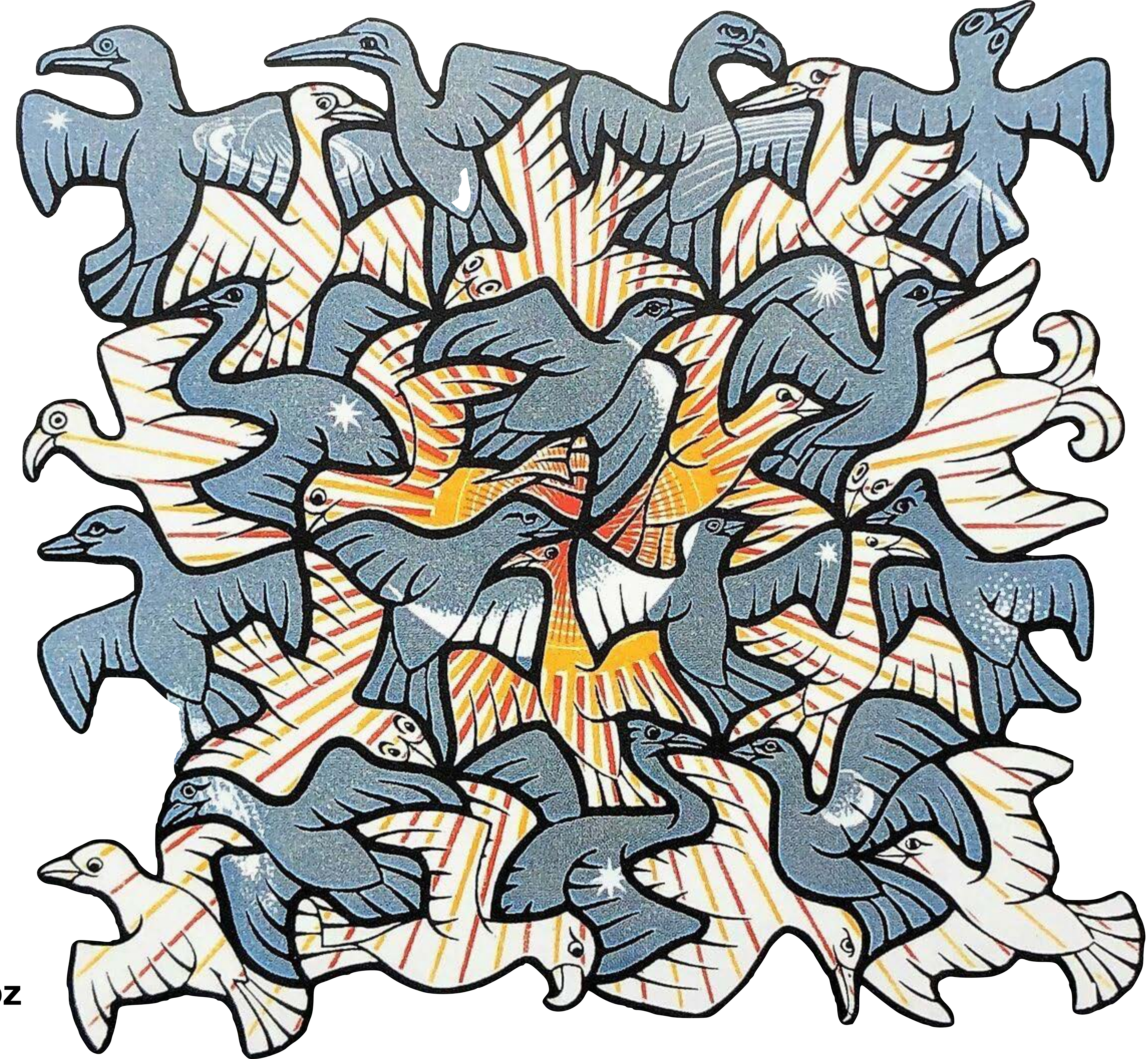
Artificial intelligence techniques for trustworthy information systems

Marco Montali
Free University of Bozen-Bolzano

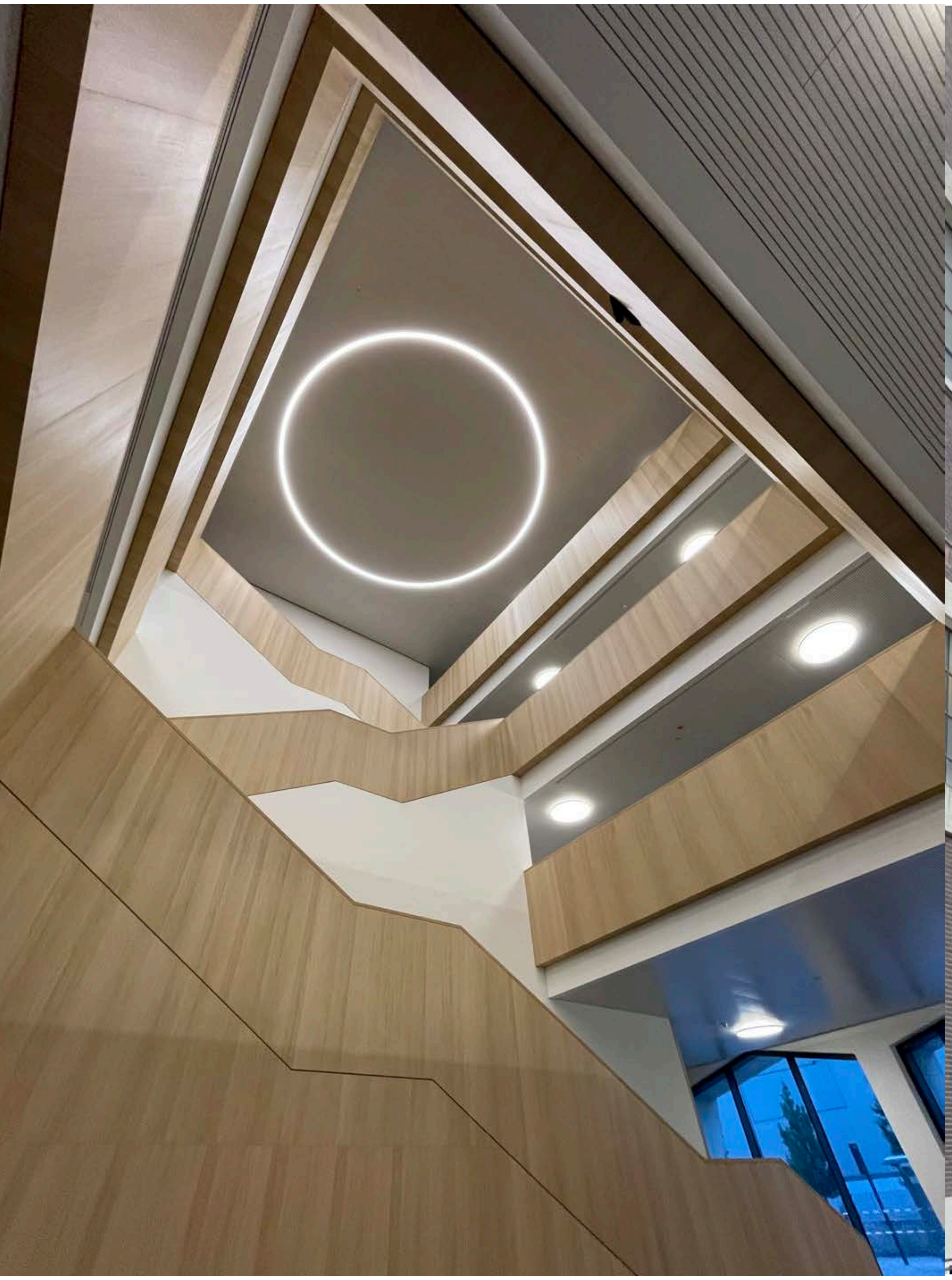
unibz

EUTRAIL workshop, 13/12/2024, unibz



General context

Faculty of engineering at unibz




850+ students, 40+ professors, 50+ researchers, 5 bachelor programmes, 5 master programmes, 3 applied masters, 3 PhD programmes


wood technology
mechanics
automation and robotics
energetics
logistics and production


automation and robotics
mechanics
energetics
wood technology
logistics and production
electronics and sensors

wood technology
mechanics
automation and robotics
energetics
logistics and production
electronics and sensors

automation and robotics
mechanics
energetics
wood technology
logistics and production
electronics and sensors
cyber-physical systems
computer science
artificial intelligence


automation and robotics
mechanics
energetics
wood technology
logistics and production
electronics and sensors
cyber-physical systems
computer science
artificial intelligence

KRDB research centre for Knowledge-based Artificial Intelligence

Part of the CS and AI institute

- Strong focus on **research excellence** (foundational and applied) and **collaborative projects**
- Wide **network of collaborations** on AI (technical, CS-oriented approach)
- Some **key facts**: Diego Calvanese PC Chair of IJCAI 2026, Chiara Ghidini vice-president of AIxIA, Marco Montali General Chair of AIxIA 2024
- **AI-lab** (funded by FESR-EFRE with 1M €): AI-based services for the territory

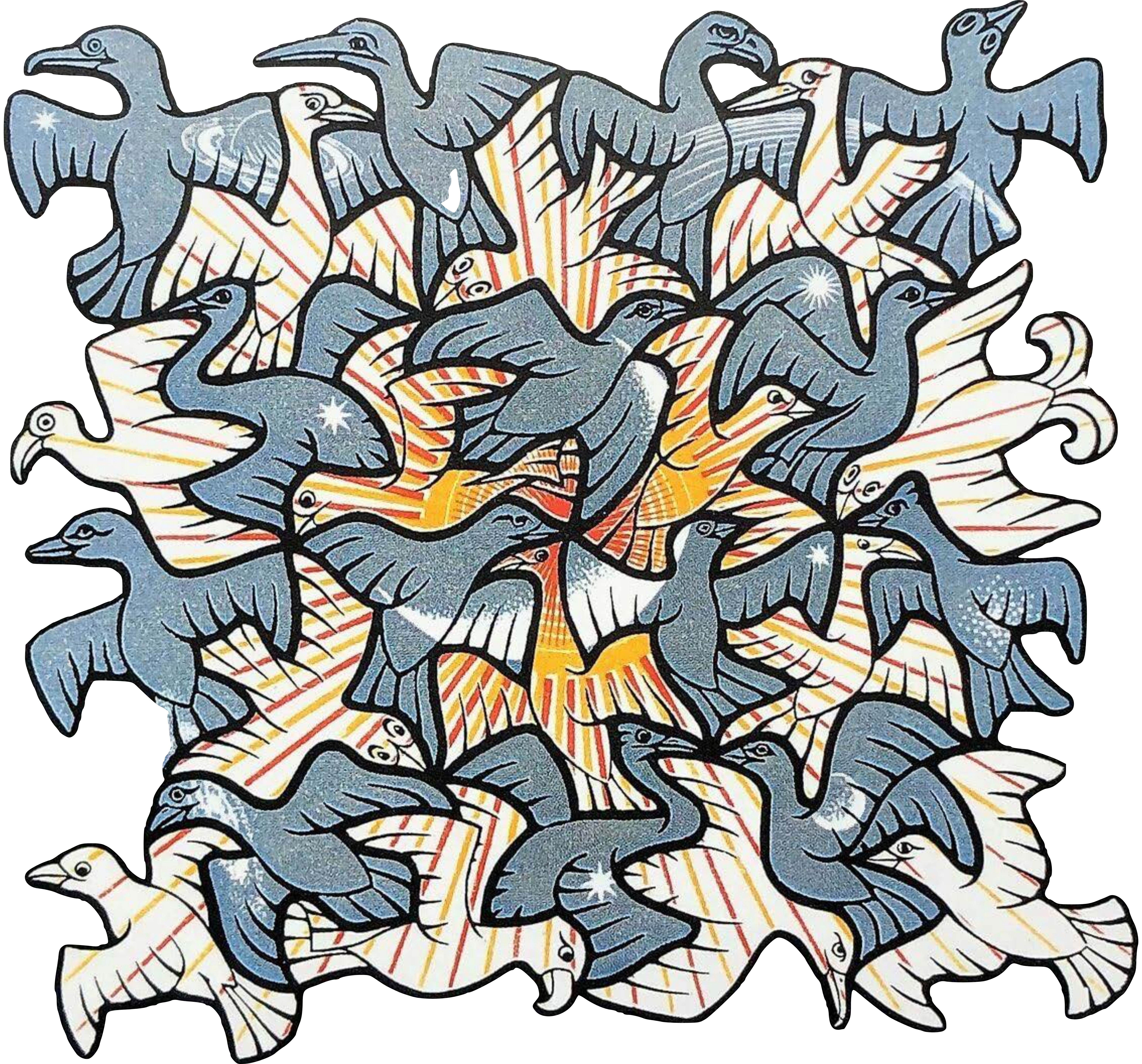
Trustworthiness as "technical robustness"

AI systems operating under "provable guarantees"

- Initial focus: **AI for data management**, **knowledge representation and automated reasoning**, **knowledge graphs**, **semantic technologies**
- Then expanded towards dynamic systems: **AI for process science (BPM/process mining)**, **temporal reasoning**, **automated planning and scheduling**
- Recently: increasing focus on **nesy integration**, **explainability**, **predictive/prescriptive techniques with guarantees**, **integration of learning, simulation, and reasoning**

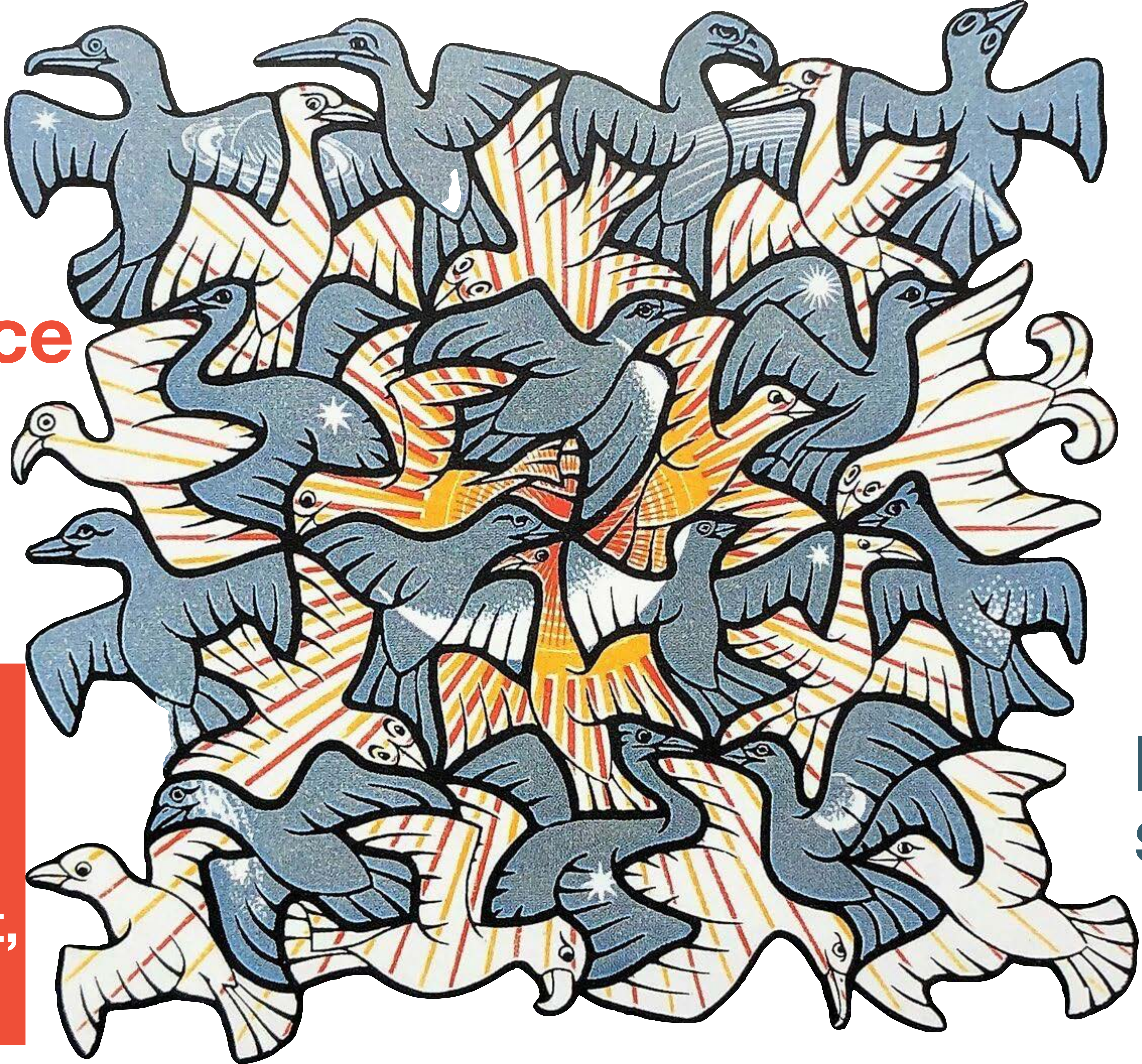
Public outreach

50+ presentations to the public, schools, industries, politics, …



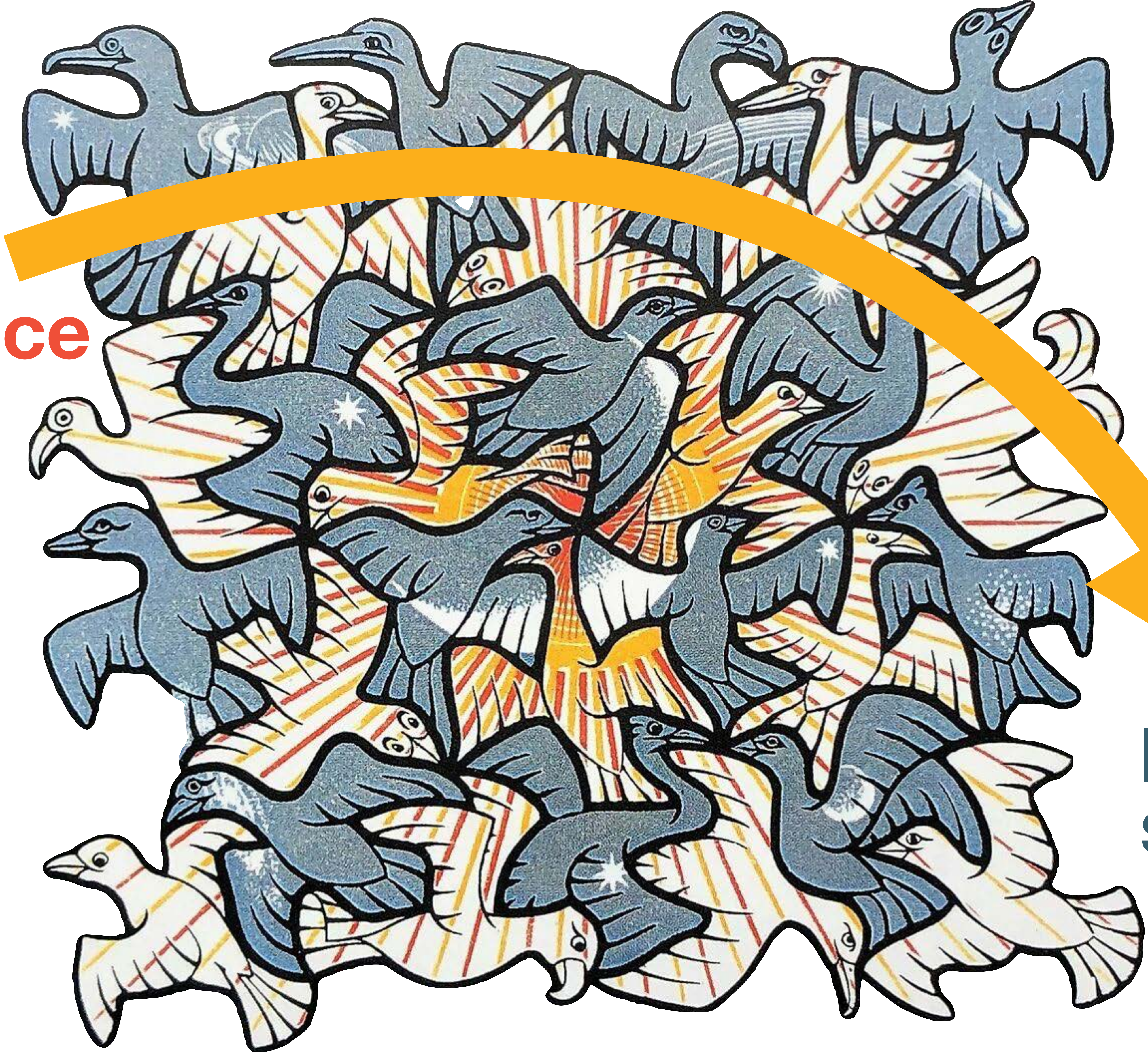
artificial intelligence

In all its flavours: symbolic, connectionist, nesy



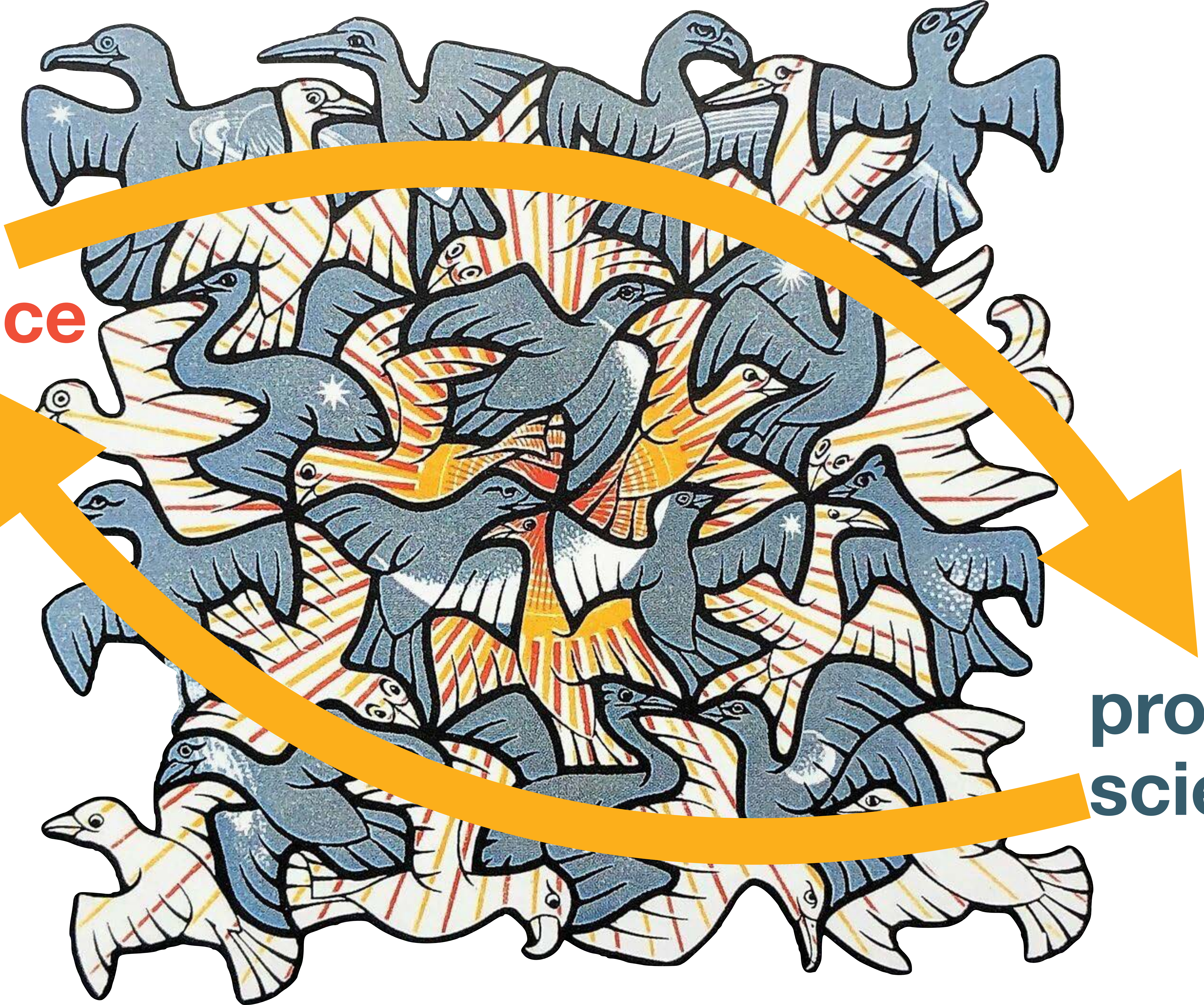
process science

artificial intelligence



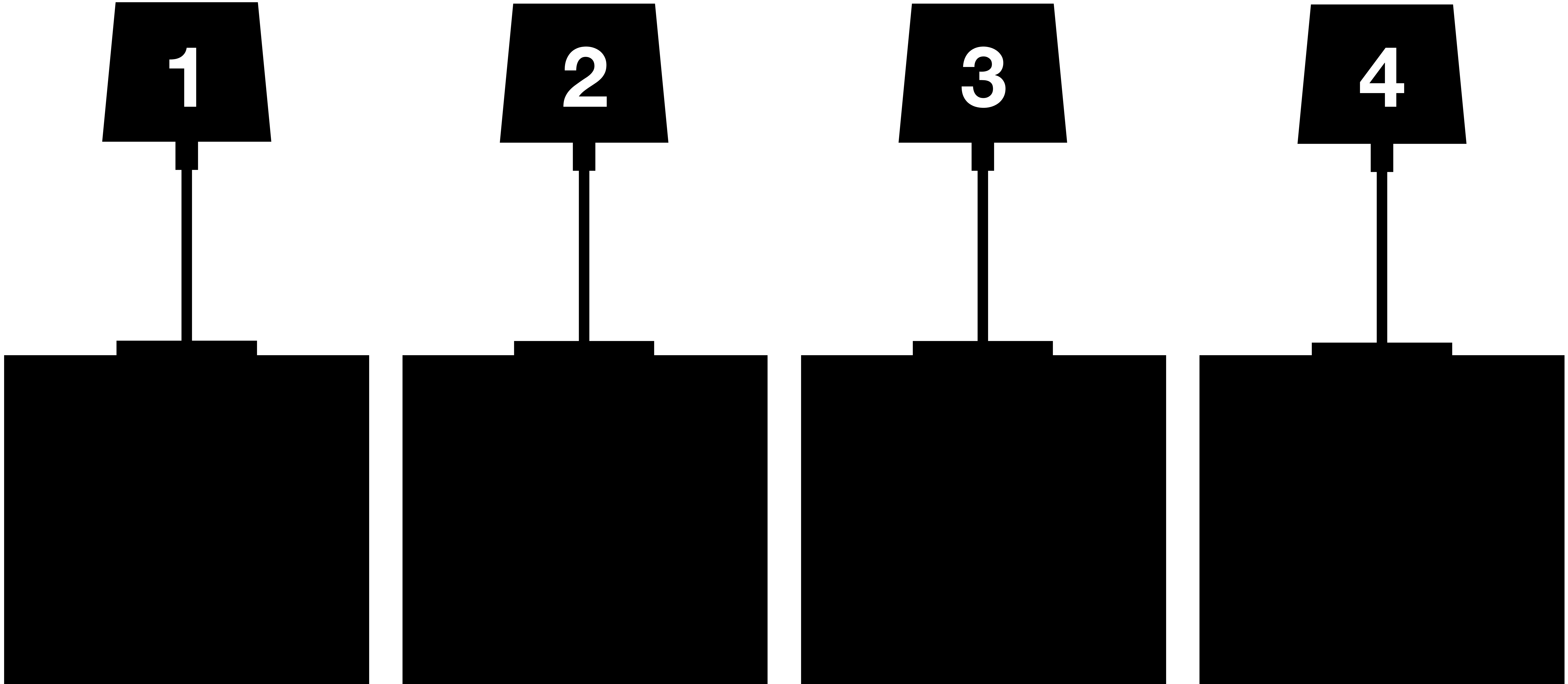
process science

artificial intelligence



process science

4 acts



4 acts

1

Process science

2

Declarative processes

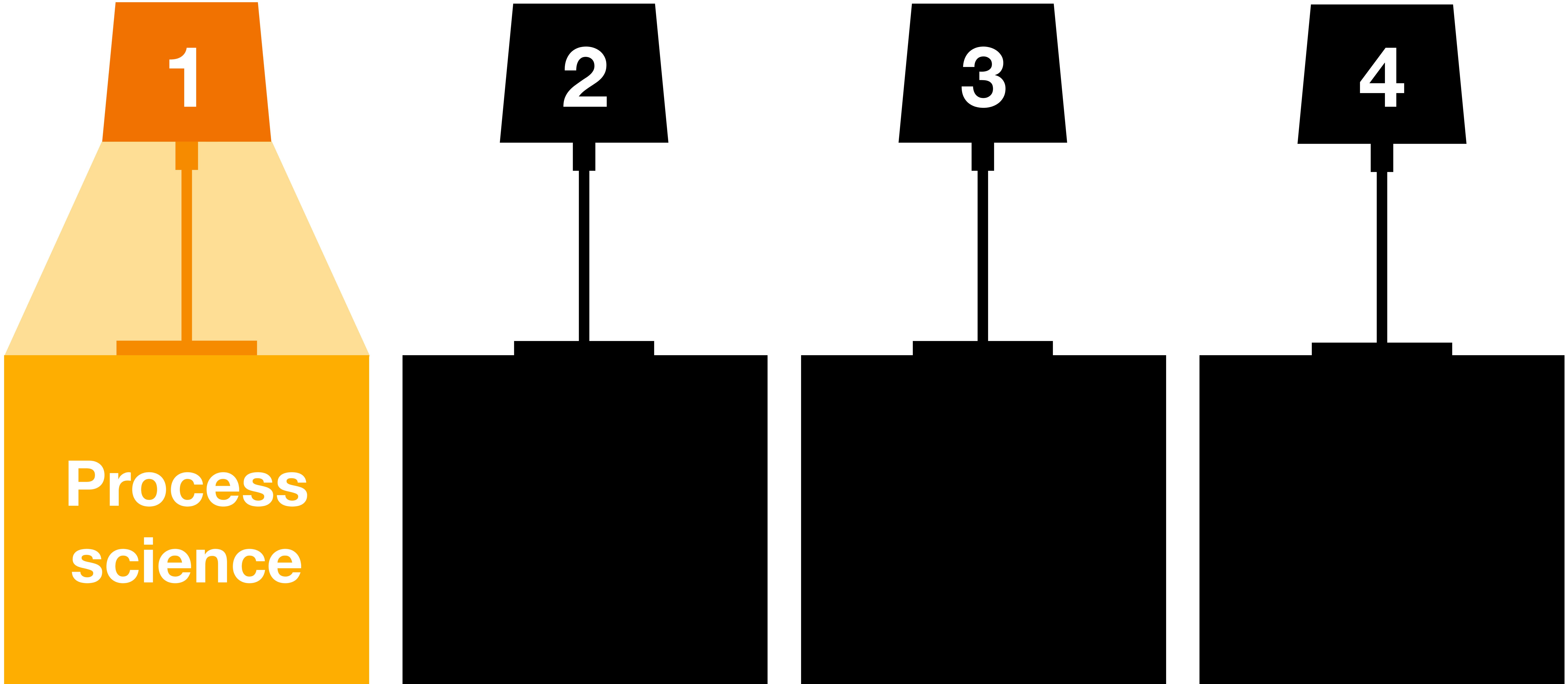
3

AI for declarative processes

4

Ongoing research

1/4

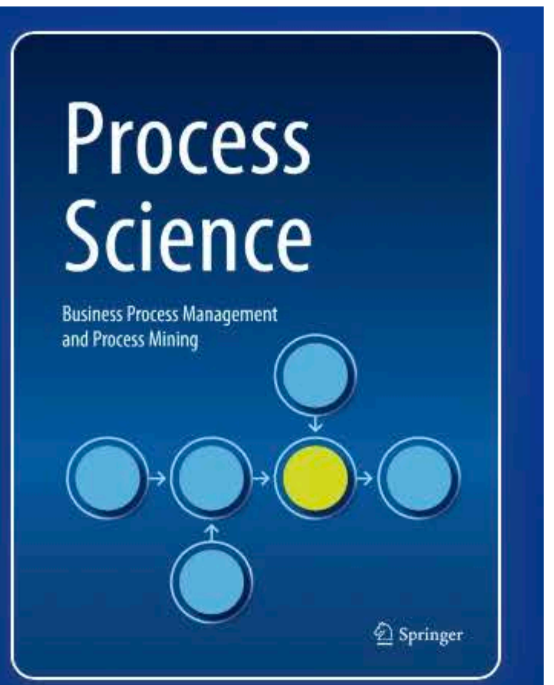


“All happens through processes, and all happens in processes”

Process science: the interdisciplinary study of socio-technical change

Editorial | Open access | Published: 25 July 2024

Volume 1, article number 1, (2024) Cite this article



Jan vom Brocke, Wil M. P. van der Aalst, Nicholas Berente, Boudewijn van Dongen, Thomas Grisold, Waldemar Kremser, Jan Mendling, Brian T. Pentland, Maximilian Roeglinger, Michael Rosemann & Barbara Weber

Abstract

Process science is the interdisciplinary study of socio-technical processes. Socio-technical processes involve coherent series of changes over time, entailing actions and events that include humans and digital technologies. The ubiquitous availability of digital trace data, combined with advanced data analytics capabilities, offer new and unprecedented opportunities to study such processes through multiple data sources. Process science is concerned with describing, explaining, and intervening in socio-technical

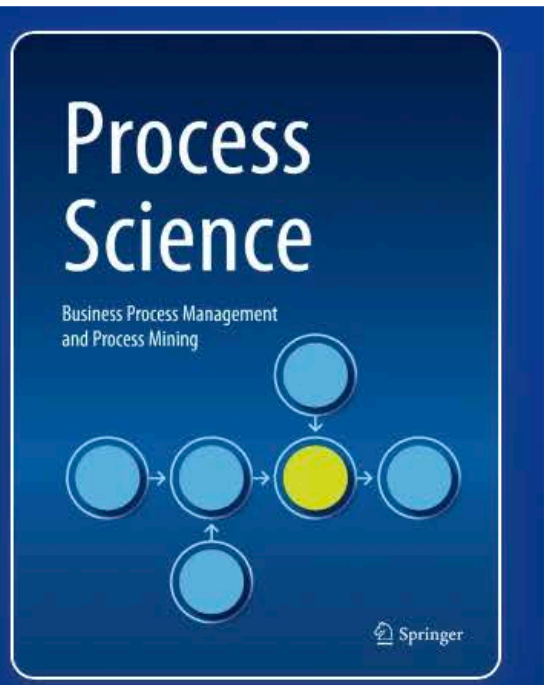
“All happens through processes, and all happens in processes”

Home > Process Science > Article

Process science: the interdisciplinary study of socio-technical change

Editorial | Open access | Published: 25 July 2024

Volume 1, article number 1, (2024) Cite this article



Process science is the interdisciplinary study of socio-technical processes. Socio-technical processes involve coherent series of changes over time, entailing actions and events that include humans and digital technologies. The ubiquitous availability of digital trace data, combined with advanced data analytics capabilities, offer new and unprecedented opportunities to study such processes through multiple data sources.

Abstract

Process science is the interdisciplinary study of socio-technical processes. Socio-technical processes involve coherent series of changes over time, entailing actions and events that include humans and digital technologies. The ubiquitous availability of digital trace data, combined with advanced data analytics capabilities, offer new and unprecedented opportunities to study such processes through multiple data sources. Process science is concerned with describing, explaining, and intervening in socio-technical

“All happens through processes, and all happens in processes”

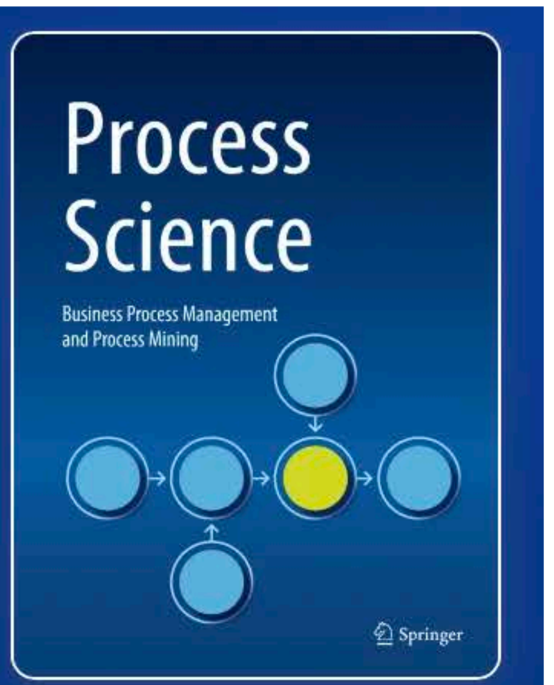
Business process management

Home > Process Science > Article

Process science: the interdisciplinary study of socio-technical change

Editorial | Open access | Published: 25 July 2024

Volume 1, article number 1, (2024) Cite this article



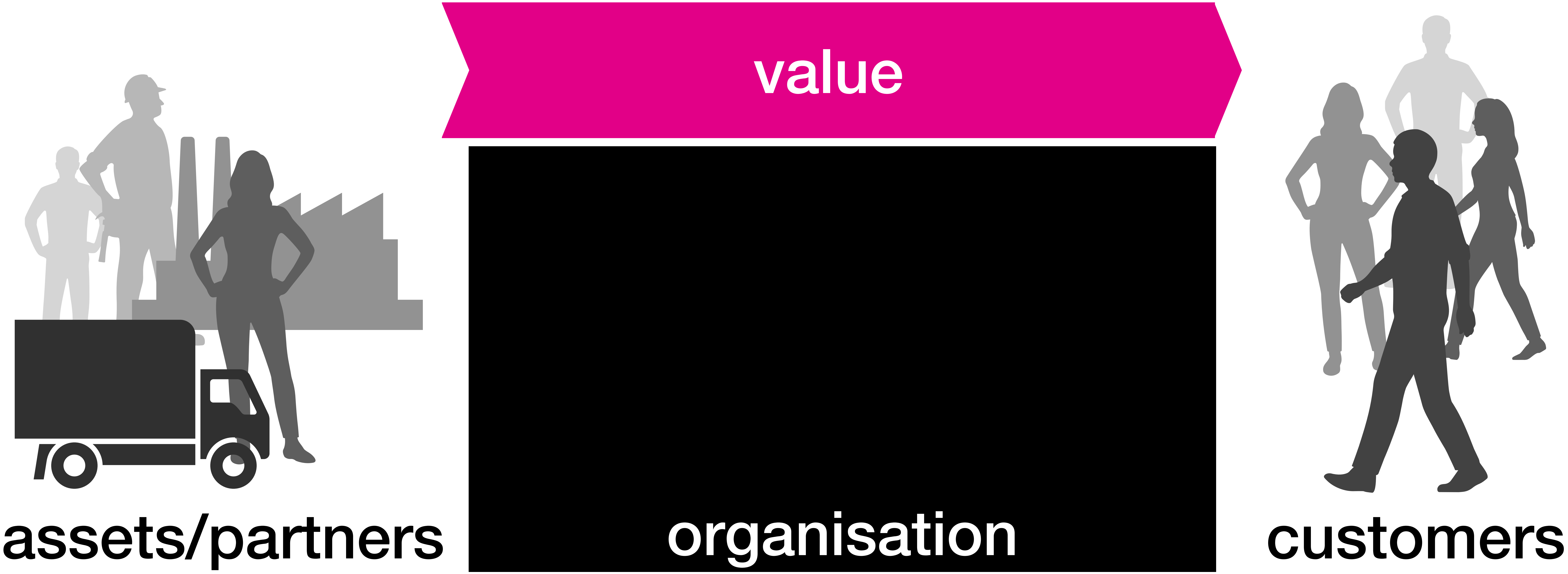
Process science is the interdisciplinary study of socio-technical processes. Socio-technical processes involve coherent series of changes over time, entailing actions and events that include humans and digital technologies. The ubiquitous availability of digital trace data, combined with advanced data analytics capabilities, offer new and unprecedented opportunities to study such processes through multiple data sources.

