Analysis of Data-Centric Workflows

Victor Vianu
UC San Diego & INRIA-Saclay

Workflows centered around data

Increasingly common:

- E-commerce
- Supply chain management
- Case management in health-care
- Scientific workflows
- Business processes

….

Complex logic → need for analysis tools

Cautionary tale: healthcare.gov !

Analysis tasks

- Automatic verification of desirable properties

  “no product can be shipped before being paid”

  “if a product is paid, it is eventually delivered or the customer is refunded”

- Comparison, customization, optimization of workflows

  “is my workflow equivalent to yours ?”

- Runtime analysis

  “what can I infer from my local view of the workflow?”

  “what should I do next?”

Long collaboration with INRIA

- Automatic Verification

  PODS 2008, ICDT 2009, TODS 2009,

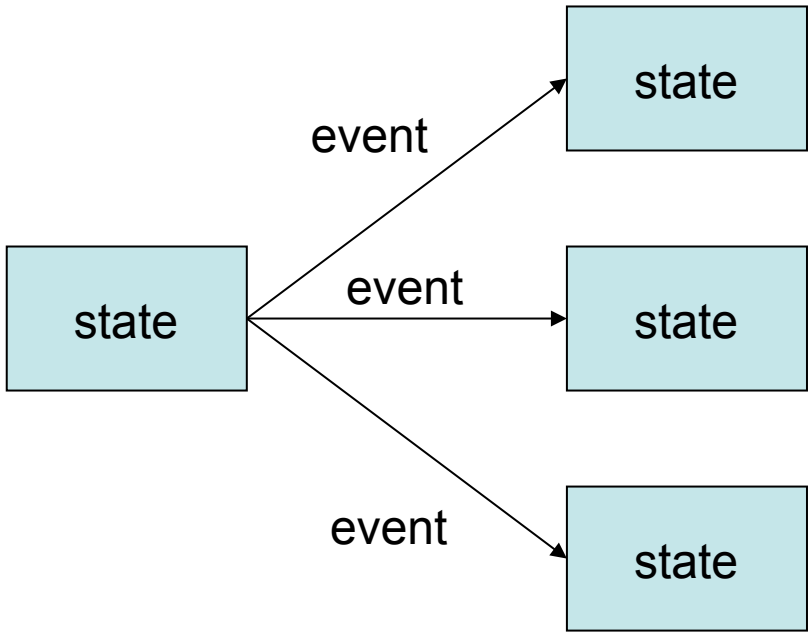
  ICDT 2011, BPM 2011, TODS 2012
- Comparison, customization, optimization of workflows

  ICDT 2011, TODS 2012
- Runtime analysis for collaborative workflows

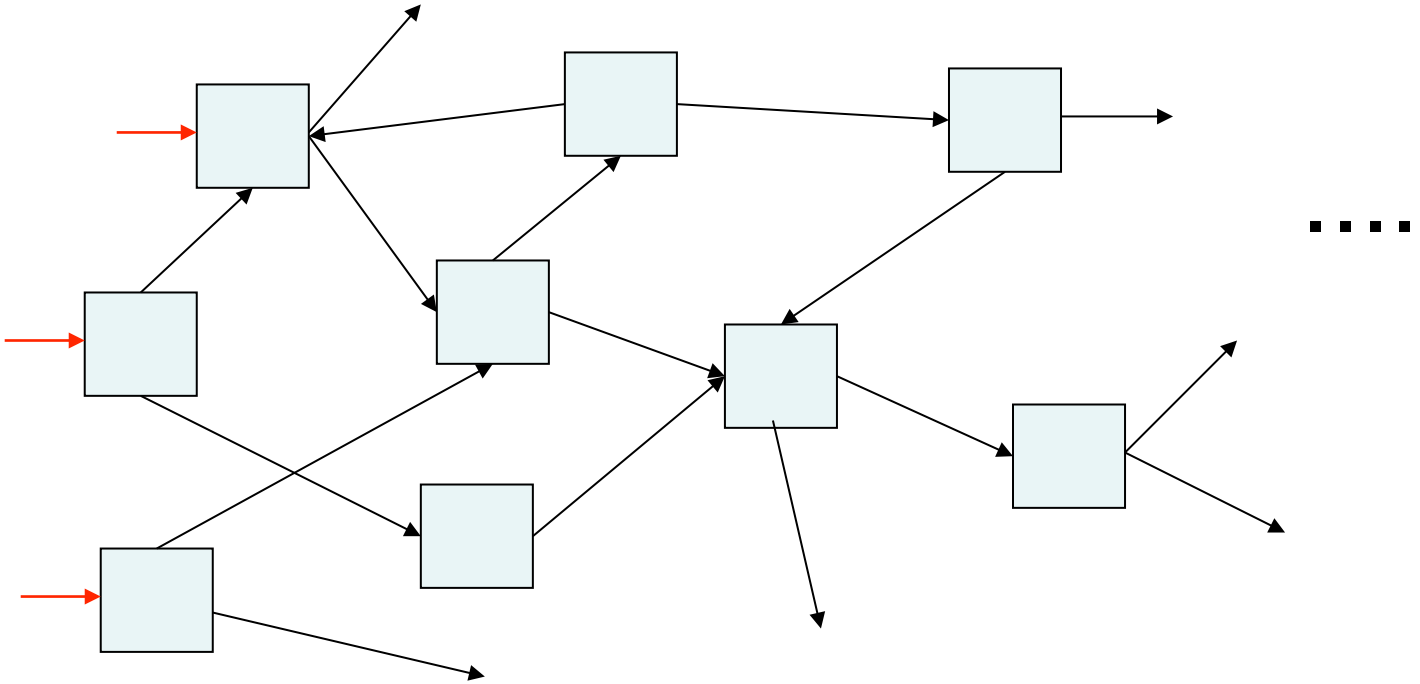
  PODS 2013

What is a data-centric workflow?

- States (data)
- Events (with data)
- Transitions



DB transition system



Infinite-state system

Specifications of data-centric workflows

Two frameworks used in our work:

- Active XML (INRIA Gemo):

  XML with embedded functions

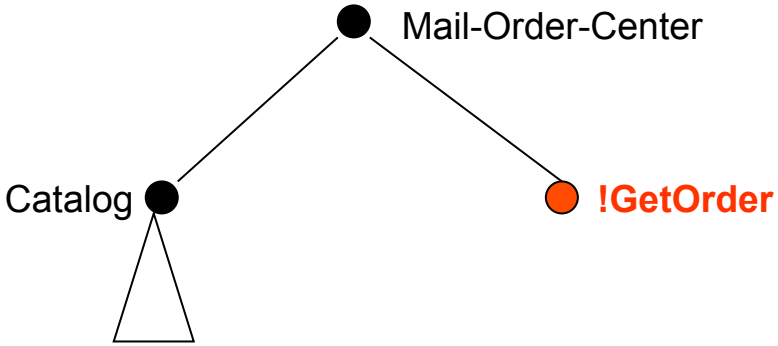
  events: function calls

- Business Artifacts (IBM Yorktown)

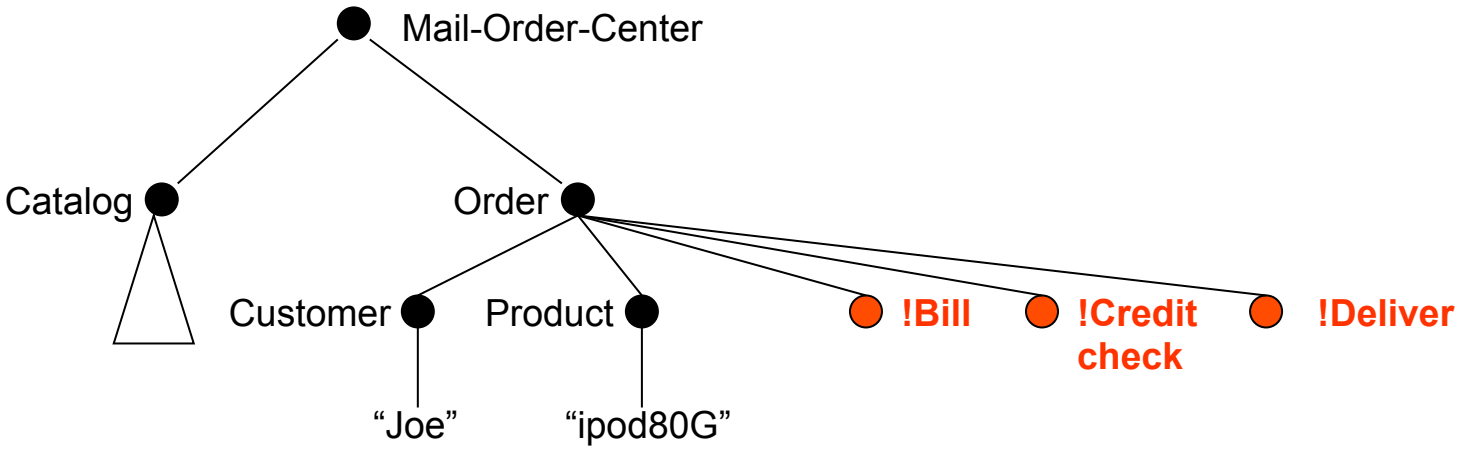
  relational database, evolving “artifact” tuple

