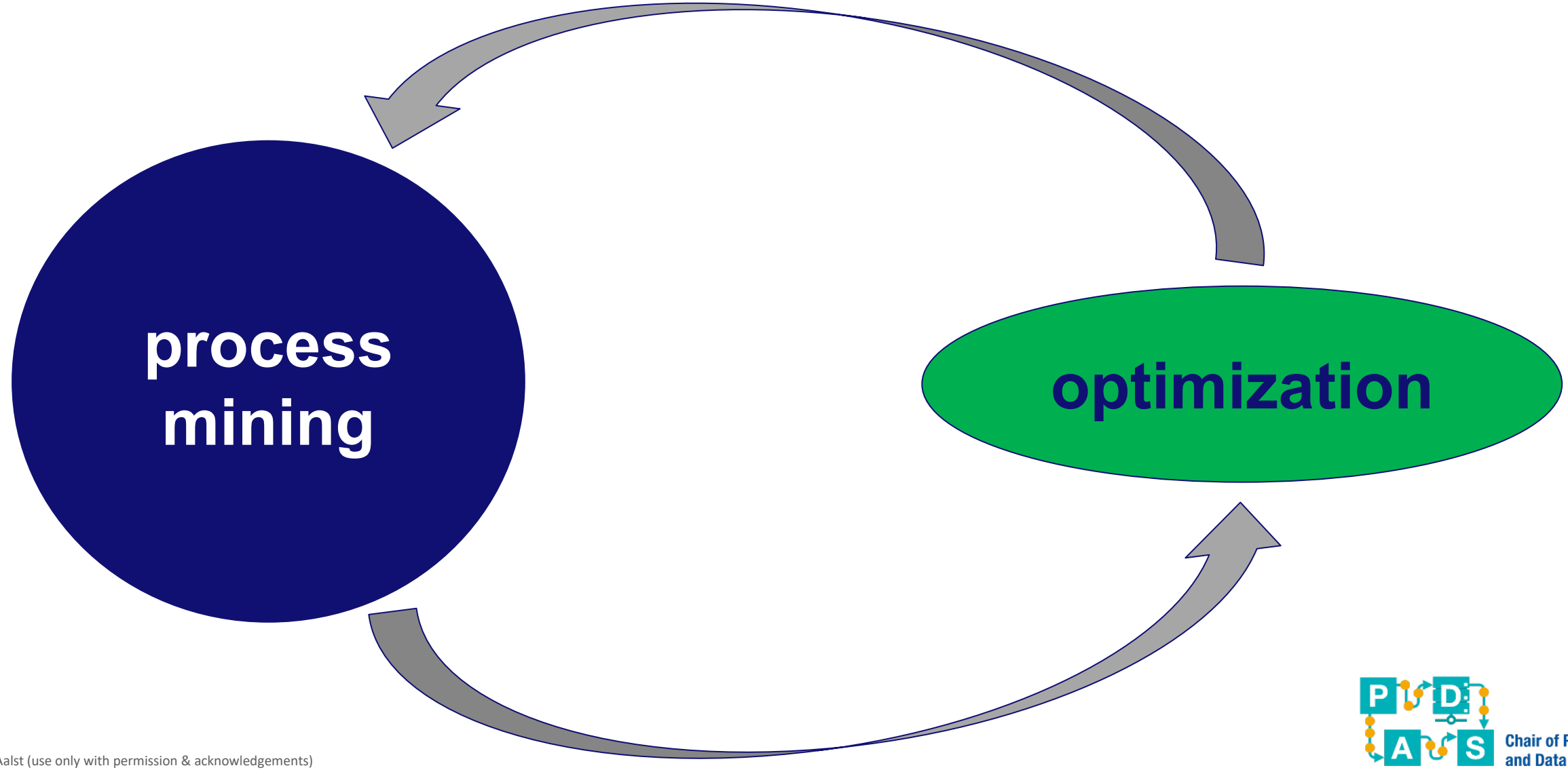
Process Mining for Optimization and Optimization for Process Mining

Wil van der Aalst
Process and Data Science @ RWTH Aachen University & Celonis

Outline



Chair of Process and Data Science

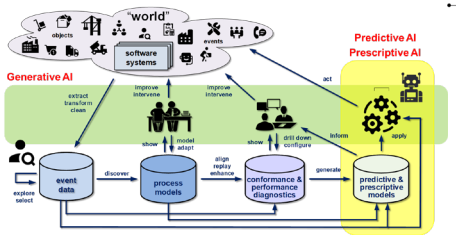
Outline

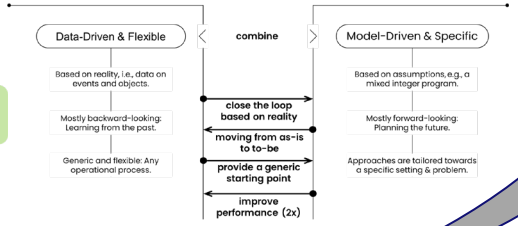
Conformance checking:
Alignments

Process discovery:
Inductive Mining and
ILP Mining

process mining

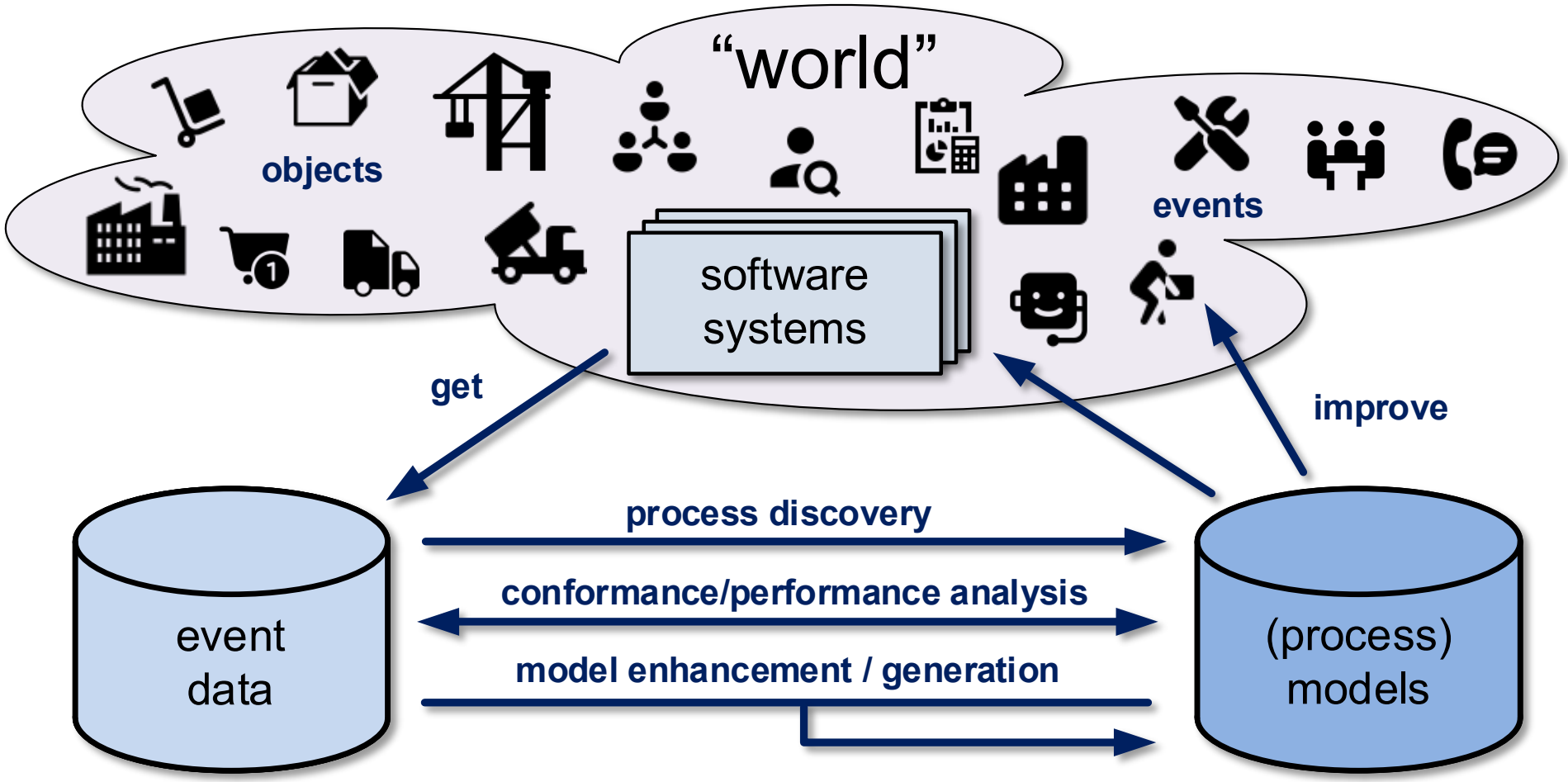
optimization





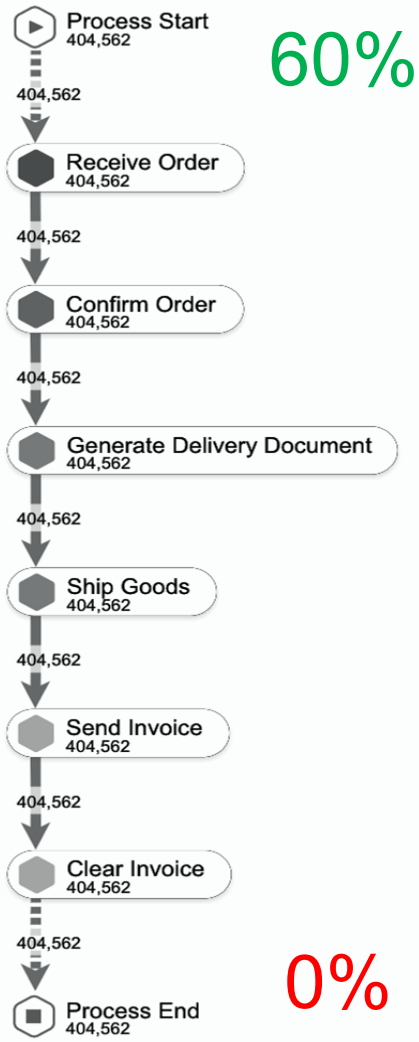
Chair of Process and Data Science

Process Mining: Overview

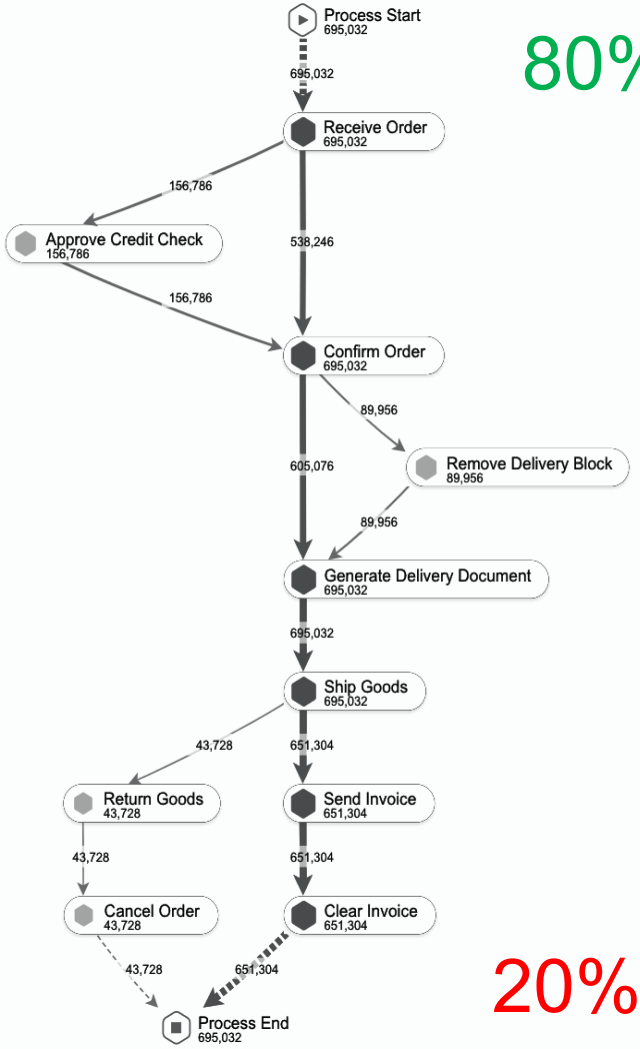


Chair of Process and Data Science

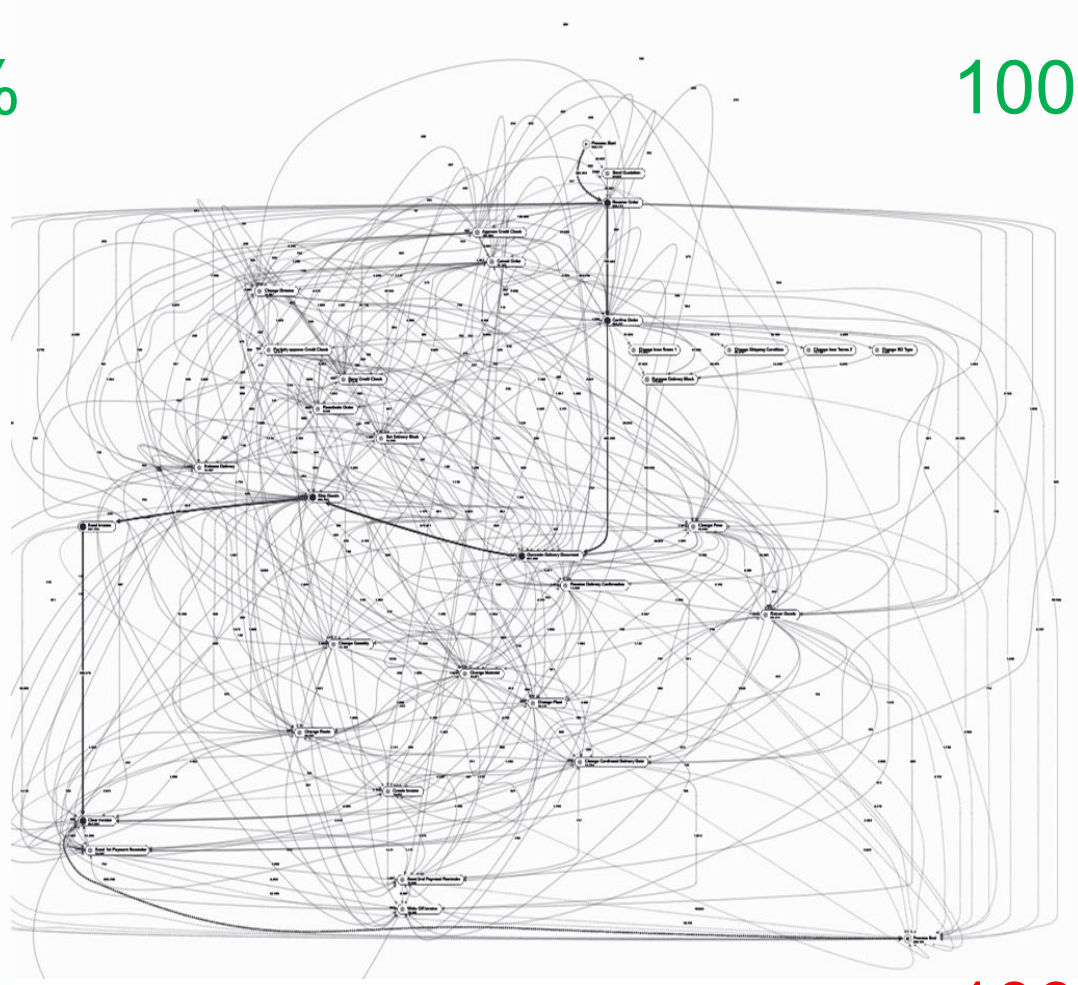
Actual processes are very different from what stakeholders think !



60%



80%



100%

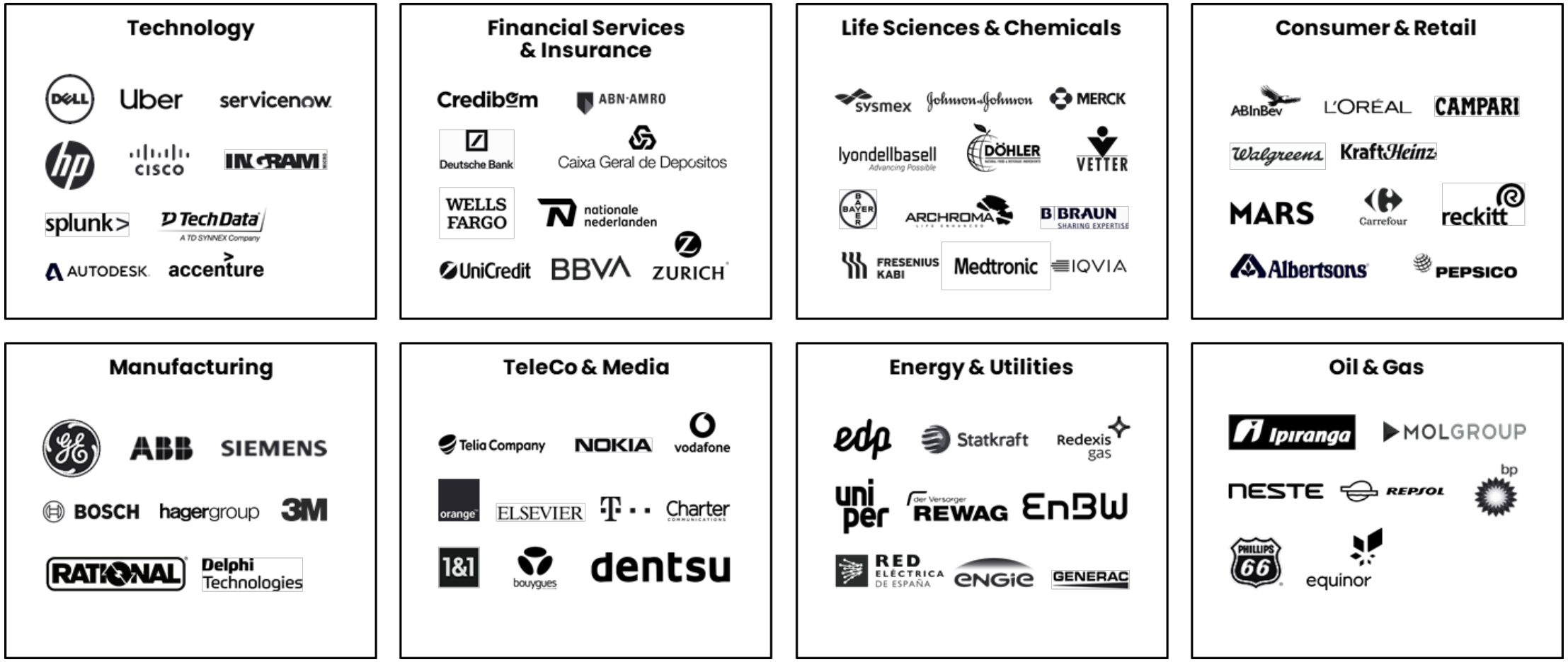
0%

20%

100%

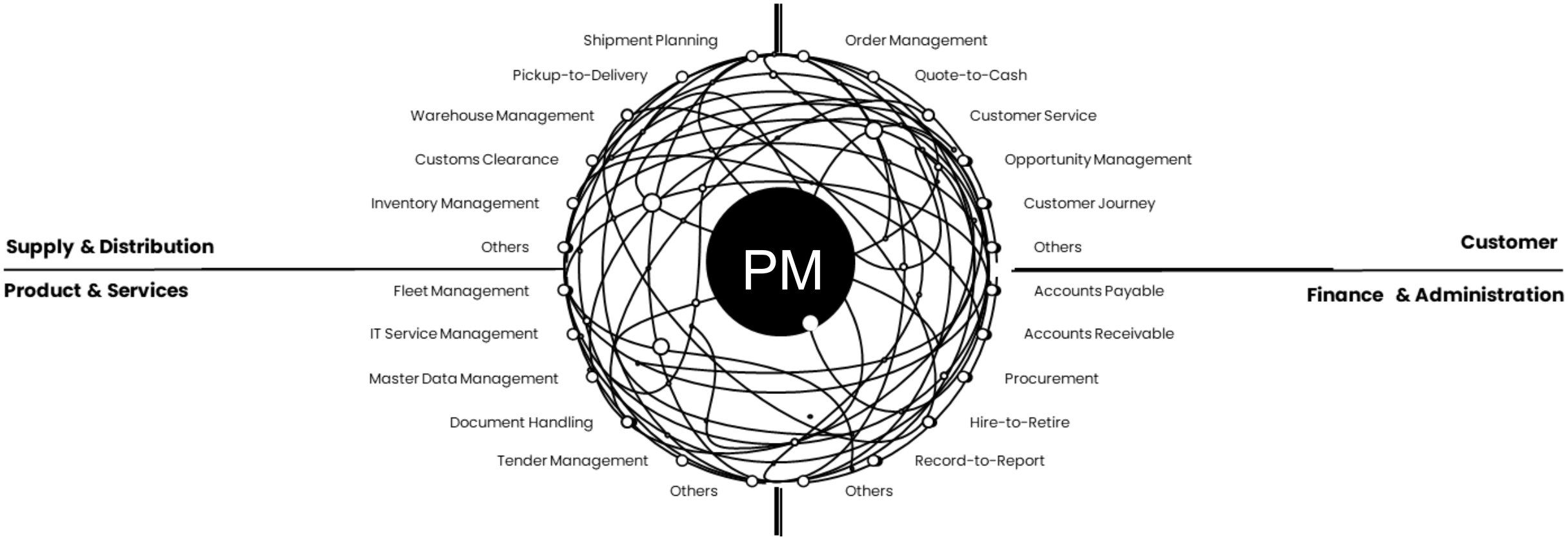
Chair of Process and Data Science

Use Cases Everywhere!



Chair of Process and Data Science

Use Cases Everywhere!

