From State- to Delta-Based Bidirectional Model Transformations: The Symmetric case 10 years after

Zinovy Diskin, Yingfei Xiong, Krzysztof Czarnecki,
Hartmut Ehrig, Frank Hermann, Fernando Orejas

In memory of Hartmut Ehrig who passed away on March 17, 2016

Models’21 MIP Talk, Fukuoka, Oct 13, 2021

How we got together

M. Antkiewicz & K.Czarnecki, **Design Space of Heterogeneous Synchronization** @ GTTSE'07

Krzysztof

Fernando

Two papers at Models'11

Zinovy

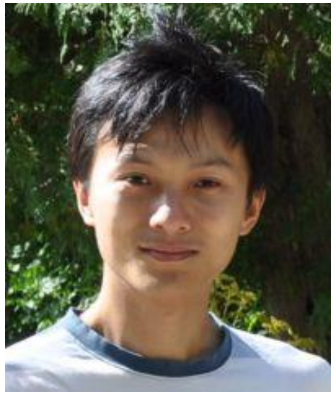
Waterloo, 2007-11

Waterloo-Berlin-Barcelona, 2011

Dagstuhl, Dec 2010

Hartmut

Z.Diskin, Y.Xiong & K.Czarnecki: **From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case** @ ICMT'10



Yingfei

Frank

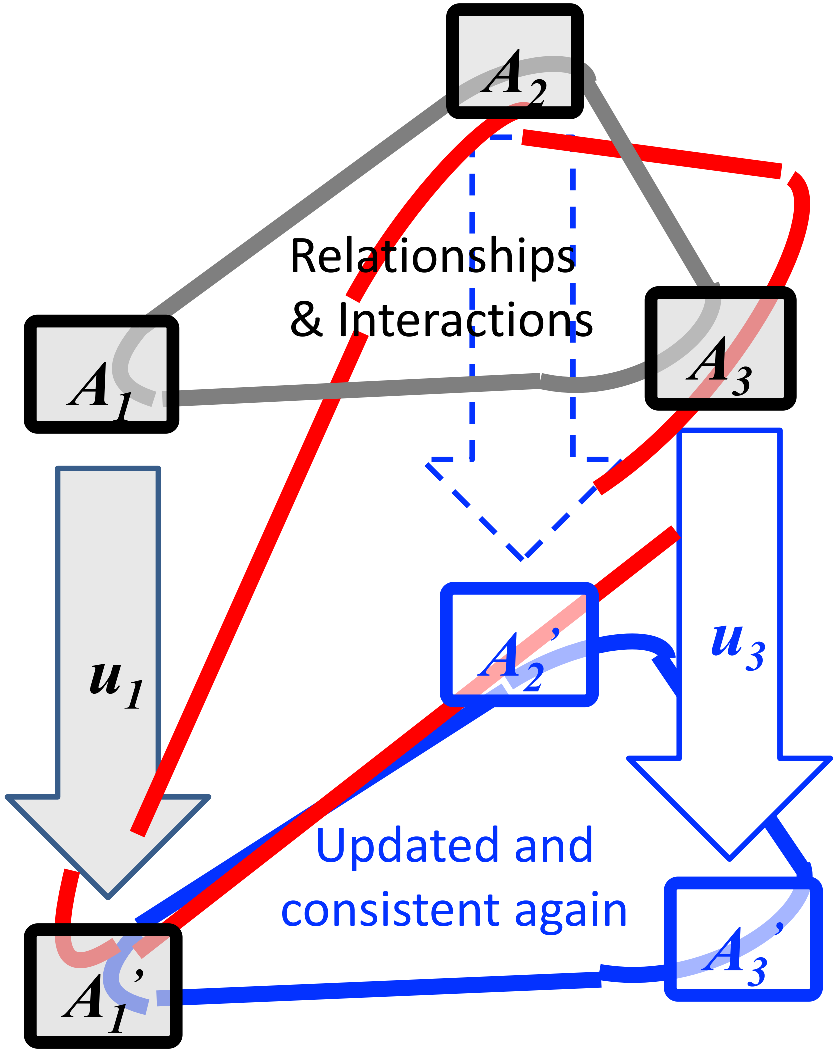
F. Hermann, H. Ehrig, F. Orejas, U. Golas: **Formal Analysis ...for Model Transformations Based on TGG @** ICGT'10

Content

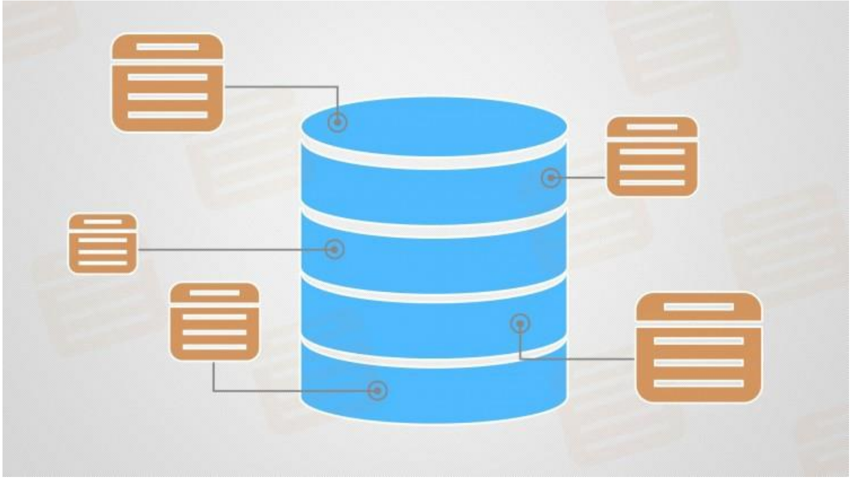
- Background
- Our contribution: Why deltas
- Sync in the large: Multi-ary delta lenses (mx) and lens composition
- Applications in DB, SE, PL
- Applications in ML and AI: looking forward

Change propagation is everywhere

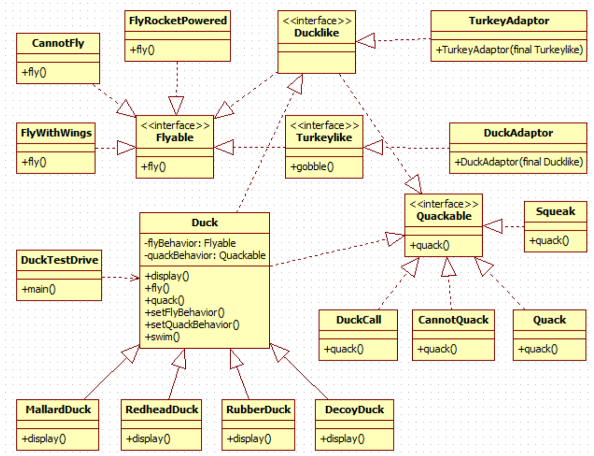
- Our world consists of interrelated objects
- Objects are subjected to changes that can violate consistency between objects
- Consistency is to be restored by change propagation
- Mathematical models of change propagations are often called *lenses*