  events: pre/post conditions

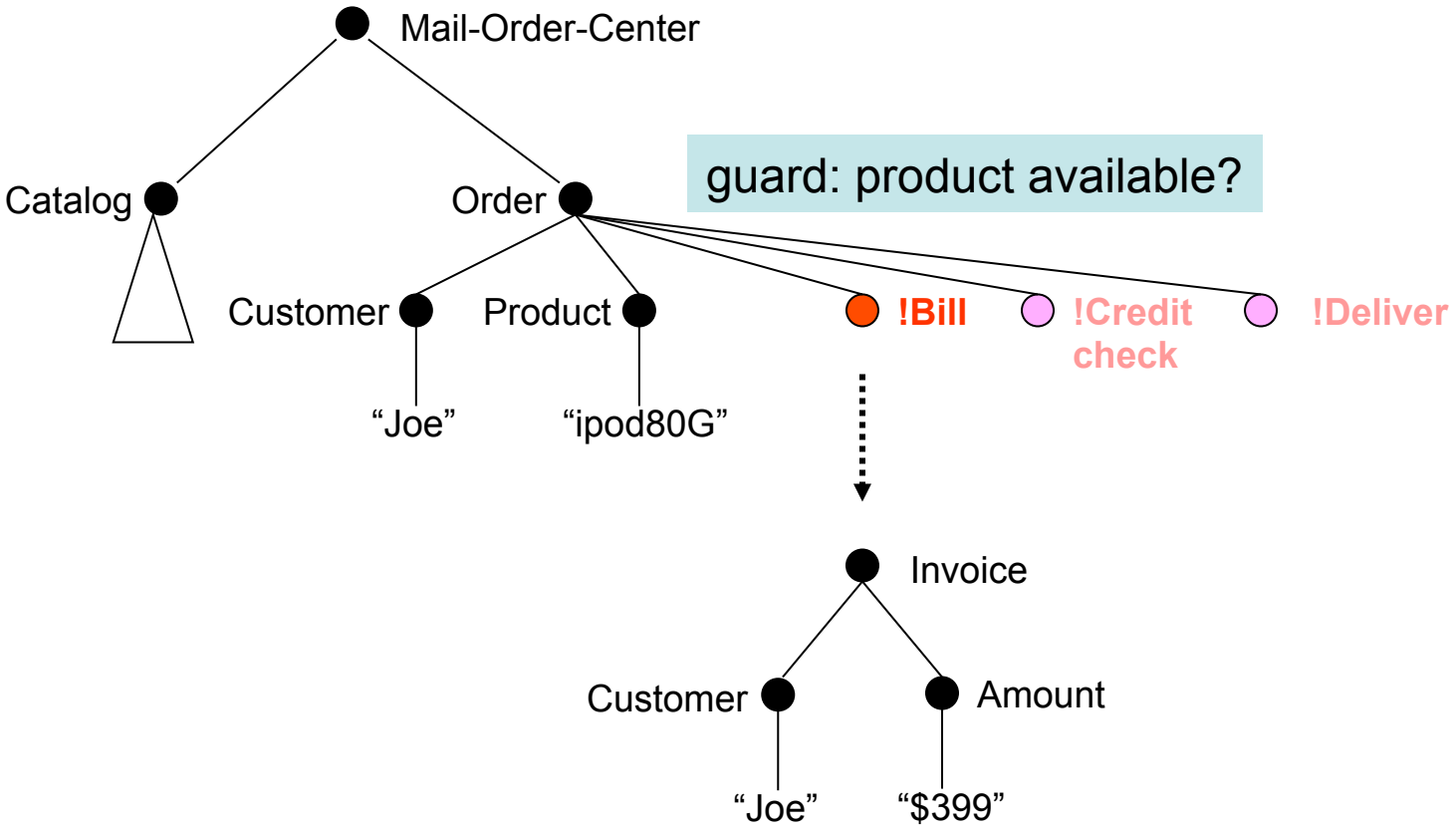
AXML workflow by example



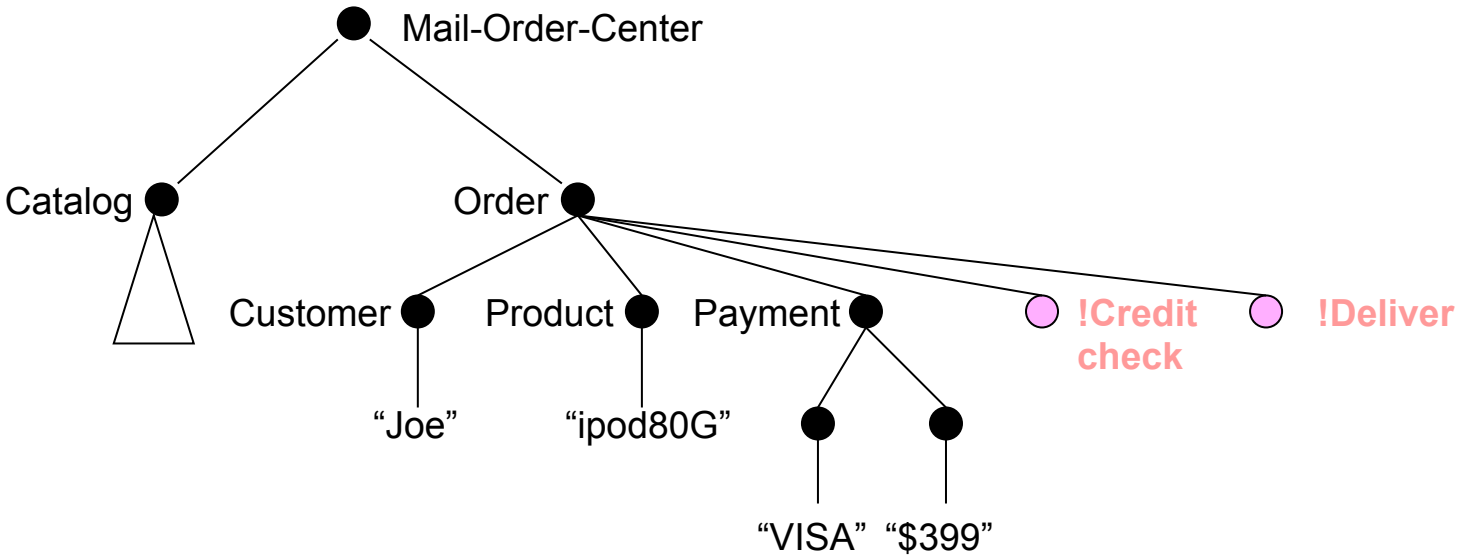
AXML workflow by example



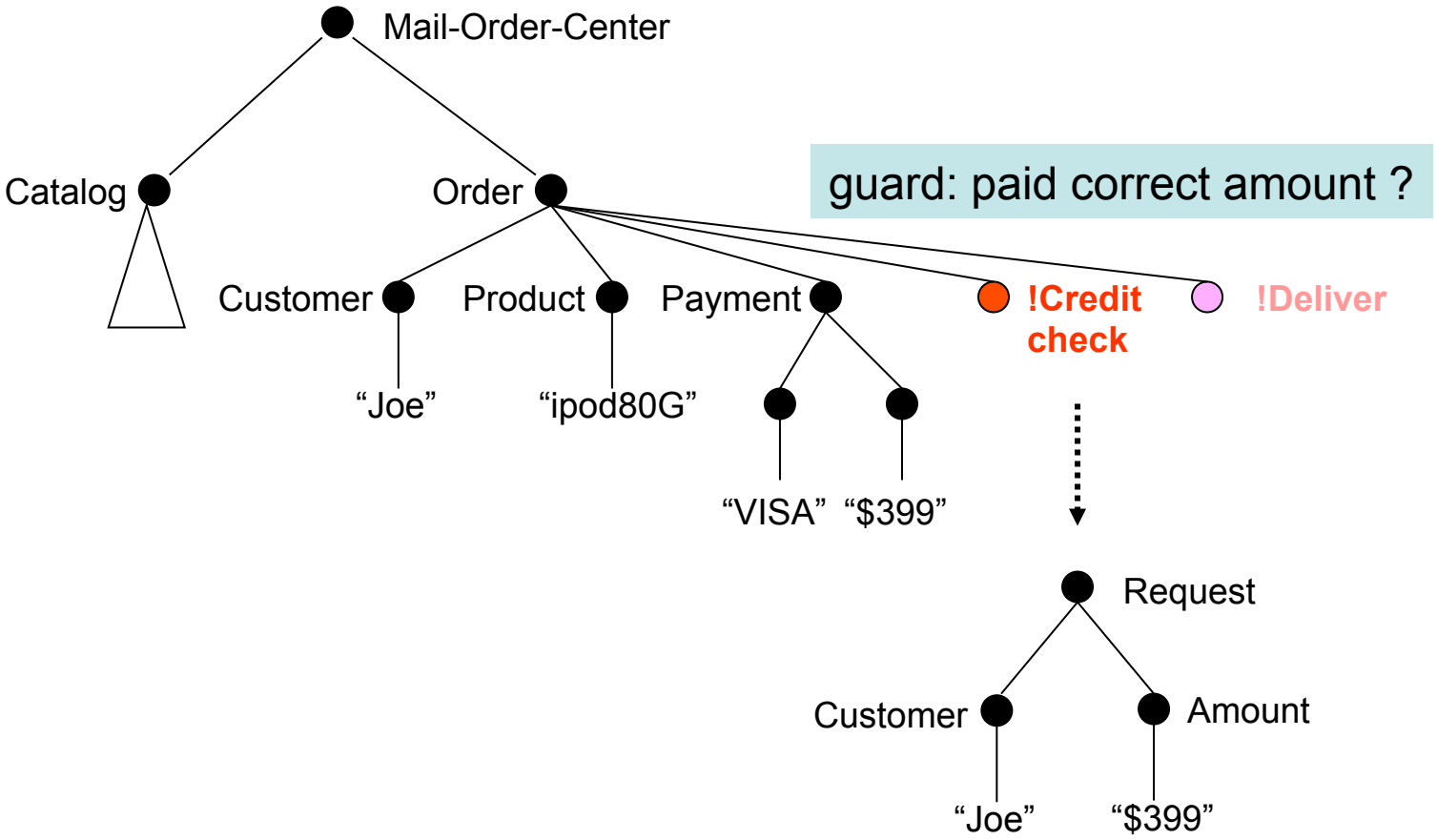
AXML workflow by example



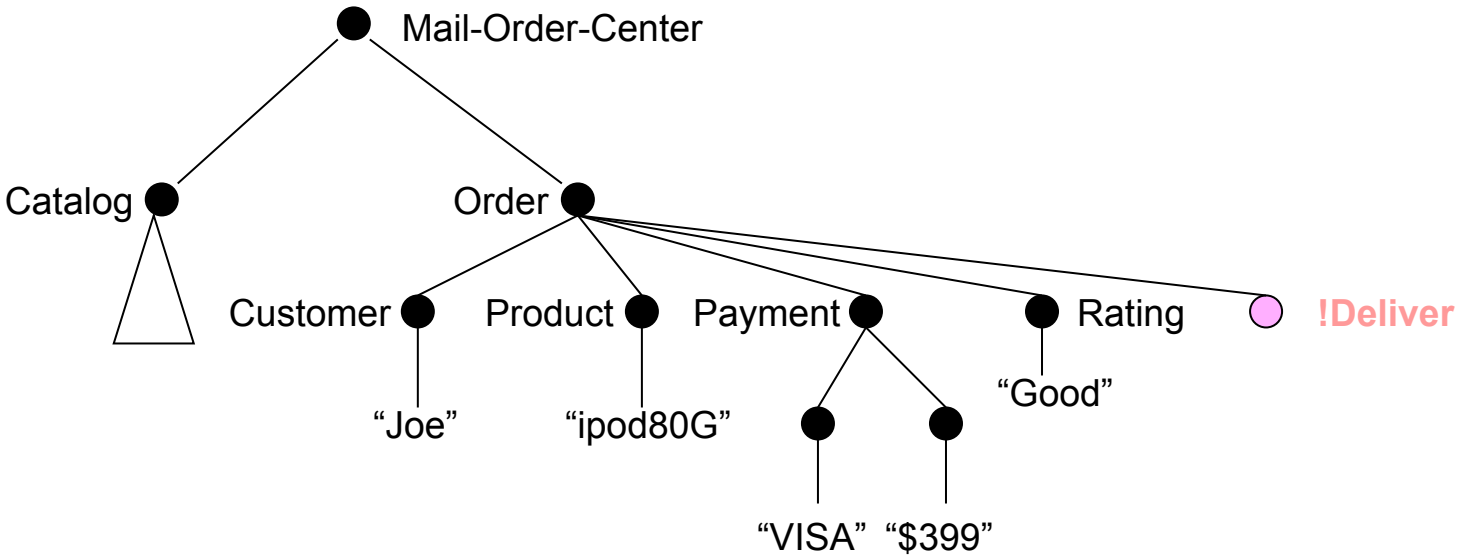
