

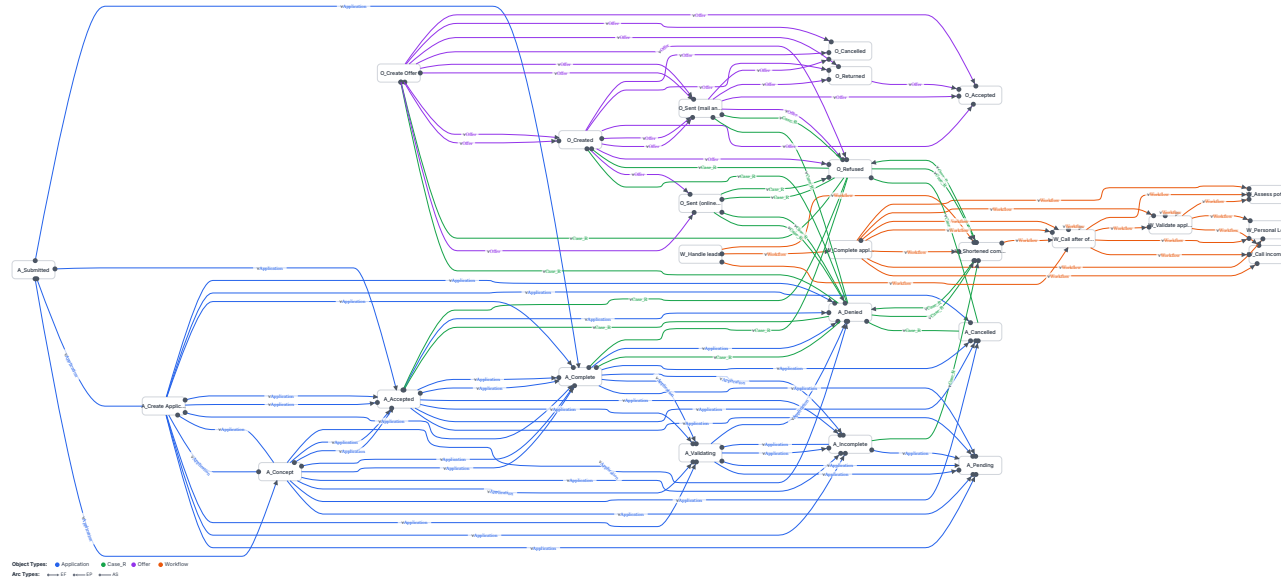
Lossless Simplification of OC-DECLARE Models

OBJECTS Workshop @ BPM 2026

Aaron Küsters, Wil M.P. van der Aalst

2026-09-28

Discovered object-centric model can get messy...



OC-DECLARE model of BPIC2017: 26 activities, 104 constraints

Why simplify?

People

Larger models are harder to understand and reason about.

Tools

Downstream analysis (e.g., conformance checking, AI-assisted) causes increased runtime/costs with larger model size.

Lossless simplification, keeping exactly the same behavior, exists for:

Petri nets

implicit place removal

Declare

transitive reduction

Object-centric

—

Why OC-DECLARE?



Commonly OC-DFGs and OC-Petri nets are discovered as one model **per object type** that are glued together.



They cannot directly express: *"an order is confirmed together with **all** its items"*.

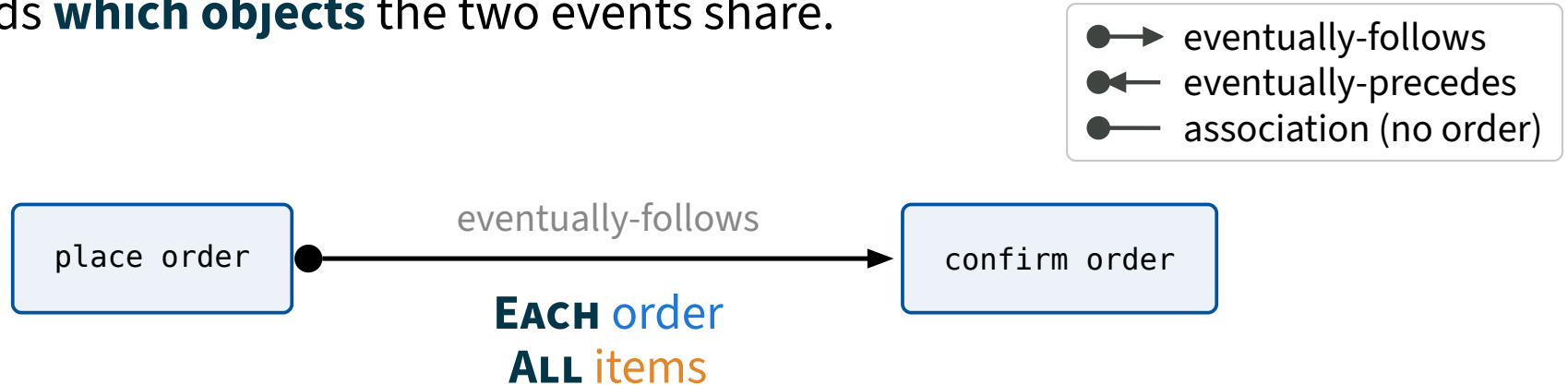


OC-DECLARE constraints can span **several object types** across different *object involvements*.

OC-DECLARE in one arc

Declare's *response*: every *a* is eventually followed by a *b*.

OC-DECLARE adds **which objects** the two events share.