Process mining

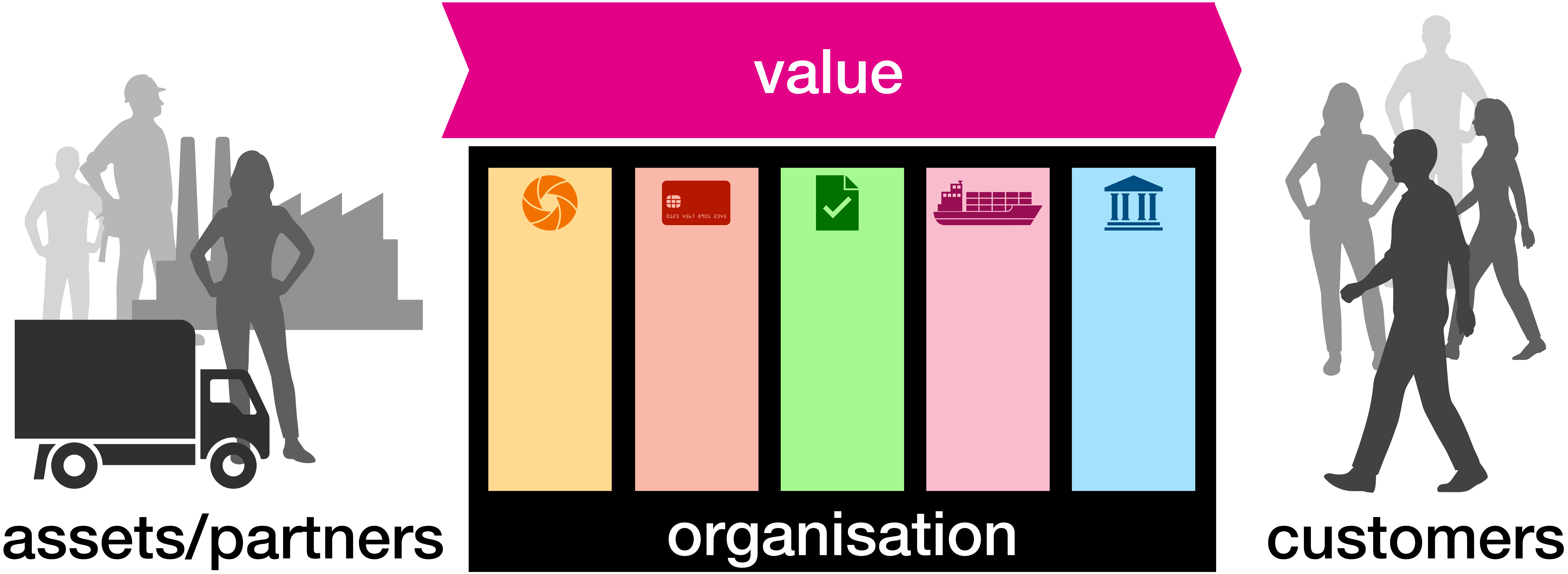
Abstract

Process science is the interdisciplinary study of socio-technical processes. Socio-technical processes involve coherent series of changes over time, entailing actions and events that include humans and digital technologies. The ubiquitous availability of digital trace data, combined with advanced data analytics capabilities, offer new and unprecedented opportunities to study such processes through multiple data sources. Process science is concerned with describing, explaining, and intervening in socio-technical

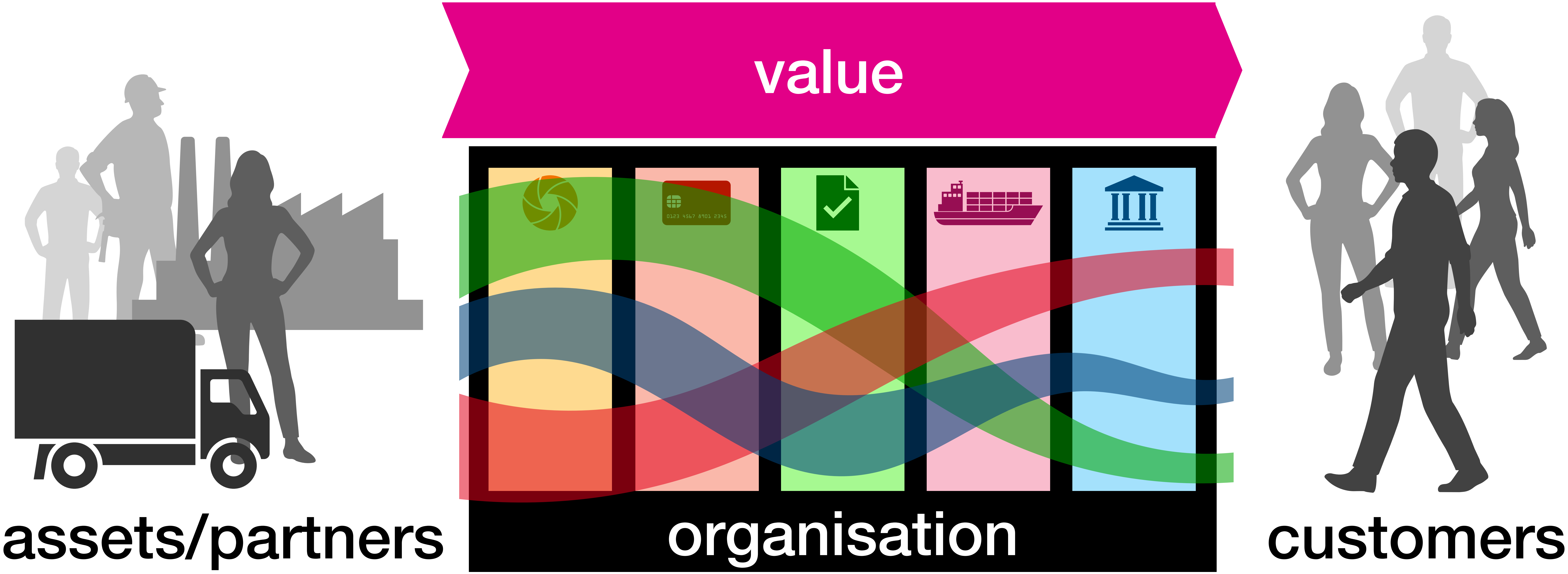
Business process



Business process



Business process



A **business process** is a collection of **related events, activities** and **decisions**, that involve a number of ***actors and objects***, and that *collectively lead to an outcome that is of* ***value to an organization or its customers***


REALITY
process discovery
(re) design
configure/ deploy
enact/ monitor
data
models
IT support
knowledge workers
managers/ analysts
ORGANISATION

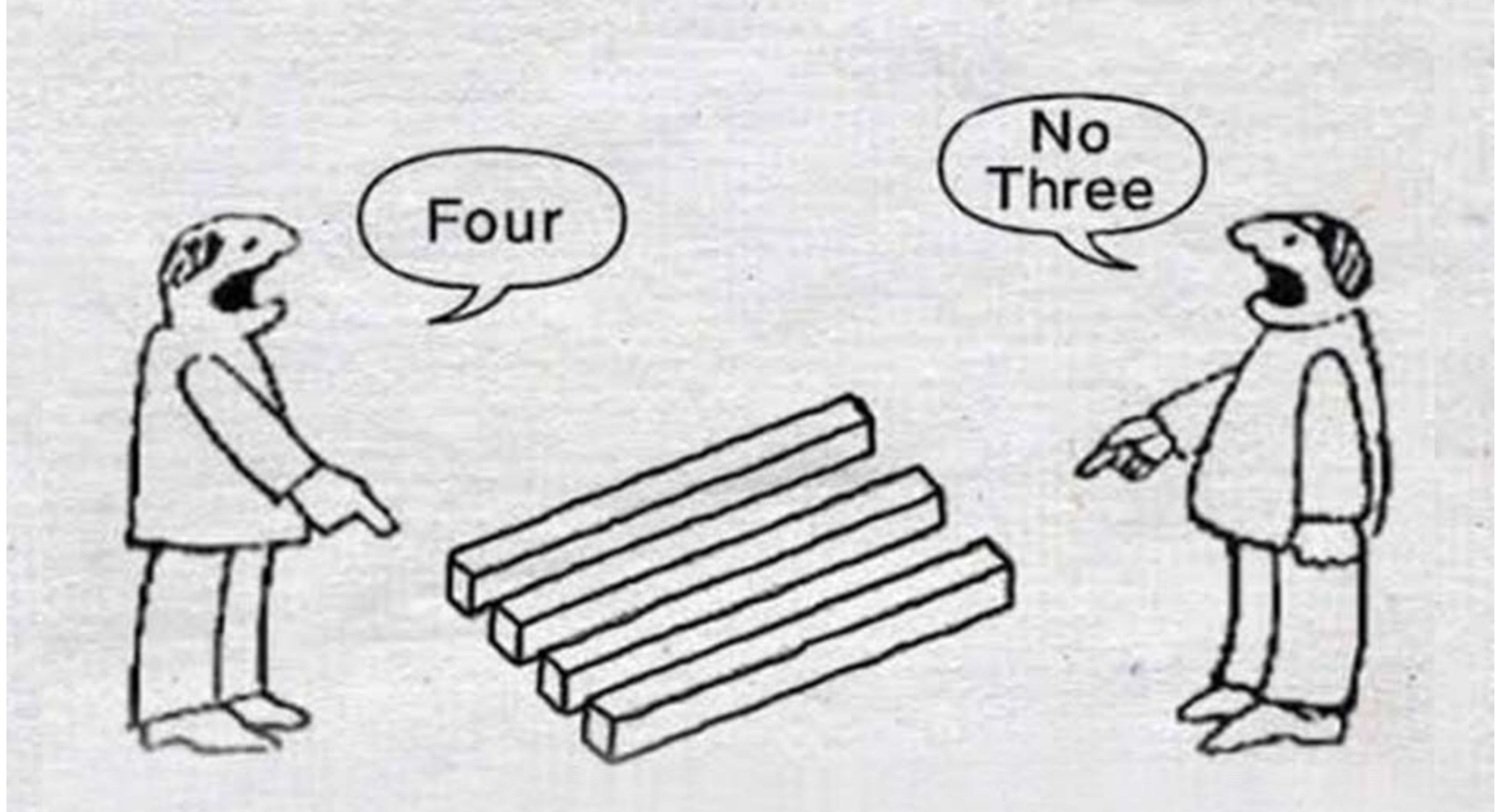
Case id	Event id	Timestamp	Activity	Resource	Cost	...
		Properties				
1	35654423	30-12-2010:11.02	register request	Pete	50	...
	35654424	31-12-2010:10.06	examine thoroughly	Sue	400	...
	35654425	05-01-2011:15.12	check ticket	Mike	100	...
	35654426	06-01-2011:11.18	decide	Sara	200	...
	35654427	07-01-2011:14.24	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	

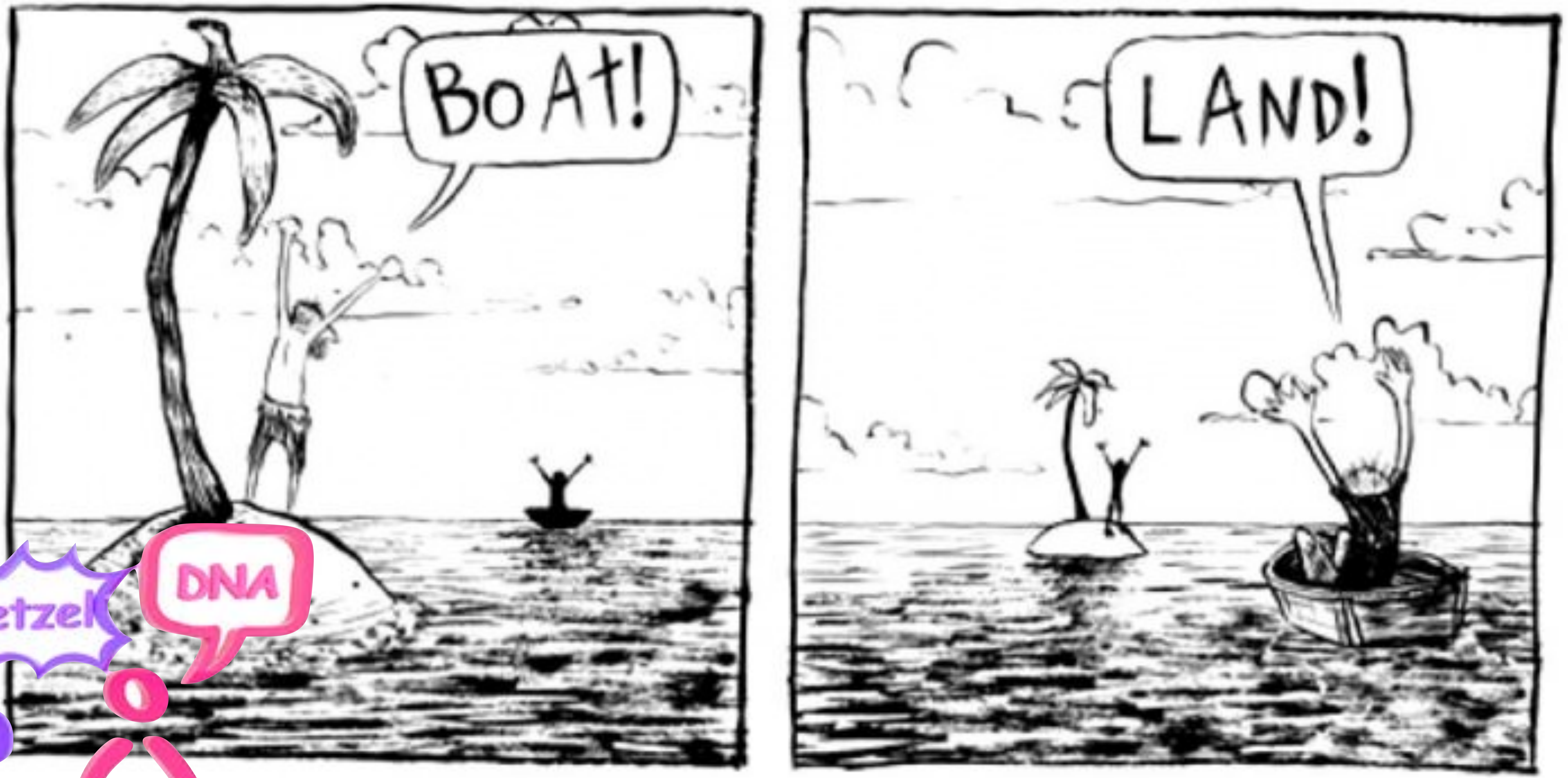

REALITY
enact/
monito
knowledge
workers
process models
event logs
(re)
design
managers/
analysts
ORGANISATION


REALITY
model-driven
process management
process models
event logs
enact/
monito
(re)
design
knowledge
workers
managers/
analysts
ORG
ATION

Model-driven process management

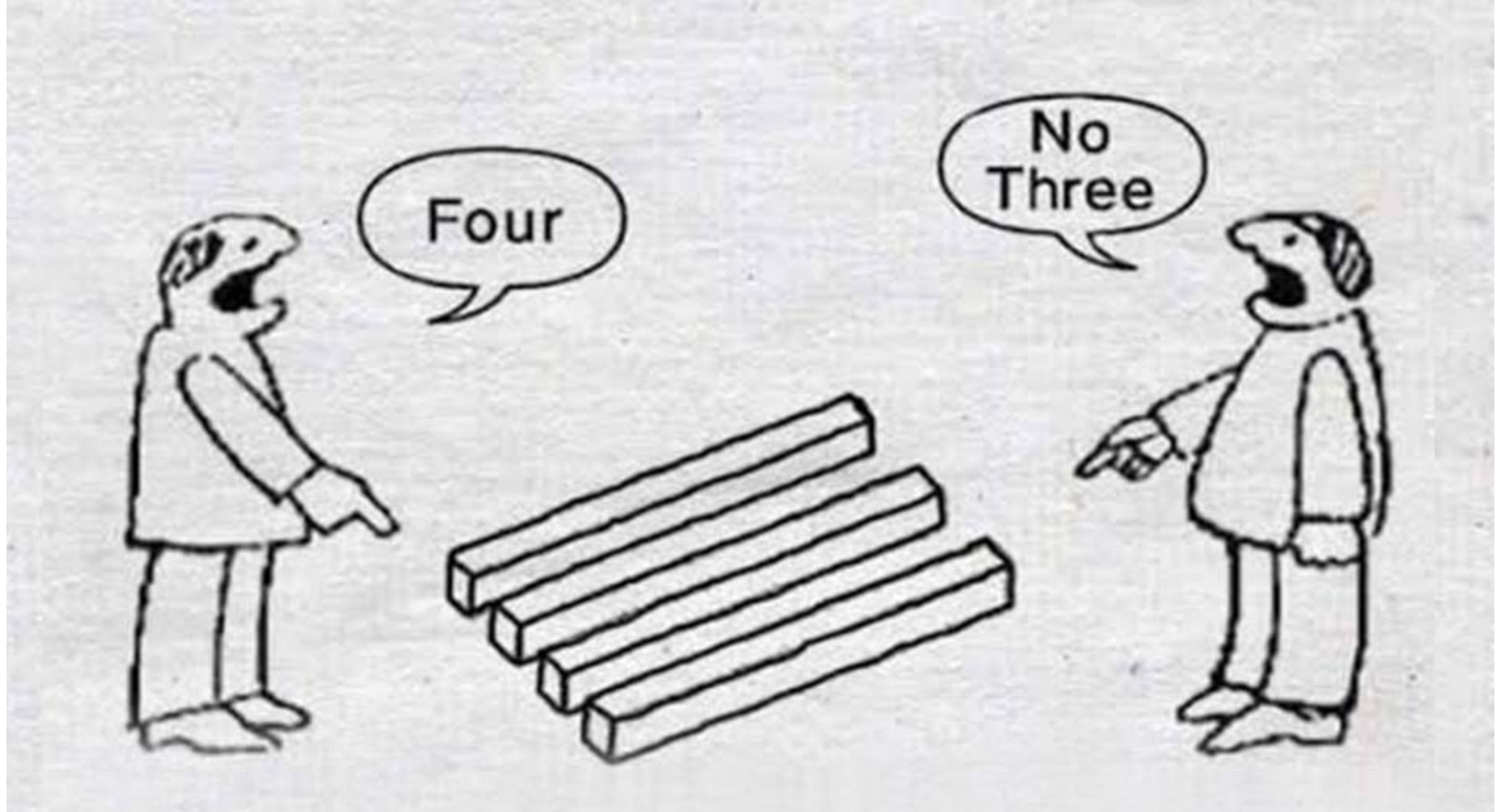
1. process discovery via conceptual modelling

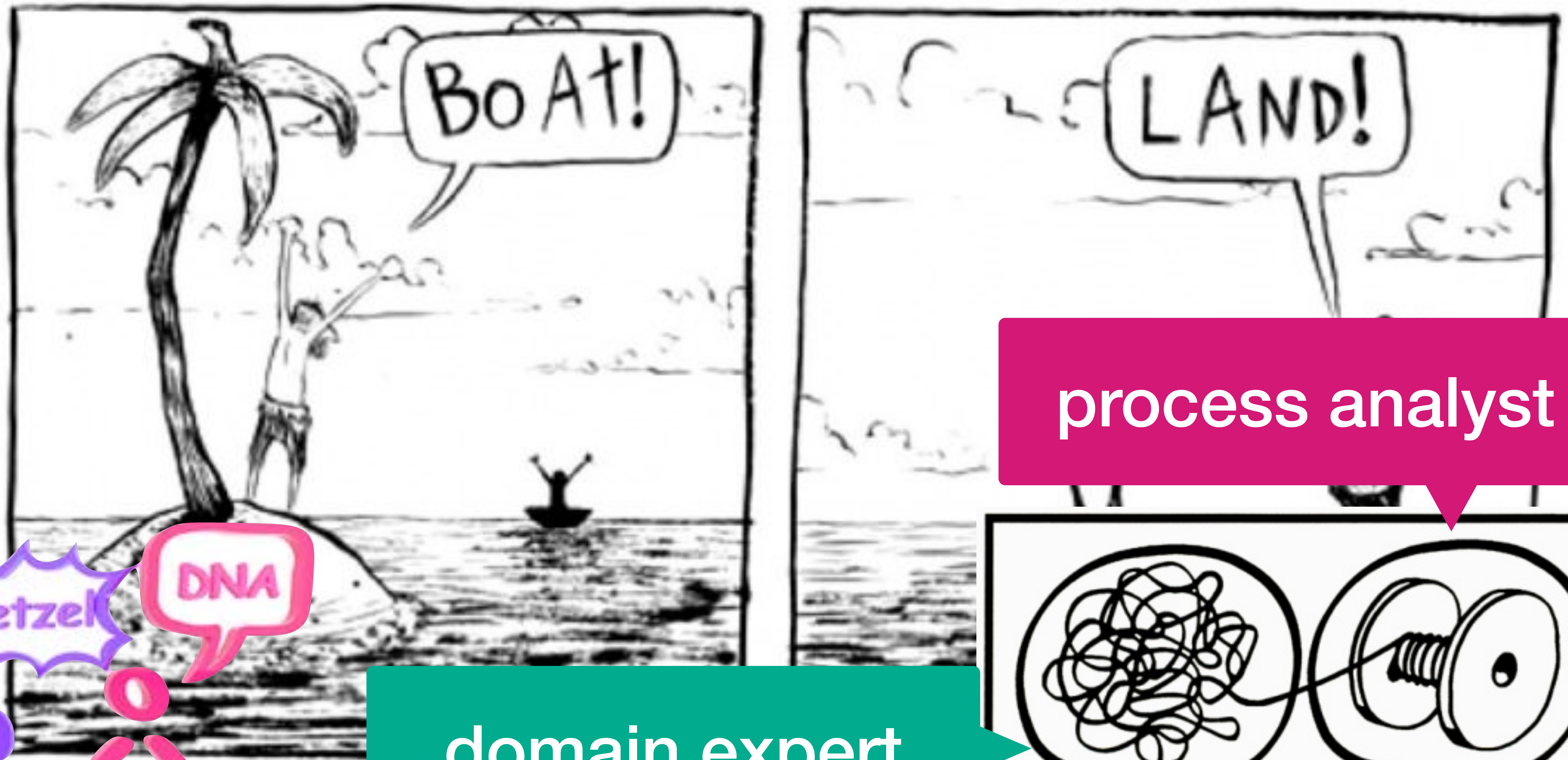


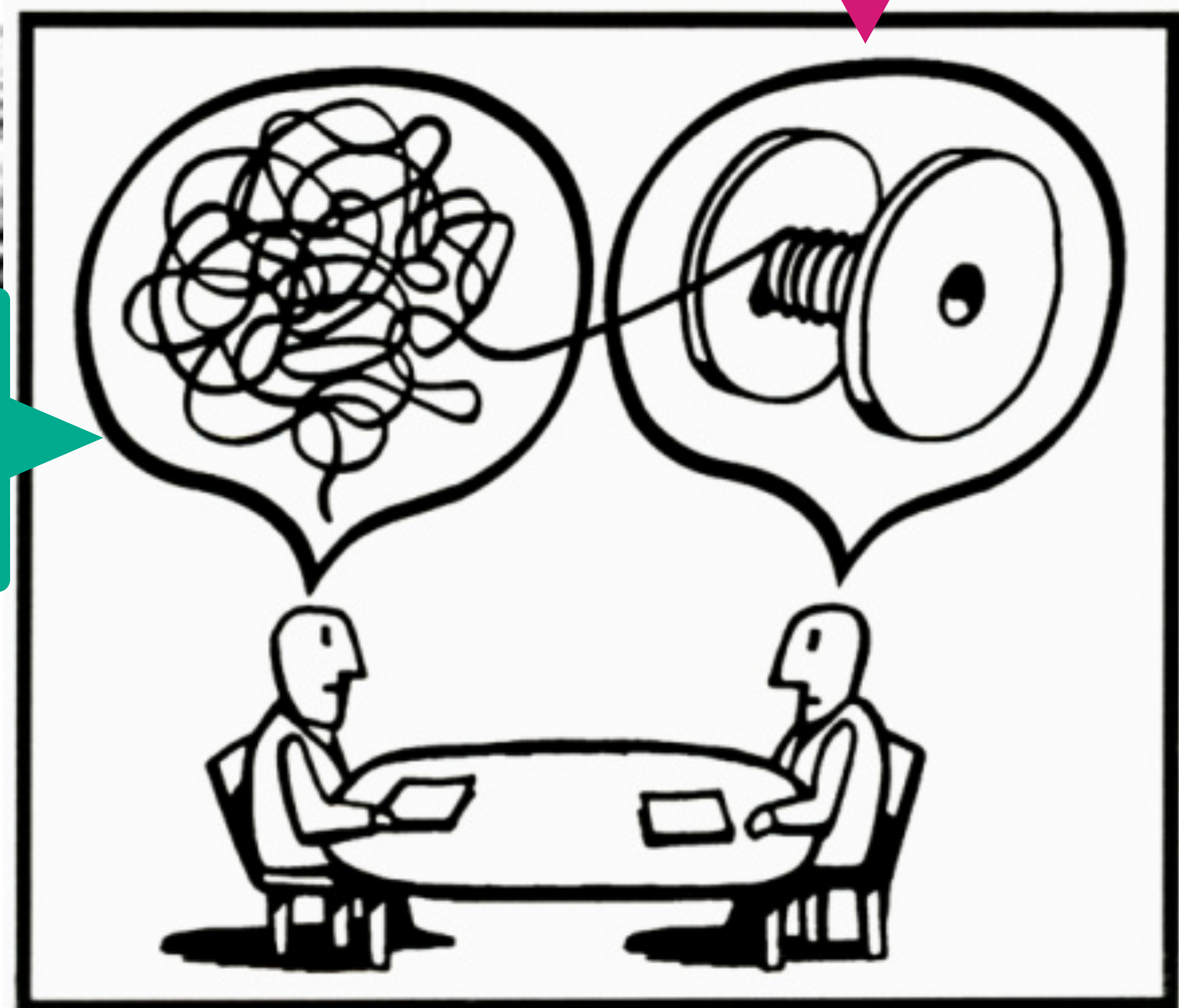


Model-driven process management

1. process discovery via conceptual modelling

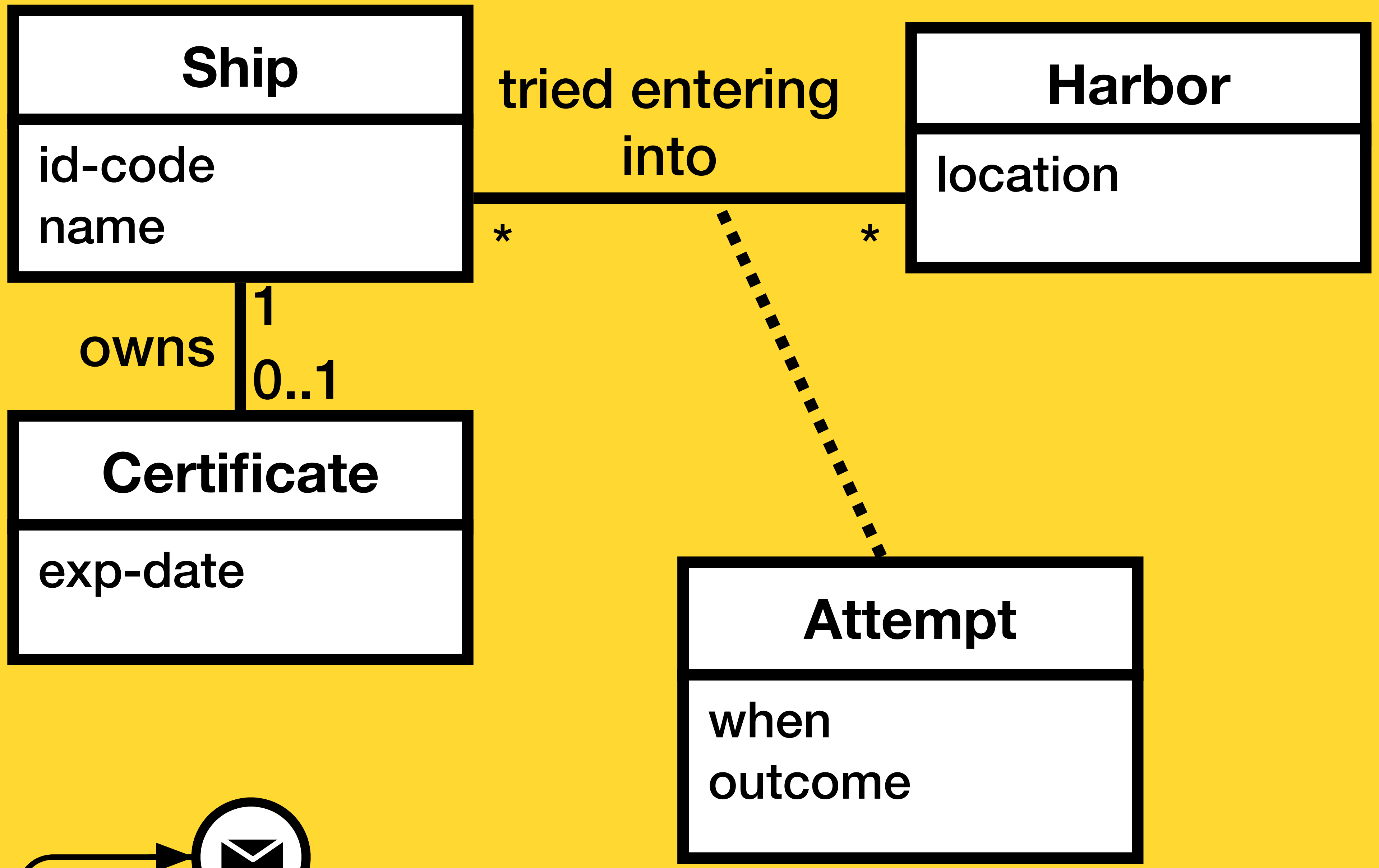


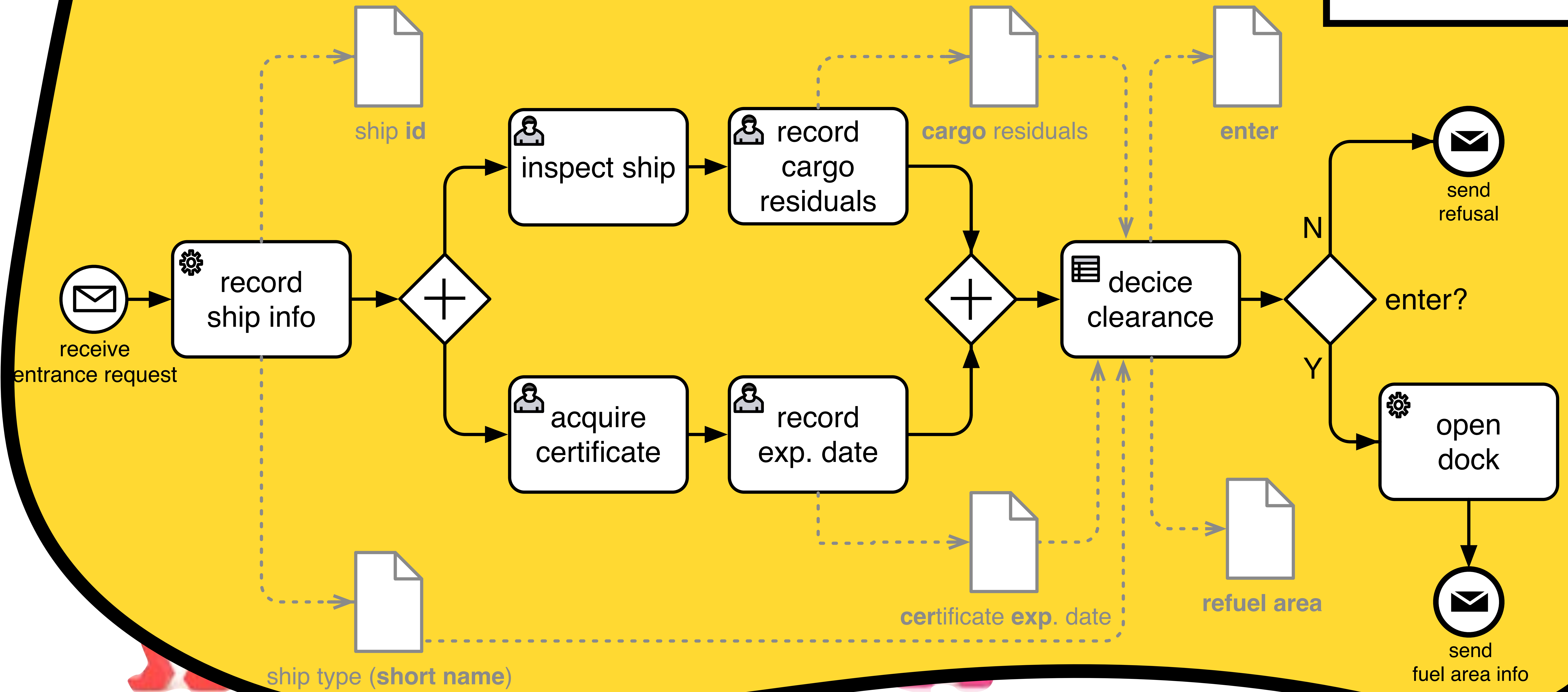


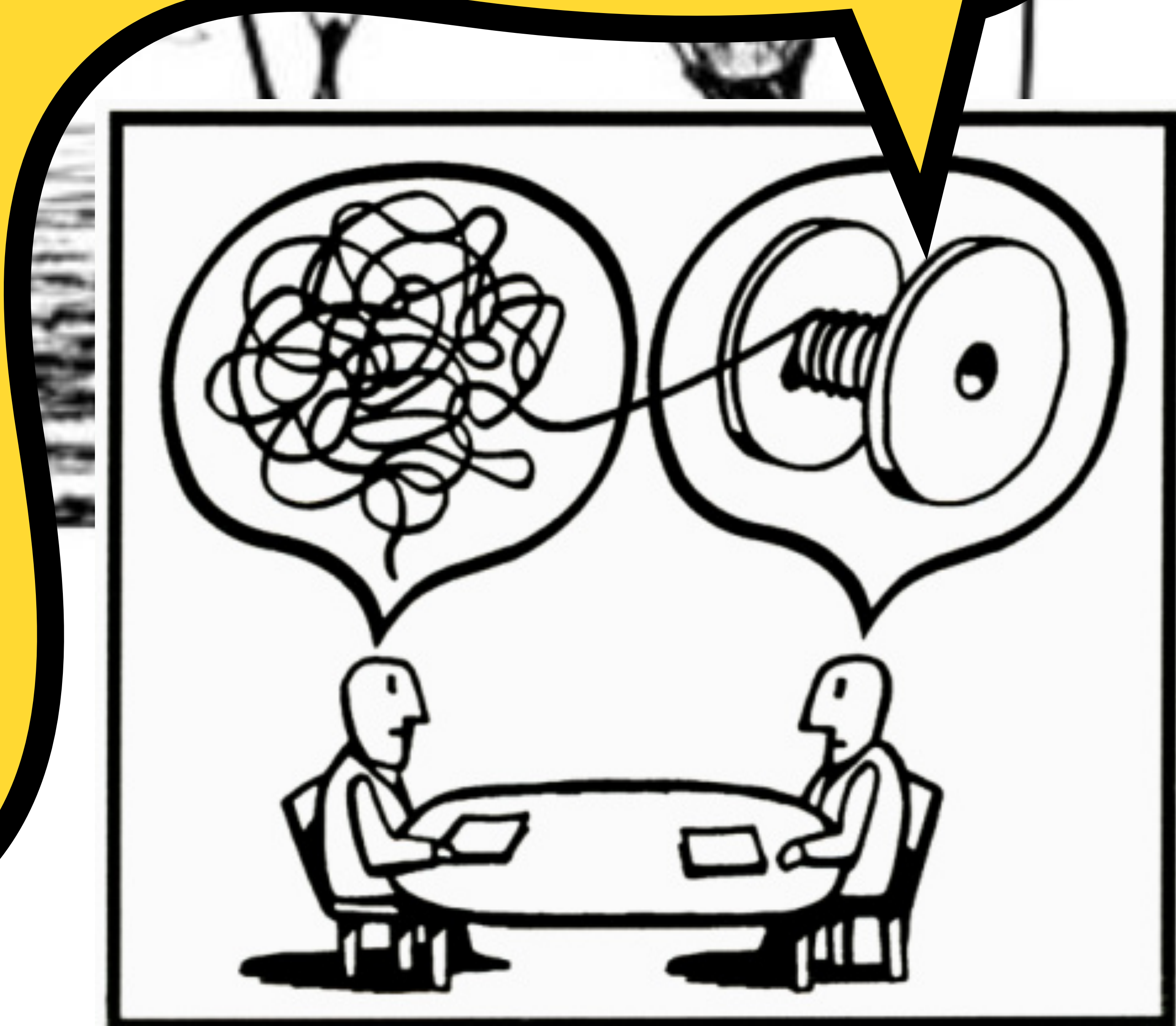


Refuel area determination

U	Enter	Length (m)	Cargo (mg/cm^2)	Refuel Area
	y,n	≥ 0	≥ 0	none, indoor, outdoor
1	n	–	–	none
2	y	≤ 350	–	indoor
3	y	> 350	≤ 0.3	indoor
4	y	> 350	> 0.3	outdoor

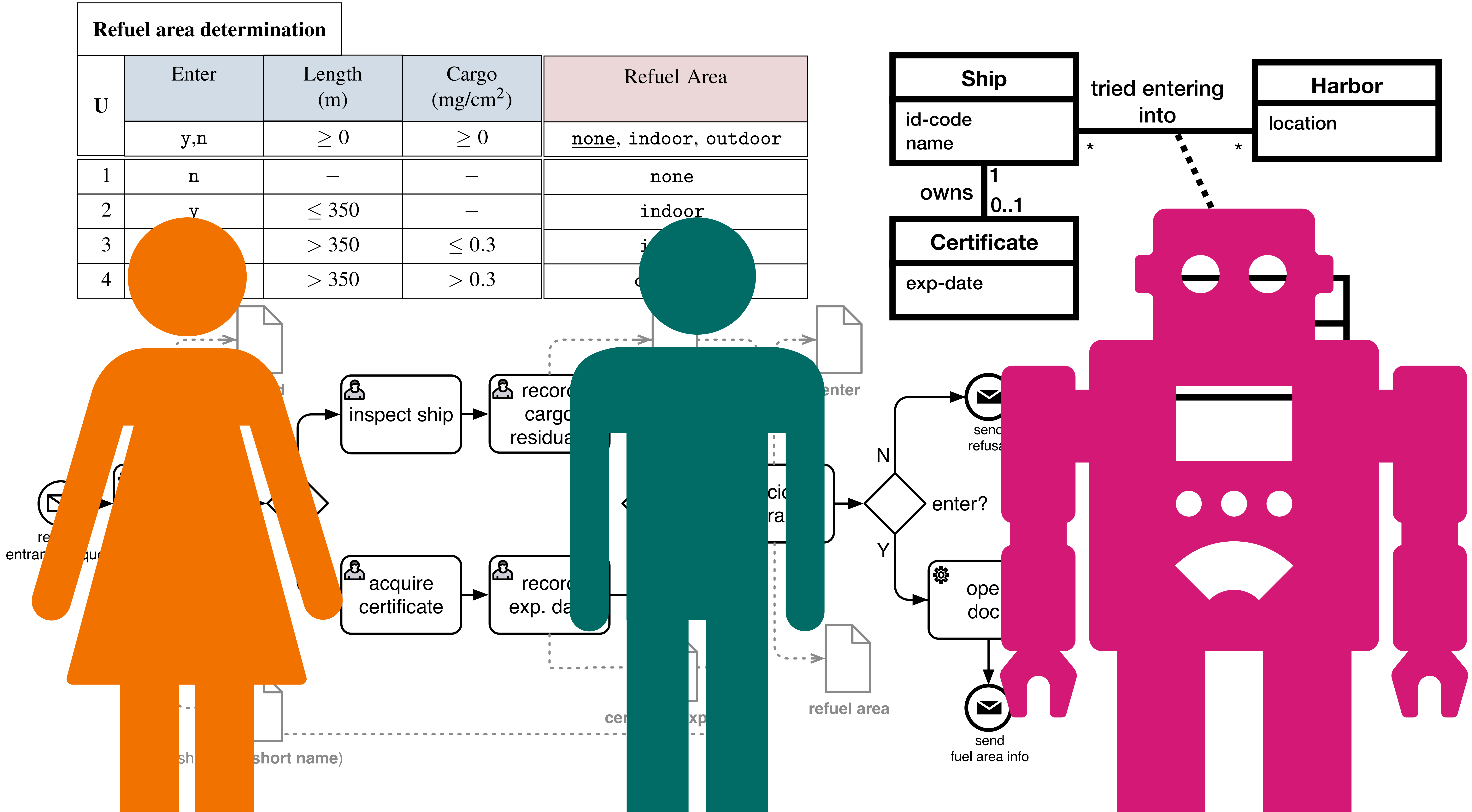






Model-driven process management

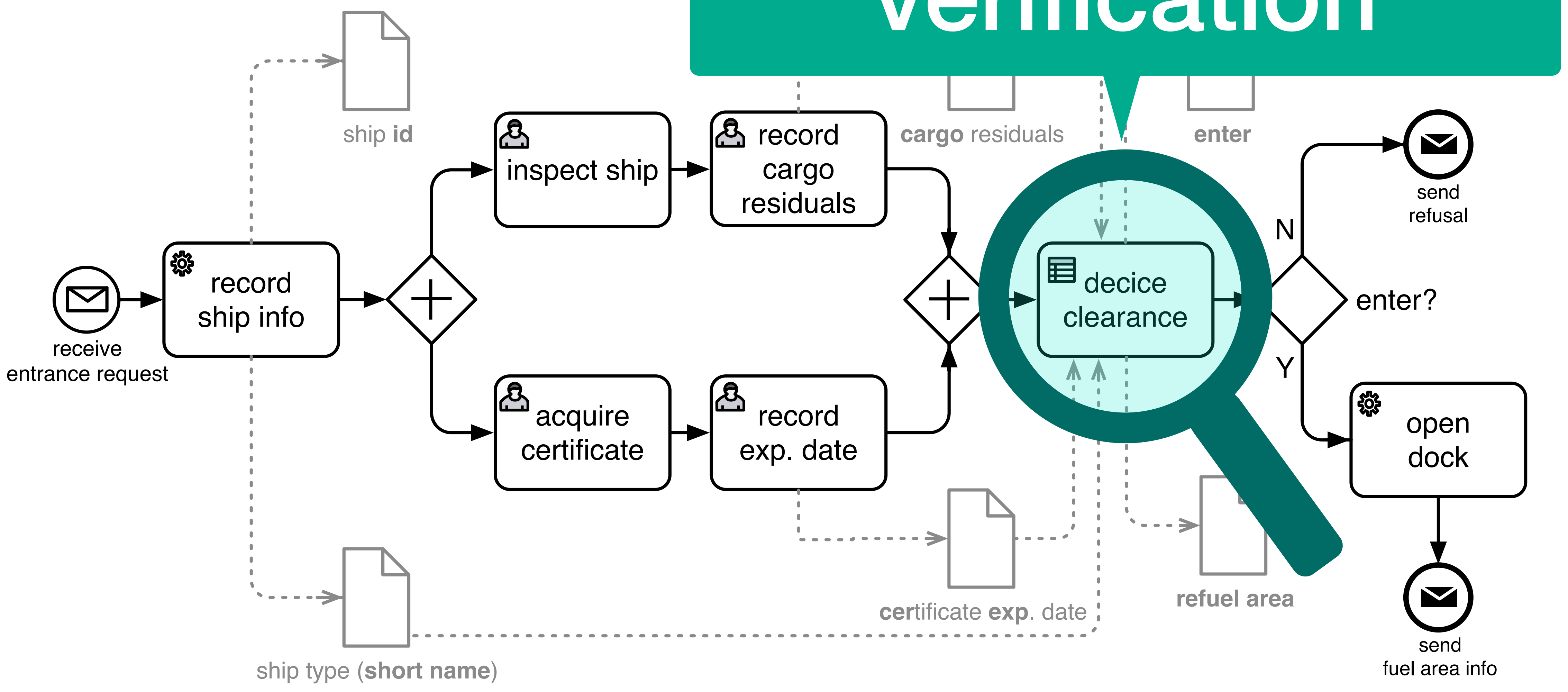
2. share and understand



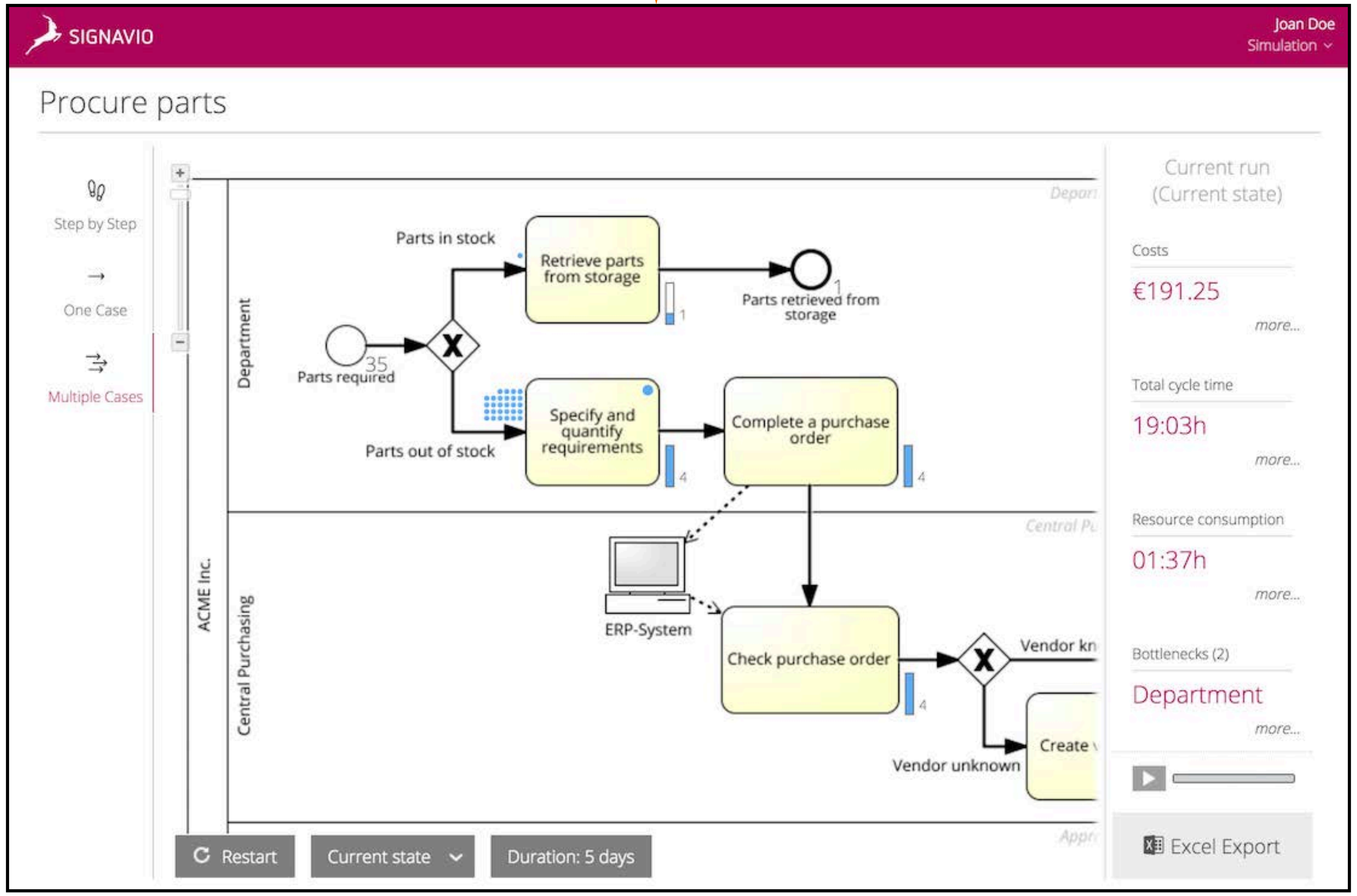
Model-driven process management

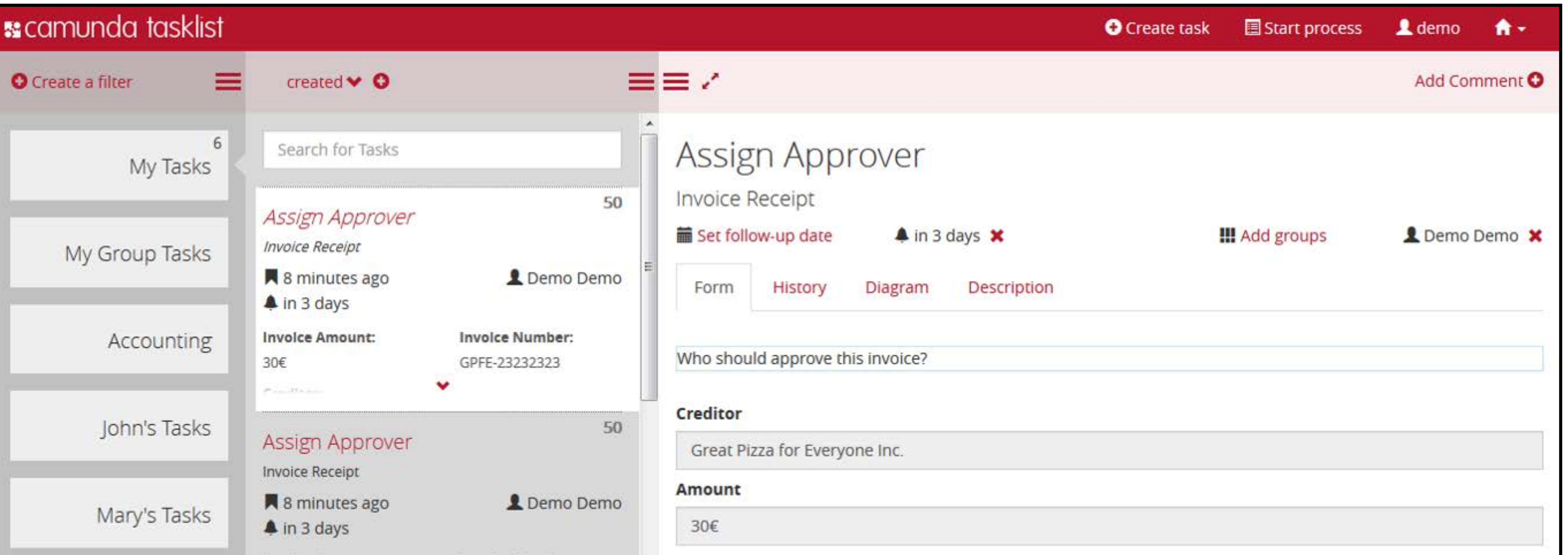
3. use models

verification



quantitative analysis/simulation



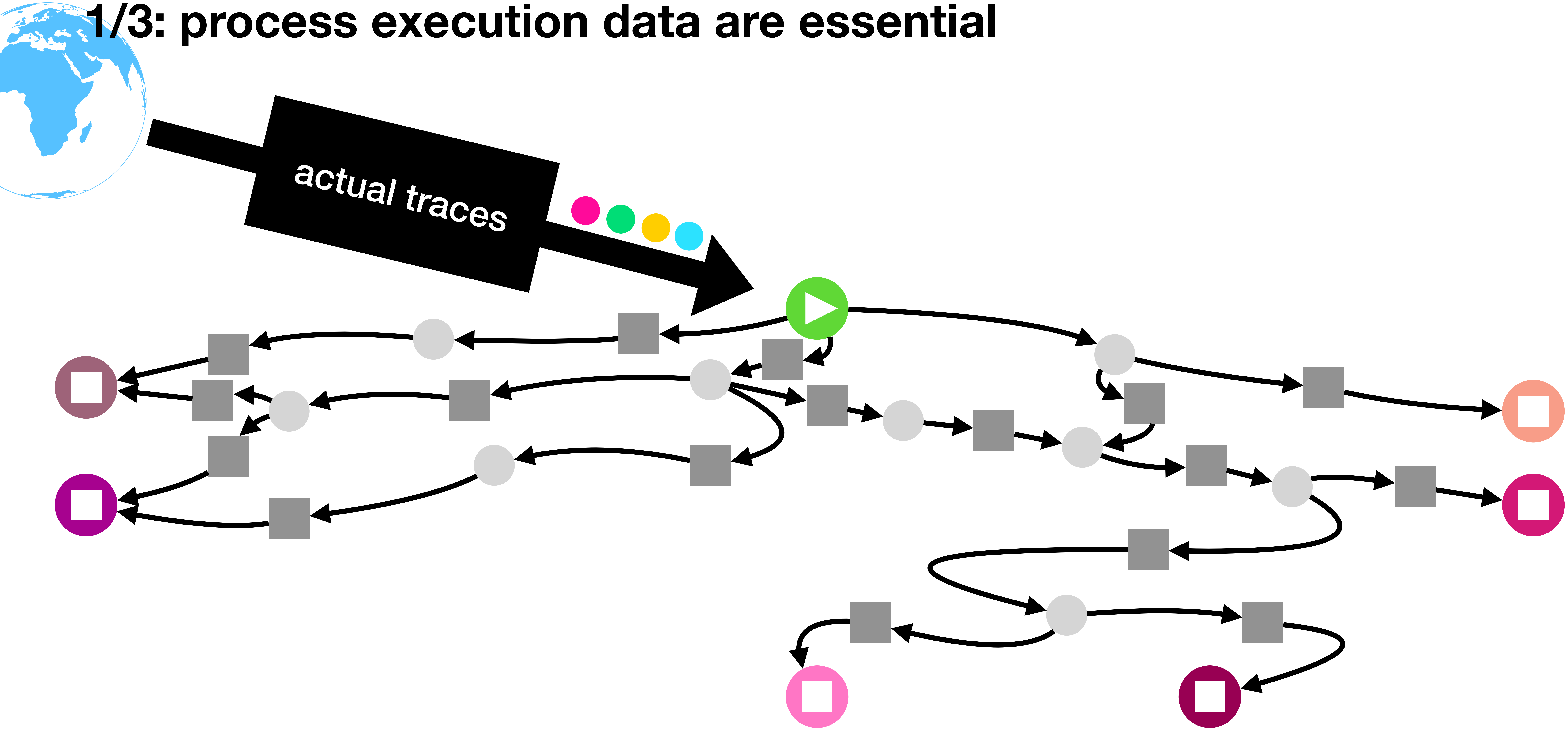


execution support

Are we really helping?