AXML workflow by example



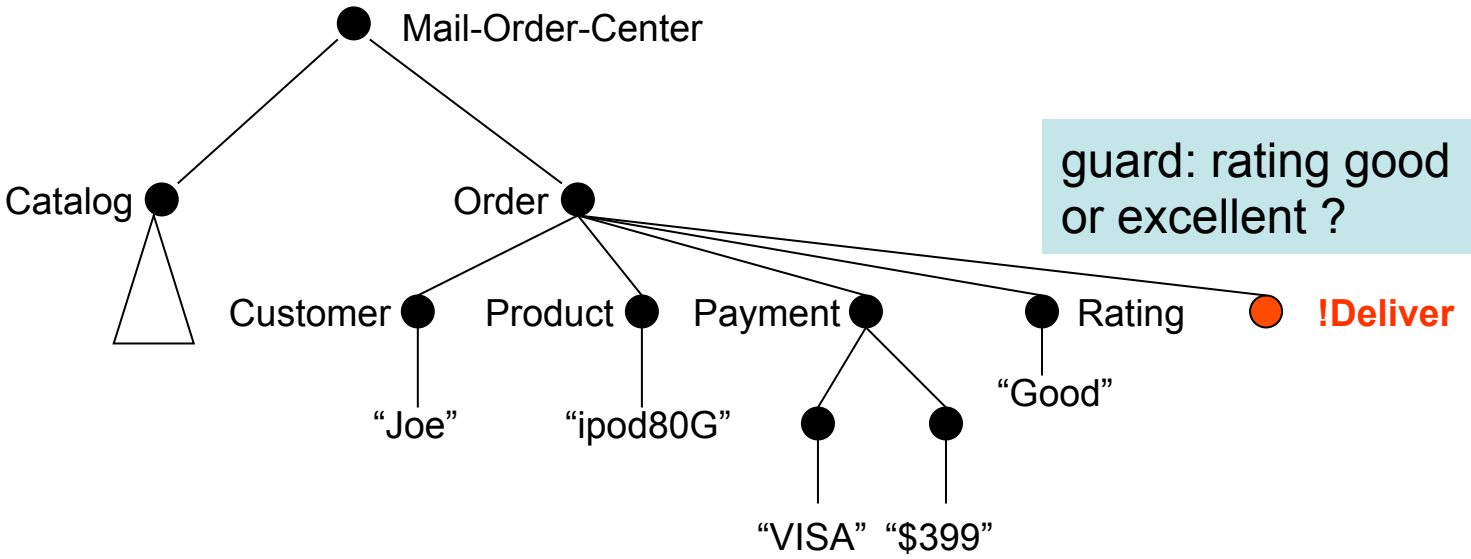
AXML workflow by example



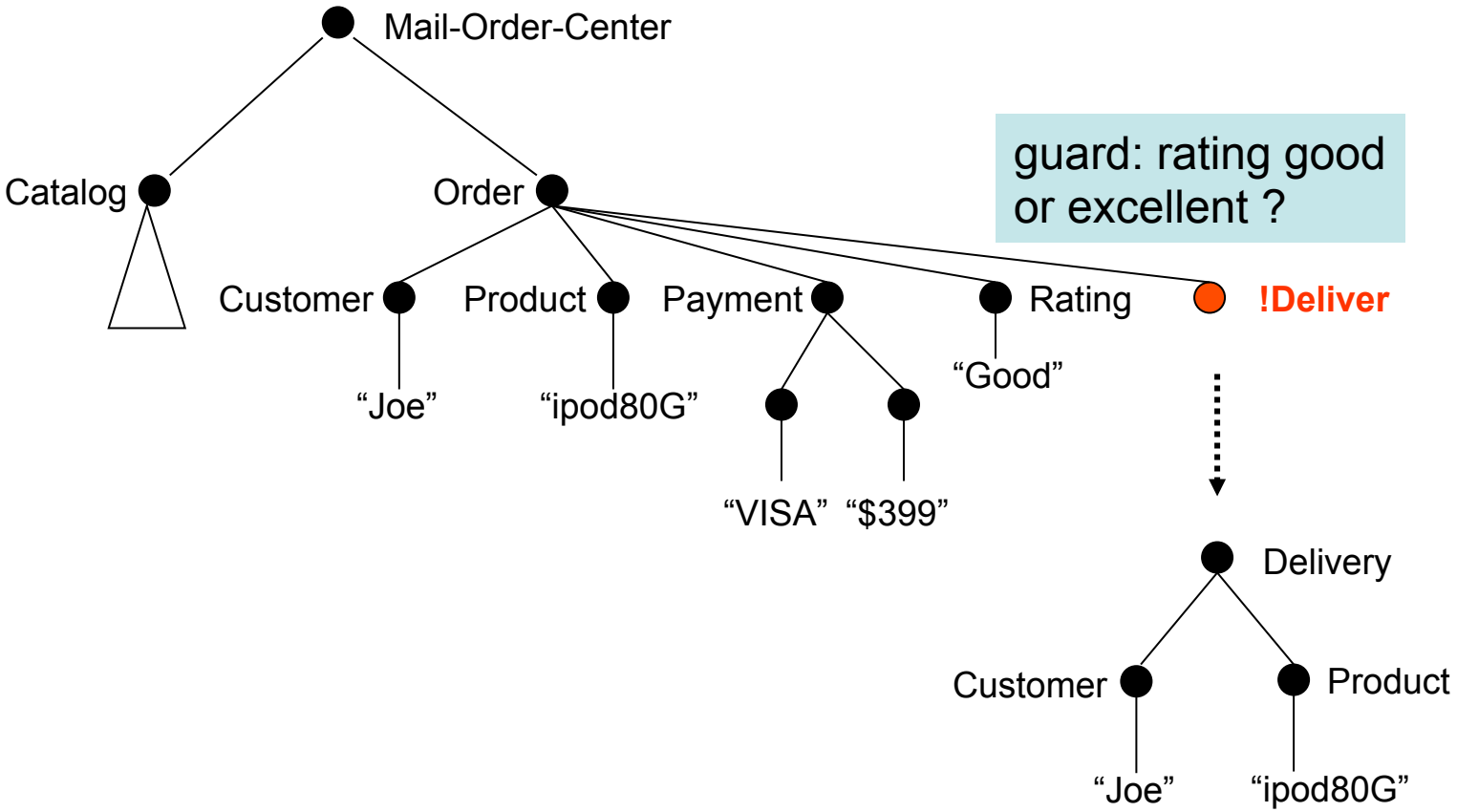
AXML workflow by example



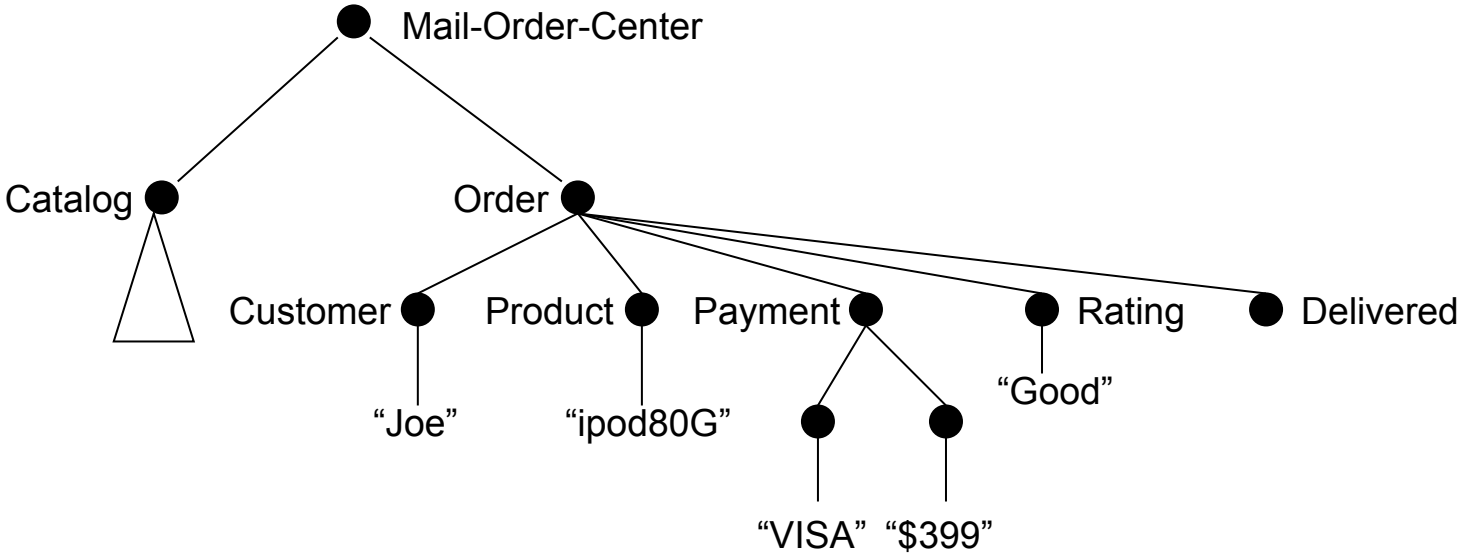
AXML workflow by example



AXML workflow by example



AXML workflow by example