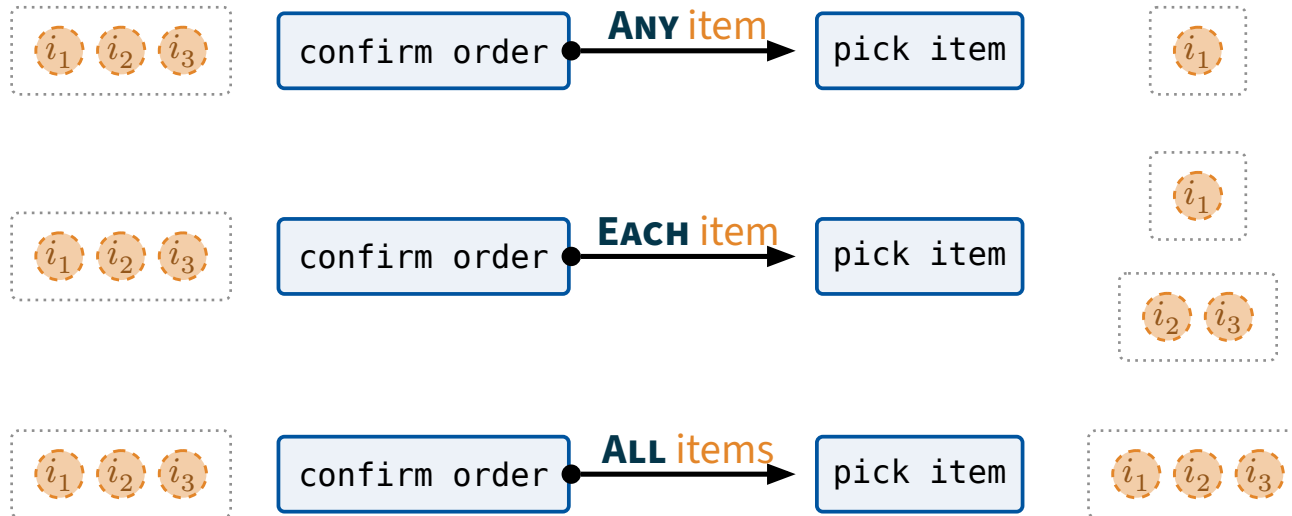
For each **order** of a place order event, a later confirm order event involves that **order** and **all items** of the place order event.

ANY, EACH, ALL

The mode fixes which **items** of the source event the target events involve.
From strict to weak: ALL $\Rightarrow$ EACH $\Rightarrow$ ANY.

source event has

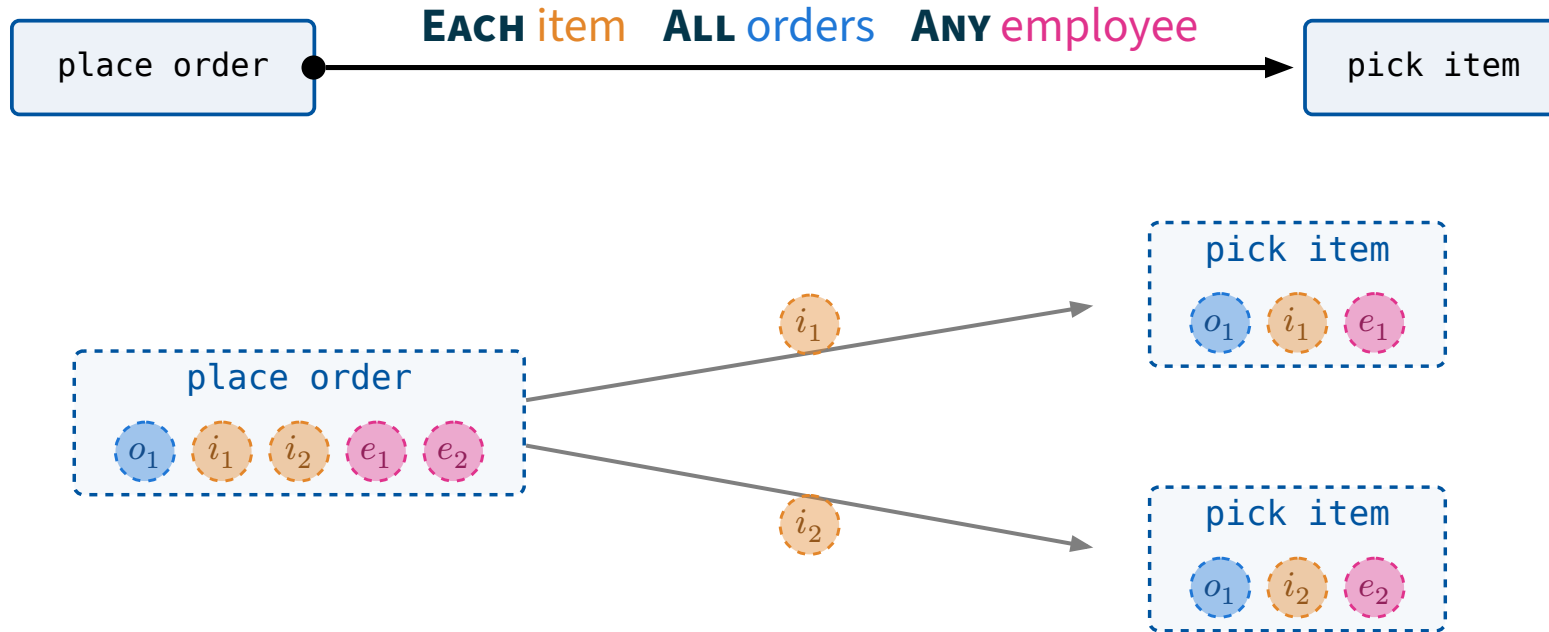
target event(s) need



One arc, multiple object types

Each object type has one involvement type.

one pick item per **item**, with **all** orders and **some** employee of the place order event.