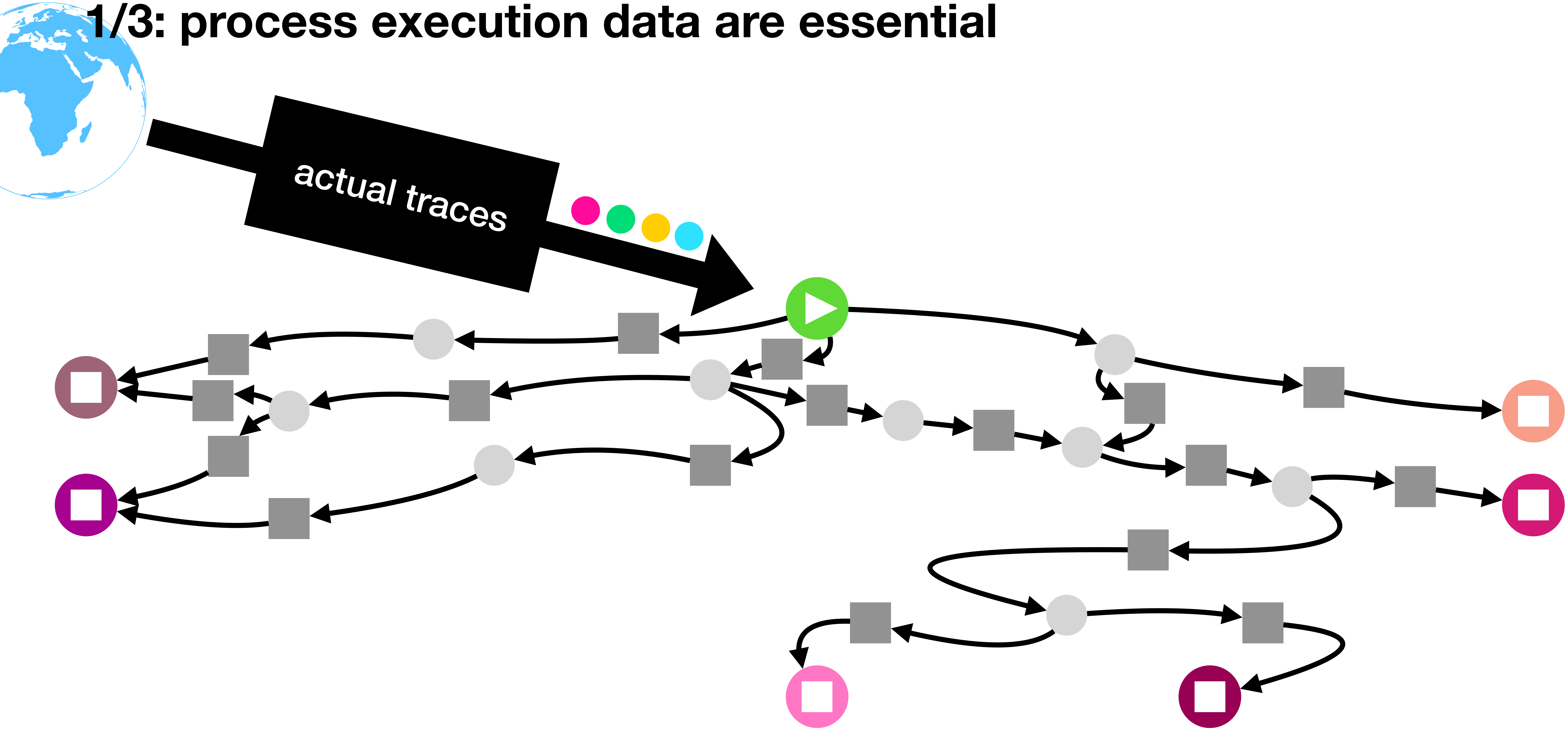
Difference between models and reality

1/3: process execution data are essential



Difference between models and reality

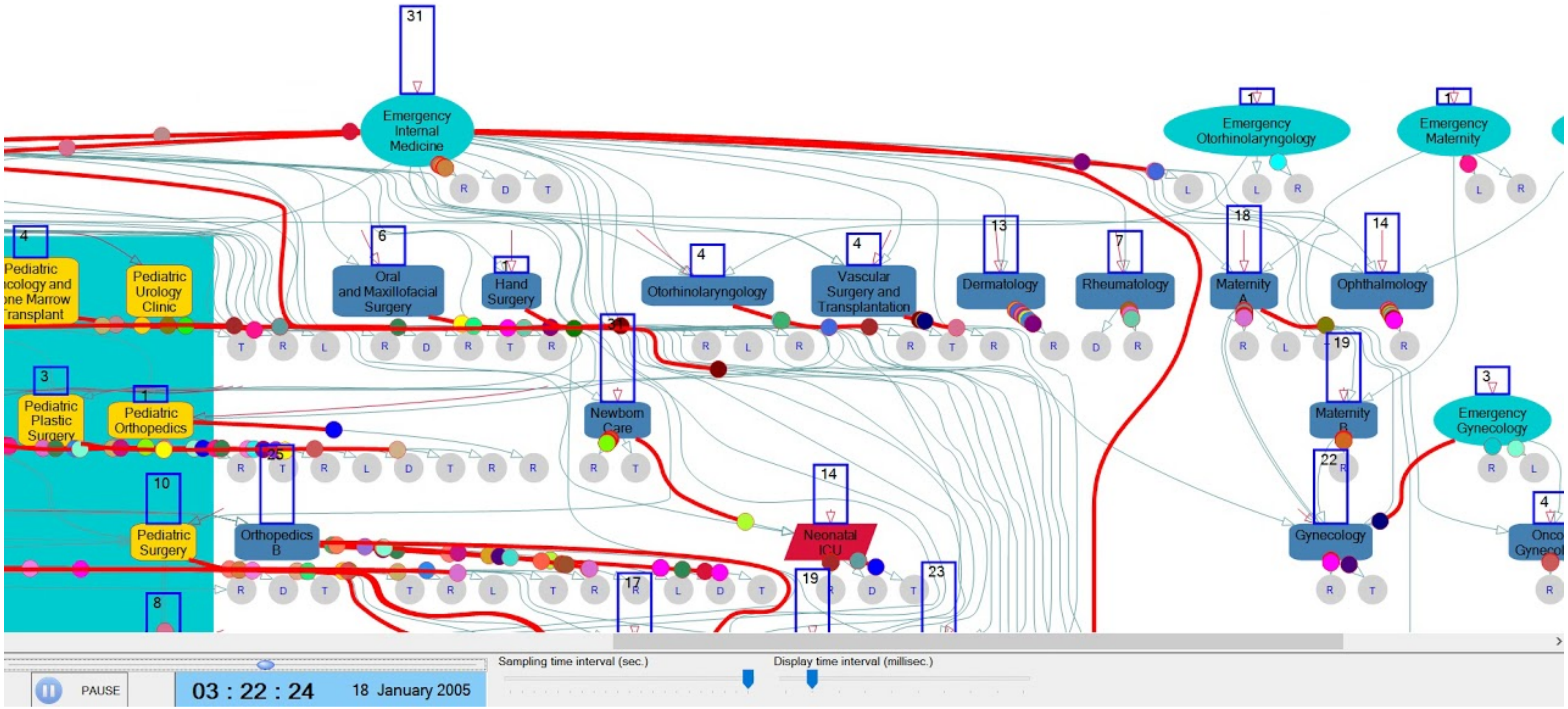
1/3: process execution data are essential



Real example…

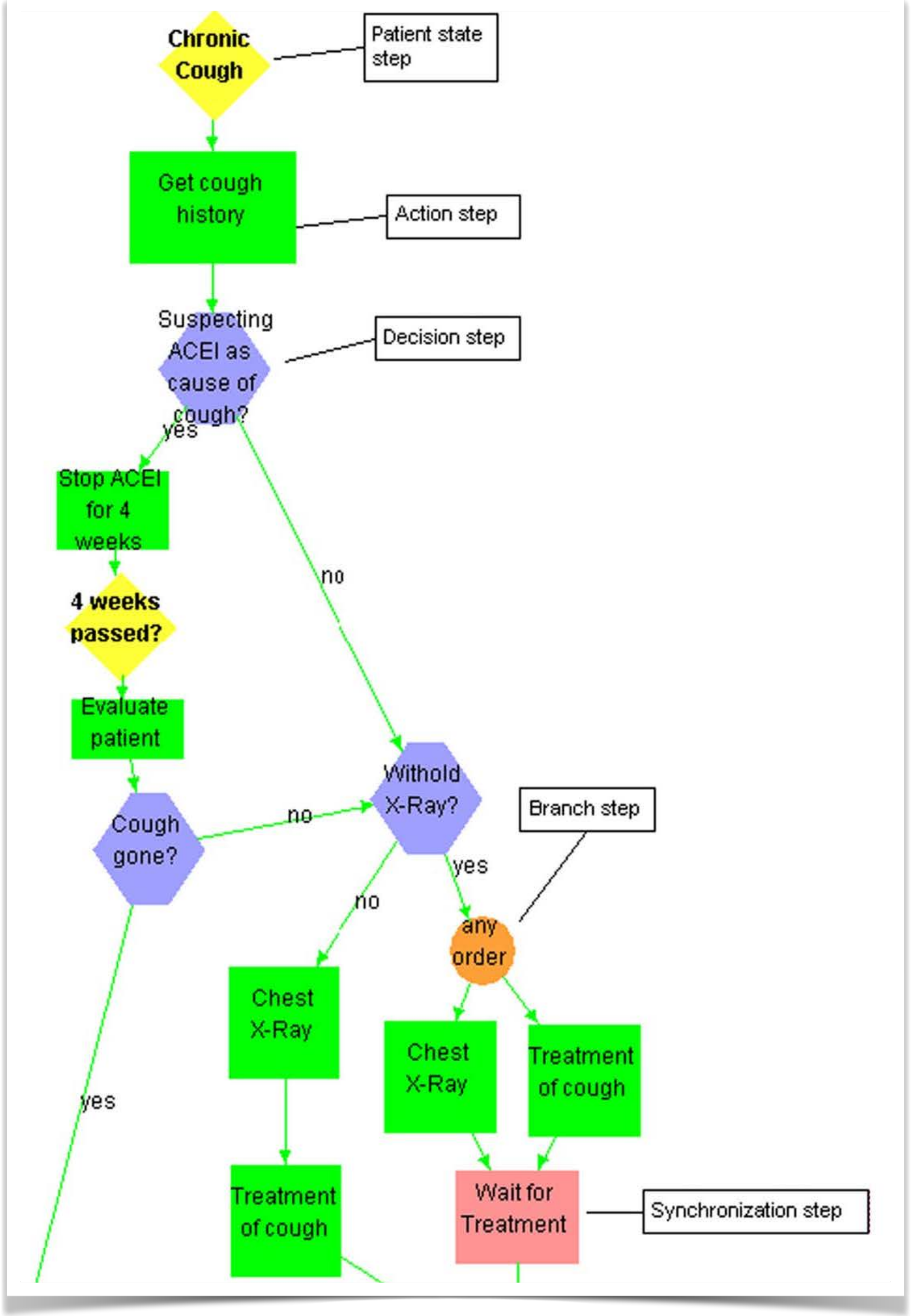
Credits: SEE-lab, Technion

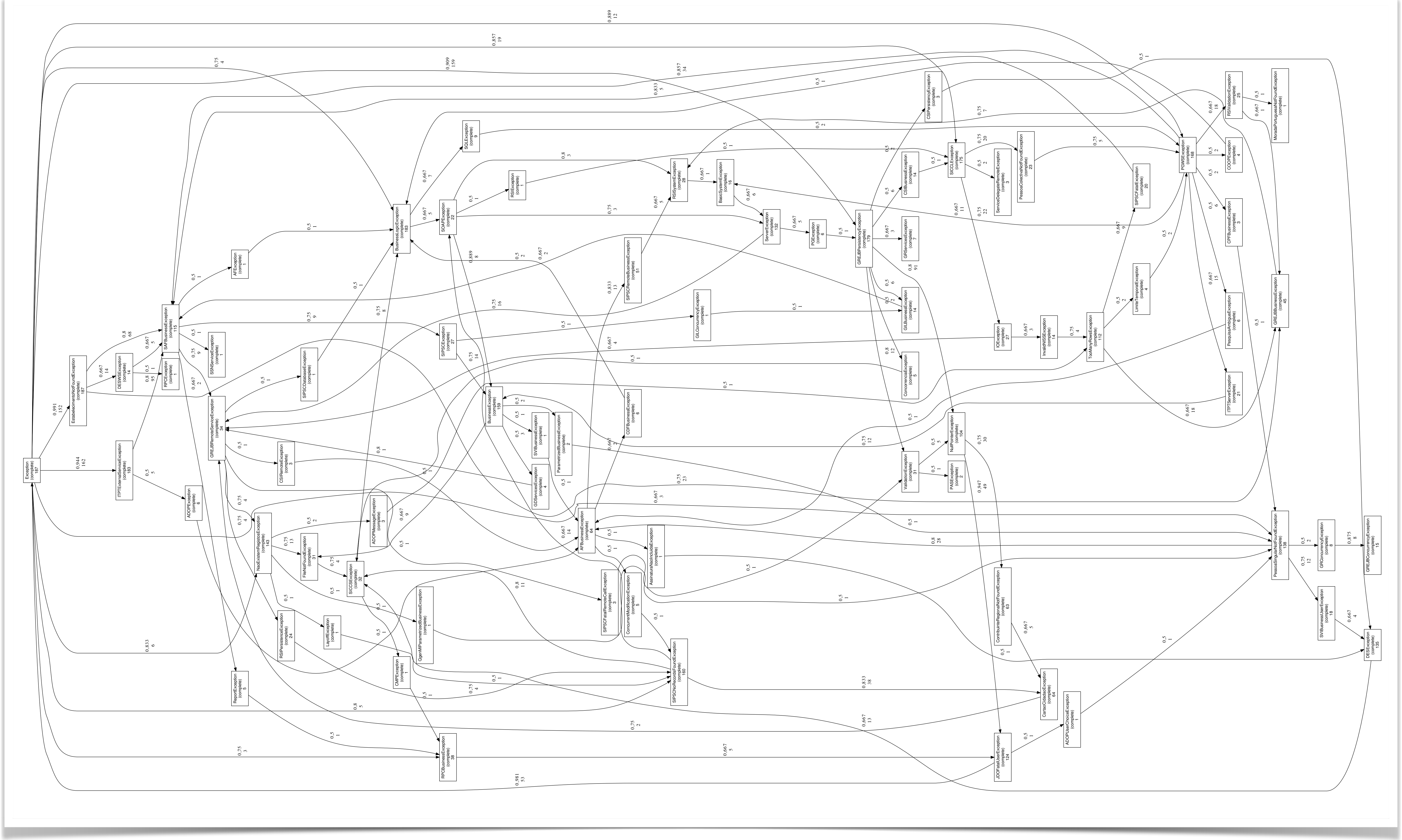
Patients Flow, HomeHospital
January 2005



Difference between models and reality

2/3: reality is often more complex than expected





Simplicity cannot be obtained by sweeping complexity under the carpet

Difference between models and reality

3/3: domain experts find their own way

Difference between models and reality

3/3: domain experts find their own way

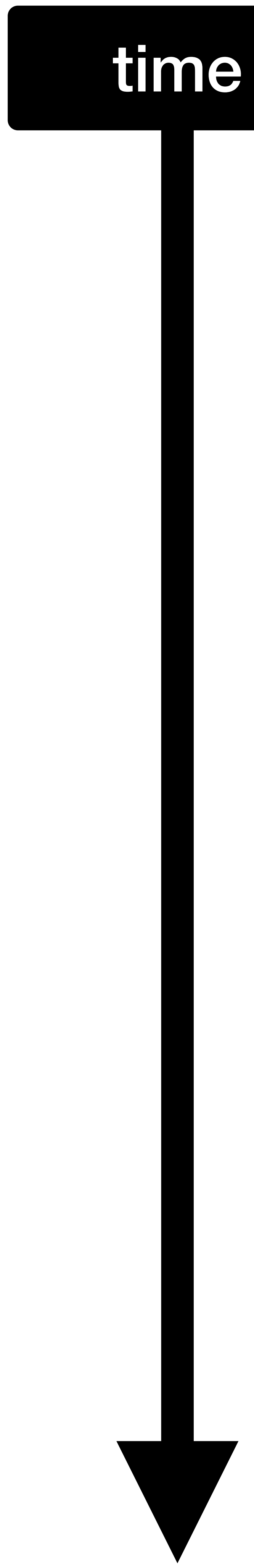

REALITY
enact/
monito
knowledge
workers
process models
event logs
(re)
design
managers/
analysts
ORGANISATION

REALITY
process models
event logs
enact/
monito
(re)
design
knowledge
workers
managers/
analysts

Event logs

https://ceur-ws.org/Vol-1859/bpmds-08-paper.pdf

Case id	Activity	Timestamp	Transaction type	Resource	…
…	…	…	…	…	
253	ER Triage	04–13-2021 11:33:50	complete	Nurse 1	…
255	Release A	04–13-2021 11:35:05	complete	Physician 02	…
259	Lactic Acid	04–13-2021 11:38:55	complete	Nurse 4	…
254	Leucocytes	04–13-2021 11:41:23	complete	Nurse 5	…
256	Lactic Acid	04–13-2021 11:52:35	complete	Nurse 4	…
257	ER Triage	04–13-2021 11:53:16	complete	Nurse 7	…
258	ER Registration	04–13-2021 11:54:47	complete	Nurse 8	…
253	Admission NC	04–13-2021 11:55:26	complete	Physician 02	…
259	Admission IC	04–13-2021 11:58:30	complete	Physician 03	…
260	CRP	04–13-2021 12:01:12	complete	Nurse 07	…
261	Release B	04–13-2021 12:02:00	complete	Physician 03	…
253	IV Liquid	04–13-2021	complete	Nurse 2	…



Event logs

https://ceur-ws.org/Vol-1859/bpmds-08-paper.pdf

subject/instance | what | when | step | who | other data attributes

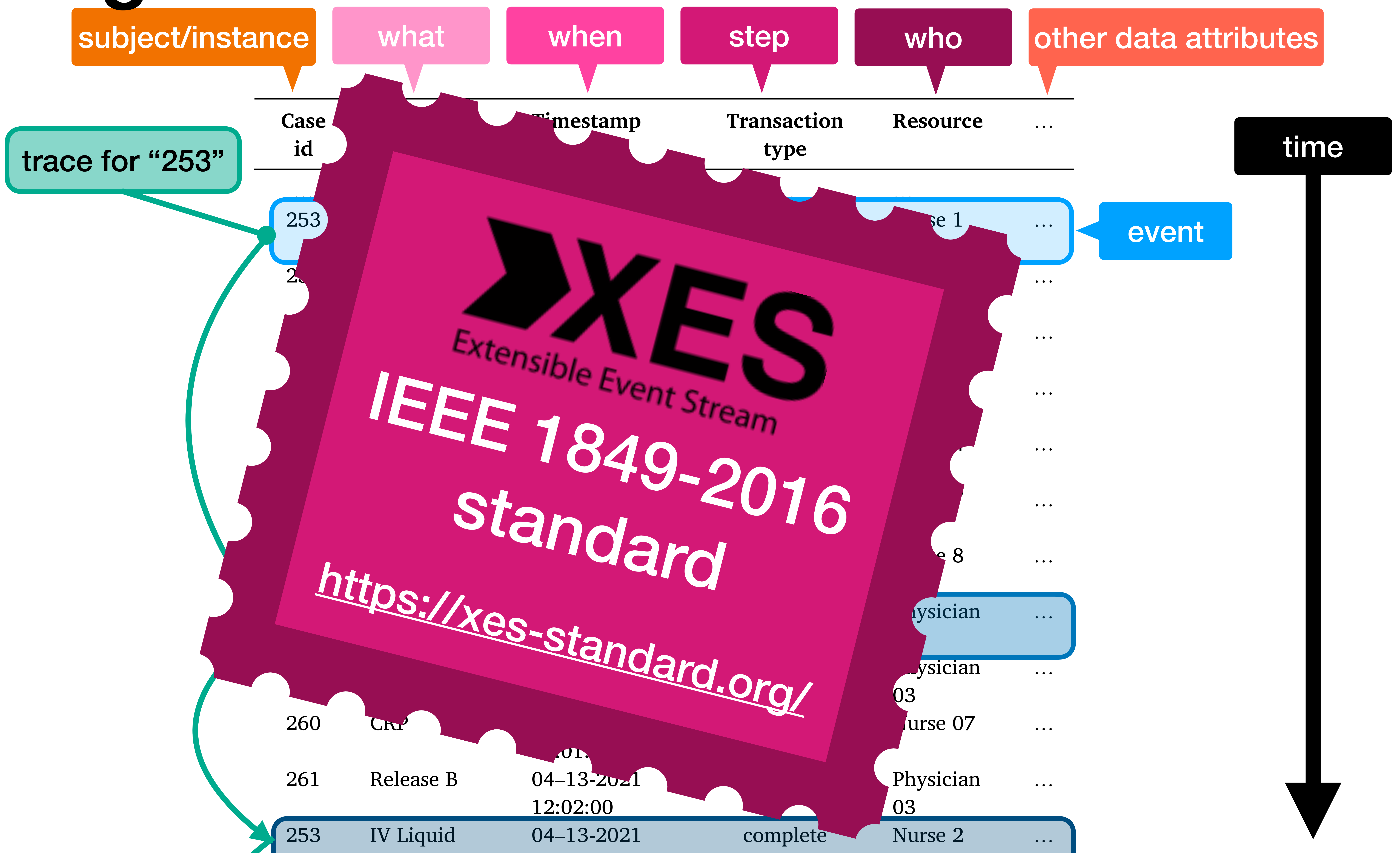
Case id	Activity	Timestamp	Transaction type	Resource	...
...	...	...	...	...	
253	ER Triage	04–13-2021 11:33:50	complete	Nurse 1	...
255	Release A	04–13-2021 11:35:05	complete	Physician 02	...
259	Lactic Acid	04–13-2021 11:38:55	complete	Nurse 4	...
254	Leucocytes	04–13-2021 11:41:23	complete	Nurse 5	...
256	Lactic Acid	04–13-2021 11:52:35	complete	Nurse 4	...
257	ER Triage	04–13-2021 11:53:16	complete	Nurse 7	...
258	ER Registration	04–13-2021 11:54:47	complete	Nurse 8	...
253	Admission NC	04–13-2021 11:55:26	complete	Physician 02	...
259	Admission IC	04–13-2021 11:58:30	complete	Physician 03	...
260	CRP	04–13-2021 12:01:12	complete	Nurse 07	...
261	Release B	04–13-2021 12:02:00	complete	Physician 03	...
253	IV Liquid	04–13-2021	complete	Nurse 2	...

event

time

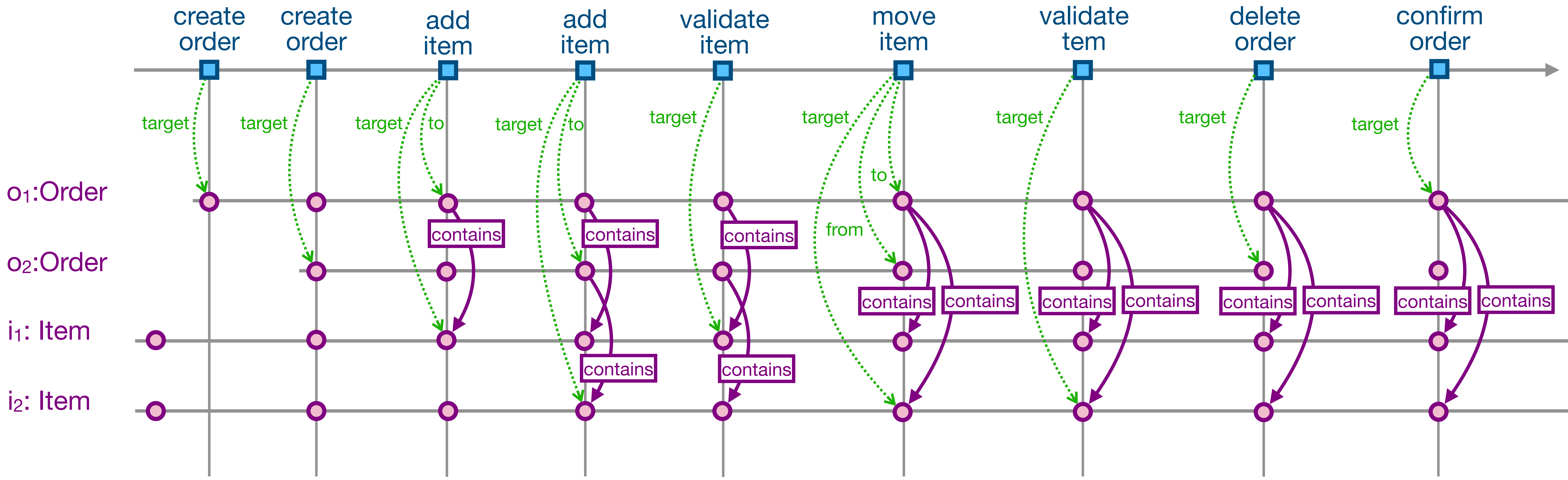
Event logs

https://ceur-ws.org/Vol-1859/bpmds-08-paper.pdf



From traces to temporal graphs

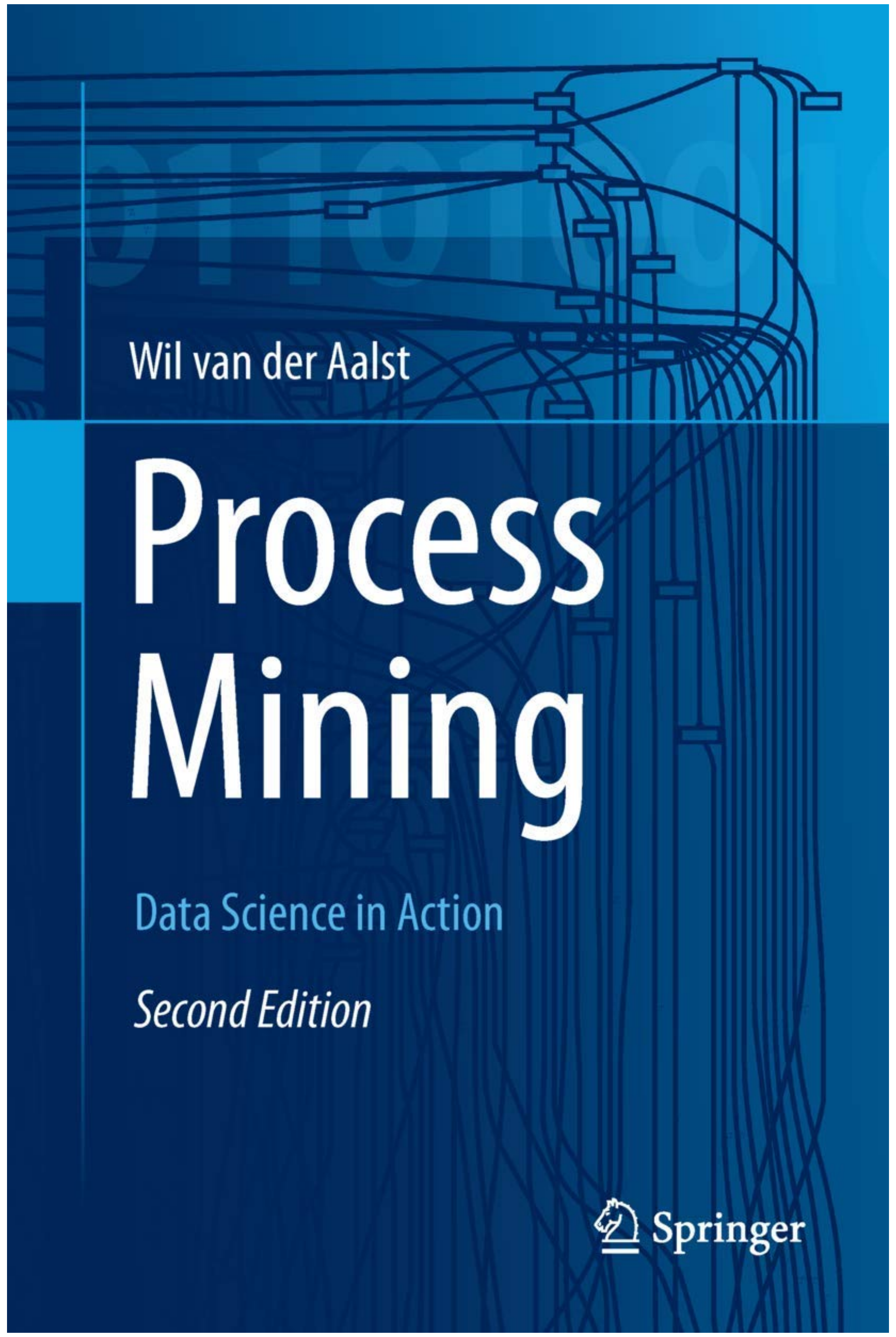
Research frontier…

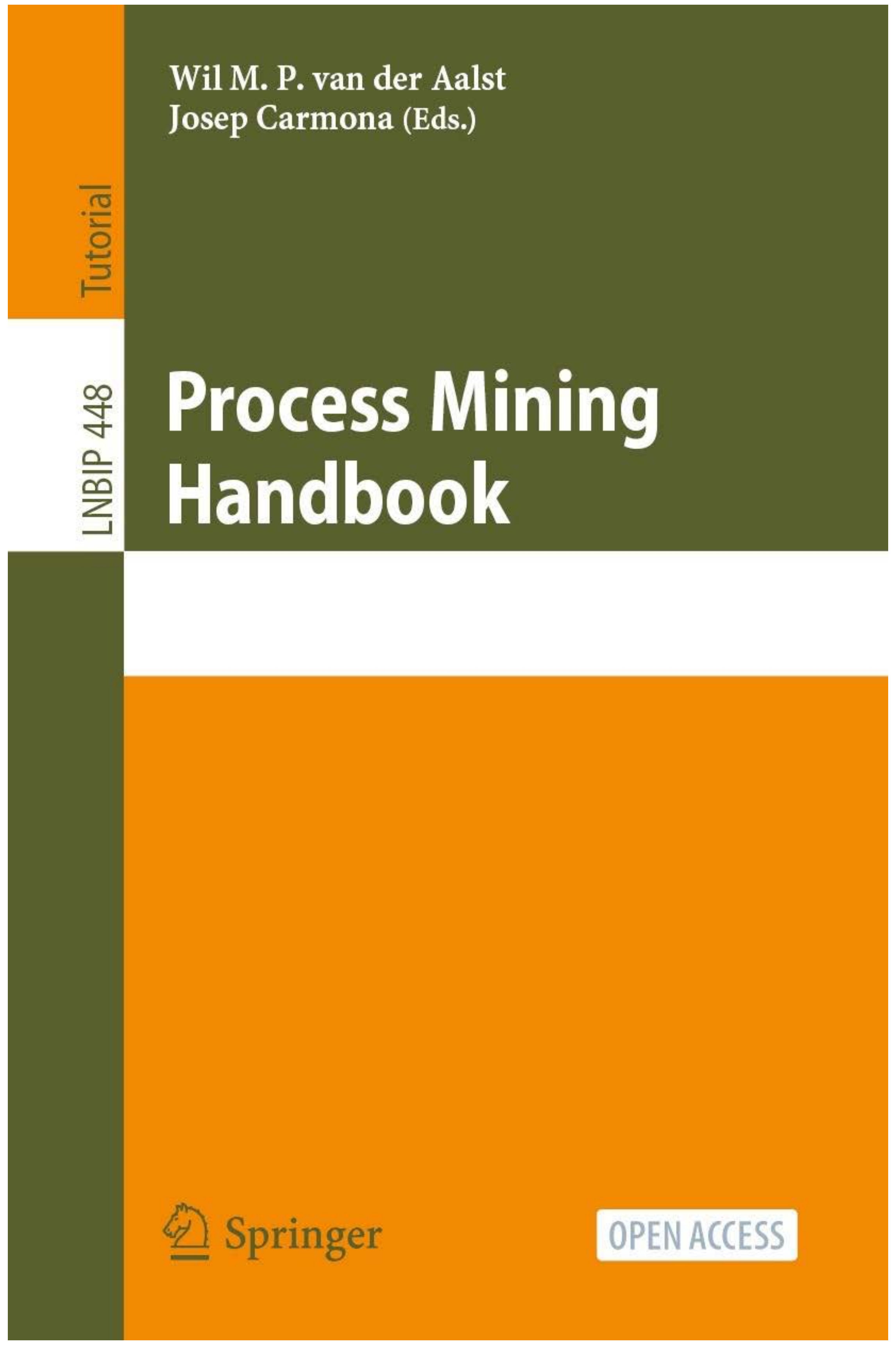


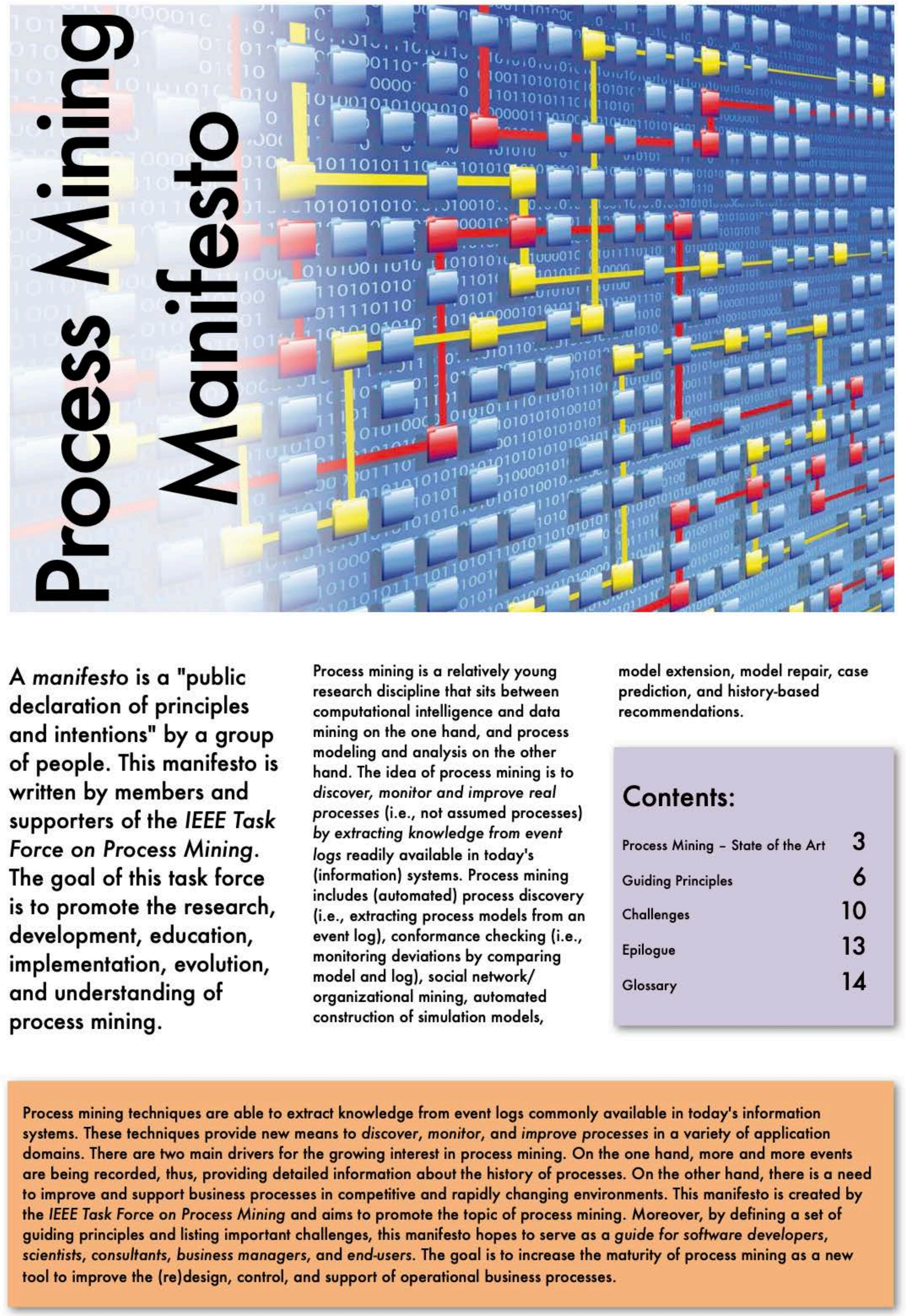

REALITY
enact/
monito
knowledge
workers
process models
event logs
(re)
design
managers/
analysts
ORGANISATION


process models
event logs
(re)
design
managers/
analysts
data-driven
process management
ORGANISATION

Process mining: “data science in action”







Process Mining Manifesto

A *manifesto* is a "public declaration of principles and intentions" by a group of people. This manifesto is written by members and supporters of the *IEEE Task Force on Process Mining*. The goal of this task force is to promote the research, development, education, implementation, evolution, and understanding of process mining.

Process mining is a relatively young research discipline that sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand. The idea of process mining is to *discover, monitor and improve real processes* (i.e., not assumed processes) *by extracting knowledge from event logs* readily available in today's (information) systems. Process mining includes (automated) process discovery (i.e., extracting process models from an event log), conformance checking (i.e., monitoring deviations by comparing model and log), social network/ organizational mining, automated construction of simulation models, model extension, model repair, case prediction, and history-based recommendations.