IBM's Business Artifacts

Tuple artifacts:

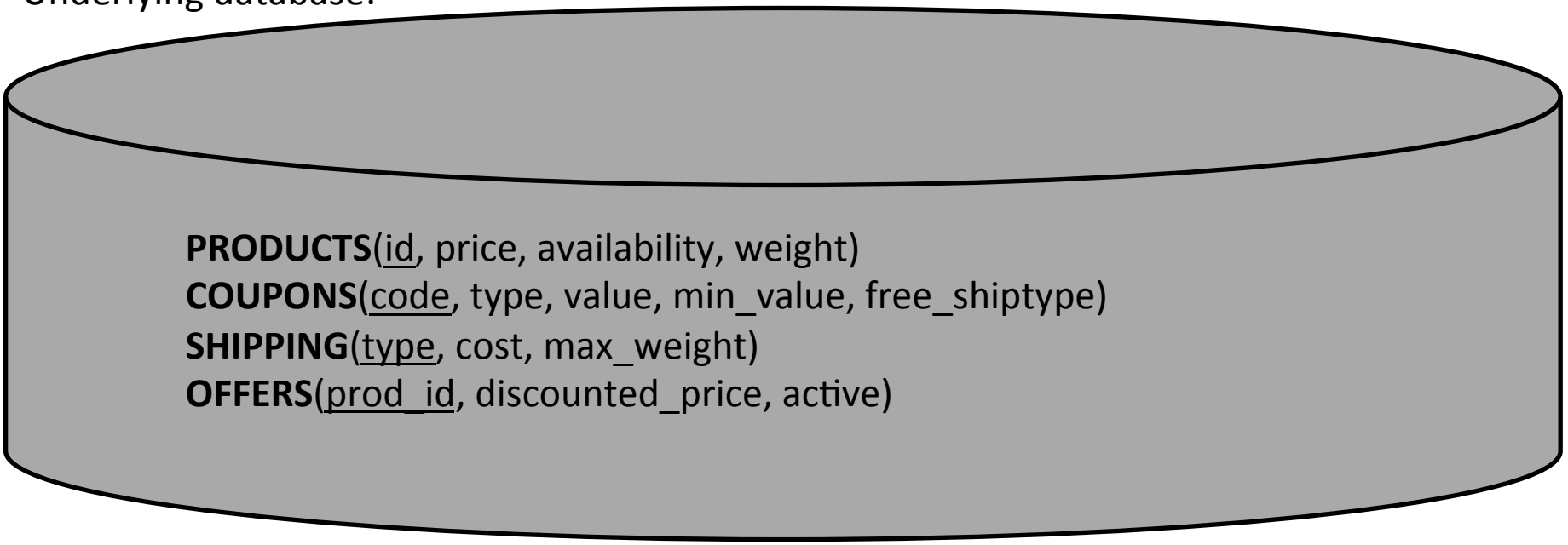
- evolving tuples of data values
- fixed underlying database
- services (pre/post conditions on tuple and a global relational database)

Example

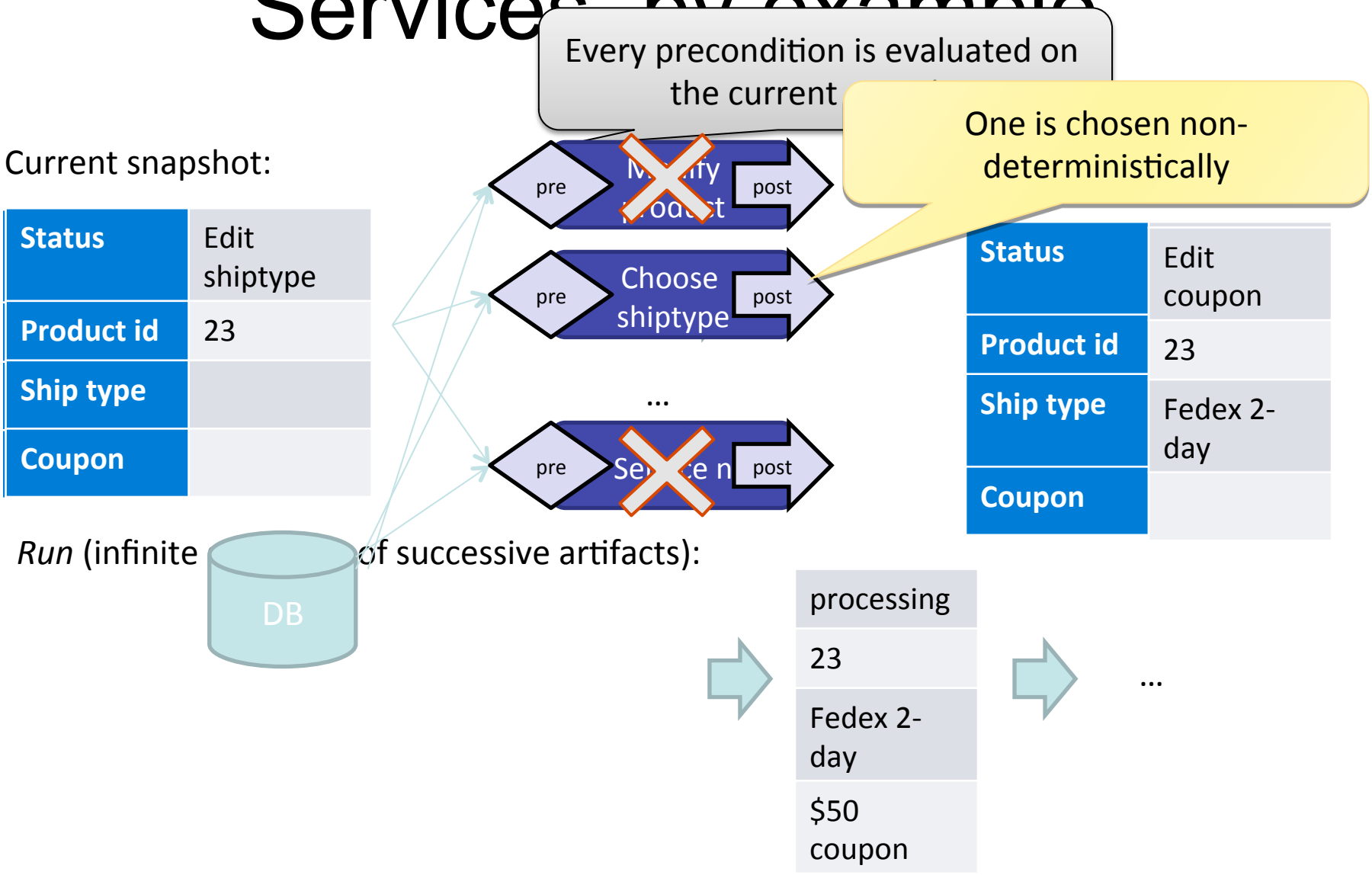
Single artifact record:

Status	Prod_id	Ship_type	Coupon	Amount owed	Amount paid	Amount refunded

Underlying database:



Services, by example



Specifying temporal properties of runs

LTL + statements about data

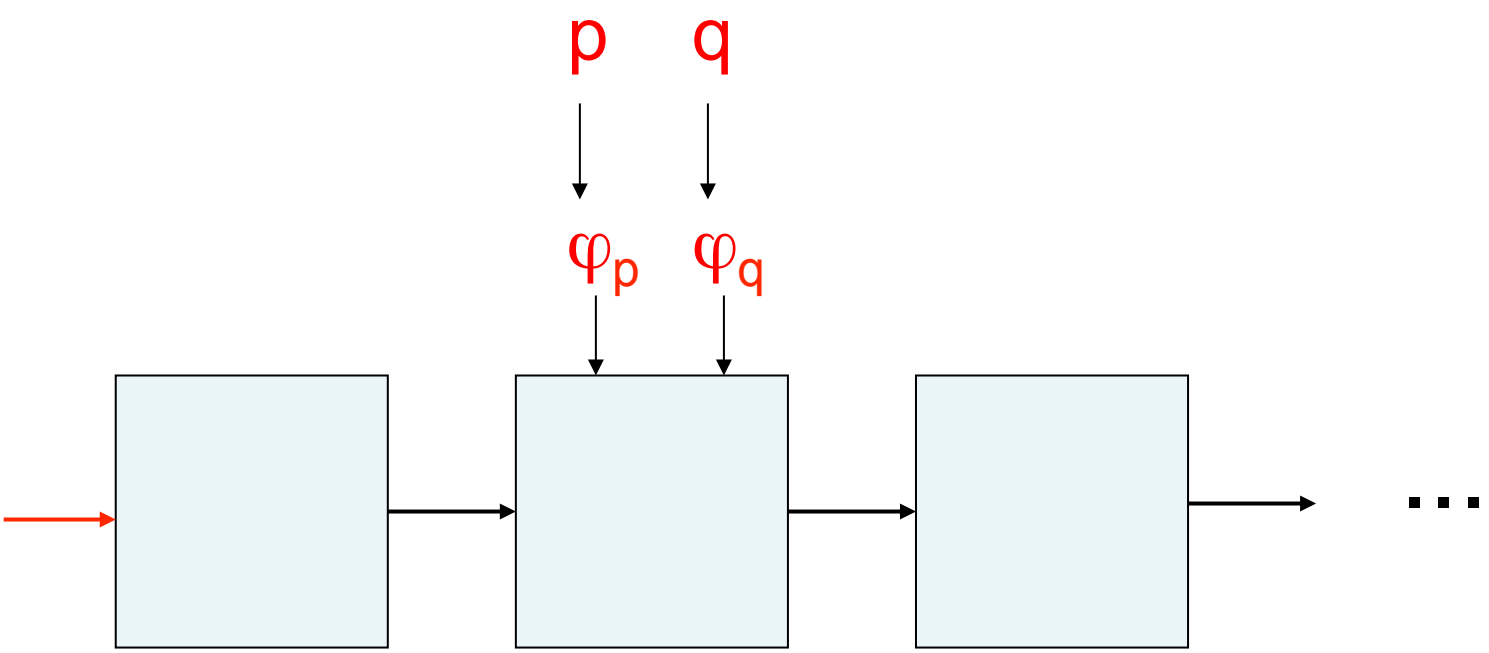
Linear-time temporal logic

always (p → eventually q)

LTL($\mathcal{L}$)

propositions → formulas in a db language

$\mathcal{L}$



LTL(FO) property:

"every paid product is eventually delivered"

$\forall x$ always (p $\rightarrow$ eventually q)

$\exists y$ (pay(x,y) $\wedge$ price(x,y)) delivered (x)

Variant for AXML workflows

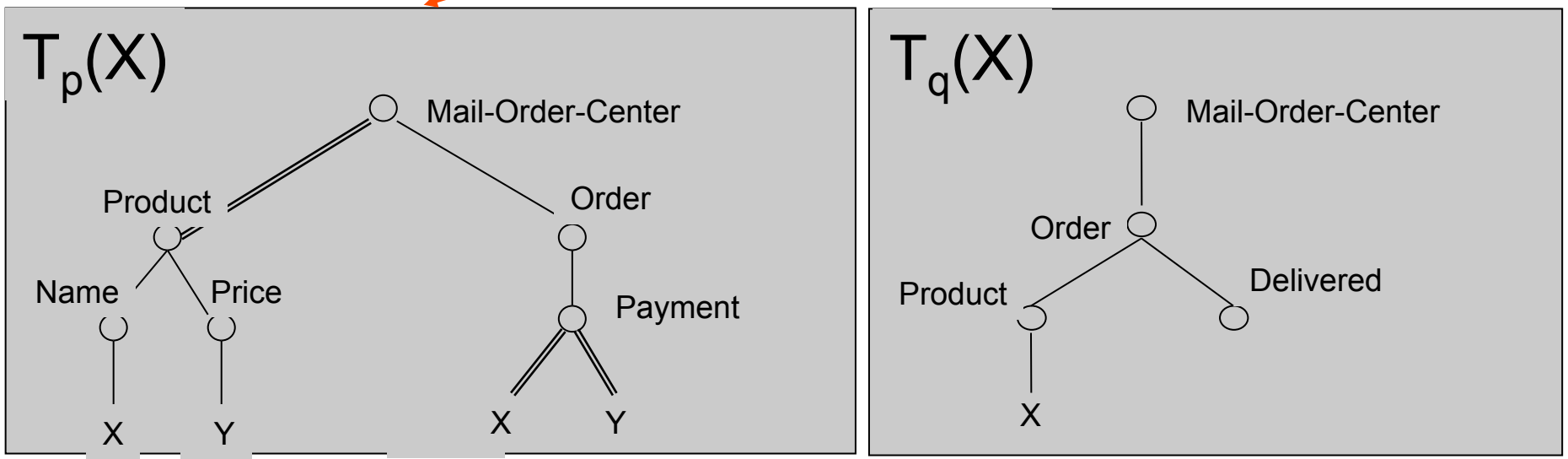
$\forall \bar{X}$ always (p → eventually q)

$T_p(\bar{X})$ $T_q(\bar{X})$

tree patterns with free variables $\bar{X}$

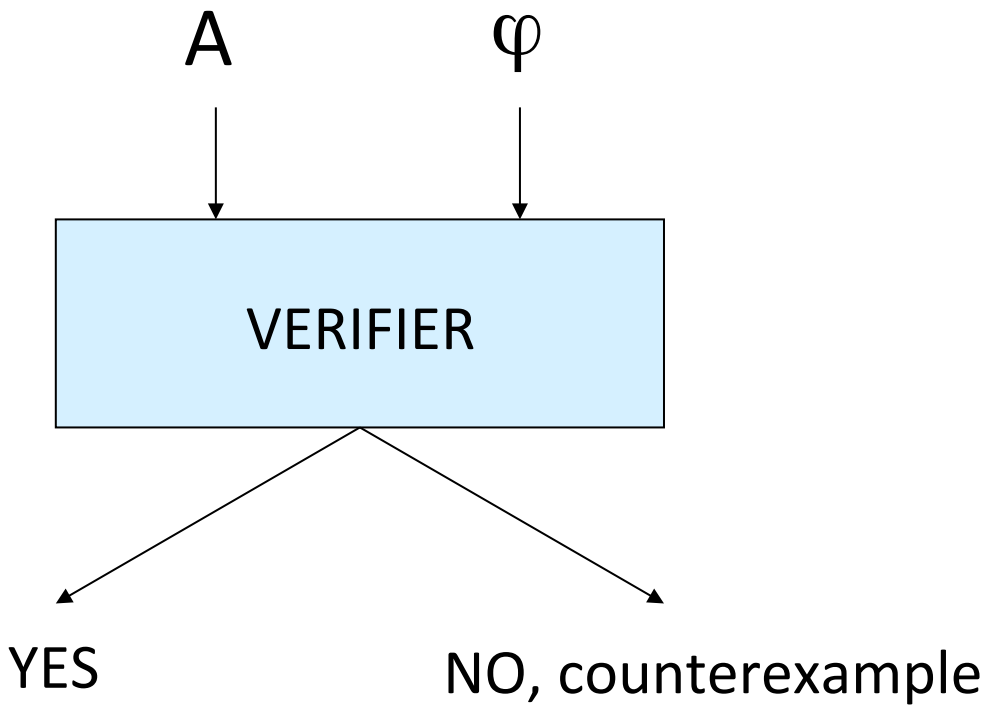
“every paid product is eventually delivered”

∀ X always(p → eventually q)



LTL(XPath)

The verification problem



Results on verification

Restrictions that guarantee decidability

- Business artifacts: UCSD + INRIA + IBM
- Active XML: UCSD + INRIA

Boundary of decidability, complexity

Results on verification

Restrictions that guarantee decidability

- Tuple artifacts: UCSD + INRIA + IBM
- Active XML: UCSD + INRIA

Efficient implementation: WAVE verifier

Symbolic model checking:
marriage of database and CAV techniques

Long collaboration with INRIA

- Automatic Verification

  PODS 2008, ICDT 2009, TODS 2009,

  ICDT 2011, BPM 2011, TODS 2012
- Comparison, customization, optimization of workflows

  ICDT 2011, TODS 2012
- Runtime analysis for collaborative workflows

  PODS 2013

How to compare different workflows?