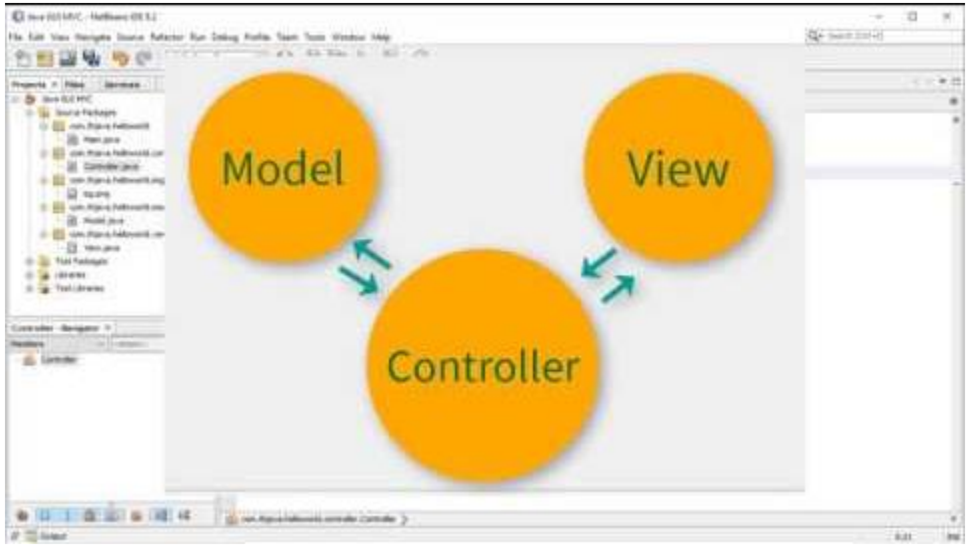
Typical Cases of Change Propagation



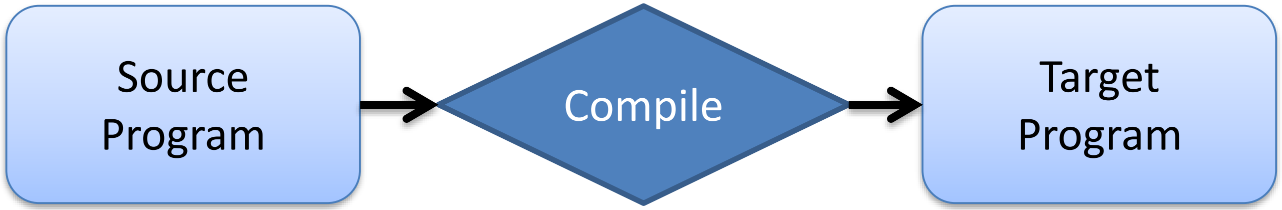
Database and its Views



UML Diagrams and Code

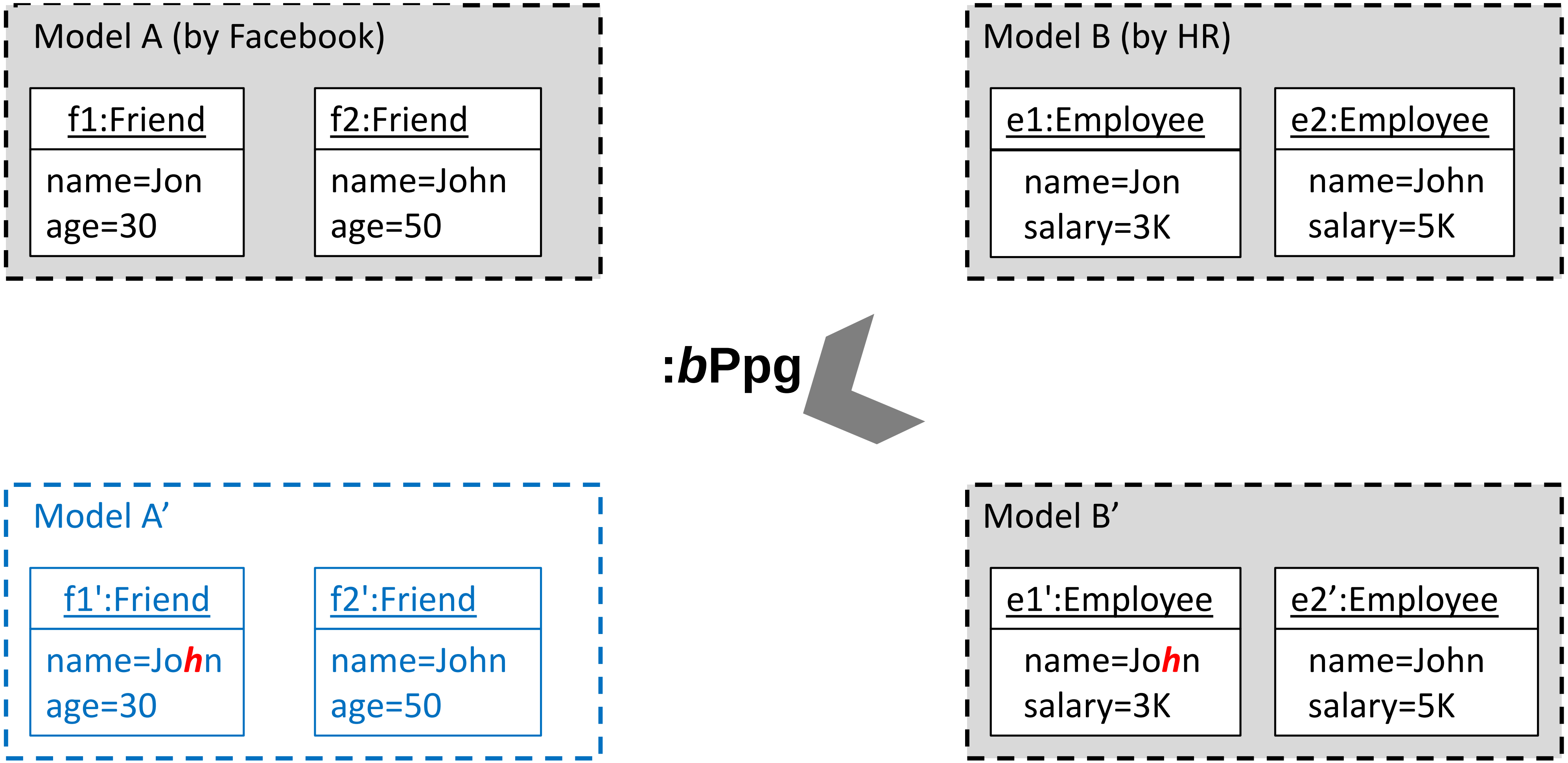


GUI and its underline model



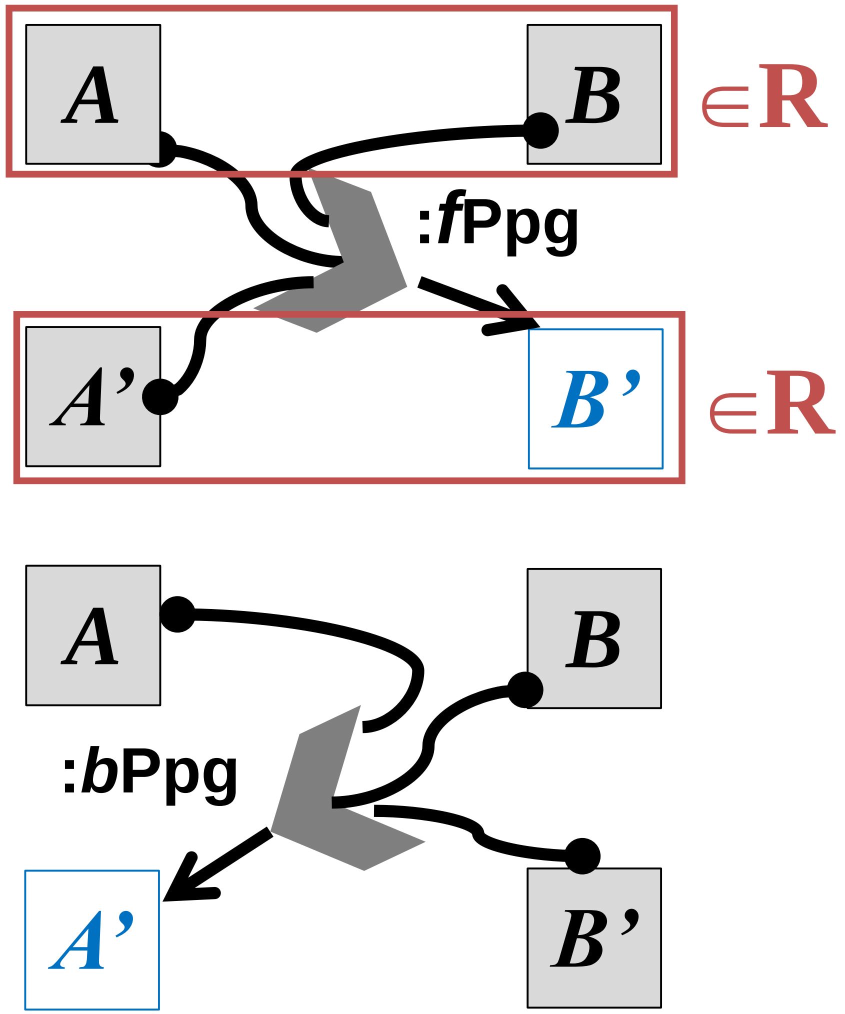
Source and target programs

Example


Model A (by Facebook)
f1:Friend
name=Jon
age=30
f2:Friend
name=John
age=50
Model B (by HR)
e1:Employee
name=Jon
salary=3K
e2:Employee
name=John
salary=5K
:bPpg
Model A’
f1':Friend
name=John
age=30
f2':Friend
name=John
age=50
Model B’
e1':Employee
name=John
salary=3K
e2’:Employee
name=John
salary=5K

State-based Bidirectional Transformation (Bx)

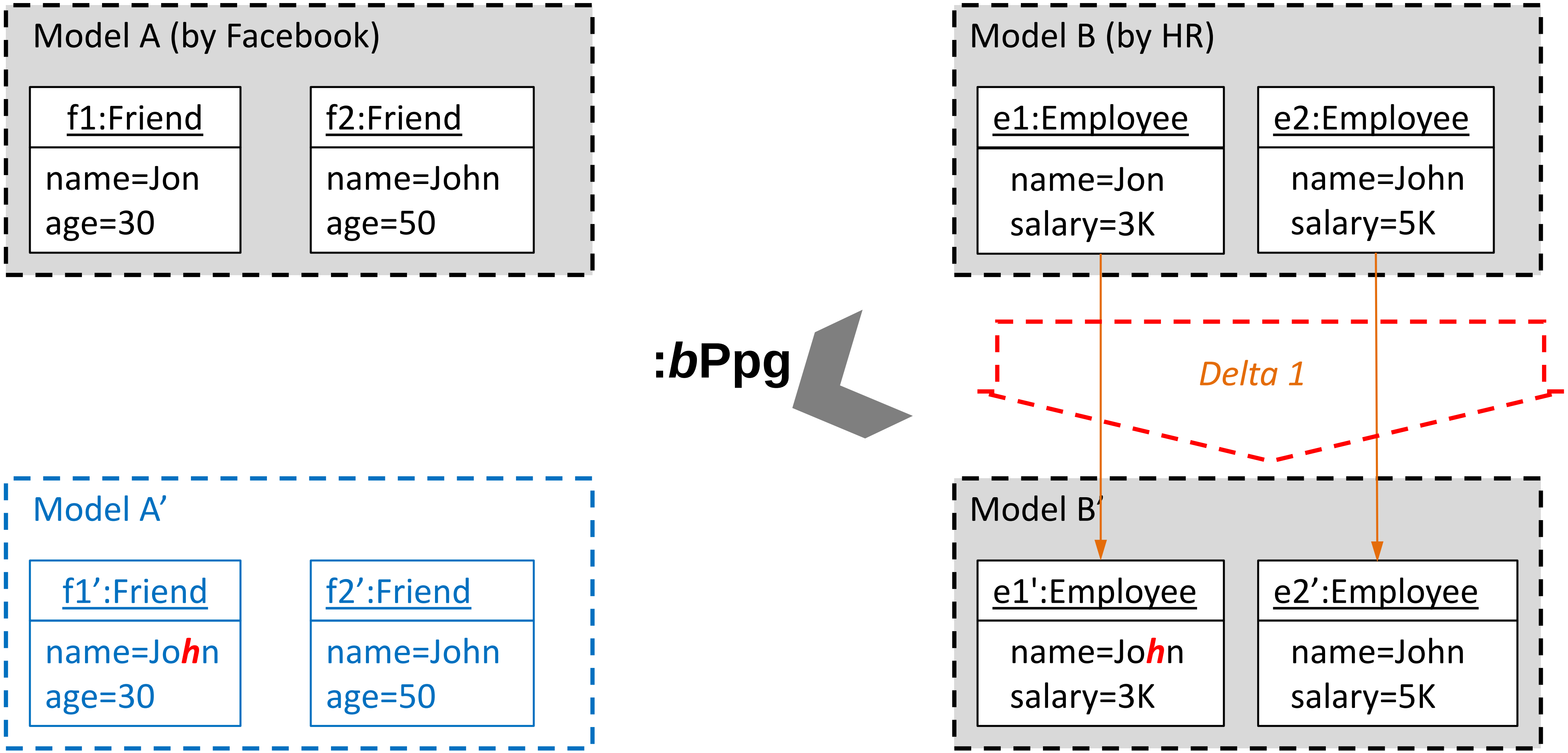
- Consistency Relation
  - ***R:A×B***
- Forward Propgation
  - ***fPpg : A×A×B→B***
- Backward Propagation
  - ***bPpg : A×B×B→A***



Content

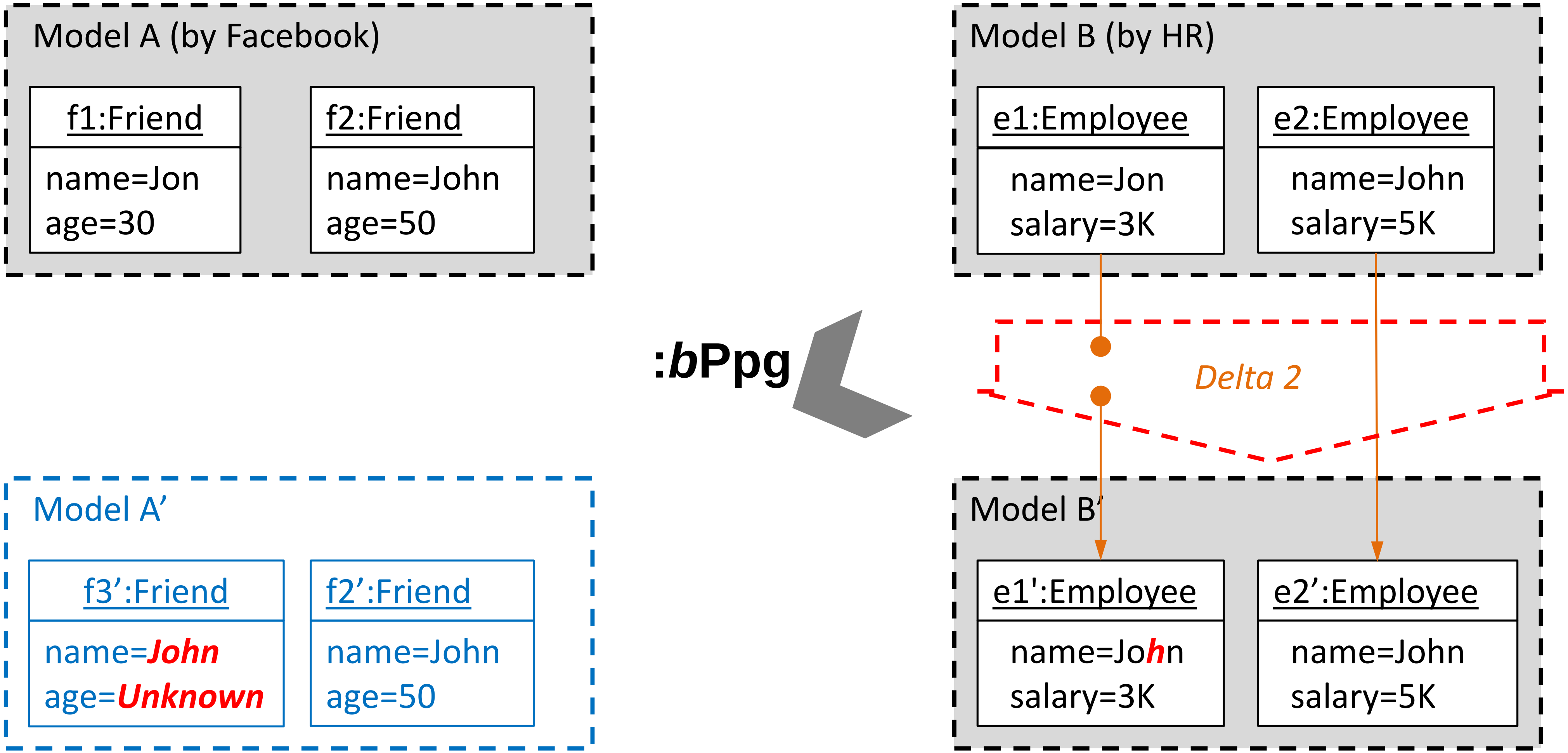
- Background
- Our contribution: Why deltas
  - Problems of state-based Bx
  - Delta-based Bx and Laws
  - Big Picture
- Sync in the large: Lens composition and multi-ary delta lenses (mx)
- Applications in DB, SE, PL
- Applications in ML: looking forward

Update Interpretation: Delta 1

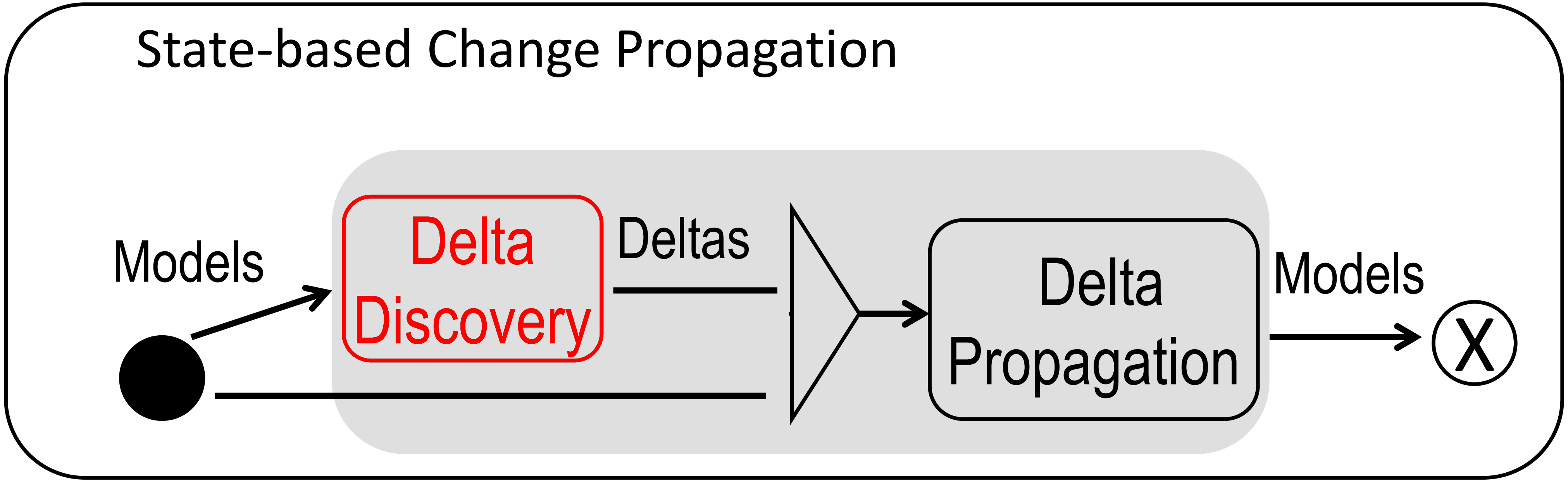


Delta was called "vertical delta" in the paper.