Contents:

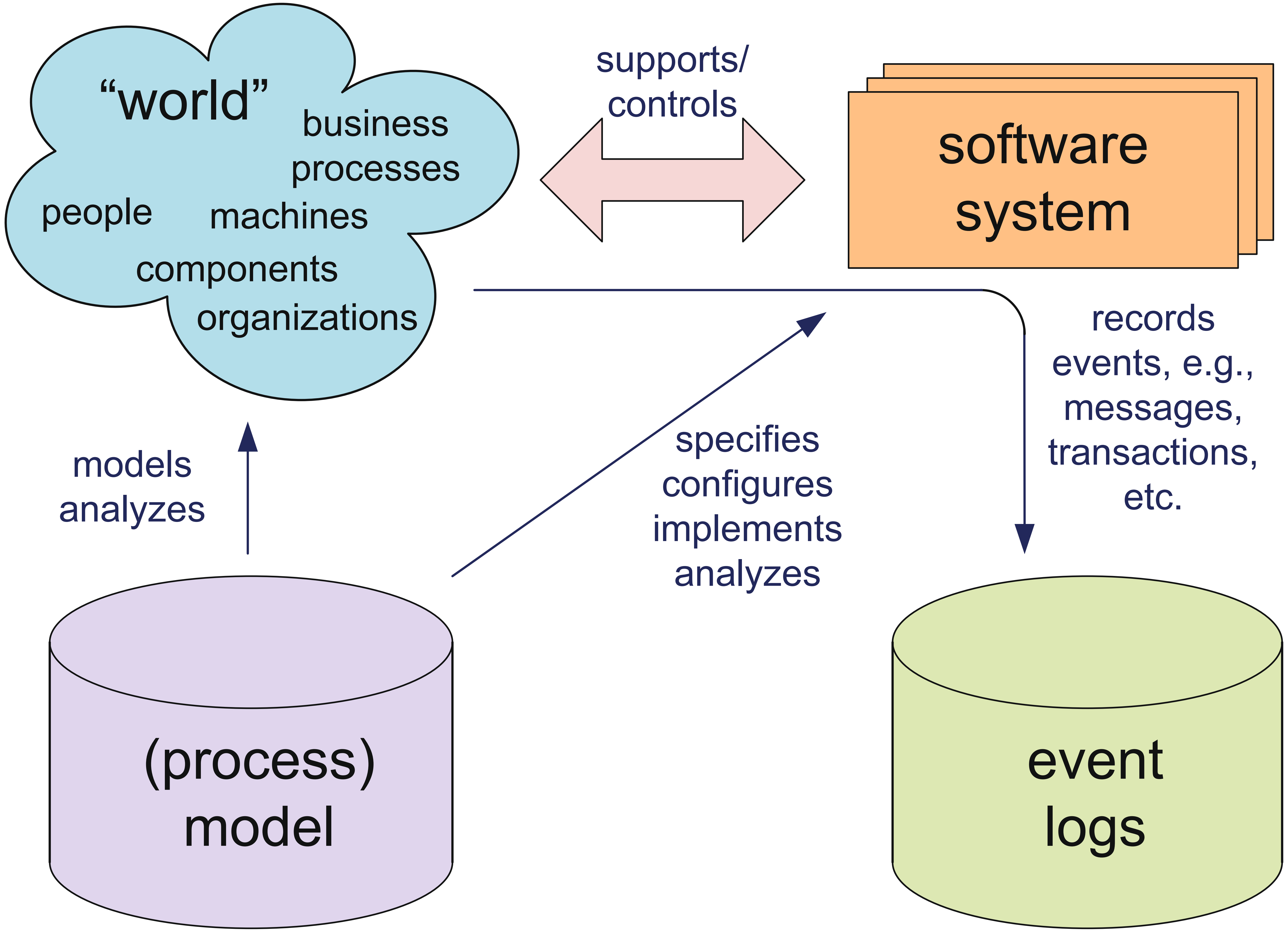
Process mining techniques are able to extract knowledge from event logs commonly available in today's information systems. These techniques provide new means to *discover*, *monitor*, and *improve processes* in a variety of application domains. There are two main drivers for the growing interest in process mining. On the one hand, more and more events are being recorded, thus, providing detailed information about the history of processes. On the other hand, there is a need to improve and support business processes in competitive and rapidly changing environments. This manifesto is created by the *IEEE Task Force on Process Mining* and aims to promote the topic of process mining. Moreover, by defining a set of guiding principles and listing important challenges, this manifesto hopes to serve as a *guide for software developers, scientists, consultants, business managers*, and *end-users*. The goal is to increase the maturity of process mining as a new tool to improve the (re)design, control, and support of operational business processes.

IEEE TASK FORCE ON PROCESS MINING

https://www.tf-pm.org

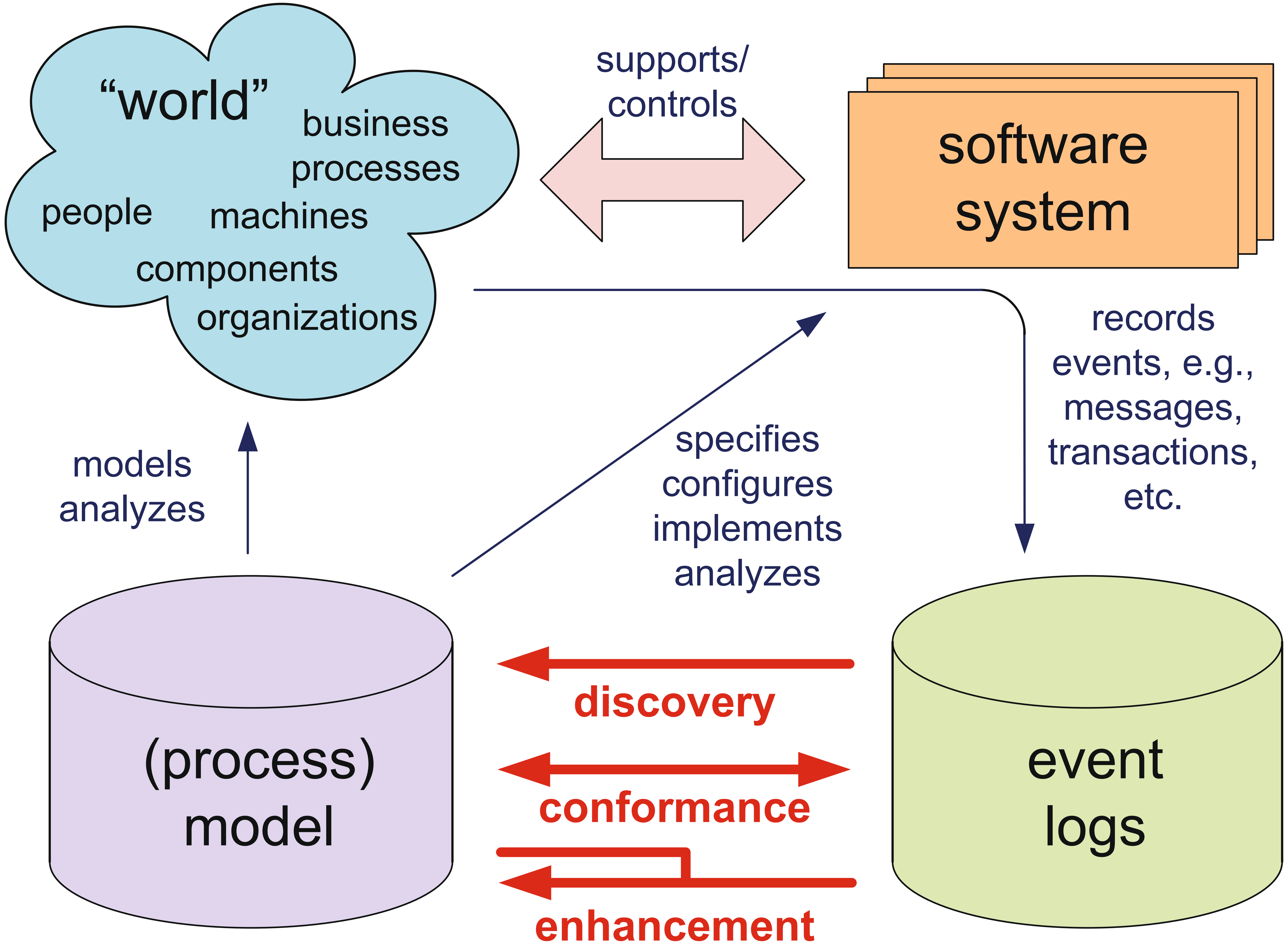
The process mining framework

Original picture by Wil van der Aalst



The process mining framework

Original picture by Wil van der Aalst

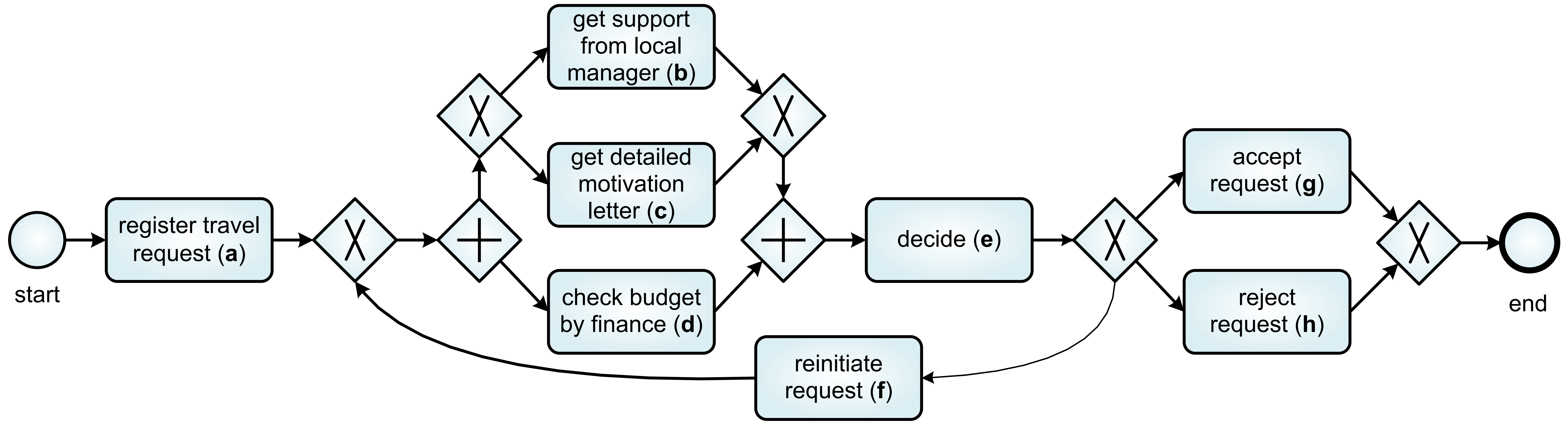


Play in: automated discovery

Model learning from trace data - typically, with positive examples only

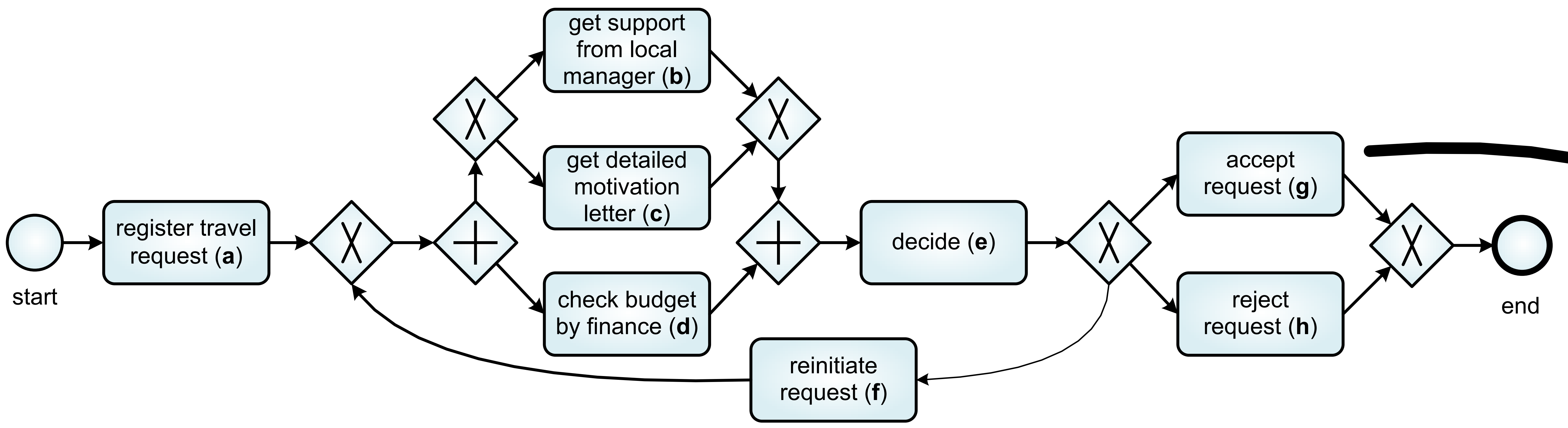
Case	Activity	Timestamp	Resource
432	register travel request (a)	18-3-2014:9.15	John
432	get support from local manager (b)	18-3-2014:9.25	Mary
432	check budget by finance (d)	19-3-2014:8.55	John
432	decide (e)	19-3-2014:9.36	Sue
432	accept request (g)	19-3-2014:9.48	Mary

...



Play out: synthetic event log generation

Synthesis of meaningful event data from a model

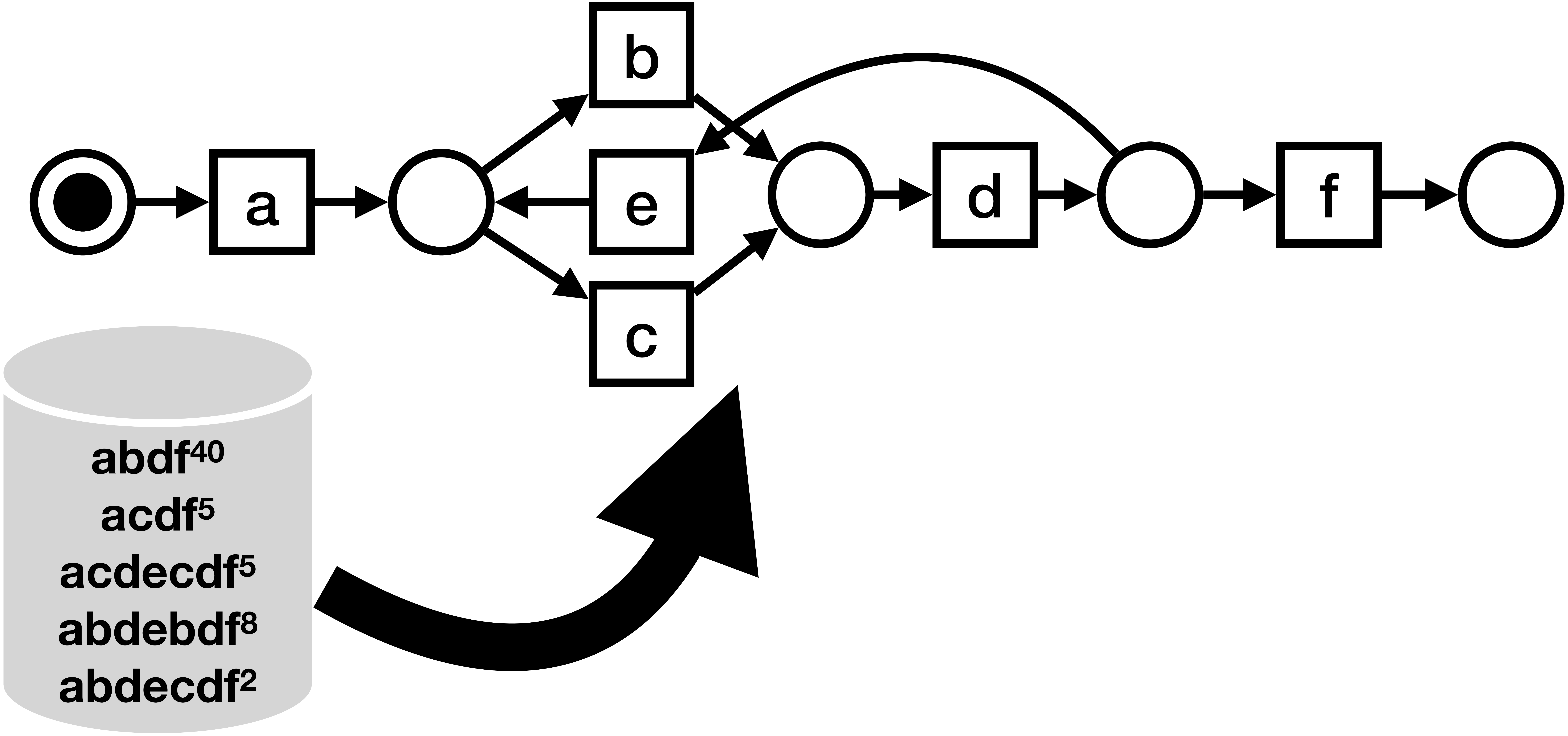


Case	Activity	Timestamp	Resource
432	register travel request (a)	18-3-2014:9.15	John
432	get support from local manager (b)	18-3-2014:9.25	Mary
432	check budget by finance (d)	19-3-2014:8.55	John
432	decide (e)	19-3-2014:9.36	Sue
432	accept request (g)	19-3-2014:9.48	Mary

...

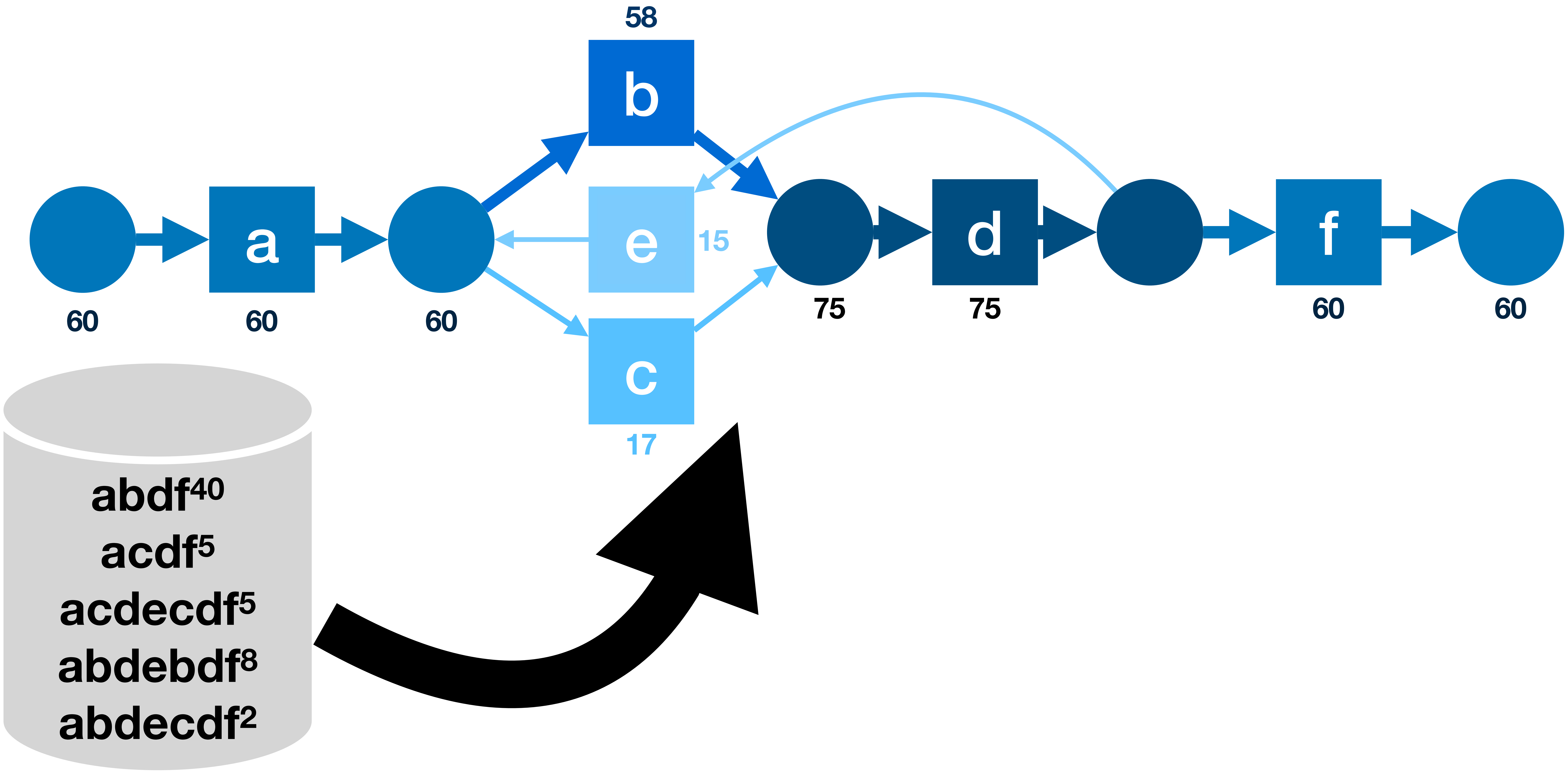
Replay: enhancement

Data augmentation of the model



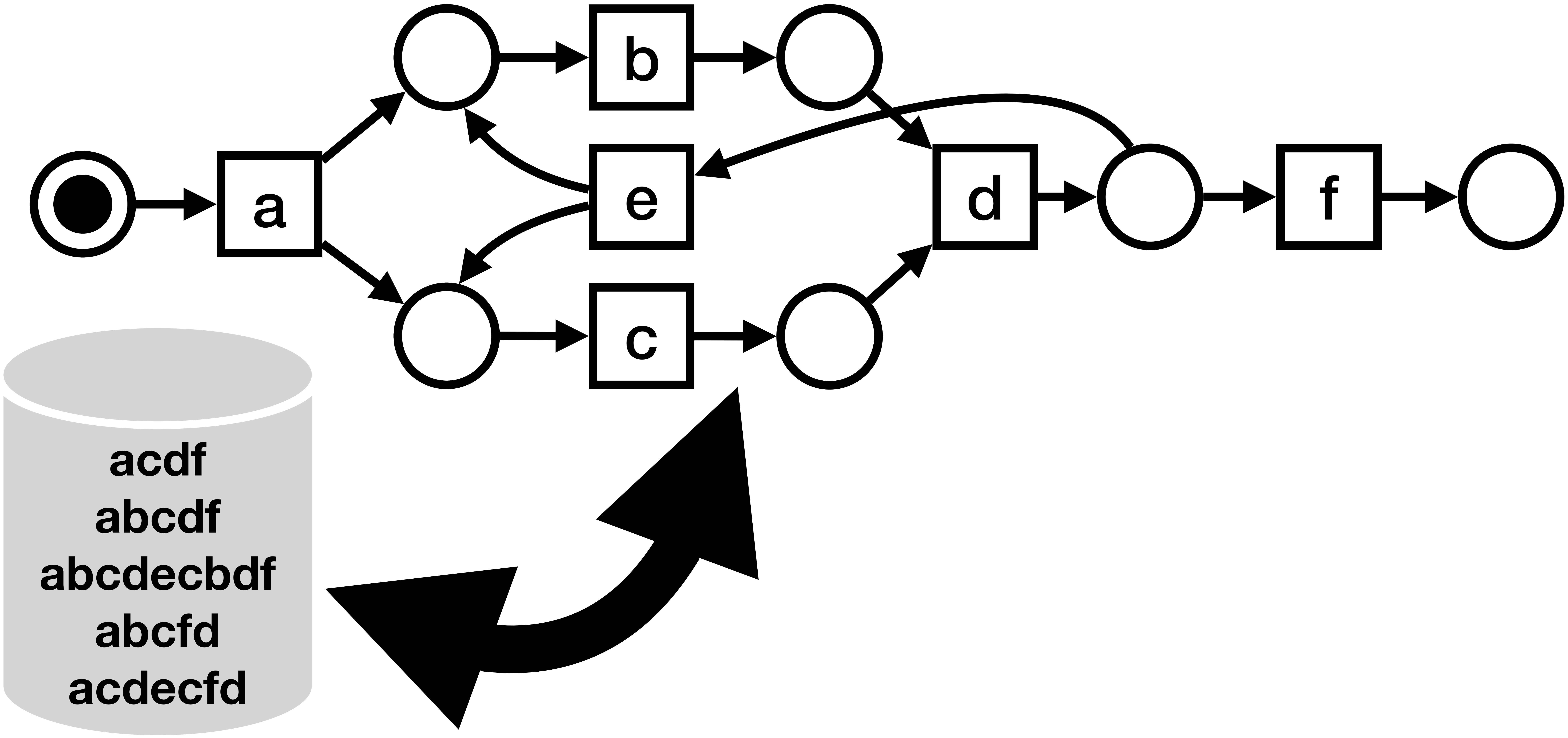
Replay: enhancement

Data augmentation of the model



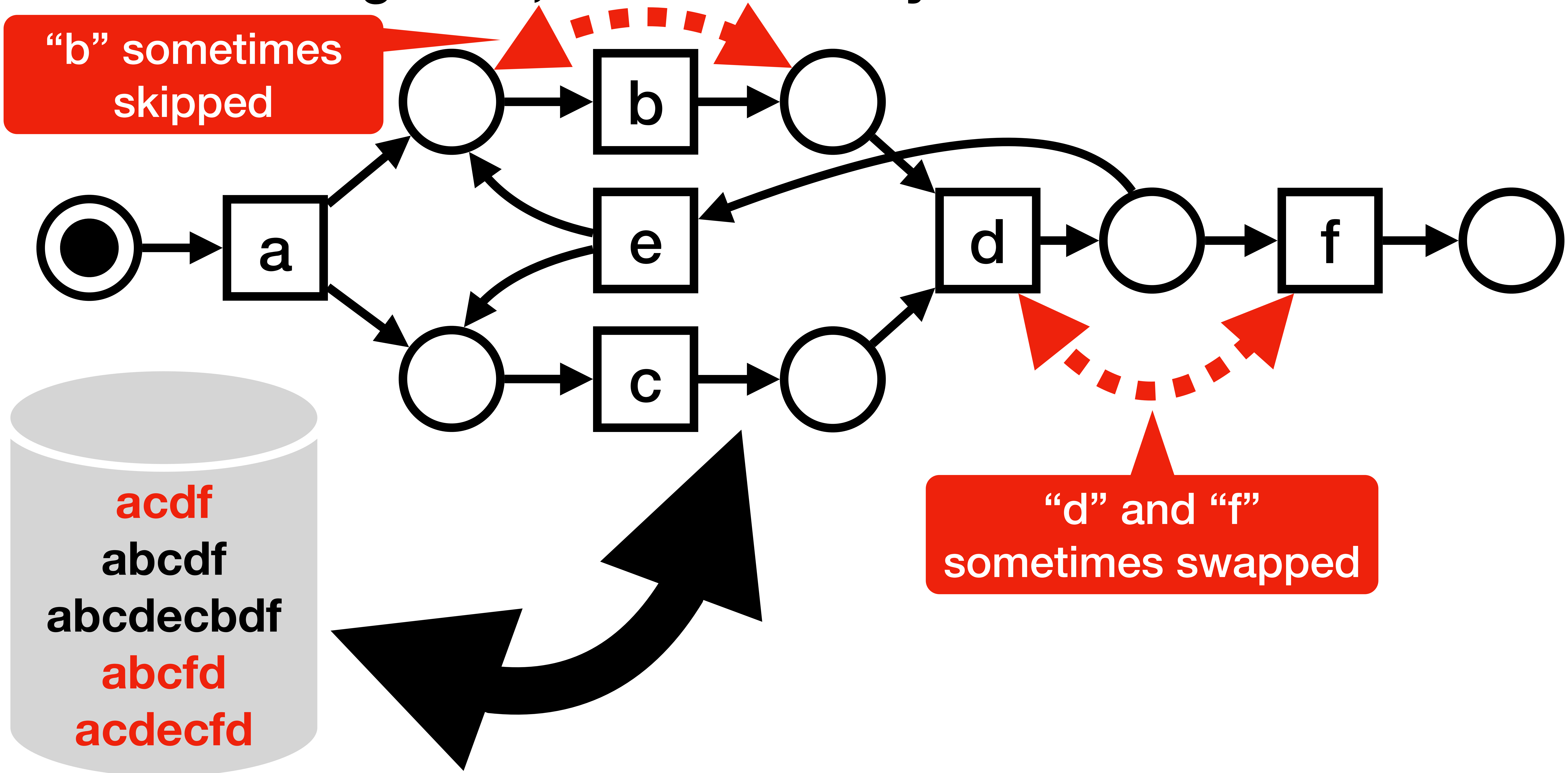
Replay: conformance checking

Behaviour recognition, deviation analysis



Replay: conformance checking

Behaviour recognition, deviation analysis



AI and process science

AI-augmentation along the BPM lifecycle

Credits: Marlon Dumas

How do my processes look like?
Where are the bottlenecks,
wastes, compliance violations,
positive and negative deviances?

(descriptive)
process mining

Automated process discovery
Conformance checking
Performance mining
Variant analysis

AI-augmentation along the BPM lifecycle

Credits: Marlon Dumas

AI-augmented Business Process Management Systems: A Research Manifesto

Authors: Marlon Dumas, Fabiana Fournier, Lior Limonad, Andrea Marrella, Marco Montali, Jana-Rebecca Rehse, Rafael Accorsi, Diego Calvanese, Giuseppe De Giacomo, Dirk Fahland, + 4 Authors Info & Claims

ACM Transactions on Management Information Systems, Volume 14, Issue 1 • Article No.: 11, Pages 1 - 19
https://doi.org/10.1145/3576047

Published: 31 January 2023 Publication History

Abstract

AI-augmented Business Process Management Systems (ABPMSs) are an emerging class of process-aware information systems, empowered by trustworthy AI technology. An ABPMS enhances the execution of business processes with the aim of making these processes more adaptable, proactive, explainable, and context-sensitive. This manifesto presents a vision for ABPMSs and discusses research challenges that need to be surmounted to realize this vision. To this end, we define the concept of ABPMS, we outline the lifecycle of processes within an

AI-augmented Business Process Management Systems: A Research Manifesto

Authors: Marlon Dumas, Fabiana Fournier, Lior Limonad, Andrea Marrella, Marco Montali, Jana-Rebecca Rehse, Rafael Accorsi, Diego Calvanese, Giuseppe De Giacomo, Dirk Fahland, + 4 Authors Info & Claims

ACM Transactions on Management Information Systems, Volume 14, Issue 1 • Article No.: 11, Pages 1 - 19
https://doi.org/10.1145/3576047

An ABPMS is a process-aware information system that relies on trustworthy AI technology to reason and act upon data, within a set of restrictions, with the aim to continuously adapt and improve a set of business processes with respect to one or more performance indicators.

AI-augmented Business Process Management Systems: A Research Manifesto

Authors: Marlon Dumas, Fabiana Fournier, Lior Limonad, Andrea Marrella, Marco Montali, Jana-Rebecca Rehse, Rafael Accorsi, Diego Calvanese, Giuseppe De Giacomo, Dirk Fahland, + 4 | Authors Info & Claims

ACM Transactions on Manage

https://doi.org/10.1145/35760

framed autonomy

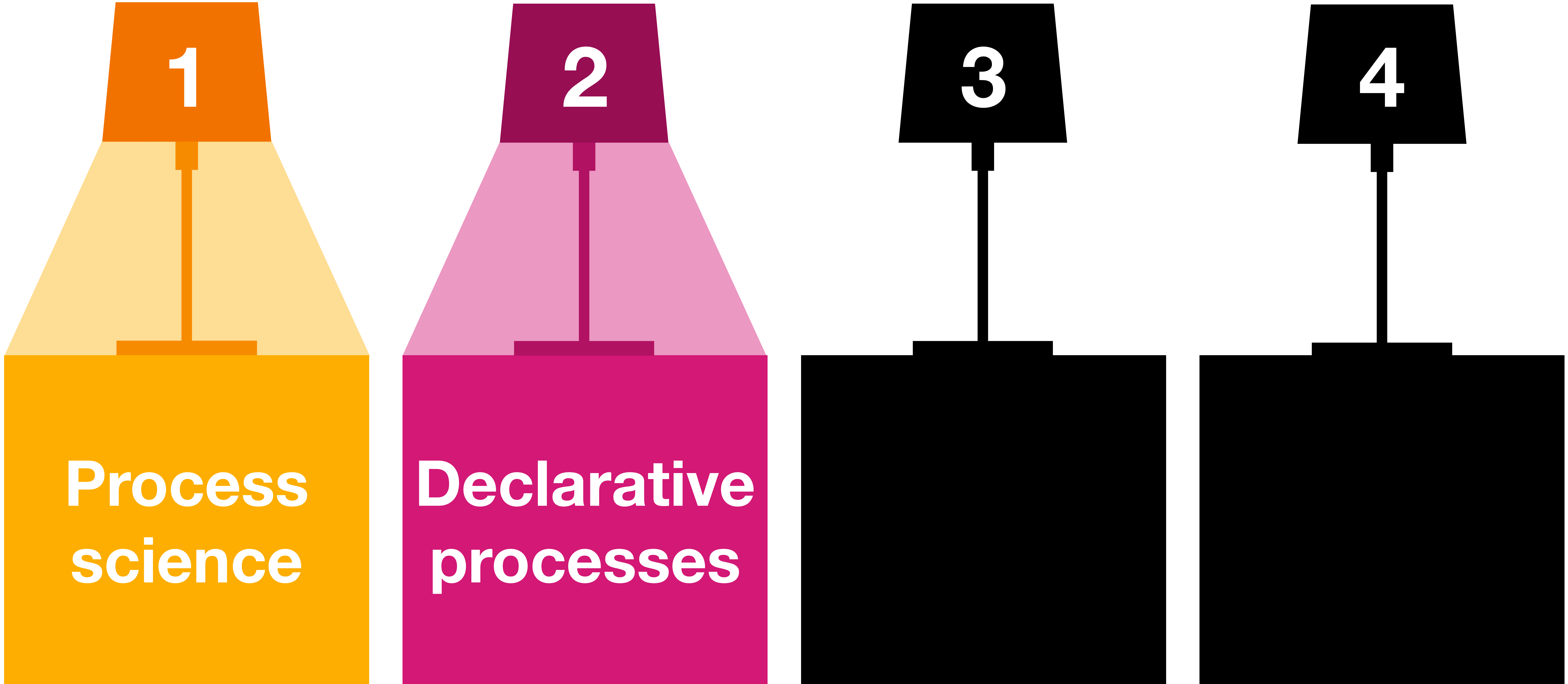
explainability

An ABPMS is a process-aware information system that relies on trustworthy AI technology to reason and act upon data, <u>within a set of restrictions</u>, with the aim to continuously adapt and improve a set of business processes with respect to one or more performance indicators.

adaptivity

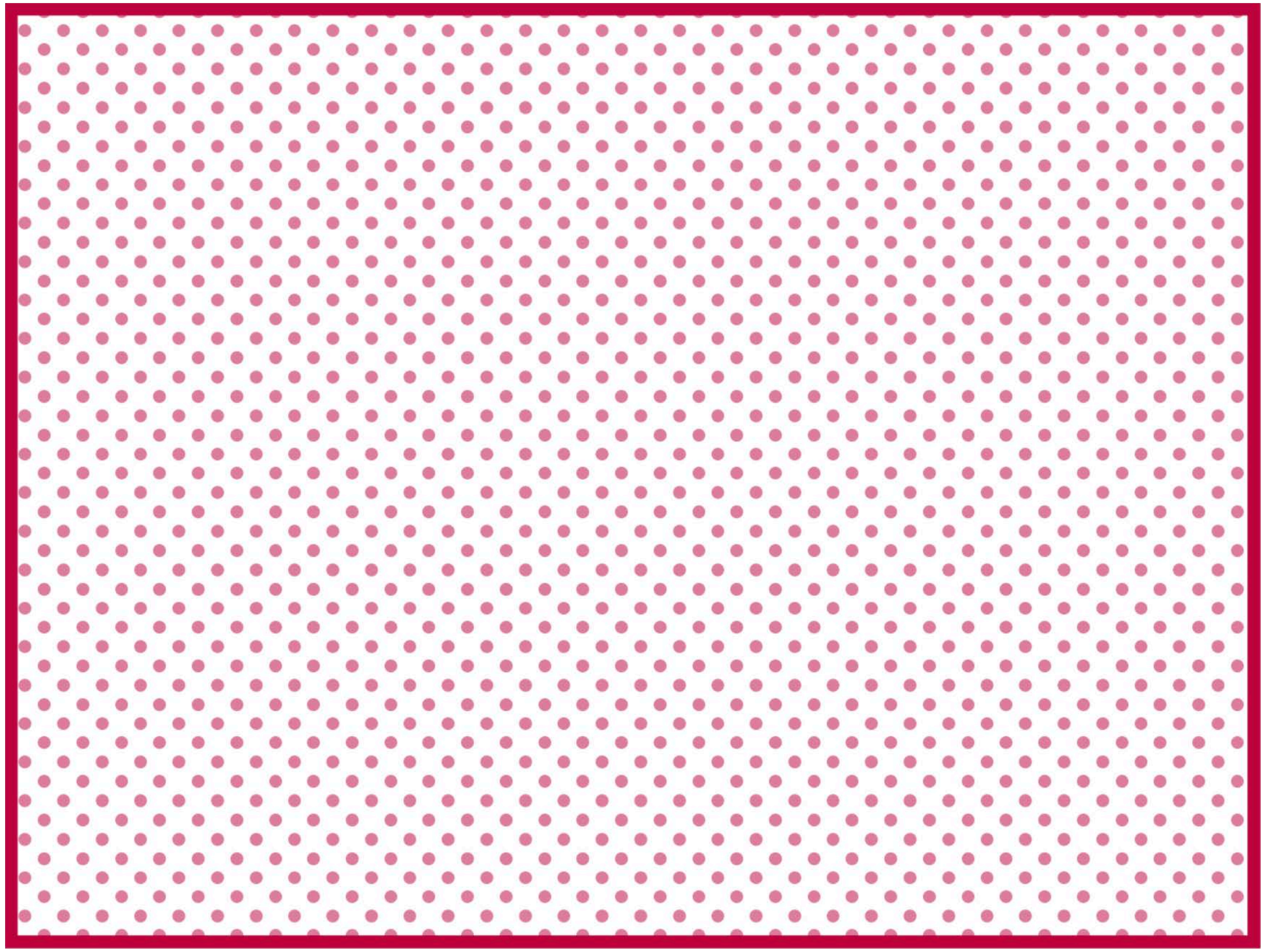
conversational actionability

2/4



What is a process?

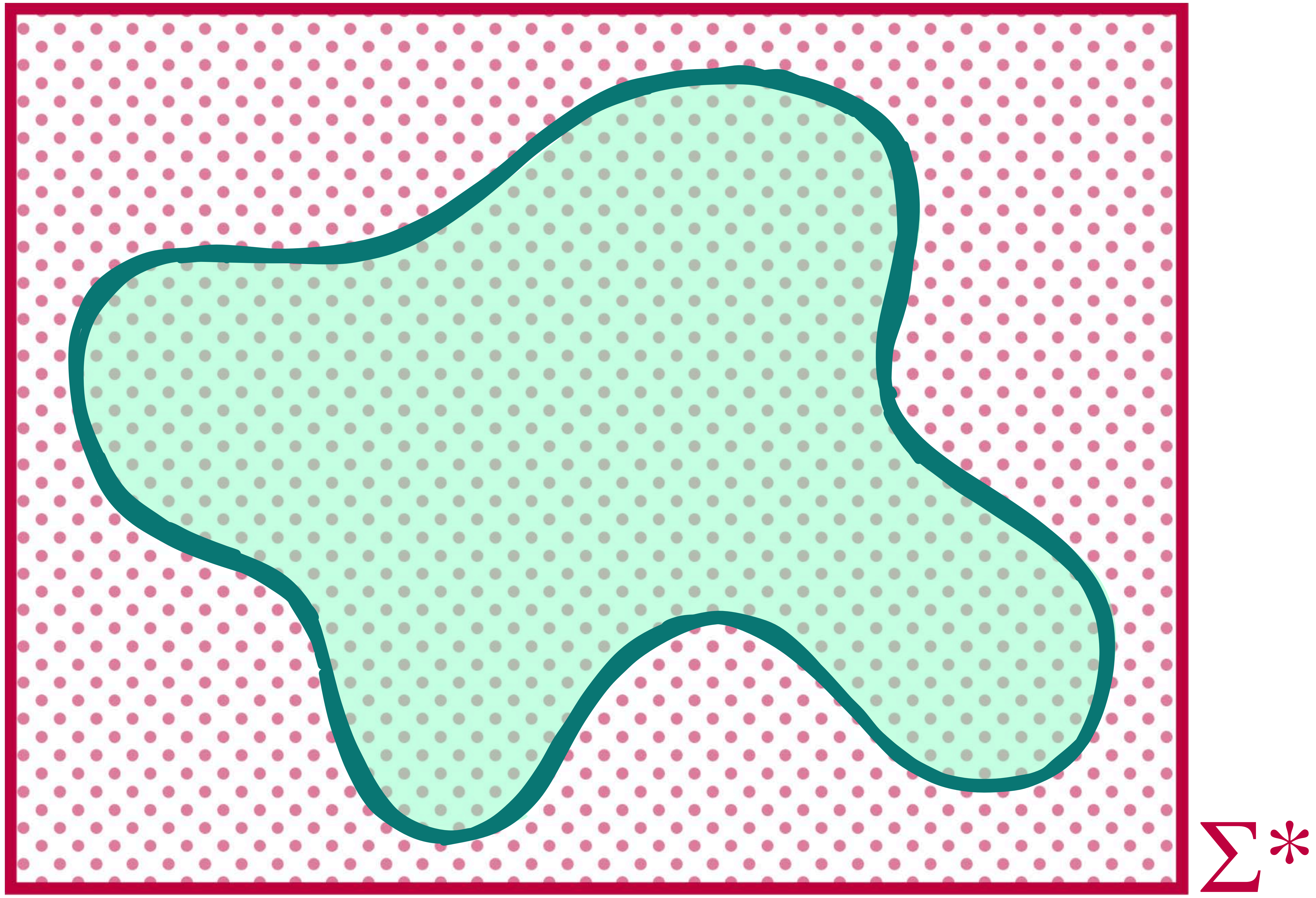
A possibly infinite set of finite traces



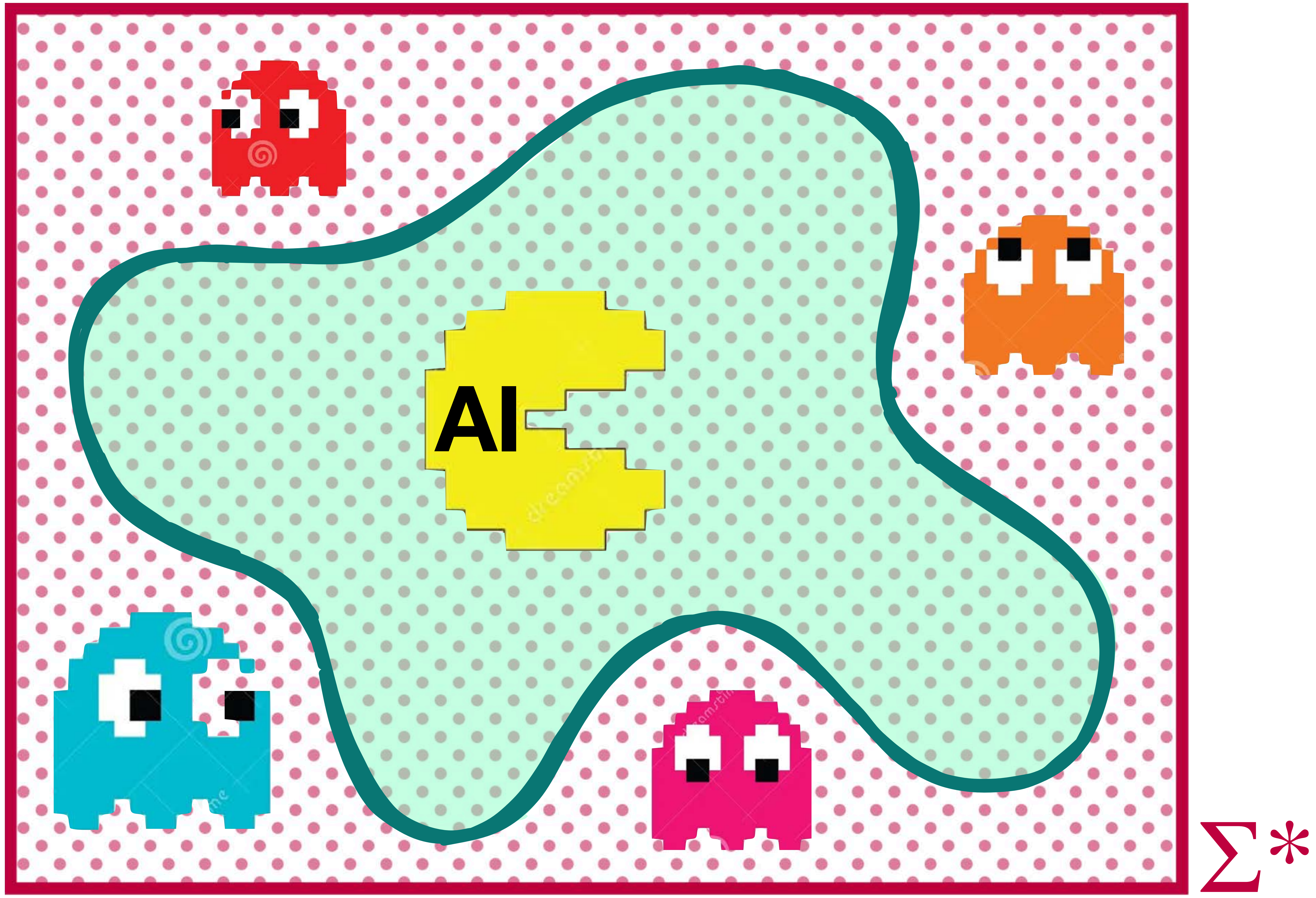
Σ^*

What is a process?