Building a beehive

Building an ant nest

Use abstraction!

Building a beehive

Building an ant nest

Use abstraction!

- Map to a common abstraction
- Define a notion of simulation

Building a beehive

Building an ant nest

Use abstraction!

Allows to compare
very different workflows

Building a beehive

Building an ant nest

AXML vs Artifacts

- AXML workflows can simulate Artifacts
  with respect to appropriate abstractions

- Artifacts cannot simulate AXML

Long collaboration with INRIA

- Automatic Verification

  PODS 2008, ICDT 2009, TODS 2009,

  ICDT 2011, BPM 2011, TODS 2012
- Comparison, customization, optimization of workflows

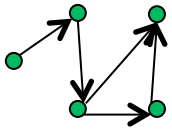
  ICDT 2011, TODS 2012
- Runtime analysis for collaborative workflows

  PODS 2013

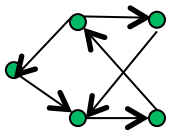
Collaborative workflows are pervasive



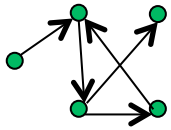

Business processes, health care, scientific workflows, government …



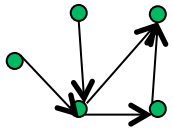
actions & data



actions & data



actions & data



actions & data

Author

PC chair

PC member

Referee

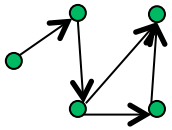
Some possible actions

Author: submit, respond, final, copyright

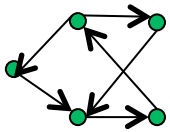
PC chair: assign, discuss, decide, notify, remind

PC member: enter review, invite referee, discuss

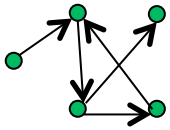
The collaborative workflow model



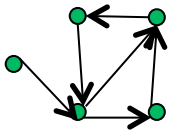
actions & data



actions & data



actions & data



actions & data

Author

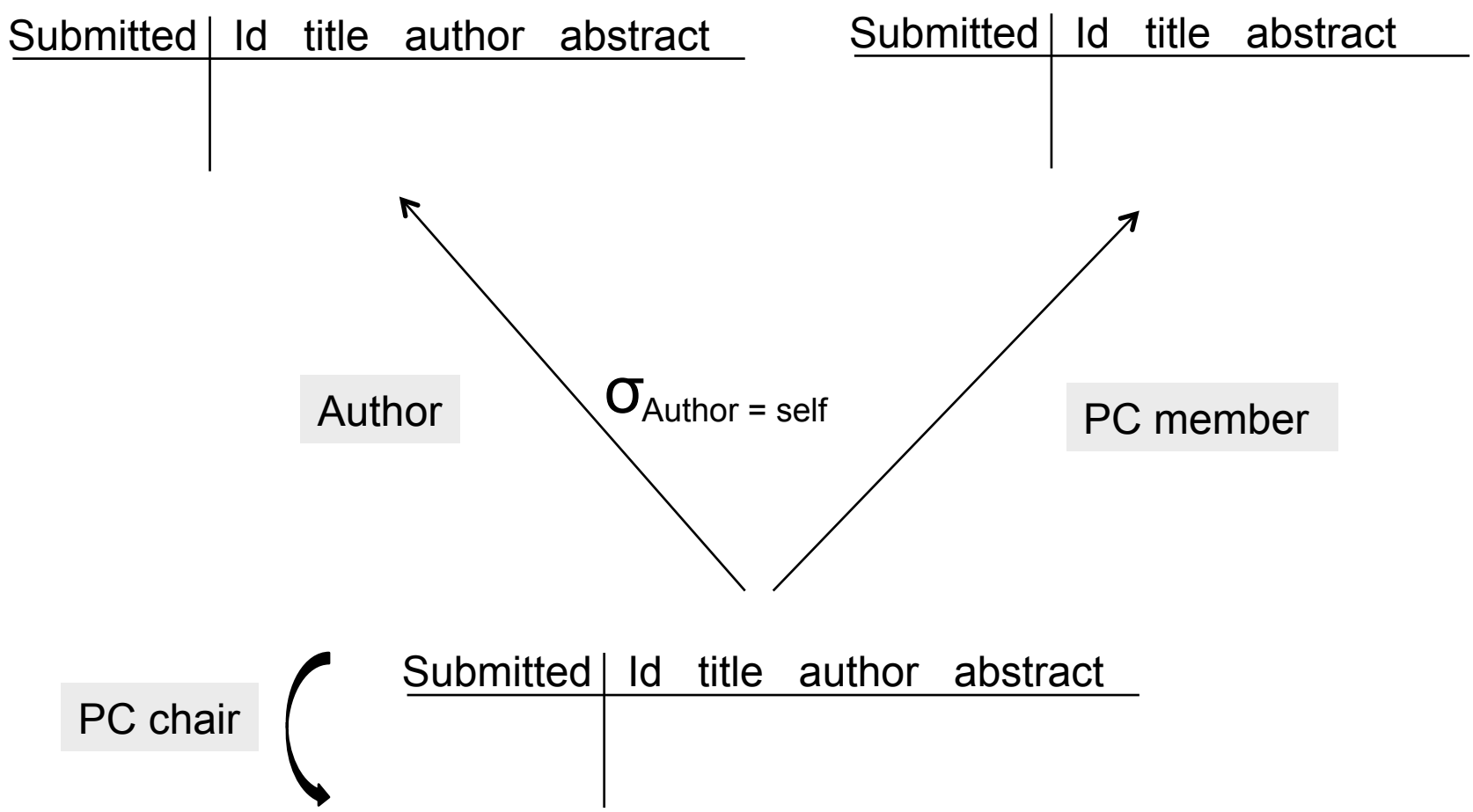
PC chair

PC member

Referee

Simple peer views

virtual database


Submitted | Id title author abstract
Submitted | Id title abstract
Author
σAuthor = self
PC member
PC chair
Submitted | Id title author abstract