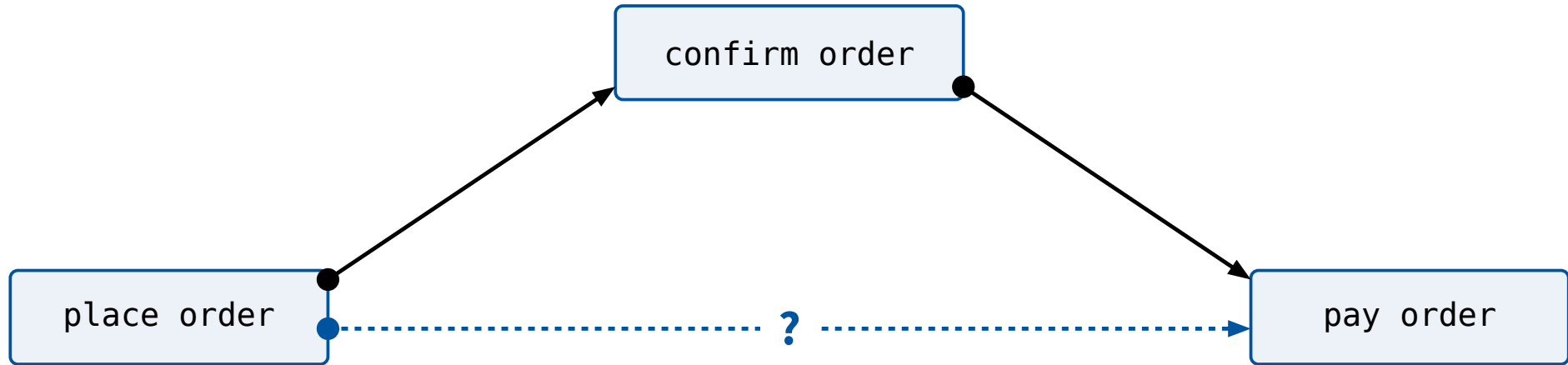
Redundant or not?

I show you a model with one **dashed arc**.
Can we delete it without allowing **any** new behavior?

Vote with your thumb:  **KEEP**  **CUT**

Warm-up: plain Declare

Round 1 / 3



Keep or cut the **dashed arc**? Vote with your thumb!



KEEP

the arc stays

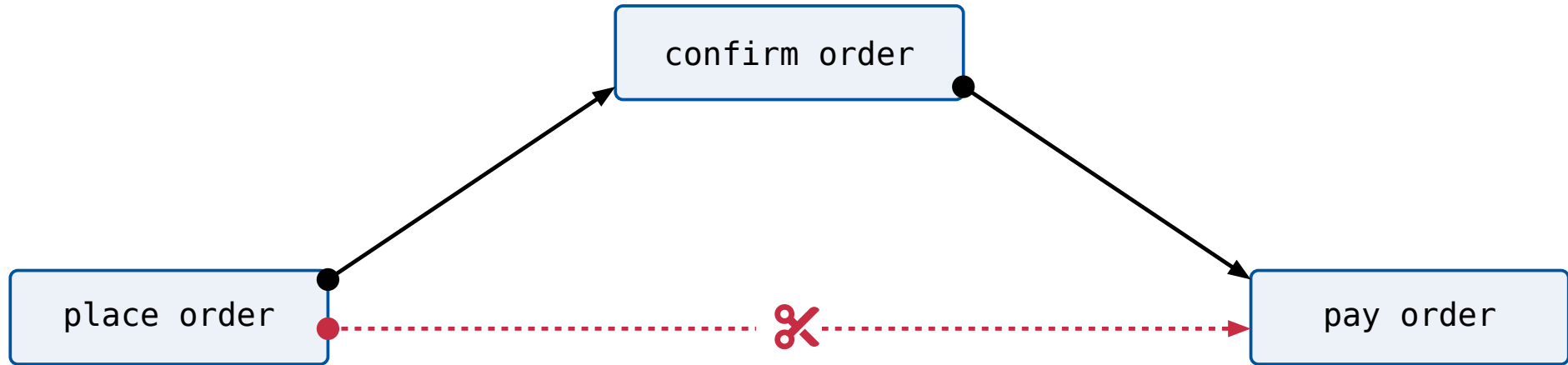


CUT

the arc goes

Warm-up: plain Declare

Round 1 / 3



CUT. Response is transitive: place order → confirm order → pay order.



KEEP

the arc stays

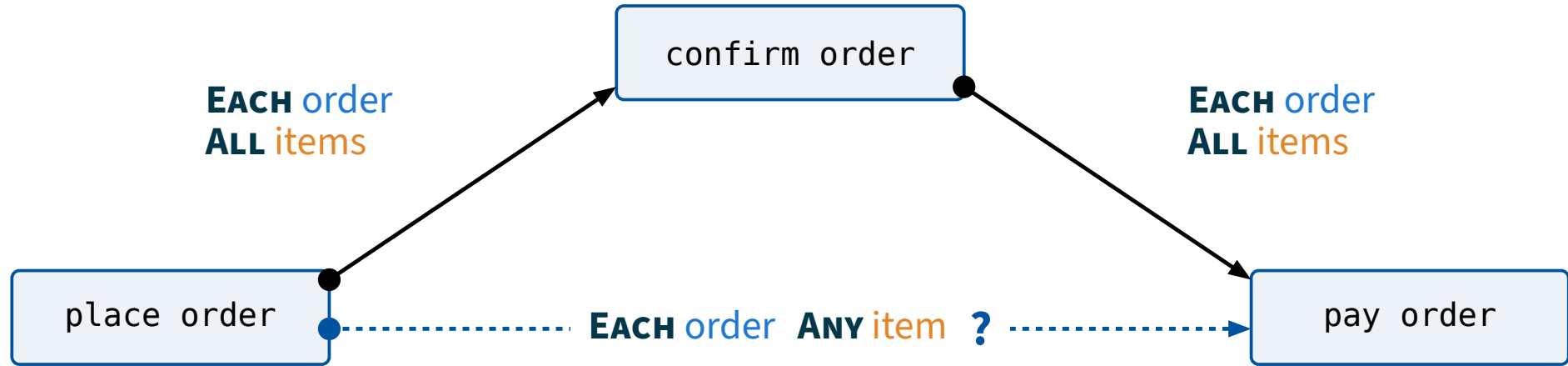


CUT

the arc goes

Adding objects

Round 2 / 3



Keep or cut the **dashed arc**? Vote with your thumb!



