm payment	Emily	2018/02/13 15:05:36.000	SAMSUNG Galaxy S4	329,00 €	1	NL-9521GC-32
6218	confirm payment	Emily	2018/02/13 15:08:11.000	APPLE iPhone 6s Plus 64 GB	969,00 €	2	NL-7948BX-10
6245	make delivery	Michael	2018/02/13 15:14:04.000	APPLE iPhone 6s 16 GB	639,00 €	3	NL-7905AX-38
6272	pay	Emily	2018/02/13 15:20:36.000	APPLE iPhone 6s 16 GB	639,00 €	1	NL-7821AC-3
6269	pay	Charlotte	2018/02/13 15:25:21.000	SAMSUNG Galaxy S4	329,00 €	1	NL-7907EJ-42
6212	prepare delivery	Sophia	2018/02/13 15:43:39.000	HUAWEI P8 Lite	334,00 €	1	NL-7905AX-38
6323	send invoice	Alexander	2018/02/13 15:46:08.000	APPLE iPhone 6s 16 GB	639,00 €	1	NL-7833HT-15
6246	confirm payment	Jack	2018/02/13 15:56:03.000	SAMSUNG Galaxy S4	329,00 €	3	NL-7833HT-15
6347	send invoice	Jack	2018/02/13 15:57:42.000	SAMSUNG Galaxy S4	329,00 €	3	NL-7905AX-38
6351	place order	Zoe	2018/02/13 16:17:37.000	APPLE iPhone 5s 16 GB	449,00 €	3	NL-9521GC-32
6204	prepare delivery	Sophia	2018/02/13 16:31:28.000	SAMSUNG Galaxy S6 32 GB	542,99	1	NL-7828AM-11a
6204	make delivery	Kaylee	2018/02/13 16:51:54.000	SAMSUNG Galaxy S6 32 GB	542,99	1	NL-7828AM-11a
6265	confirm payment	Lily	2018/02/13 16:55:55.000	SAMSUNG Galaxy S4	329,00 €	4	NL-9521GC-32
6250	confirm payment	Jack	2018/02/13 17:03:26.000	MOTOROLA Moto G	199,00 €	4	NL-7942GT-2
6328	send invoice	Lily	2018/02/13 17:30:16.000	APPLE iPhone 6s 64 GB	858,00 €	4	NL-9514BV-16
6352	place order	Aiden	2018/02/13 17:53:22.000	APPLE iPhone 6 16 GB	639,00 €	2	NL-9514BV-16
6317	send invoice	Jack	2018/02/13 18:45:30.000	APPLE iPhone 6s 64 GB	858,00 €	5	NL-7907EJ-42
6353	place order	Sophia	2018/02/13 20:16:20.000	APPLE iPhone 5s 16 GB	449,00 €	4	NL-7751AR-19
...

event =
case +
activity +
timestamp +



Example: Vacancies and Applications

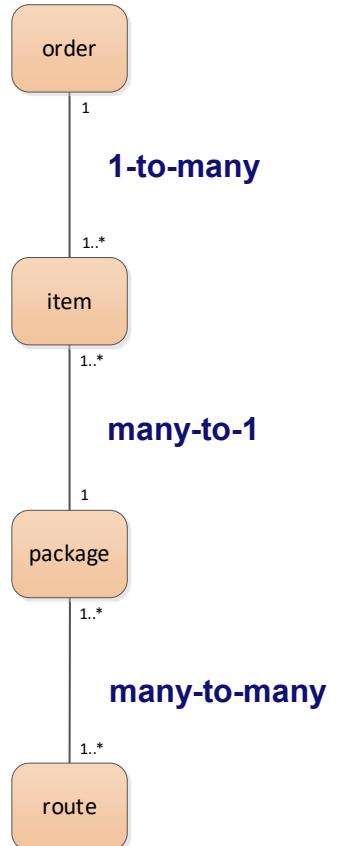
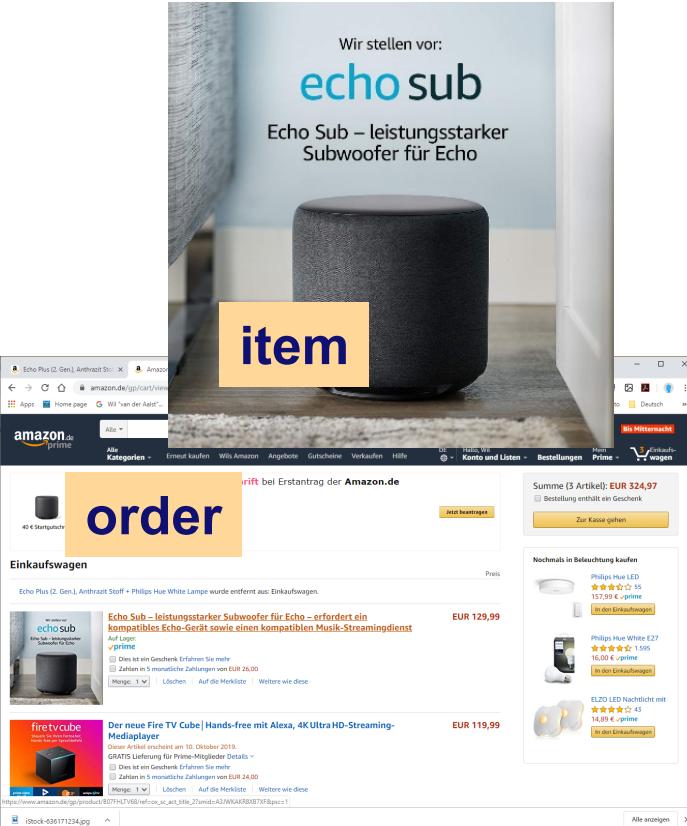
activities



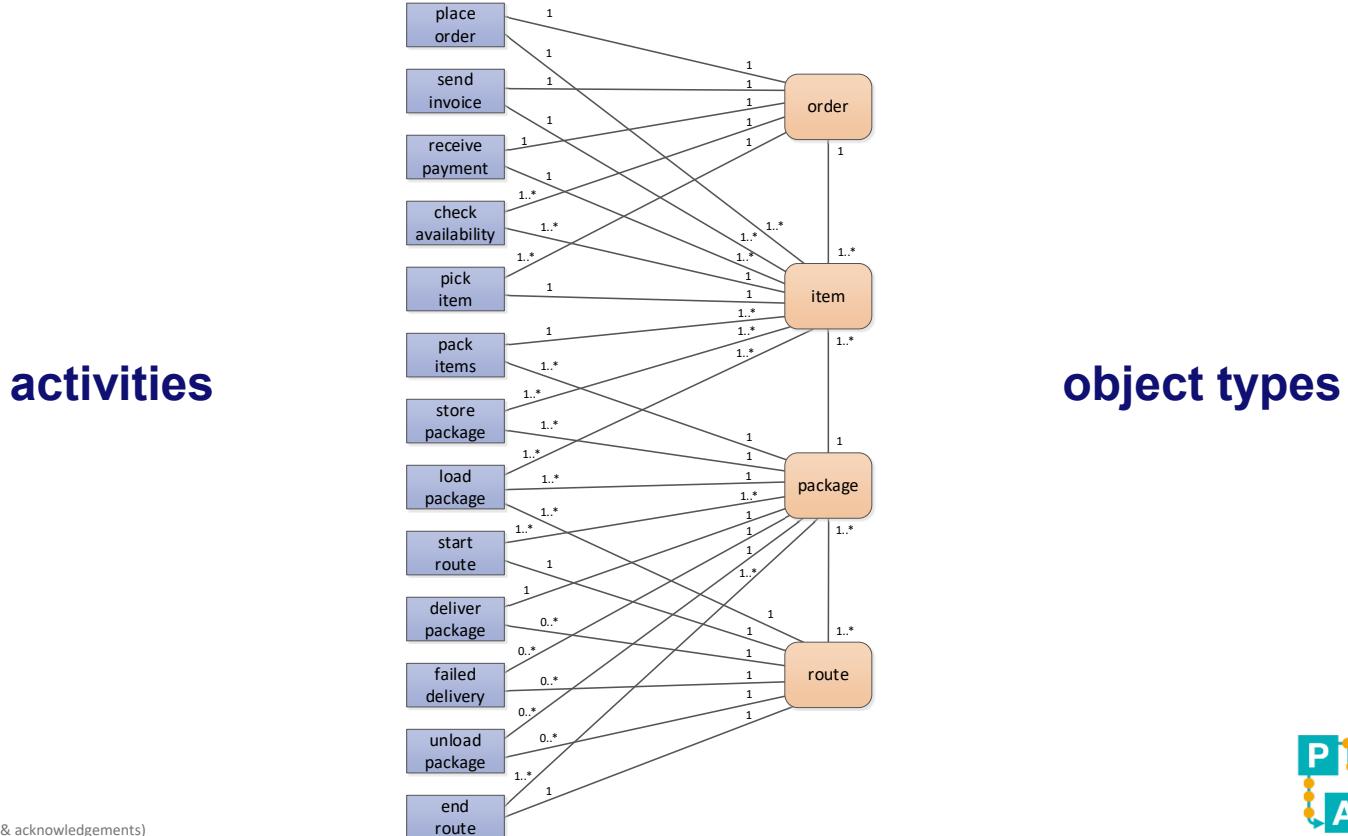
What case id ?

Still easy: 1-to-many rather than many-to-many

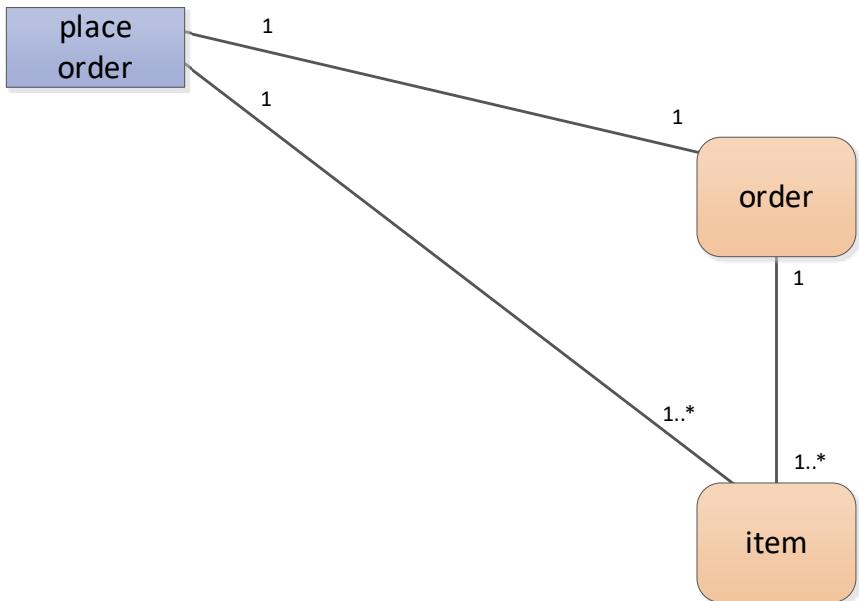
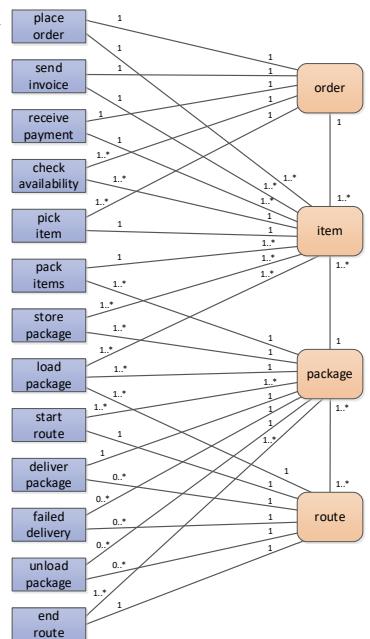
Slightly more complex example