Update Interpretation: Delta 2

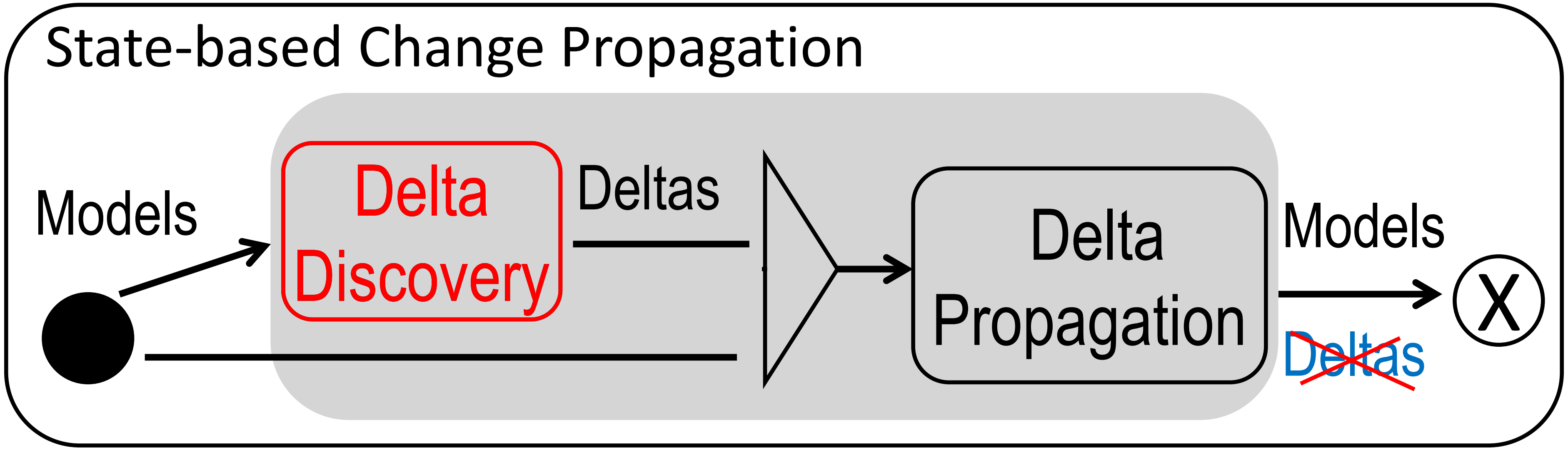


Problem 1 of State-based BX



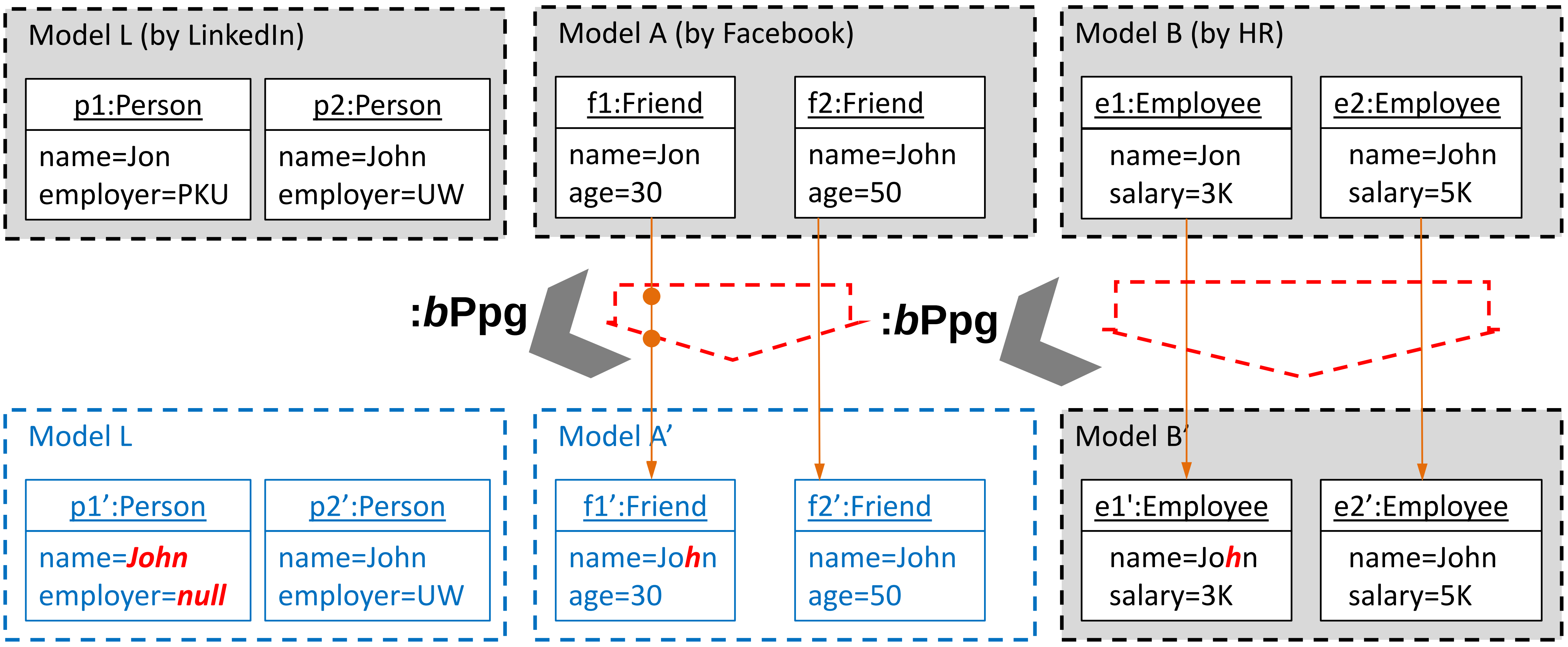
- Semantics of DD is mixed into DP, complicating the latter