KEEP

the arc stays



CUT

the arc goes

Adding objects

Round 2 / 3



CUT. Both hops carry the order and **all** its items, the direct arc only needs **one**.



KEEP

the arc stays

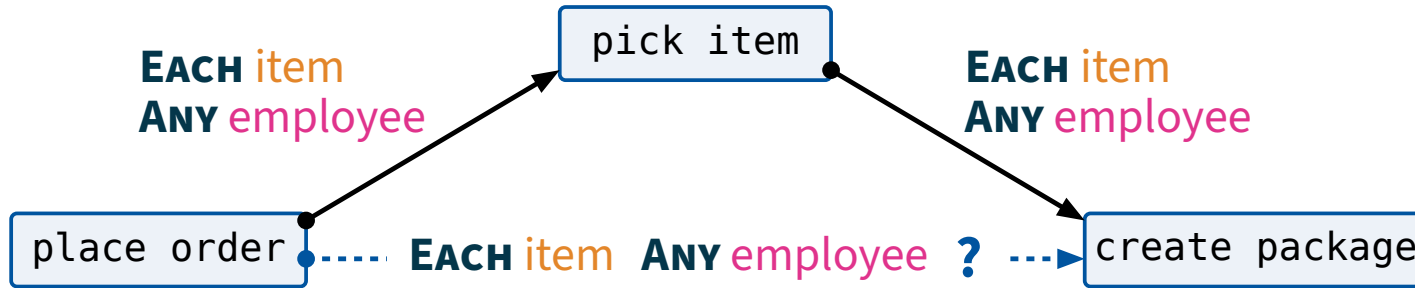


CUT

the arc goes

Adding employees

Round 3 / 3



Keep or cut the **dashed arc**? Vote with your thumb!



KEEP

the arc stays

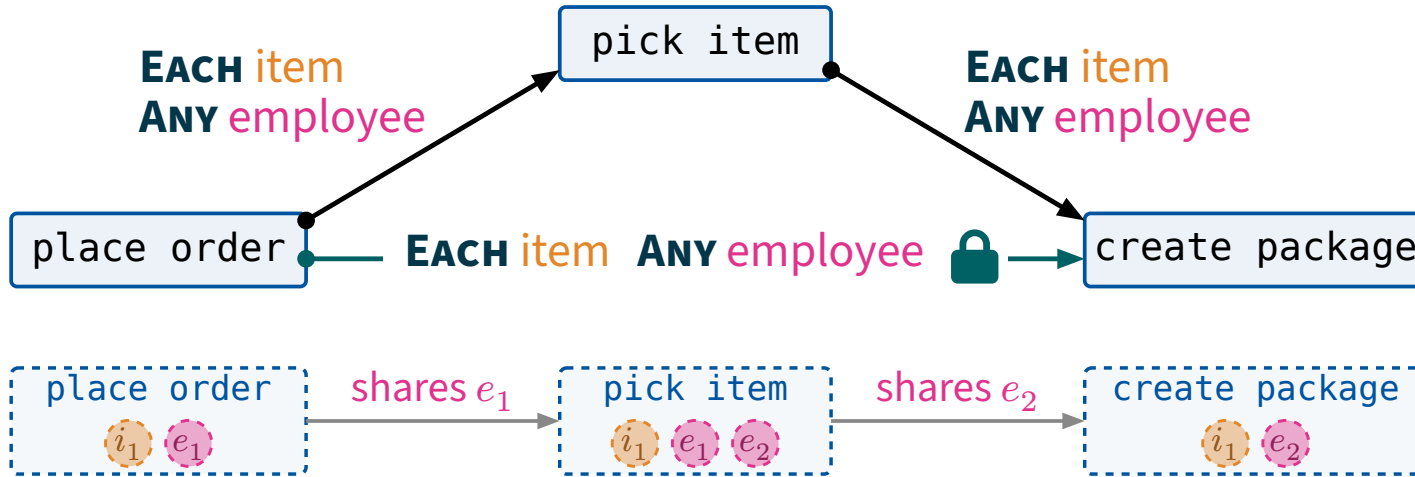


CUT

the arc goes

Adding employees

Round 3 / 3



KEEP! Each hop shares *some* employee, but not the same one.



KEEP

the arc stays

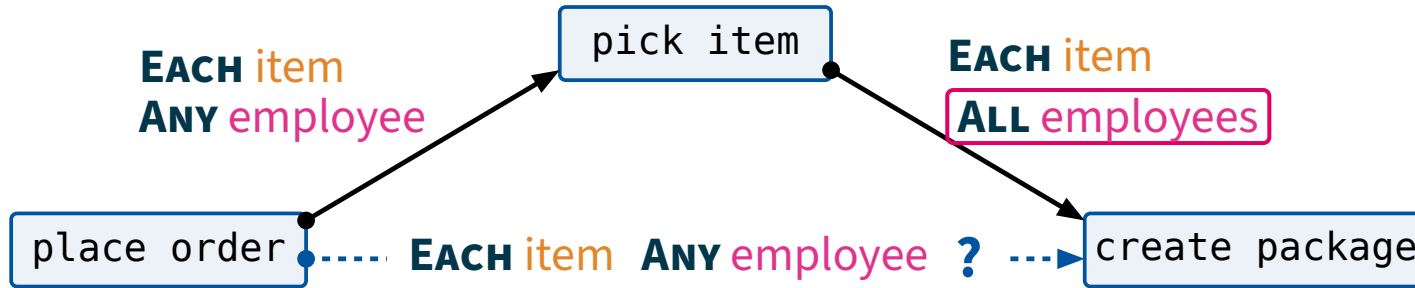


CUT

the arc goes

Adding employees

Round 3 / 3



Follow-up: the second hop now needs ALL employees. Keep or cut?