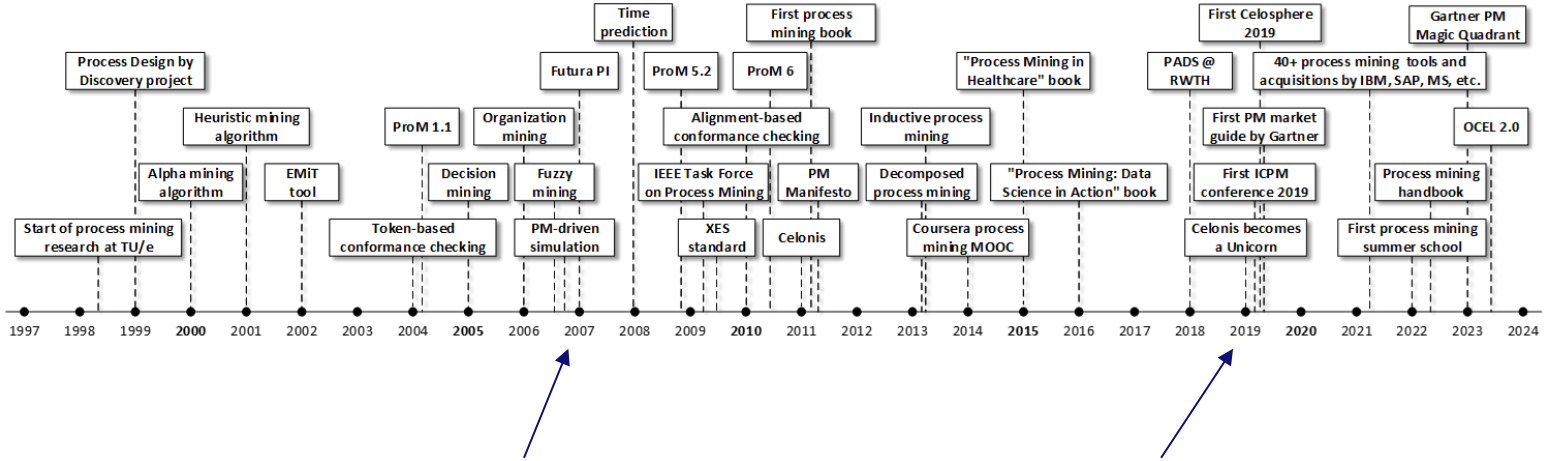


Chair of Process and Data Science

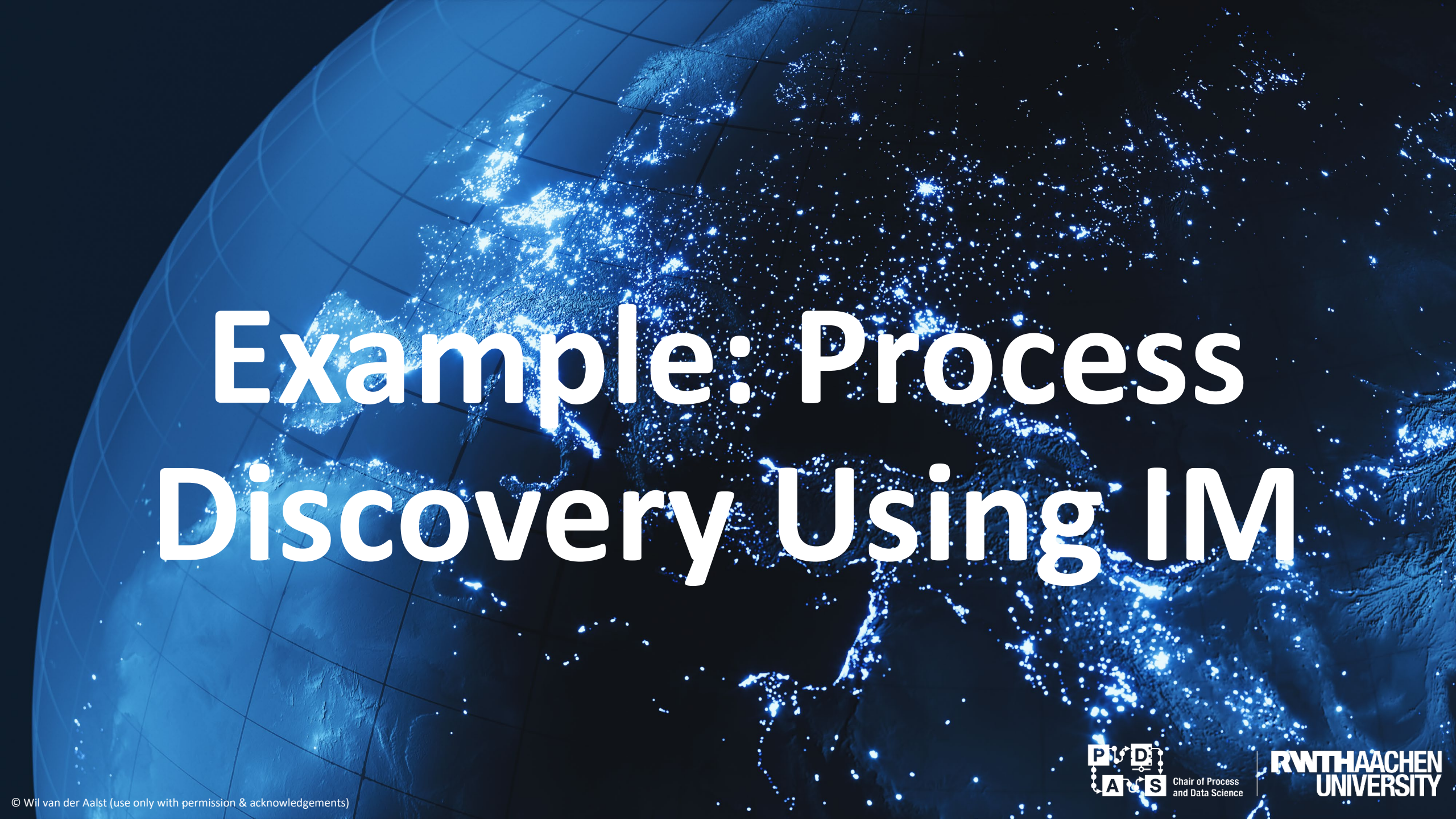

INFORM
celonis
VU
UNITED
lufthansa.com
D-ABVU
SP
Lufthansa
lufthansa.com
D-AISP
NA
First to fly A320neo
Less noise. Less fuel. Less CO_2.
D-AINA
Lufthansa

A new category of tools and scientific discipline …

Gartner Magic Quadrant



Chair of Process and Data Science

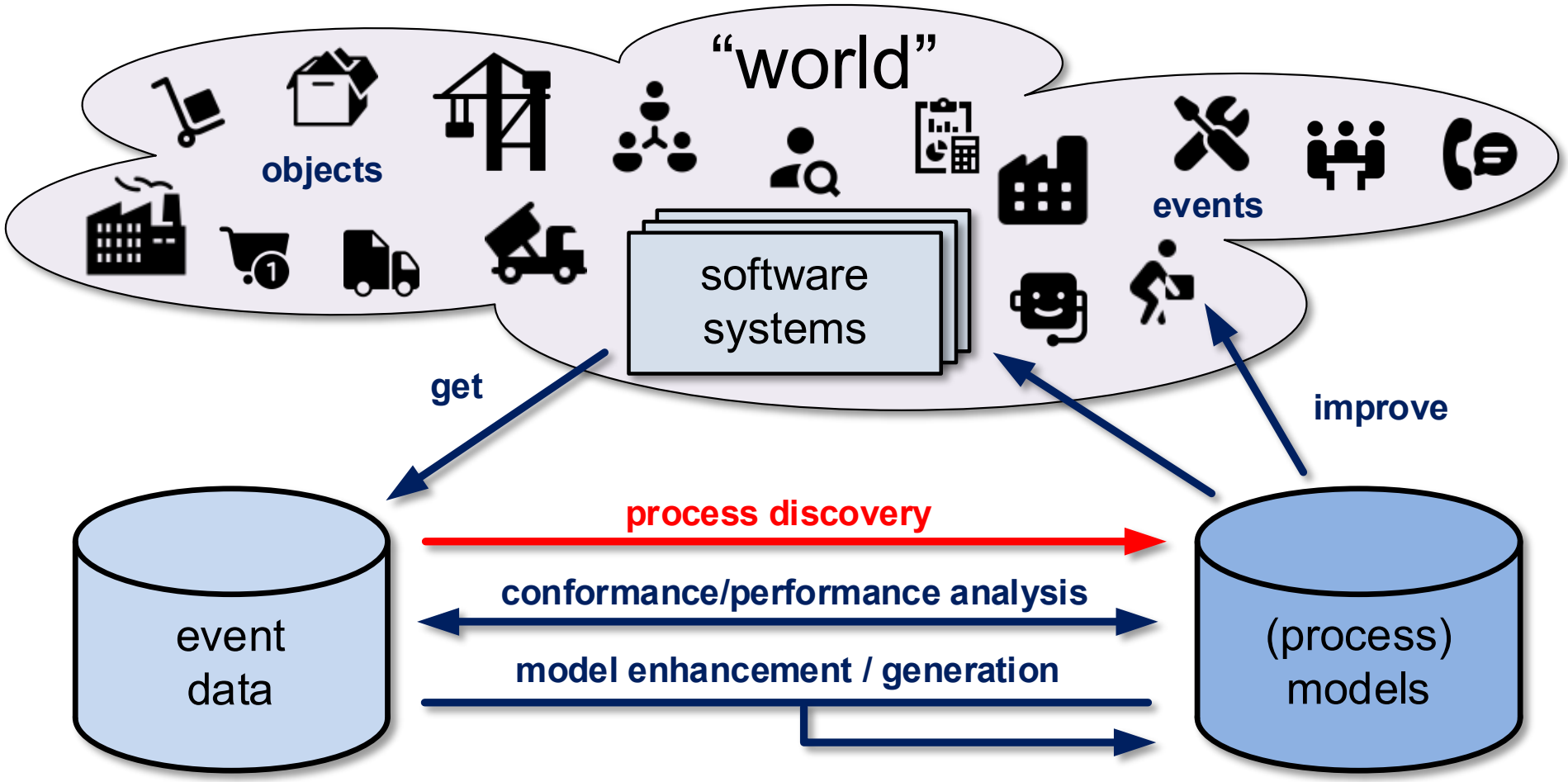


Example: Process Discovery Using IM

Chair of Process and Data Science

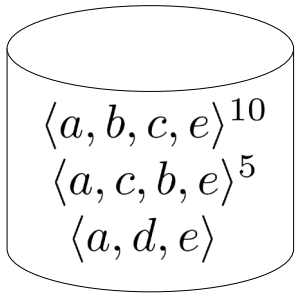
RWTH AACHEN UNIVERSITY

Process Discovery

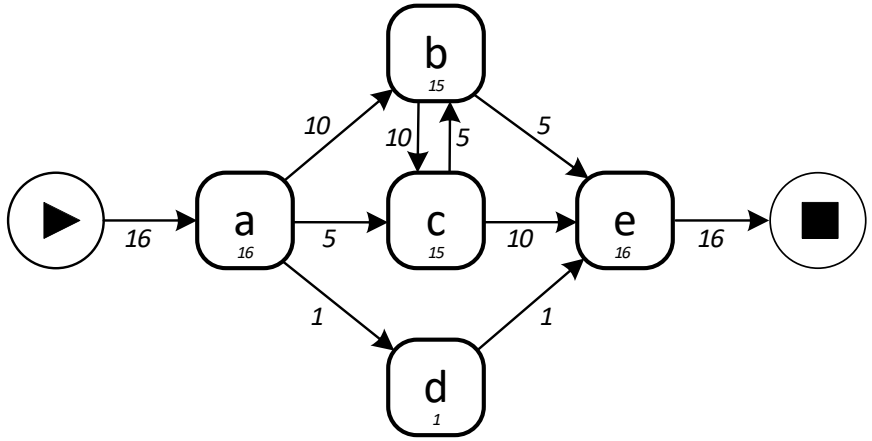


Chair of Process and Data Science

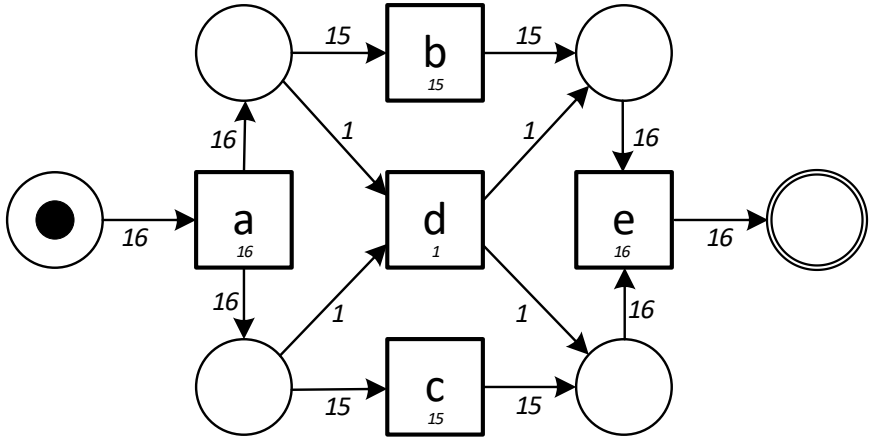
The main idea (informal)



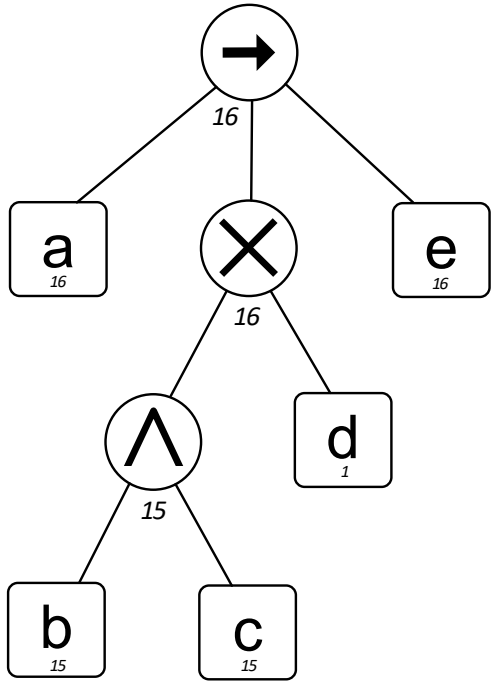
(a) Event log L_1



(b) Directly-Follows Graph (DFG): M_1



(c) Accepting Petri Net (APN): M_2



(d) Process Tree (PT): M_3

Chair of Process and Data Science

Inductive Mining (IM)

- Decompose the event log into smaller events logs until the problem get trivial.
- Four types of cuts corresponding to the operators: → (sequential composition), × (exclusive choice), ∧ (parallel composition), and ↺ (redo loop).
- In each step the activities are partitioned into subsets until they are singletons.

Developed by Sander Leemans in the context of his PhD thesis (NWO project "Don't Search for the Undesirable! Avoiding "Blind Alleys" in Process Mining" 2012-2017).

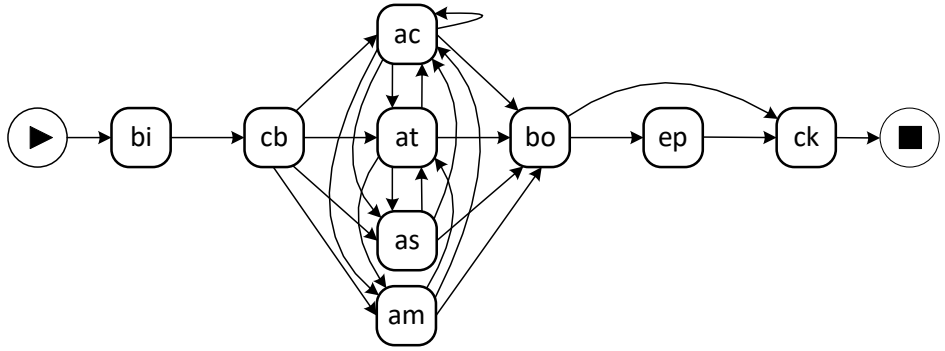

Presented at high speed: You only need to get the main idea!

Event log

Activities: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Create a DFG for the whole event log

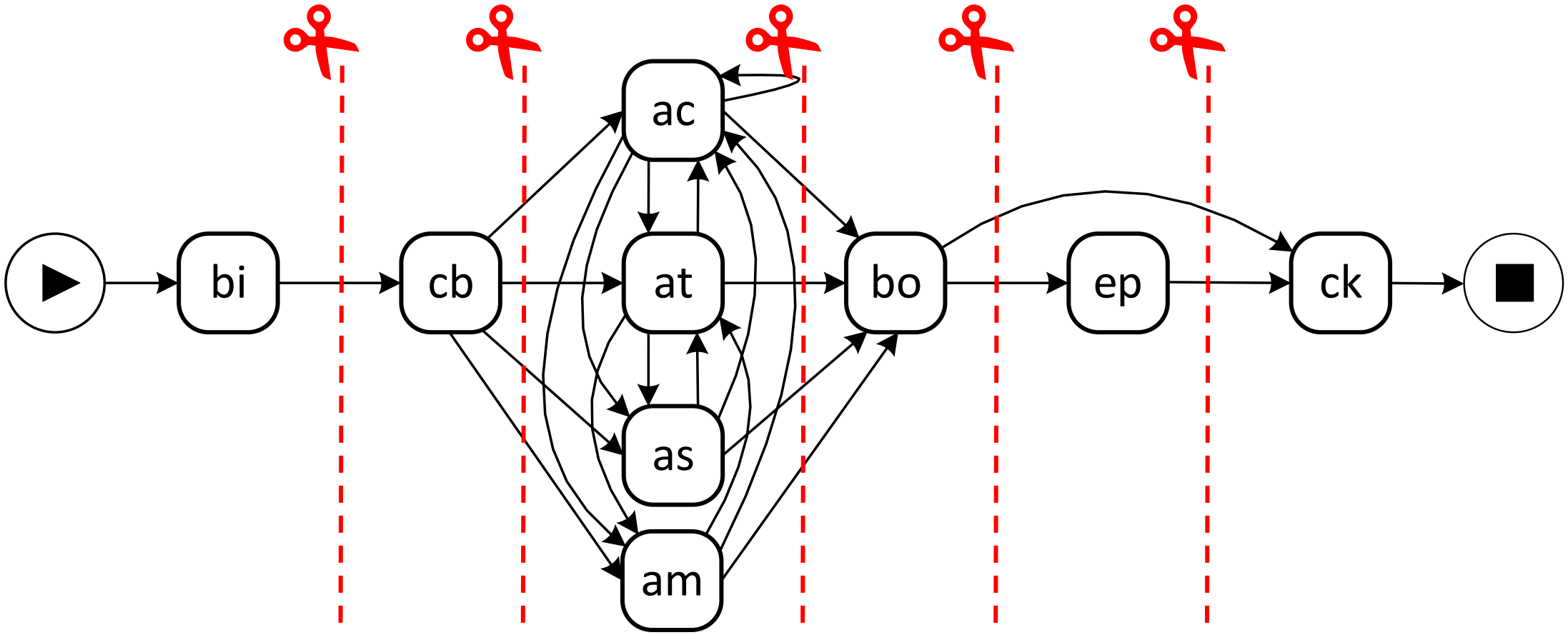
bi cb ac at as bo ep ck
bi cb ac as at bo ep ck
bi cb at ac as bo ep ck
bi cb at as ac bo ck
bi cb as ac at bo ep ck
bi cb as at ac bo ep ck
bi cb ac at am bo ep ck
bi cb ac am at bo ep ck
bi cb at ac am bo ck
bi cb at am ac bo ep ck
bi cb am ac at bo ep ck
bi cb am at ac bo ep ck
bi cb at as ac ac bo ep ck
bi cb as ac ac at ac bo ck
bi cb as at ac ac bo ep ck



Frequencies omitted for readability

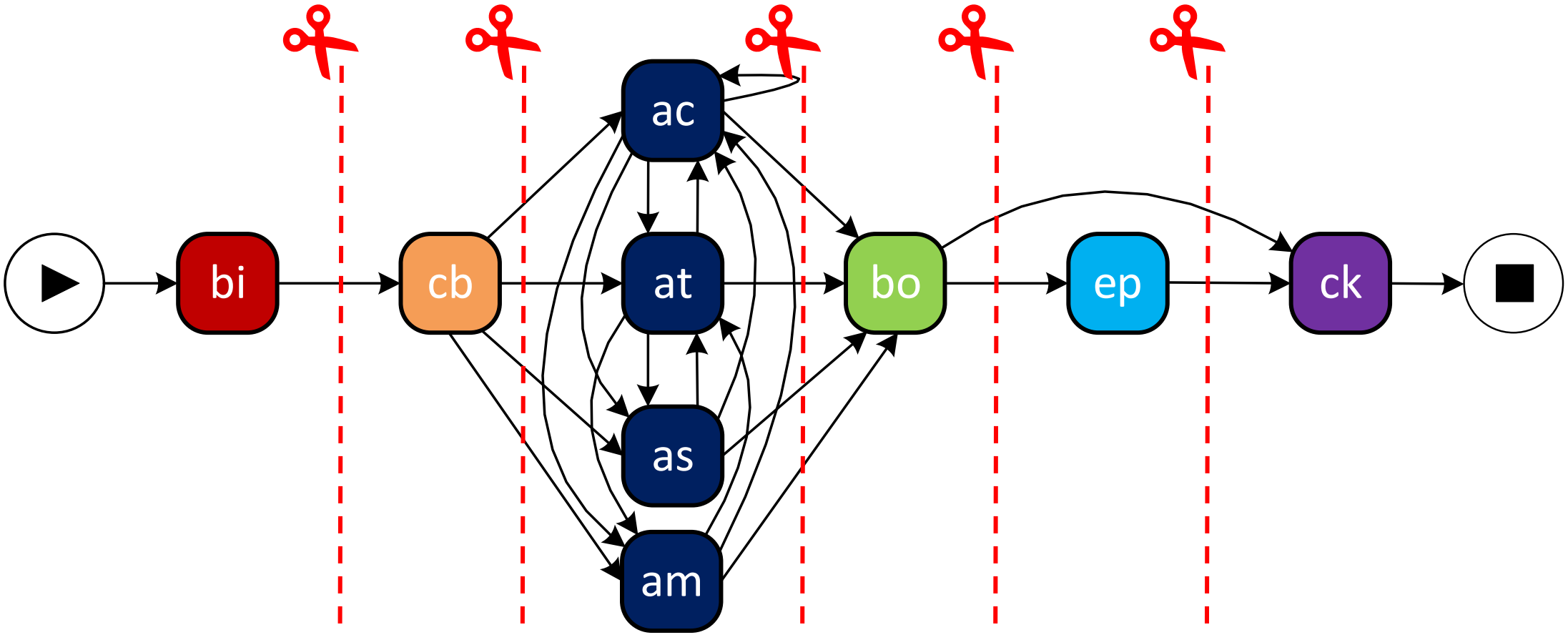
Chair of Process and Data Science

Apply a sequence cut



There is a sequence cut when the DFG can be split into sequential parts where only “forward connections” are possible. Note that we need to use the non-reflexive transitive closure of F.

Sequence cut partitions activities in six subsets

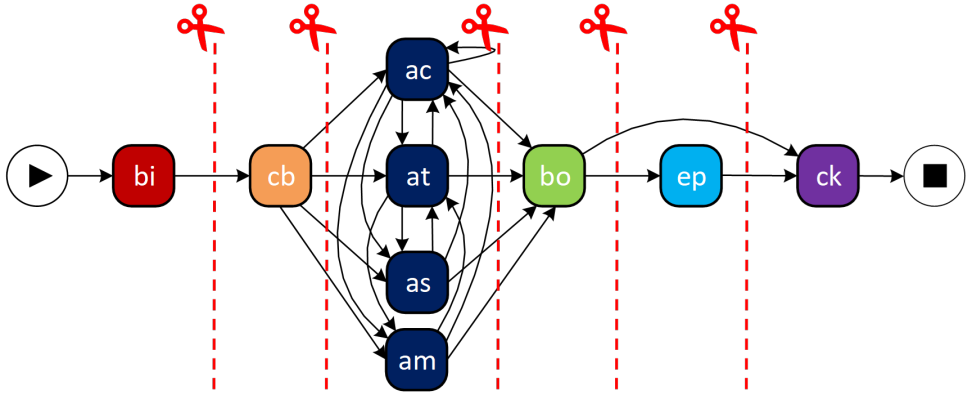


Chair of Process and Data Science

Color the events based on the partitioning

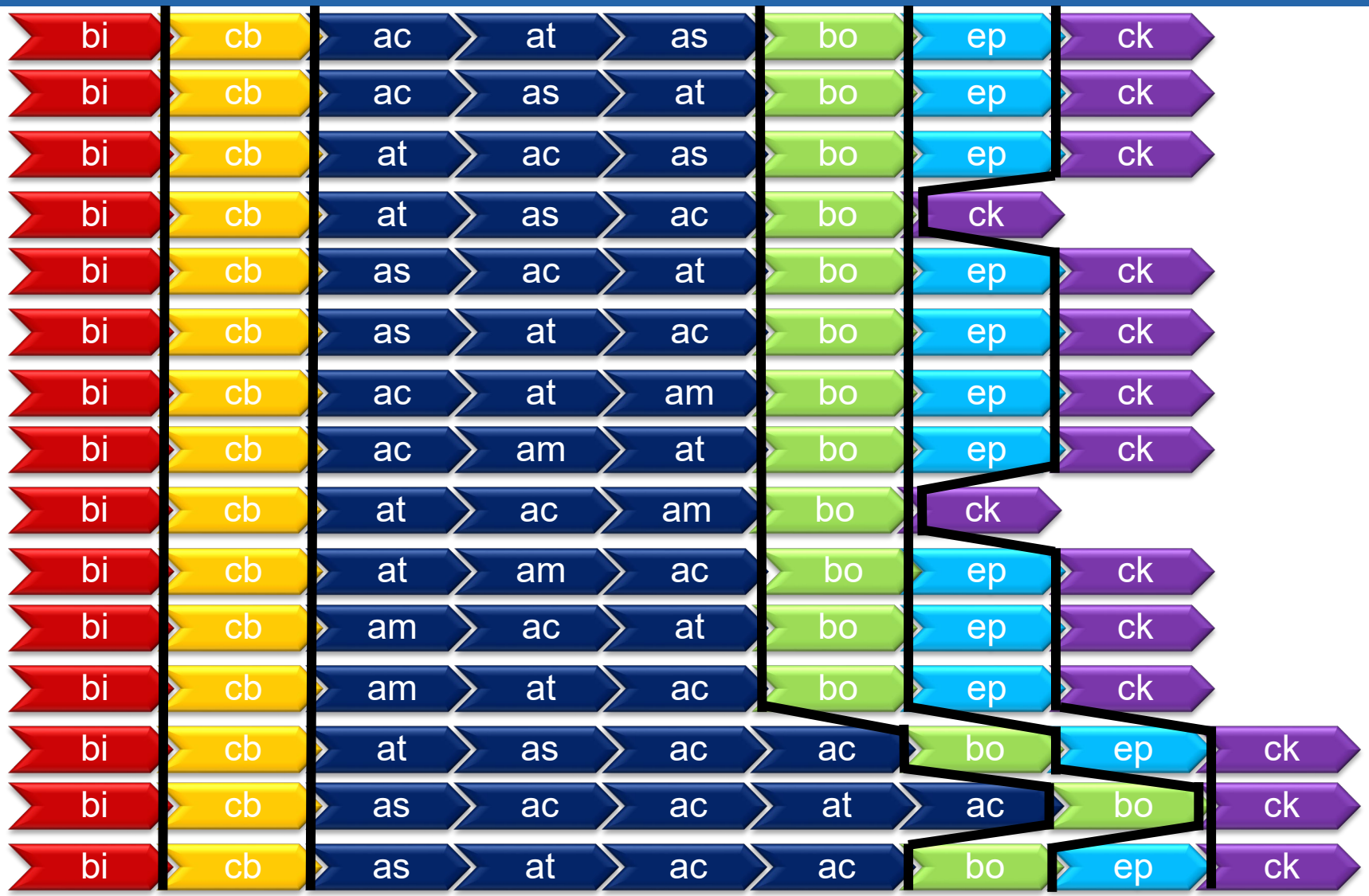
bi cb ac at as bo ep ck
bi cb ac as at bo ep ck
bi cb at ac as bo ep ck
bi cb at as ac bo ck
bi cb as ac at bo ep ck
bi cb as at ac bo ep ck
bi cb ac at am bo ep ck
bi cb ac am at bo ep ck
bi cb at ac am bo ck
bi cb at am ac bo ep ck
bi cb am ac at bo ep ck
bi cb am at ac bo ep ck
bi cb at as ac ac bo ep ck
bi cb as ac ac at ac bo ck
bi cb as at ac ac bo ep ck