Problem 2 of State-based Bx

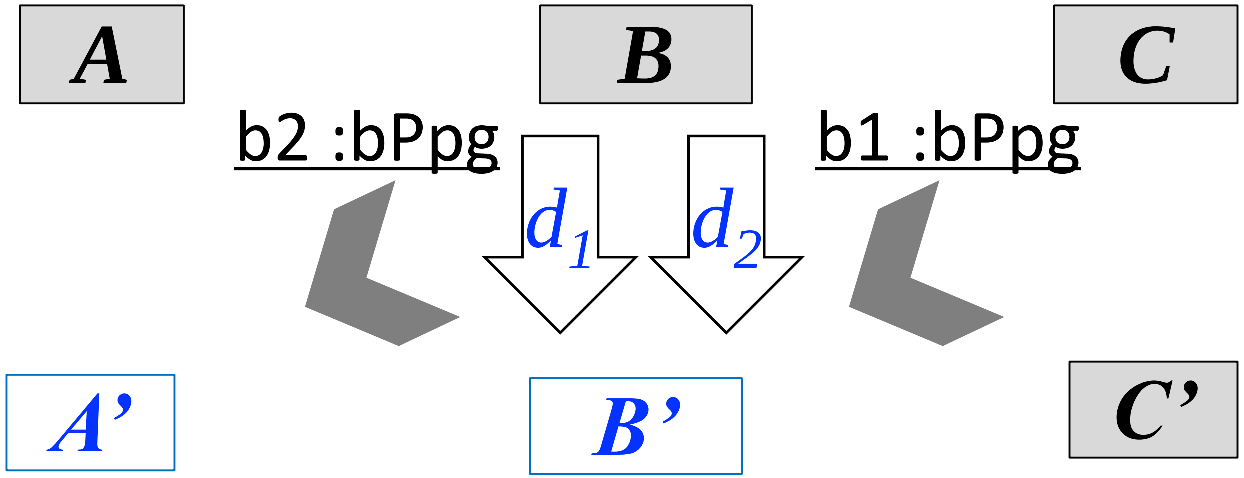


- New deltas are discarded, causing composition problem

Inconsistent Horizontal Composition



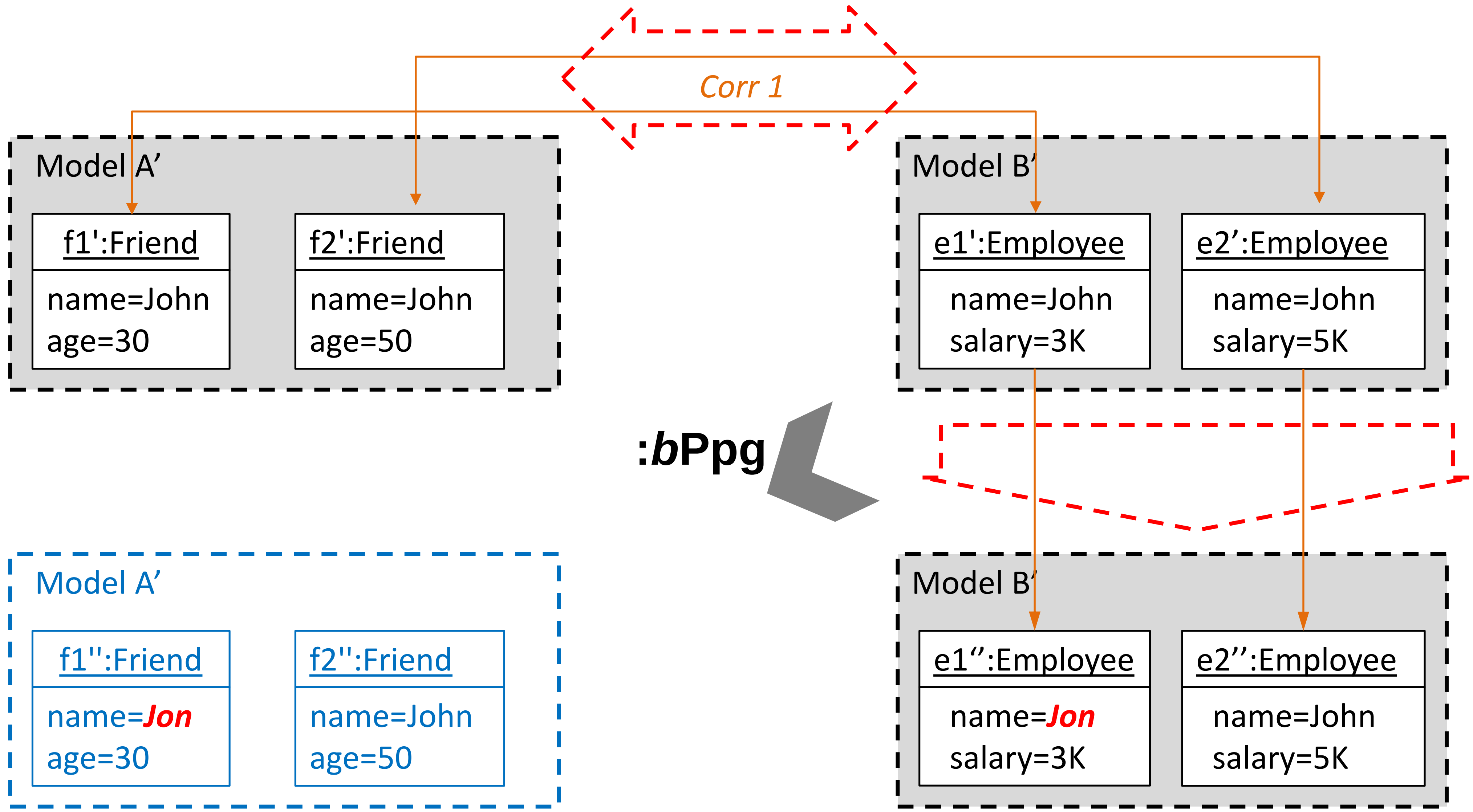
Inconsistent Horizontal Composition



Efficiency: b2 has to compute the delta again

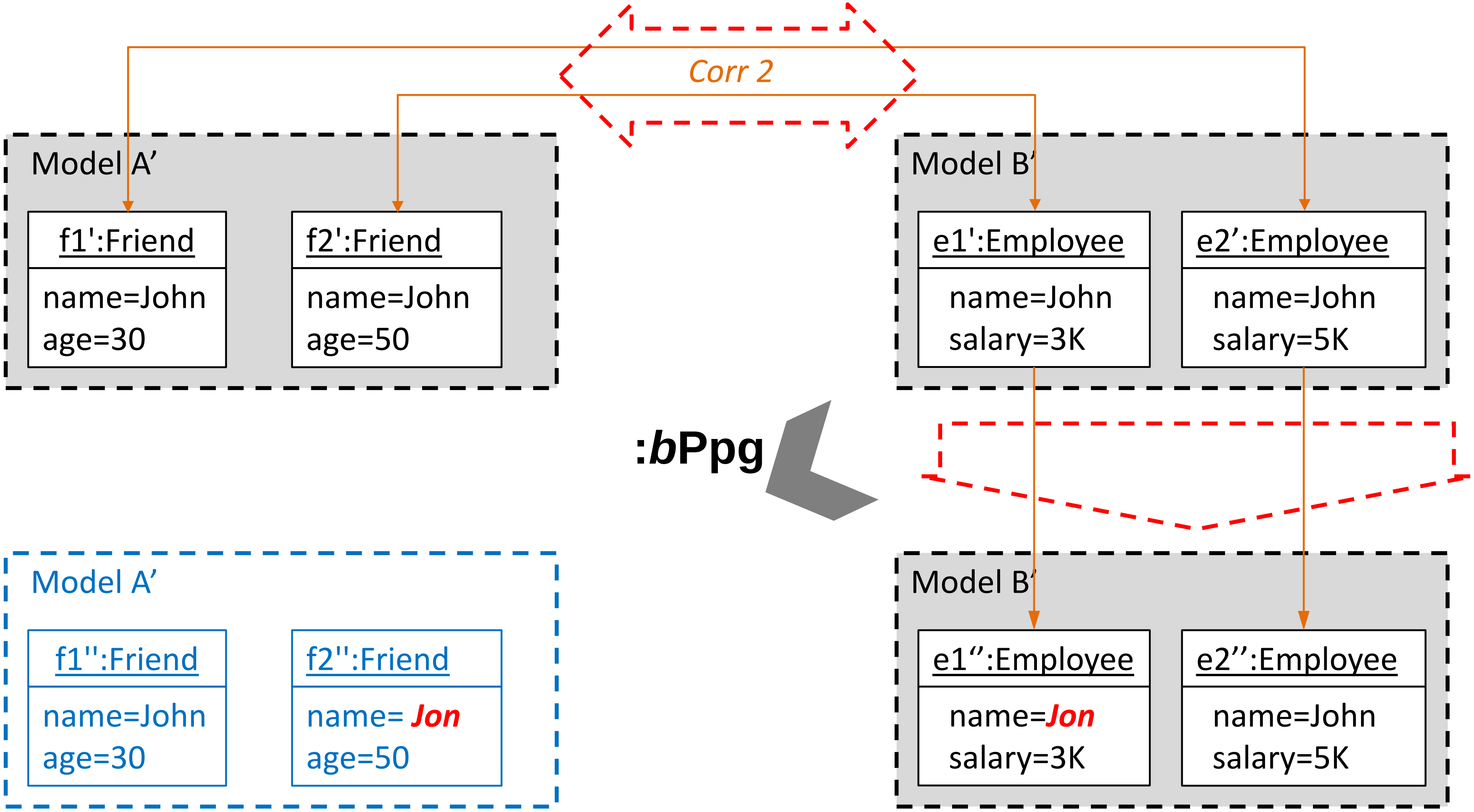
Semantics: b2 may compute a different delta

Correspondence Interpretation: Corr 1

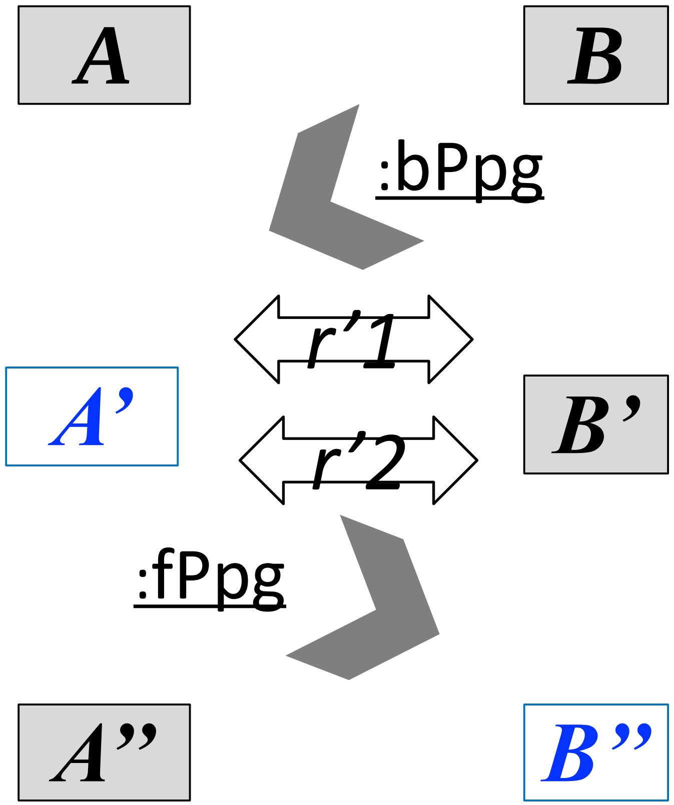


Correspondence was called “horizontal delta” in the paper and was sometimes called “traceability” in other papers.

Correspondence Interpretation: Corr 2



Erroneous vertical composition

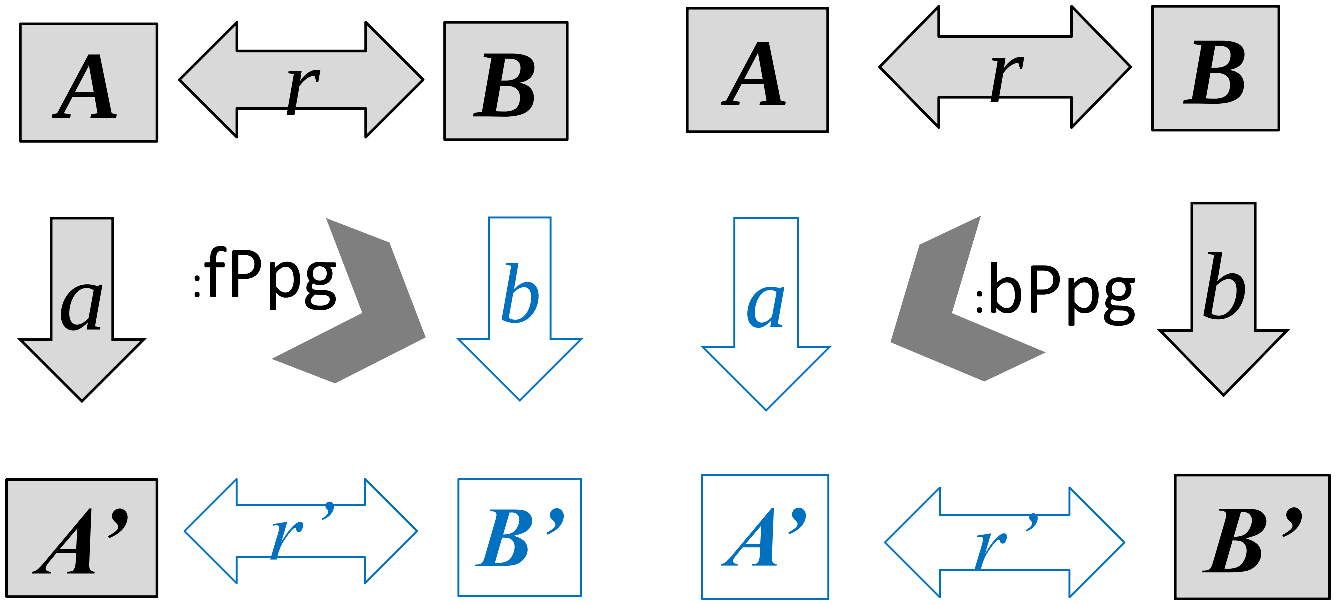


Content

- Background
- Our contribution: Why deltas
  - Problems of state-based Bx
  - Delta-based Bx and Laws
  - Big Picture
- Sync in the large: Lens composition and multi-ary delta lenses (mx)
- Applications in DB, SE, PL
- Applications in ML: looking forward