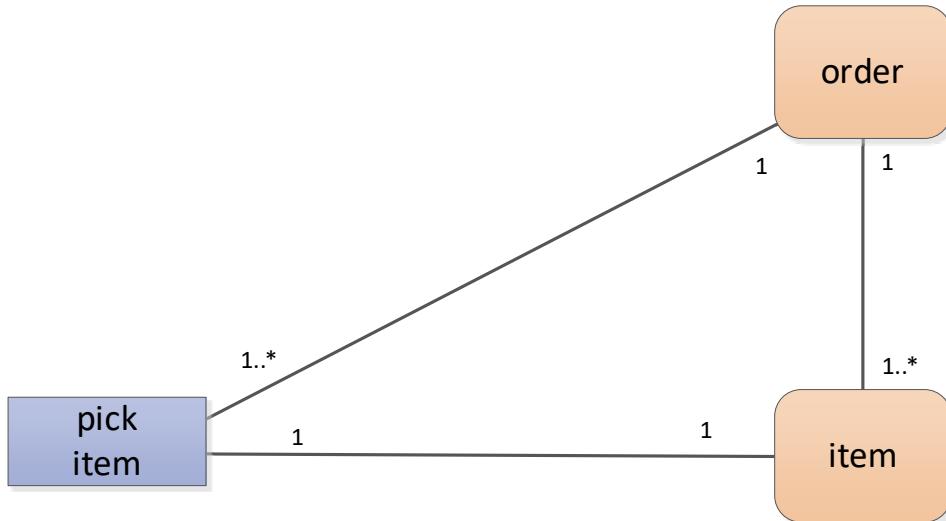
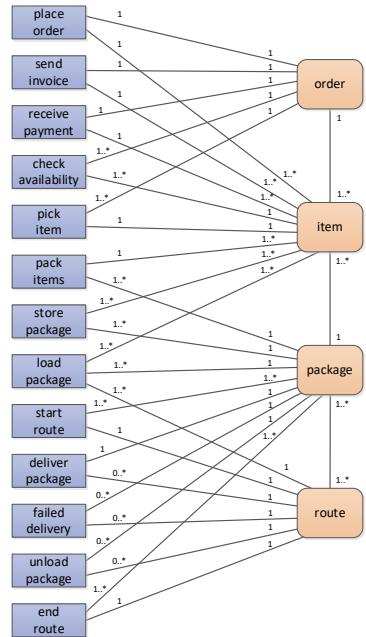
Slightly more complex example



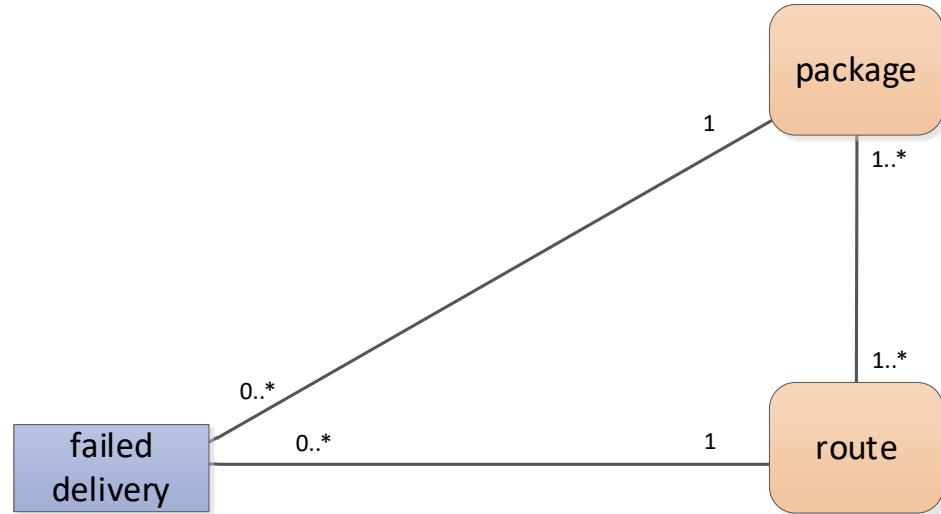
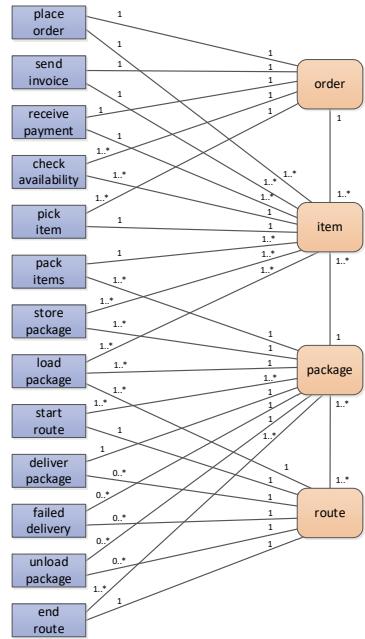
Example activity: Place order



Example activity: Pick item

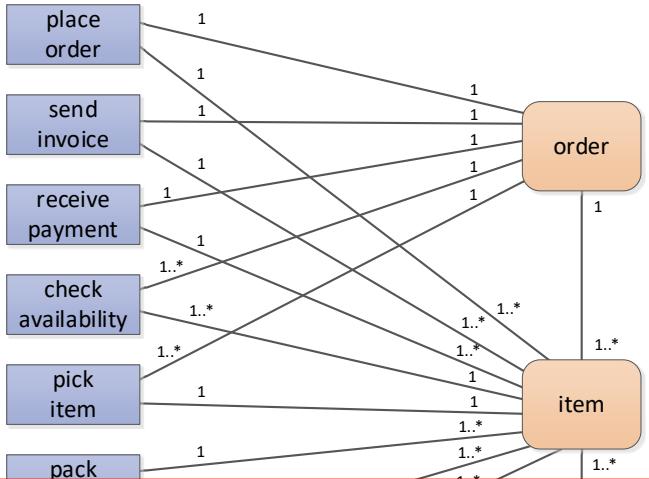


Example activity: Failed delivery



Another logging format: Beyond XES

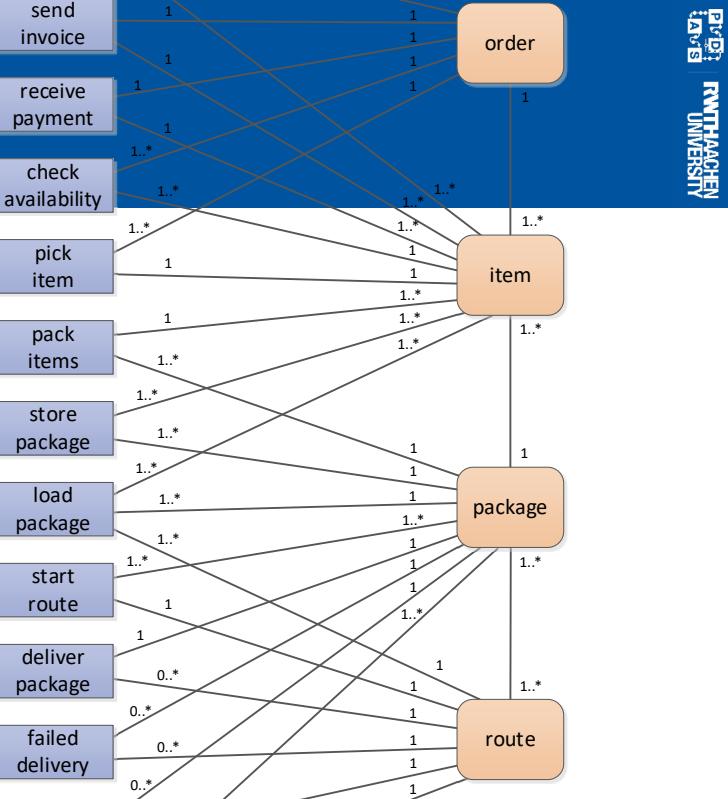
event identifier	activity name	timestamp	objects involved				attribute values	
			order	item	package	route	customer	costs
...
9911	place_order	20-7-2019:08.15	{o1}	{i1, i2}	Ø	Ø	Apple	3500€
9912	check_availability	20-7-2019:09.35	{o1}	{i1}	Ø	Ø		
9913	place_order	20-7-2019:09.38	{o2}	{i3, i4, i5}	Ø	Ø	Google	4120€
9914	check_availability	20-7-2019:10.20	{o1}	{i2}	Ø	Ø		
9915	pick_item	20-7-2019:11.05	{o1}	{i1}	Ø	Ø		
9916	check_availability	20-7-2019:11.19	{o2}	{i3}	Ø	Ø		
9917	pick_item	20-7-2019:11.55	{o2}	{i3}	Ø	Ø		
9918	check_availability	20-7-2019:13.15	{o2}	{i4}	Ø	Ø		
9919	pick_item	20-7-2019:14.20	{o2}	{i4}	Ø	Ø		
9920	check_availability	20-7-2019:15.25	{o2}	{i5}	Ø	Ø		
9921	check_availability	20-7-2019:16.34	{o1}	{i2}	Ø	Ø		
9922	pick_item	20-7-2019:16.68	{o1}	{i2}	Ø	Ø		
9923	pack_items	20-7-2019:16.44	Ø	{i1, i2, i3}	{p1}	Ø		
9924	store_package	20-7-2019:16.55	Ø	{i1, i2, i3}	{p1}	Ø		
9925	start_route	20-7-2019:16.56	Ø	Ø	{p1}	{r1}		
9926	load_package	21-7-2019:08.00	Ø	{i1, i2, i3}	{p1}	{r1}		
9927	send_invoice	21-7-2019:08.17	{o1}	{i1, i2}	Ø	Ø		
9928	place_order	21-7-2019:08.28	{o3}	{i6}	Ø	Ø		
9929	failed_delivery	21-7-2019:08.88	Ø	Ø	{p1}	{r1}		
9930	unload_package	21-7-2019:08.56	Ø	Ø	{p1}	{r1}		
9931	end_route	21-7-2019:09.16	Ø	Ø	{p1}	{r1}		
9932	check_availability	21-7-2019:10.26	{o3}	{i6}	Ø	Ø		
9933	receive_payment	21-7-2019:11.55	{o1}	{i1, i2}	Ø	Ø		
9934	check_availability	22-7-2019:08.19	{o2}	{i3}	Ø	Ø		
9935	pick_item	22-7-2019:08.44	{o2}	{i3}	Ø	Ø		
9936	send_invoice	22-7-2019:08.55	{o2}	{i3, i4, i5}	Ø	Ø		
9937	receive_payment	22-7-2019:09.15	{o3}	{i3, i4, i5}	Ø	Ø		
9938	check_availability	22-7-2019:10.35	{o3}	{i6}	Ø	Ø		
9939	pick_item	22-7-2019:11.29	{o3}	{i6}	Ø	Ø		
9940	pack_items	22-7-2019:11.29	{o3}	{i6}	Ø	Ø		
9941	pack_items	22-7-2019:09.11	Ø	{i4, i5, i6}	Ø	Ø		
9942	send_invoice	22-7-2019:11.45	Ø	{i6}	Ø	Ø		
9943	store_package	22-7-2019:09.19	Ø	{i4, i5, i6}	Ø	Ø		
9944	start_route	22-7-2019:09.28	Ø	Ø	Ø	Ø		
9945	load_package	22-7-2019:10.05	Ø	{i1, i2, i3}	Ø	Ø		
9946	load_package	23-7-2019:10.09	Ø	{i4, i5, i6}	Ø	Ø		
9947	deliver_package	23-7-2019:11.26	Ø	Ø	Ø	Ø		
9948	deliver_package	24-7-2019:09.87	Ø	Ø	Ø	Ø		
9949	end_route	24-7-2019:09.48	Ø	Ø	Ø	Ø		
9950	receive_payment	24-7-2019:09.55	{o3}	{i6}	Ø	Ø		
...