Sequence cut



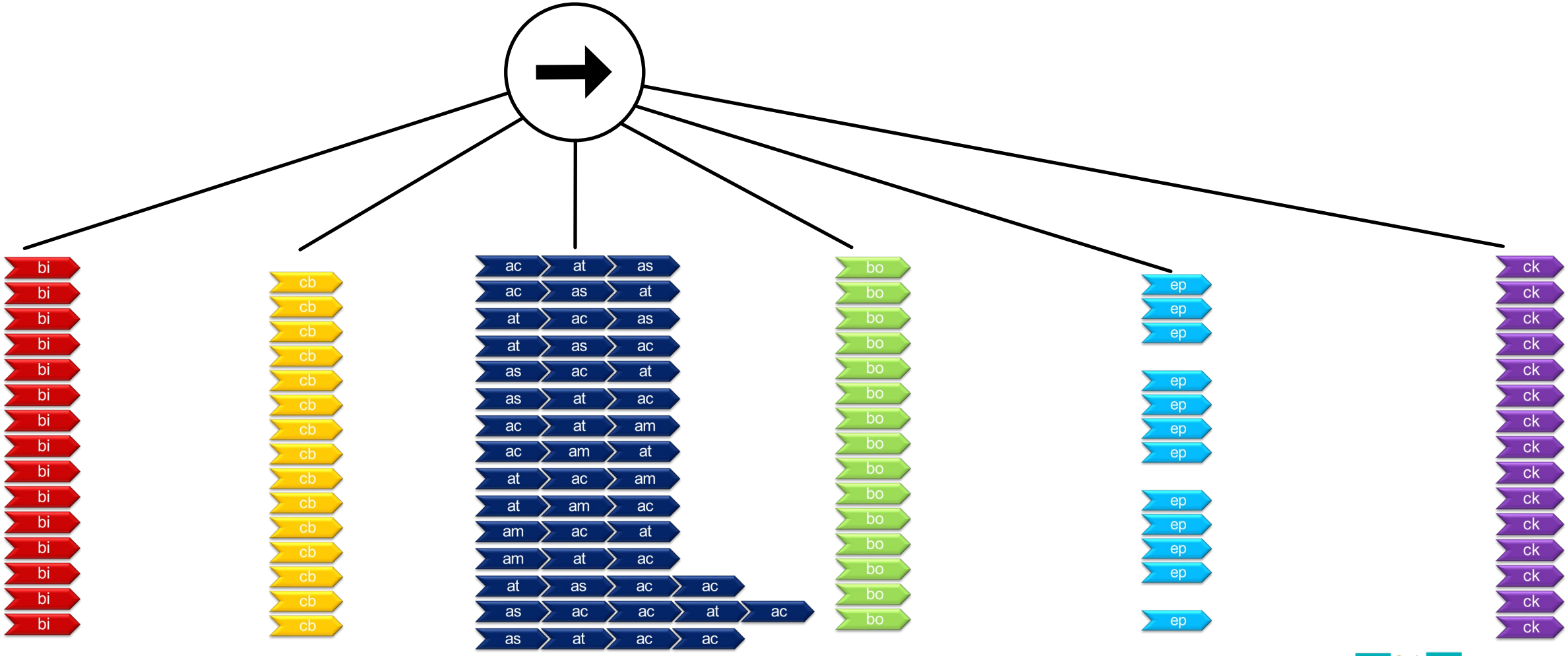
Chair of Process and Data Science

Split the event log based on the partitioning



Chair of Process and Data Science

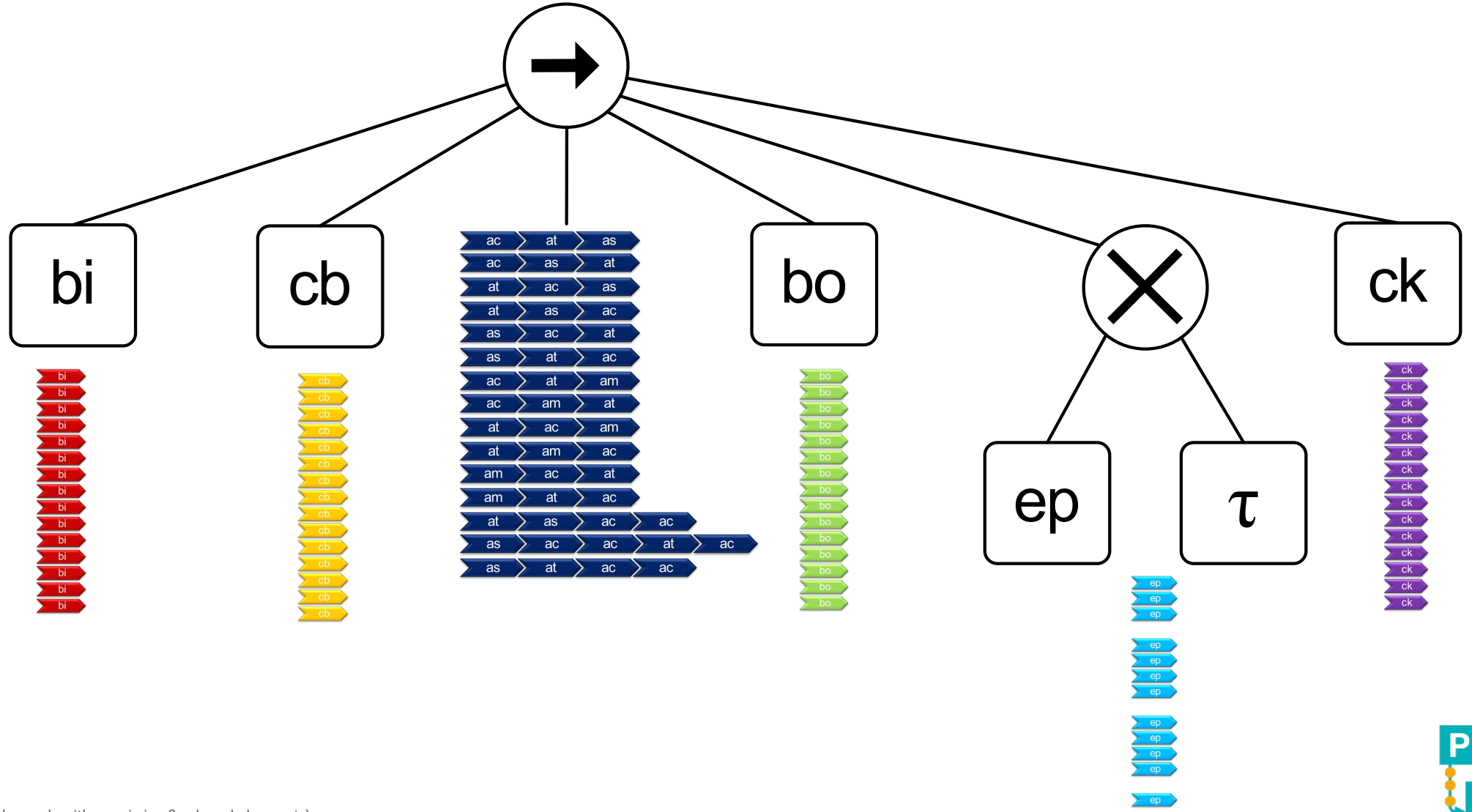
Recursion: Apply algorithm to all sublogs



Five of the projected event logs refer to a single activity (base case).

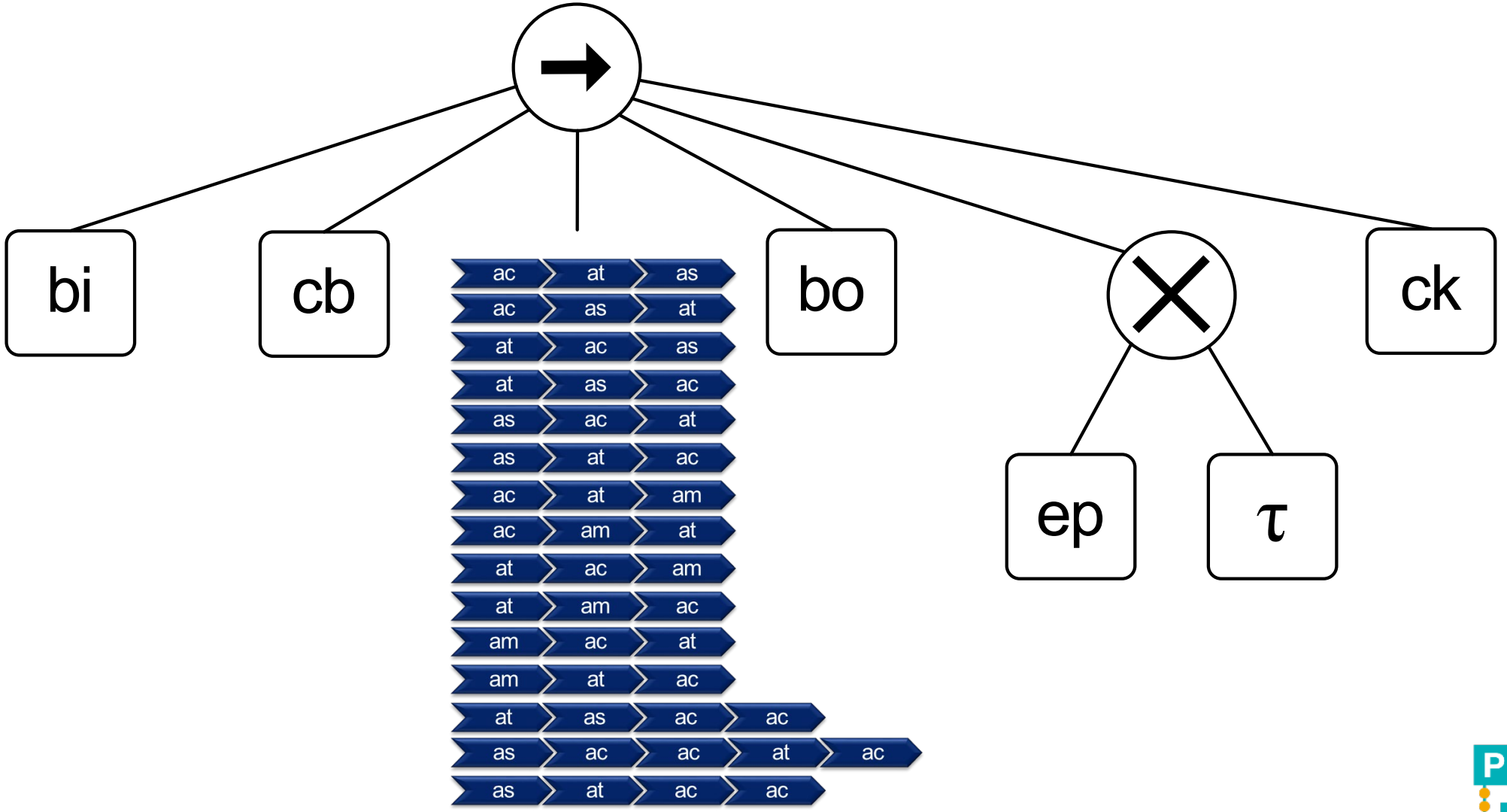
Chair of Process and Data Science

Handling the base cases (ep can be skipped)



Chair of Process and Data Science

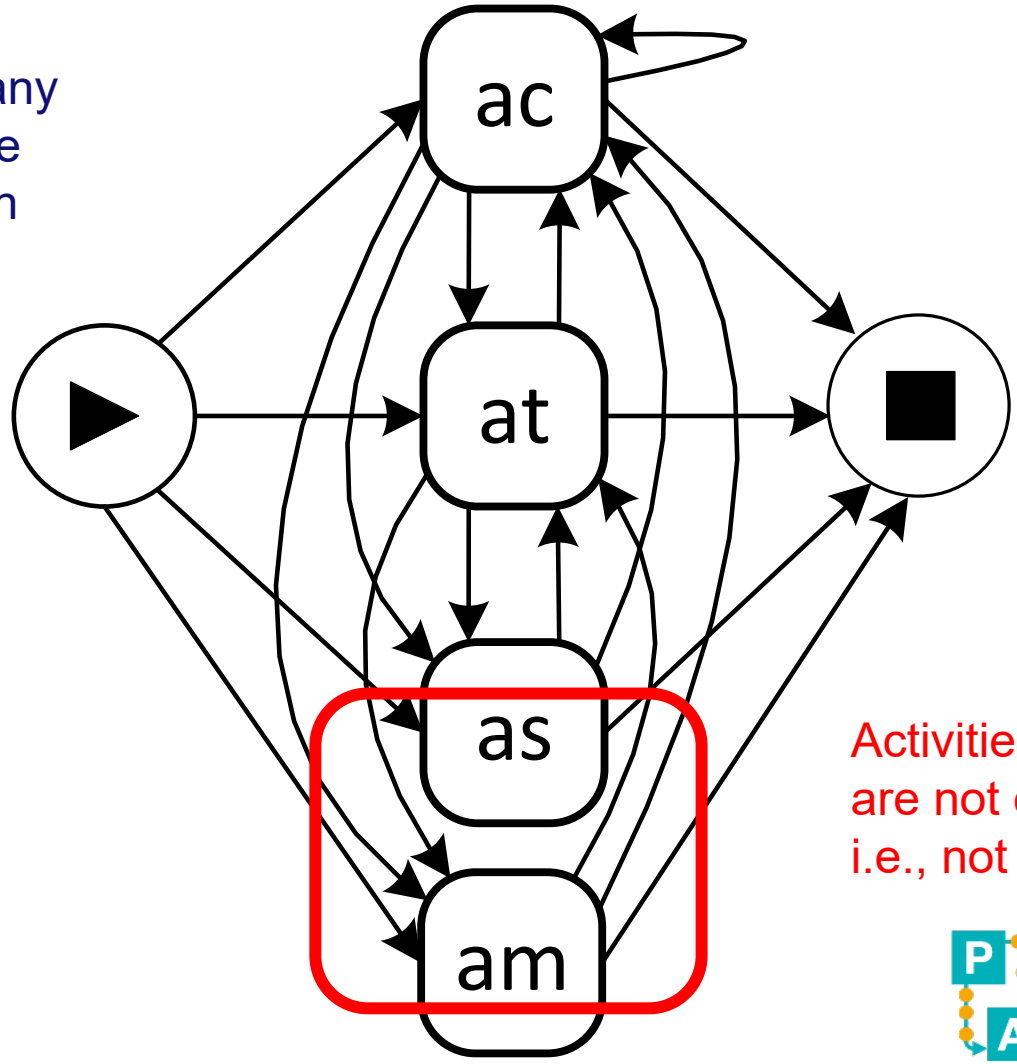
Only the blue event log remains



Continue with the blue event log

ac at as
ac as at
at ac as
at as ac
as ac at
as at ac
ac at am
ac am at
at ac am
at am ac
am ac at
am at ac
at as ac ac
as ac ac at ac
as at ac ac

There is a parallel cut when the DFG can be split into concurrent parts where any activity in one part can be followed by any activity in another part.



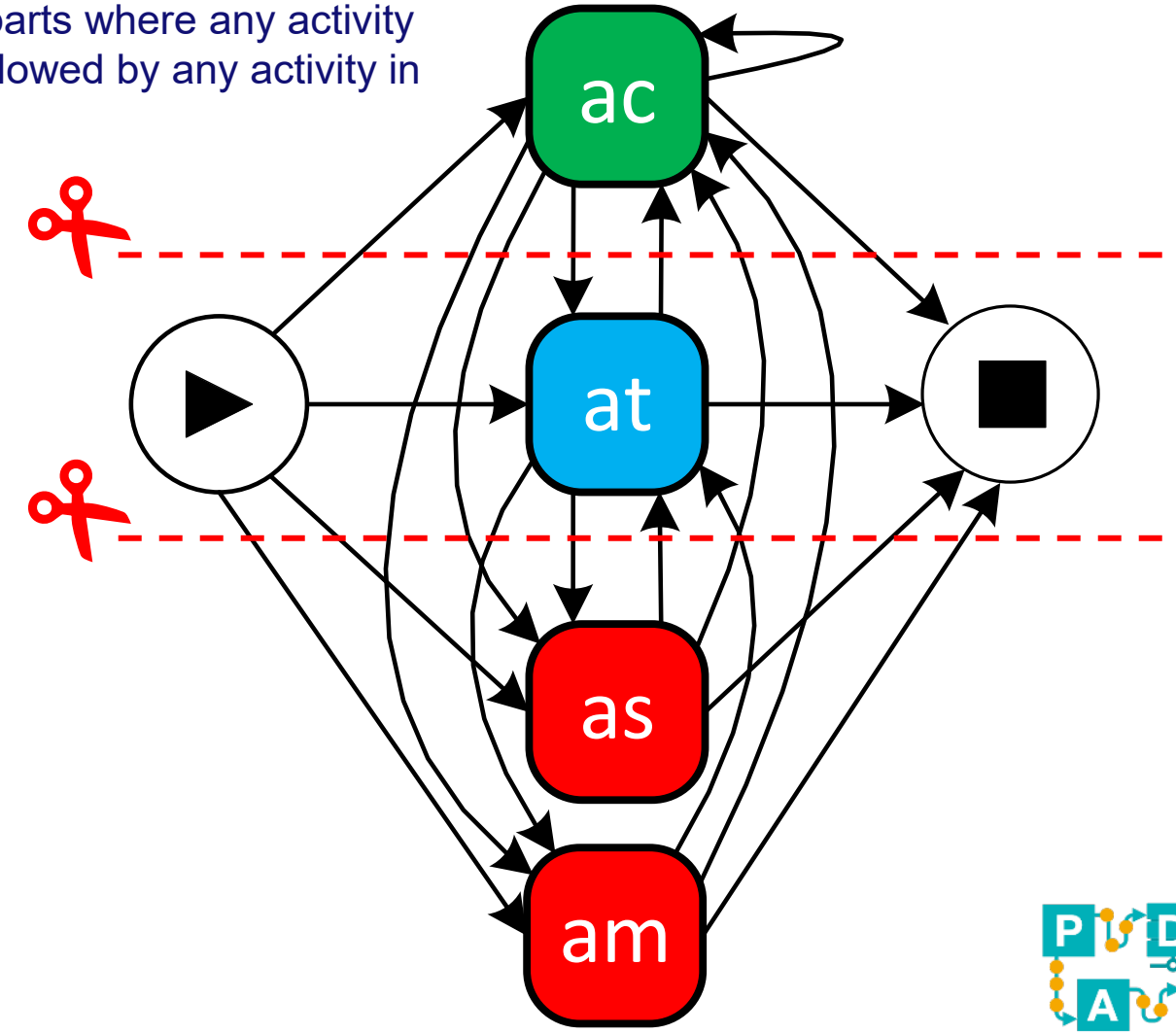
Activities as and am are not connected, i.e., not concurrent

Chair of Process and Data Science

Apply a parallel cut resulting in three activity groups

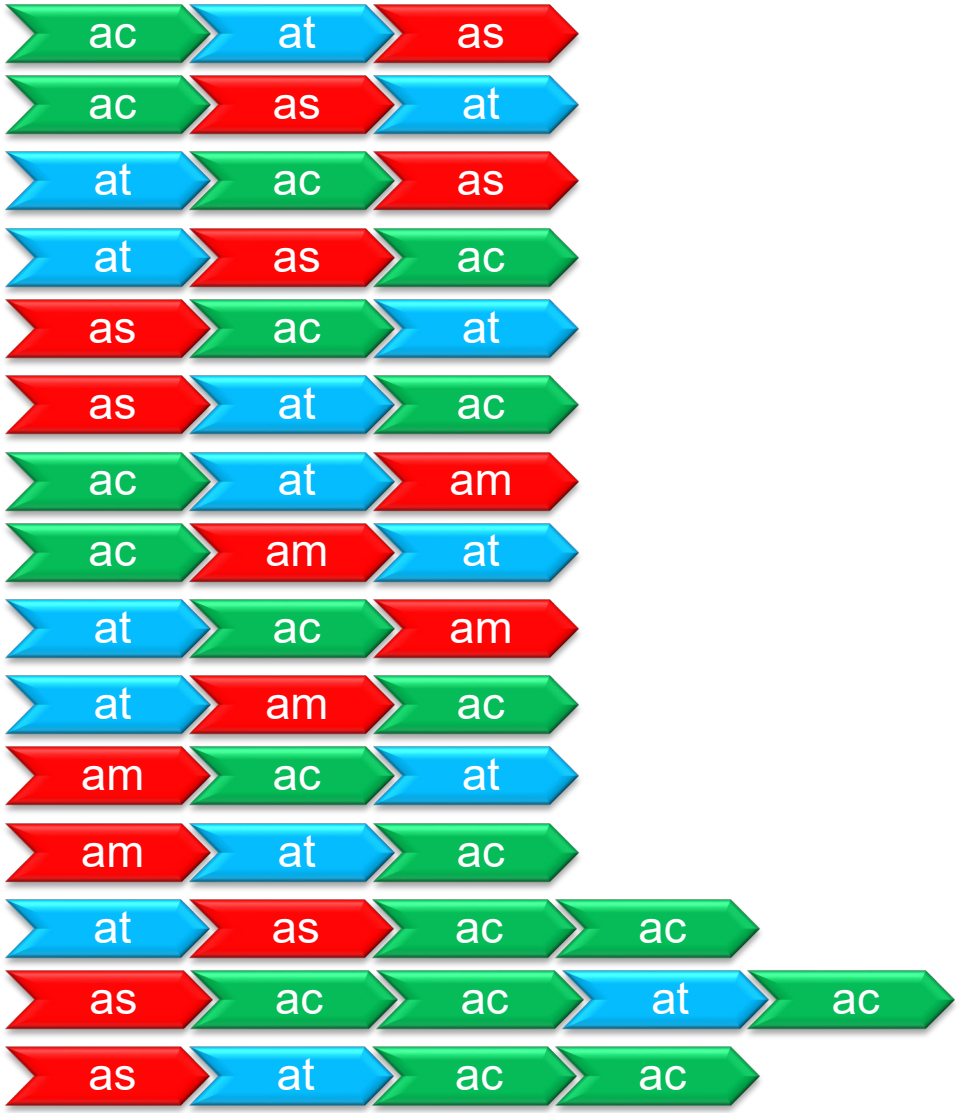
ac at as
ac as at
at ac as
at as ac
as ac at
as at ac
ac at am
ac am at
at ac am
at am ac
am ac at
am at ac
at as ac ac
as ac ac at ac
as at ac ac

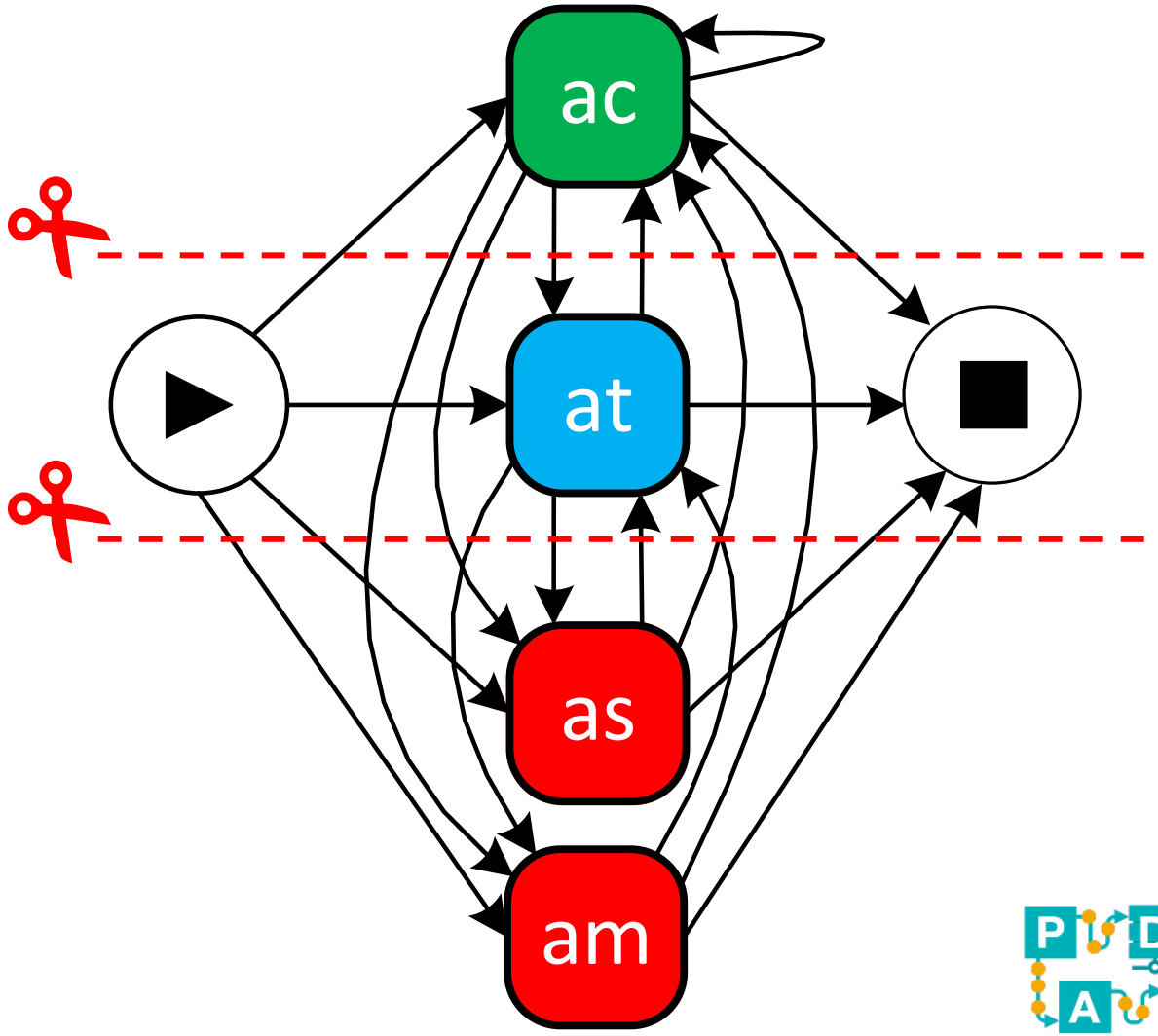
There is a parallel cut when the DFG can be split into concurrent parts where any activity in one part can be followed by any activity in another part.