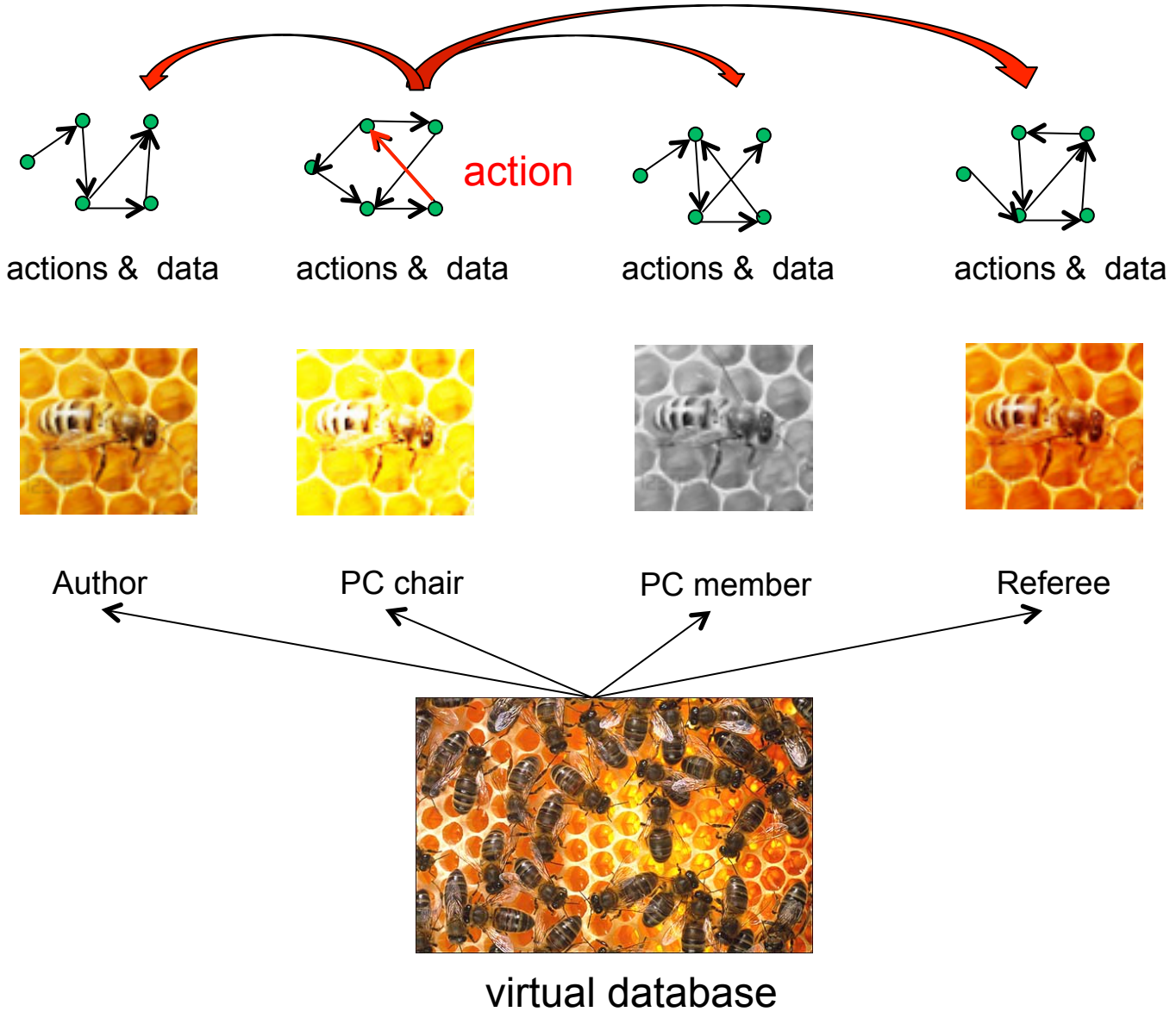
Peer actions

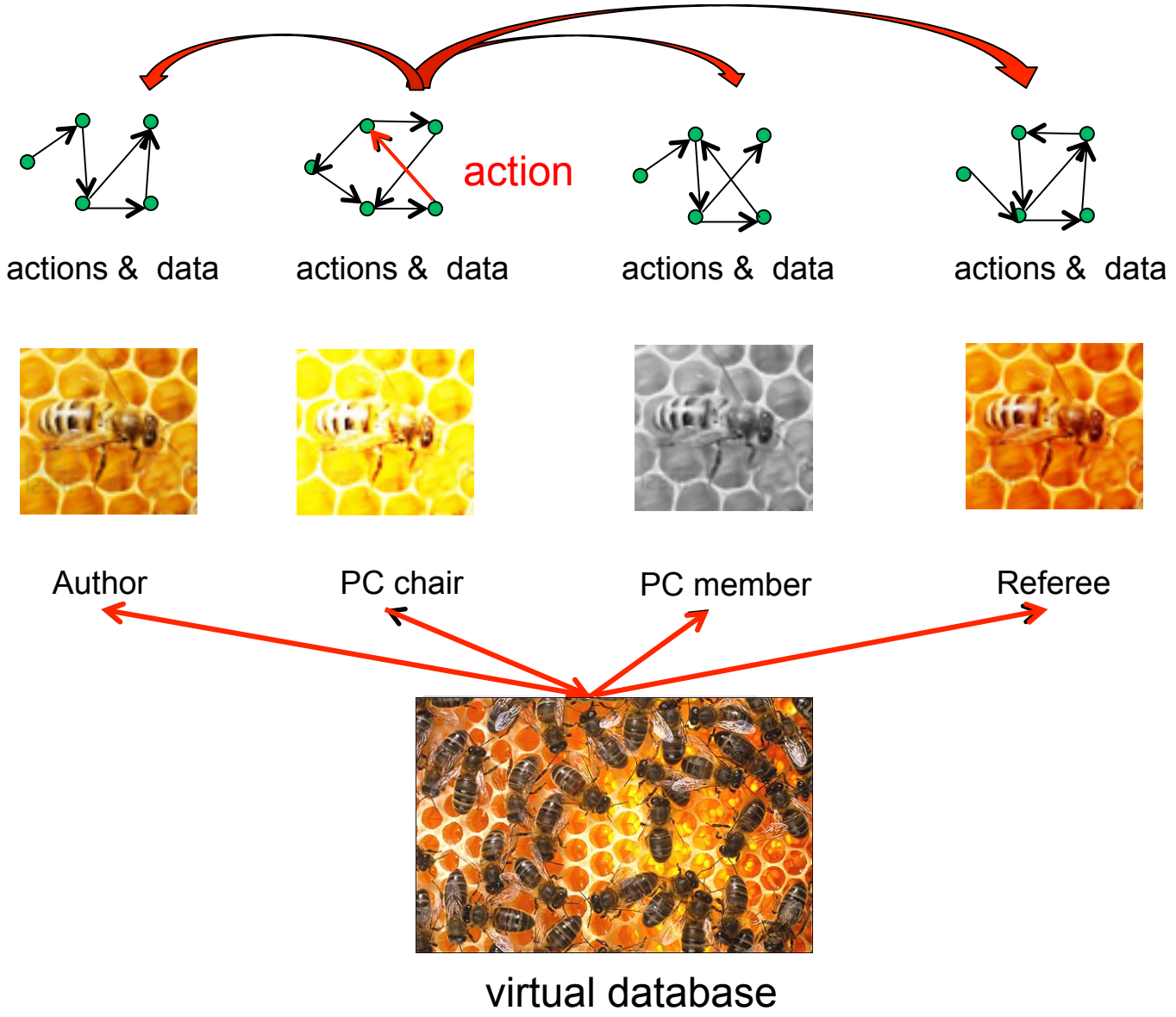
Update :- *Condition*

sequence of insertions/deletions into the view

conjunction of literals in the view

PC Chair Assign(Id, PCmember), Status(Id, assigned)
:- Submitted(Id, title, author, abstract), ¬ Status(Id, assigned)
PC(PCmember), ¬ Conflict(author, PCmember)


action
actions & data
actions & data
actions & data
actions & data
Author
PC chair
PC member
Referee
virtual database


action
actions & data
actions & data
actions & data
actions & data
Author
PC chair
PC member
Referee
virtual database

Focus:
runtime peer reasoning about global run based on local observations

- Monitoring events

  Has my paper been accepted?

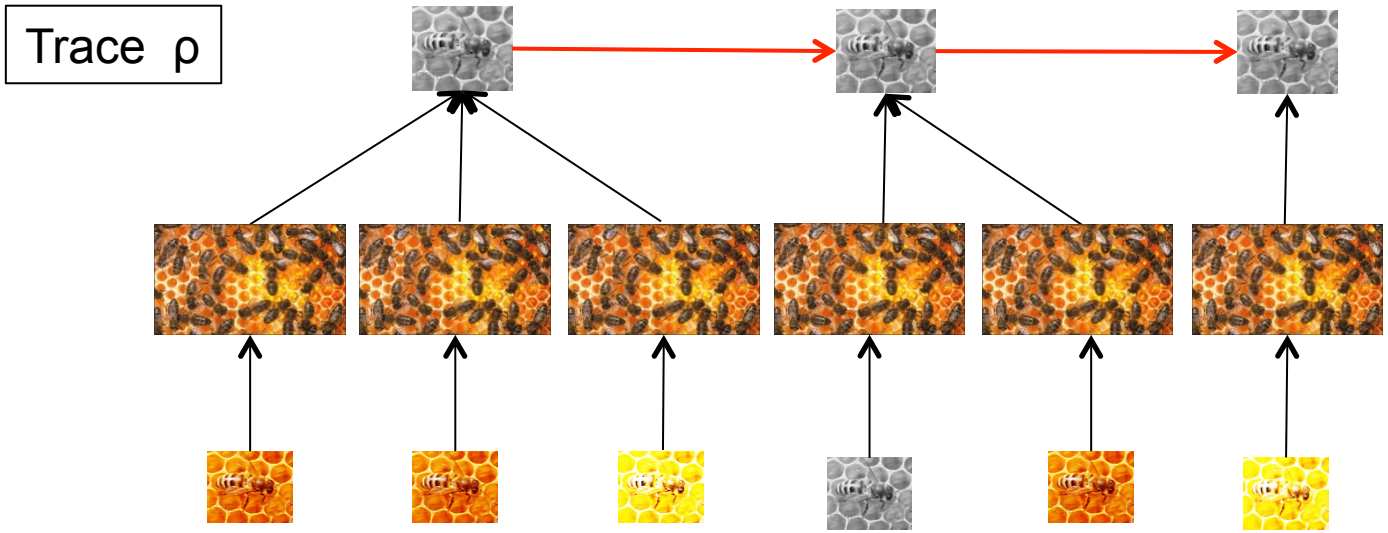
- Diagnosing anomalous behavior

  A paper was assigned to a PC member with a COI: could the assignment have been made by the PC chair, or was it necessarily made by a track chair?

- Competitive advantage

  Is my own submission as a PC member still under discussion?

Runtime peer reasoning about global run based on local observations



Infinitely many compatible global runs

Specifying properties of global runs
PLTL-FO

PLTL:

past linear-time propositional temporal logic

PLTL-FO:

PLTL with propositions interpreted as FO statements

Specifying properties of global runs
PLTL-FO

- My paper (submission 12) has not yet been accepted nor rejected:

always $^{-1}$ $\neg \exists$ review (Accept(12, review) $\vee$ Reject(12, review))

Specifying properties of global runs
PLTL-FO

- My paper (submission 12) has not yet been accepted nor rejected:

always $^{-1}$ $\neg \exists$ review (Accept(12, review) $\vee$ Reject(12, review))

- Semantics of a formula φ: relative to a trace ρ

Cert(φ) : φ holds in every global run compatible with ρ

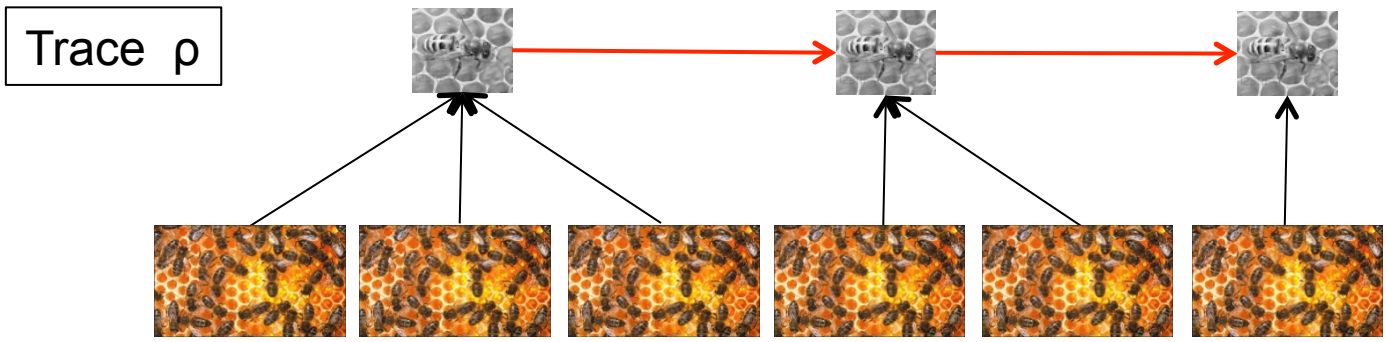
Poss(φ) : φ holds in some global run compatible with ρ

Goal

Given a workflow specification W,
a trace ρ and an PLTL-FO property φ of global runs
evaluate Cert(φ) and Poss(φ)

Theorem: Cert(φ) and Poss(φ) can be evaluated
in PSPACE with respect to φ and ρ.