event identifier	activity name	timestamp	objects involved			
			order	item	package	route
...
9911	place_order	20-7-2019:08.15	{o1}	{i1, i2}	Ø	Ø
9912	check_availability	20-7-2019:09.35	{o1}	{i1}	Ø	Ø
9913	place_order	20-7-2019:09.38	{o2}	{i3, i4, i5}	Ø	Ø

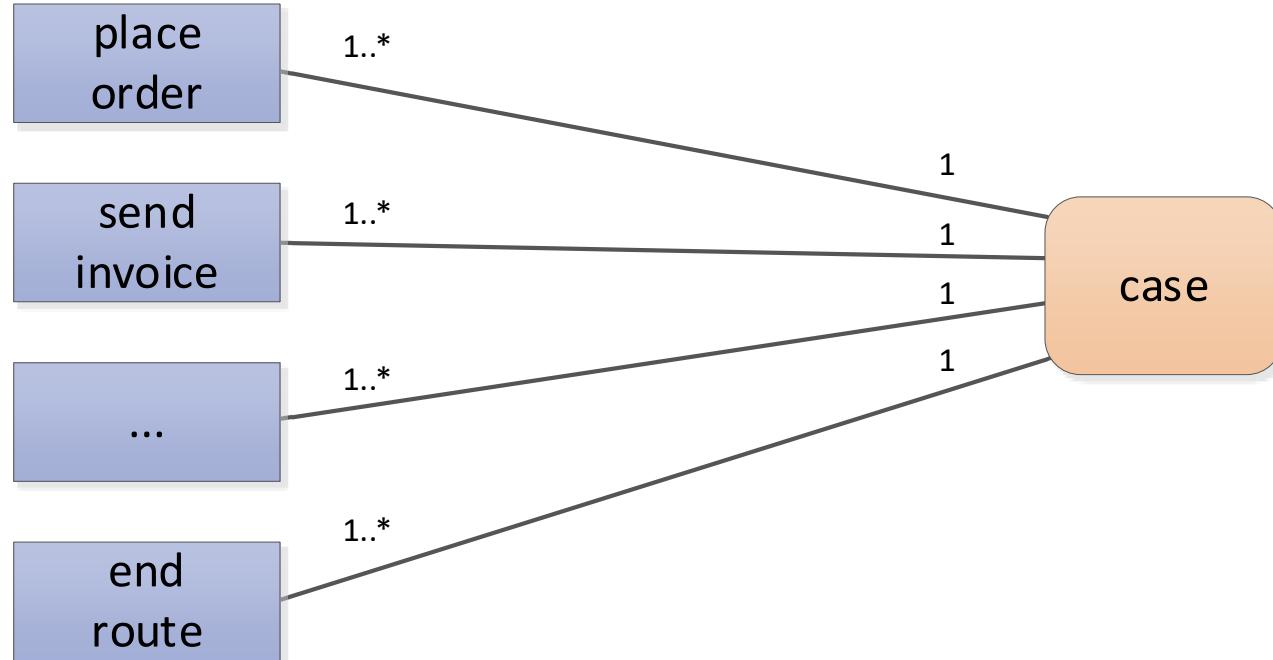
Another logging format

event identifier	activity name	timestamp	objects involved				attribute values	
			order	item	package	route	customer	costs
...
9911	place_order	20-7-2019:08.15	{o1}	{i1, i2}	0	0	Apple	3500€
9912	check_availability	20-7-2019:09.35	{o1}	{i1}	0	0		
9913	place_order	20-7-2019:09.98	{o2}	{i3, i4, i5}	0	0	Google	4120€
9914	check_availability	20-7-2019:10.20	{o1}	{i2}	0	0		
9915	pick_item	20-7-2019:11.05	{o1}	{i1}	0	0		
9916	check_availability	20-7-2019:11.19	{o2}	{i3}	0	0		
9917	pick_item	20-7-2019:11.55	{o2}	{i3}	0	0		
9918	check_availability	20-7-2019:13.15	{o2}	{i4}	0	0		
9919	pick_item	20-7-2019:14.20	{o2}	{i4}	0	0		
9920	check_availability	20-7-2019:15.25	{o2}	{i5}	0	0		
9921	check_availability	20-7-2019:16.84	{o1}	{i2}	0	0		
9922	pick_item	20-7-2019:16.93	{o1}	{i2}	0	0		
9923	pack_items	20-7-2019:16.44	Ø	{i1, i2, i3}	{p1}	0		
9924	store_package	20-7-2019:16.55	Ø	{i1, i2, i3}	{p1}	0		
9925	start_route	20-7-2019:16.56	Ø	Ø	{p1}	{r1}		
9926	load_package	21-7-2019:08.00	Ø	{i1, i2, i3}	{p1}	{r1}		
9927	send_invoice	21-7-2019:08.17	{o1}	{i1, i2}	0	0		
9928	place_order	21-7-2019:08.28	{o3}	{i6}	0	0	Microsoft	1894€
9929	failed_delivery	21-7-2019:08.88	Ø	Ø	{p1}	{r1}		
9930	unload_package	21-7-2019:08.56	Ø	Ø	{p1}	{r1}		
9931	end_route	21-7-2019:09.16	Ø	Ø	{p1}	{r1}		
9932	check_availability	21-7-2019:10.26	{o3}	{i6}	0	0		
9933	receive_payment	21-7-2019:11.55	{o1}	{i1, i2}	0	0		
9934	check_availability	22-7-2019:08.19	{o2}	{i5}	0	0		
9935	pick_item	22-7-2019:08.44	{o2}	{i5}	0	0		
9936	send_invoice	22-7-2019:08.55	{o2}	{i3, i4, i5}	0	0		
9937	receive_payment	22-7-2019:09.15	{o2}	{i3, i4, i5}	0	0		
9938	check_availability	22-7-2019:10.05	{o3}	{i6}	0	0		
9939	pick_item	22-7-2019:11.29	{o3}	{i6}	0	0		
9941	pack_items							
9942	send_invoice							
9943	store_package							
9944	start_route							
9945	load_package							
9946	load_package							
9947	deliver_package							
9948	deliver_package							
9949	end_route							
9950	receive_payment							
...	...							



9923	pack_items	20-7-2019:16.44	Ø	{i1, i2, i3}	{p1}	Ø
9924	store_package	20-7-2019:16.55	Ø	{i1, i2, i3}	{p1}	Ø
9925	start_route	20-7-2019:16.56	Ø	Ø	{p1}	{r1}
9926	load_package	21-7-2019:08.00	Ø	{i1, i2, i3}	{p1}	{r1}

Traditional view: One case id per event



Flattening does not work

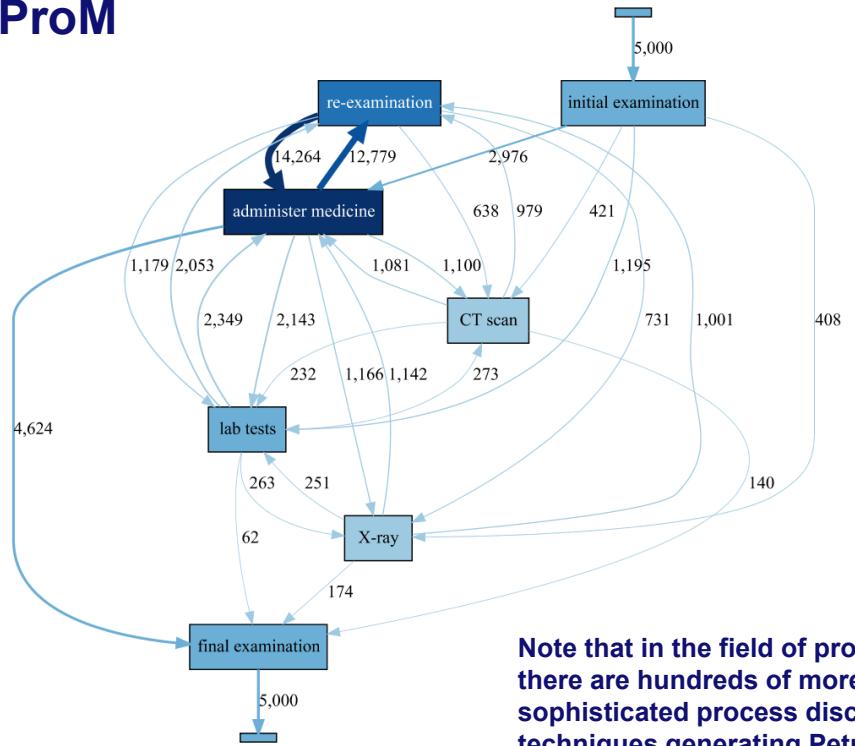
event identifier	activity name	timestamp	objects involved			
			order	item	package	route
...
9911	place_order	20-7-2019:08.15	{o1}	{i1, i2}	Ø	Ø
9912	check_availability	20-7-2019:09.35	{o1}	{i1}	Ø	Ø
9913	place_order	20-7-2019:09.38	{o2}	{i3, i4, i5}	Ø	Ø

- One can **create a conventional event log** by picking one object type as case identifier. If an event has multiple objects of that type, then simply create one event for each object.
- Possible problems:
 - **Deficiency:** Events without a case id are lost.
 - **Convergence:** Events referring to multiple case objects are replicated leading to unintentional duplication.
 - **Divergence:** Events of the same case referring to different (secondary) objects. These seem to be related but are not. This results in “spaghetti” & loops.