Delta-based Bx

R: a set of deltas between two model spaces

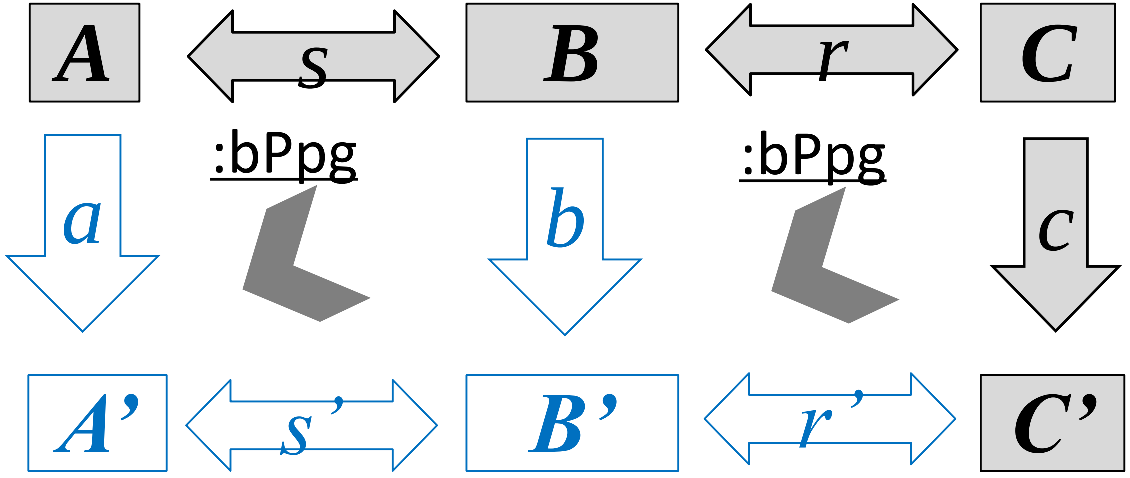


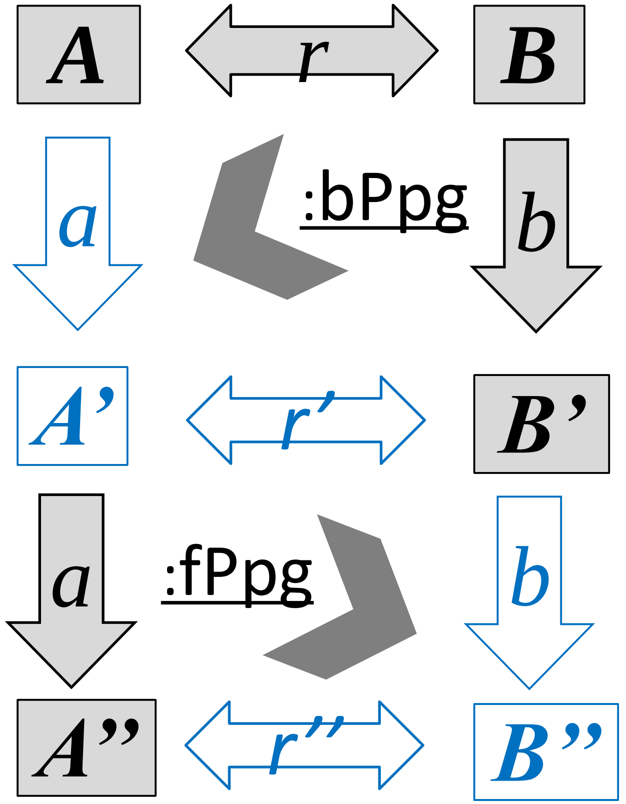
Forward update propagation

Backward update propagation

Benefits 1

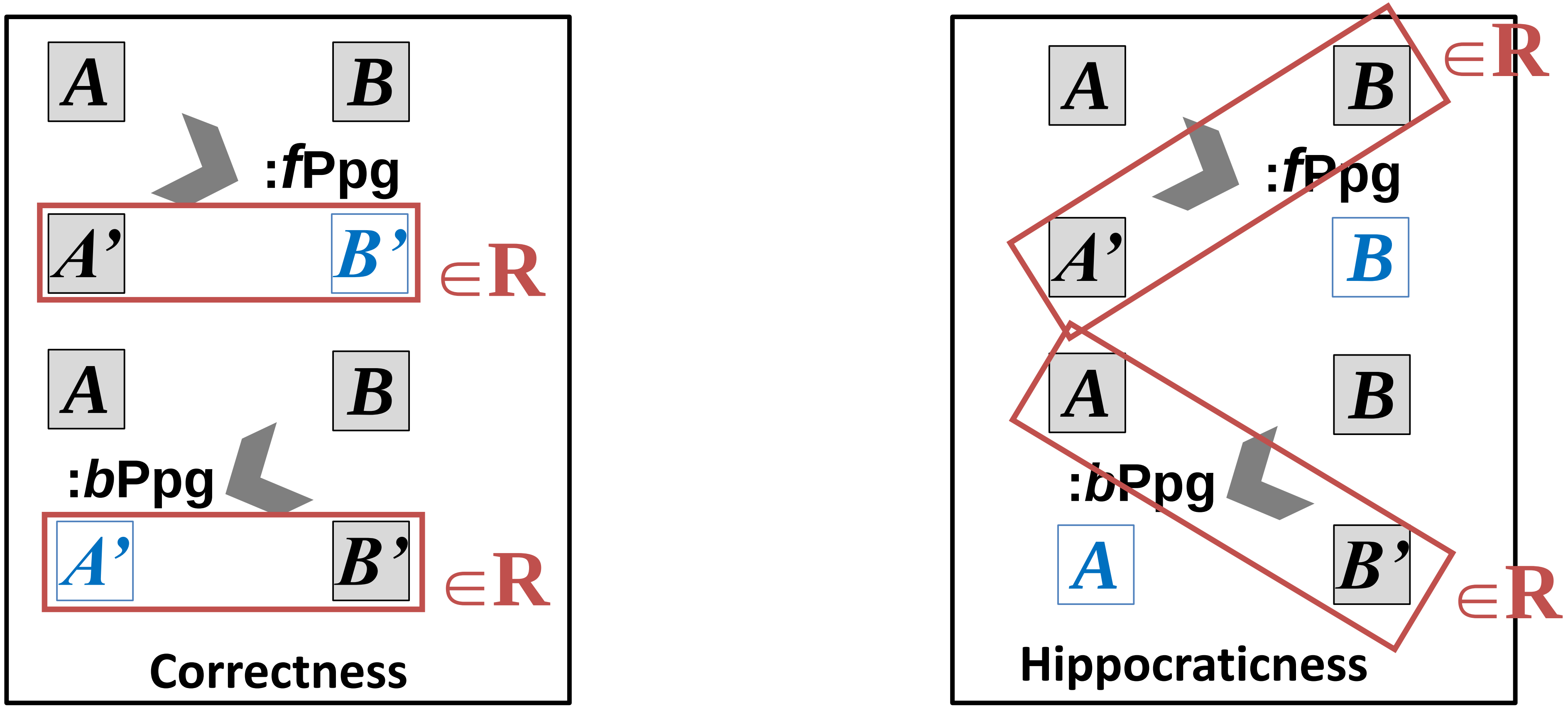
- Semantics of **delta discovery** and **propagation** are separated
  - When state-based BX is needed, we can add a DD component.
- No composition problem





Laws

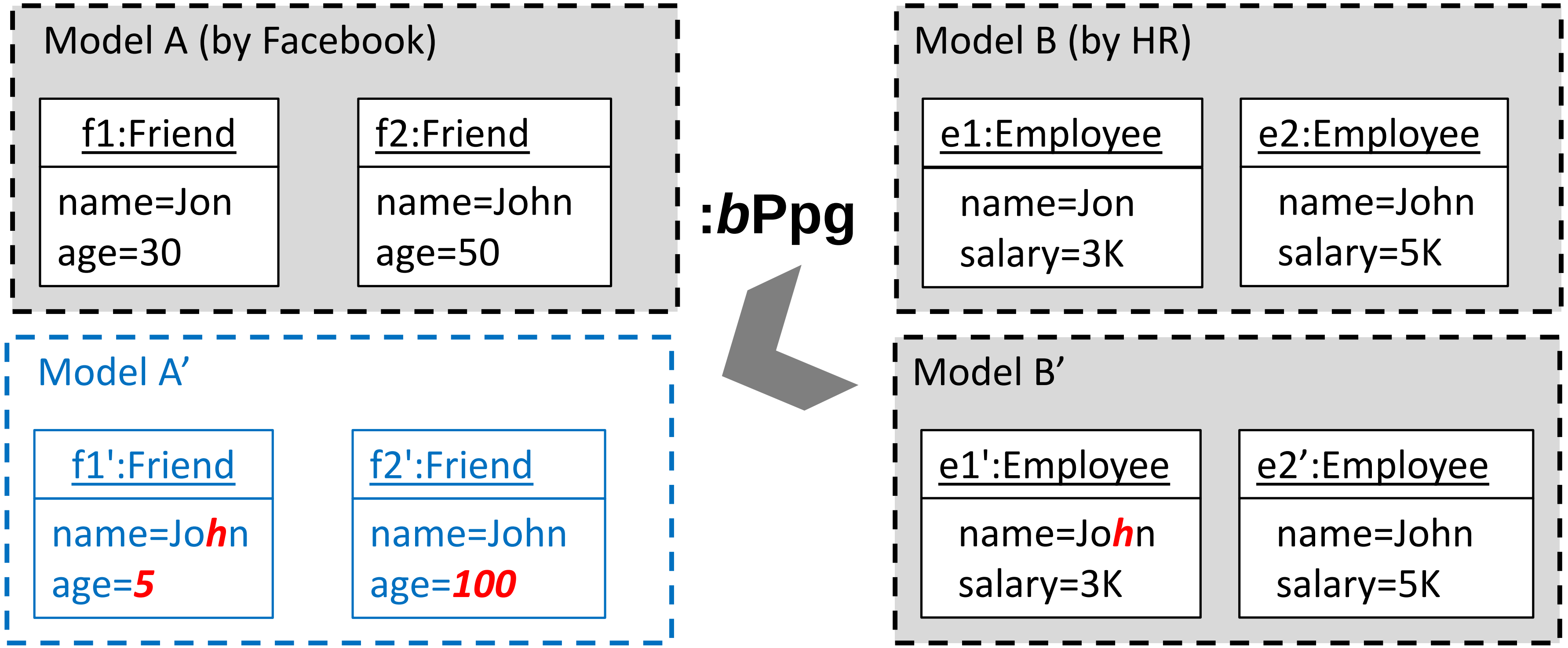
- Laws define the behavior of Bx



- Well-behaved Bx: satisfying the above laws

Benefits 2

- Laws are needed to prevent unnecessary changes



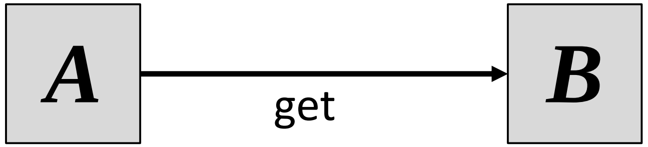
- Delta-based Bx allows laws that are impossible for state-based Bx
  - Weak undoability
  - Weak invertibility

Content

- Background
- Our contribution: Why deltas
  - Problems of state-based Bx
  - Delta-based Bx and Laws
  - Big Picture
- Sync in the large: Lens composition and multi-ary delta lenses (mx)
- Applications in DB, SE, PL
- Applications in ML: looking forward

Asymmetric and Symmetric Bx

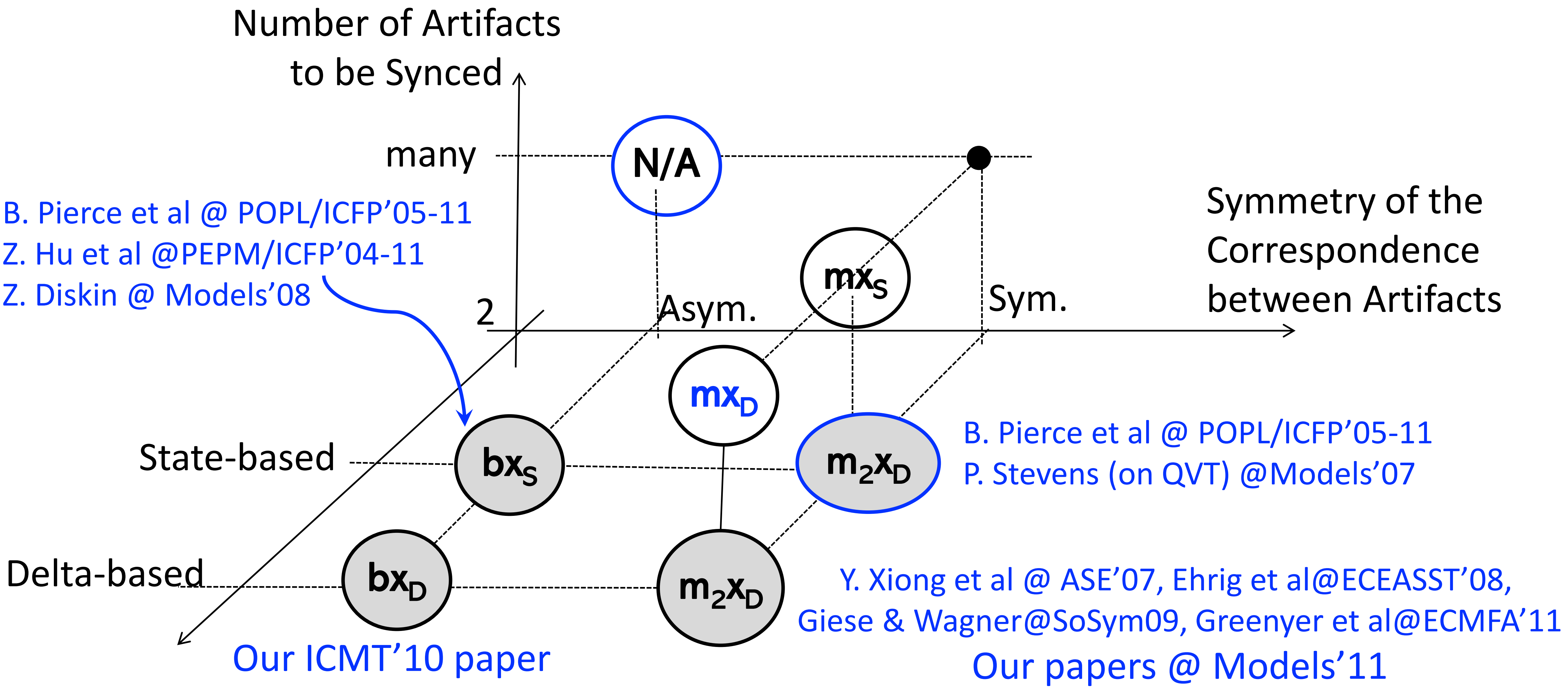
Symmetric Bx
Consistency is defined by a binary relation R



Asymmetric Bx
Consistency is defined by a function get

Symmetric bx	Asymmetric bx
R	$\left(s, get(s)\right) \in S \times V$
fPpg	$get: S \rightarrow V$
bPpg	$put: S \times V \rightarrow S$
Correctness	$PutGet: get\left(put(s, v)\right) = v$
Hippocraticness	$GetPut: put\left(s, get(s)\right) = s$