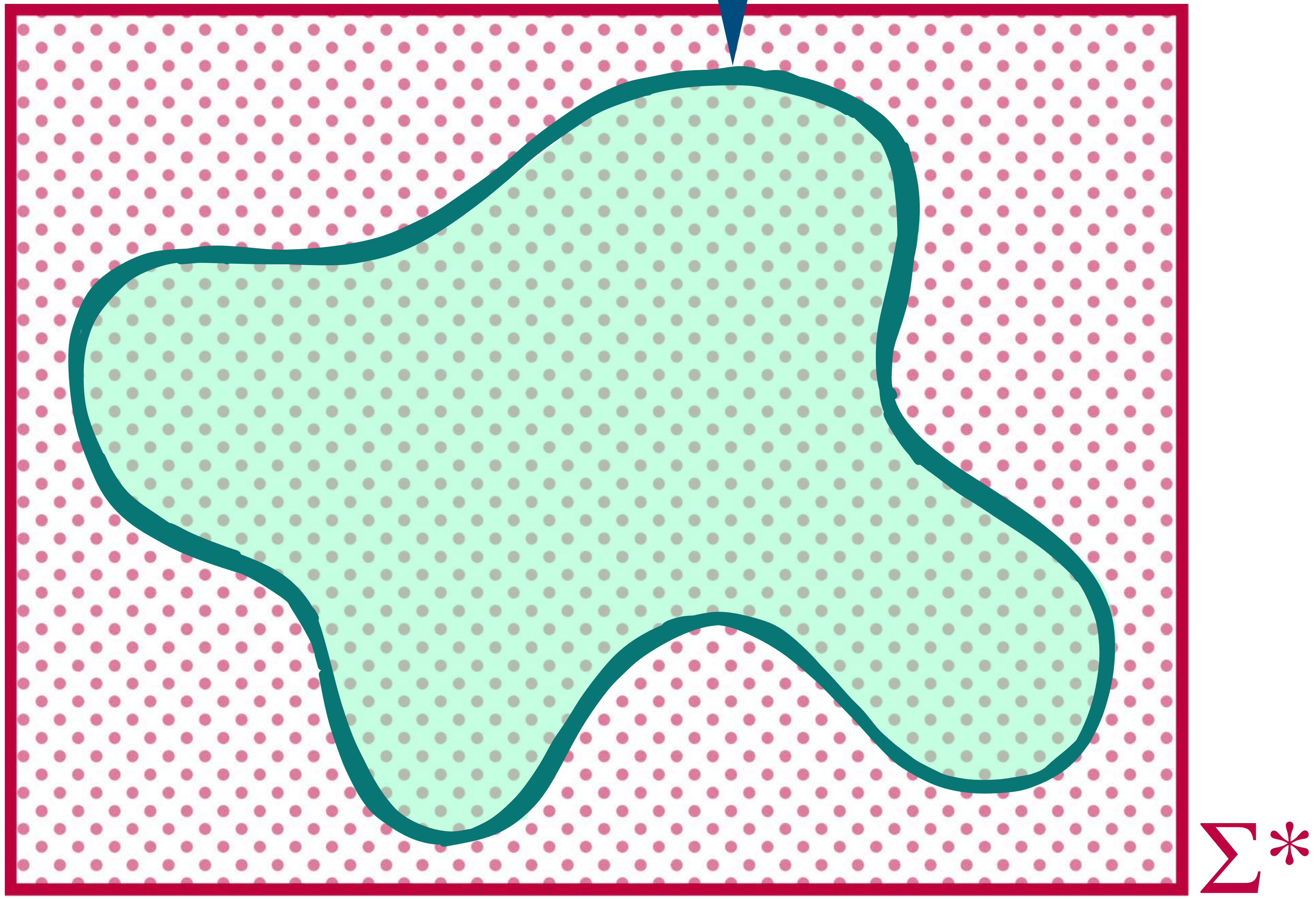
A possibly infinite set of finite traces



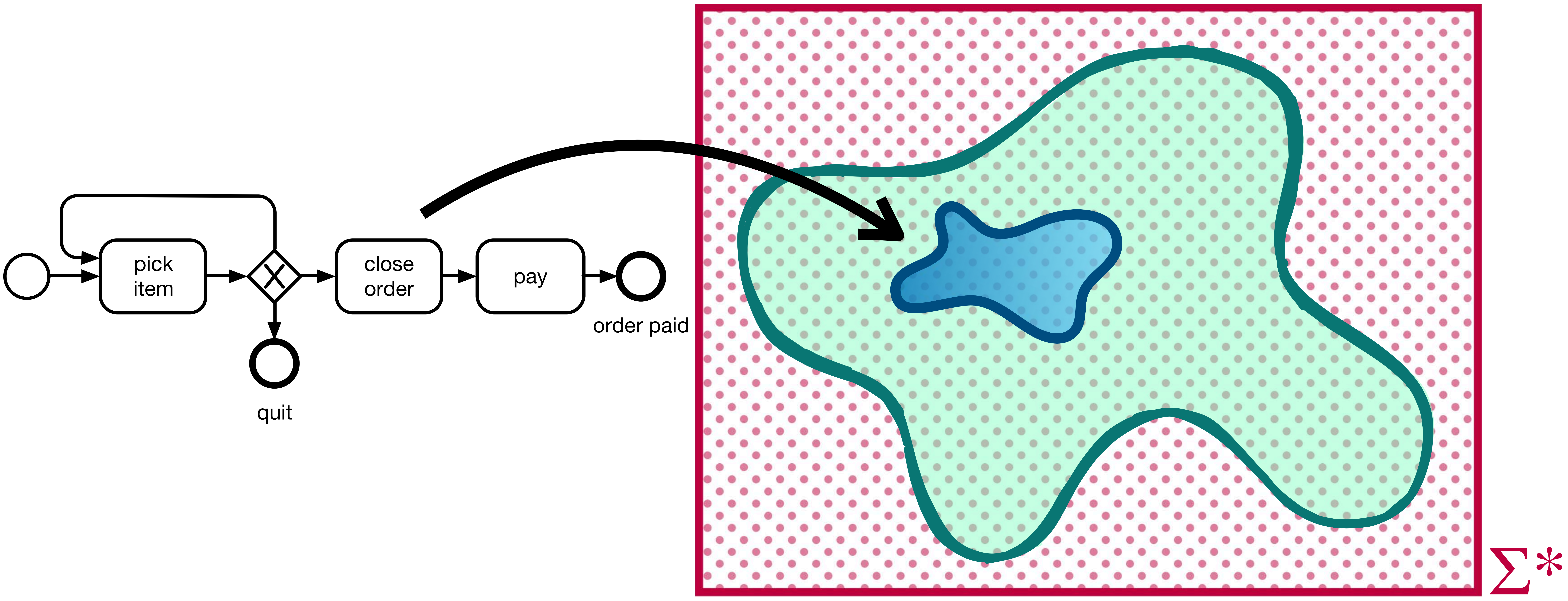
Framing in AI-augmented process science



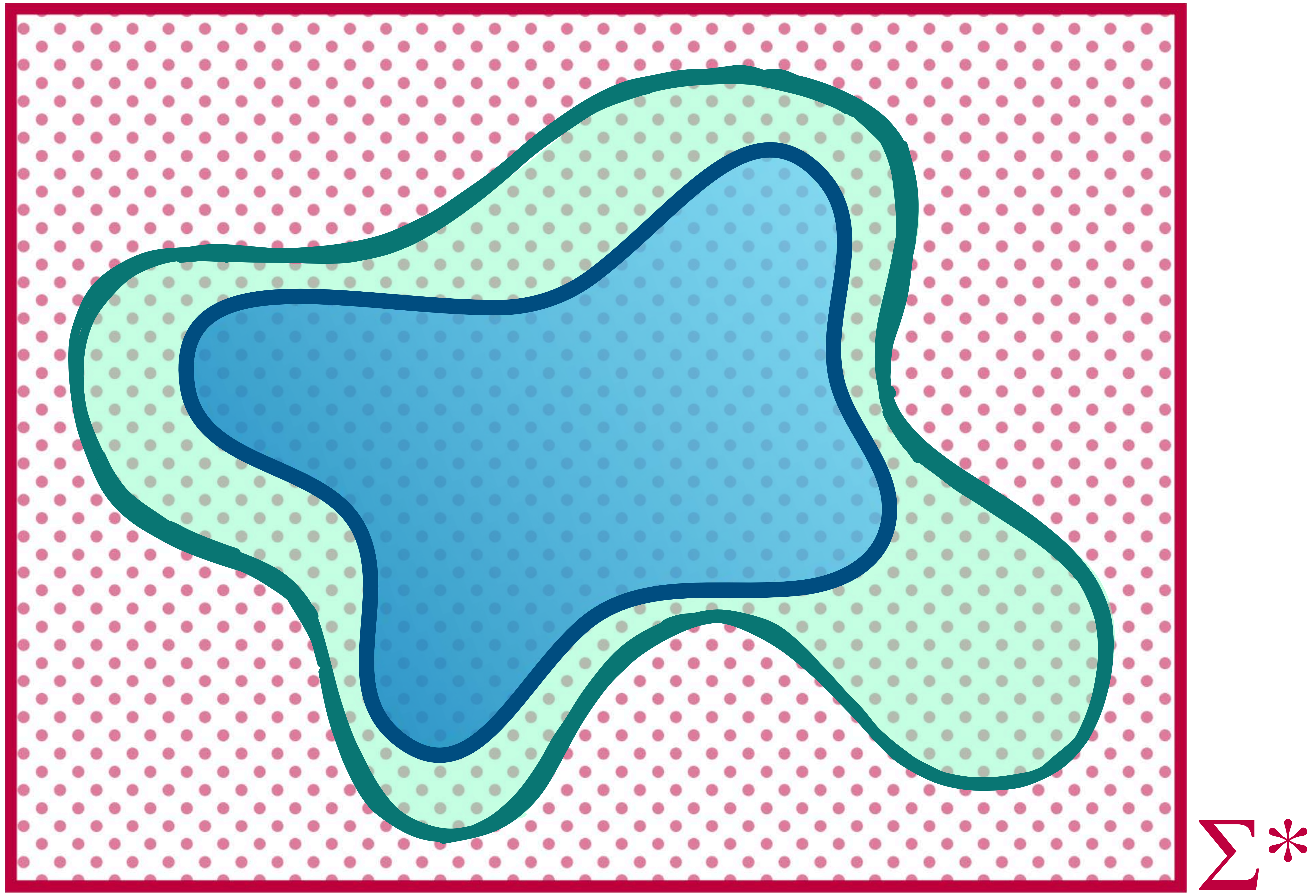
Key question: how to capture this set?



Procedural process specification

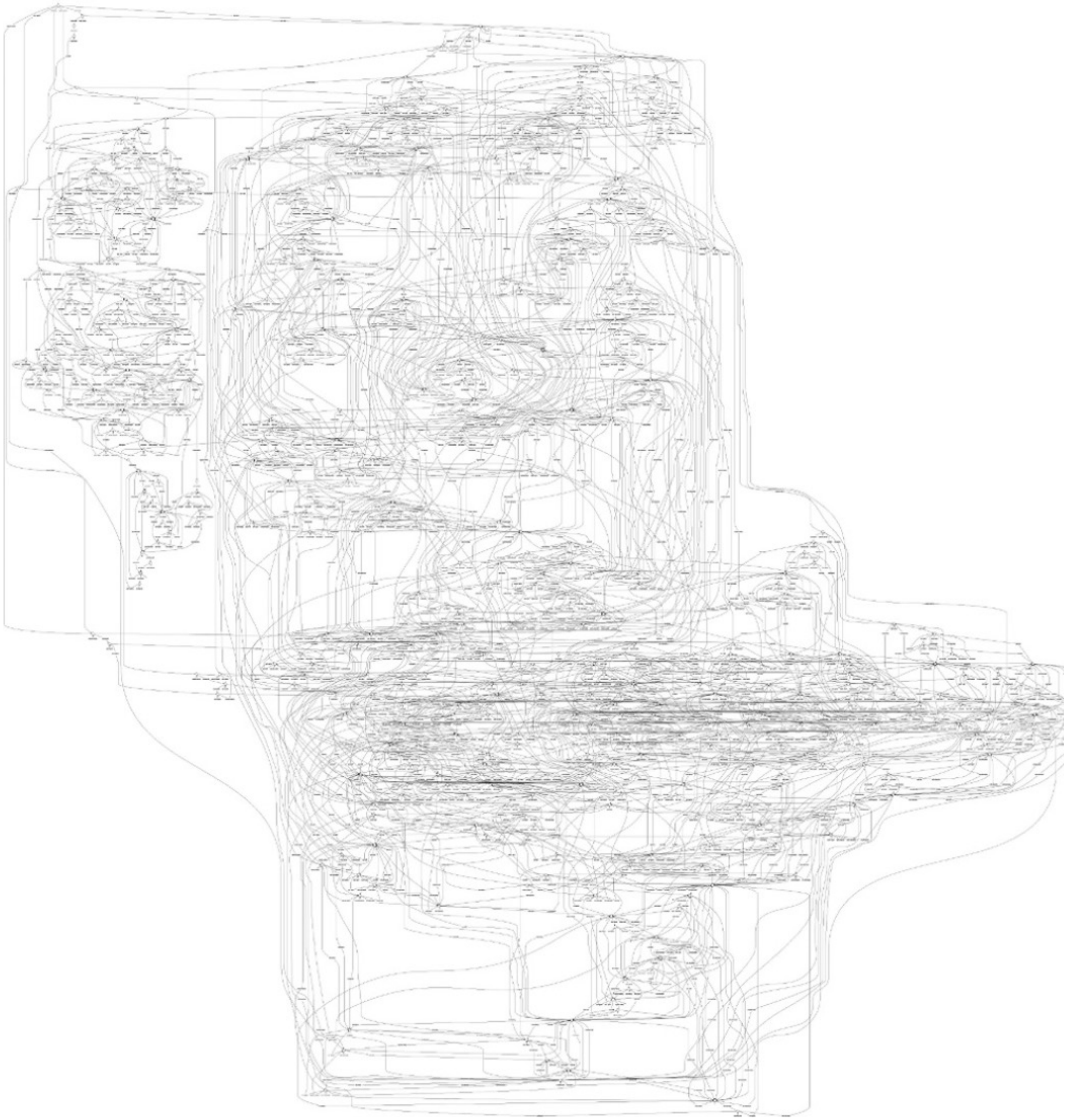


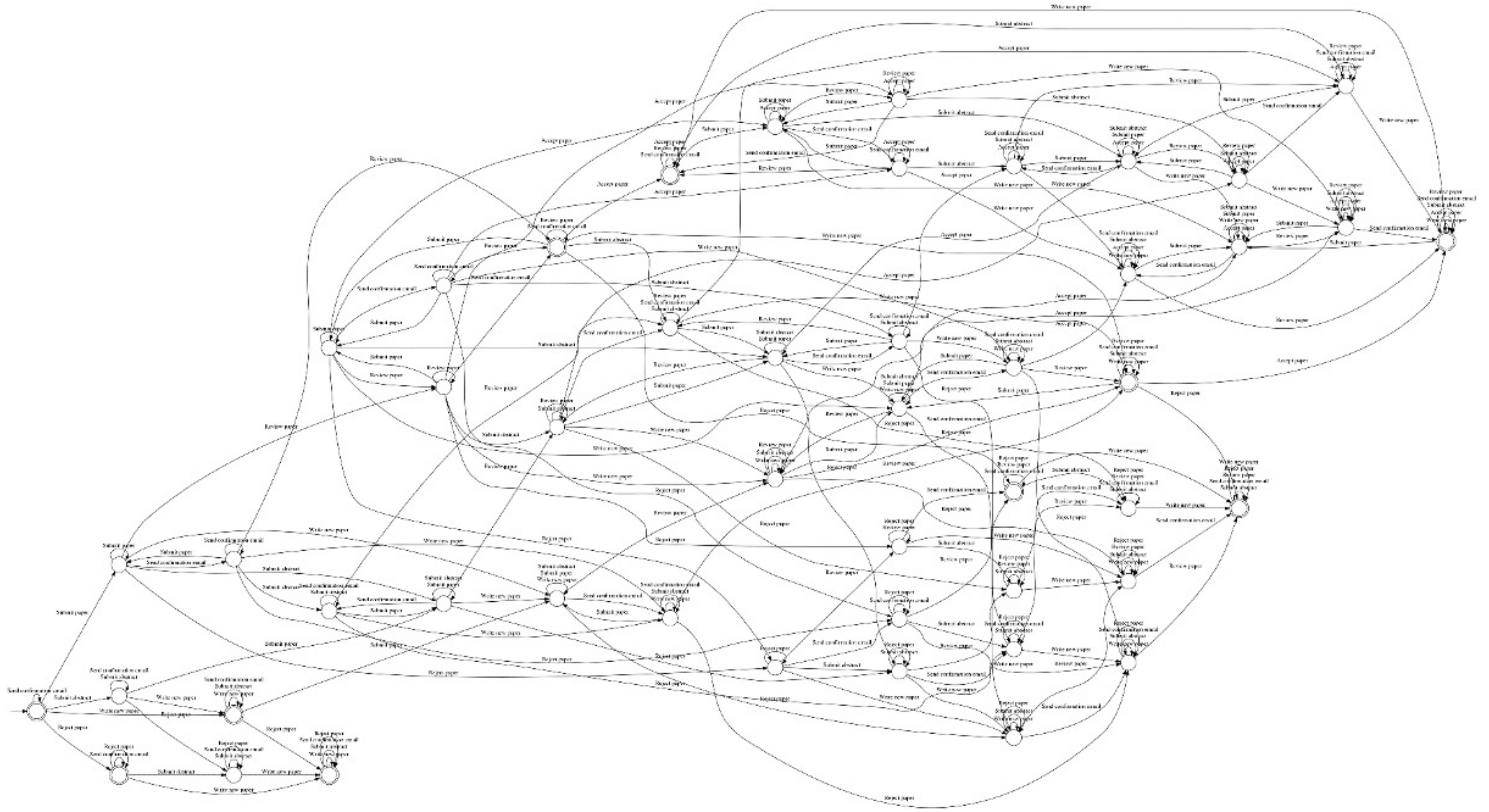
Generalisation



Generalisation

Flexibility is everywhere





Simplicity cannot be obtained by sweeping complexity under the carpet

Our goal

represents

Compact specification

Reality

“Framing” via declarative specifications

process

procedural specification

declarative specification

"Framing" via declarative specifications

process

procedural specification

declarative specification

“Framing” via declarative specifications

process

procedural specification

declarative specification

“Framing” via declarative specifications

process

procedural specification

declarative specification

The "Declare" approach

[Pesic et al, EDOC 2007] [___,Springer Book 2010] [___,TWEB 2011]

A set of temporal constraints

templates grounded on the activities of interest

- Constraints have to be **all satisfied** over a **complete execution trace**
- Compositional approach by **conjunction**

Which constraints are useful?

Patterns in Property Specifications for Finite-State Verification*

Matthew B. Dwyer
Kansas State University
Department of Computing and Information Sciences
Manhattan, KS 66506-2302
+1 785 532 6350
dwyer@cis.ksu.edu

George S. Avrunin
University of Massachusetts
Department of Mathematics and Statistics
Amherst, MA 01003-4515
+1 413 545 4251
avrunin@math.umass.edu

James C. Corbett
University of Hawai'i
Department of Information and Computer Science
Honolulu, HI 96822
+1 808 956 6107
corbett@hawaii.edu

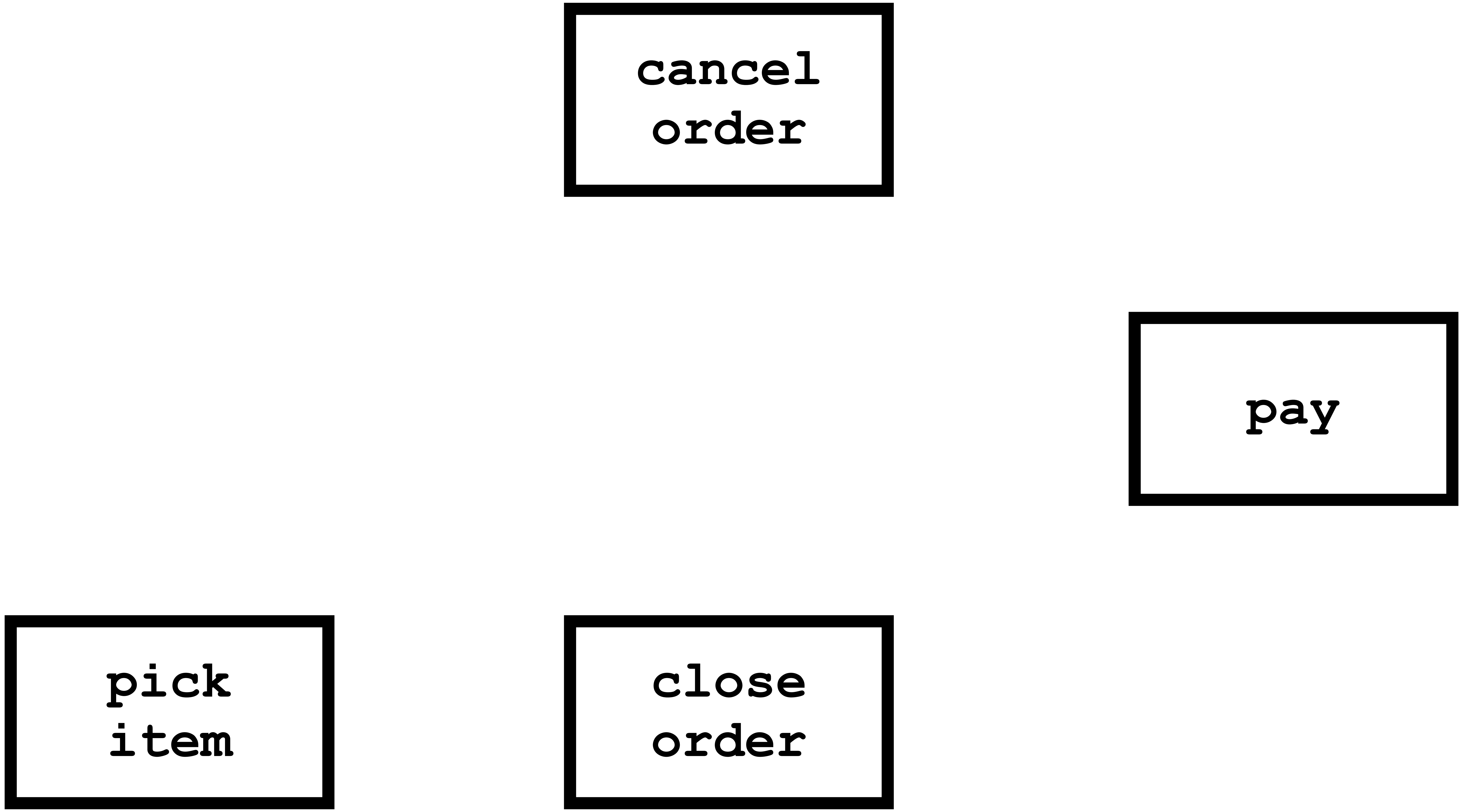
ABSTRACT

Model checkers and other finite-state verification tools allow developers to detect certain kinds of errors automatically. Nevertheless, the transition of this technology from research to practice has been slow. While there are a number of potential causes for reluctance to adopt such formal methods, we believe that a primary cause is that practitioners are unfamiliar with specification processes, notations, and strategies. In a recent paper, we proposed a pattern-based approach to the presentation, codification and reuse of property specifications for finite-state verification. Since then, we have carried out a survey of available specifications, collecting over 500 examples of property specifications. We found that most are instances of our proposed patterns. Furthermore, we have updated our pattern system to accommodate new patterns and variations of existing patterns encountered in this survey. This paper reports the results of the survey and the current status of our pattern system.

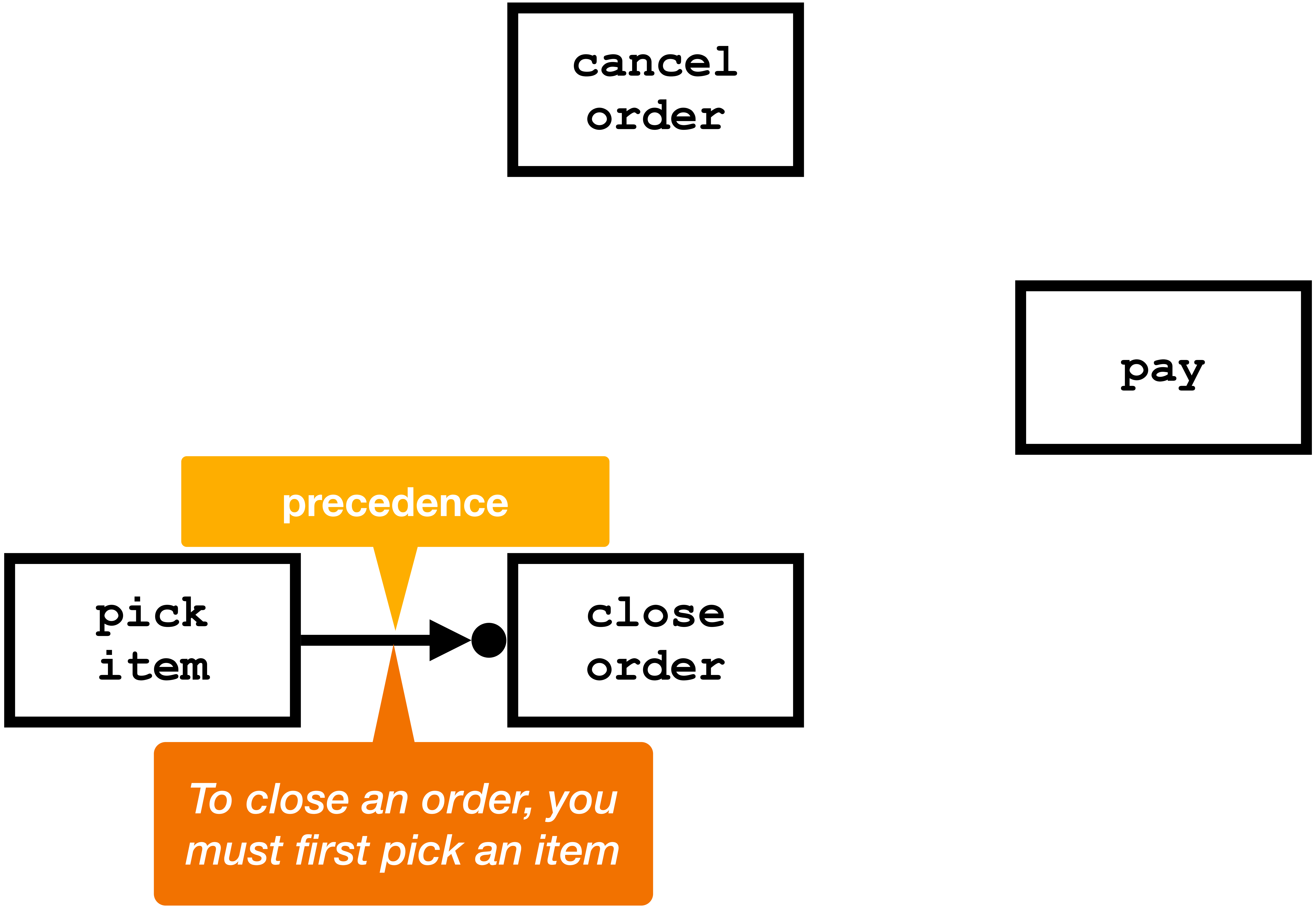
cess support for formal methods.

We believe that the recent availability of tool support for finite-state verification provides an opportunity to overcome some of these barriers. Finite-state verification refers to a set of techniques for proving properties of finite-state models of computer systems. Properties are typically specified with temporal logics or regular expressions, while systems are specified as finite-state transition systems of some kind. Tool support is available for a variety of verification techniques including, for example, techniques based on model checking [19], bisimulation [4], language containment [14], flow analysis [10], and inequality necessary conditions [1]. In contrast to mechanical theorem proving, which often requires guidance by an expert, most finite-state verification techniques can be fully automated, relieving the user of the need to understand the inner workings of the verification process. Finite-state verification techniques are especially critical in the development of concurrent

Example

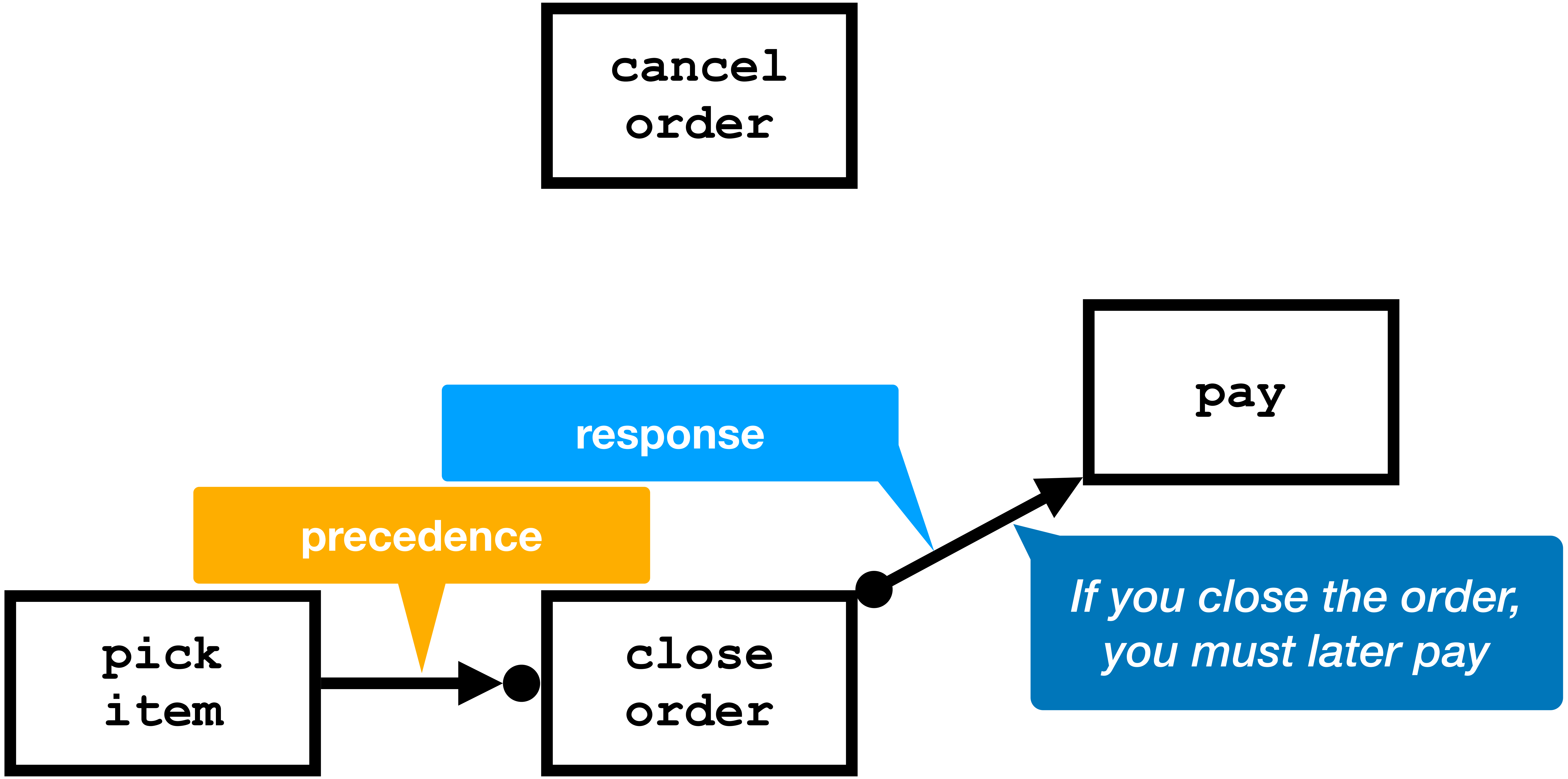


Example

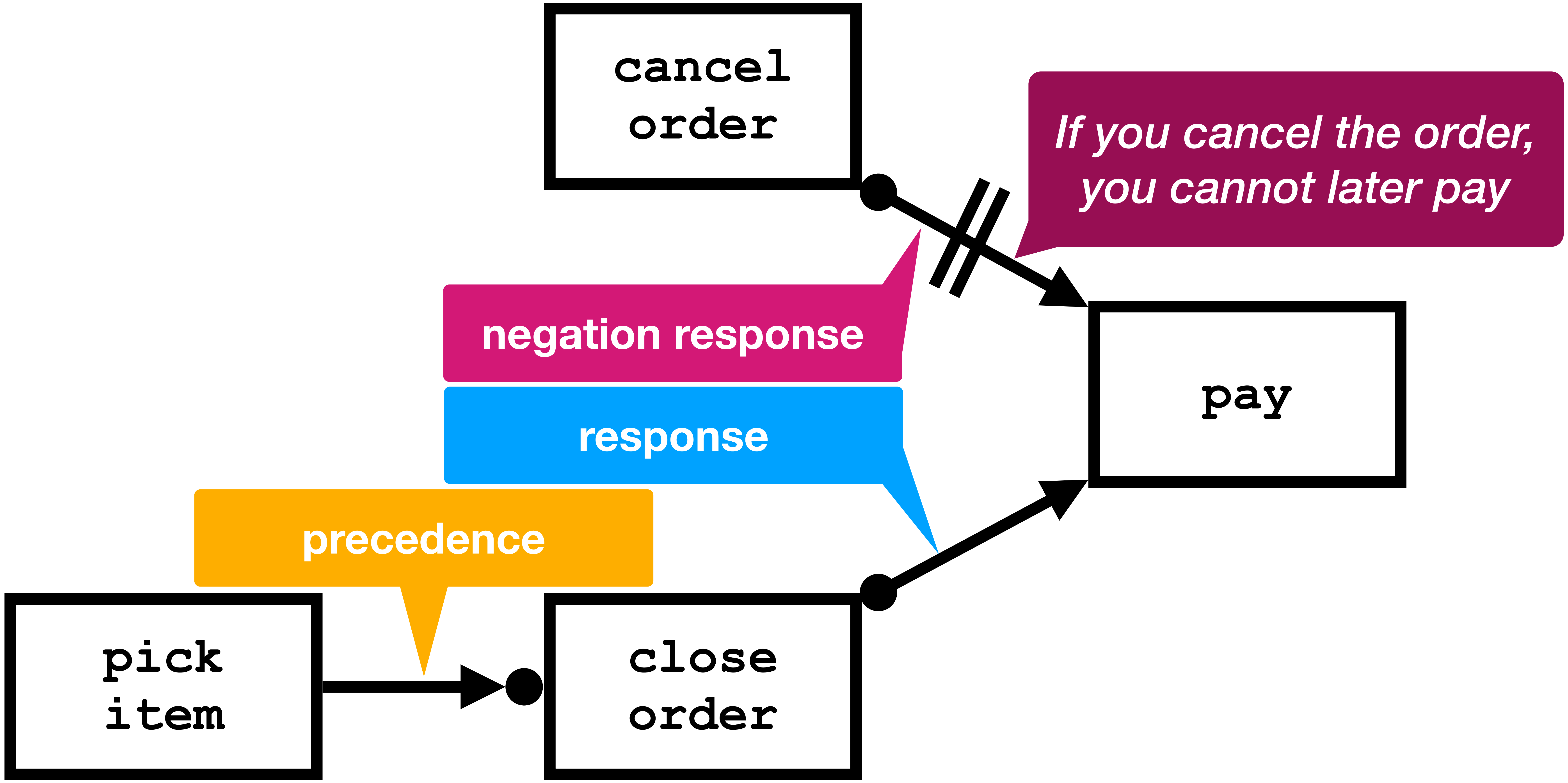


cancel order
pay
precedence
pick item
close order
To close an order, you must first pick an item

Example

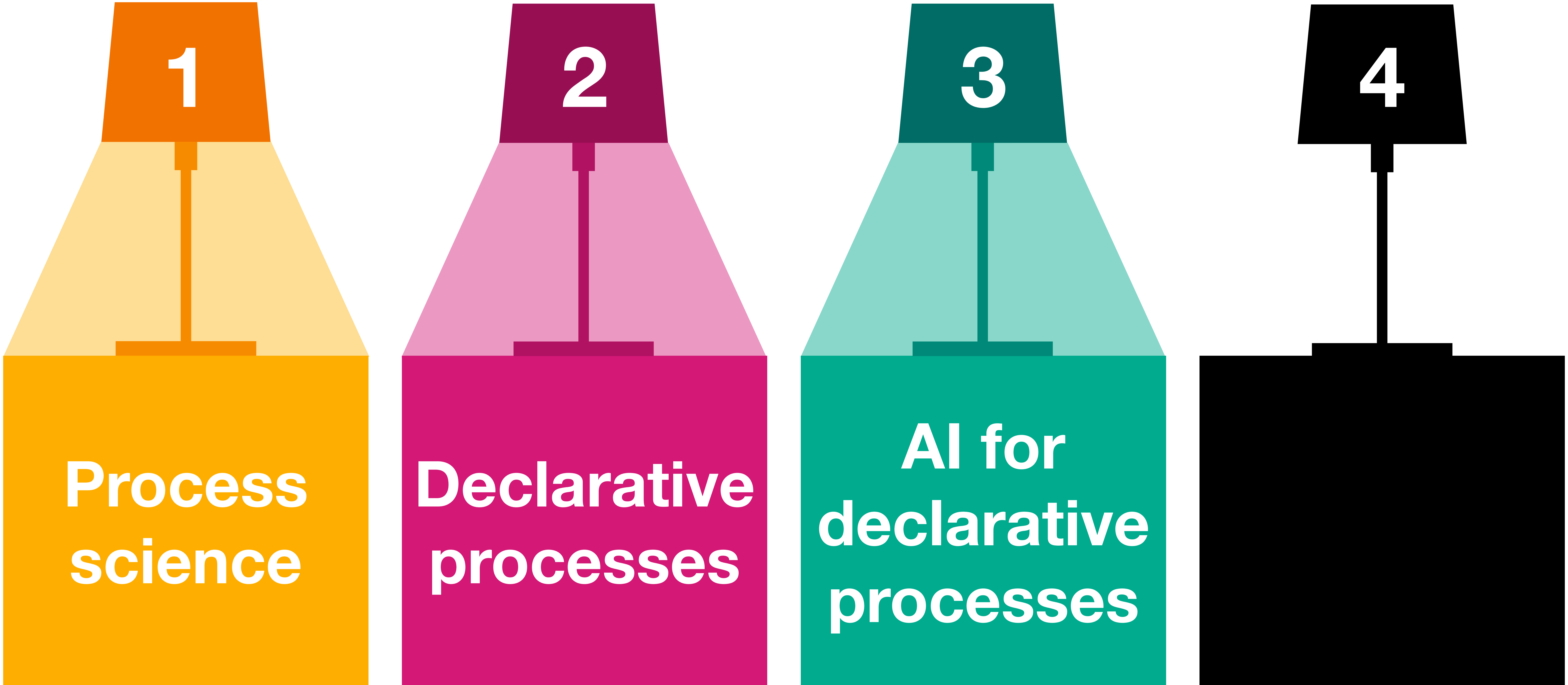

cancel
order
response
pay
precedence
pick
item
close
order
If you close the order,
you must later pay

Example



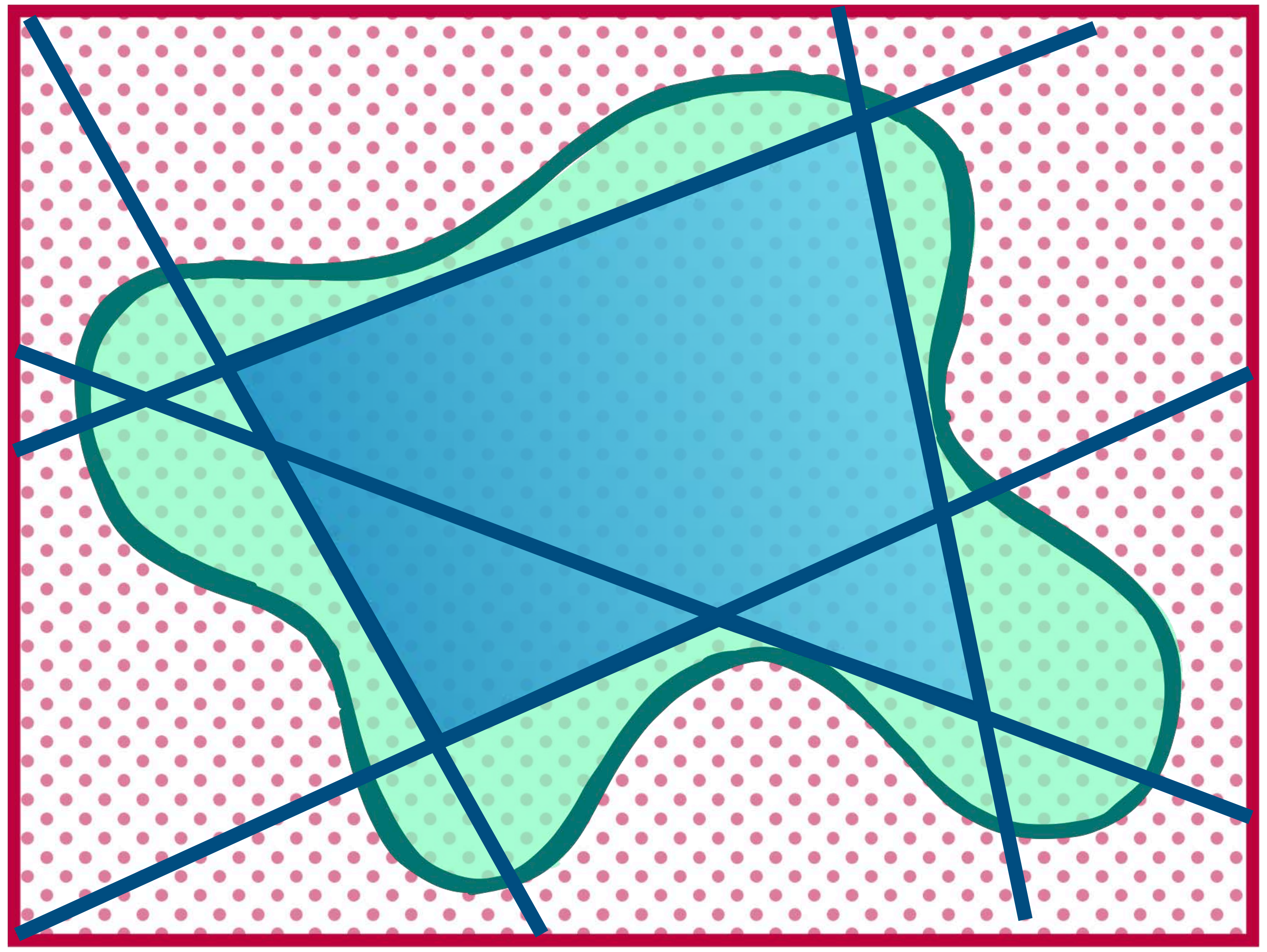
cancel order
If you cancel the order, you cannot later pay
negation response
pay
response
precedence
pick item
close order

3/4


1
2
3
4
Process science
Declarative processes
AI for declarative processes

Representation question

How to characterise the set of traces in the specification?



Declare specification

Which constraints are useful?