Chair of Process and Data Science

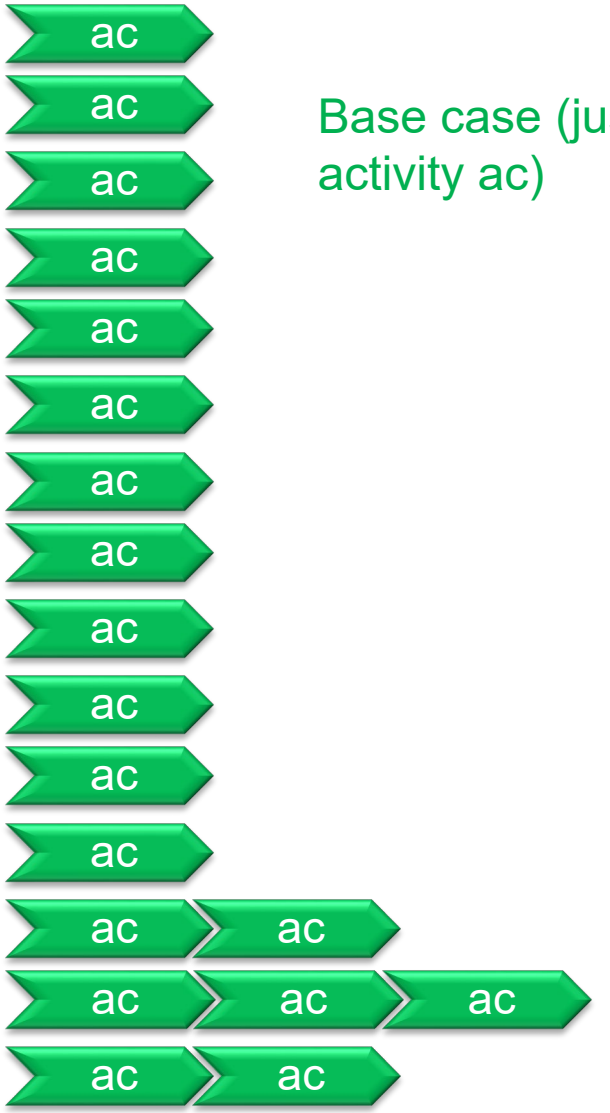
Apply a parallel cut resulting in three activity groups



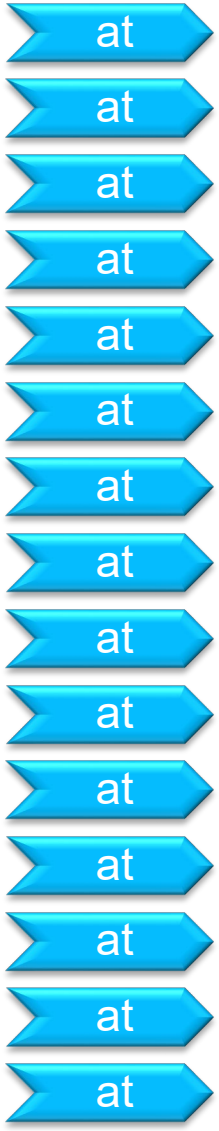


Chair of Process and Data Science

Three new event logs are created



Base case (just activity ac)

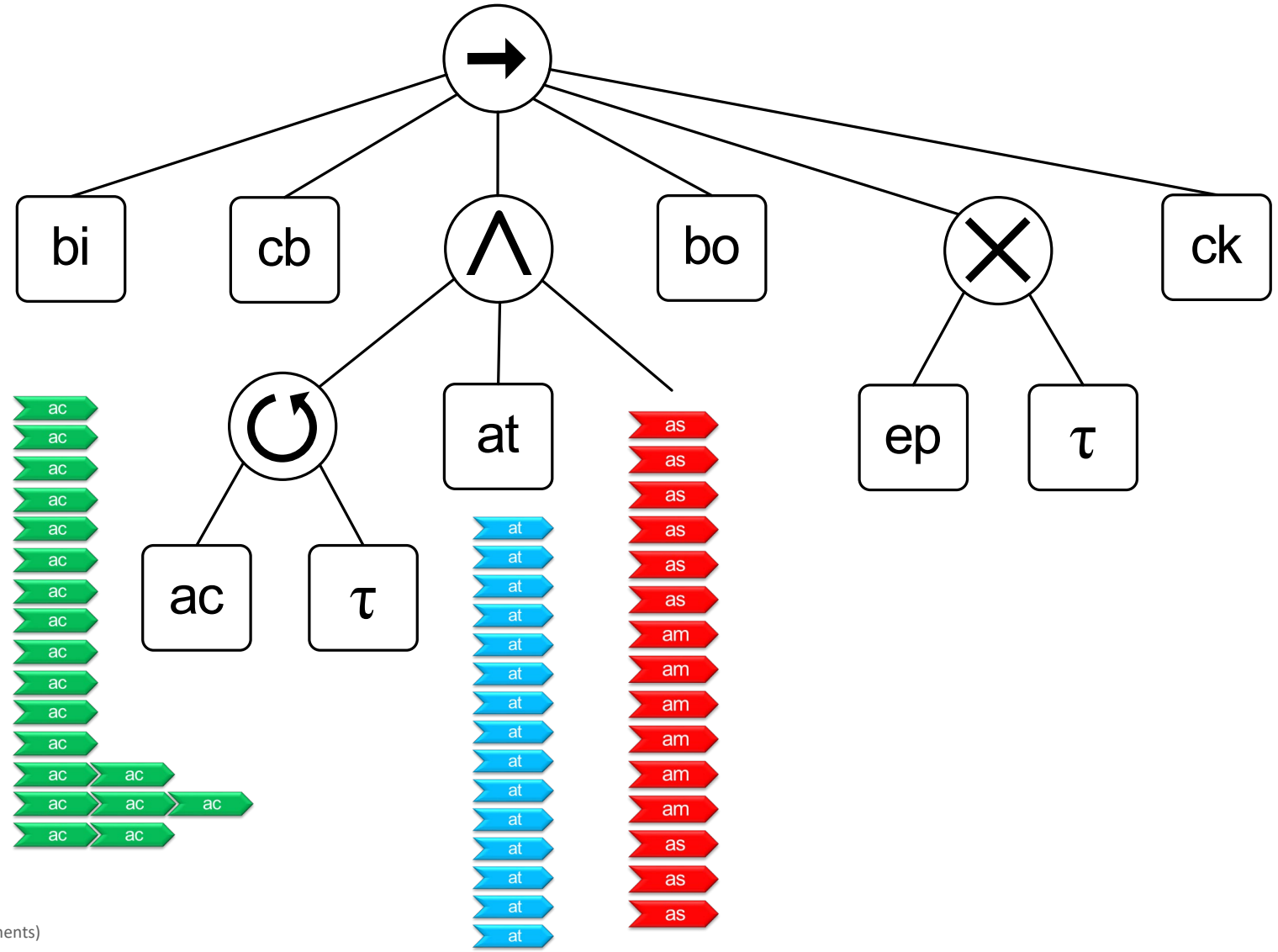


Base case (just activity at)

Not a base case, still two activities as and am.

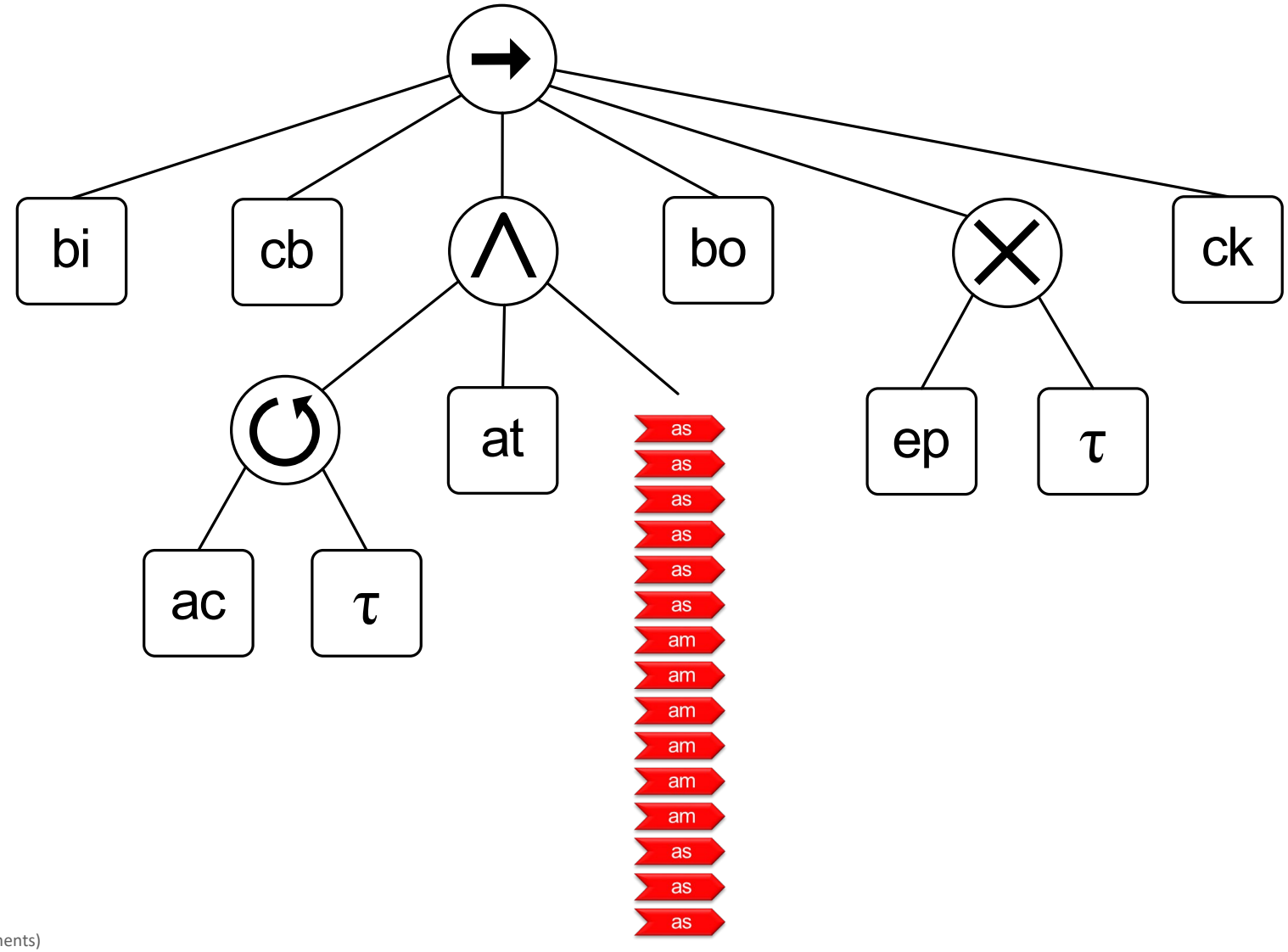
Chair of Process and Data Science

Handling the base cases (ac can be repeated)



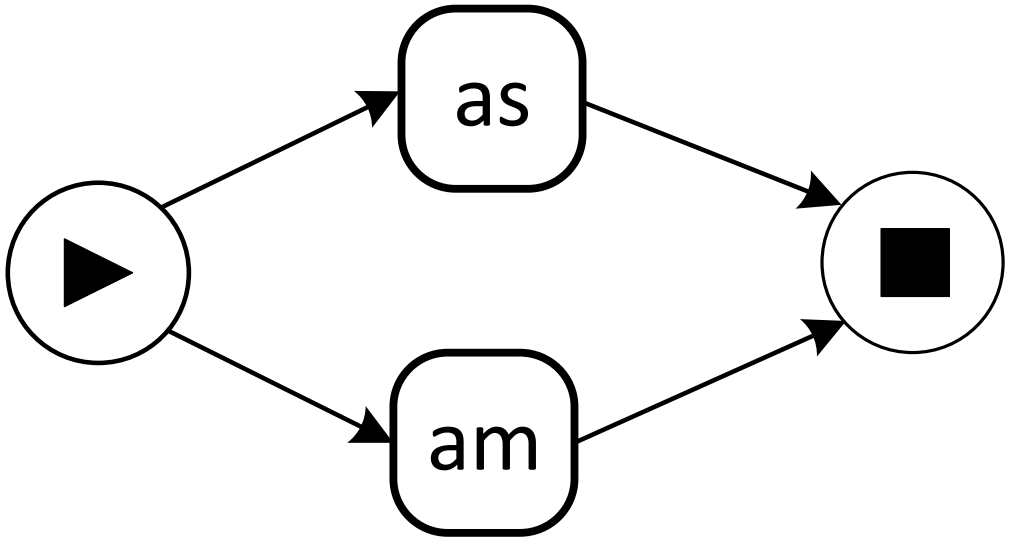
Chair of Process and Data Science

Only the red event log remains



Continue with the red event log

as
as
as
as
as
as
am
am
am
am
am
am
as
as
as

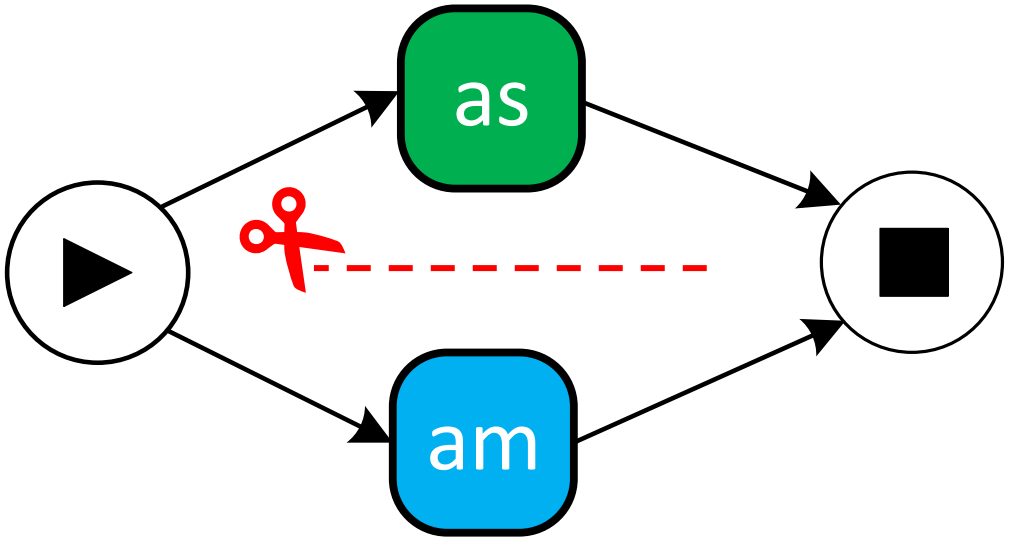


Chair of Process and Data Science

We find an exclusive-choice cut

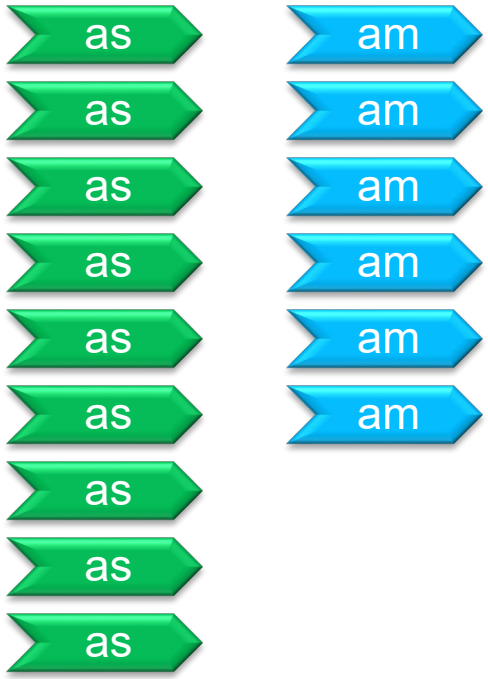
as
as
as
as
as
as
am
am
am
am
am
am
am
as
as
as

There is an exclusive-choice cut when the DFG can be split into disconnected parts after leaving out the artificial start and end.

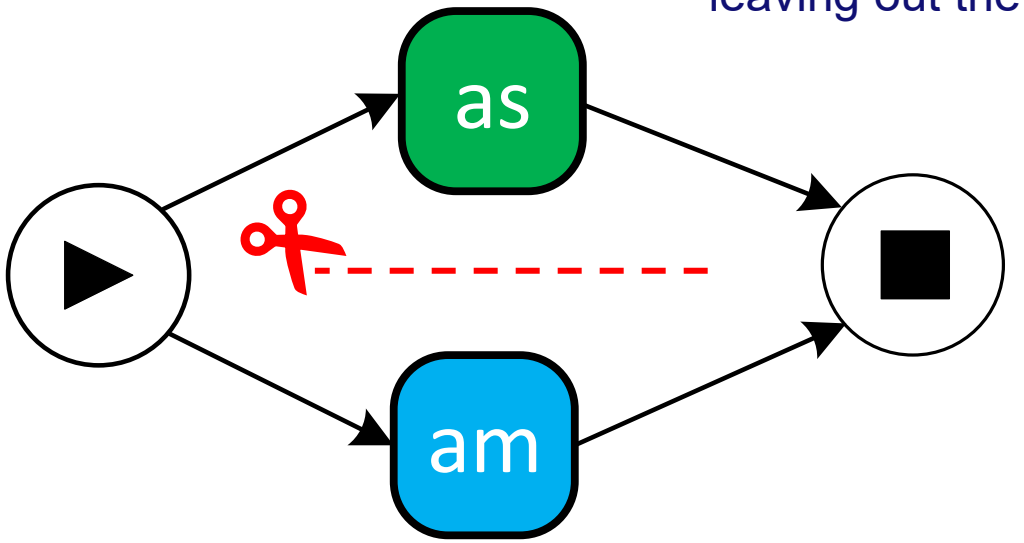


Chair of Process and Data Science

We find an exclusive-choice cut

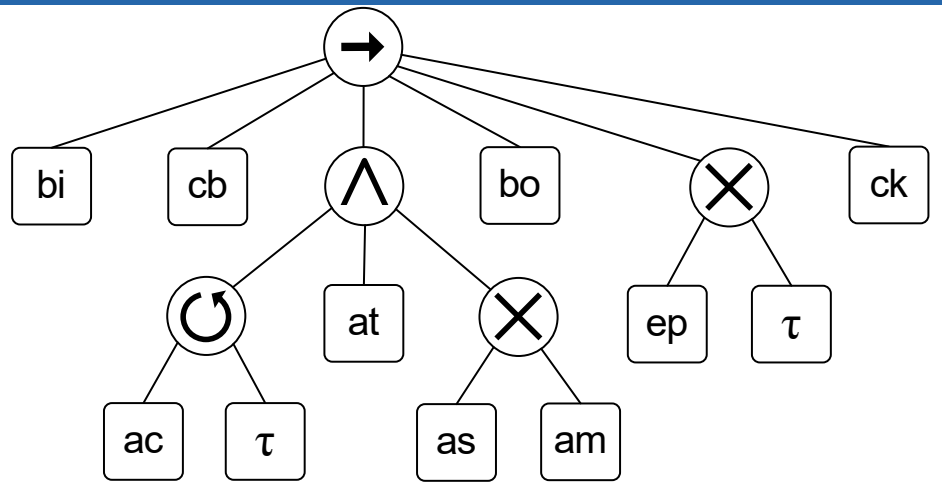


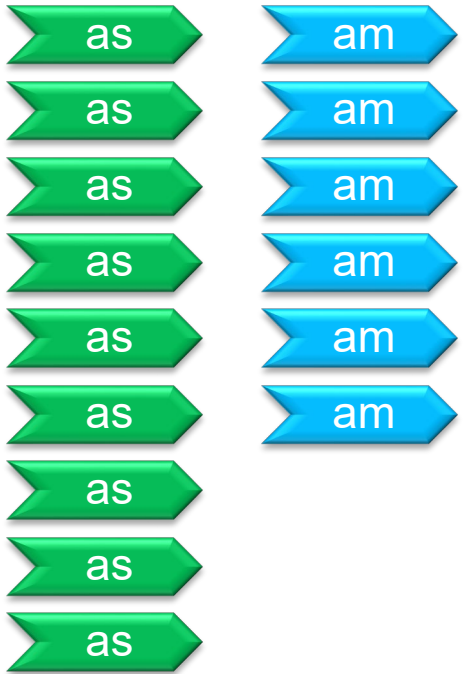
There is an exclusive-choice cut when the DFG can be split into disconnected parts after leaving out the artificial start and end.



Note that projection is now different than for the sequence and parallel cuts.