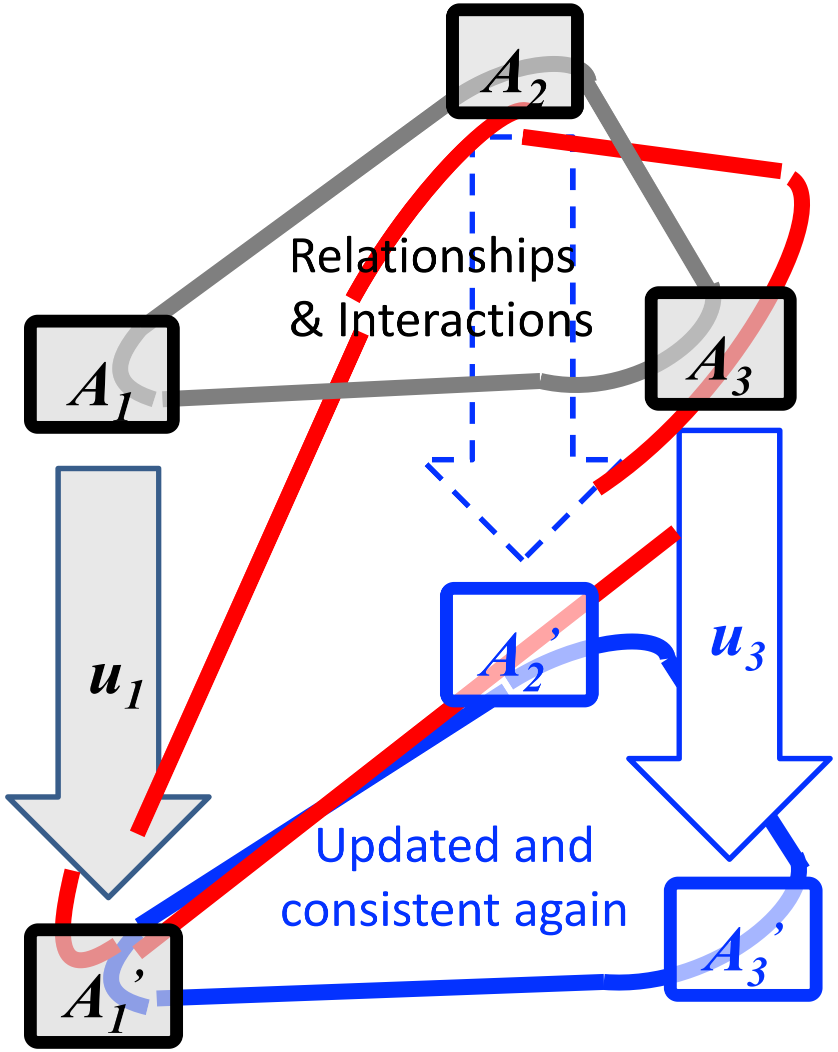
History and Classification



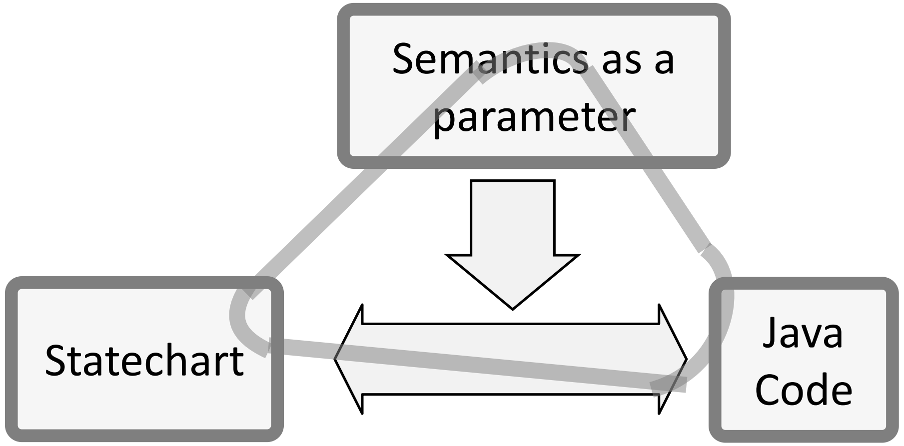
Content

- Background
- Our contribution: Why deltas
- Sync in the large: Multi-ary delta lenses (mx) and lens composition
  – Why Mx (with amendment)
  – Mx-bx lens composition: Building sync systems from components
  – Concurrent updates
- Applications in DB, SE, PL
- Applications in ML and AI: looking forward

What is Mx and why we need it

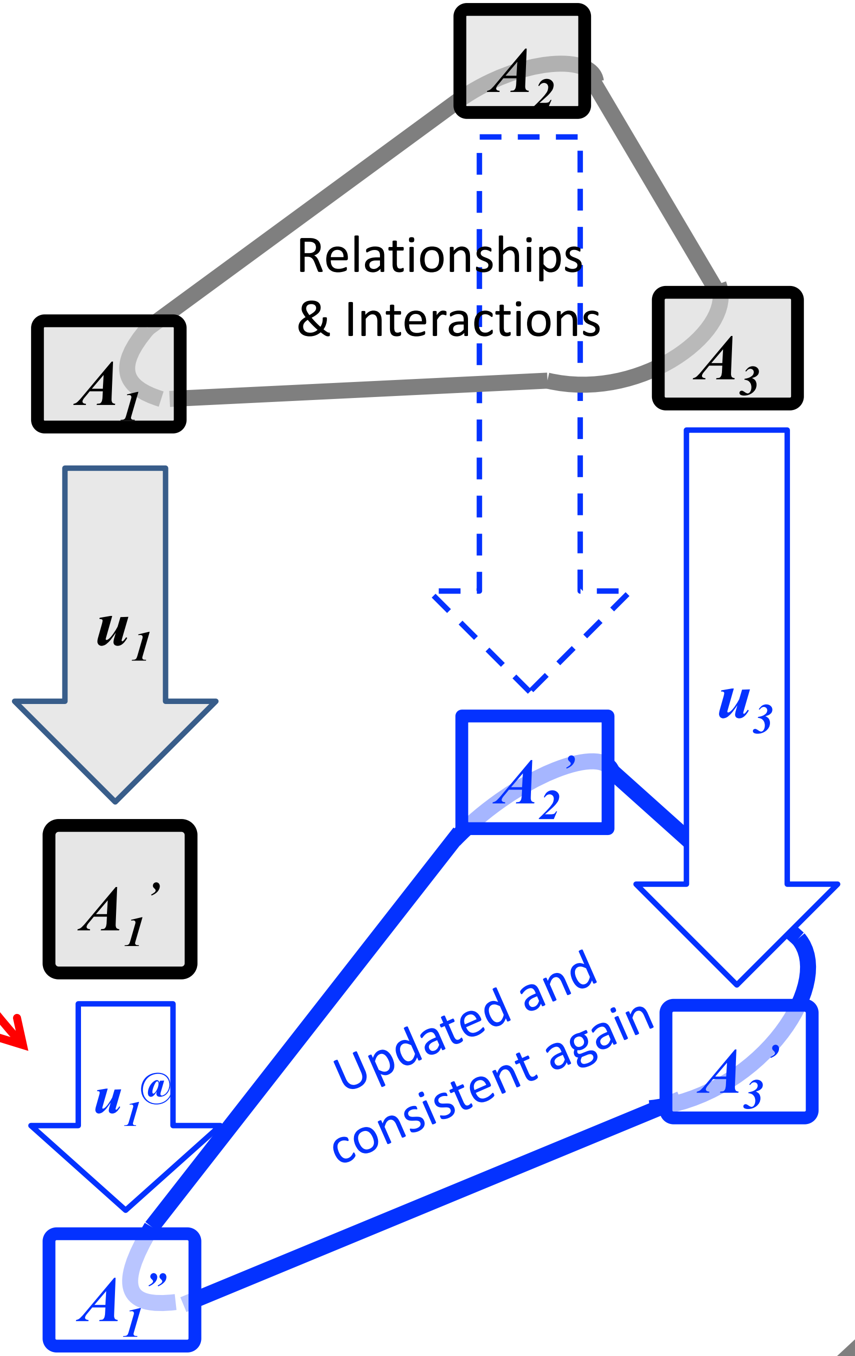


- Multiview modelling in DB and SE
  - One UML model = M view models ….
- Safety assurance: a set /megamodel **D** of design models, a set **S** of safety models, and a set of **N** of normative docs are all inter-related and to be kept in sync (megamodel sync).
- Some (many?) binary sync case are actually m-ary

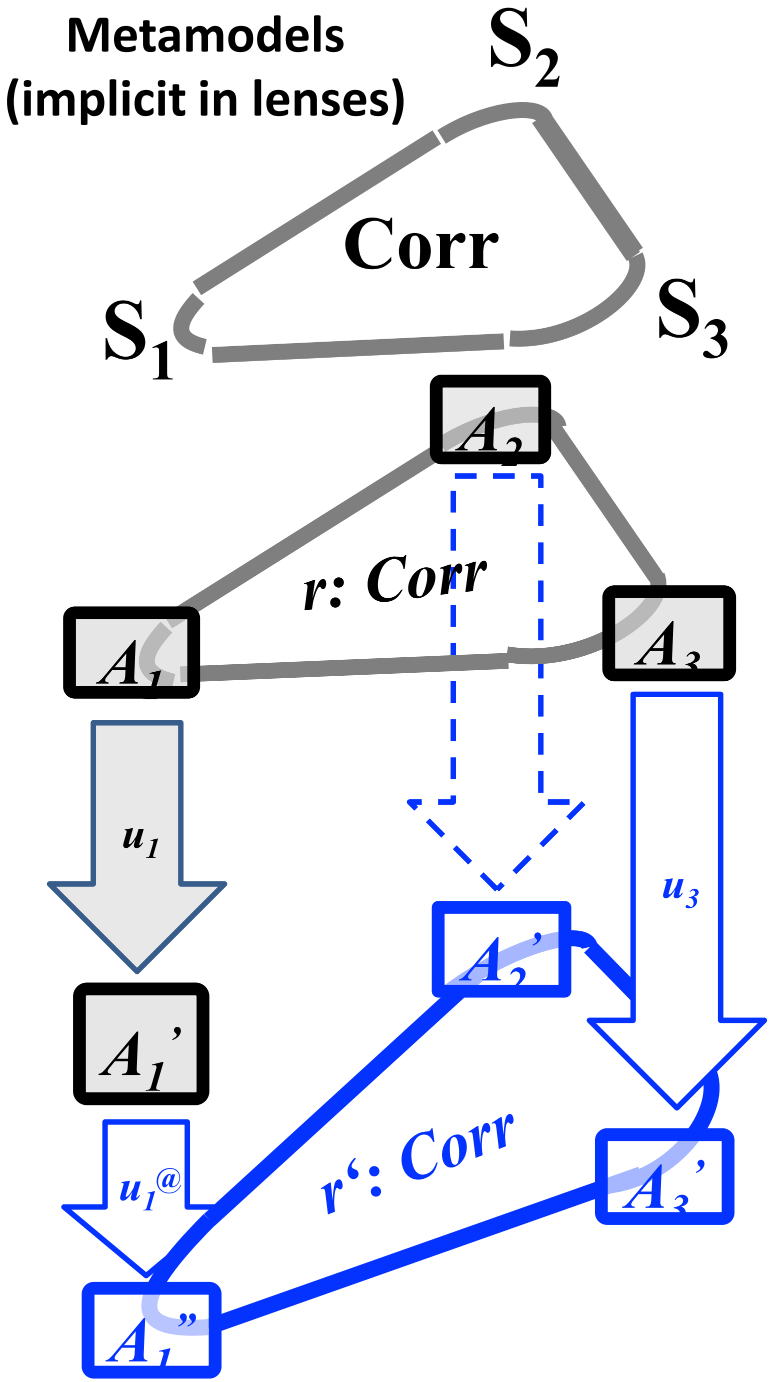


Change propagation with amendment

- Change propagation to restore consistency may involve ***amendment*** of the initial update
  - due, mainly, to **different granularity of updates** in different model spaces
  - $\boldsymbol{u_1^{@}}$ is to be consistent with $\boldsymbol{u_1}$: no undoing



An mx lens with amendment comprises the following data:

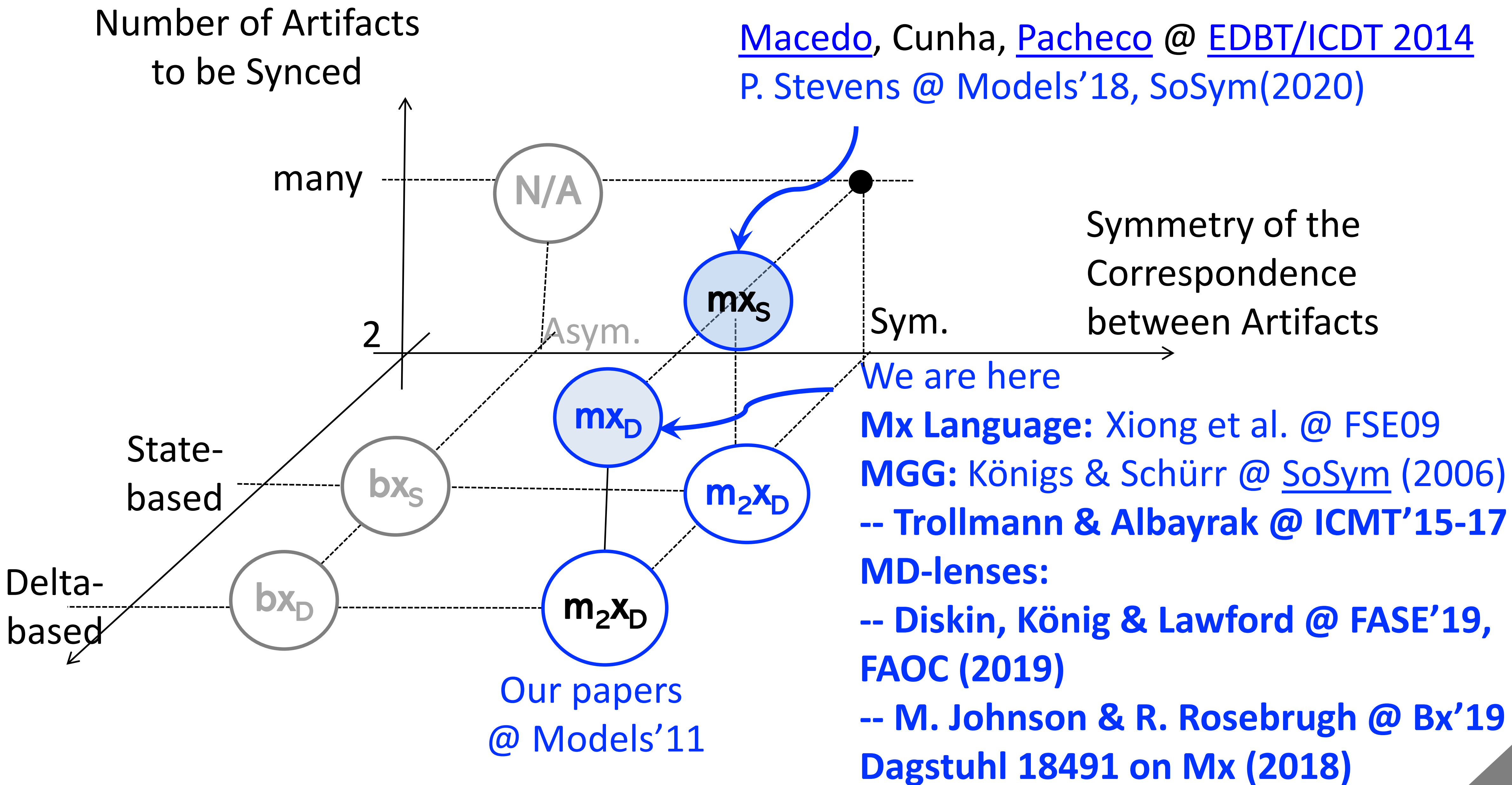


- N spaces of models and updates (determined by metamodels, which are implicit))
- Corrs between spaces
- Propagation operations, e.g.,
  - ppg_{12}: Corr x U_1 --> U_2,
  - ppg_{13}: Corr x U_1 --> U_3,
  - ppg_{11}: Corr x U_1 --> U_1 (amend.)
  - ppg_{1*} : Corr x U_1 --> Corr
  - (with a lot of incidence conditions)

	source		
target	11	21	31
	12	22	32
	13	23	33

- A lens is called ***well-behaved (wb)*** if it satisfies a set of equational laws generalizing those for the binary case

From state- to delta-based mx with mx_D lenses

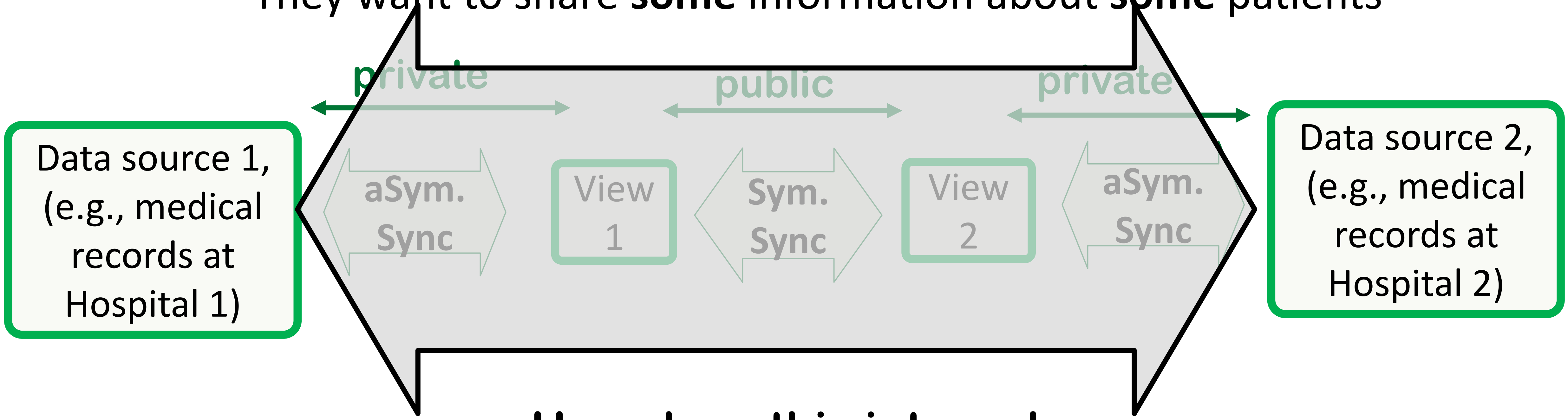


Content

- Background
- Our contribution: Why deltas
- Sync in the large: Multi-ary delta lenses (mx) and lens composition
  - Why Mx (with amendment)
  - Lens composition: Building sync systems from components
  - Concurrent updates
- Applications in DB, SE, PL
- Applications in ML and AI: looking forward

Why we need lens composition: Example

They want to share **some** information about **some** patients



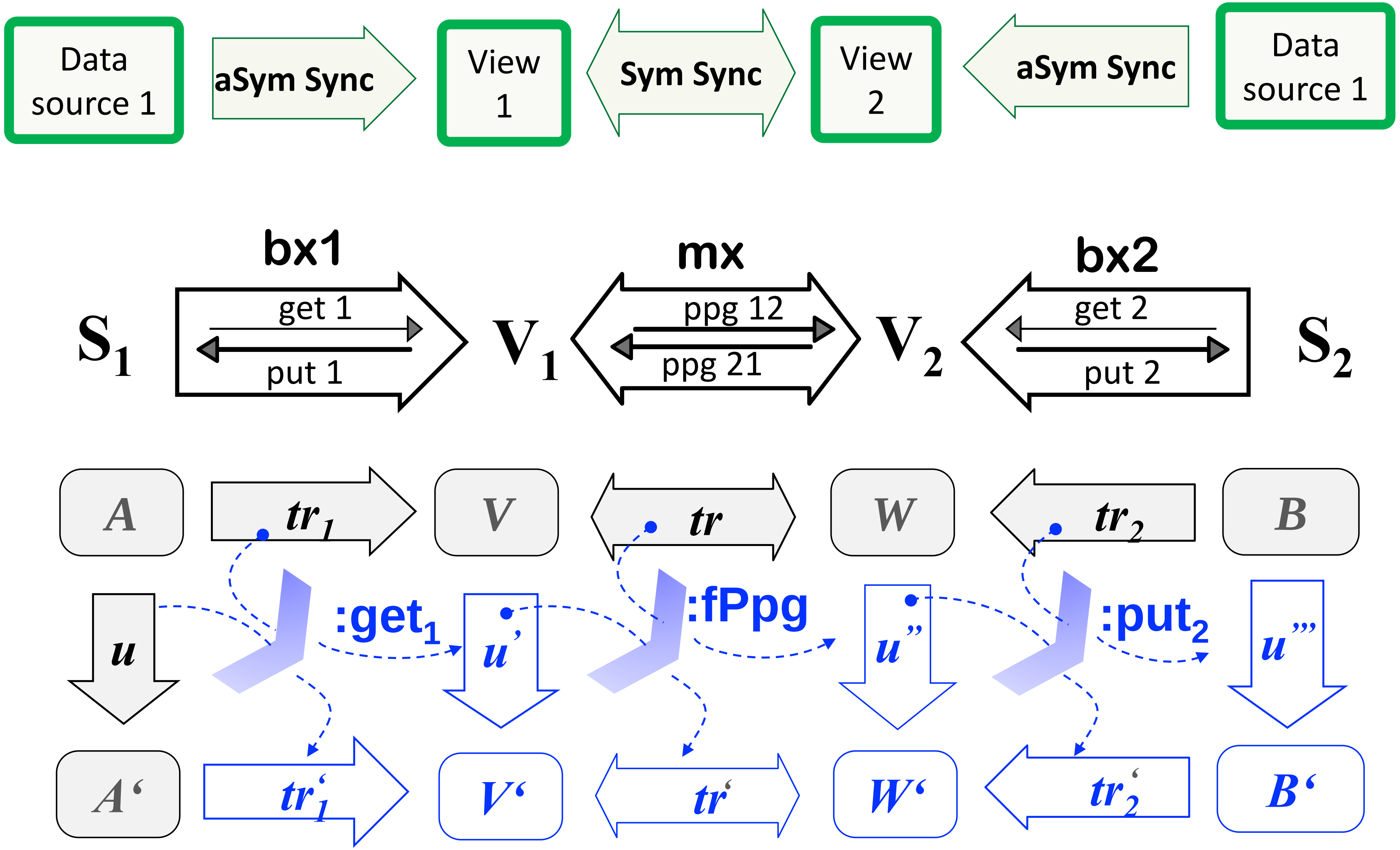
How does this integral sync behave ?

Why we need lens composition: Example

World

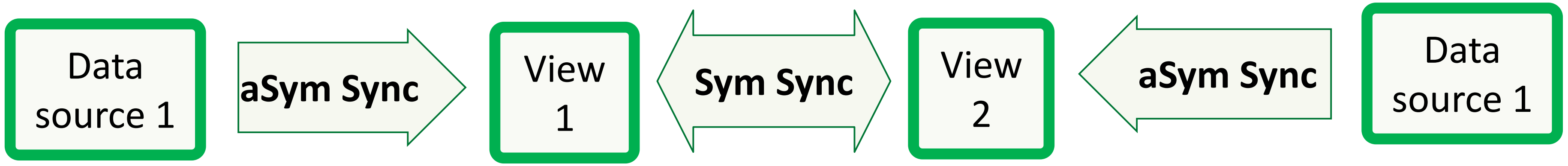
Formal model:
Model spaces
(nodes) ***and***
lenses **(arrows)**

Inside lenses:
How operations
over model
states & updates
are composed

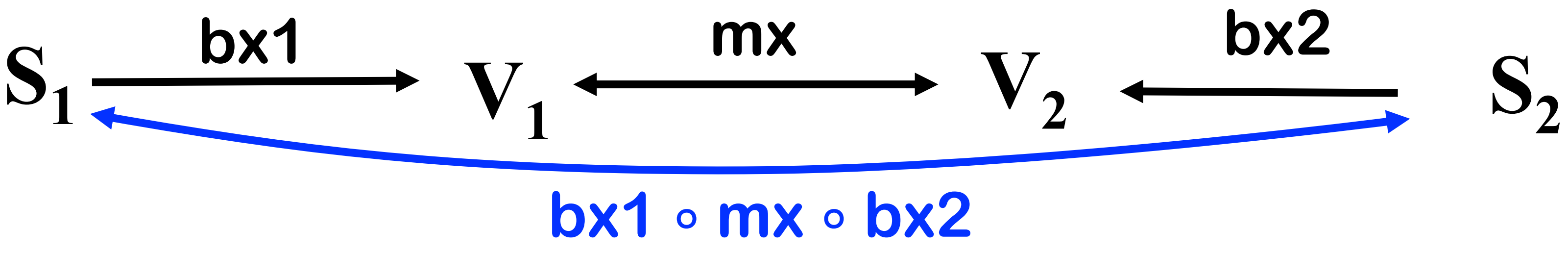


Why we need lens composition: Example

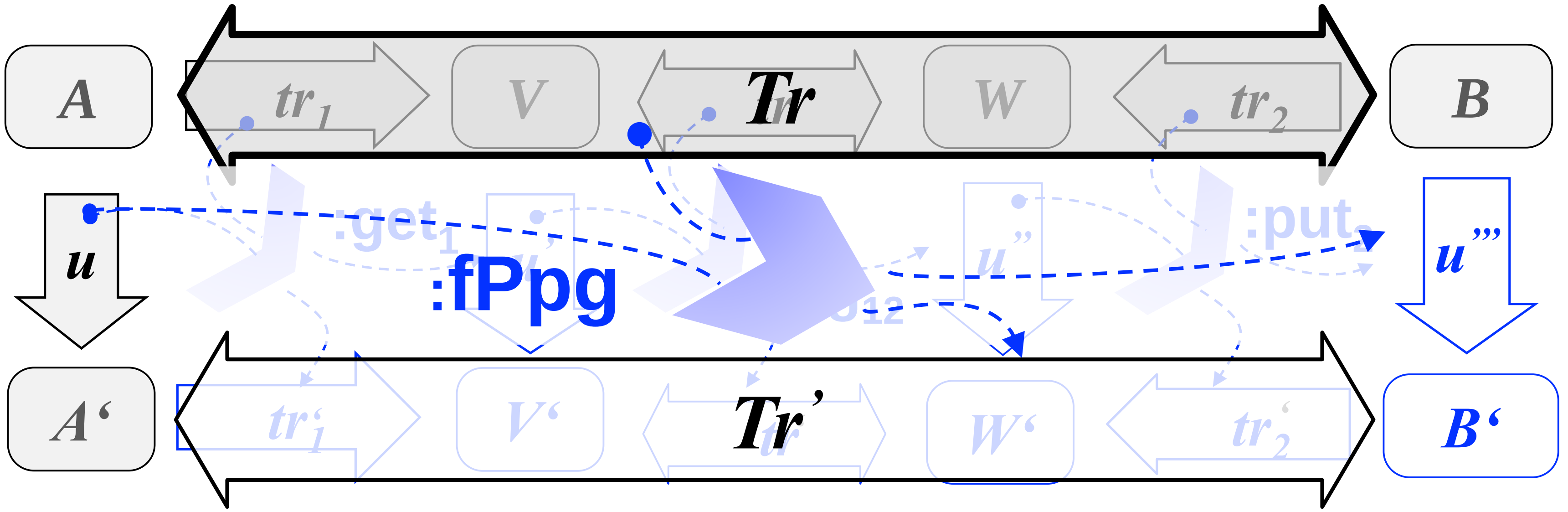
World



Formal model: Model spaces (nodes) and lenses (arrows)