 **KEEP**

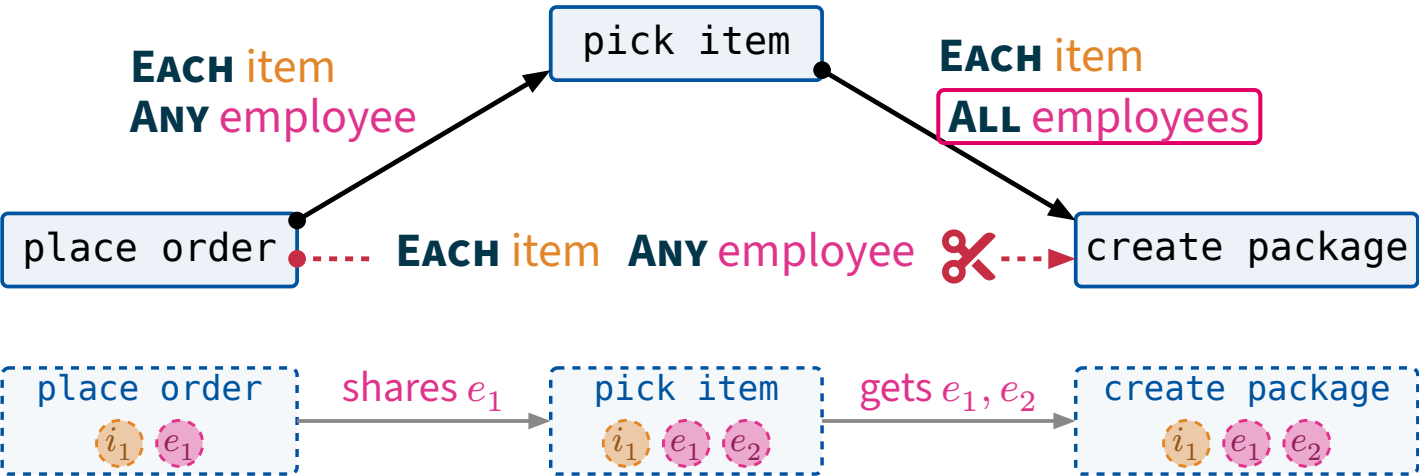
the arc stays

 **CUT**

the arc goes

Adding employees

Round 3 / 3



CUT. create package gets **all** employees of pick item, so also e_1 .

KEEP
 the arc stays

CUT
 the arc goes

Reduction rules

A direct arc is implied by a path $d_1, \dots, d_n$ from its source to its target if:

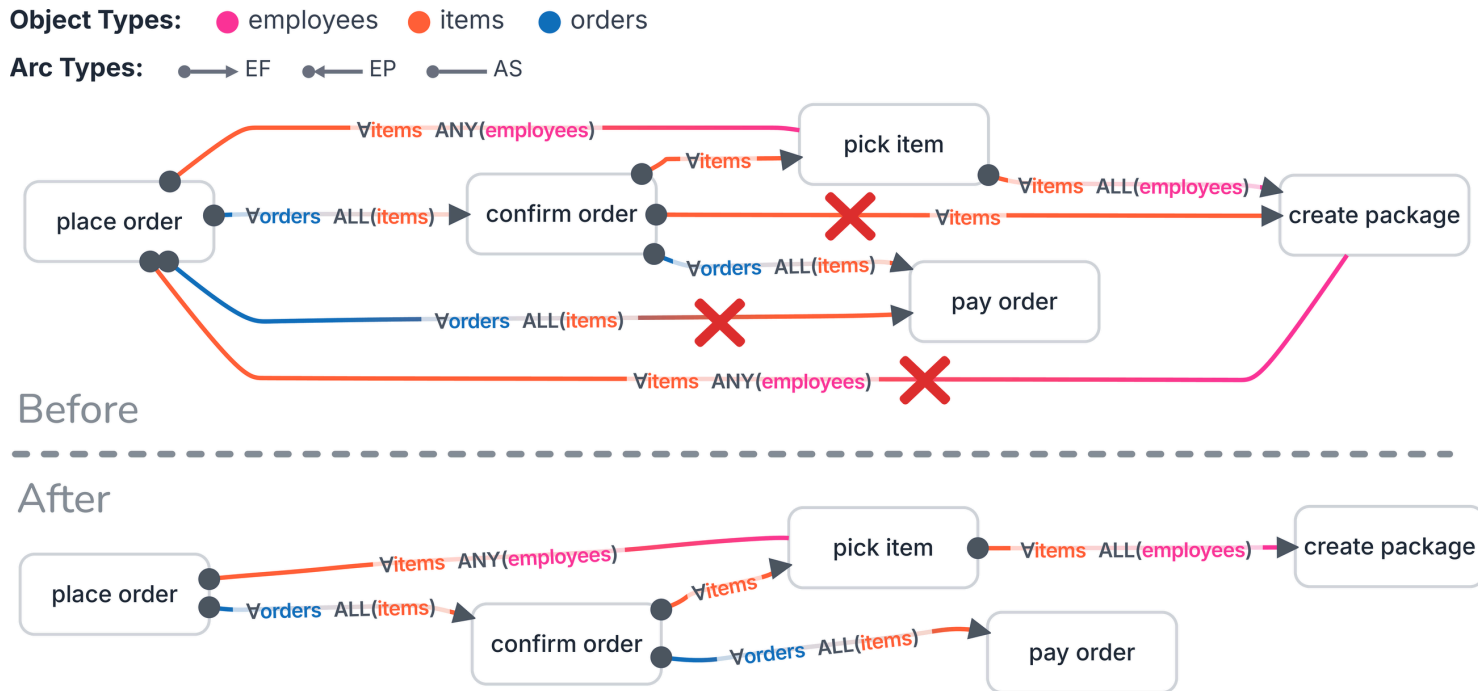
- **same arrow type** on every hop, or the direct arc is an association **Round 1**
- ↓≡ every hop is **at least as strict** per object type **Round 2**
- ⚠ a type with ANY on the direct arc may be ANY **only on the first hop** **Round 3**

✓ **Theorem:** the reduced model allows exactly the same logs.