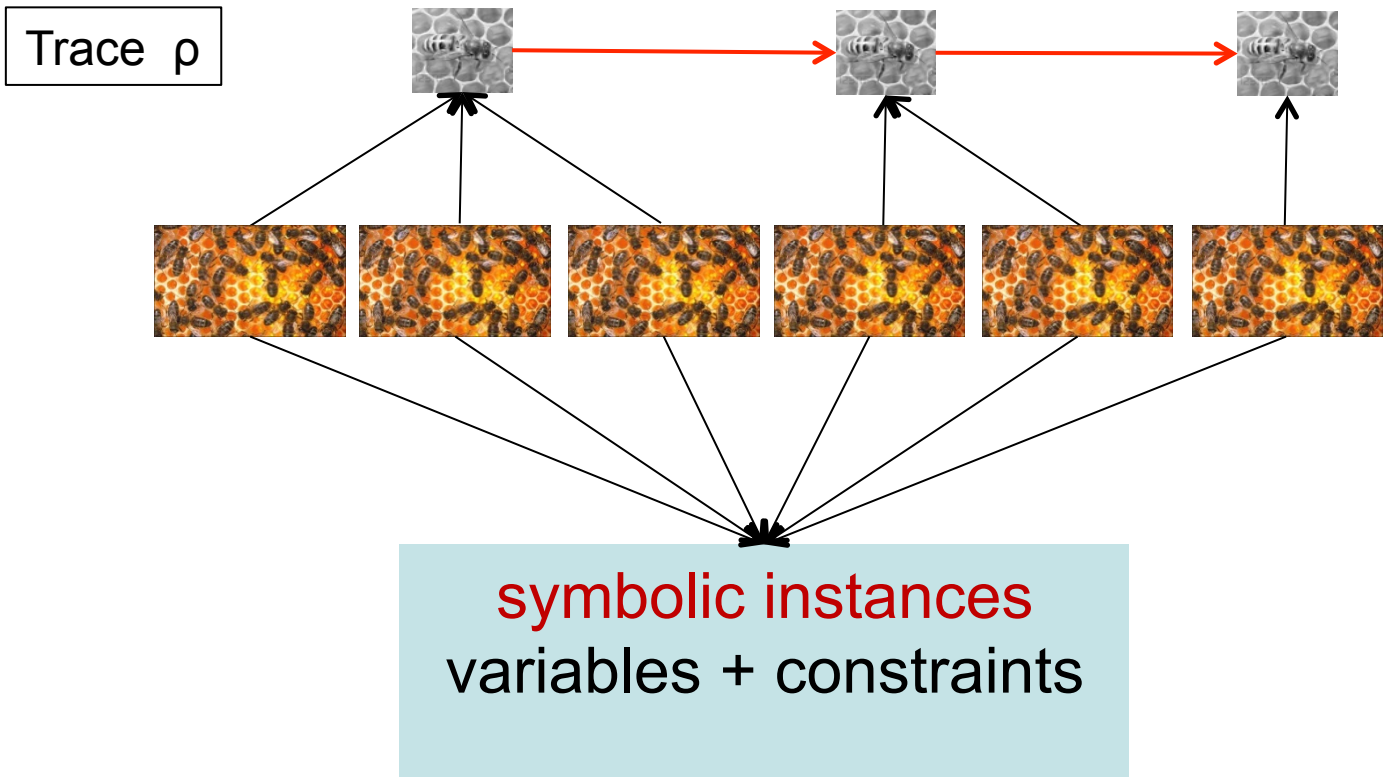
Proof idea



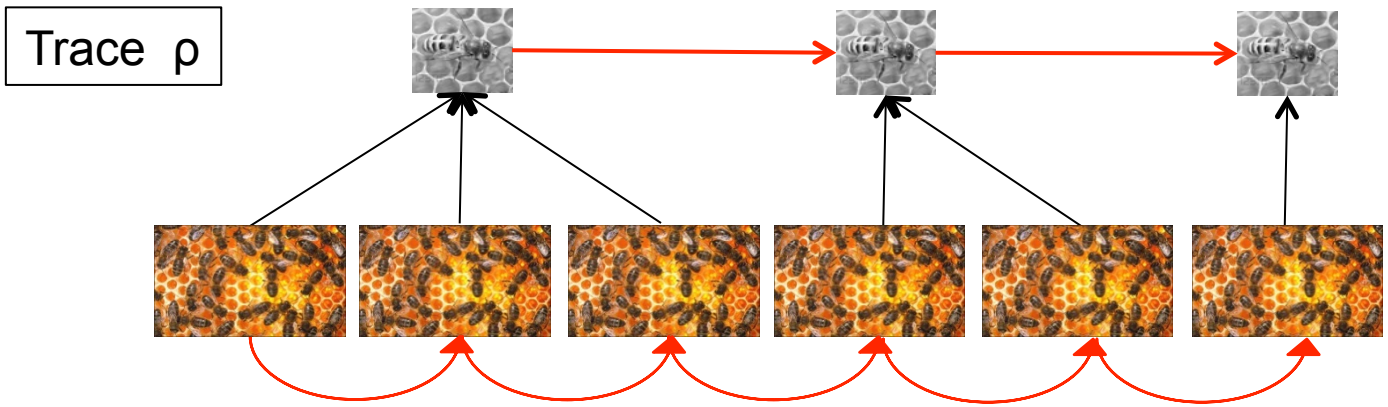
Infinitely many compatible global runs

Symbolic transition system

Proof idea

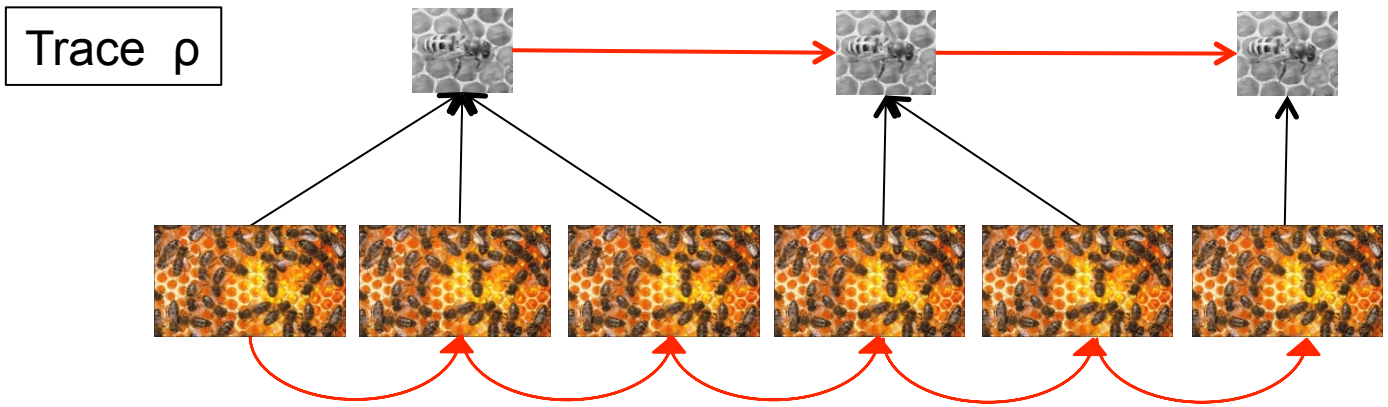


Proof idea



Transitions
among symbolic instances
each computed in PSPACE

Proof idea



Finite transition system on which
Poss(φ) can be evaluated
by a nondeterministic PSPACE search

$$\mathrm{Cert}(\varphi) \;=\; \neg\, \mathrm{Poss}(\neg\, \varphi)$$

Other results

- Incremental evaluation
- Pre-emptive monitoring

φ is not known a priori

Can be done for classes of properties sharing a temporal type: underlying propositional PLTL formula

Example: **always^{-1} (p → sometime $^{-1}$ q)**

FO FO

Other results

- Incremental evaluation
- Pre-emptive monitoring
- Using knowledge about global run in workflow specifications

PC member

As long as I don't know that my paper is accepted, I will reject every other paper

Reject(ID) :- paper(ID, author), author ≠ *self,*
¬ cert ("my paper is accepted")

epistemic atom

Summary

- Models for data-centric workflows
- Automatic verification
- Framework for comparing workflows
- Runtime analysis for collaborative workflows

Better tools for the design and analysis of data-centric workflows

Summary

- Models for data-centric workflows
- Automatic verification
- Framework for comparing workflows
- Runtime analysis for collaborative workflows

Exciting area, much more to be done!

Thank you !

Merci !