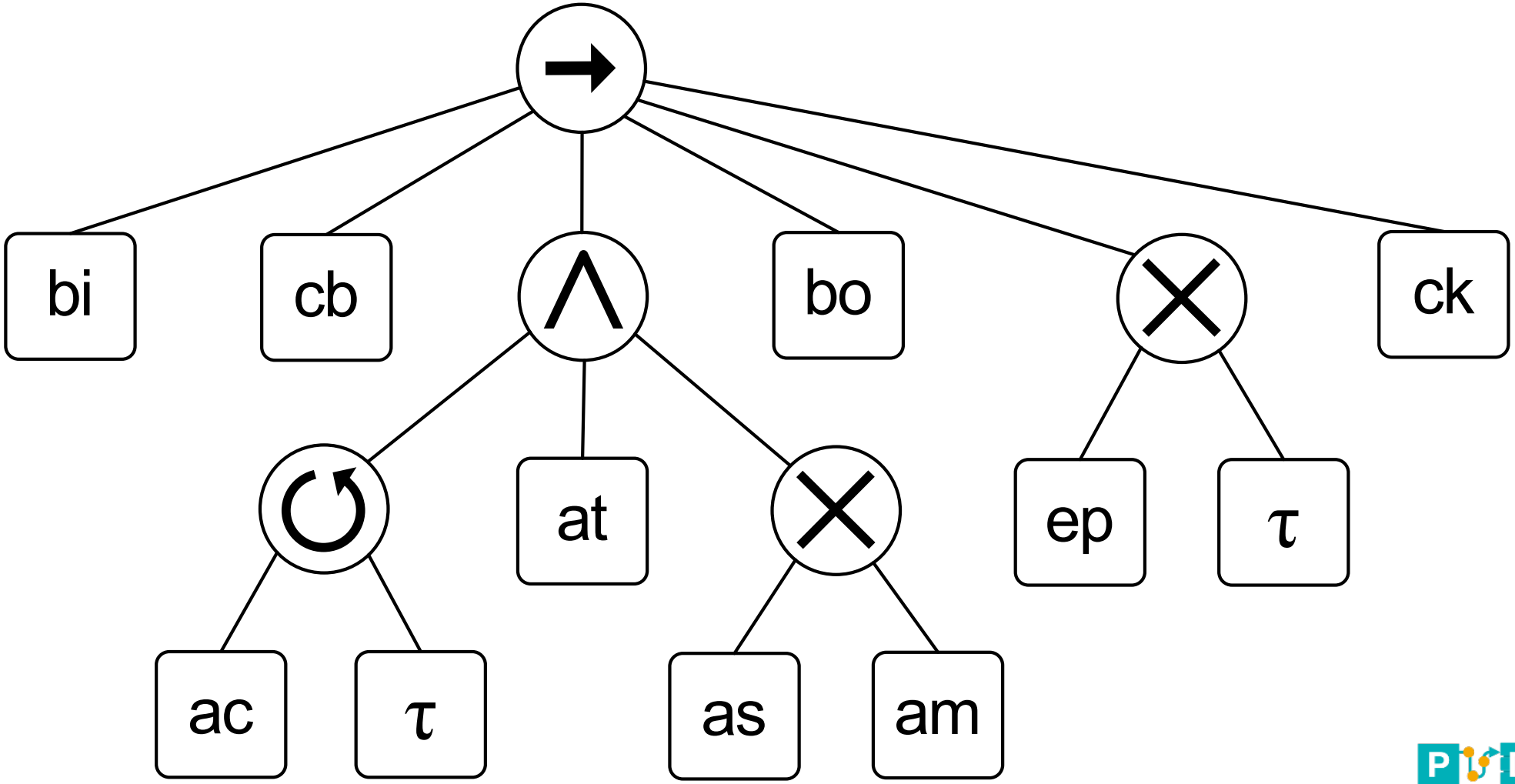
We end up with two base cases



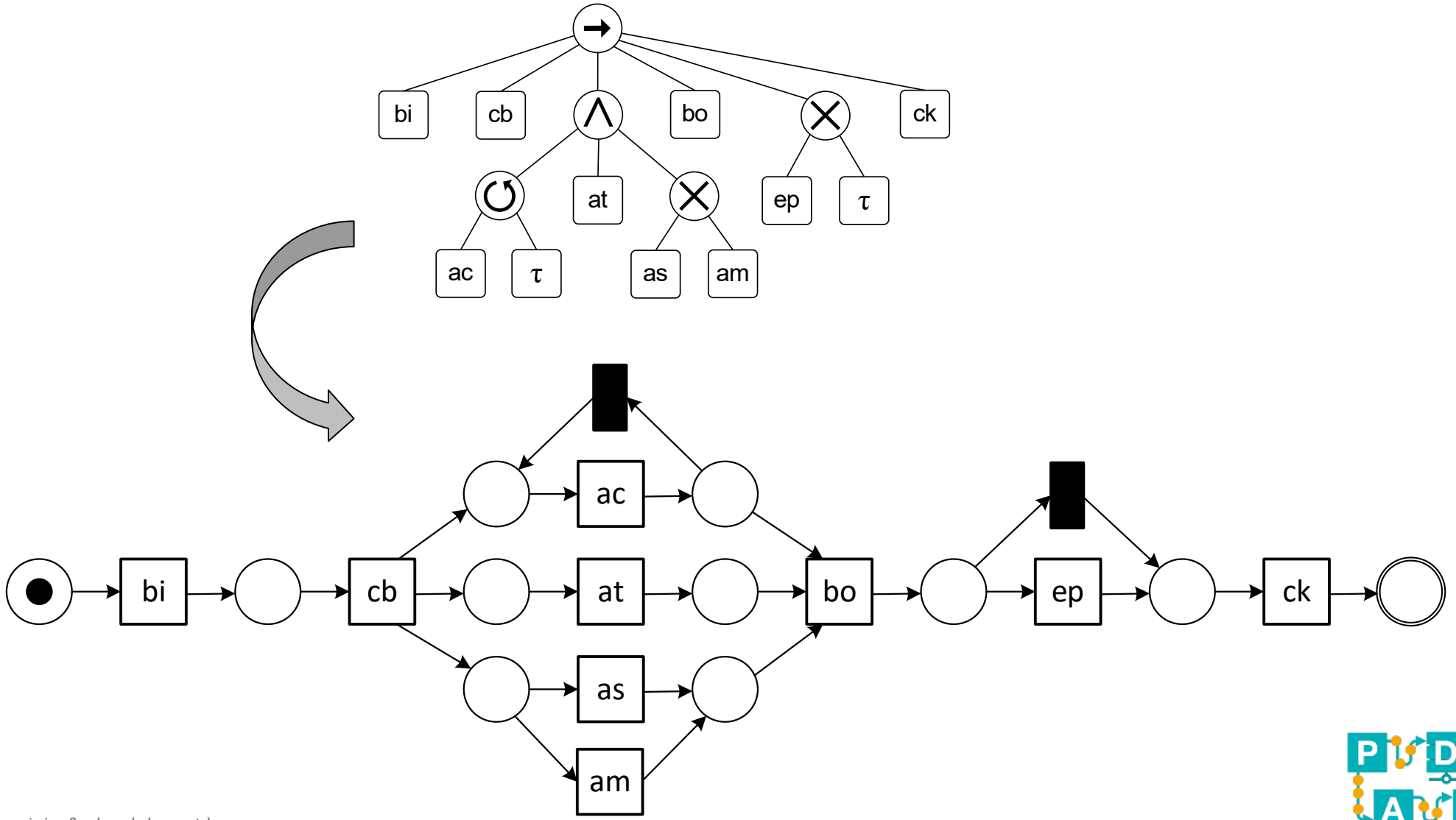


Chair of Process and Data Science

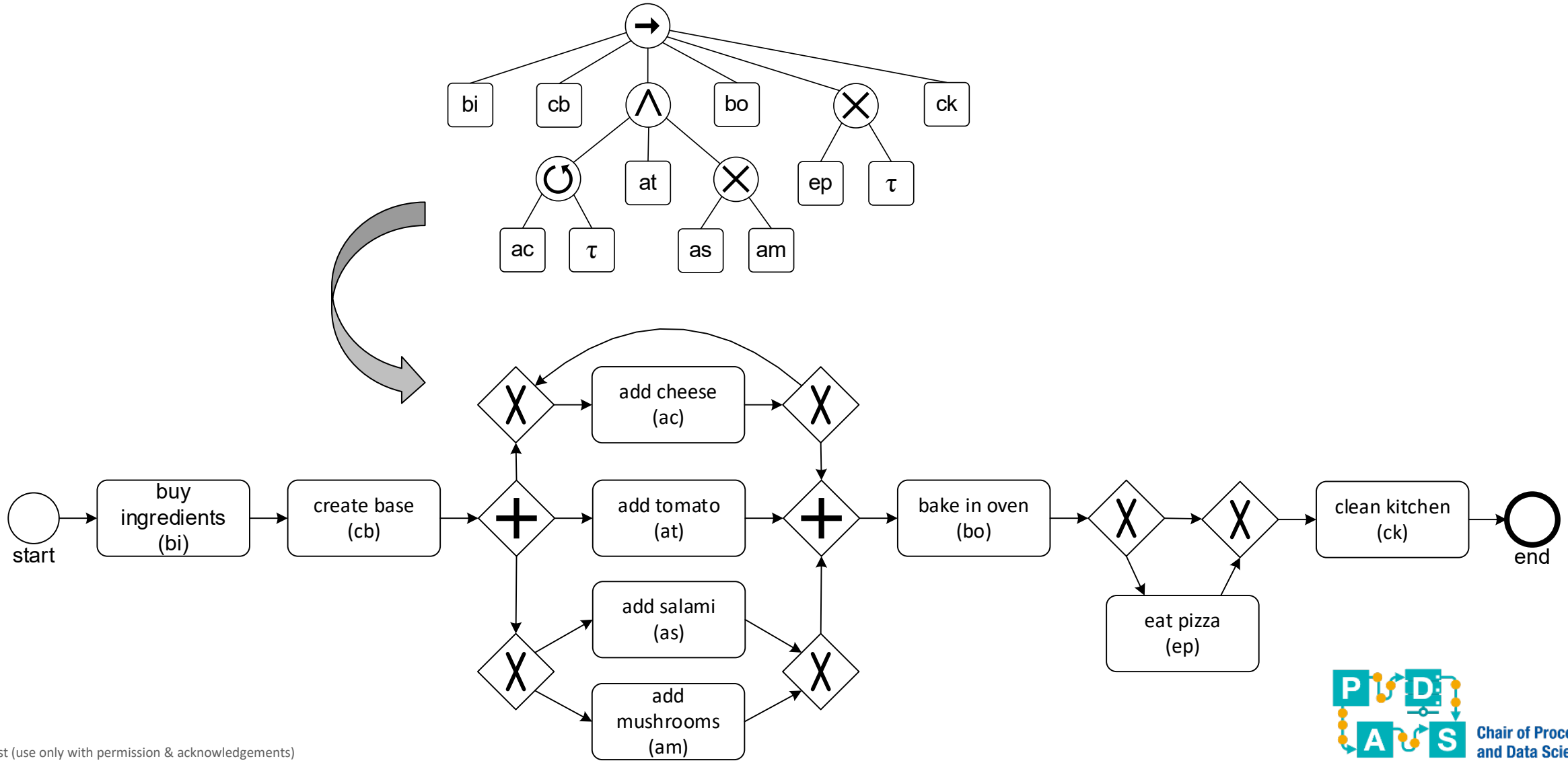
The process tree returned by the Inductive Mining algorithm



Can be visualized using Petri nets or BPMN



Can be visualized using Petri nets or BPMN

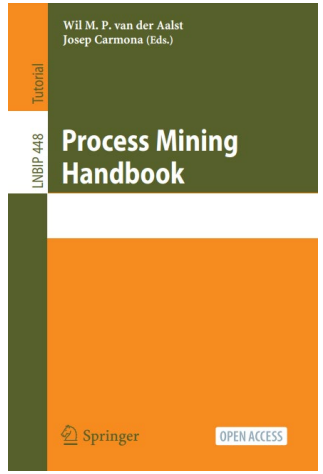


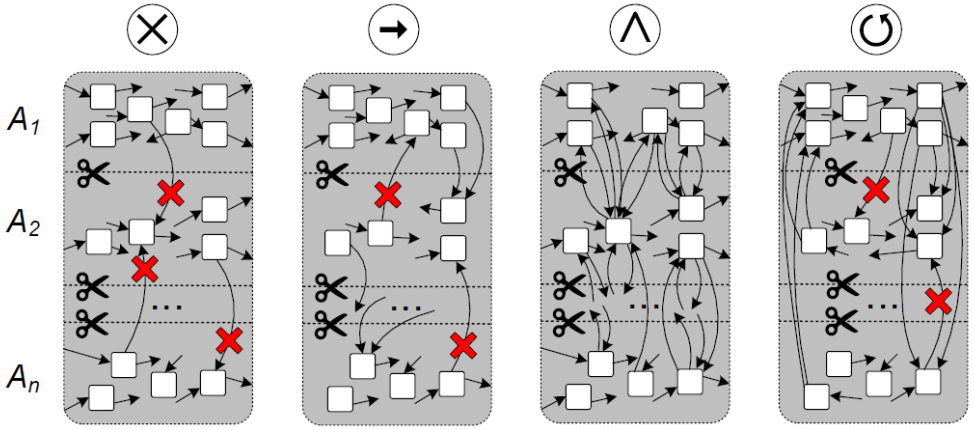
Chair of Process and Data Science

Formal definition of the algorithm

Definition 23 (Sequence, Exclusive-Choice, Parallel, and Redo-Loop Cuts). *Let $L \in \mathcal{B}(\mathcal{U}_{act}{}^*)$ be an event log having a DFG $disc_{DFG}(L) = (A, F)$ based on L (note that $A = act(L)$) with start activities $A^{start} = \{a \in A \mid (\blacktriangleright, a) \in F\}$ and end activities $A^{end} = \{a \in A \mid (a, \blacksquare) \in F\}$. An n-ary $\oplus$-cut of L is a partition of A into $n \geq 2$ pairwise disjoint subsets $A_1, A_2, \ldots, A_n$ (i.e., $A = \bigcup_{i \in \{1,\ldots,n\}} A_i$ and $A_i \cap A_j = \emptyset$ for $i \neq j$) with $\oplus \in \{\rightarrow, \times, \wedge, \circlearrowleft\}$. Such a $\oplus$-cut is denoted $(\oplus, A_1, A_2, \ldots A_n)$. For each type of operator $\oplus \in \{\rightarrow, \times, \wedge, \circlearrowleft\}$ specific conditions apply:*

- *An* exclusive-choice cut *of L is a cut $(\times, A_1, A_2, \ldots A_n)$ such that*
  - $\forall_{i,j \in \{1,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow (a, b) \notin F$.
- *A* sequence cut *of L is a cut $(\rightarrow, A_1, A_2, \ldots A_n)$ such that*
  - $\forall_{i,j \in \{1,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i < j \Rightarrow ((a, b) \in F^+ \wedge (b, a) \notin F^+)$. *(Note that F^+ is the non-reflexive transitive closure of F, i.e., $(a, b) \in F^+$ means that there is a path from a to b in the DFG.)*
- *A* parallel cut *of L is a cut $(\wedge, A_1, A_2, \ldots A_n)$ such that*
  - $\forall_{i \in \{1,\ldots n\}}\ A_i \cap A^{start} \neq \emptyset \wedge A_i \cap A^{end} \neq \emptyset$ *and*
  - $\forall_{i,j \in \{1,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow (a, b) \in F$.
- *A* redo-loop cut *of L is a cut $(\circlearrowleft, A_1, A_2, \ldots A_n)$ such that*
  - $A^{start} \cup A^{end} \subseteq A_1$,
  - $\forall_{i,j \in \{2,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow (a, b) \notin F$,
  - $\{a \in A_1 \mid (a, b) \in F \wedge b \notin A_1\} = A^{end}$,
  - $\{a \in A_1 \mid (b, a) \in F \wedge b \notin A_1\} = A^{start}$,
  - $\forall_{(a,b) \in F}\ a \in A_1 \wedge b \notin A_1 \Rightarrow \forall_{a' \in A^{end}}\ (a', b) \in F$, *and*
  - $\forall_{(b,a) \in F}\ a \in A_1 \wedge b \notin A_1 \Rightarrow \forall_{a' \in A^{start}}\ (b, a') \in F$.





(a) exclusive-choice cut (b) sequence cut (c) parallel cut (d) redo-loop cut

Chair of Process and Data Science

Strong theoretical guarantees

- Guaranteed to be able to replay the whole event log, i.e., perfect recall (if desired).
- “Rediscoverability” of process trees with unique activities (if the log is directly-follows complete).
- Scalable enough for real-life events logs (more scalable variants available) and able to filter out infrequent behavior.
- Implemented in open source tools like ProM and PM4py, and several commercial tools, including Celonis.

Question: Is this machine learning?

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Example: ILP Mining

Definition (Basic ILP formulation) Let $L \subseteq T^*$ be an event log (cardinalities are not considered). A and A' are the matrices defined for the language of L. ILP_L is the integer programming problem:

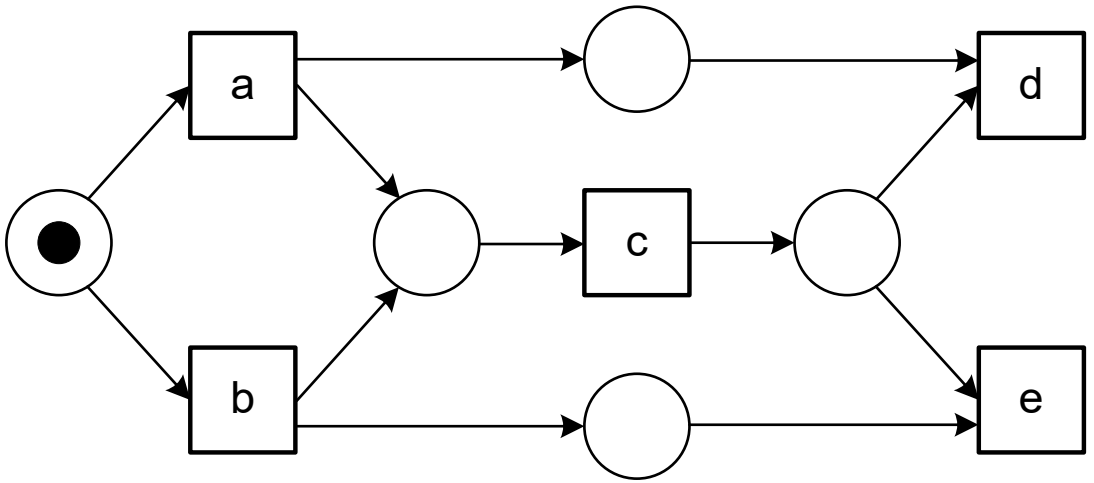
$$
\begin{array}{lrl}
\textbf{Minimize} & c + \mathbf{1}^T \cdot (c \cdot \mathbf{1} + A \cdot (\mathbf{x} - \mathbf{y})) & \text{minimal and ``interesting'' regions} \\
\textbf{such that} & c \cdot \mathbf{1} + A' \cdot \mathbf{x} - A \cdot \mathbf{y} \geq \mathbf{0} & (\mathbf{x}, \mathbf{y}, c) \text{ is a region} \\
& \mathbf{1}^T \cdot \mathbf{x} + \mathbf{1}^T \cdot \mathbf{y} \geq 1 & \text{there is at least one edge} \\
& \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} & \mathbf{x} \in \{0,1\}^T \\
& \mathbf{0} \leq \mathbf{y} \leq \mathbf{1} & \mathbf{y} \in \{0,1\}^T \\
& 0 \leq c \leq 1 & c \in \{0,1\}
\end{array}
$$

Joint work with Boudewijn van Dongen, Cor Hurkens, Bas van Zelst, et al.

Finding places in a Petri net

Example Integer Linear Program (ILP) to discover places in a Petri net

$$L = \{\langle a, c, d\rangle, \langle b, c, e\rangle\}$$



Minimize $c + (1,1,1,1,1,1) \cdot (c \cdot \begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix} + \begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\1&0&1&0&0\\0&1&1&0&0\\1&0&1&1&0\\0&1&1&0&1\end{pmatrix} \cdot \begin{pmatrix}x_a - y_a\\x_b - y_b\\x_c - y_c\\x_d - y_d\\x_e - y_e\end{pmatrix})$

such that:

$$c \cdot \begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix} + \begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\1&0&0&0&0\\0&1&0&0&0\\1&0&1&0&0\\0&1&1&0&0\end{pmatrix} \cdot \begin{pmatrix}x_a\\x_b\\x_c\\x_d\\x_e\end{pmatrix} - \begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\1&0&1&0&0\\0&1&1&0&0\\1&0&1&1&0\\0&1&1&0&1\end{pmatrix} \cdot \begin{pmatrix}y_a\\y_b\\y_c\\y_d\\y_e\end{pmatrix} \geq \begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix}$$

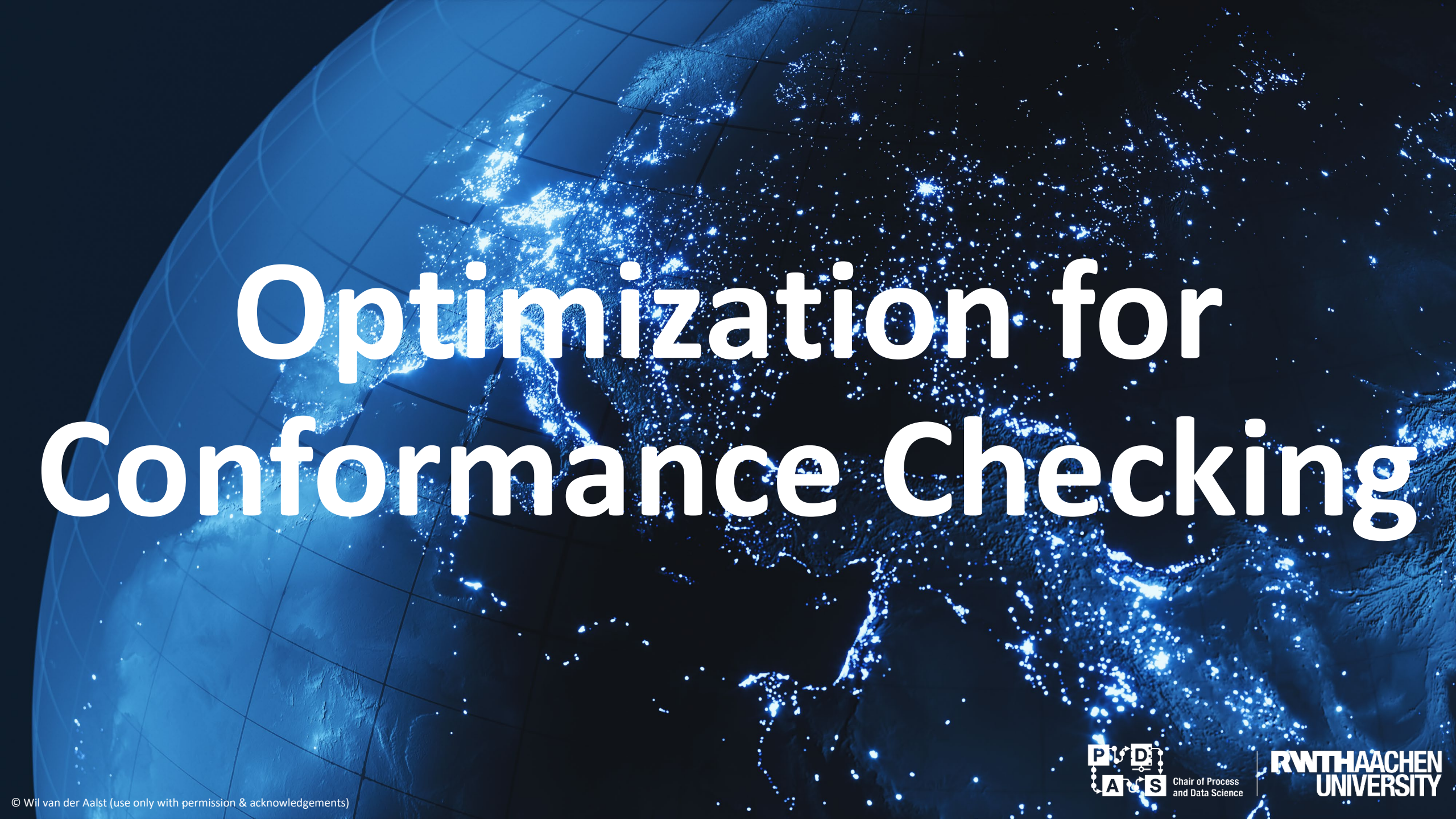
$$(1,1,1,1,1) \cdot \begin{pmatrix}x_a\\x_b\\x_c\\x_d\\x_e\end{pmatrix} + (1,1,1,1,1) \cdot \begin{pmatrix}y_a\\y_b\\y_c\\y_d\\y_e\end{pmatrix} \geq 1$$

$$\begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix} \leq \begin{pmatrix}x_a\\x_b\\x_c\\x_d\\x_e\end{pmatrix} \leq \begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}$$

$$\begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix} \leq \begin{pmatrix}y_a\\y_b\\y_c\\y_d\\y_e\end{pmatrix} \leq \begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}$$

$$0 \leq c \leq 1$$

each place is a solution
(skipping some details, e.g., empty at the end)