Patterns in Property Specifications for Finite-State Verification*

Matthew B. Dwyer
Kansas State University
Department of Computing and Information Sciences
Manhattan, KS 66506-2302
+1 785 532 6350
dwyer@cis.ksu.edu

George S. Avrunin
University of Massachusetts
Department of Mathematics and Statistics
Amherst, MA 01003-4515
+1 413 545 4251
avrunin@math.umass.edu

James C. Corbett
University of Hawai'i
Department of Information and Computer Science
Honolulu, HI 96822
+1 808 956 6107
corbett@hawaii.edu

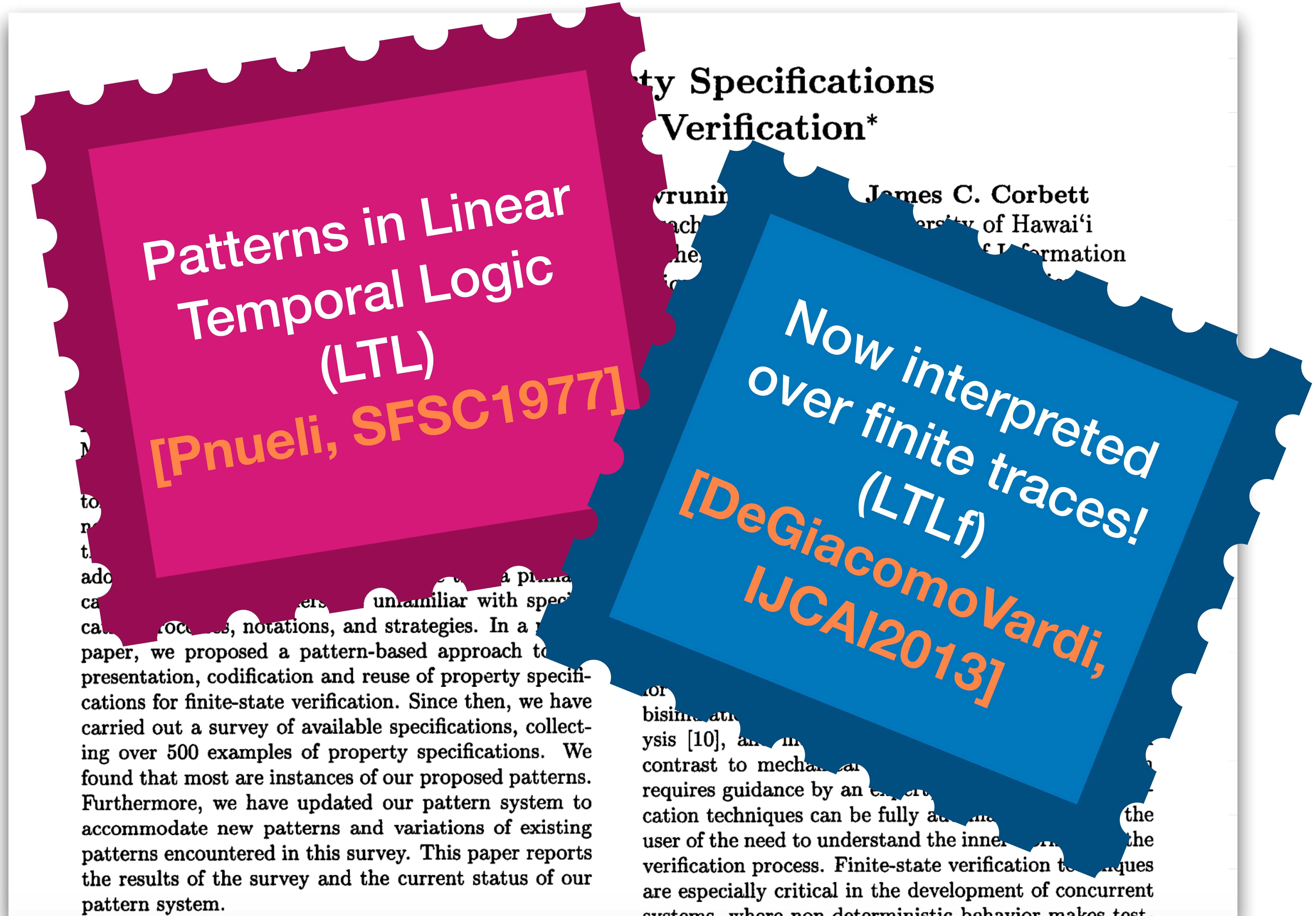
ABSTRACT

Model checkers and other finite-state verification tools allow developers to detect certain kinds of errors automatically. Nevertheless, the transition of this technology from research to practice has been slow. While there are a number of potential causes for reluctance to adopt such formal methods, we believe that a primary cause is that practitioners are unfamiliar with specification processes, notations, and strategies. In a recent paper, we proposed a pattern-based approach to the presentation, codification and reuse of property specifications for finite-state verification. Since then, we have carried out a survey of available specifications, collecting over 500 examples of property specifications. We found that most are instances of our proposed patterns. Furthermore, we have updated our pattern system to accommodate new patterns and variations of existing patterns encountered in this survey. This paper reports the results of the survey and the current status of our pattern system.

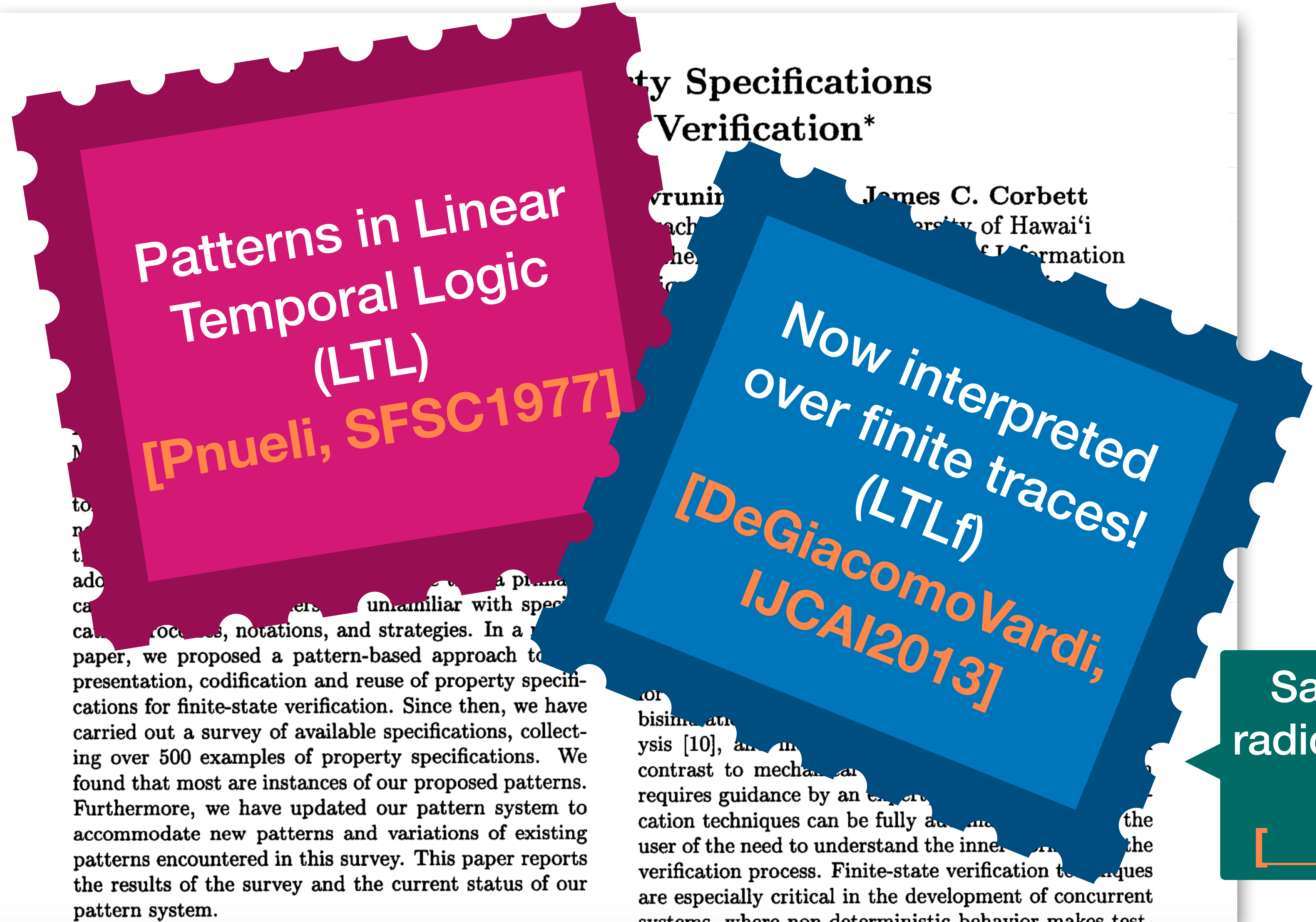
cess support for formal methods.

We believe that the recent availability of tool support for finite-state verification provides an opportunity to overcome some of these barriers. Finite-state verification refers to a set of techniques for proving properties of finite-state models of computer systems. Properties are typically specified with temporal logics or regular expressions, while systems are specified as finite-state transition systems of some kind. Tool support is available for a variety of verification techniques including, for example, techniques based on model checking [19], bisimulation [4], language containment [14], flow analysis [10], and inequality necessary conditions [1]. In contrast to mechanical theorem proving, which often requires guidance by an expert, most finite-state verification techniques can be fully automated, relieving the user of the need to understand the inner workings of the verification process. Finite-state verification techniques are especially critical in the development of concurrent

Which constraints are useful?



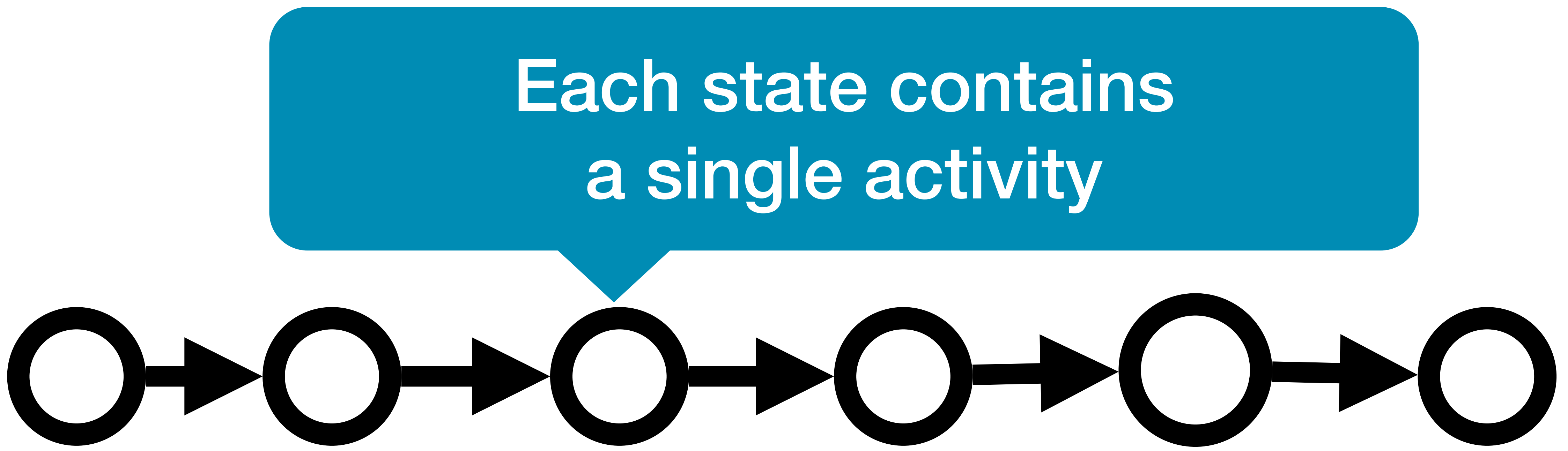
Which constraints are useful?



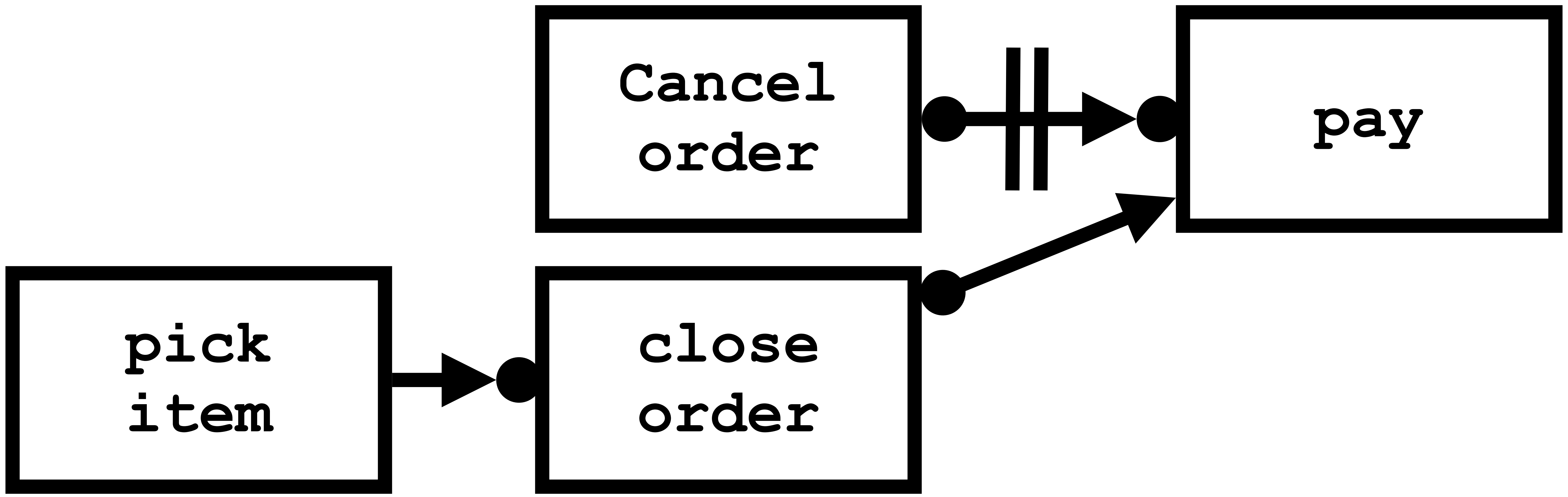
Same syntax, radically different meaning
[___,AAAI2014]

Semantics of Declare via LTLf

Atomic **propositions** are **activities**

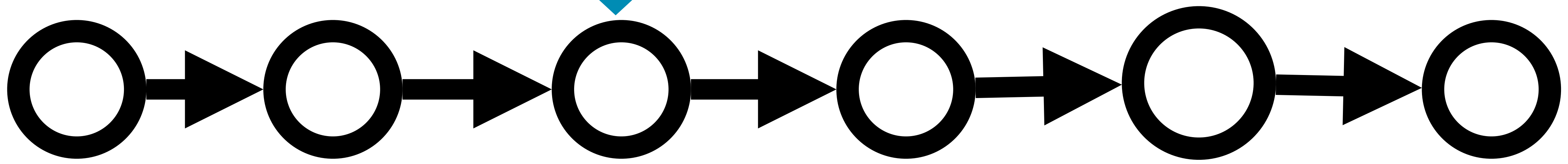


Each **constraint** is an **LTLf formula** (built from template formulae)



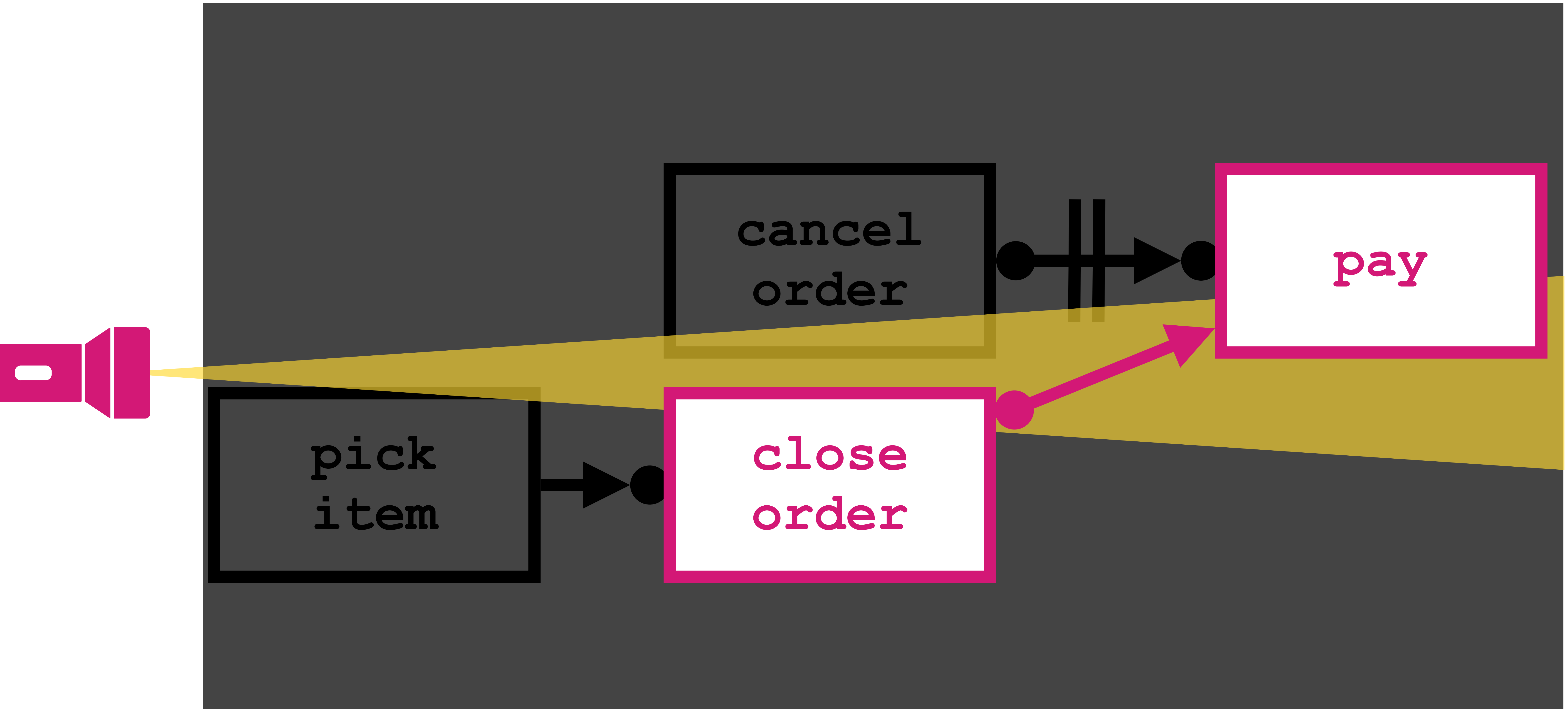
Semantics of Declare via LTLf

Each state contains a single activity



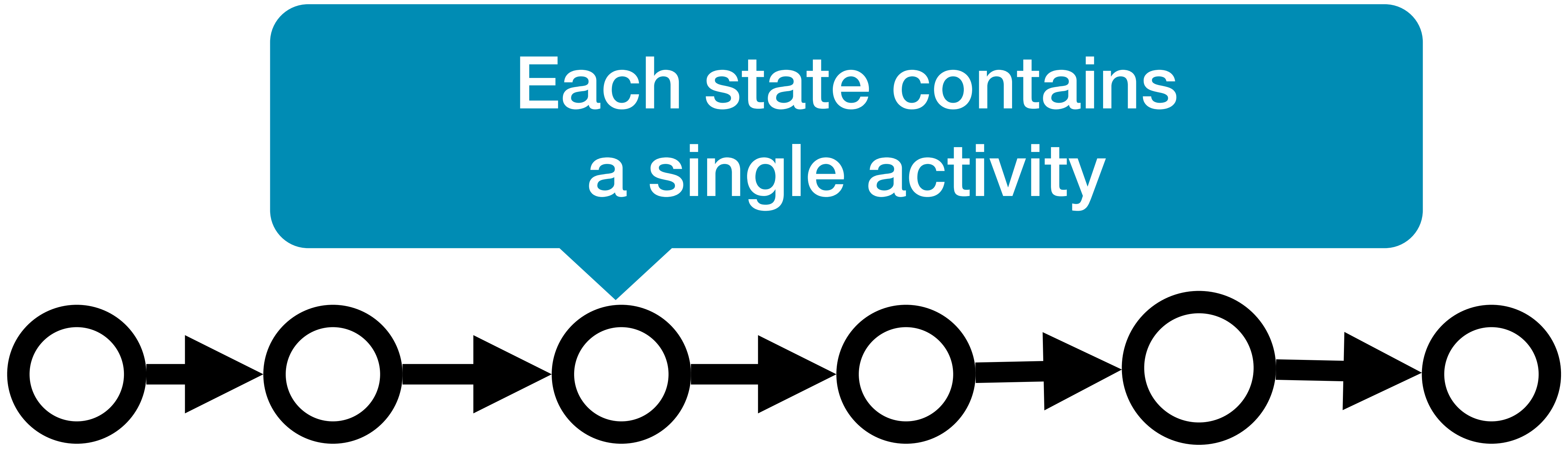
Atomic **propositions** are **activities**

Each **constraint** is an **LTLf formula** (built from template formulae)



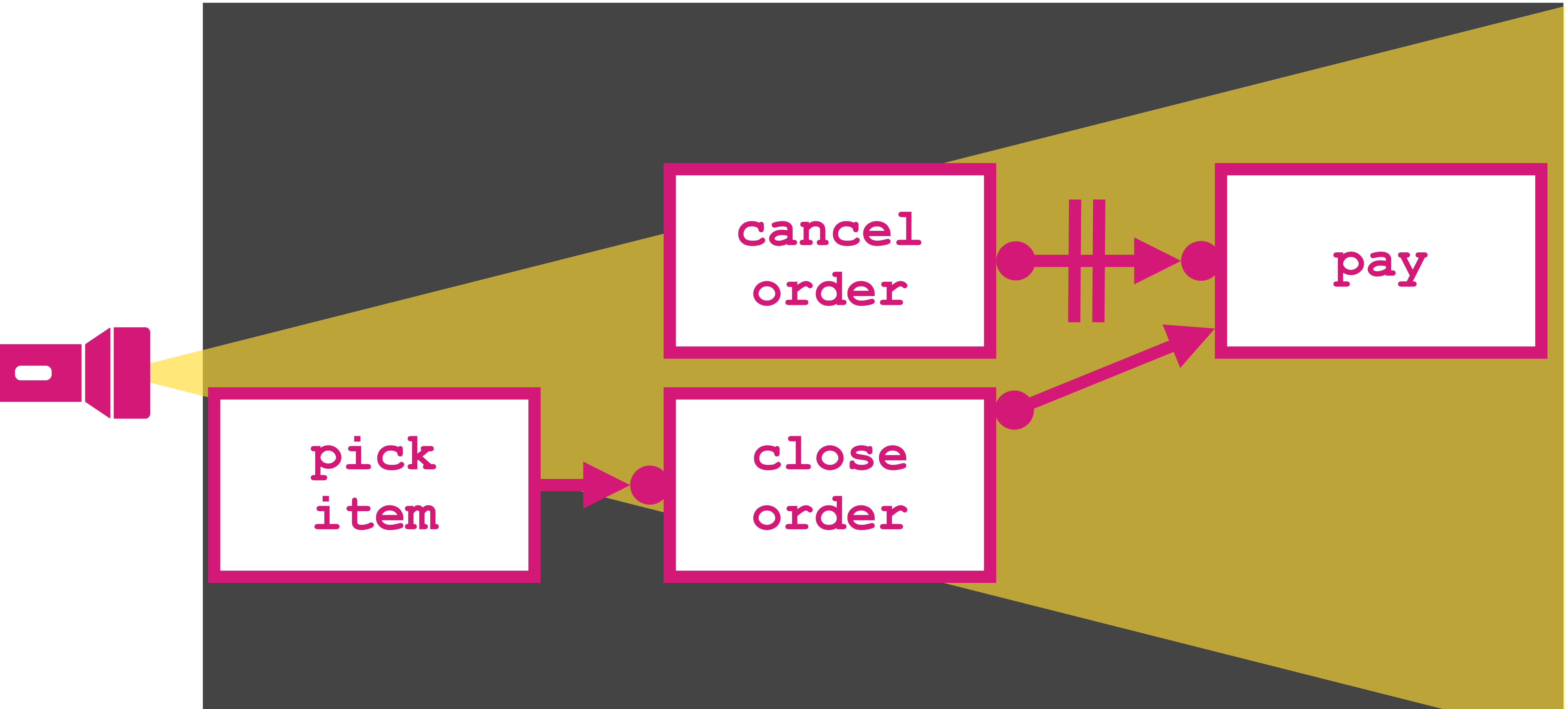
$\Box(\mathrm{close} \rightarrow \Diamond \mathrm{pay})$

Semantics of Declare via LTLf



Atomic **propositions** are **activities**

A **Declare specification** is the **conjunction** of its **constraint formulae**



$$\square(\mathtt{close} \rightarrow \lozenge \mathtt{pay}) \wedge (\neg \mathtt{close}) \mathscr{W} \mathtt{item} \wedge \square(\mathtt{cancel} \rightarrow \neg \square \mathtt{pay})$$

Semantics of Declare via LTLf

…ch state contains
…a single activity

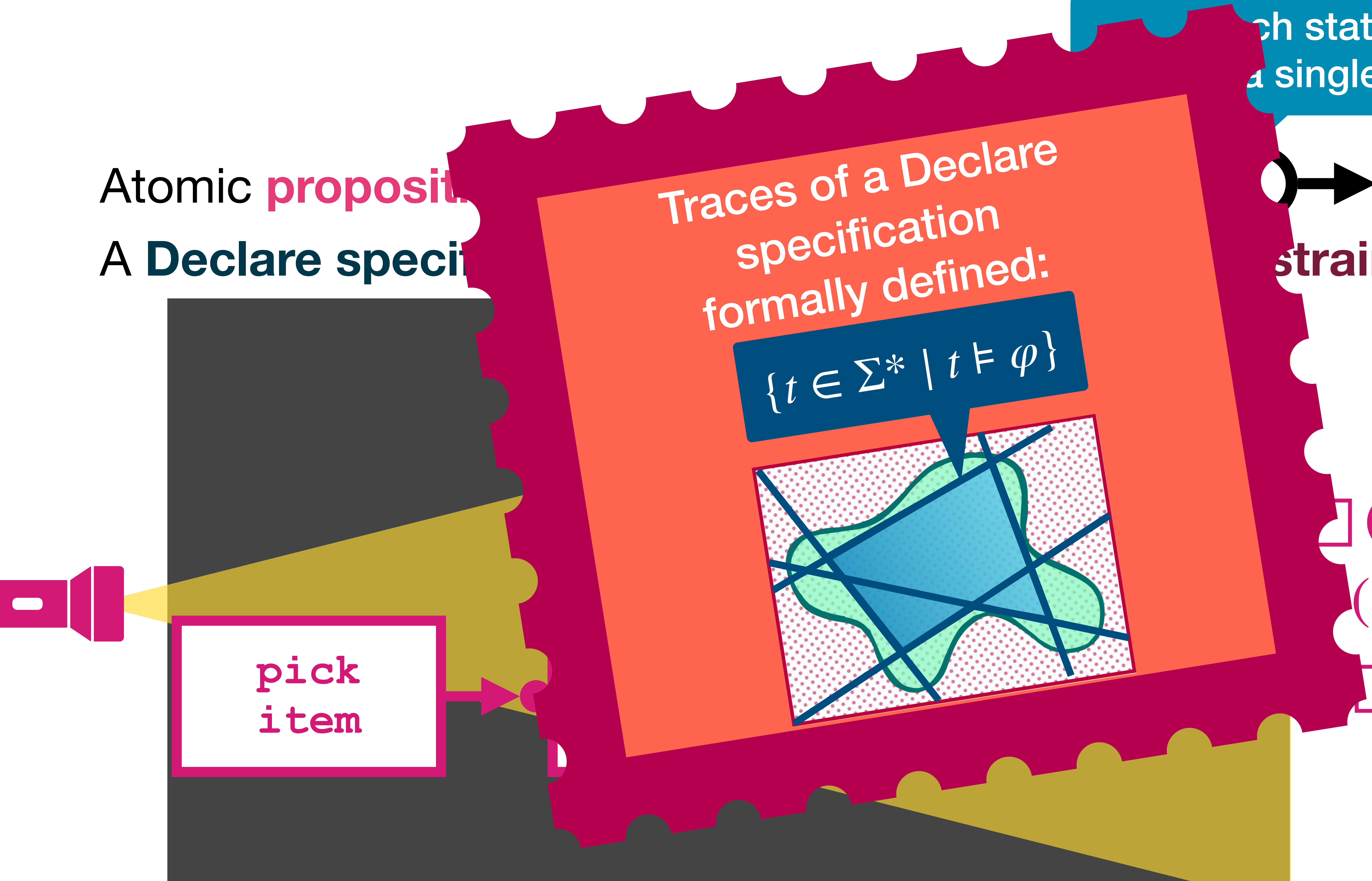
Atomic proposit…

A **Declare speci…** …straint formulae

$\square(\mathtt{close} \rightarrow \Diamond \mathtt{pay})$

$(\neg \mathtt{close}) \mathcal{W} \mathtt{item}$

$\square(\mathtt{cancel} \rightarrow \neg \square \mathtt{pay})$



An unconventional use of logics

(In process science)

From …

Temporal logics for **specifying** (un)desired **properties of a dynamic system**

… to …

Temporal logics for **specifying the dynamic system itself**

AI for Declare/LTLf-based processes

An incomplete picture…

AI for Declare/LTLf-based processes

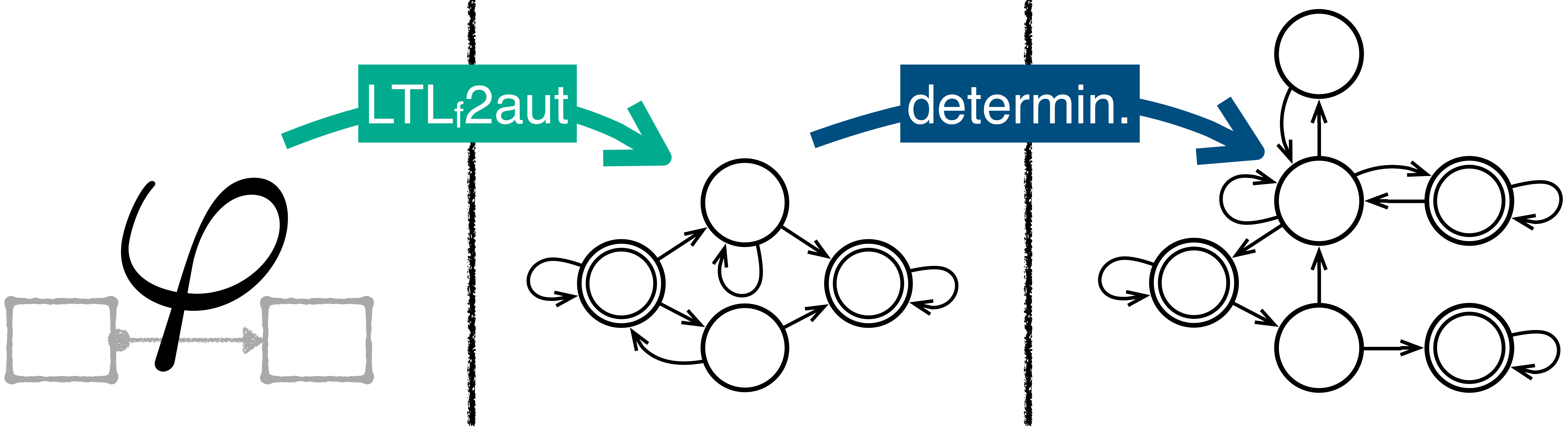
An incomplete picture…

From LTLf/Declare to automata

[DeGiacomoVardi,IJCAI2013] [____,TOSEM2022]

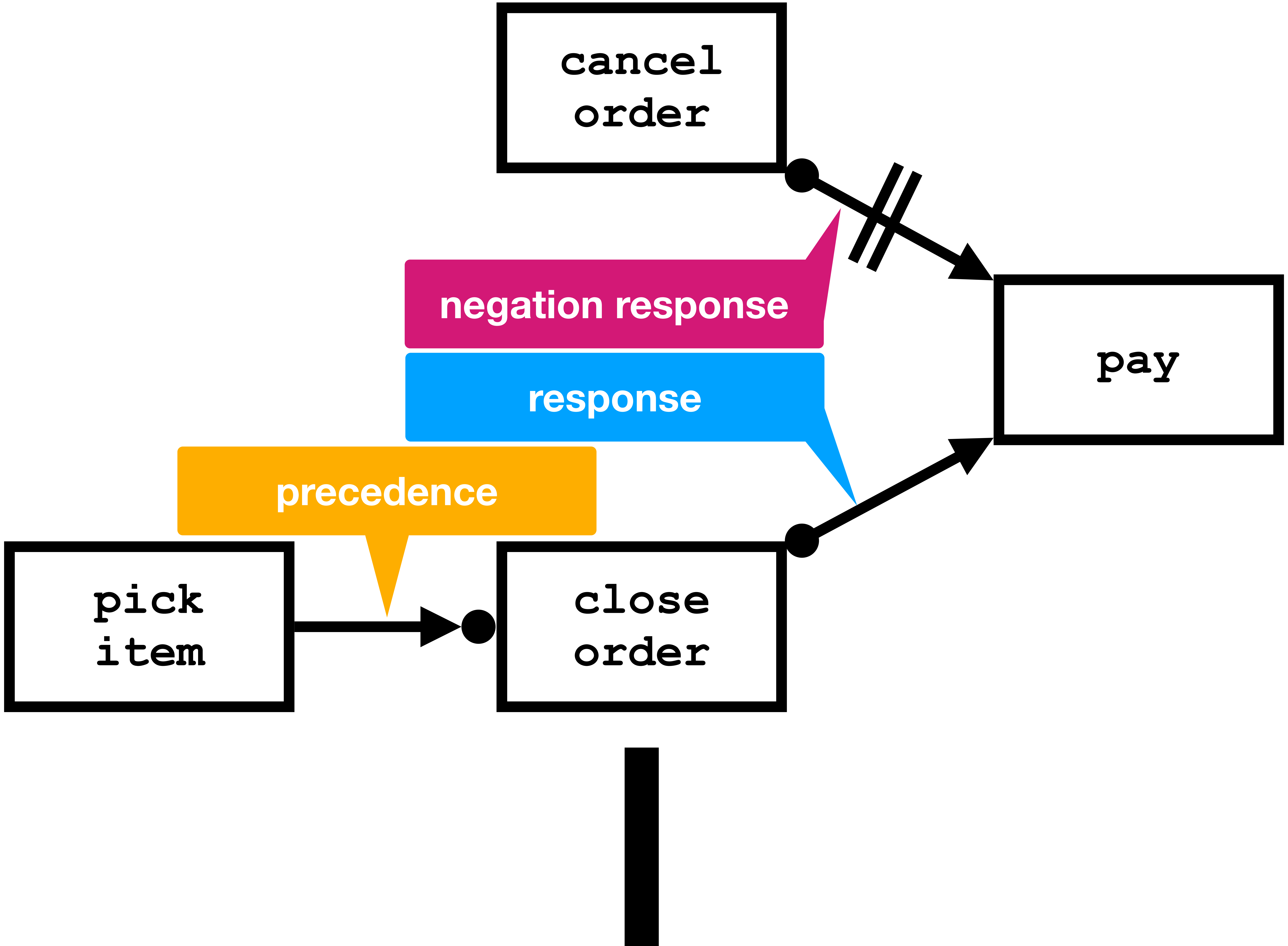
$$t \models \varphi \text{ iff } t \in \mathcal{L}(\mathcal{A}_\varphi)$$

LTL_f	NFA	DFA
	nondeterministic	deterministic



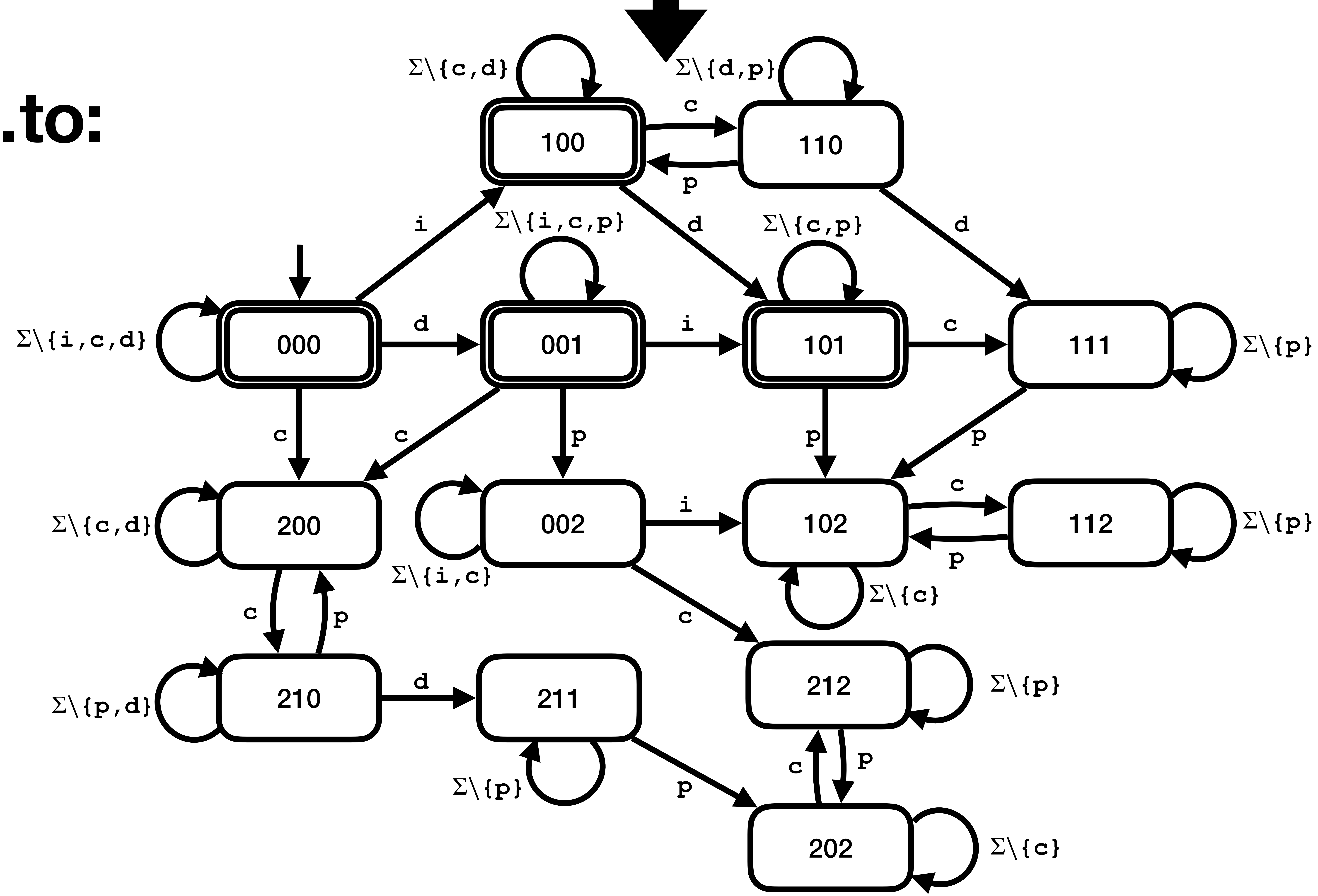
Thanks to finite traces: good old finite-state automata

From...



cancel order
negation response
pay
response
precedence
pick item
close order

...to:



Σ\{c,d}
100
c
p
110
Σ\{d,p}
i
Σ\{i,c,p}
d
Σ\{c,p}
d
Σ\{i,c,d}
000
d
001
i
101
c
111
Σ\{p}
c
c
p
p
p
Σ\{c,d}
200
002
i
102
c
p
112
Σ\{p}
Σ\{i,c}
Σ\{c}
c
p
c
Σ\{p,d}
210
d
211
212
Σ\{p}
Σ\{p}
p
c
p
202
Σ\{c}

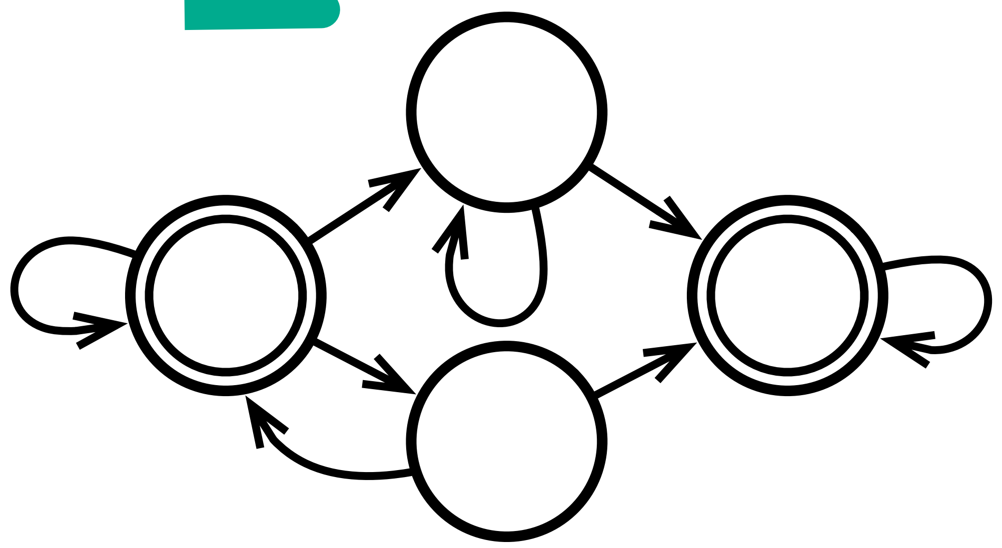
Vision realised!

$$t \models \varphi \text{ iff } t \in \mathcal{L}(\mathcal{A}_\varphi)$$

LTL$_f$2aut

NFA

nondeterministic



determin.