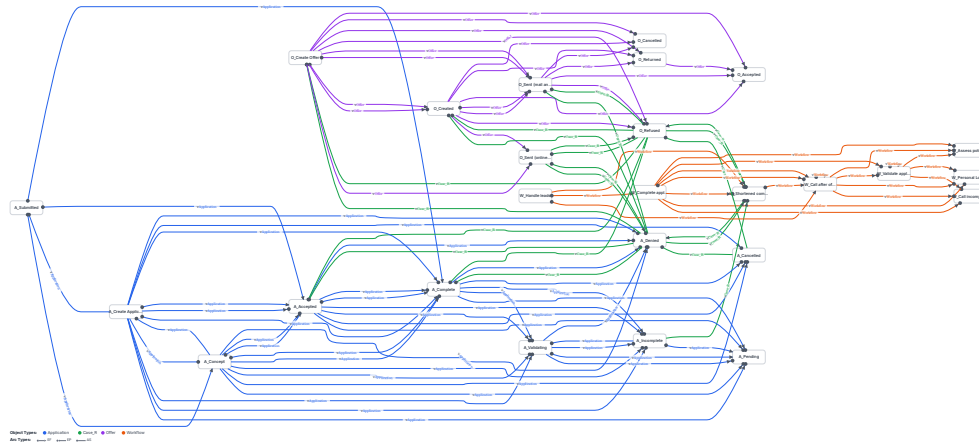
$$\mathcal{L}((A, \mathcal{D} \downarrow)) = \mathcal{L}((A, \mathcal{D}))$$

The full example

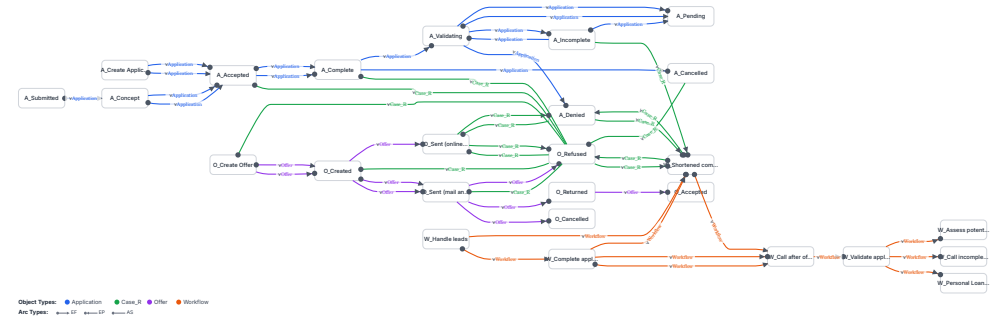
In this example, 3 out of 8 arcs are removed. ($\forall$ = EACH).



Results: BPIC2017 before and after

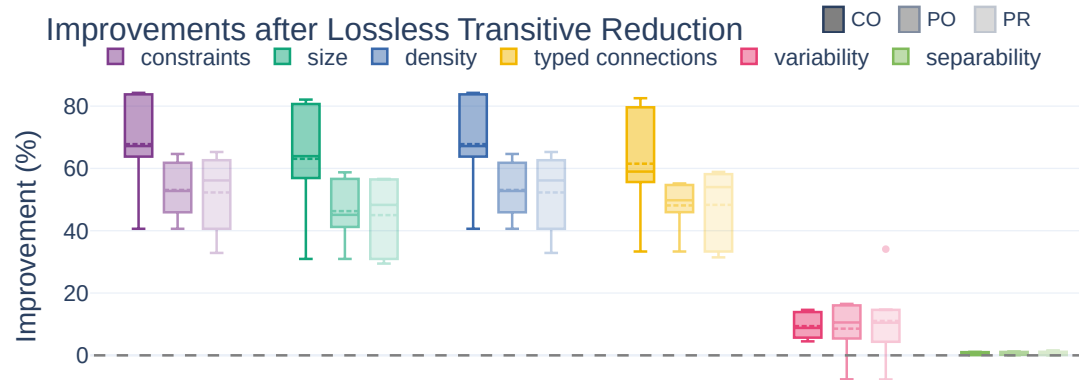


discovered: **104** constraints



after lossless reduction: **49** constraints

Results on six event logs



Six logs: P2P, O2C, Logistics, BPIC2014, BPIC2017, BPIC2019. **CO**: complete discovery. **PO** / **PR**: lossy pruning after / during discovery.