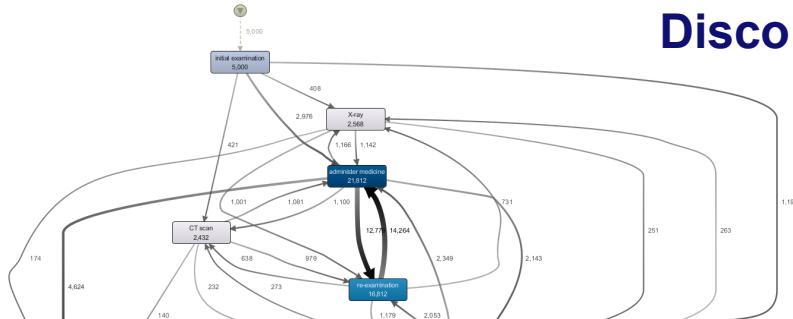
Challenge: Object-Centric Process Discovery

Take a simple Directly Follows Graph (DFG)

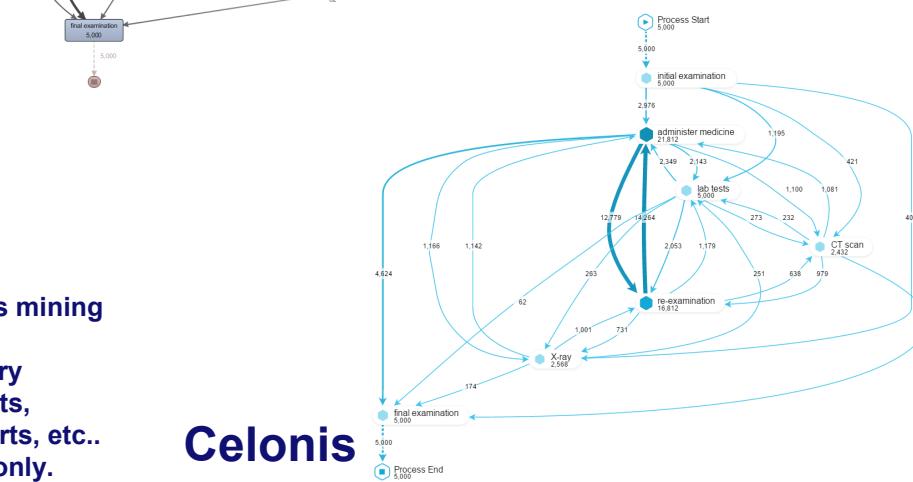
ProM



Disco

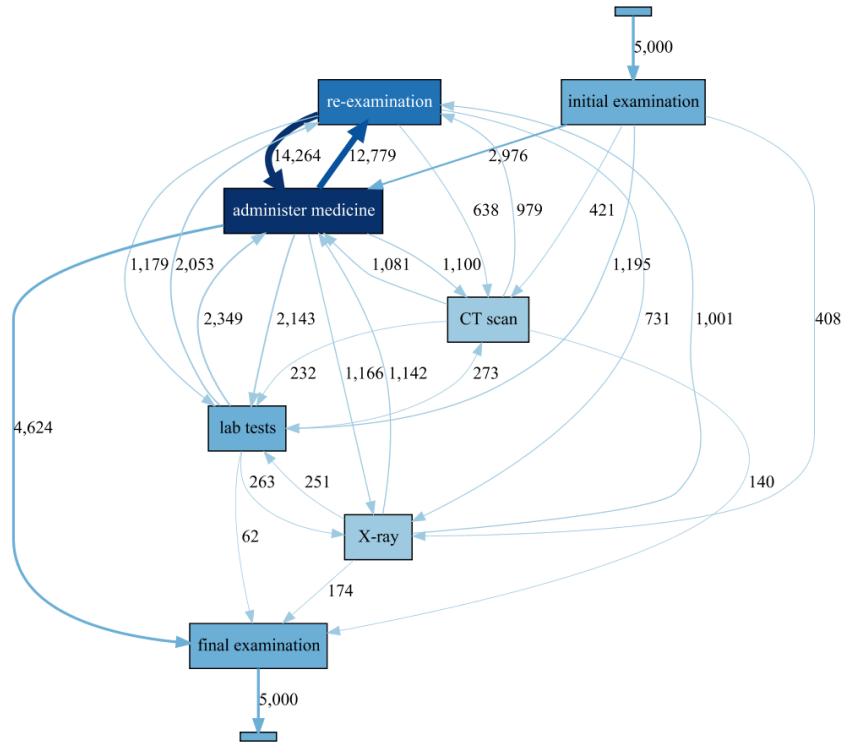


Celonis



Note that in the field of process mining there are hundreds of more sophisticated process discovery techniques generating Petri nets, process trees, BPMN, statecharts, etc.. DFGs are used for illustration only.

Take a simple Directly Follows Graph (DFG)



- Nodes refer to activities
- Arcs refer to direct successions within a case.
- Example: (cases grouped per trace)
 $[\langle a, b, d \rangle^{10}, \langle a, a, b, c, d \rangle^5]$
 - a occurs 20 times, etc.
 - (a, a) occurs 5 times, (a, b) occurs 15 times, etc.

Design choices Object Centric DFG

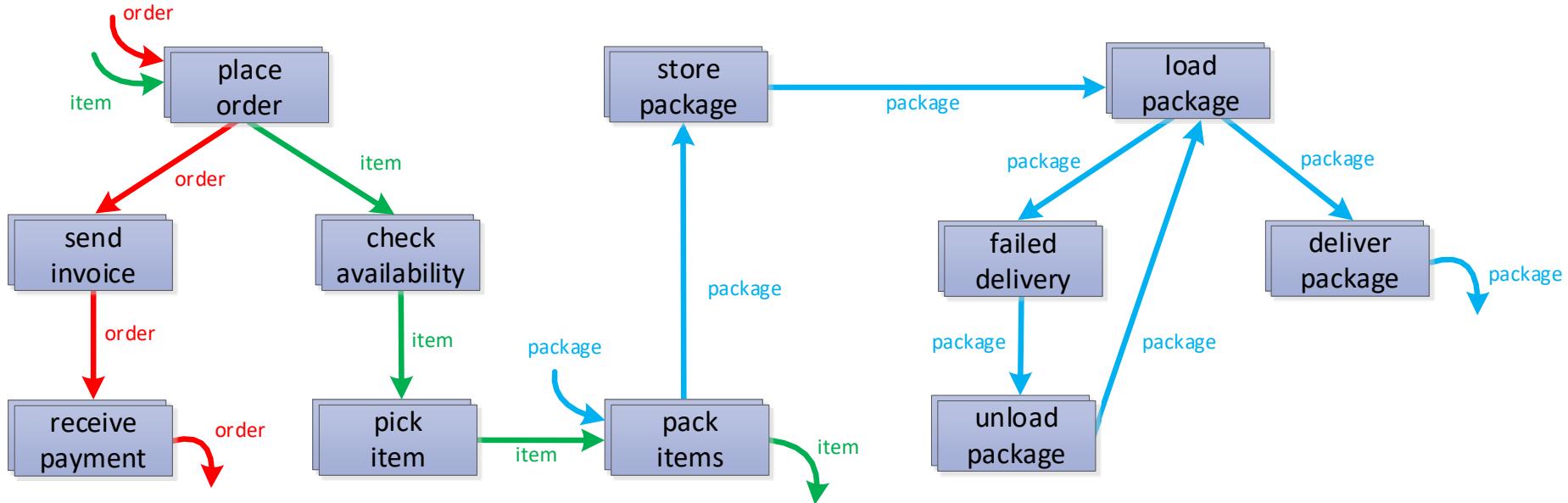
- We want to lift the DFG notion to **multiple object types**.
- Let's make the following choices:
 - **Nodes** refer to **activities** and each activity appears once in the diagram.
 - The **frequency** of an activity a matches the number of a events.
 - **Arcs** connect activities and correspond to **object types**.
 - The **frequency** of an arc represents the number **event pairs** directly following each other for some **object** of the selected type.

Design choices Object Centric DFG

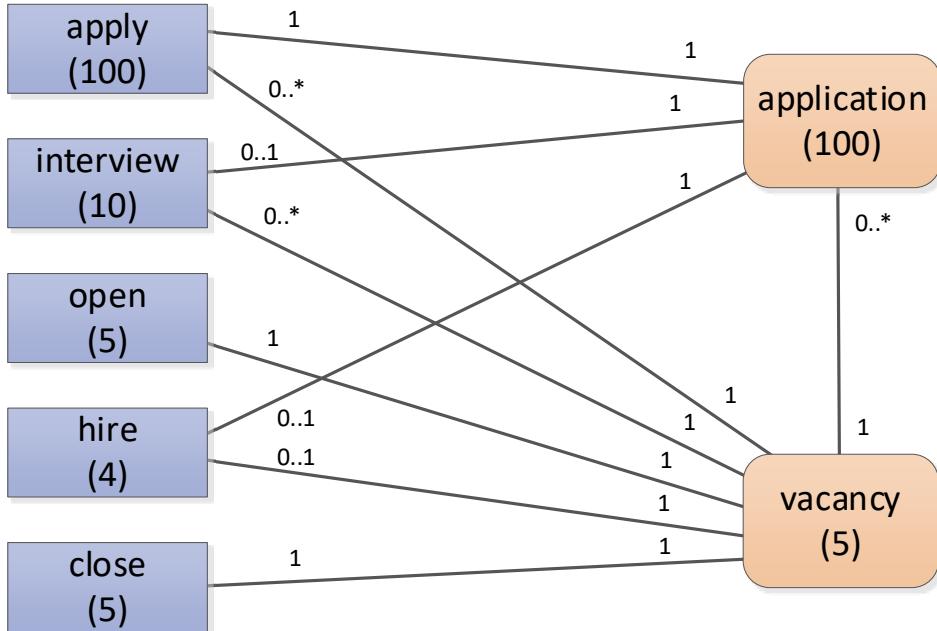
- Let's make the following choices:
 - Nodes refer to activities and each activity appears once in the diagram.
(Nodes may also refer to object types and if so, one can pick one object type and still have one node per activity or allow for multiple nodes per activity referring to different object types.)
 - The frequency of an activity a matches the number of a events.
(Instead of counting events one can also count event-object combinations. Many possibilities that depend on the previous step.)
 - Arcs connect activities and correspond to object types.
(Arcs could also refer to sets of object types.)
 - The frequency of an arc represents the number event pairs directly following each other for some object of the selected type.*
(There are many possibilities strongly depending on the choices made above, e.g., one can count objects related through events.)