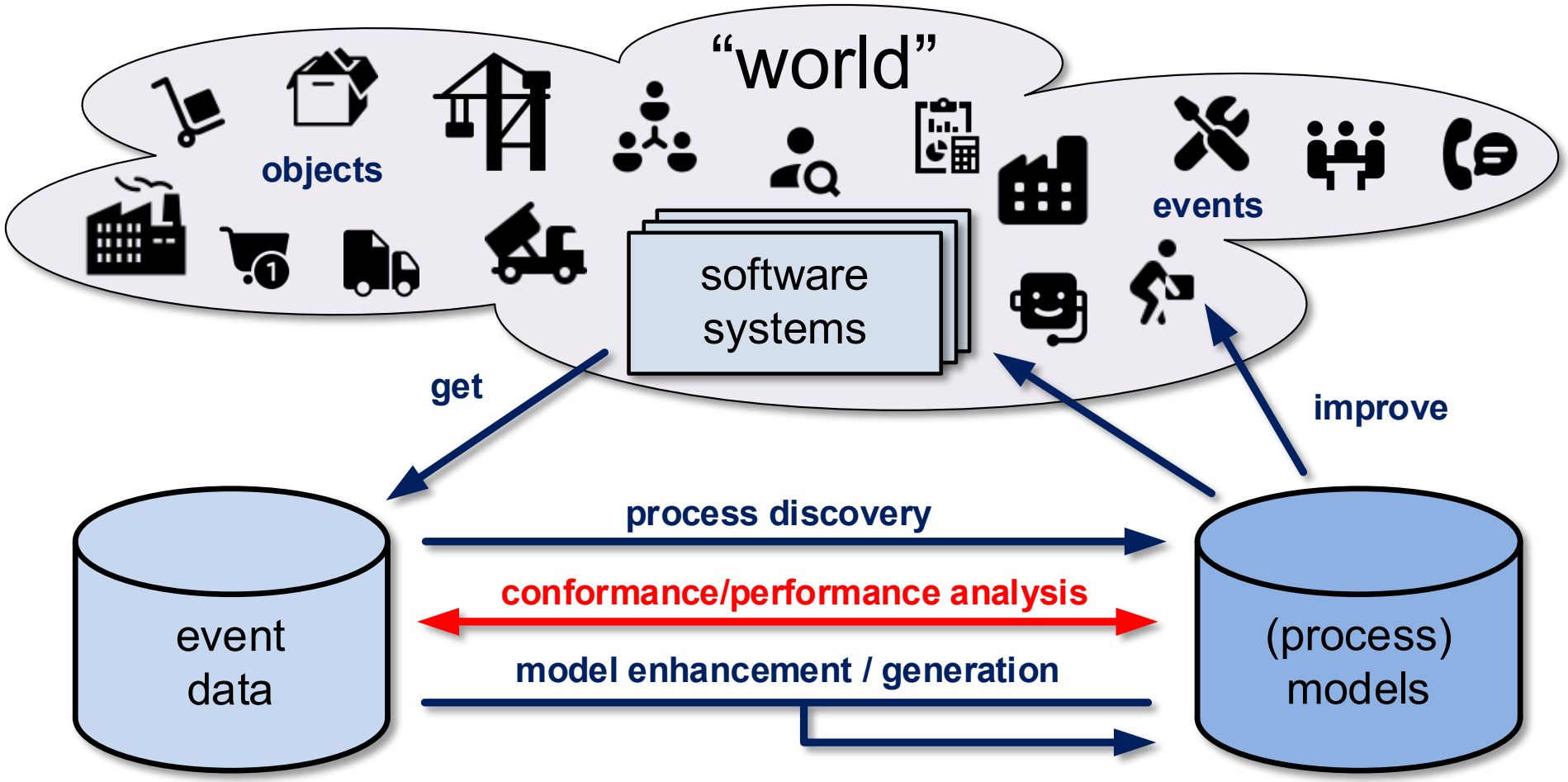


Optimization for Conformance Checking

Chair of Process and Data Science

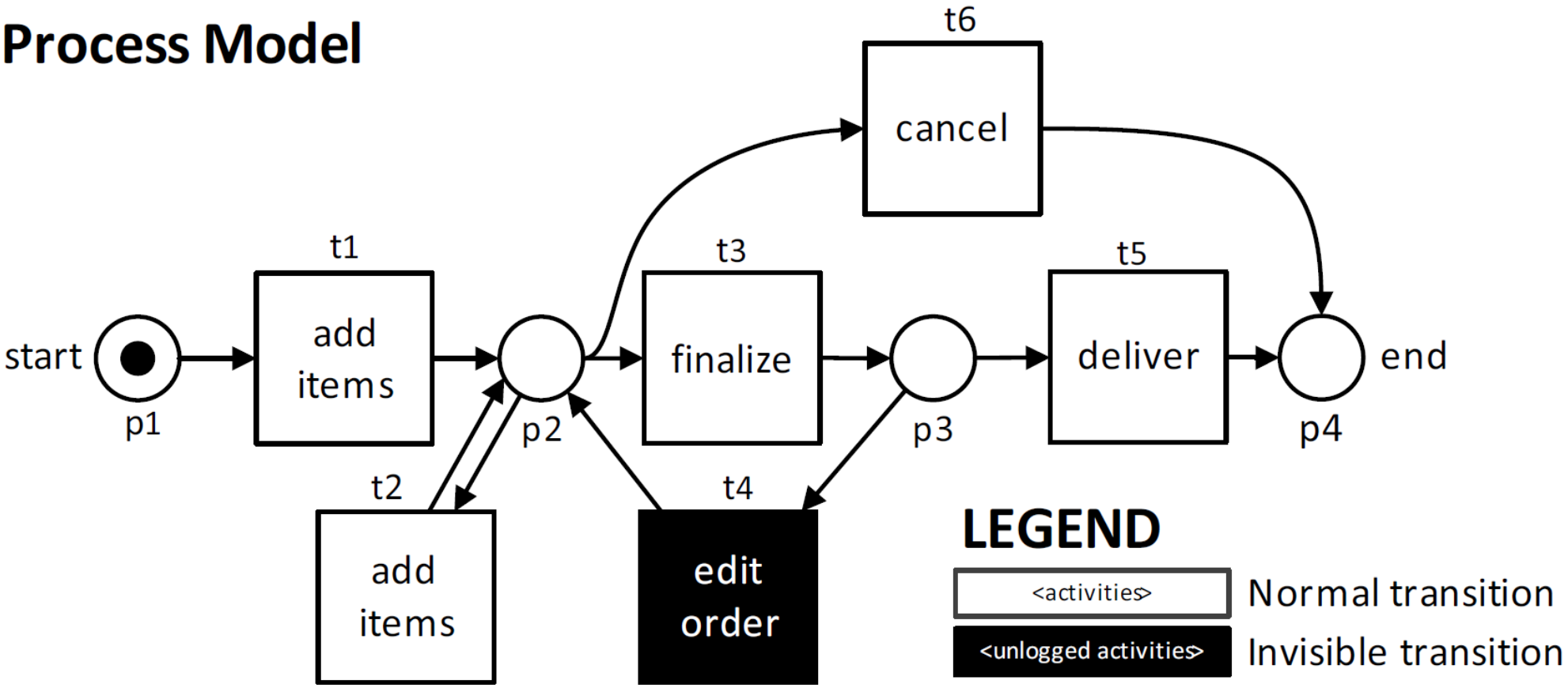
RWTH AACHEN UNIVERSITY

Conformance Checking



Chair of Process and Data Science

Computing alignments

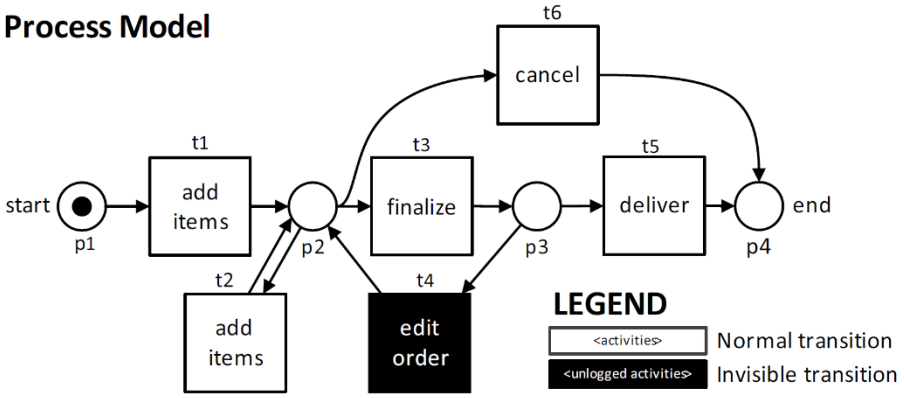


Trace : *<add items, add items, finalize, finalize>*

Joint work with Arya Adriansyah and Boudewijn van Dongen

Chair of Process and Data Science

Two example alignments



Trace : *<add items, add items, finalize, finalize>*

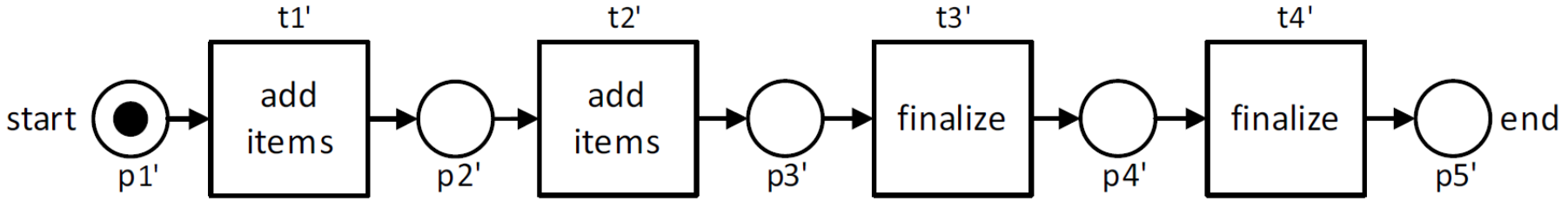
$$\gamma_1 = \begin{array}{|c|c|c|c|c|} \hline add\ items & add\ items & finalize & finalize & \gg \\ \hline add\ items & add\ items & finalize & \gg & deliver \\ t_1 & t_2 & t_3 & & t_5 \\ \hline \end{array}$$

$$\gamma_2 = \begin{array}{|c|c|c|c|c|c|} \hline add\ items & add\ items & finalize & \gg & finalize & \gg \\ \hline add\ items & add\ items & finalize & edit\ order & finalize & deliver \\ t_1 & t_2 & t_3 & t_4 & t_3 & t_5 \\ \hline \end{array}$$

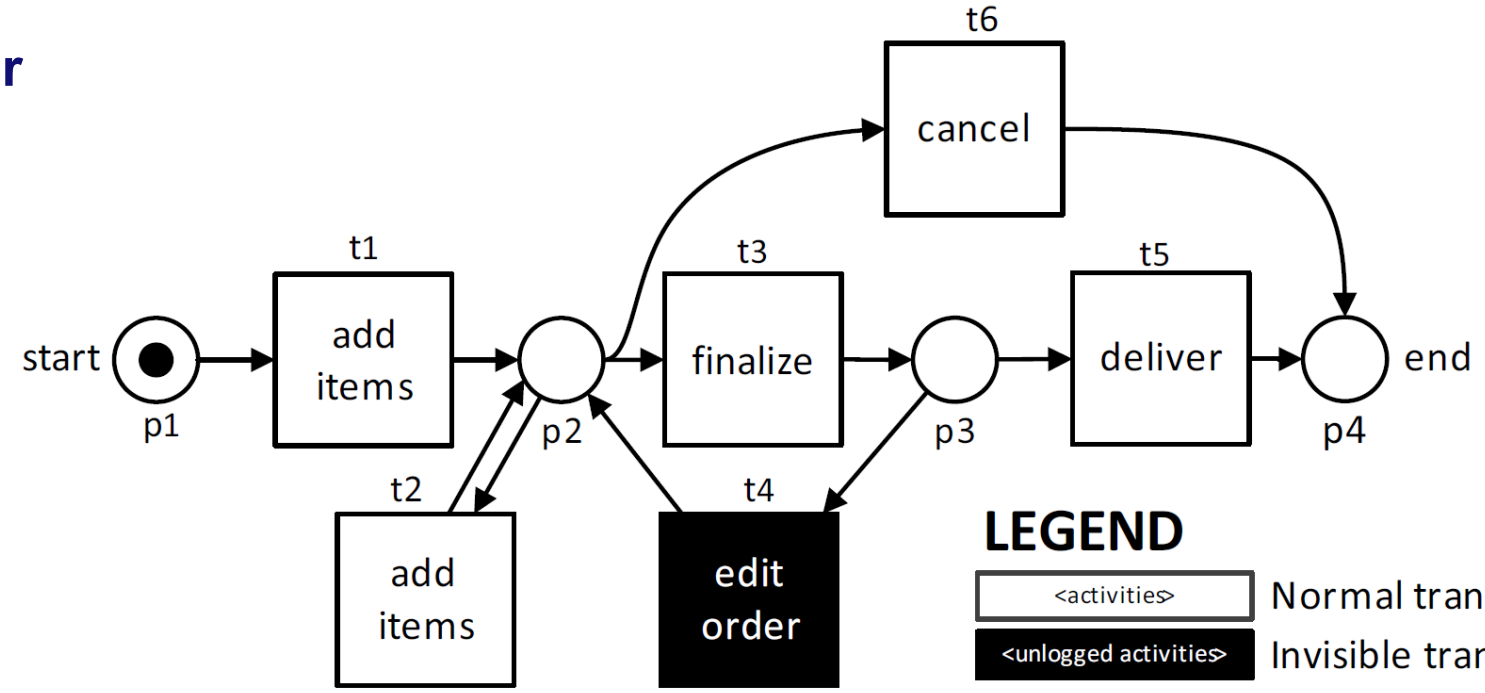
Chair of Process and Data Science

Two Petri nets

Observed behavior

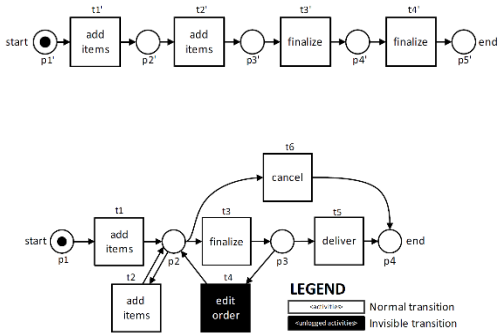


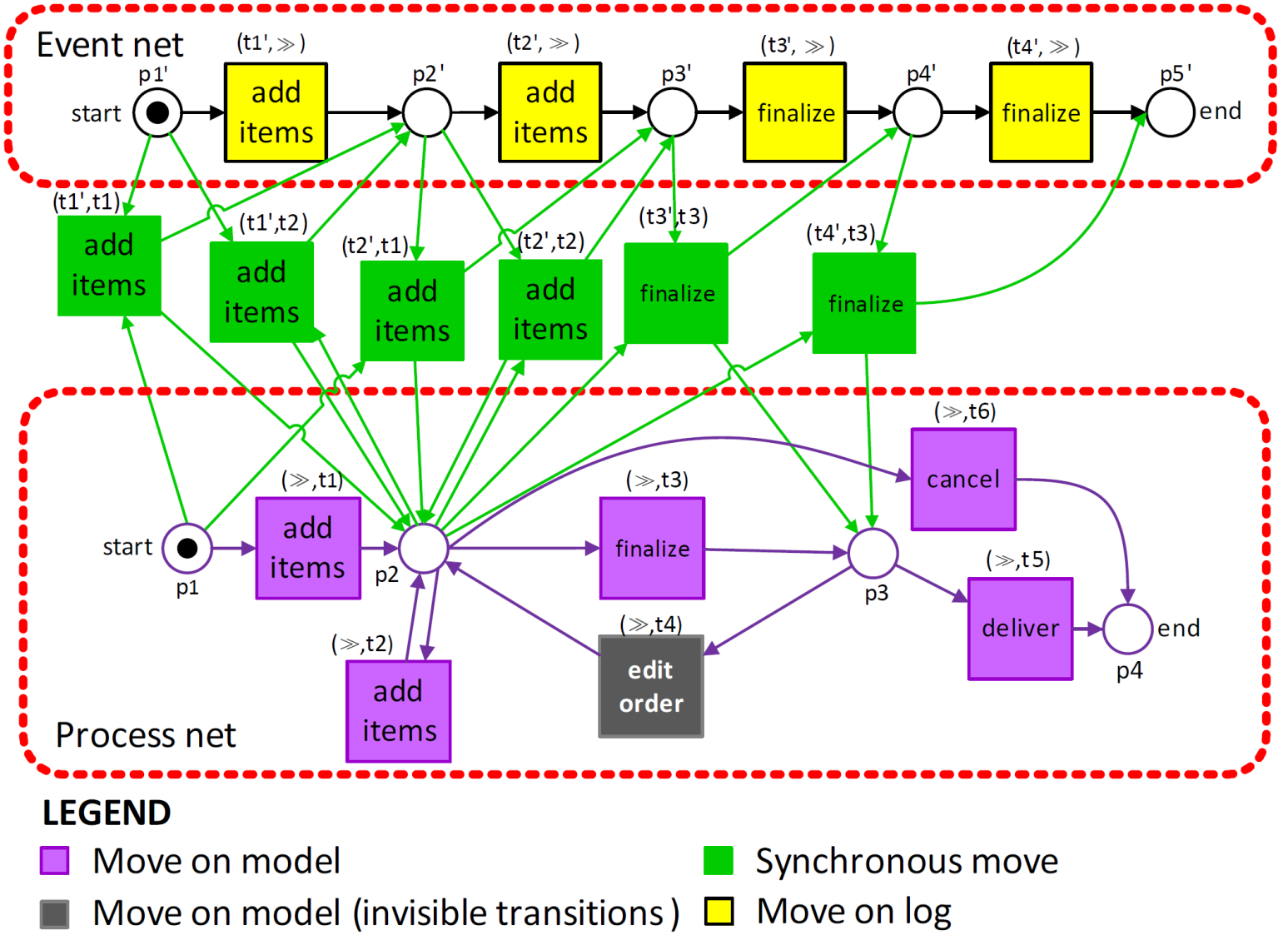
Modeled behavior



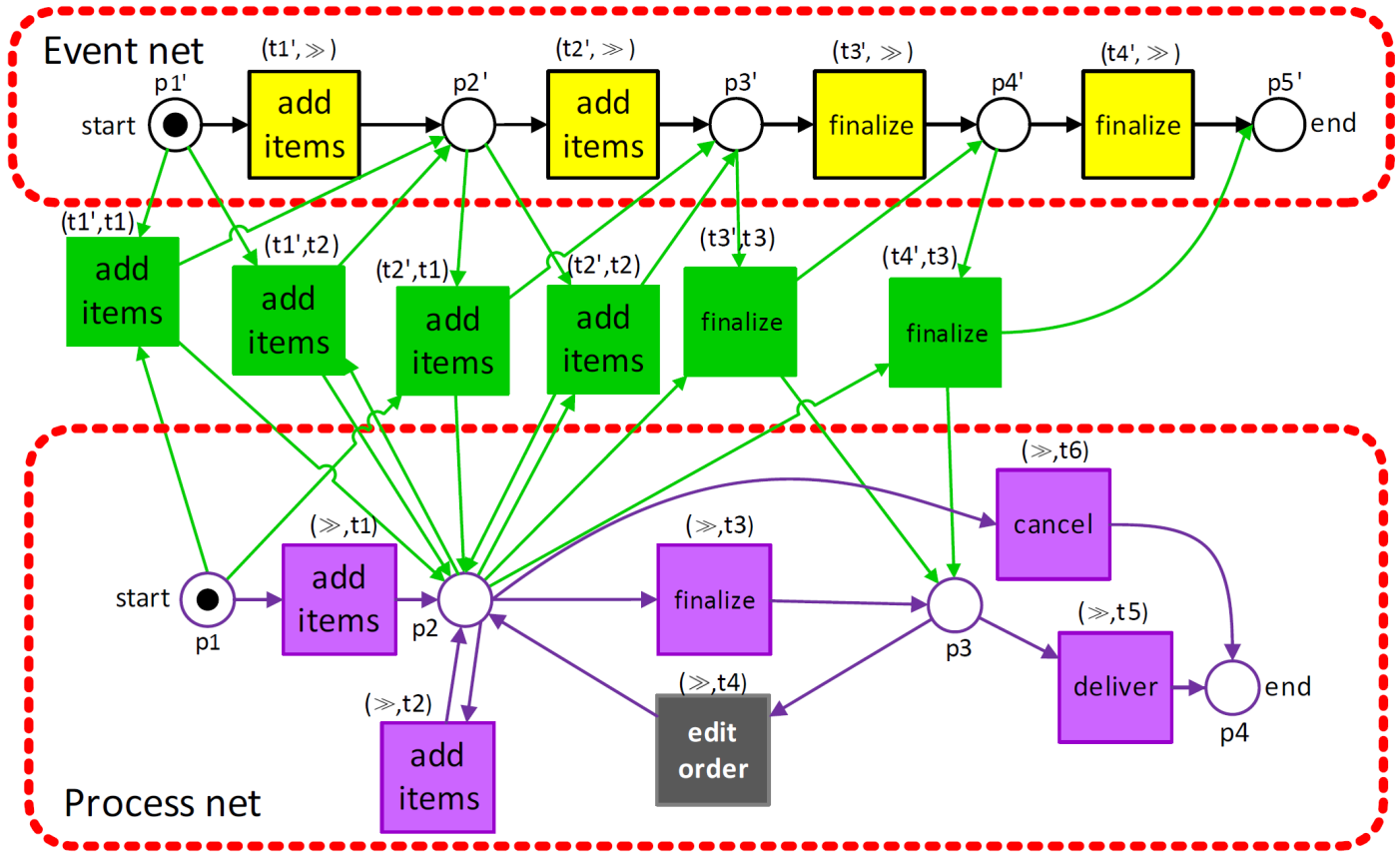
Chair of Process and Data Science

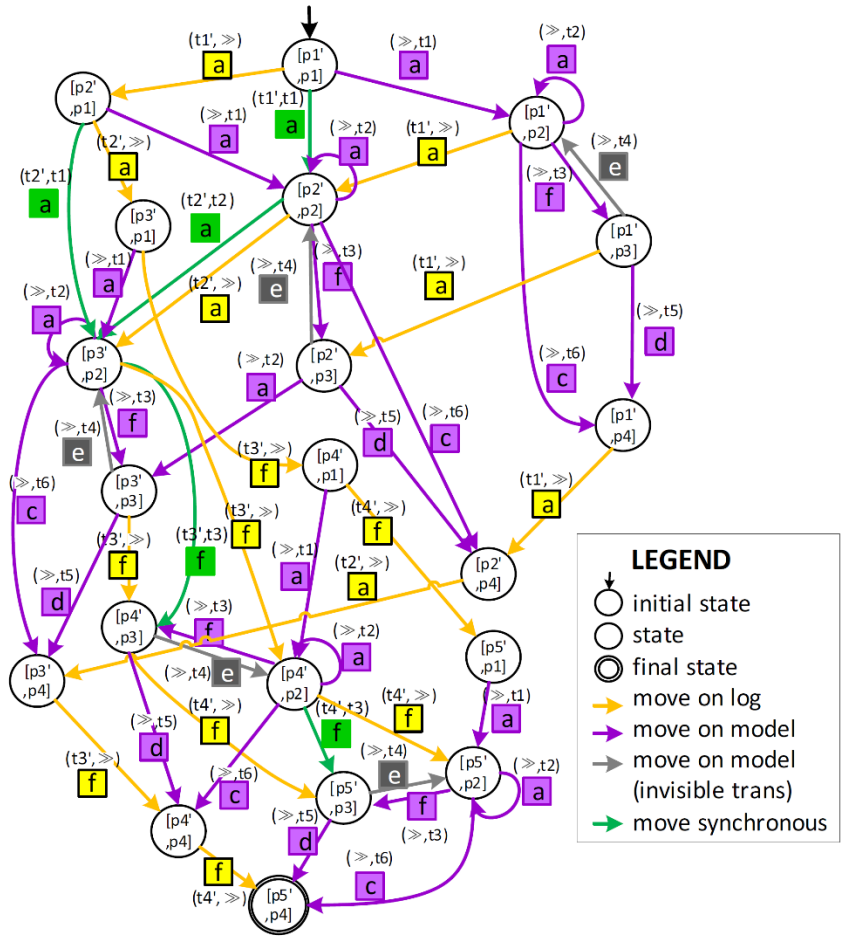
Product net of the process model and the trace model





A shortest path problem
(in a possibly infinite state space)





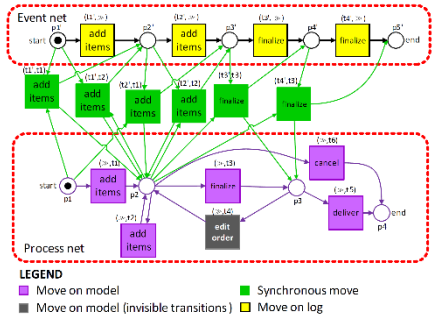
LEGEND

- Move on model
- Move on model (invisible transitions)
- Synchronous move
- Move on log

add items	*add items*	*finalize*	≫	*finalize*	≫
add items t_1	*add items* t_2	*finalize* t_3	*edit order* t_4	*finalize* t_3	*deliver* t_5

Chair of Process and Data Science

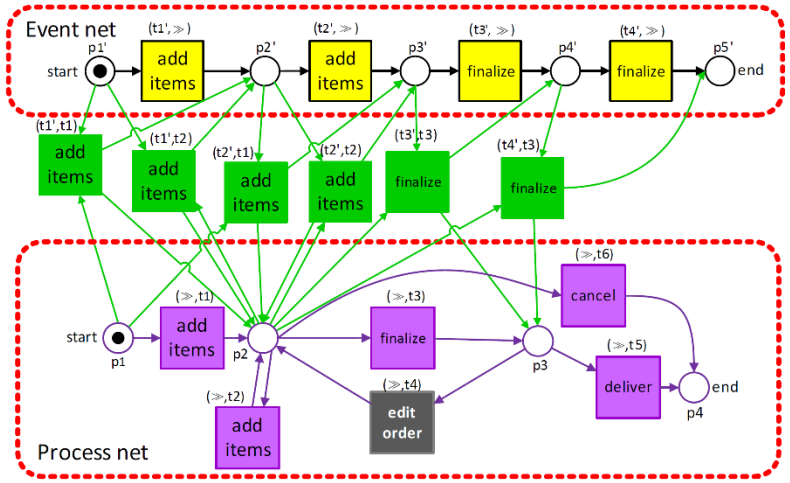
Marking equation of the product net



$$\begin{pmatrix}1\\0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix}+\begin{pmatrix}-1&0&0&0&-1&-1&0&0&0&0&0&0&0&0&0&0\\1&-1&0&0&1&1&-1&-1&0&0&0&0&0&0&0&0\\0&1&-1&0&0&0&1&1&-1&0&0&0&0&0&0&0\\0&0&1&-1&0&0&0&0&1&-1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&-1&0&-1&0&0&0&-1&0&0&0&0&0\\0&0&0&0&1&0&1&0&-1&-1&1&0&-1&1&-1&0\\0&0&0&0&0&0&0&0&1&1&0&0&1&-1&0&-1\\0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\end{pmatrix}\begin{pmatrix}t1'\\t2'\\t3'\\t4'\\t1't1\\t1't2\\t2't1\\t2't2\\t3't3\\t4't3\\t1\\t2\\t3\\t4\\t5\\t6\end{pmatrix}=\begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\1\end{pmatrix}$$