-67%

median constraints (CO)

-84%

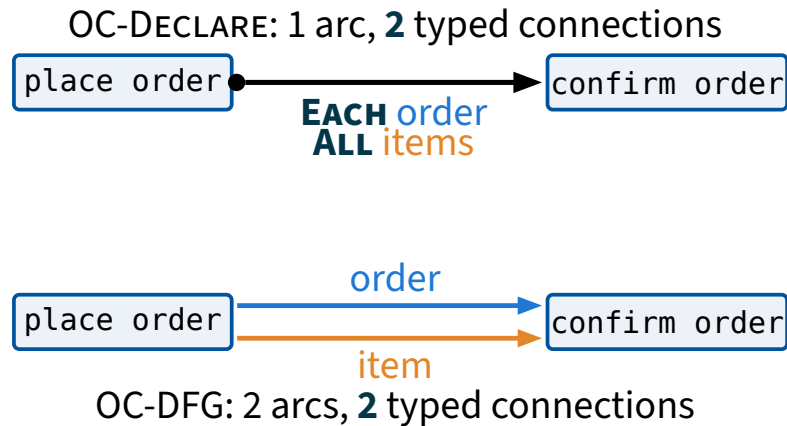
BPIC2014: 12,541 → 1,974

< 0.3 s

reduction time per log

Typed connections

Typed connections: the number of object types on each arc, summed over all arcs.
OC-DFG and OCPN arcs carry one object type each, so there it is the number of arcs.

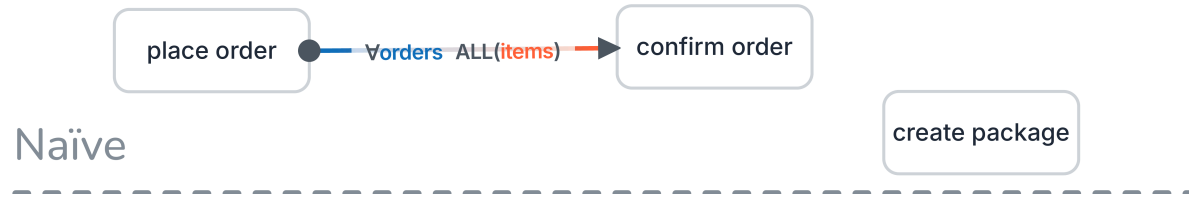


Log	OC-DECLARE	OC-DFG	OCPN
P2P	30	52	86
O2C	152	244	160
Logistics	59	76	134
BPIC2014	3050	16836	3394
BPIC2017	121	412	150
BPIC2019	622	2708	362

After reduction (CO), OC-DECLARE has fewer typed connections than the OC-DFG on all six logs and than the OCPN on five.

Zooming in: Activity Projection

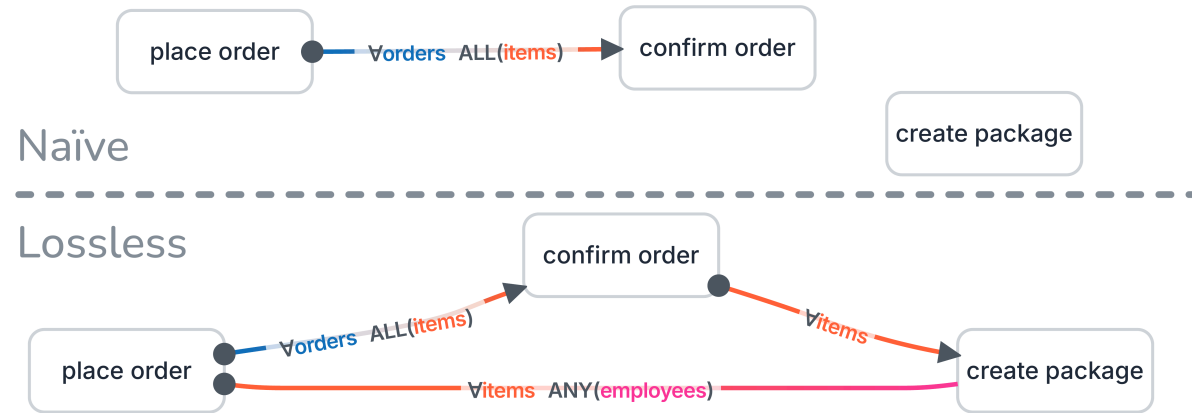
Even after reduction, BPIC2014 keeps 1,974 constraints. Analysts often look at a few activities only, e.g., place order, confirm order and create package.



Dropping every arc that touches another activity cuts off create package: its dependencies ran through the hidden activities.

Zooming in: Activity Projection

Even after reduction, BPIC2014 keeps 1,974 constraints. Analysts often look at a few activities only, e.g., place order, confirm order and create package.



A lossless projection keeps them: create package stays connected through arcs implied by paths.

Lossless Activity Projection



Step 1 uses the same rules as the game, so no dependency between the chosen activities is lost.

On the six logs, naive filtering misses **48-74%** of the constraints for random activity subsets and **10-36%** for the most frequent activities. The lossless projection keeps all of them.

Open questions

- 👁️ **Complexity metrics:** size and density ignore object types
- ⚖️ **Comparing formalisms:** arcs differ in how much they say
- 🚫 **Other constraints:** directly-follows, negative

Try it



Integrated in OCPQ:

<https://ocpq.aarkue.eu/>

Implemented in Rust4PM: Run

discovery, reduction and projection from Python, Rust and graphical tools

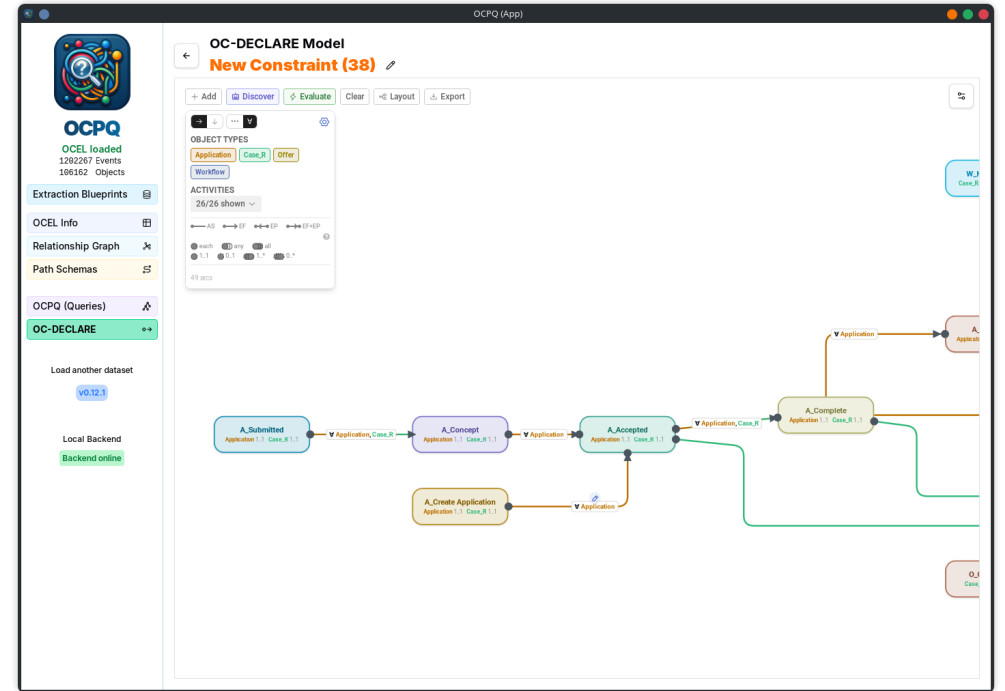


<https://rust4pm.aarkue.eu/>



Code + evaluation results:

github.com/aarkue/oc-declare-red



Takeaways

- ✂ About two thirds of discovered OC-DECLARE arcs are **implied by others**. Removing them does not change the allowed behavior.
- ⚠ ANY is **not transitive**, so plain graph transitive reduction would drop arcs that have to stay.
- ✂ The same rules give a **lossless activity projection**.

👍 Run lossless reduction after every OC-DECLARE discovery.

Thank you

Thanks!

 github.com/aarkue/oc-declare-red

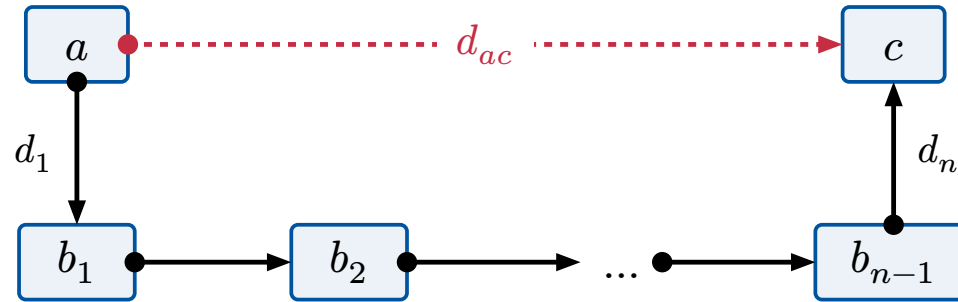
 ocpq.aarkue.eu

 kuesters@pads.rwth-aachen.de



Bonus

Formal: transitivity relation



$d_{ac} \in \mathbf{Trans}(\{d_1, \dots, d_n\})$ if (1) $d_{ac} \neq d_i$, (2) $ar_{ac} \preceq ar_i$, (3) $oi_{ac} \preceq oi_i$ for all i , and (4) $oi_{ac}^{\text{Any}} \cap oi_i^{\text{Any}} = \emptyset$ for $i \geq 2$.

Arrow domination: $ar' \preceq ar$ iff $ar = ar'$ or $ar' = \text{assoc.}$ Involvement: $\text{ANY} \preceq \text{EACH} \preceq \text{ALL}$ per type.

Proof sketch: language equivalence

✓ Fitness ($\supseteq$)

$\mathcal{D} \downarrow \subseteq \mathcal{D}$: constraints are only removed.

🎯 Precision ($\subseteq$)

Induction on path length. Case $n = 2$ fixes shared objects on hop 1. Condition (4) makes ANY types EACH/ALL on hop 2, so they reach the target.

Reduction: process arcs in fixed order, drop an arc if a path of still-active arcs implies it. $O(m^2r)$ for m arcs and r object types. Unique for acyclic models.

Full evaluation table

Dataset	$ \mathcal{ET} $	$ \mathcal{OT} $	$ E $	$ O $	$ \mathcal{D}\downarrow $			Typed Connections $TC(\mathcal{D}\downarrow)$				
					CO	PO	PR	CO	PO	PR	OC-DFG	OCPN
P2P	10	7	14671	9543	19	19	19	30	30	30	52	86
O2C	11	6	21008	10840	70	55	25	152	128	37	244	160
Logistics	14	7	35413	13910	42	35	34	59	52	41	76	134
BPIC2014	330	7	690622	228885	1974	2422	2573	3050	3454	3165	16836	3394
BPIC2017	26	4	1202267	106162	110	48	49	121	59	49	412	150
BPIC2019	42	4	1595923	330685	276	199	243	622	509	388	2708	362

$|\mathcal{D}\downarrow|$ and typed connections after reduction, per discovery mode: complete (CO), post-pruned (PO), pre-pruned (PR). OC-DFG and OCPN counts via pm4py.

Discovery modes: CO, PO, PR

- ▼ **CO**: complete discovery. **PO** / **PR**: prune after / during discovery with a lossy filter that drops constraints matching more than $N = 20$ target events.
- ⚠ Fewer typed connections than CO, but not always fewer constraints after reduction: Pruning also removes **transitive bridges**.
- ↓ BPIC2014: CO 12,541 → 1,974 (– 84%). PO/PR start from 6,342 / 6,339 and end at 2,422 / 2,573.
- ⌚ Discovery: 40 ms to 20 s, PR fastest (under 6 s).