Example output



Smaller example



- **100 applications for 5 vacancies, 10 interviews, 4 applicants are hired.**
- **Assume an equal distribution of applications (20:20:20:20:20) and interviews (2:2:2:2:2) over the 5 vacancies.**
- **For one vacancy nobody is hired.**
- **Assume interviews are done after the deadline for applications and the hiring decision is made after all interviews.**

Partial model: Activities only

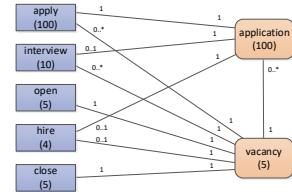
open
(5)

apply
(100)

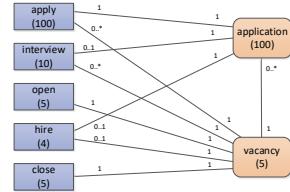
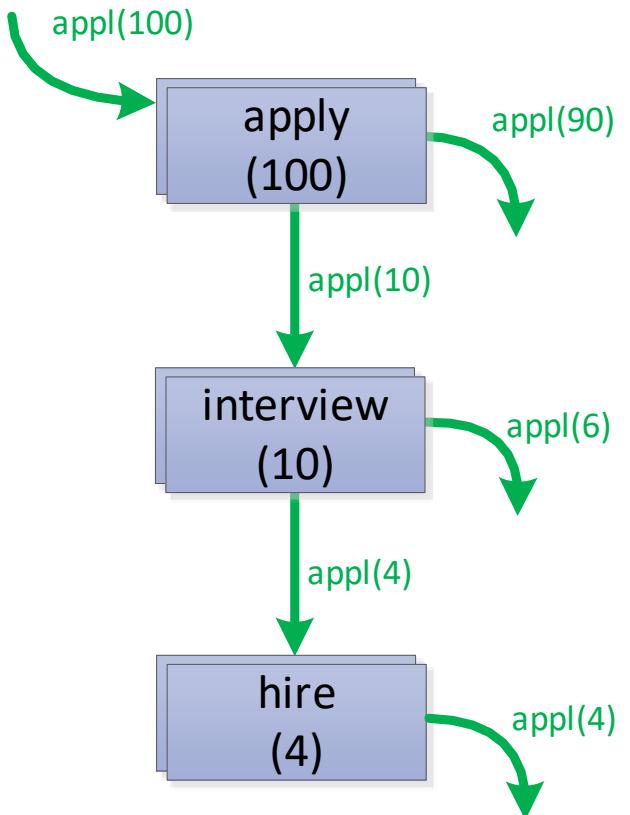
interview
(10)

close
(5)

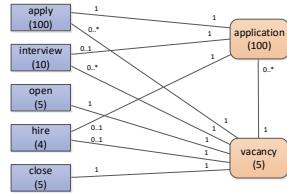
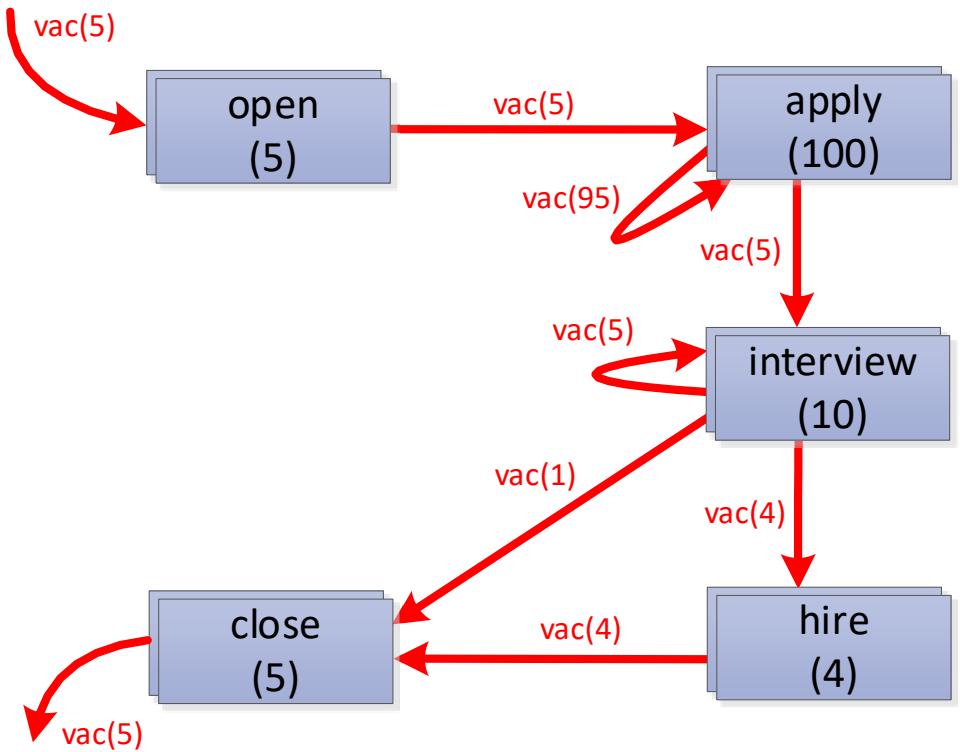
hire
(4)



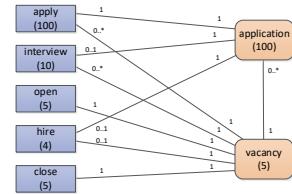
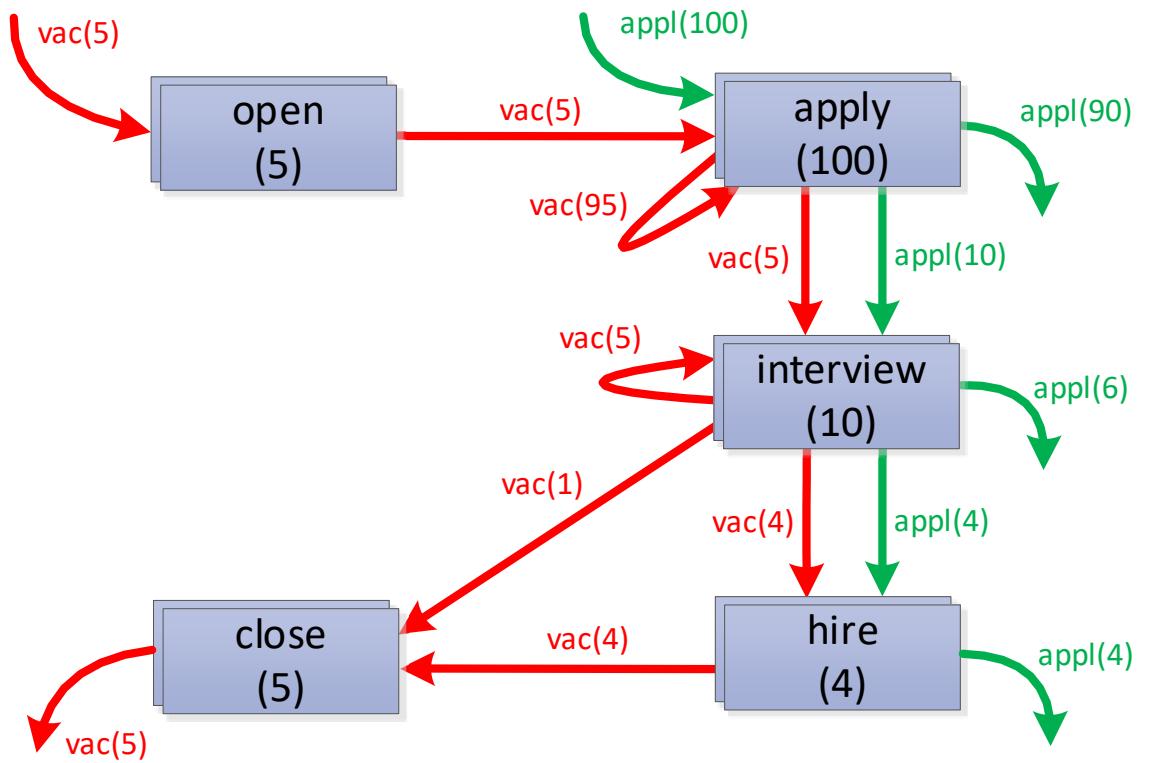
Partial model: Applications



Partial model: Vacancies

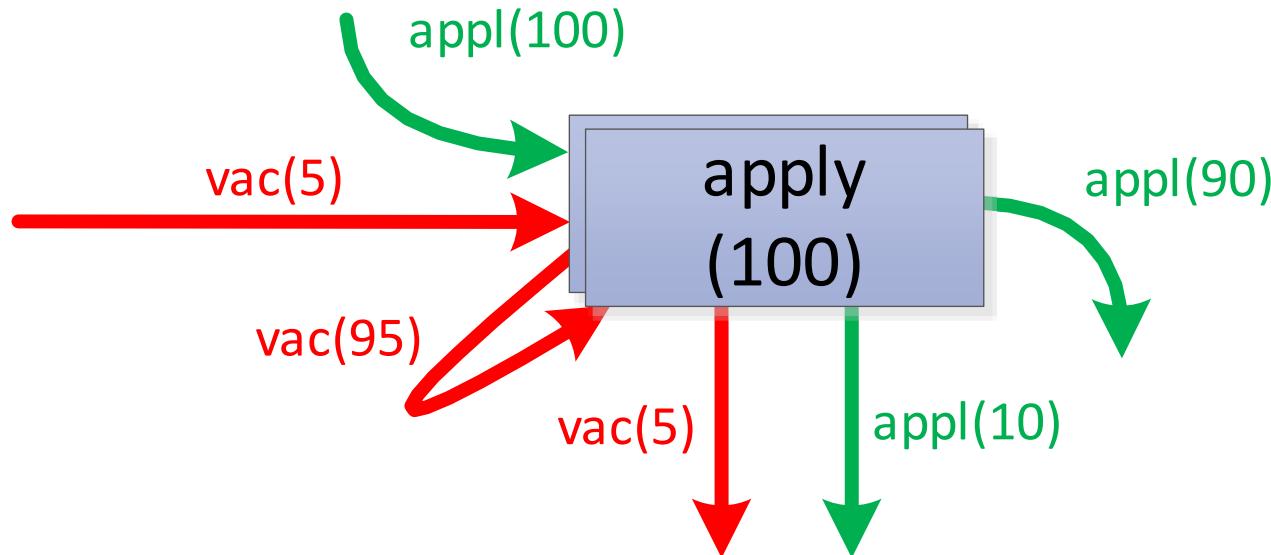


Complete model



Because of the design choices numbers add up

(not always the case)



Note that there are $2 \times (2 \times 2) = 8$ possible execution modes.
Delays and object-related statistics can be computed trivially.

$$\begin{aligned} \text{vac: } 5 + 95 &= 100 = 95 + 5 \\ \text{appl: } 100 &= 100 = 90 + 10 \end{aligned}$$

The paper

Object-Centric Process Mining: Dealing With Divergence and Convergence in Event Data

Wil M.P. van der Aalst^{1,2}[0000-0002-098x-6940]

¹ Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany
² Fraunhofer Institute for Applied Information Technology, Sankt Augustin, Germany
wdaalst@pads.rwth-aachen.de

Abstract. Process mining techniques use event data to answer a variety of process-related questions. Process discovery, conformance checking, model enhancement, and operational support are used to improve performance and compliance. Process mining starts from recorded events that are each characterized by a case identifier, an activity name, a timestamp, and optional parameters. In many cases, multiple events for the same case are described by different identifiers leading to different views on the same process. Moreover, one event may be related to different cases (convergence) and, for a given case, there may be multiple instances of the same activity within a case (divergence). To create a traditional process model, the event data need to be “flattened”. There are typically multiple choices possible, leading to different views that are disconnected. Therefore, one needs to look at divergence and convergence data need to be exacted multi times (from the different views). Different approaches have been proposed to tackle the problem. This paper discusses the gap between real event data and the event logs required by traditional process mining techniques. The main purpose is to create awareness and to provide ways to characterize event data. A specific logging format is proposed where events can be related to objects of different types. Moreover, basic notations and a baseline discovery approach are presented to facilitate discussion and understanding.

Keywords: Process Mining · Process Discovery · Divergence · Convergence · Artifact-Centric Modeling.