Any path in the product net from start to end is a solution of this system of inequalities

Efficiently computing optimal alignments



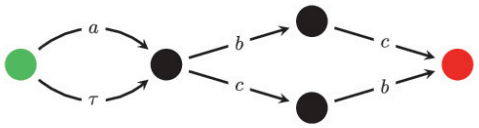
- **The marking equation helps to limit the search space (cut off paths not leading to the end).**
- **Provides very effective lower bounds to reach the end (exploited by e.g. A*)**

$$\begin{pmatrix}1\\0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}-1&0&0&0&-1&-1&0&0&0&0&0&0&0&0&0&0\\1&-1&0&0&1&1&-1&-1&0&0&0&0&0&0&0&0\\0&1&-1&0&0&0&1&1&-1&0&0&0&0&0&0&0\\0&0&1&-1&0&0&0&0&1&-1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&-1&0&-1&0&0&0&-1&0&0&0&0&0\\0&0&0&0&1&0&1&0&-1&-1&1&0&-1&1&-1&0\\0&0&0&0&0&0&0&0&1&1&0&0&1&-1&0&-1\\0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\end{pmatrix} \begin{pmatrix}t1'\\t2'\\t3'\\t4'\\t1't1\\t1't2\\t2't1\\t2't2\\t3't3\\t4't3\\t1\\t2\\t3\\t4\\t5\\t6\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\\1\end{pmatrix}$$

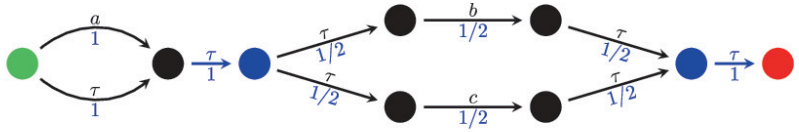
Chair of Process and Data Science

Another Optimization-Based Approach

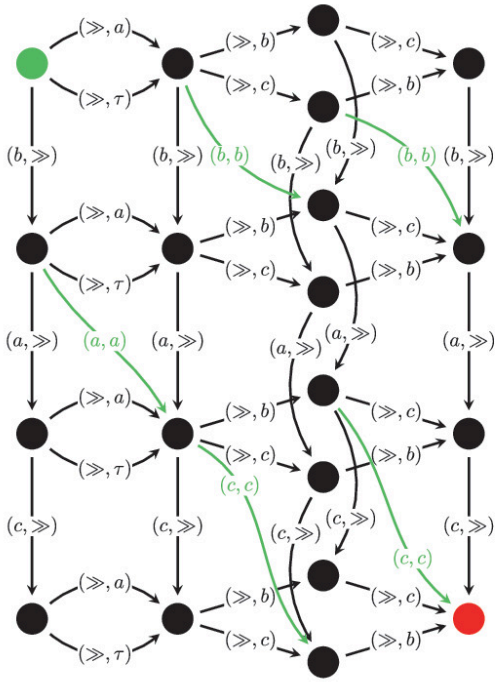
(Based on work with Christopher Schwanen and Wied Pakusa)



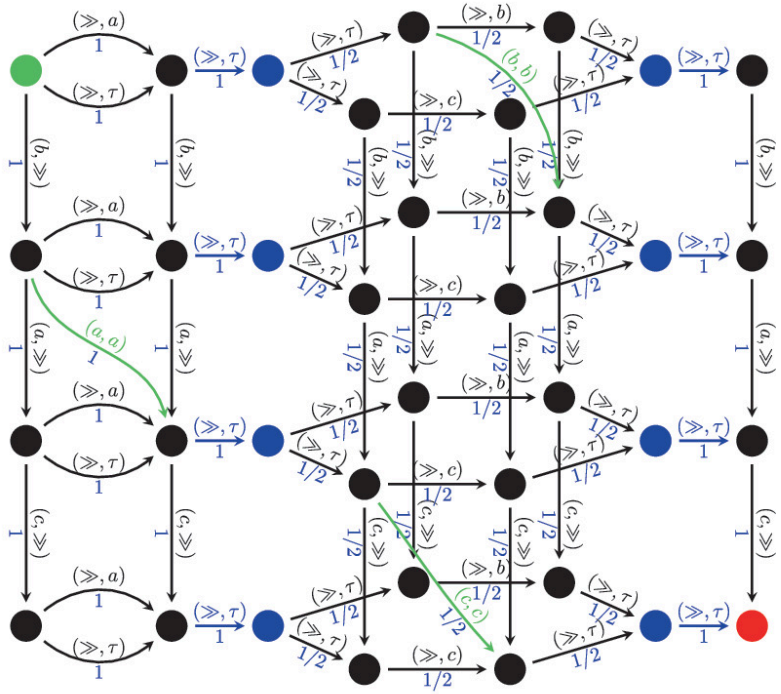
(a) Transition system $\mathcal{TS}(PT)$.



(b) Process tree network $\mathcal{N}(PT)$.



(c) Synchronous product of transition systems TS_σ and $\mathcal{TS}(PT)$.



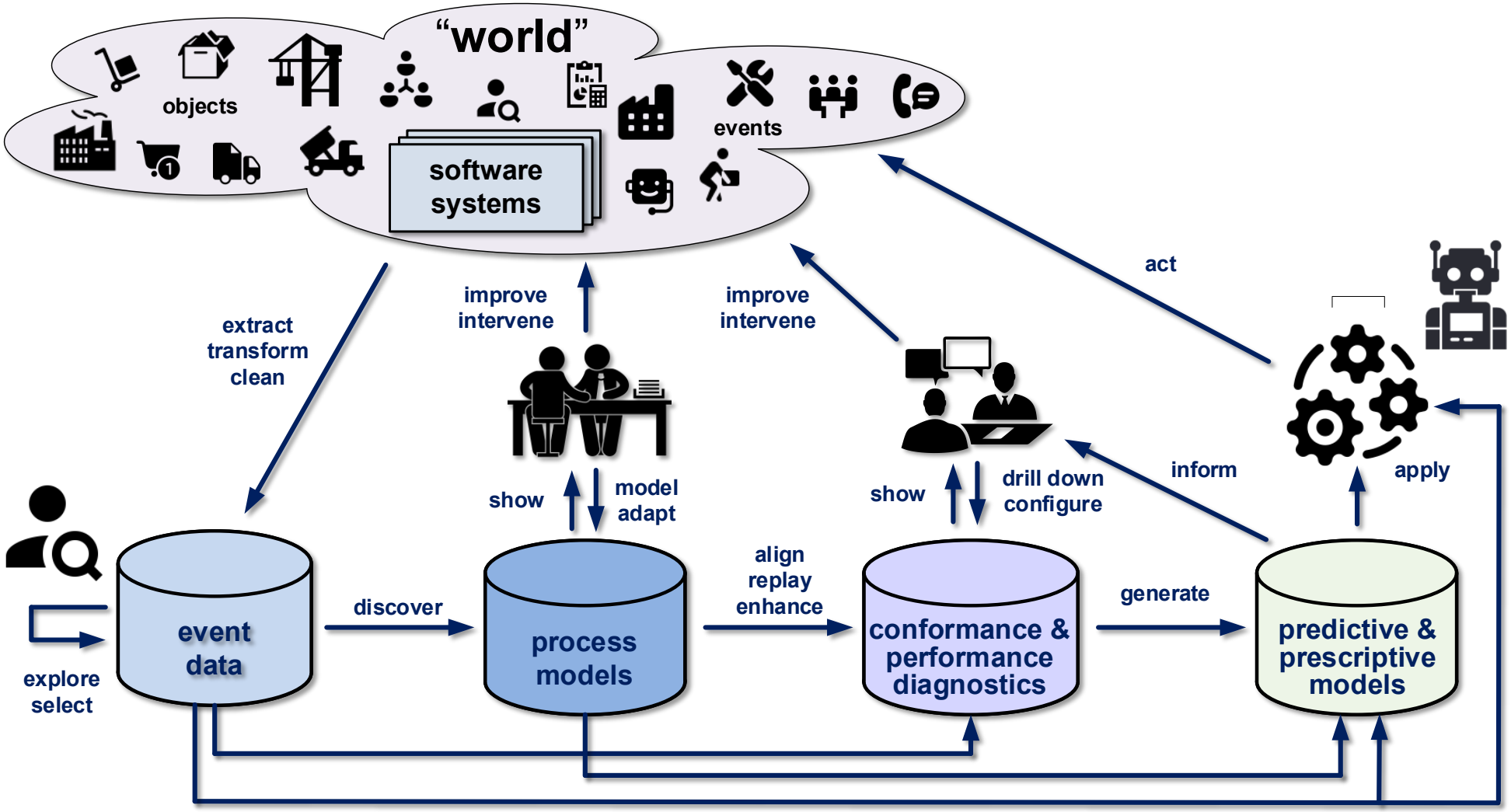
(d) Synchronous network $\mathcal{N}(\sigma, PT)$.

- **Alignments on process trees can be expressed as solutions of Mixed Integer Linear Programs (MILP).**
- **More specifically, we encode the alignment problem as a minimum-cost network flow problem.**
- **For process trees without parallel executions, our MILP formulation becomes a Linear Program (LP), which can be solved efficiently.**
- **The approach can be extended to sound free-choice nets.**

Chair of Process and Data Science

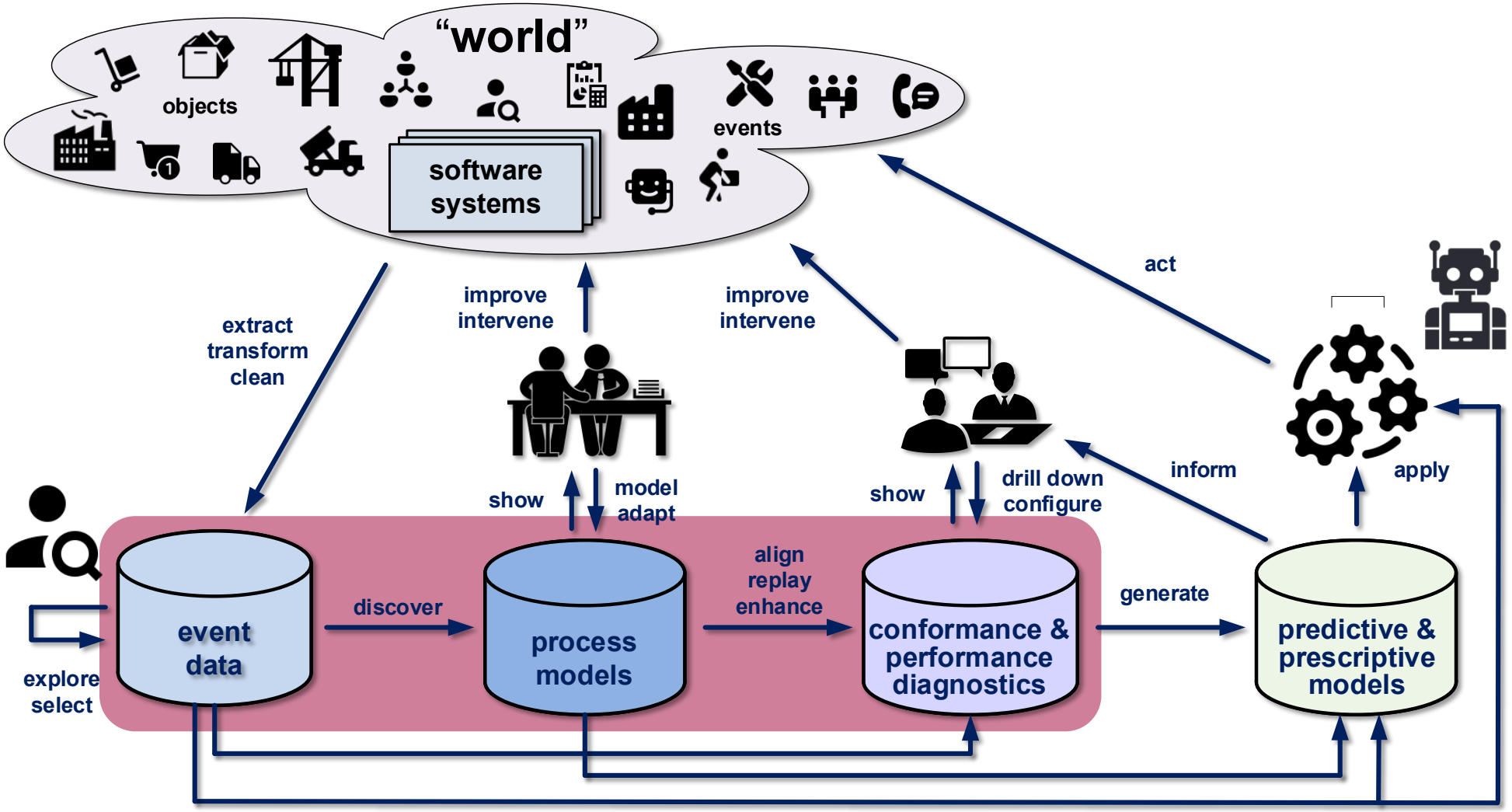
RWTH AACHEN UNIVERSITY

Refining the Overview of Process Mining



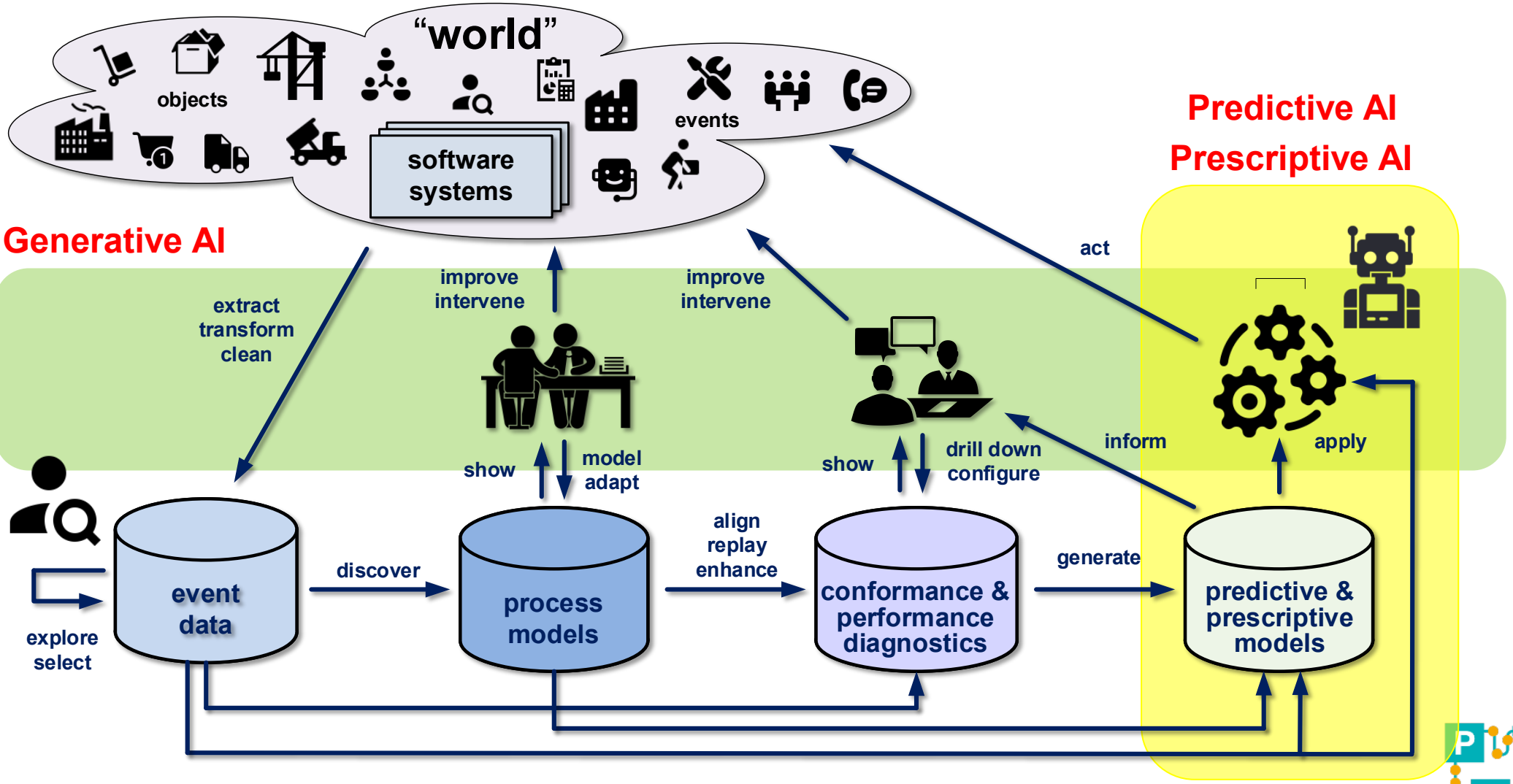
Chair of Process and Data Science

Focus thus far



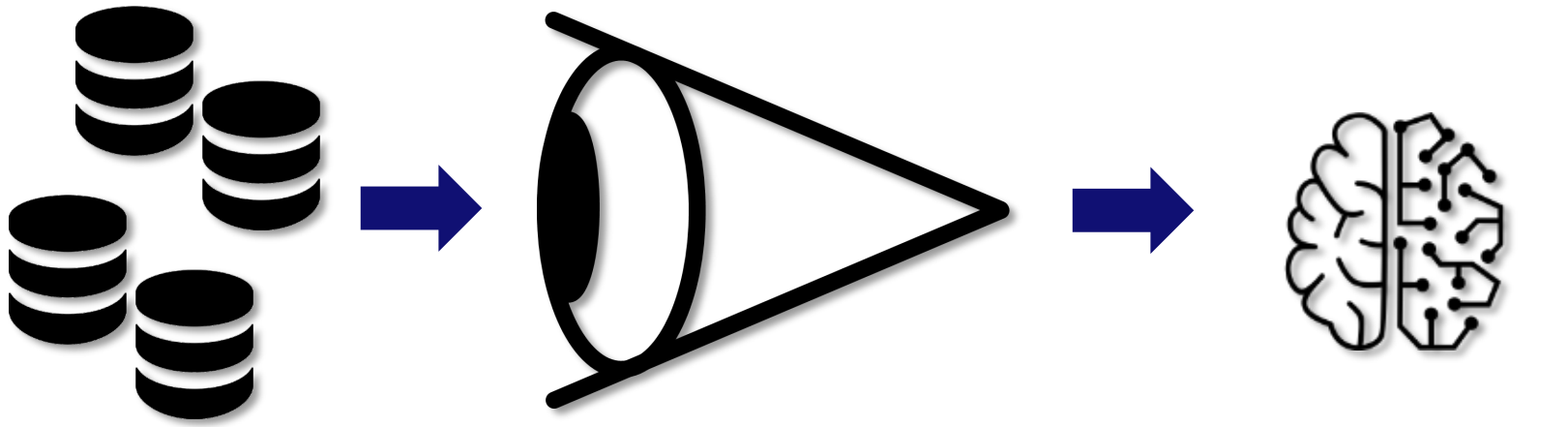
Chair of Process and Data Science

Predictive, Prescriptive, and Generative AI



Chair of Process and Data Science

Process mining provides the lens to look at enterprise data and processes



complex reality

event data and process models

machine learning

Chair of Process and Data Science

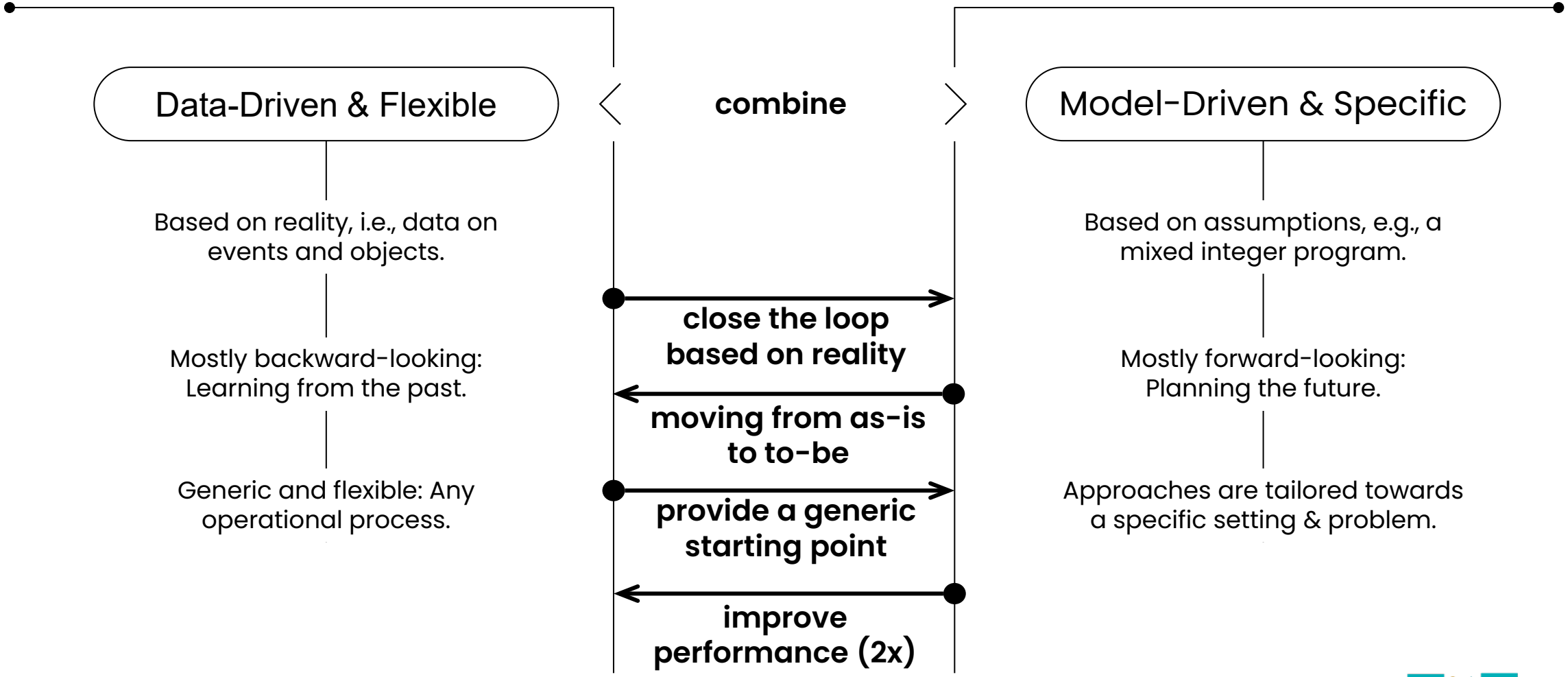
RWTH AACHEN UNIVERSITY

Challenges Optimization

- Involves ad-hoc modeling tailored towards a particular problem.
- No out-of-the-box initial visualization or analysis.
- No standard input representation like in process mining.
- Assumes a “controllable world”, often no feedback loop when the “planned reality” deviates from the “real reality”.