DFA

deterministic

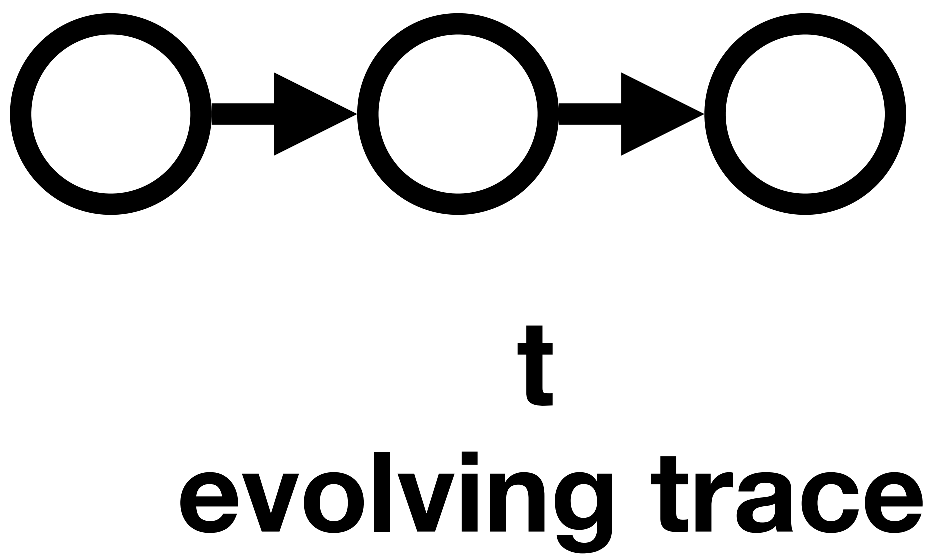
Thanks to finite traces: good old finite-state automata

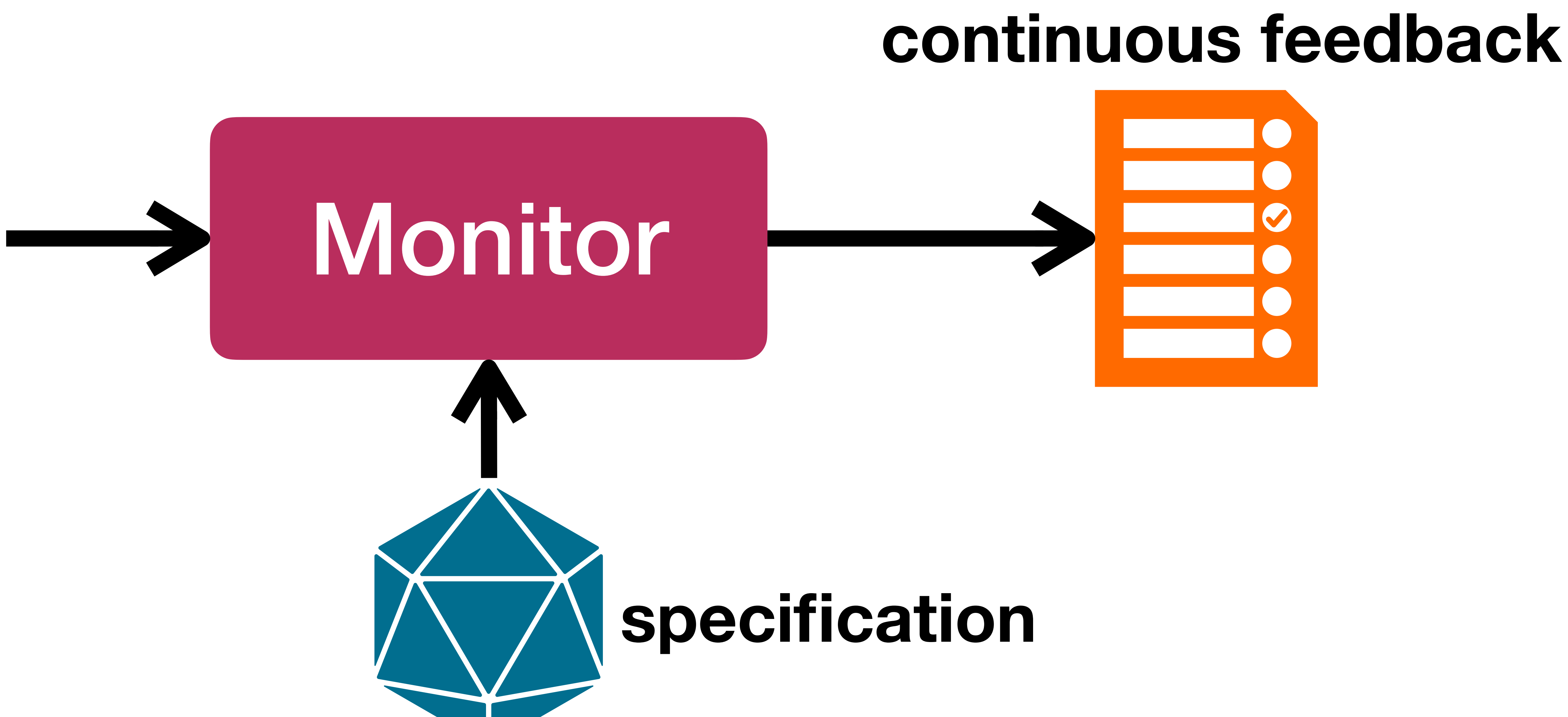
conformance monitoring

Track a **running process execution** to **check conformance** with **properties** of interest

- Goal: Detect and report **fine-grained feedback** and **deviations**

- Complementary to **predictive monitoring!**

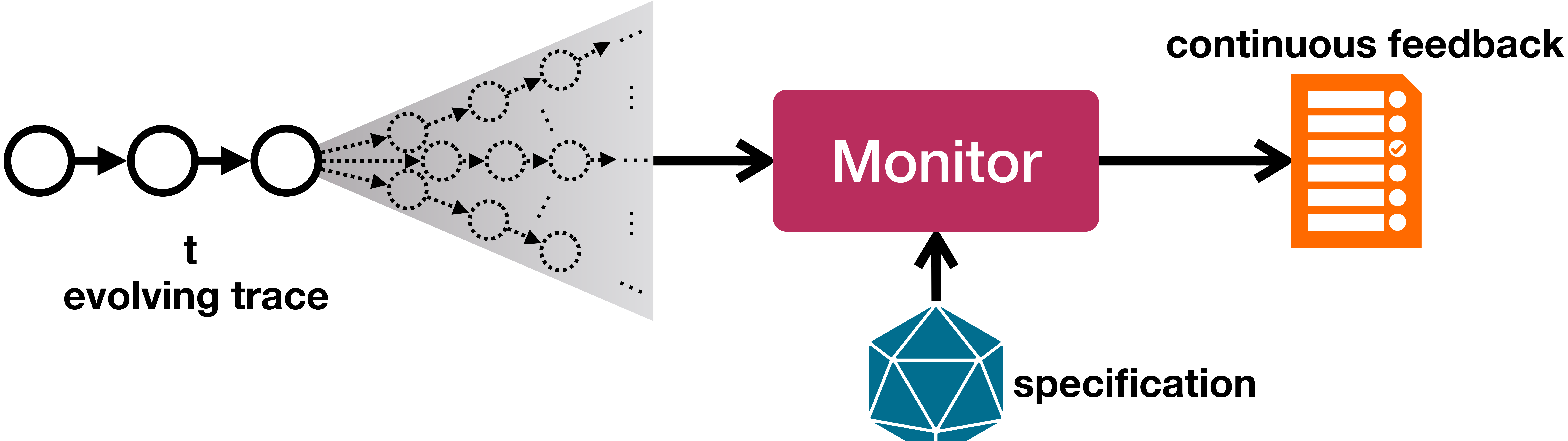




(Anticipatory) conformance monitoring

Track a **running process execution** to **check conformance** with **properties** of interest

- Goal: Detect and report **fine-grained feedback** and **deviations** **also considering the possible future continuations**
- Complementary to **predictive monitoring!**



Fine-grained feedback

Refined analysis of the “truth value” of a property

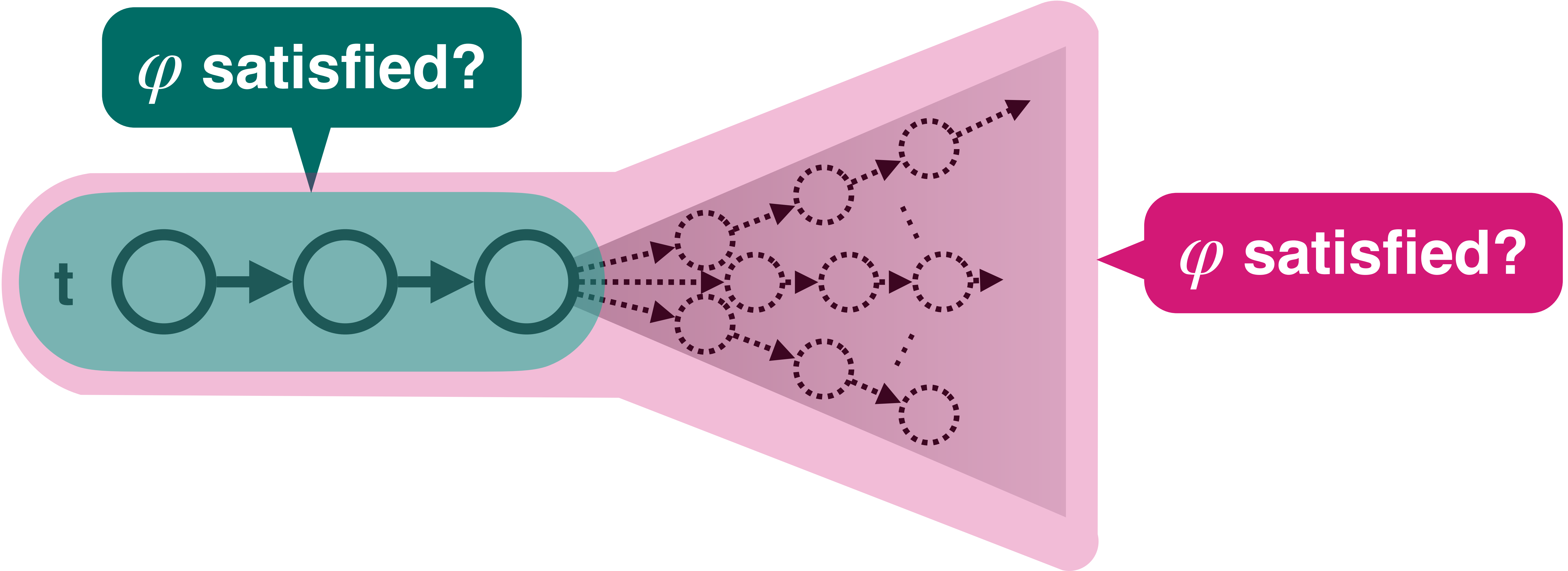
- looking into (all) **possible futures**

Fine-grained feedback

Refined analysis of the “truth value” of a property

- looking into (all) **possible futures**

Consider a partial trace **t**, and an LTLf formula φ…

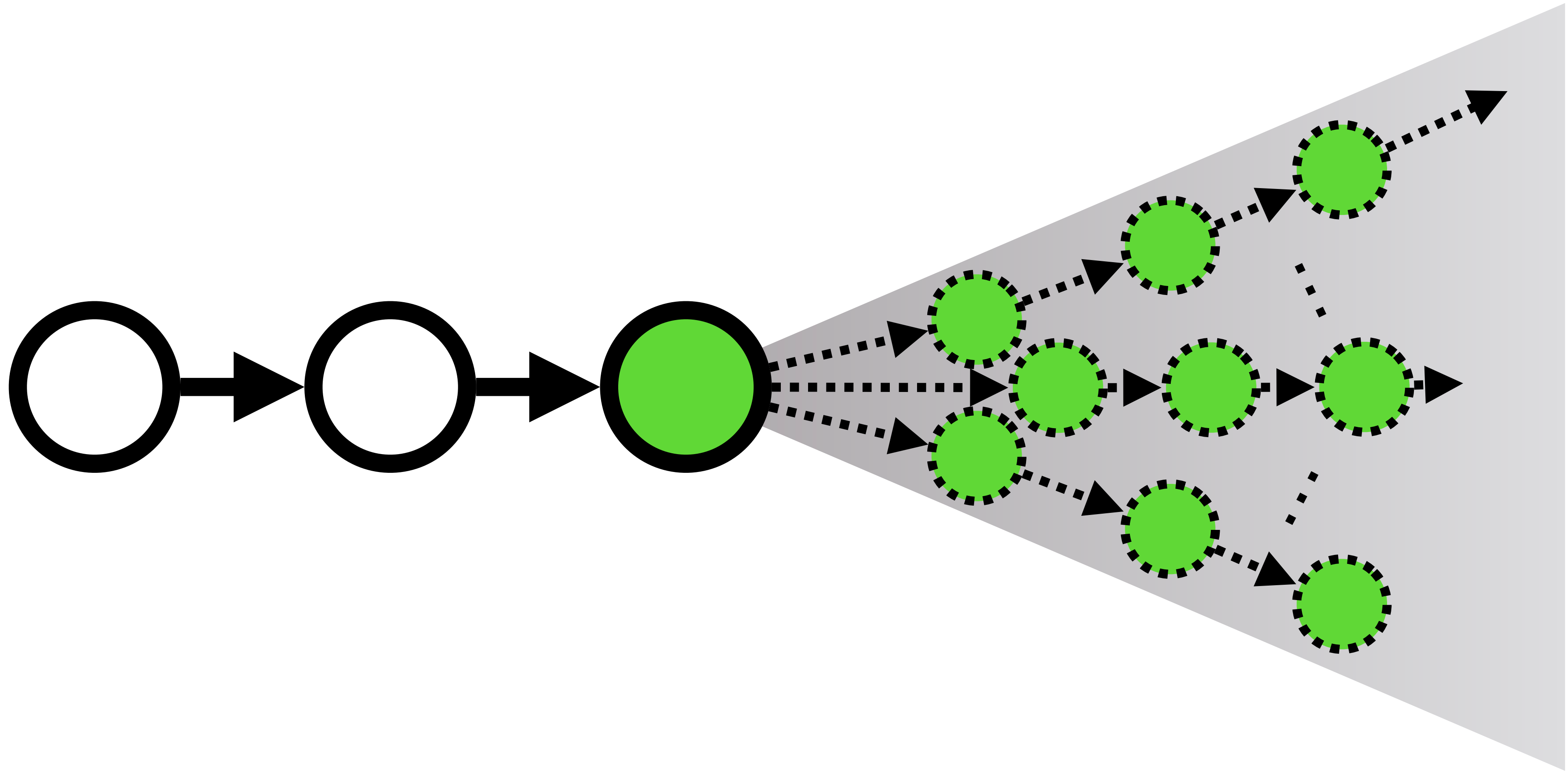


RV-LTL(f) truth values

[BauerEtAl,InfCom2010] [___,BPM2011] [___,TOSEM2022]

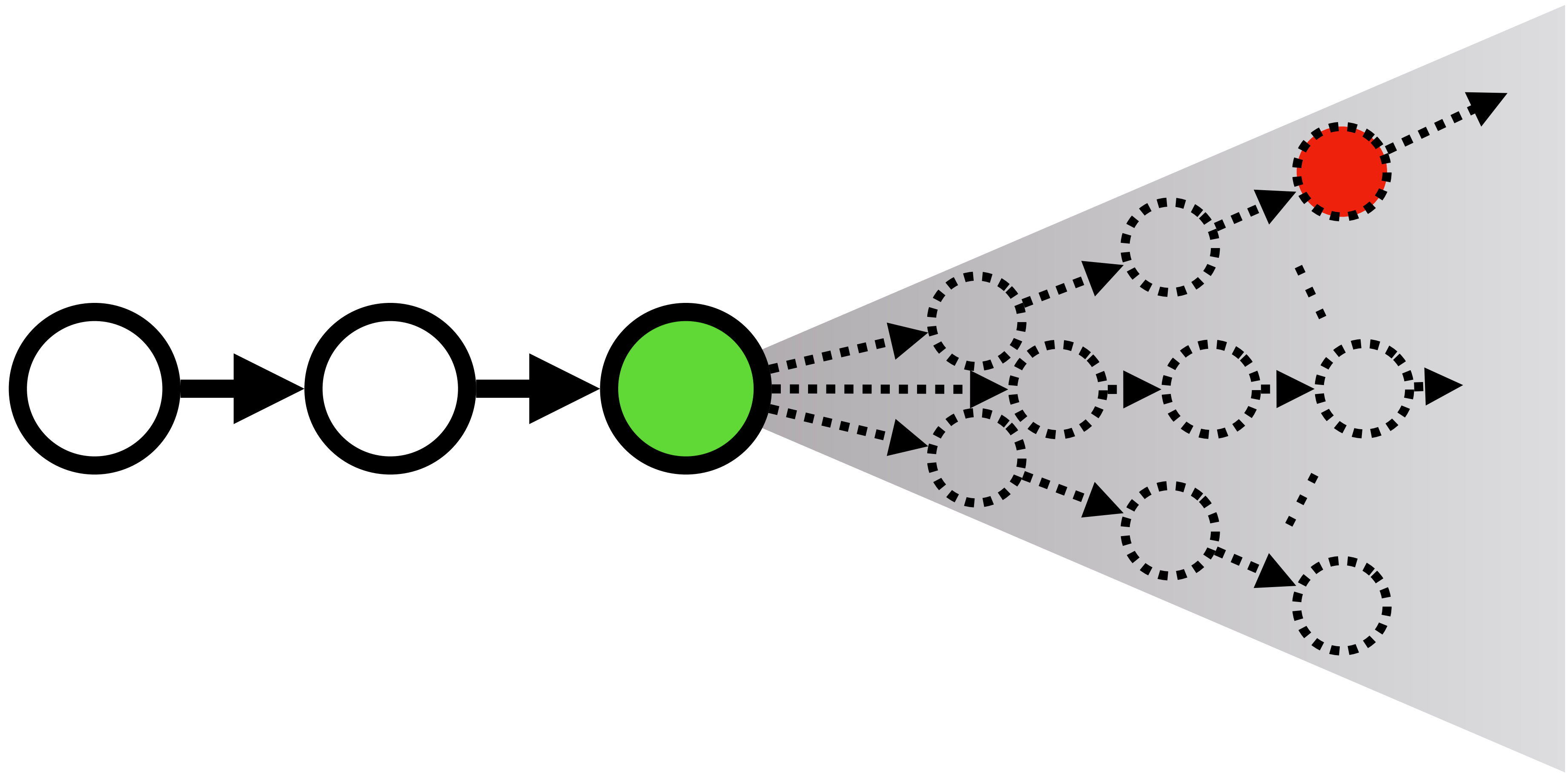
φ **permanently satisfied** by **t**

- **t** satisfies φ
- no matter how **t** continues, φ stays satisfied



φ **currently satisfied** by **t**

- **t** satisfies φ
- there is a continuation of **t** that violates φ

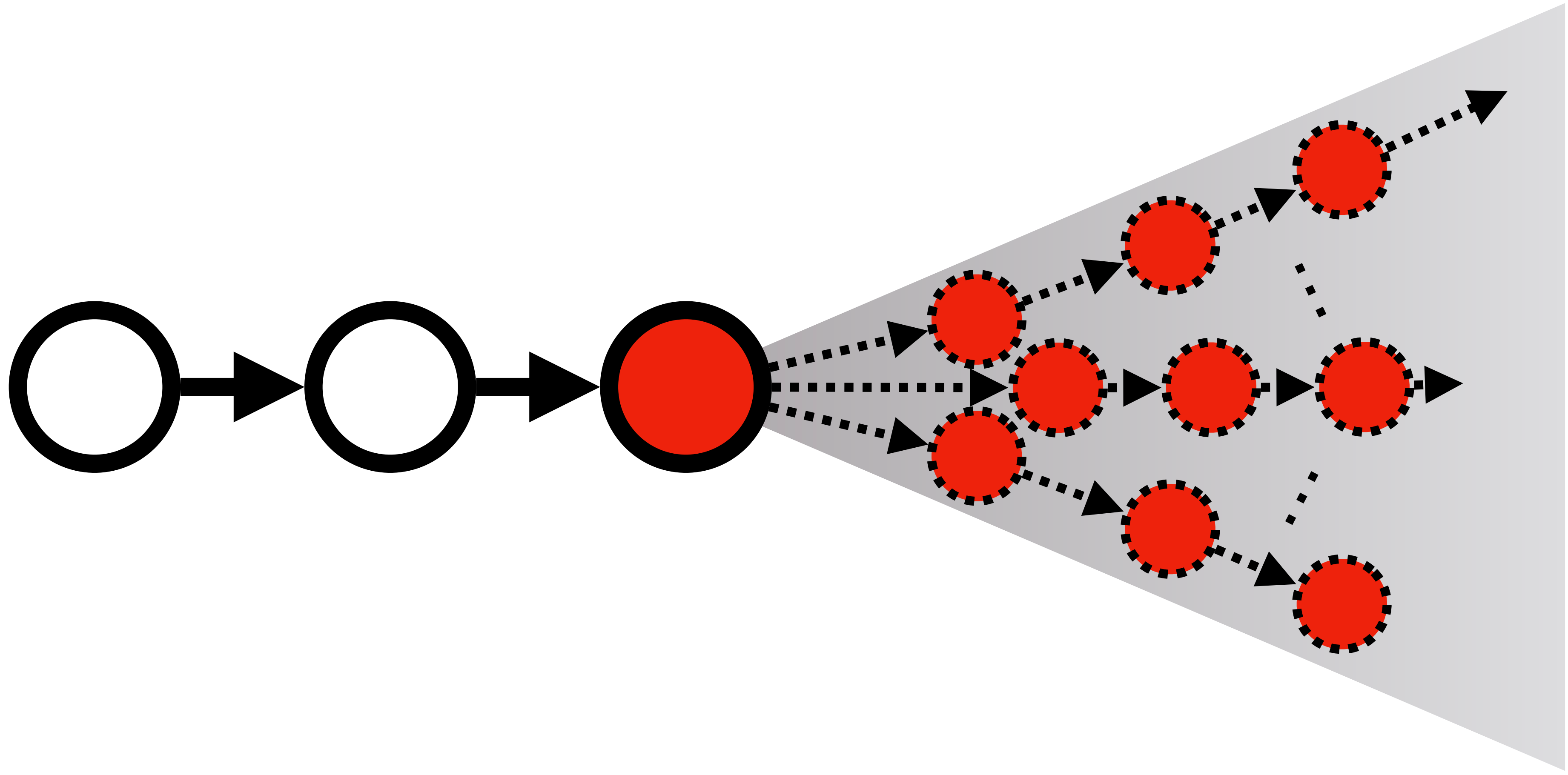


RV-LTL(f) truth values

[BauerEtAl,InfCom2010] [___,BPM2011] [___,TOSEM2022]

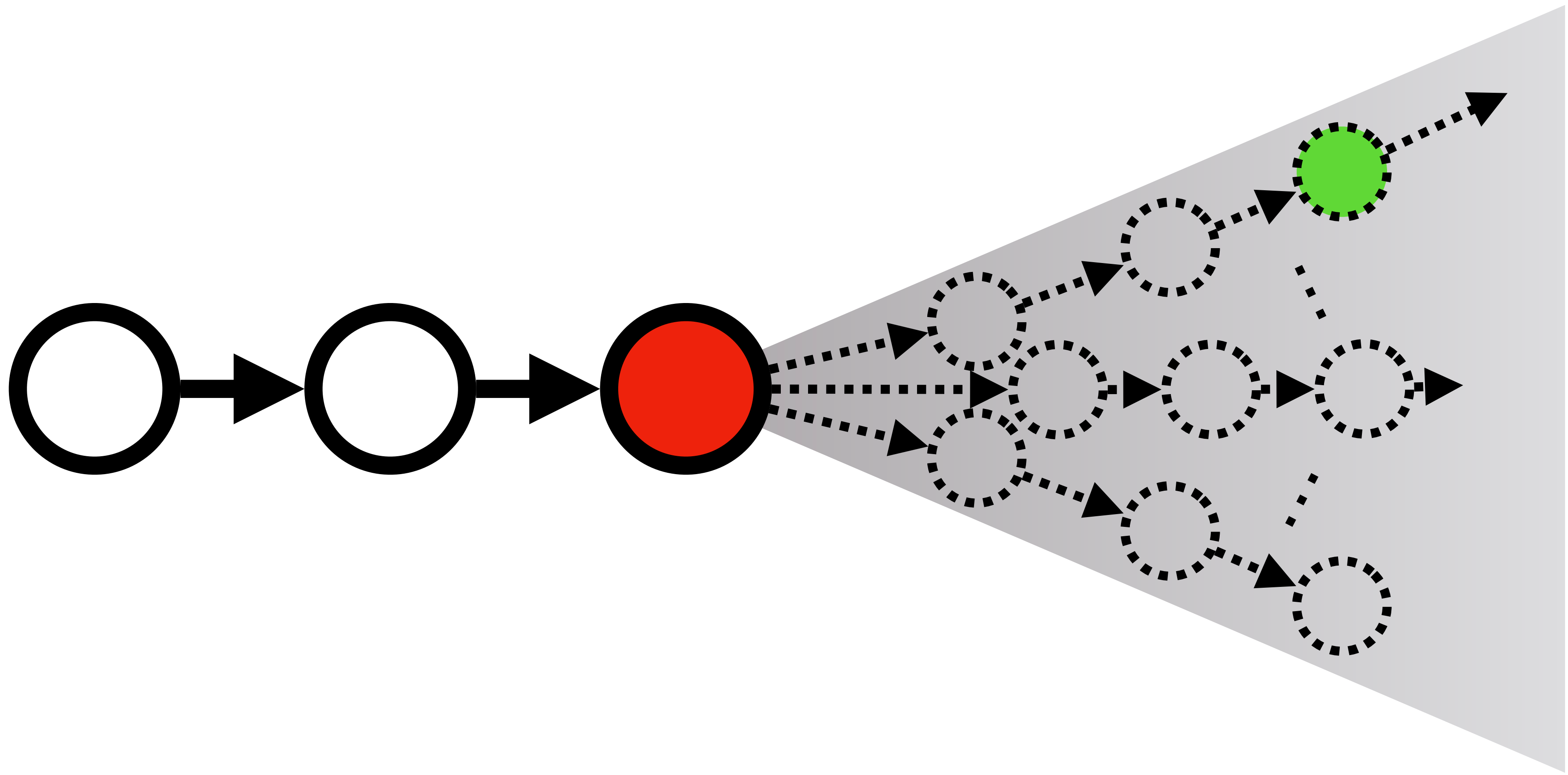
φ **permanently violated** by **t**

- **t** violates φ
- no matter how **t** continues, φ stays violated

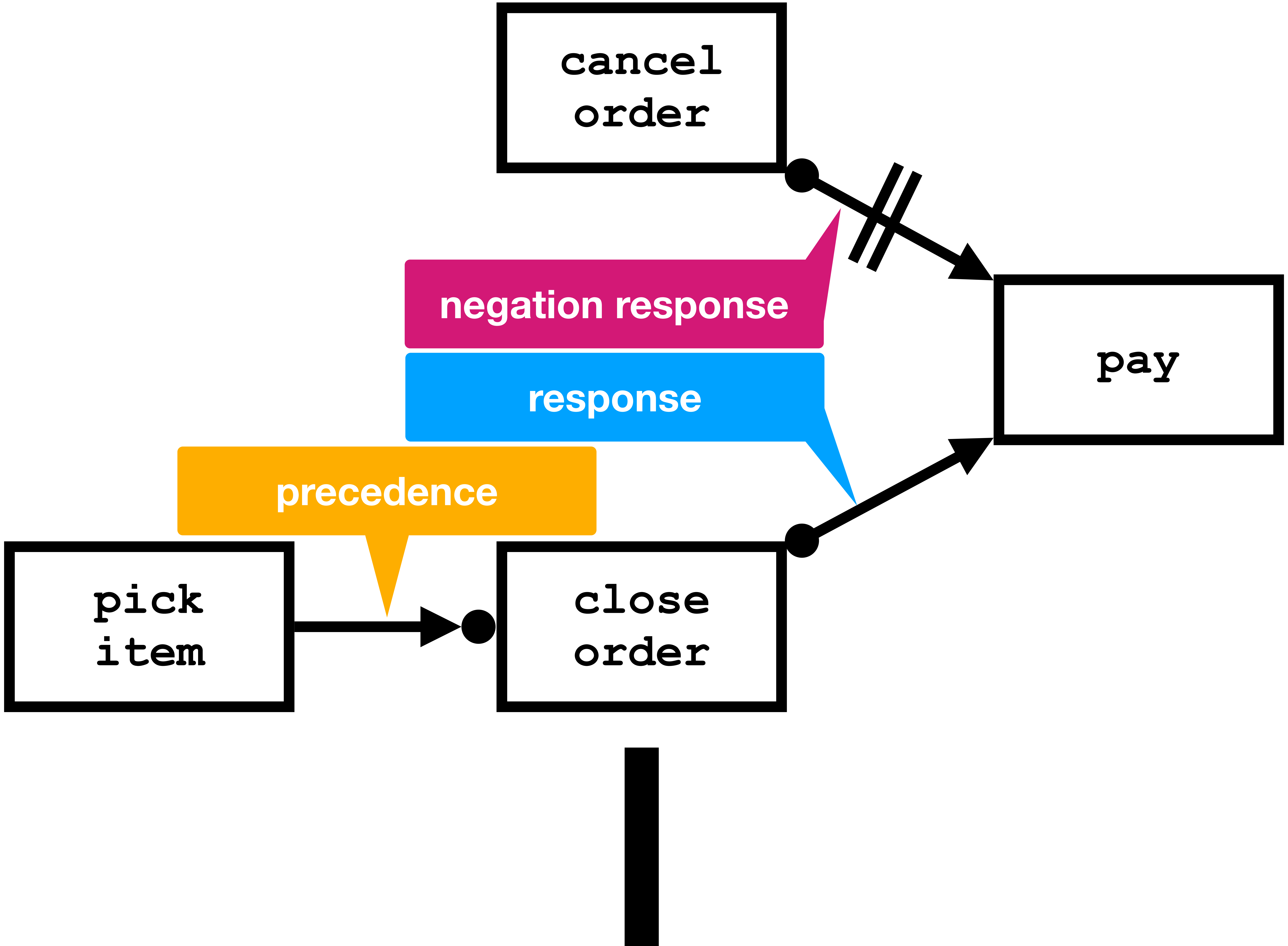


φ **currently violated** by **t**

- **t** violates φ
- there is a continuation of **t** that satisfies φ

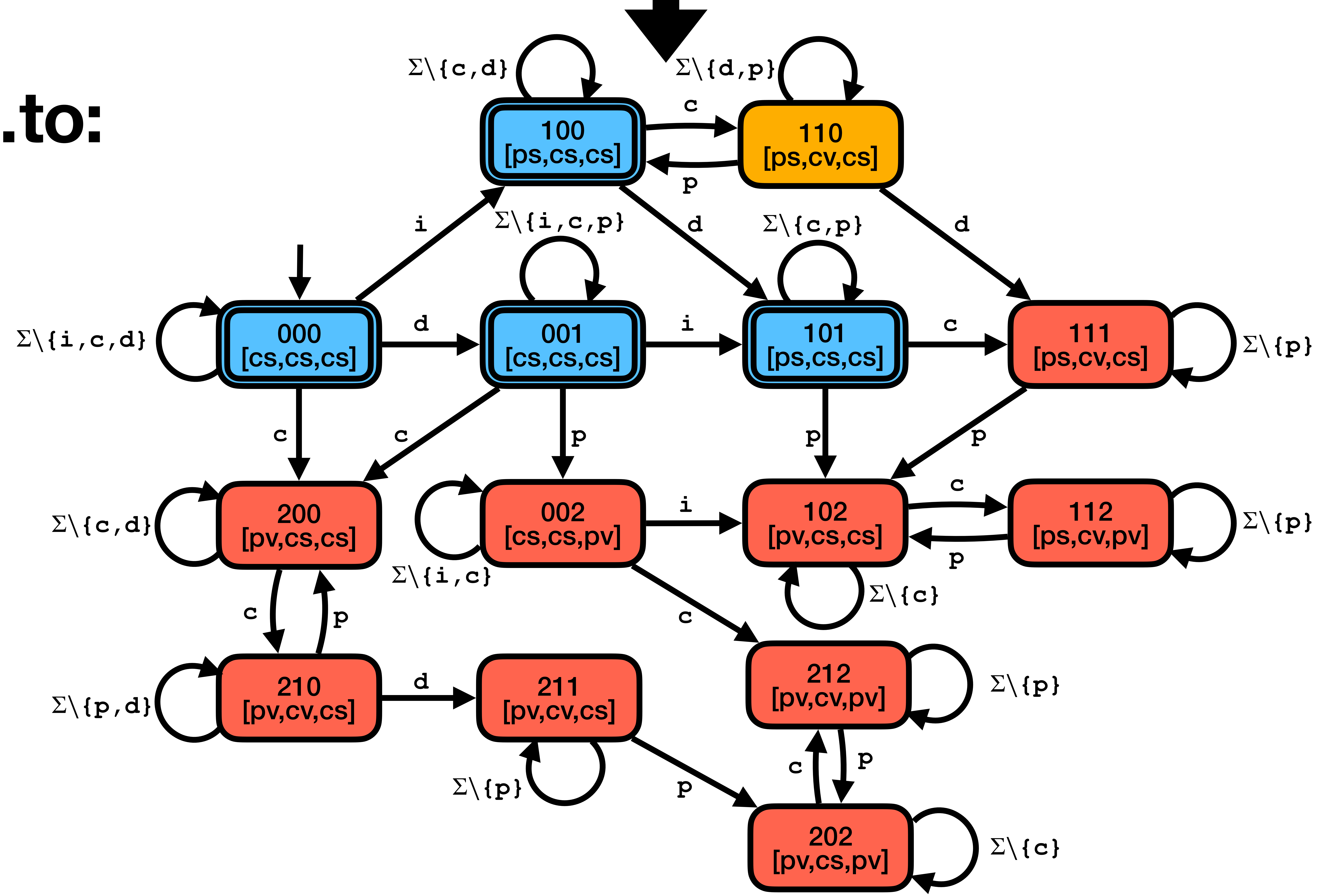


From...



cancel order
negation response
pay
response
precedence
pick item
close order

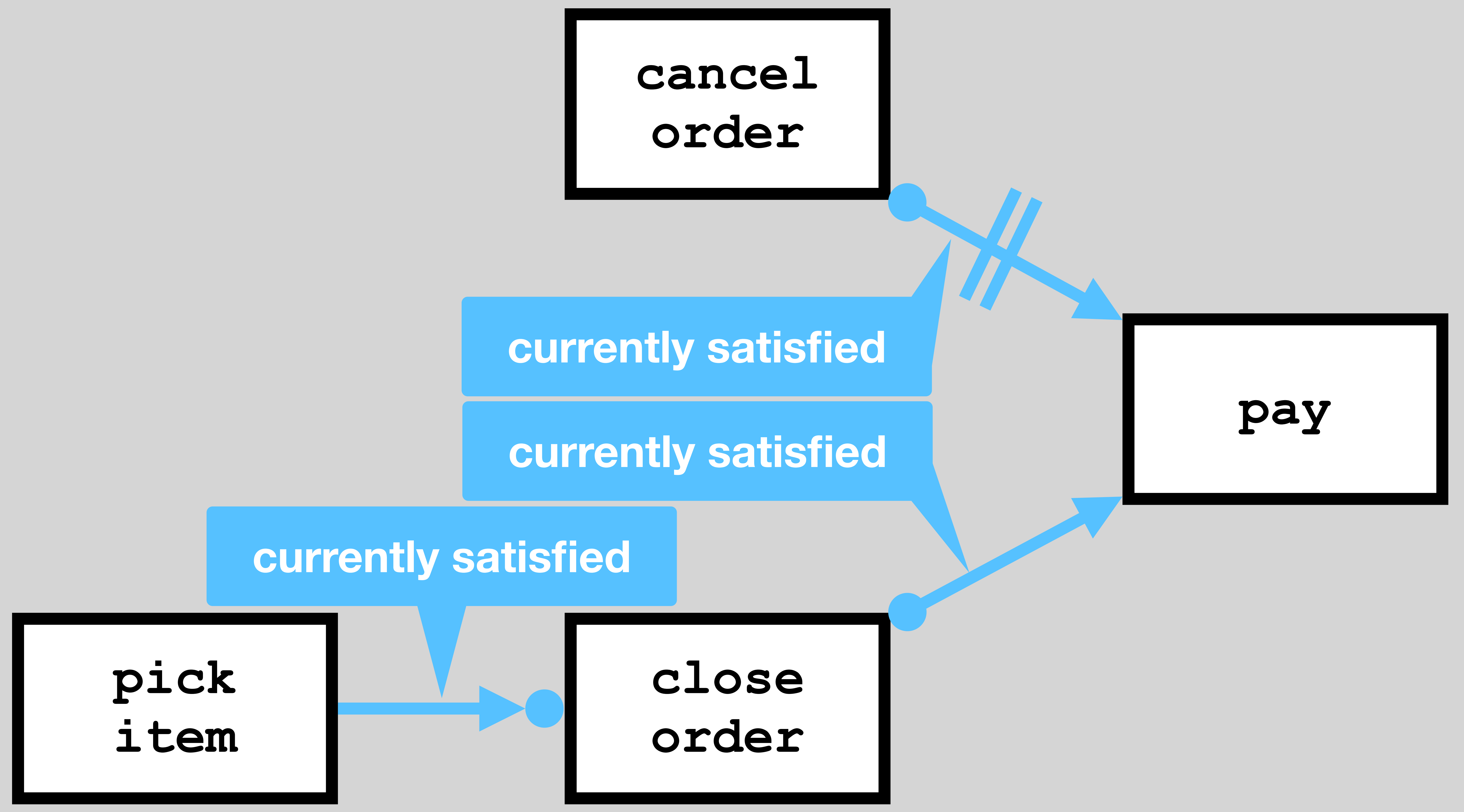
...to:



100
[ps,cs,cs]
110
[ps,cv,cs]
000
[cs,cs,cs]
001
[cs,cs,cs]
101
[ps,cs,cs]
111
[ps,cv,cs]
200
[pv,cs,cs]
002
[cs,cs,pv]
102
[pv,cs,cs]
112
[ps,cv,pv]
210
[pv,cv,cs]
211
[pv,cv,cs]
212
[pv,cv,pv]
202
[pv,cs,pv]
Σ\{c,d}
Σ\{d,p}
Σ\{i,c,p}
Σ\{c,p}
Σ\{i,c,d}
Σ\{p}
Σ\{i,c}
Σ\{c}
Σ\{p,d}
i
c
d
p