1 Introduction

Operational processes are often characterized by the 80/20 rule, also known as the Pareto principle. Often, 80% of the observed process executions (cases) can be described by less than 20% of the observed process variants. This implies that the remaining 20% of the observed process executions account for 80% of the observed process variants. Therefore, it is often relatively easy to create a precise and simple process model describing 80% of the cases. However, to add the remaining 20% of the cases, discovery techniques create models that are either complex and overfitting or severely underfitting. Standard processes such as the *Purchase-to-Pay* (*P2P*) and *Order-to-Cash* (*O2C*) seem simple at first: Just a

Wil M. P. van der Aalst: Object-Centric Process Mining: Dealing with Divergence and Convergence in Event Data. Software Engineering and Formal Methods - 17th International Conference, SEFM 2019, Oslo, Norway, September 18-20, 2019, Proceedings. Lecture Notes in Computer Science 11724, Springer 2019, ISBN 978-3-030-30445-43-25



Chair of Process
and Data Science

Related work

- W.M.P. van der Aalst. Object-Centric Process Mining: Dealing With Divergence and Convergence in Event Data. In P.C. Ölveczky and G. Salaün, editors, Software Engineering and Formal Methods (SEFM 2019), volume 11724 of Lecture Notes in Computer Science, pages 1-23. Springer-Verlag, Berlin, 2019.
- W.M.P. van der Aalst, P. Barthelmeß, C.A. Ellis, and J. Wainer. Proctels: A Framework for Lightweight Interacting Workflow Processes. International Journal of Cooperative Information Systems, 10(4):443-482, 2001.
- W.M.P. van der Aalst, G. Li, and M. Montali. Object-Centric Behavioral Constraints. CoRR, abs/1703.05740, 2017.
- A. Berti and W.M.P. van der Aalst. StarStar Models: Using Events at Database Level for Process Analysis. In P. Ceravolo, M.T. Gomez Lopez, and M. van Keulen, editors, International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2018), volume 2270 of CEUR Workshop Proceedings, pages 60-64. CEUR-WS.org, 2018.
- **Alessandro Berti and Wil van der Aalst. Discovering Multiple Viewpoint Models from Relational Databases. In P. Ceravolo, M.T. Gomez Lopez, and M. van Keulen, editors, Postproceedings International Symposium on Data-driven Process Discovery and Analysis, Lecture Notes in Business Information Processing. Springer-Verlag, Berlin, 2019.**
- E. González López de Murillas, H.A. Reijers, and W.M.P. van der Aalst. Connecting Databases with Process Mining: A Meta Model and Toolset. In R. Schmidt, W. Guedria, I. Bider, and S. Guerreiro, editors, Enterprise, Business-Process and Information Systems Modeling (BPMDS 2015), volume 248 of Lecture Notes in Business Information Processing, pages 231-249. Springer-Verlag, Berlin, 2016.
- M.L. van Eck, N. Sidorova, and W.M.P. van der Aalst. Guided Interaction Exploration in Artifact-centric Process Models. In IEEE Conference on Business Informatics (CBI 2017), pages 109-118. IEEE Computer Society, 2017.
- D. Fahland. Describing Behavior of Processes with Many-to-Many Interactions. In S. Donatelli and S. Haar, editors, Applications and Theory of Petri Nets 2019, volume 11522 of Lecture Notes in Computer Science, pages 3-24. Springer-Verlag, Berlin, 2019.
- D. Fahland, M. De Leoni, B. van Dongen, and W.M.P. van der Aalst. Behavioral Conformance of Artifact-Centric Process Models. In A. Abramowicz, editor, Business Information Systems (BIS 2011), volume 87 of Lecture Notes in Business Information Processing, pages 37-49. Springer-Verlag, Berlin, 2011.
- G. Li, E. González López de Murillas, R. Medeiros de Carvalho, and W.M.P. van der Aalst. Extracting Object-Centric Event Logs to Support Process Mining on Databases. In J. Mendling and H. Mouratidis, editors, Information Systems in the Big Data Era, CAiSE Forum 2018, volume 317 of Lecture Notes in Business Information Processing, pages 182-199. Springer-Verlag, Berlin, 2018.
- G. Li, R. Medeiros de Carvalho, and W.M.P. van der Aalst. Automatic Discovery of Object-Centric Behavioral Constraint Models. In W. Abramowicz, editor, Business Information Systems (BIS 2017), volume 288 of Lecture Notes in Business Information Processing, pages 43-58. Springer-Verlag, Berlin, 2017.
- X. Lu, M. Nagelkerke, D. van de Wiel, and D. Fahland. Discovering Interacting Artifacts from ERP Systems. IEEE Transactions on Services Computing, 8(6):861-873, 2015.
- M.L. van Eck, N. Sidorova, and W.M.P. van der Aalst. Multi-instance Mining: Discovering Synchronisation in Artifact-Centric Processes. In F. Daniel, Q.Z. Sheng, and H. Motahari, editors, Business Process Management Workshops, International Workshop on Business Process Intelligence (BPI 2018), volume 342 of Lecture Notes in Business Information Processing, pages 18-30. Springer-Verlag, Berlin, 2018.



Discovering Object-Centric Petri Nets



extract from
data sources

Each row corresponds to an event which
refers to one activity and any number of
objects of different types).

activity	time	orders	items	customers	products	price	weight
1. delivery	2019-12-24 17:27:18	991269	{8849494,884995,884996,884997,884998}	[Wil van der Aalt], [Christine Döbrent], [Kefeng Ding], [Tobias Brockhoff], [Mehnaz Bafareh], [Eduardo Sandoval], [Wilma Bader]	[Fire Stick 4K,Echo Show 5,Echo Plus,iPad mini,iPhone 8]	€ 933,96	3,70
13501 exec item	2019-12-24 17:33:55	9912691	{884956}	[Wil van der Aalt]	[Kindle Paperwhite]	€ 120,00	0,50
13503 exec item	2019-12-24 17:34:00	9912690	{884954}	[Wil van der Aalt]	[Fire Stick 4K]	€ 120,00	0,50
13504 exec item	2019-12-24 18:05:38	9912523	{884953}	[Wil van der Aalt]	[Kindle Paperwhite]	€ 120,00	0,50
13505 pay order	2019-12-24 18:05:49	9912526	{884817,884858,884859,884860}	[Mehnaz Bafareh]	[Echo Studio,Kindle,MacBook Pro,iPad,iPhone]	€ 583,00	
13506 exec item	2019-12-24 18:18:30	9912517	{884864}	[Wil van der Aalt]	[Echo Studio,Kindle]	€ 583,00	
13507 exec item	2019-12-24 18:20:23	9912523	{884865}	[Wil van der Aalt]	[Kindle Paperwhite]	€ 120,00	0,50
13508 exec item	2019-12-24 18:33:59	9912528	{884840}	[Wil van der Aalt]	[iPad mini]	€ 449,00	0,28
13509 pay order	2019-12-24 18:47:59	9912511	{884916,884957,884958,884959,884960,884961,884962}	[Wil van der Aalt]	[Kindle Paperwhite,iPad,Kindle,Echo Show 5,Echo Studio,iPhone]	€ 1,168,95	1,04
13510 exec item	2019-12-24 19:01:21	9912510	{884903}	[Wil van der Aalt]	[iPhone X]	€ 579,00	0,21
13511 exec item	2019-12-24 20:39:01	9912524	{884903}	[Wil van der Aalt]	[Kefeng Ding]	€ 4,629,99	4719,00
13512 package delivered	2019-12-24 20:46:47	991119,991030,991209,991254,991213,991206	{884386,884020,884749,884930,884926,884925,884766,884927,884736}	[Kefeng Ding]	[iPad Air,Dot,MacBook Pro,iPad Air,Kindle Paperwhite,iPad Air,iPad]	€ 4,629,99	4719,00
13513 exec item	2019-12-24 22:08:21	9912711	{885008,885010,885002,885003}	[Wil van der Aalt]	[iPhone 11 Pro,iPad,iPhone]	€ 1,168,95	1,04
13514 exec item	2019-12-24 22:08:22	9912711	{885008,885010,885002,885003}	[Wil van der Aalt]	[iPhone 11 Pro,iPad,iPhone]	€ 1,168,95	1,04
13515 exec item	2019-12-24 22:08:23	9912711	{885008,885010,885002,885003}	[Wil van der Aalt]	[iPhone 11 Pro,iPad,iPhone]	€ 1,168,95	1,04
13516 confirm order	2019-12-24 22:08:24	9912711	{885008,885010,885002,885003}	[Wil van der Aalt]	[iPhone 11 Pro,iPad,iPhone]	€ 1,168,95	1,04
13517 pay order	2019-12-24 22:08:25	9912711	{885008,885010,885002,885003}	[Wil van der Aalt]	[iPhone 11 Pro,iPad,iPhone]	€ 1,168,95	1,04
13518 package fail	2019-12-24 22:08:26	9912711	{885008,885010,885002,885003}	[Wil van der Aalt]	[iPhone 11 Pro,iPad,iPhone]	€ 1,168,95	1,04

object-centric Petri net

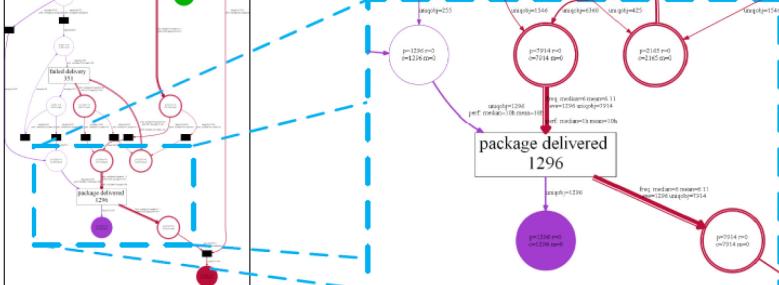
object-centric Petri net

automatically
discovered object-
centric Petri net

object-centric event log

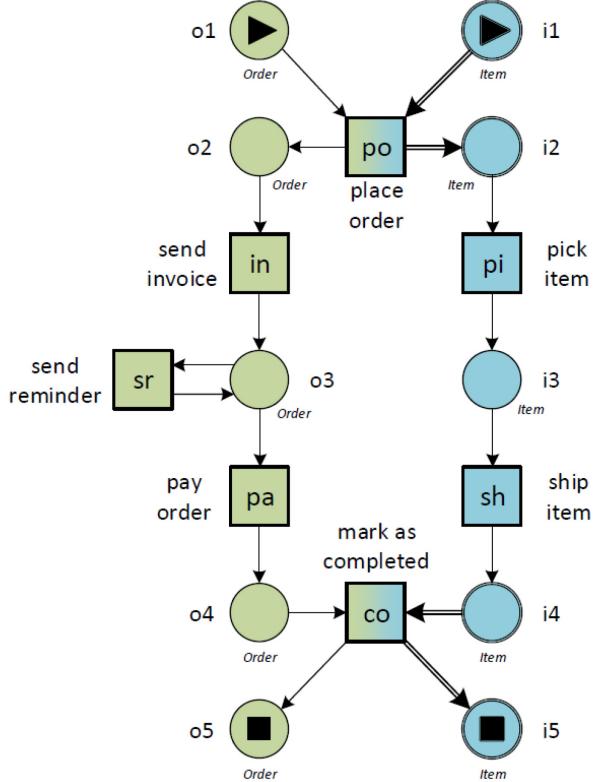
One of the 21887 events:

- activity: package delivered
- time: 2019-12-24 20:46:47
- orders involved: {991119,991030,991209,991254,991213,991206}
- items involved: {884386,884020,884749,884930,884926,884925,884766,884927,884736}
- packages involved: {660784}
- customers involved: {Kefeng Ding}
- products involved: {iPad Air,Echo Dot,MacBook Pro,iPad Air,Kindle Paperwhite,iPad Air,iPad,iPhone 11 Pro}
- total price: € 6.829,99
- total weight: 4,719 KG