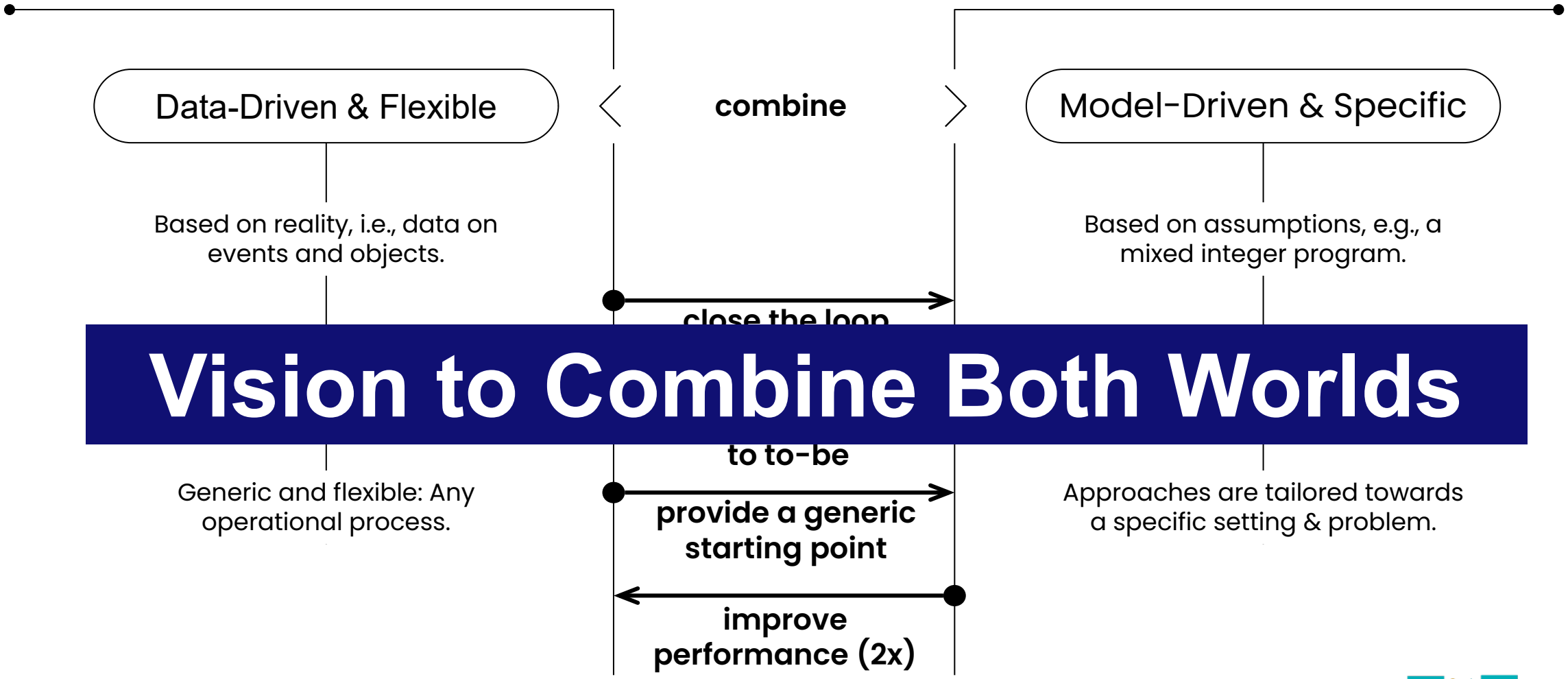

INFORM
celonis
VU
UNITED
lufthansa.com
D-ABVU
SP
Lufthansa
lufthansa.com
D-AISP
NA
First to fly A320neo
Less noise. Less fuel. Less CO_2.
D-AINA
Lufthansa

Process Mining Optimization



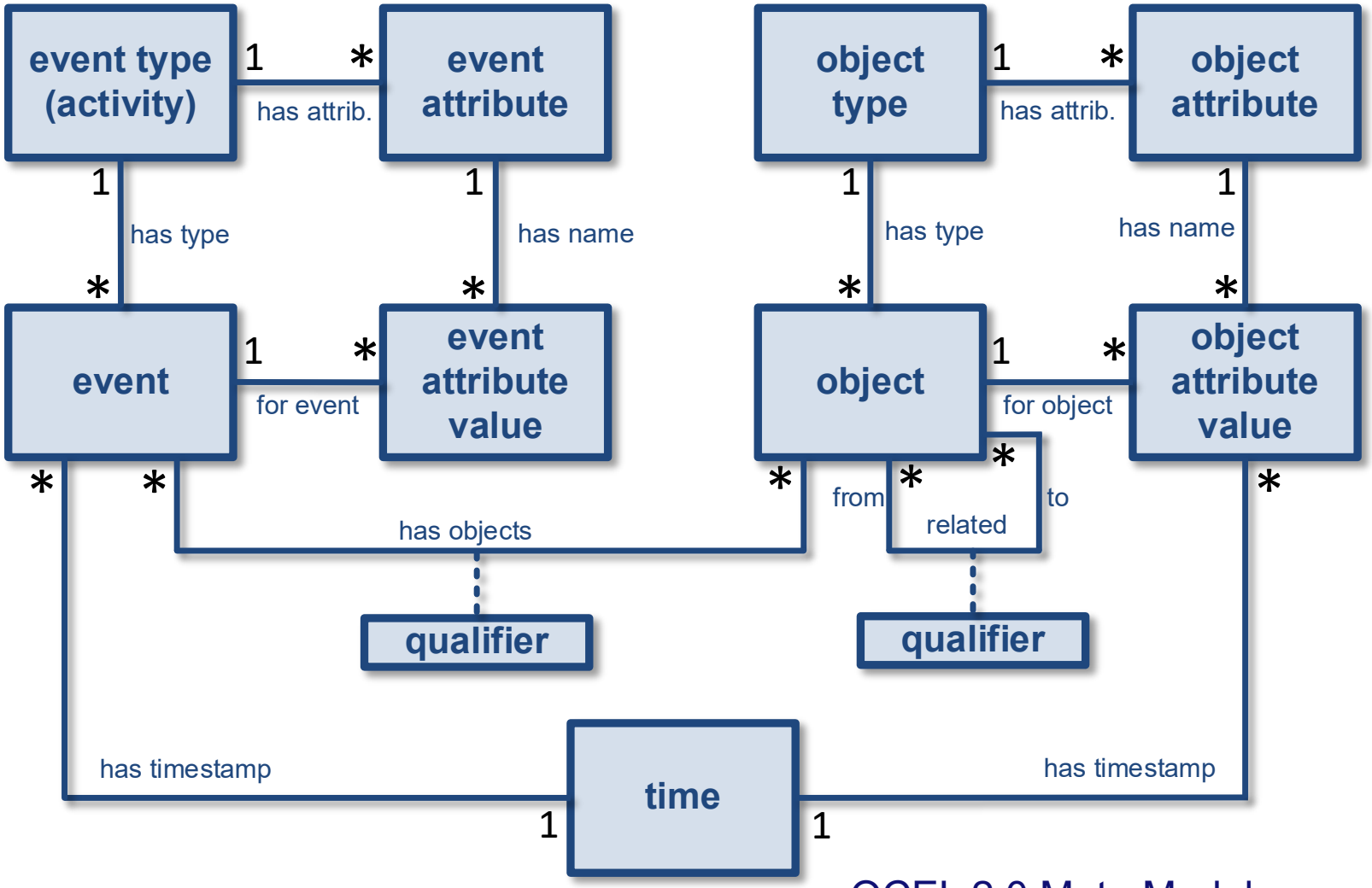
Chair of Process and Data Science

Process Mining Optimization

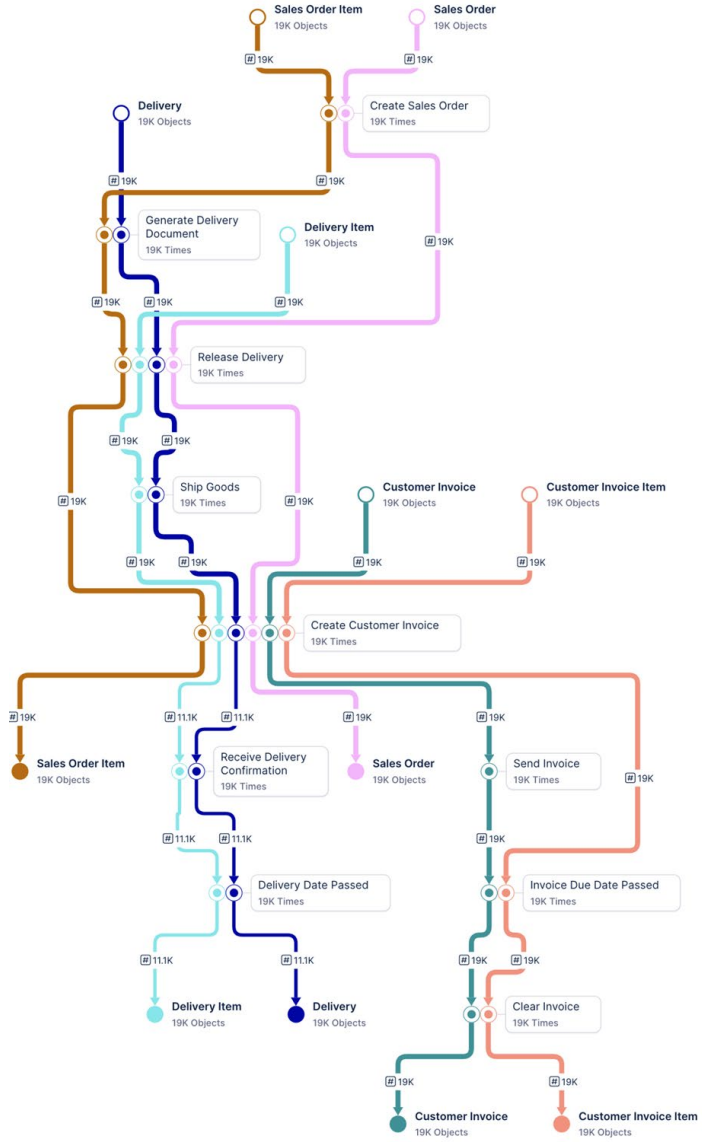


Chair of Process and Data Science

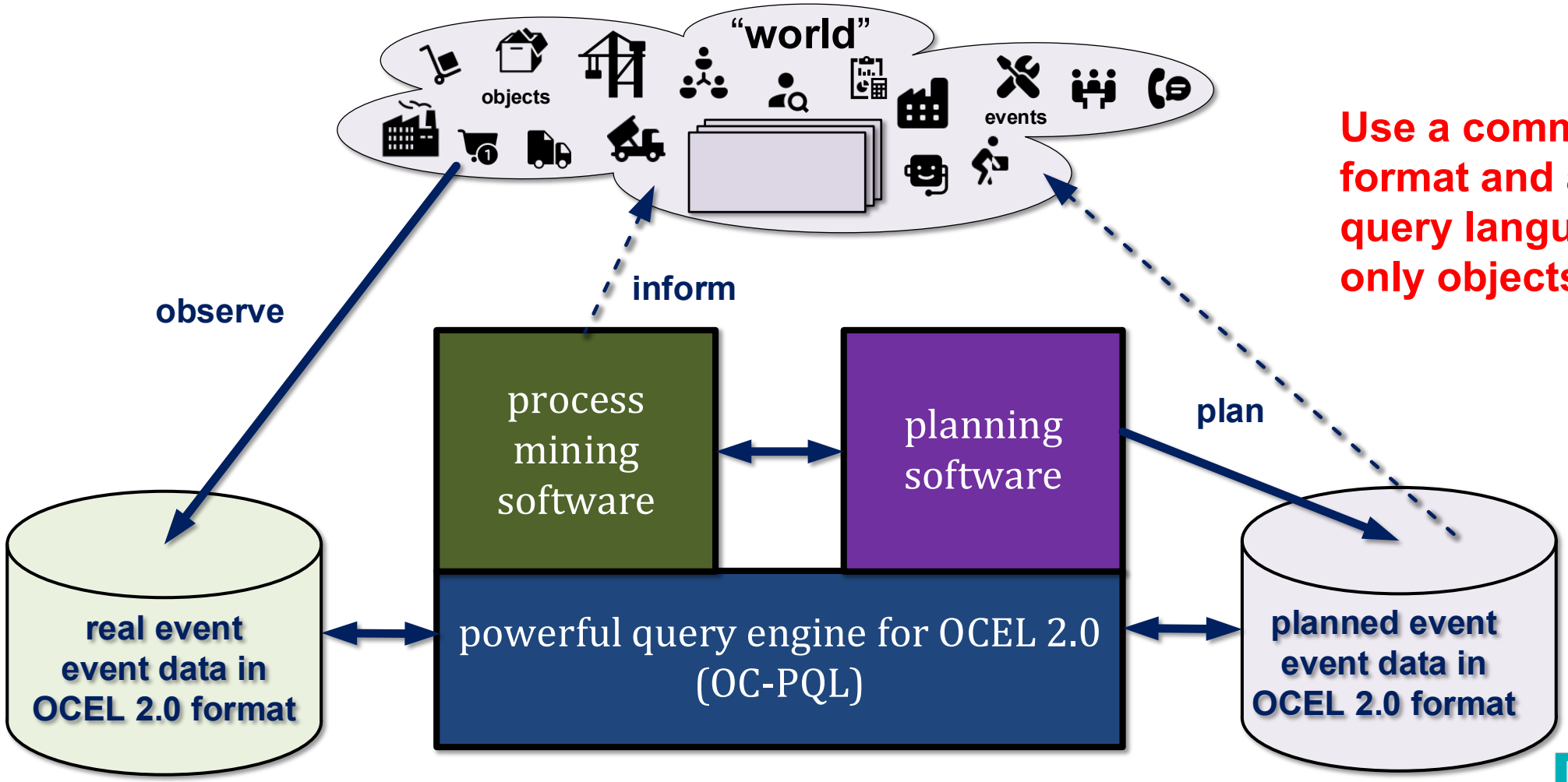
Object-Centric Event Data (e.g. OCEL 2.0) Have Proven To Be Extremely Powerful



OCEL 2.0 Meta Model



Use A Common Language For An “Optimized Reality” and the “Real Reality”



Use a common data format and a common query language using only objects and events

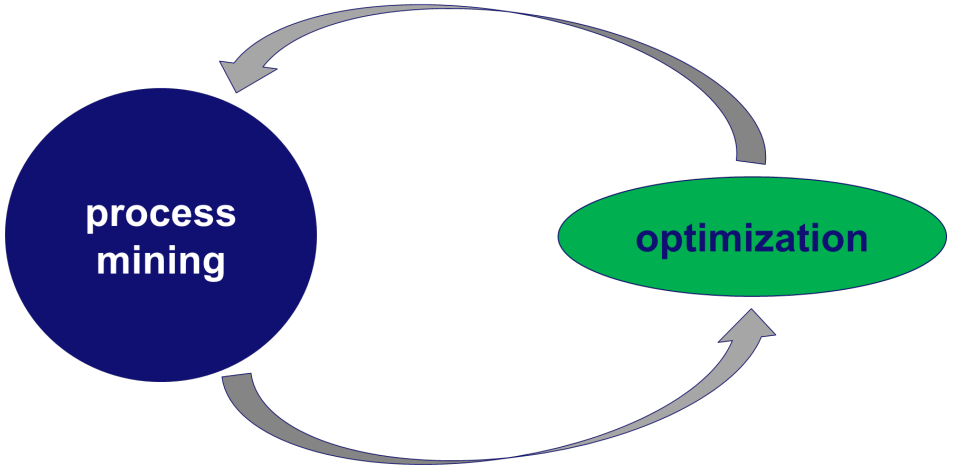
Chair of Process and Data Science

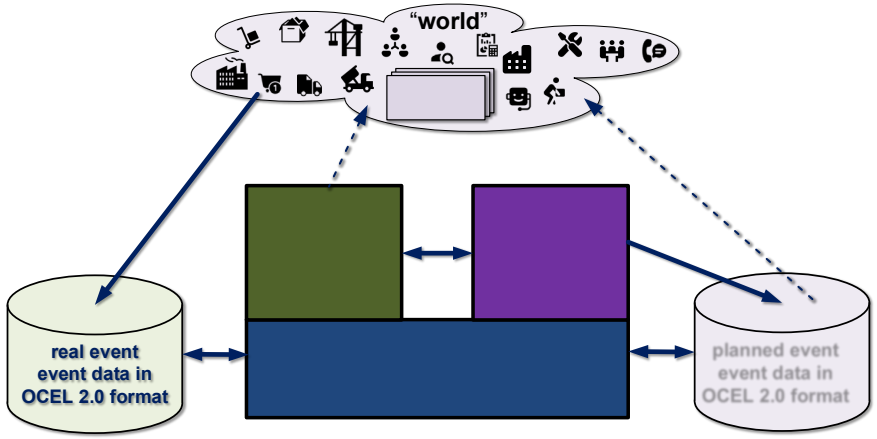
Topic for collaboration?

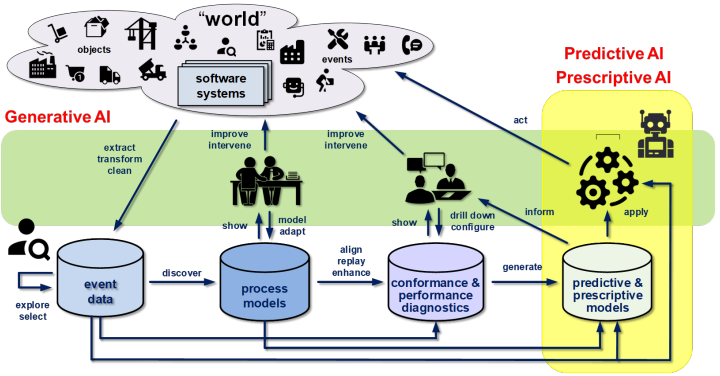
Chair of Process and Data Science

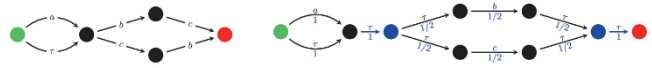
RWTH AACHEN UNIVERSITY

Conclusion



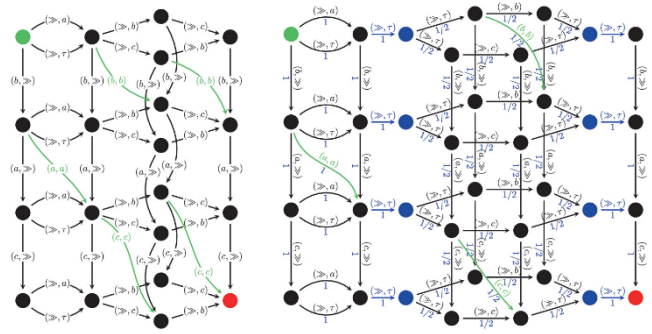






(a) Transition system $\mathcal{TS}(PT)$.

(b) Process tree network $\mathcal{N}(PT)$.



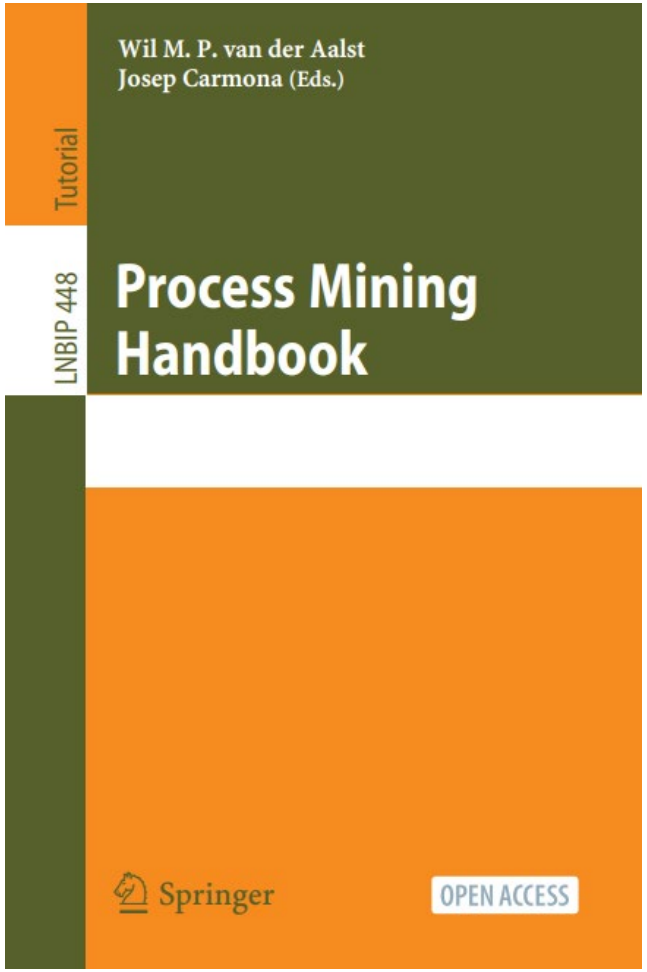
(c) Synchronous product of transition systems TS_σ and $\mathcal{TS}(PT)$.

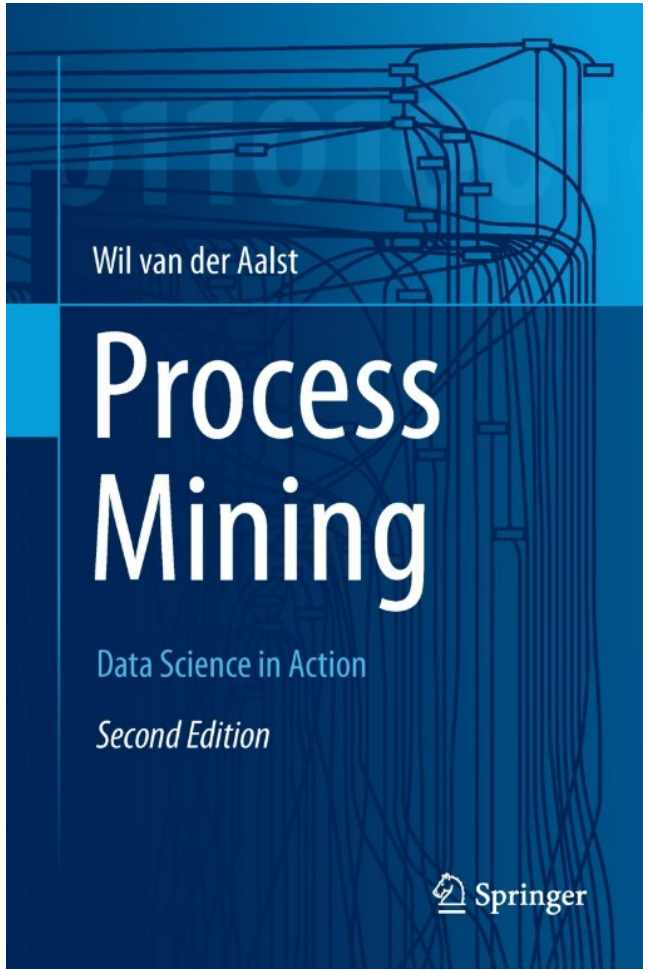
(d) Synchronous network $\mathcal{N}(\sigma, PT)$.

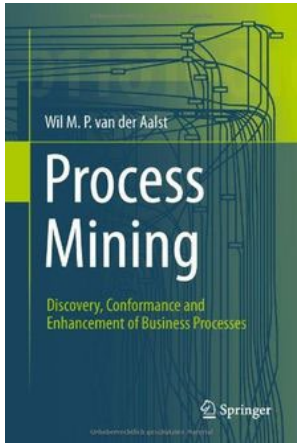
Chair of Process and Data Science

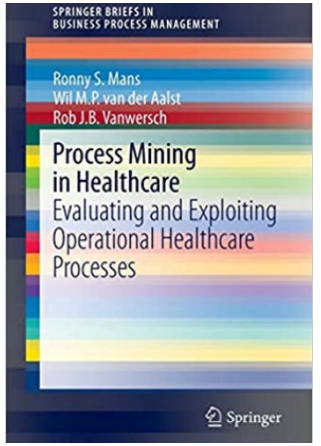
Chair of Process and Data Science
RWTH AACHEN UNIVERSITY

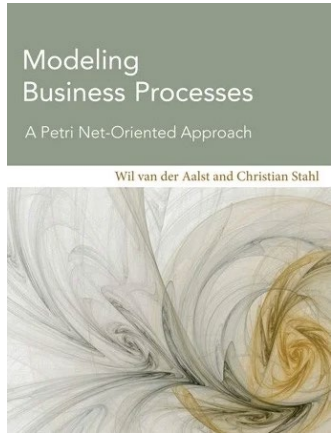
Books (not intended to be complete)

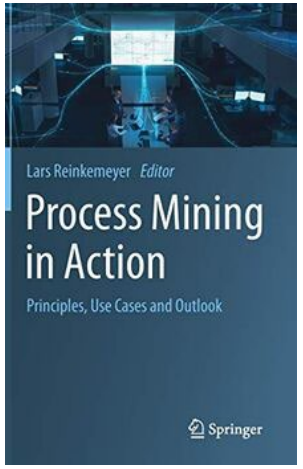


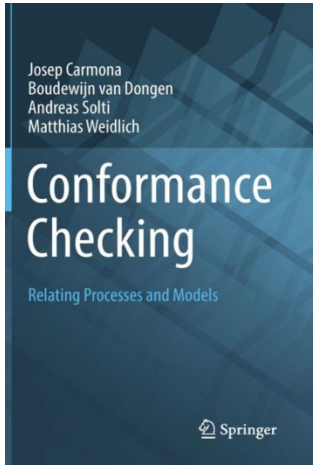












Chair of Process and Data Science

Online courses

- **Coursera course “Process Mining: Data science in Action”**
- **Three RWTHx edX courses**

coursera

edX

Chair of Process and Data Science

Websites

- www.processmining.org
- www.process-mining-summer-school.org
- www.tf-pm.org
- www.promtools.org
- www.celonis.com/academic-signup
- xes-standard.org
- ocel-standard.org
- www.pads.rwth-aachen.de
- www.vdaalst.com

Chair of Process and Data Science