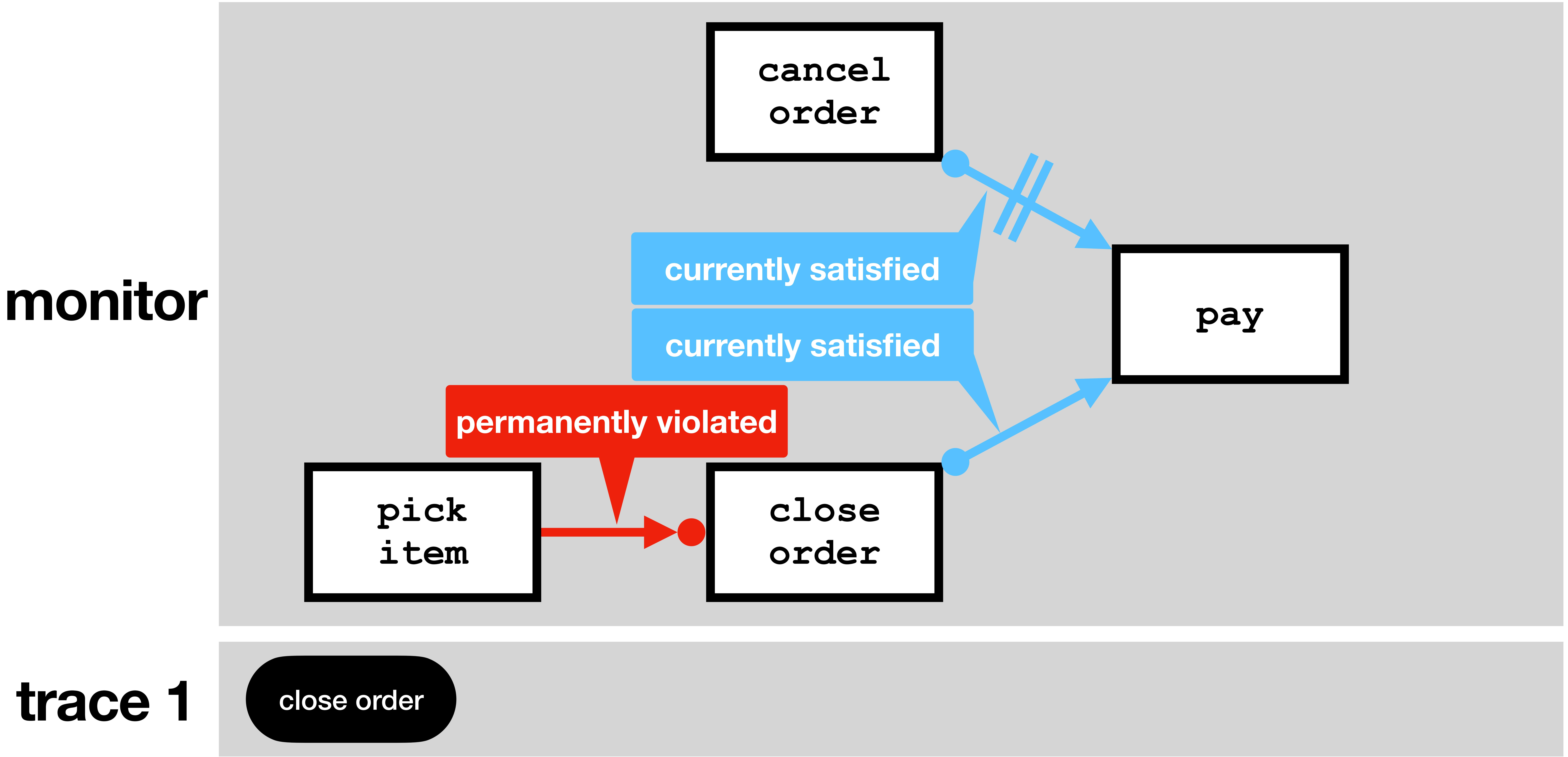
Monitor for anticipatory conformance

monitor



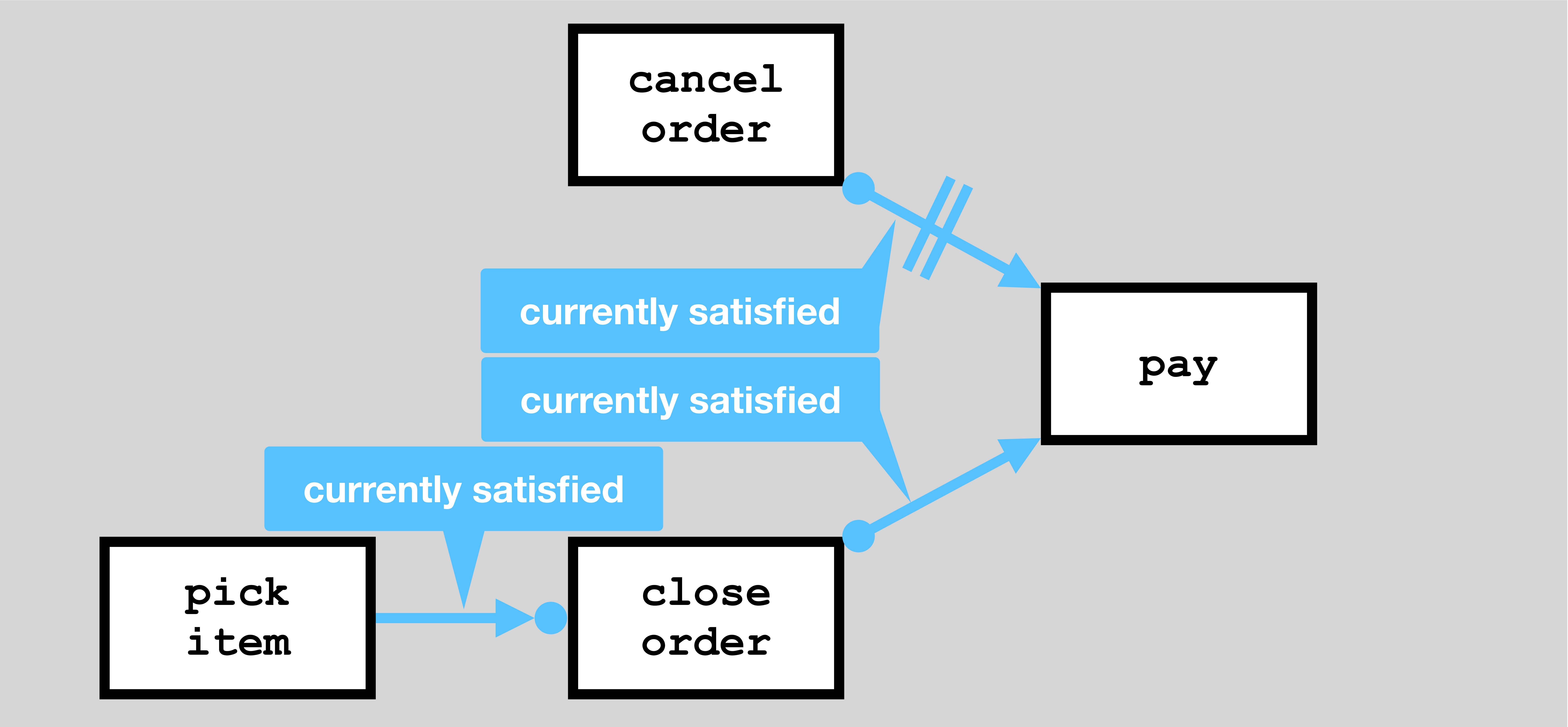
trace 1

Monitor for anticipatory conformance



Monitor for anticipatory conformance

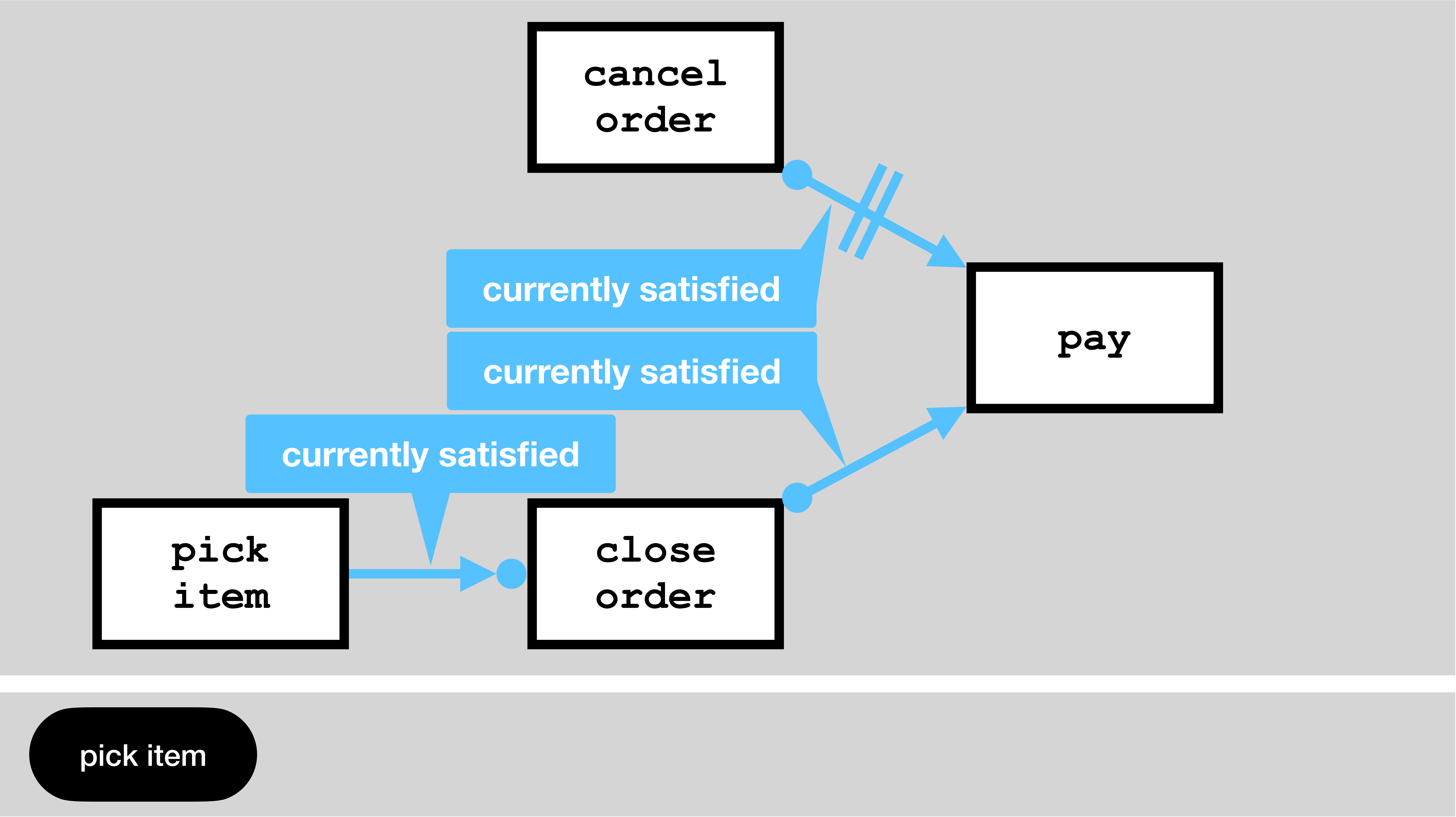
monitor



trace 2

Monitor for anticipatory conformance

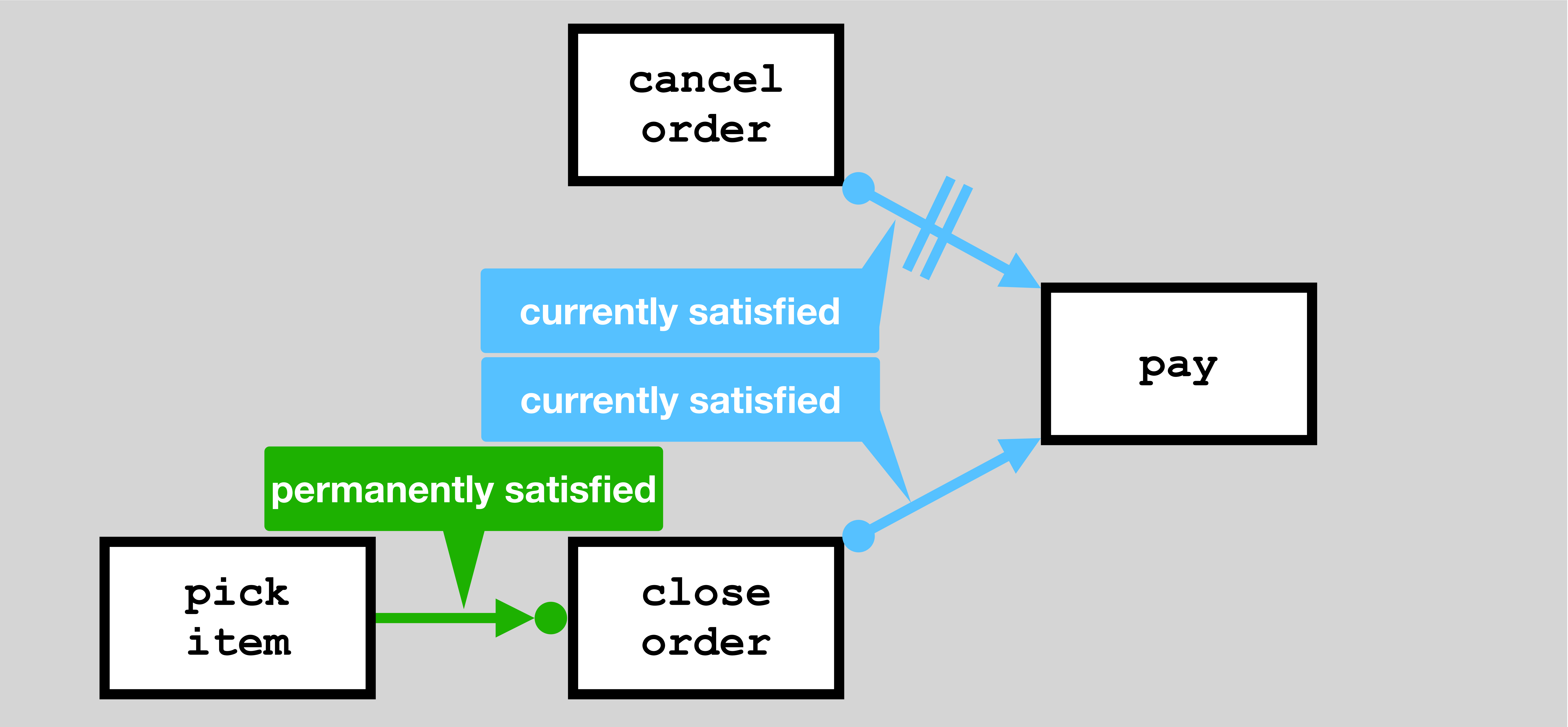
monitor



trace 2

Monitor for anticipatory conformance

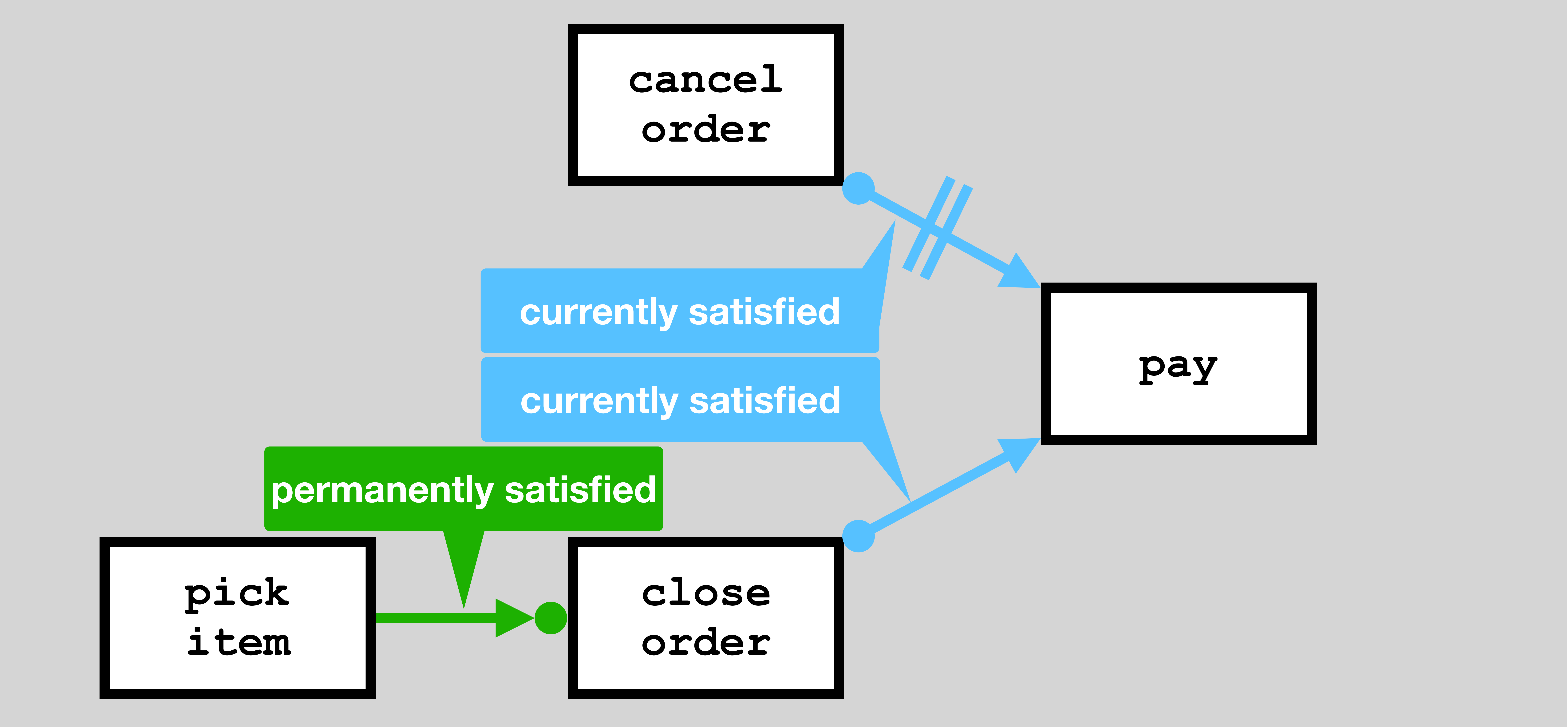
monitor



trace 2

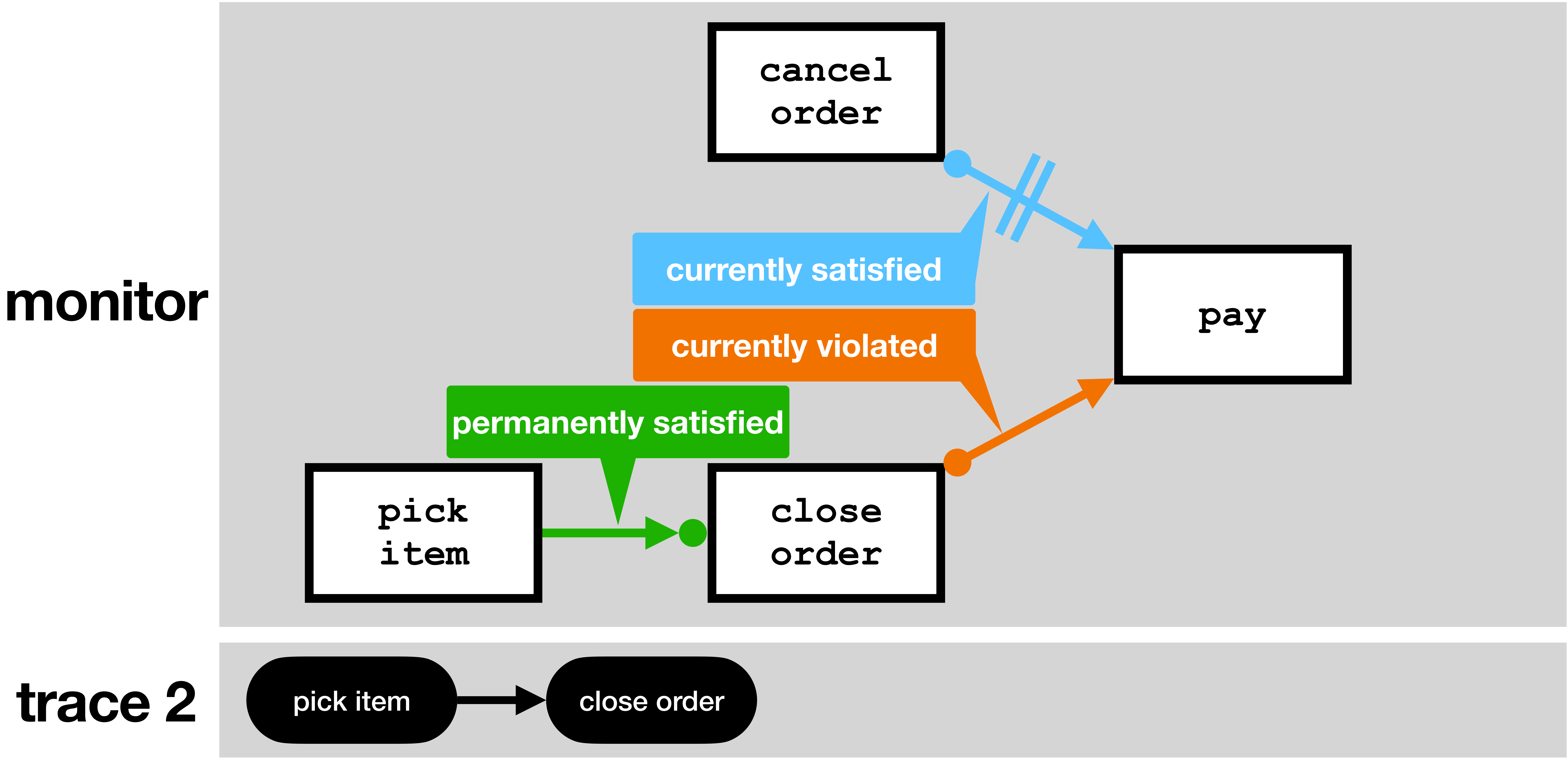
Monitor for anticipatory conformance

monitor

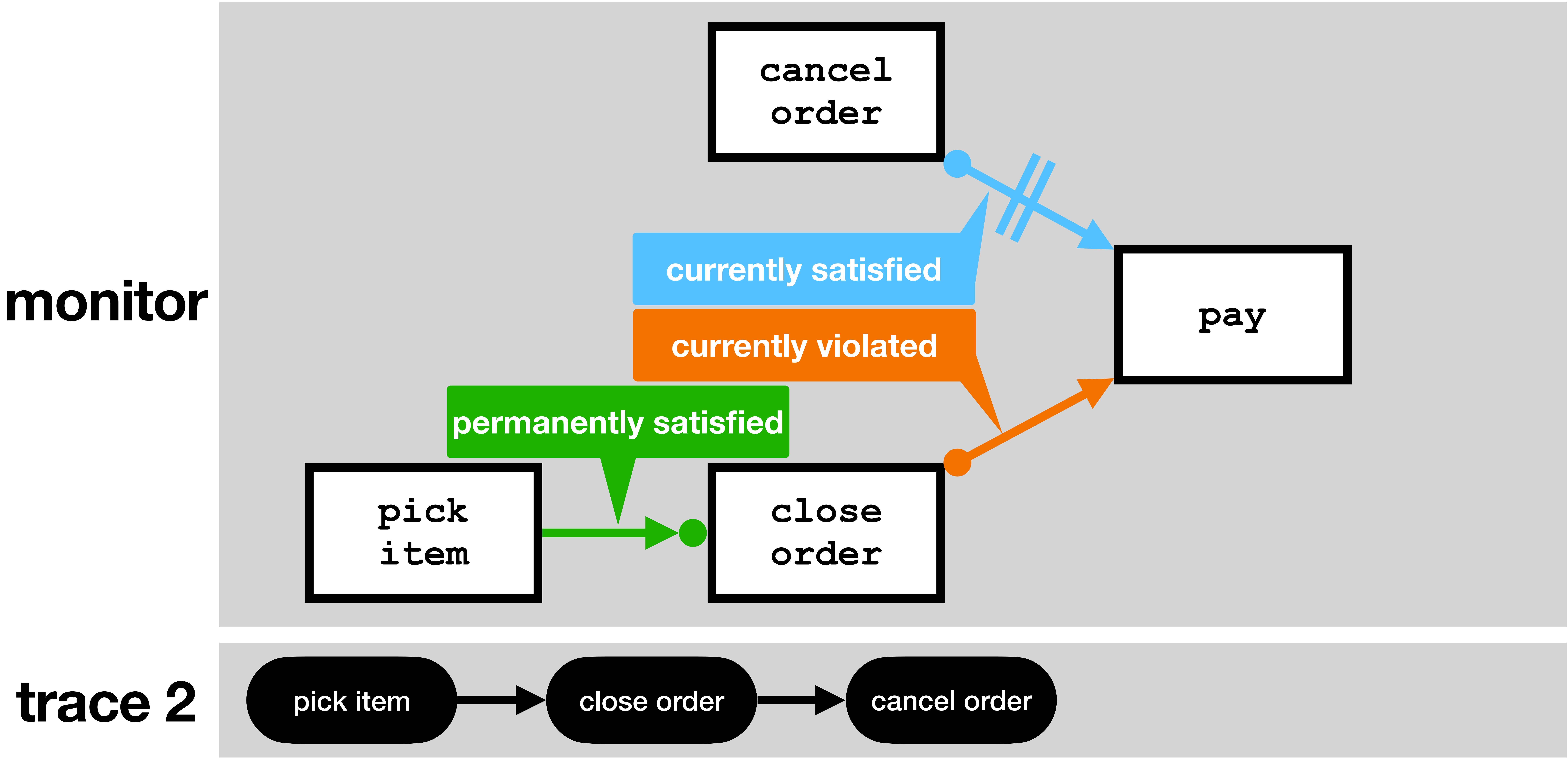


trace 2

Monitor for anticipatory conformance

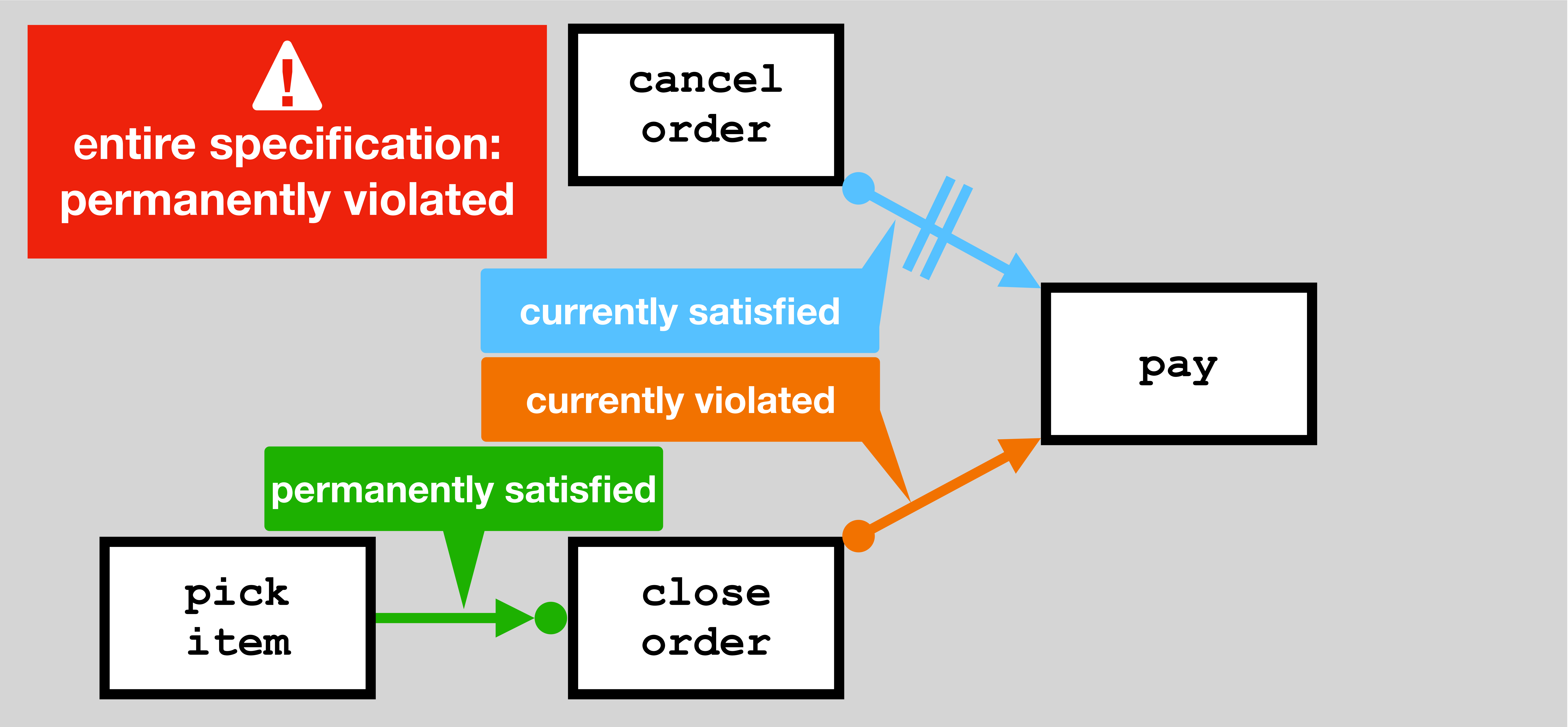


Monitor for anticipatory conformance

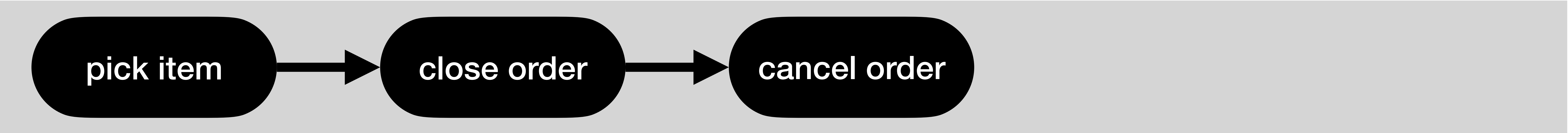


Monitor for anticipatory conformance

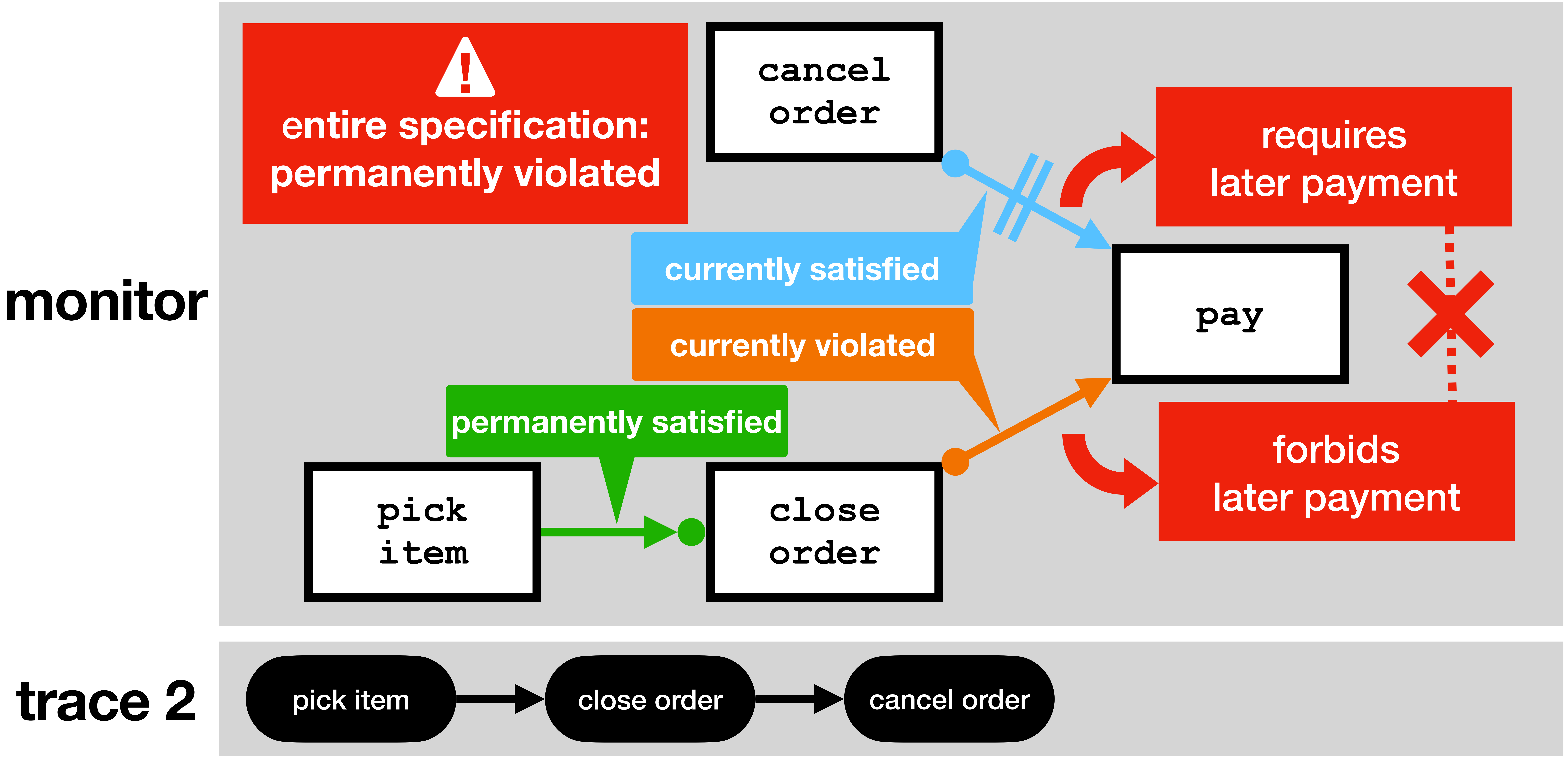
monitor



trace 2

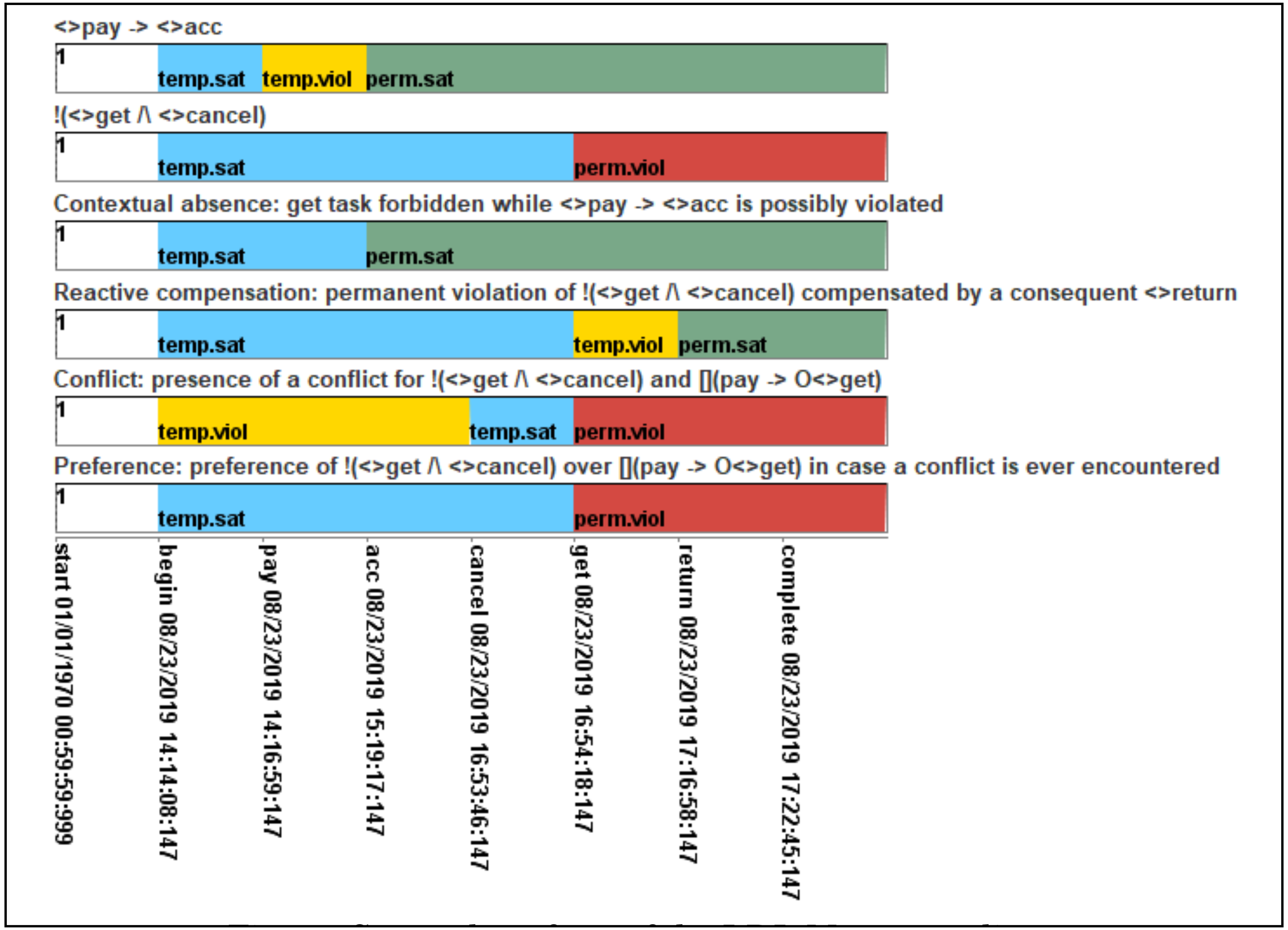


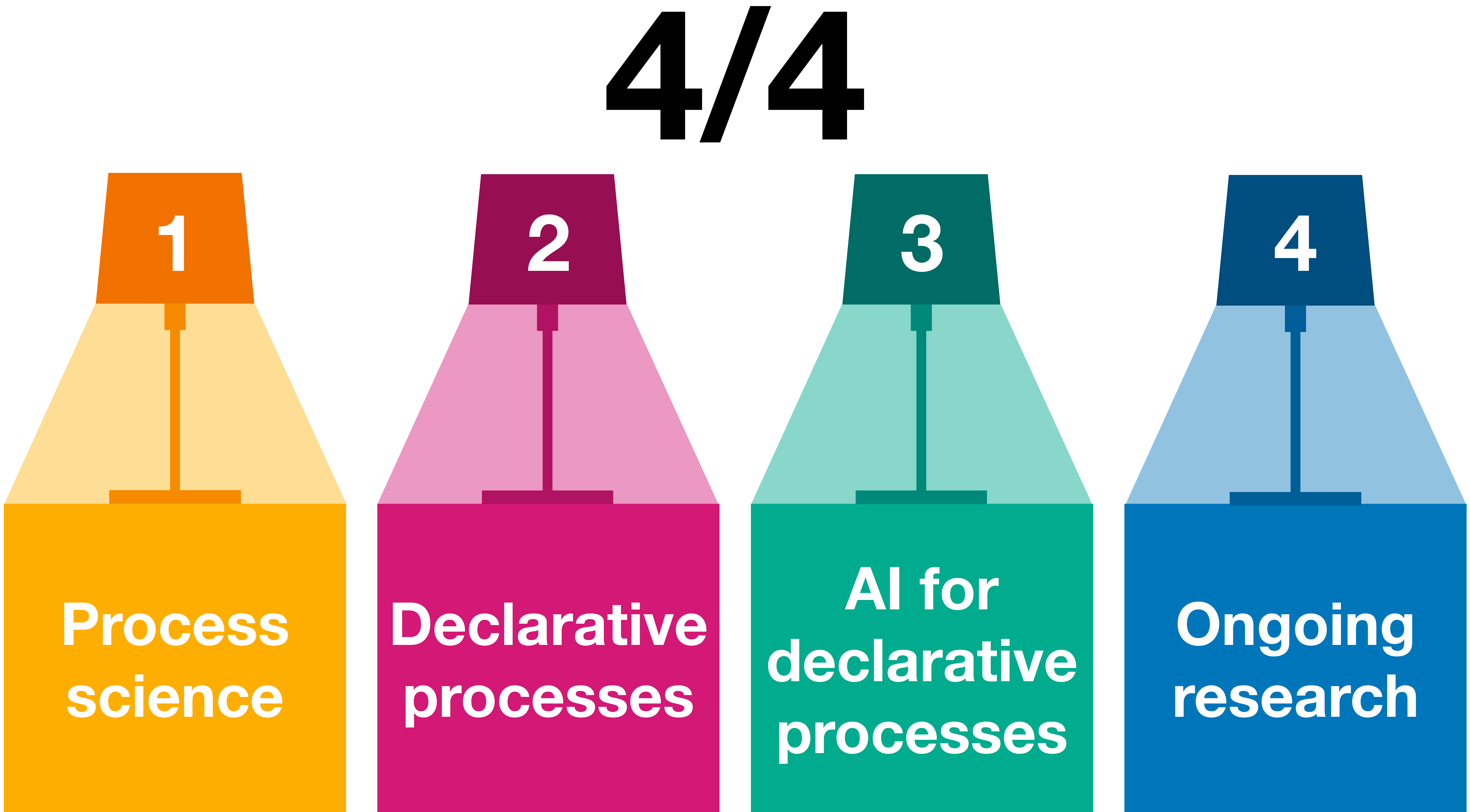
Monitor for anticipatory conformance



Fully implemented in the RuM toolkit

rulemining.org

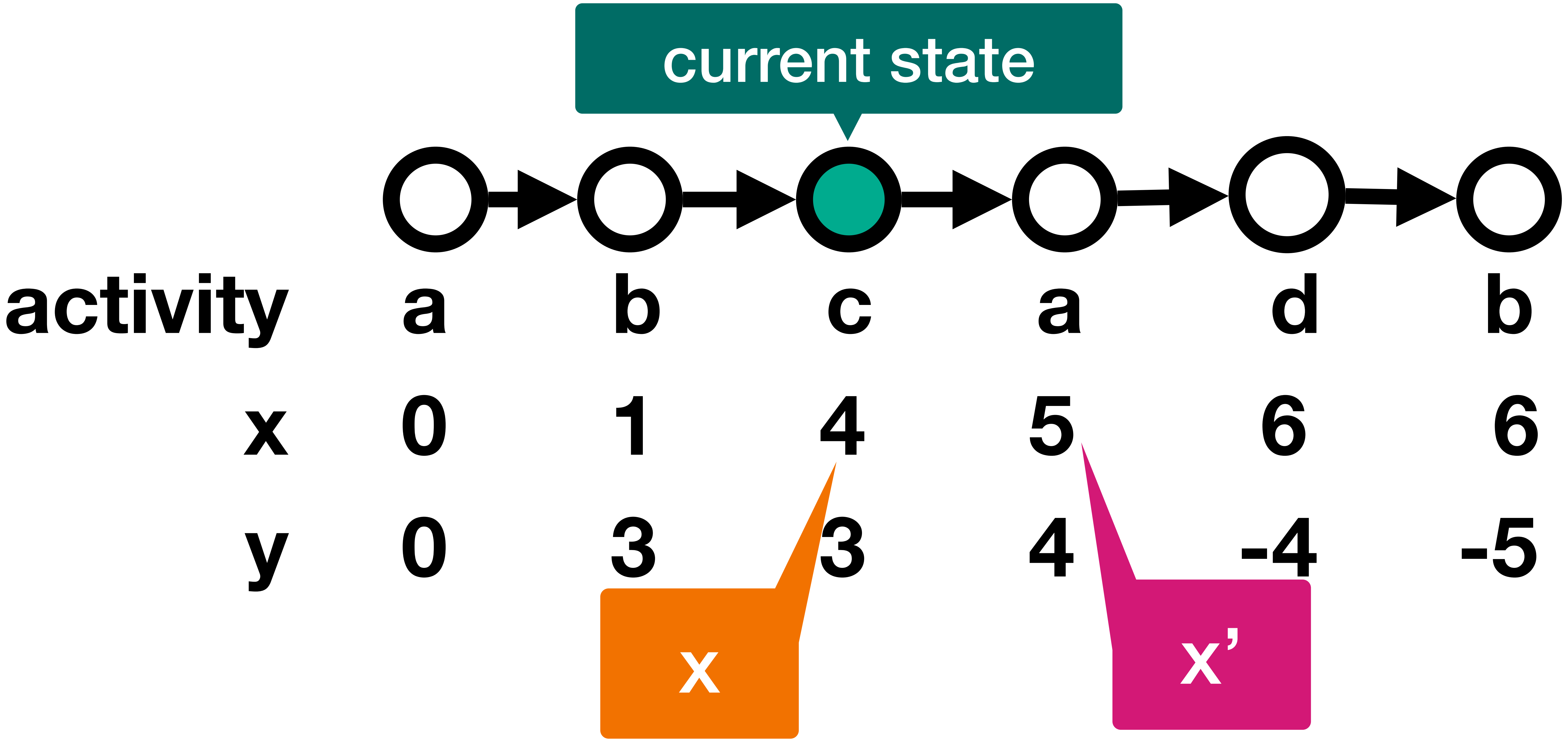



4/4
1
Process science
2
Declarative processes
3
AI for declarative processes
4
Ongoing research

Current developments

Data-awareness on events payload

[___,AAAI2022][___,AAAI2023][___,AAAI2024]



Data-LTLf: constrain the evolution of data variables over time (modulo theories)

$$\mathsf{G}(x' > x) \wedge \mathsf{F}(x = 2)$$

Current developments

Data-awareness on multiple objects and relations

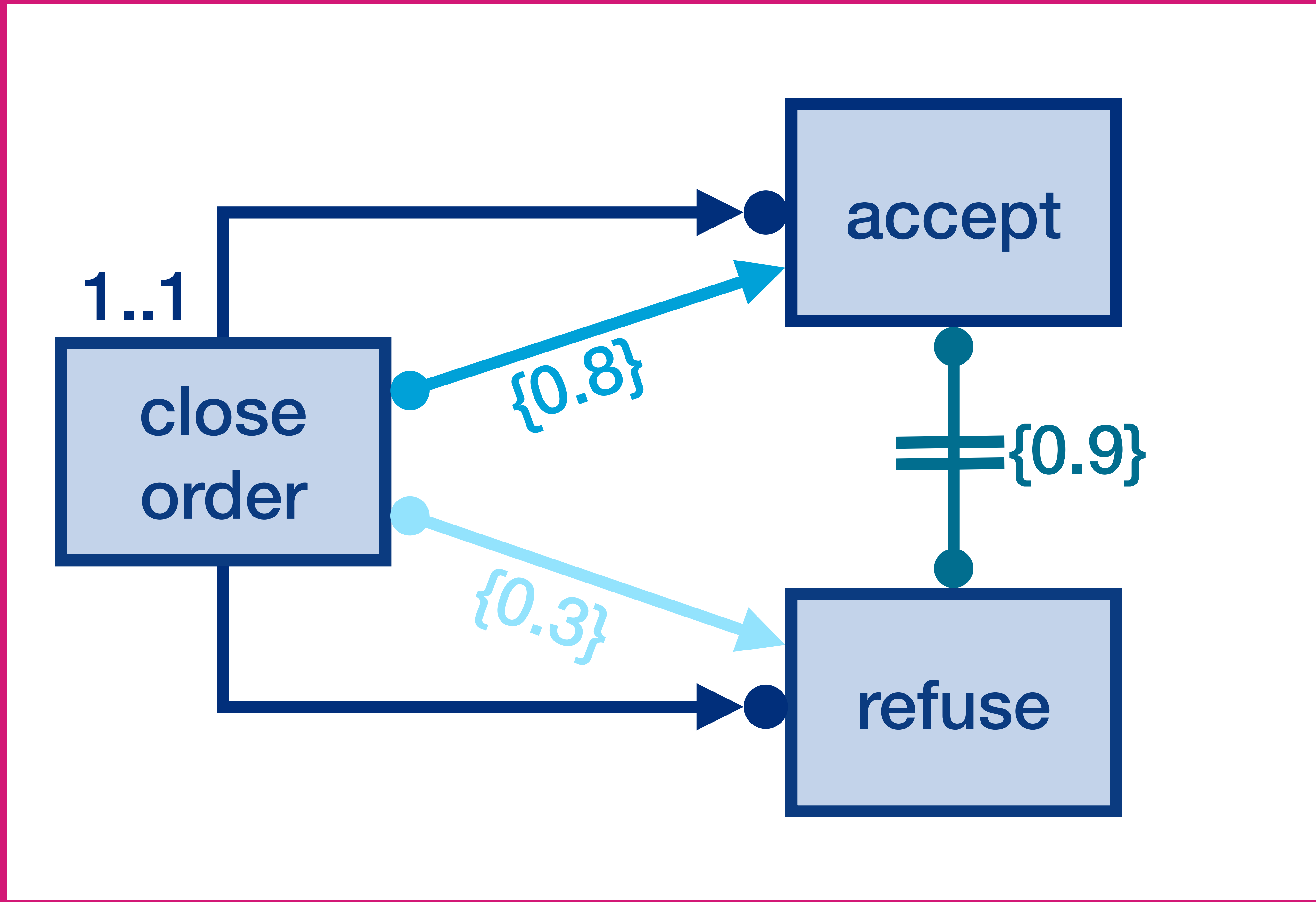
[___,DL2017][___,BPM2019][___,IJCAI2022]

FO-LTLf and **temporal description logics** to constrain events and objects over time (**temporal constraints** over **data models**)

Current developments

Uncertainty

In the specification

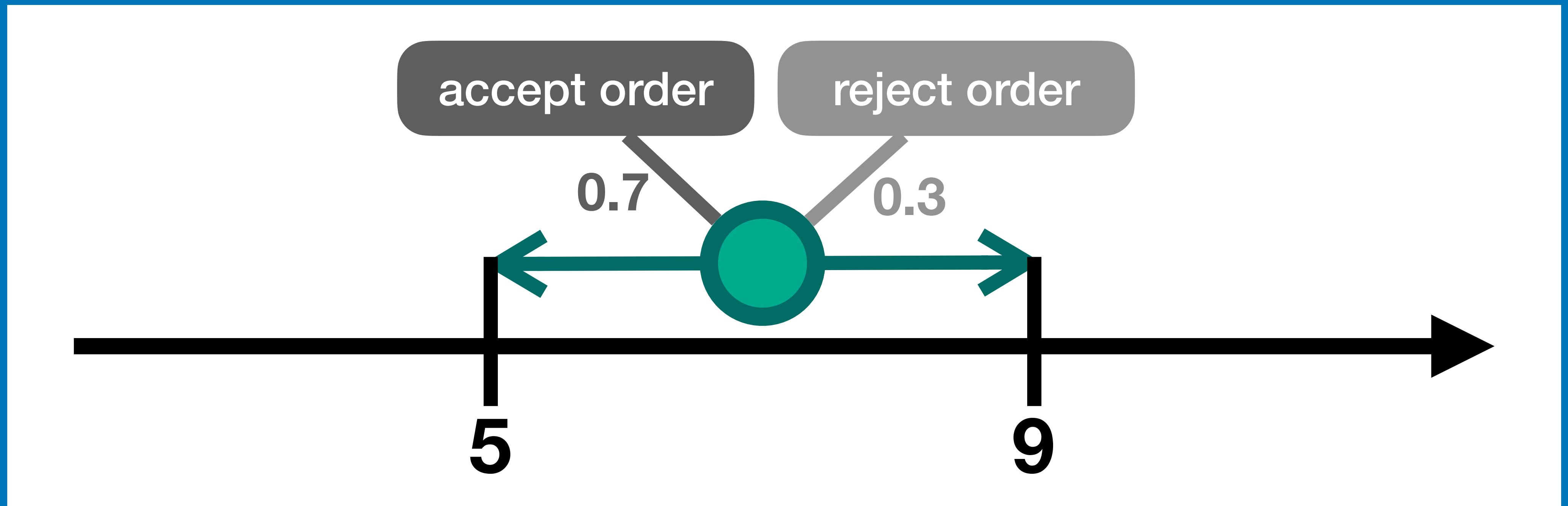


probabilistic LTLf/Declare

[___,AAAI2020]

[___,BPM2020][___,IS2022]

In the event data



probabilistic events

[___,BPM2022][___,EAAI2023]

LTLf over fuzzy event logs

[___,BPM2024]

Current developments

Conversational systems for declarative process mining [___,BPMDemo2023]

C-4PM

You can ask me multiple things, for example:

- Describe the process
- List the activities in the process
- Can you give me some conformant traces?
- Is it possible that ER Triage occurs before IV Liquids?
- Can activity IV Antibiotics be performed before activity Admission NC?
- In which cases ER Triage occurs right after ER Registration?
- Will IV Antibiotics eventually happen twice?

Can activity IV Antibiotics be performed before activity Admission NC?

The specification allows for that behavior. Here are some cases in which, If AdmissionNC happens at least once then IVAntibiotics has to happen or happened before AdmissionNC.

ER Registration, Leucocytes, CRP, LacticAcid, ER Triage, ER Sepsis Triage, IV Liquid, IV Antibiotics, Admission NC, CRP, Leucocytes, Leucocytes, CRP, Leucocytes, CRP, CRP, Leucocytes, Leucocytes, CRP, CRP, Leucocytes, Release A

ER Registration, ER Triage, CRP, LacticAcid, Leucocytes, ER Sepsis Triage, IV Liquid, IV Antibiotics, Admission NC, CRP, CRP, Release A

Learning, explainability, reasoning - combined

Counterfactual generation for explainability under temporal logical guarantees [____,AAAI2025]

Starting point

- **Counterfactual explanations** - key XAI technique (e.g., to suggest changes to input data that could alter predictions towards better outcome)
- **Input data** in process science: **execution traces (discrete time sequences)**

Observation: in a process, **not all traces make sense**

- **Temporal constraints** as background knowledge that separates the possible from the impossible

Result: **genetic algorithms for counterfactual trace generation infused with temporal reasoning** to ensure that **only conforming traces can be proposed**

Conclusions

Connection between **AI** and **process science**

Synergy between **model-driven** and **data-driven** techniques for the **analysis of system dynamics**

The case of **declarative processes**: **temporal logics** and **automata** to the rescue

Conclusions

Connection between **AI** and **process science**

Synergy between **model-driven** and **data-driven** techniques for the **analysis of system dynamics**

The case of **declarative processes**: **temporal logics** and **automated reasoning** to the rescue