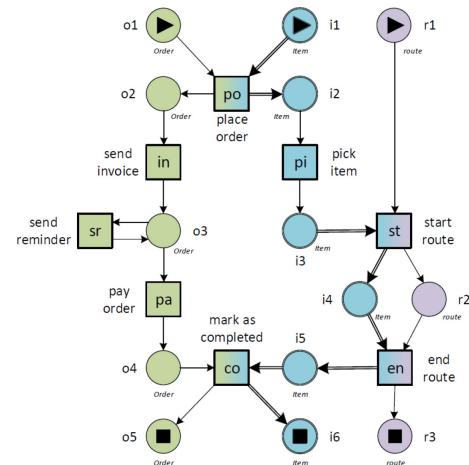
Example

activity	timestamp	order	item
...
place order	25-11-2019:09.35	{99001}	{88124, 88125, 88126}
pick item	25-11-2019:10.35	Ø	{88126}
place order	25-11-2019:11.35	{99002}	{88127, 88128}
pick item	26-11-2019:010.25	Ø	{88124}
send invoice	27-11-2019:08.12	{99001}	Ø
send invoice	28-11-2019:09.35	{99002}	Ø
pick item	29-11-2019:09.35	Ø	{88127}
send reminder	29-11-2019:10.35	{99002}	Ø
pick item	29-11-2019:11.15	Ø	{88128}
ship item	29-11-2019:12.35	Ø	{88124}
pick item	29-11-2019:13.30	Ø	{88125}
send reminder	29-11-2019:14.35	{99001}	Ø
ship item	29-11-2019:15.15	Ø	{88125}
send reminder	29-11-2019:16.15	{99002}	Ø
ship item	29-11-2019:17.45	Ø	{88126}
ship item	29-11-2019:18.00	Ø	{88128}
send reminder	30-11-2019:09.35	{99002}	Ø
ship item	30-11-2019:10.05	Ø	{88127}
pay order	30-11-2019:11.45	{99002}	Ø
pay order	30-11-2019:12.55	{99001}	Ø
mark as completed	01-12-2019:09.35	{99001}	{88124, 88125, 88126}
place order	02-12-2019:10.40	{99003}	{88129}
mark as completed	04-12-2019:11.05	{99002}	{88127, 88128}
place order	06-12-2019:14.18	{99004}	{88130, 88131, 88132, 88133, 88134}
...



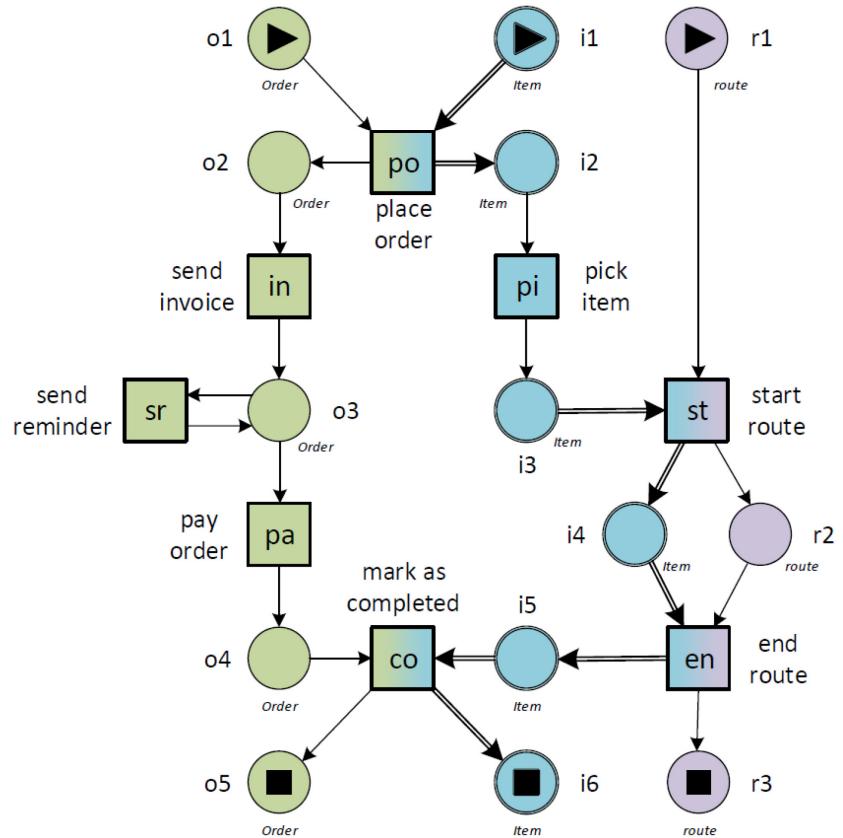
A bit more complicated

activity	timestamp	order	item	route
...
place order	25-11-2019:09.35	{99001}	{88124, 88125, 88126}	∅
place order	25-11-2019:11.35	{99002}	{88127, 88128}	∅
...
start route	25-11-2019:11.35	∅	{88124, 88127}	{66222}
end route	25-11-2019:11.35	∅	{88124, 88127}	{66222}
...
start route	25-11-2019:11.35	∅	{88125, 88126, 88128}	{66223}
end route	25-11-2019:11.35	∅	{88125, 88126, 88128}	{66223}
...
mark as completed	01-12-2019:09.35	{99001}	{88124, 88125, 88126}	∅
mark as completed	04-12-2019:11.05	{99002}	{88127, 88128}	∅
...

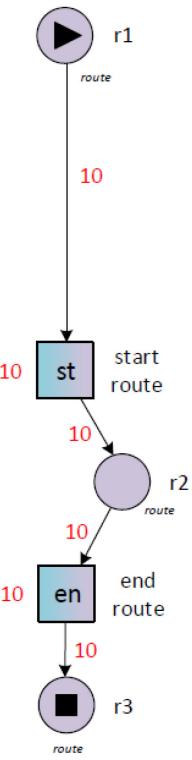
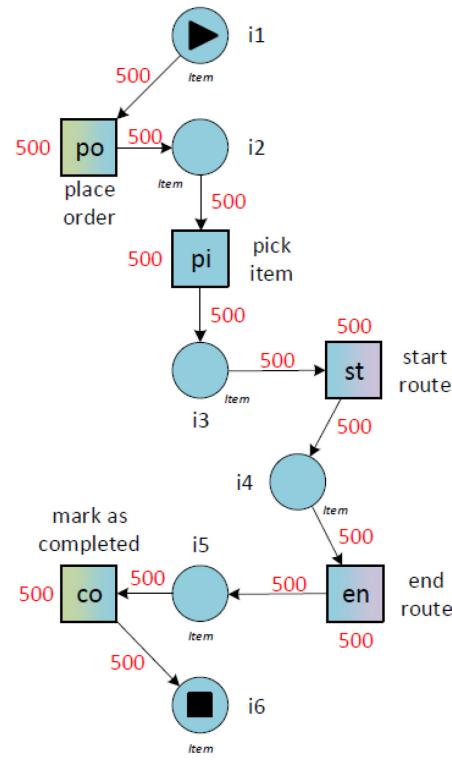
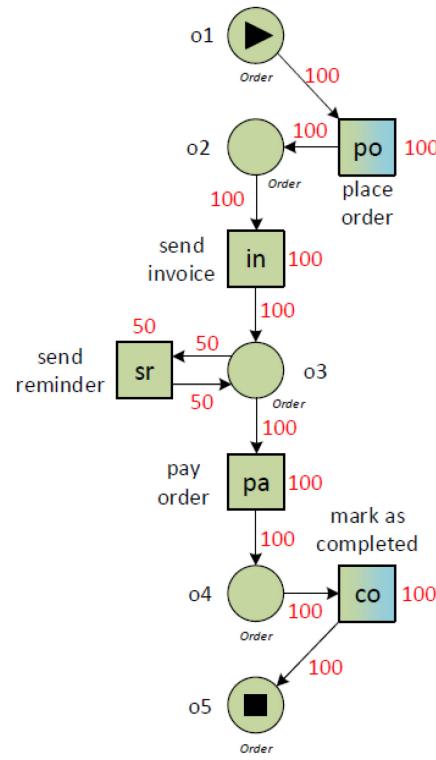


A bit more complicated

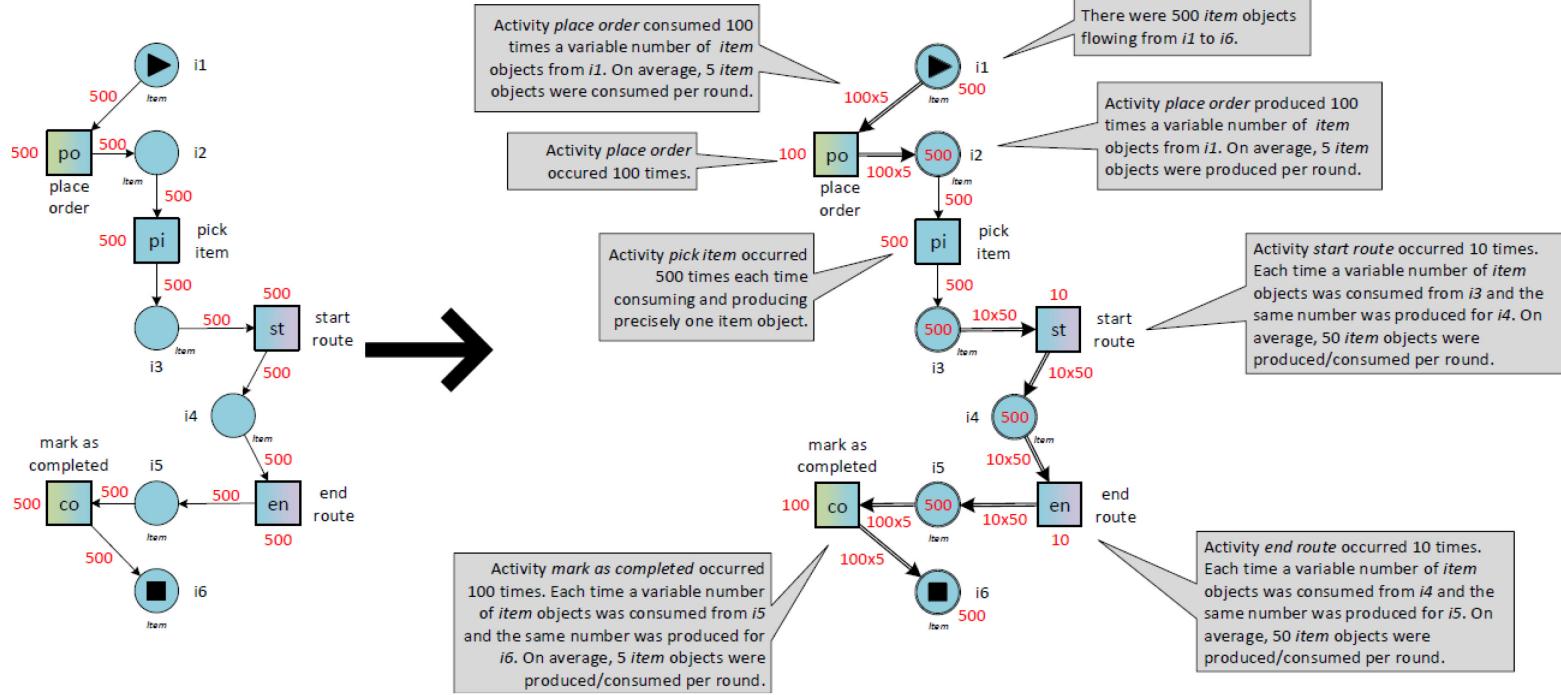
activity	timestamp	order	item	route
...
place order	25-11-2019:09.35	{99001}	{88124, 88125, 88126}	∅
place order	25-11-2019:11.35	{99002}	{88127, 88128}	∅
...
start route	25-11-2019:11.35	∅	{88124, 88127}	{66222}
end route	25-11-2019:11.35	∅	{88124, 88127}	{66222}
...
start route	25-11-2019:11.35	∅	{88125, 88126, 88128}	{66223}
end route	25-11-2019:11.35	∅	{88125, 88126, 88128}	{66223}
...
mark as completed	01-12-2019:09.35	{99001}	{88124, 88125, 88126}	∅
mark as completed	04-12-2019:11.05	{99002}	{88127, 88128}	∅
...



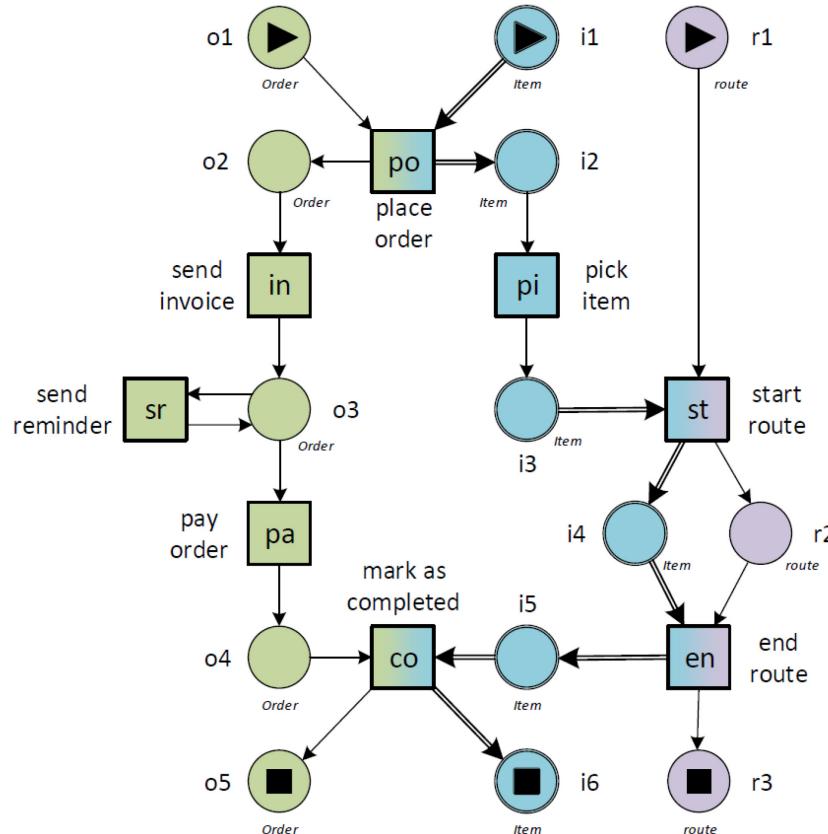
Merging accepting Petri nets



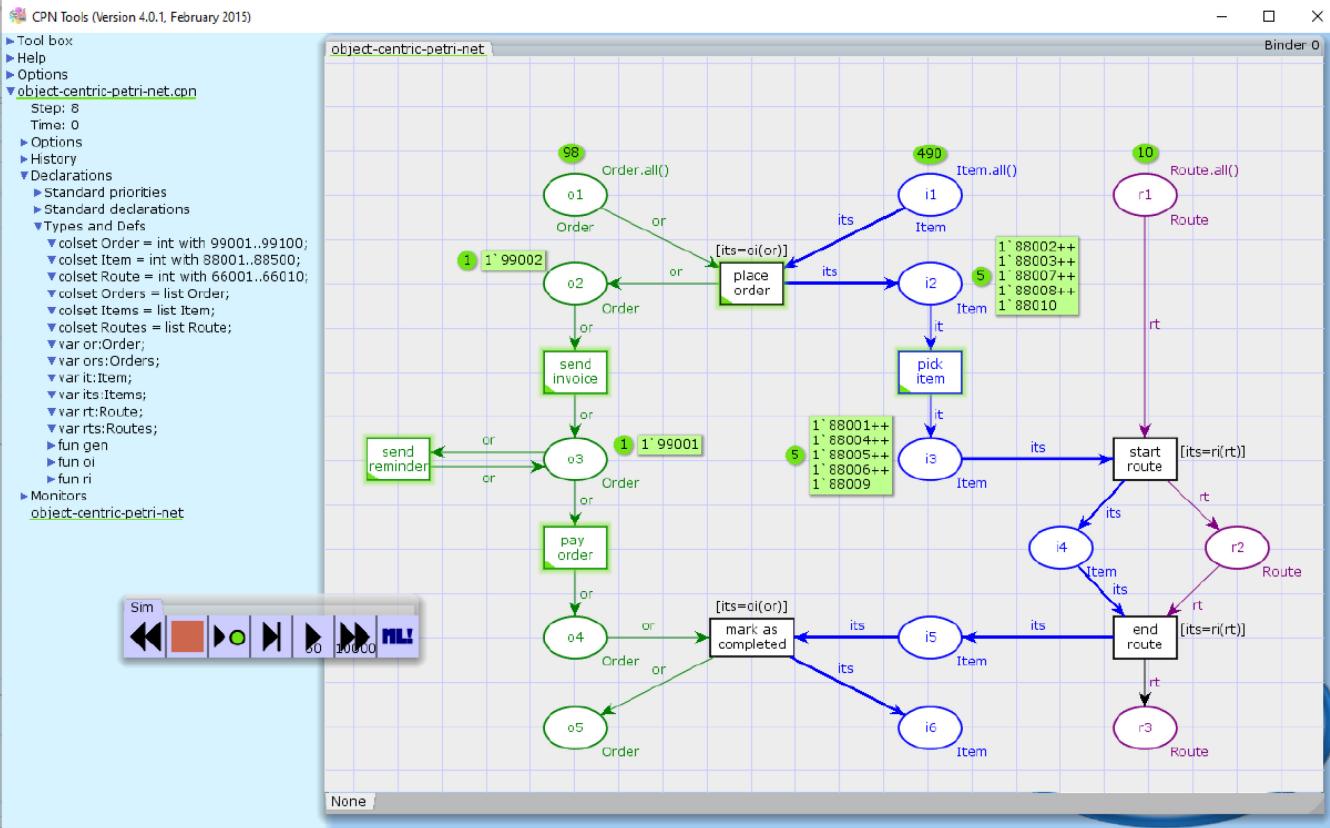
Merging accepting Petri nets



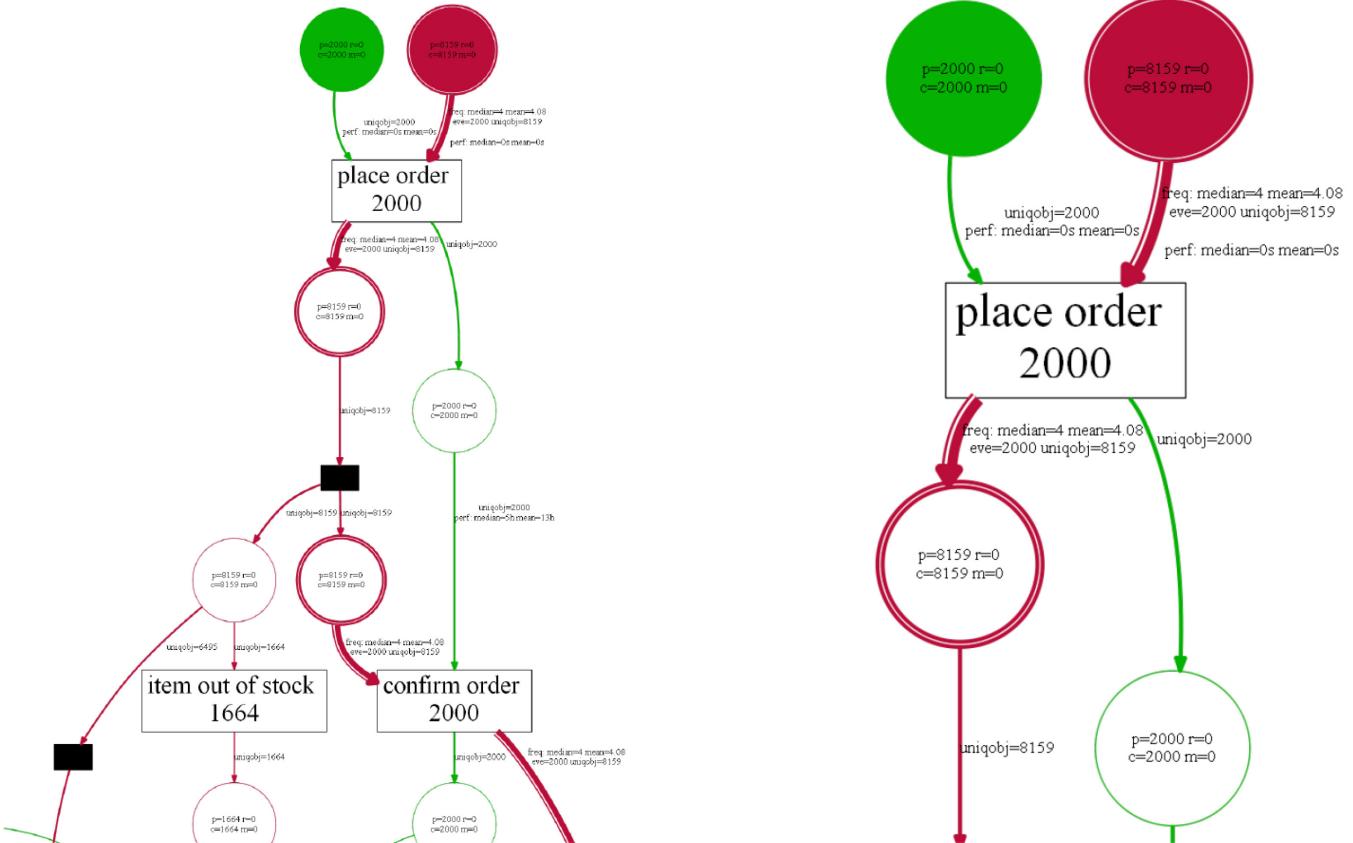
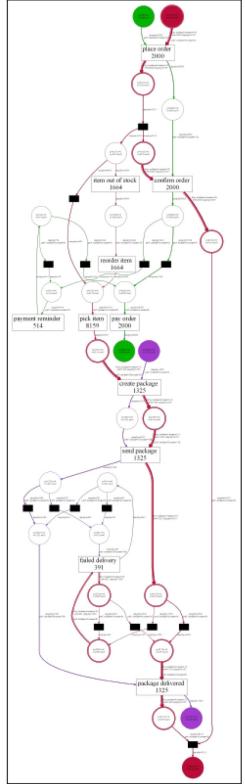
The result after merging



Related to good old CPNs



Implementation (by Alessandro Berti)



Implementation (by Alessandro Berti)