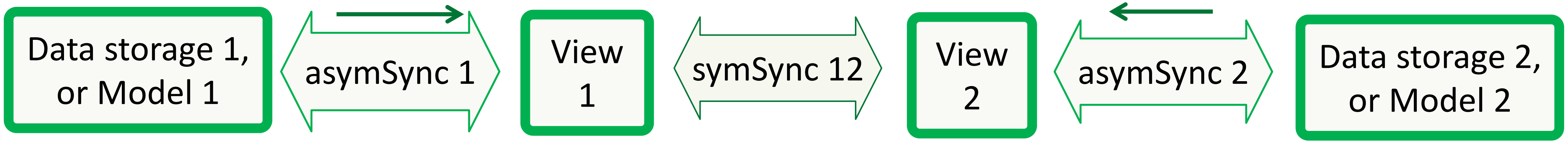


Inside lenses: Hiding details

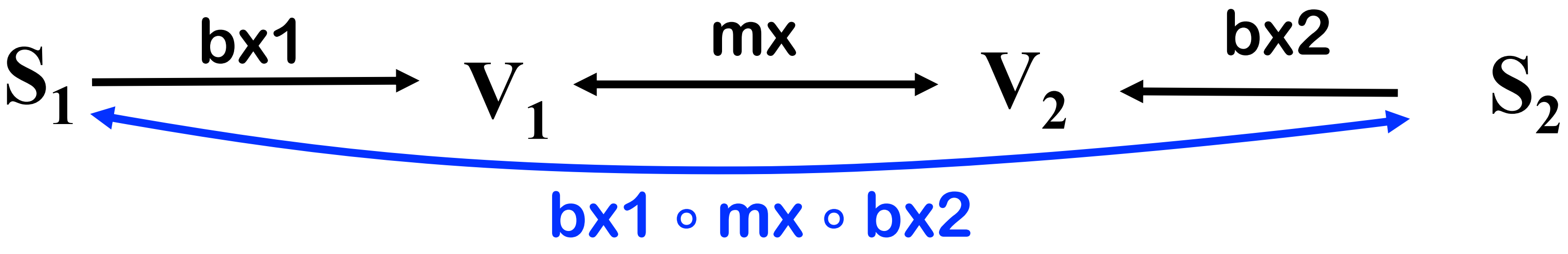


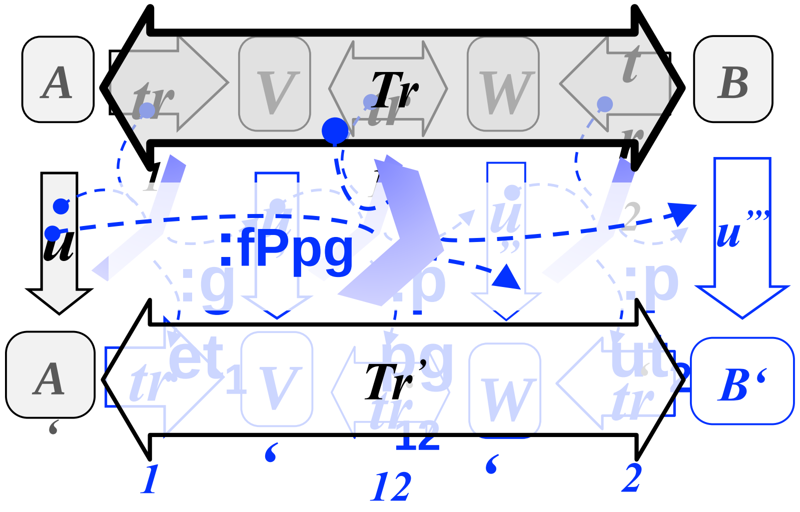
Why we need lens composition: Example

World



Formal model: Model spaces (nodes) and lenses (arrows)

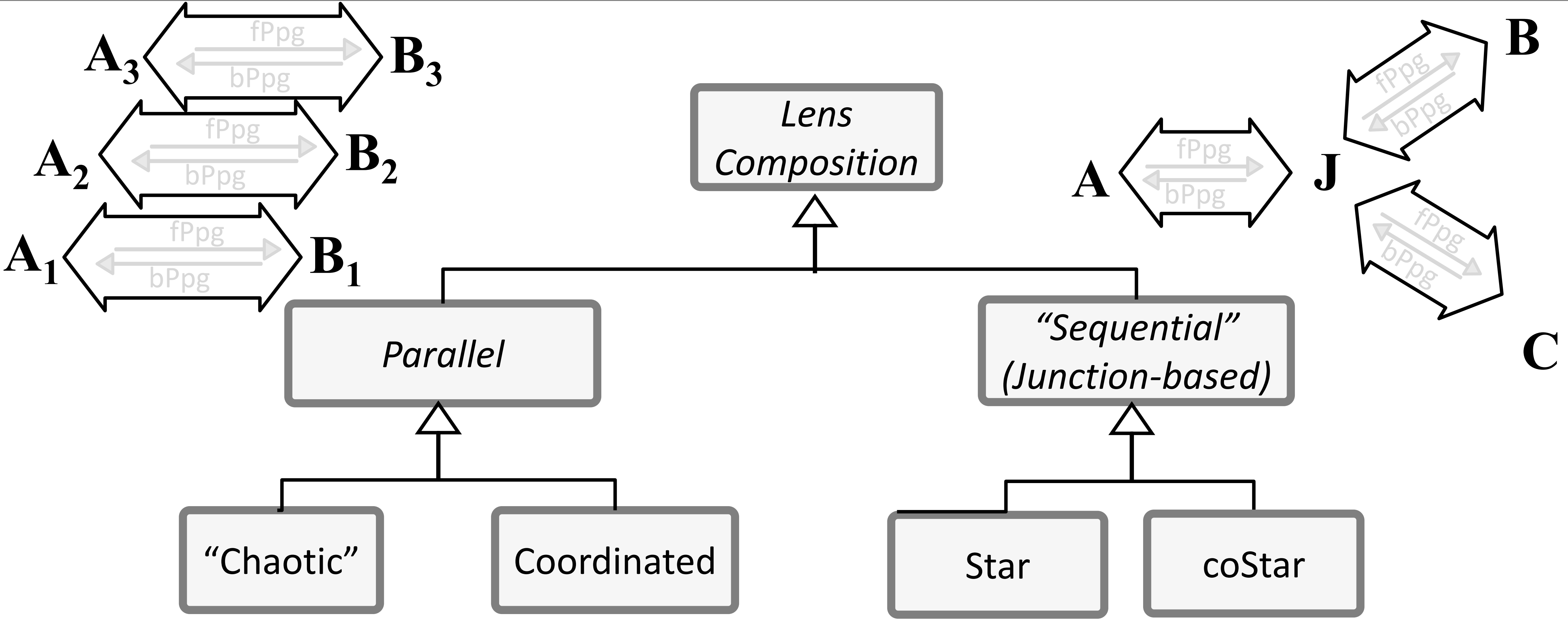




Theorem (composing binary mx).

- Composition MX = bx1 ∘ mx ∘ bx 2 is an mx-lens.
- If some (junction) conditions are satisfied, MX is **well-behaved** as soon as lenses bx1, mx12, and bx2 are such

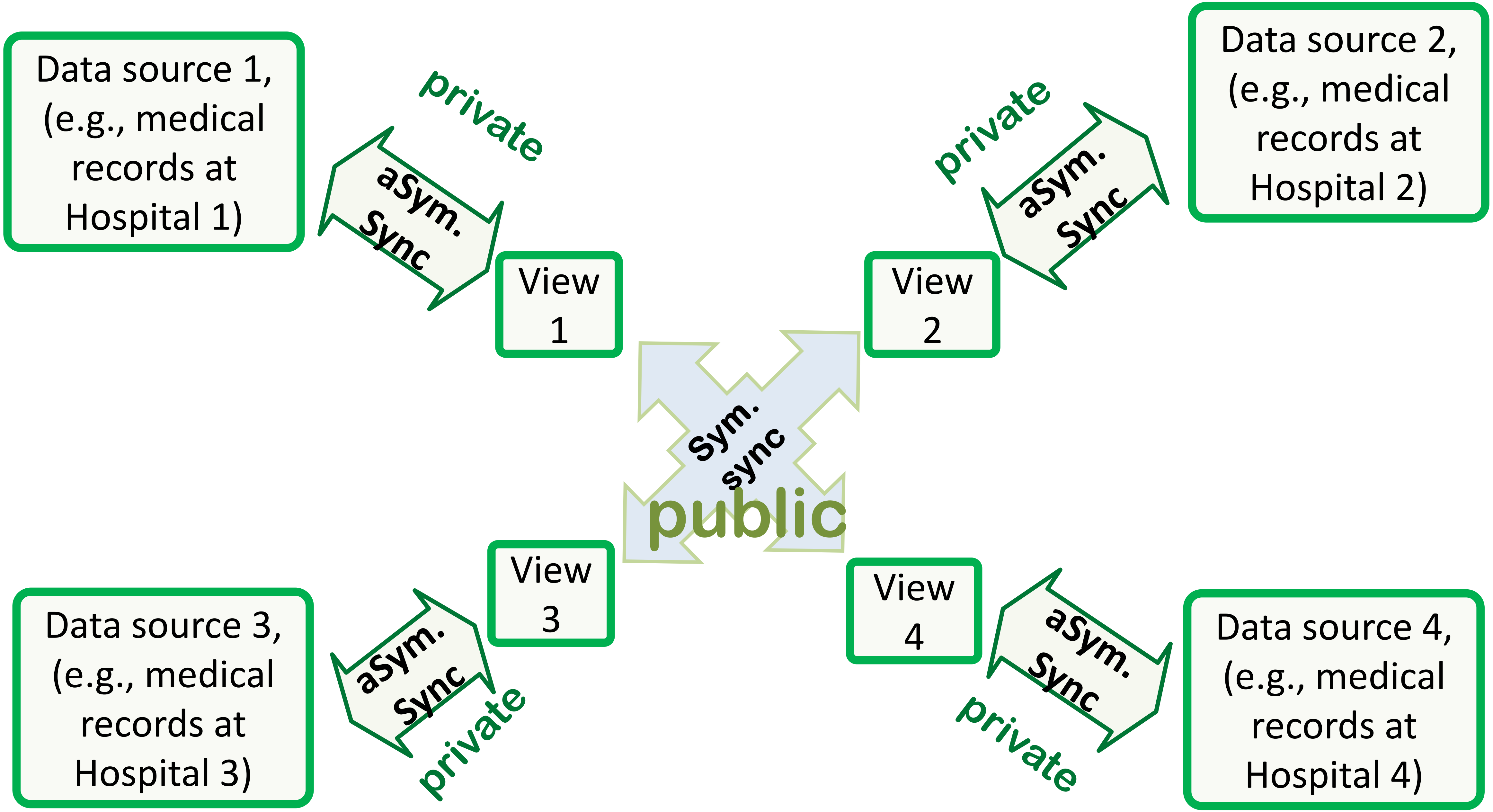
Lens composition types



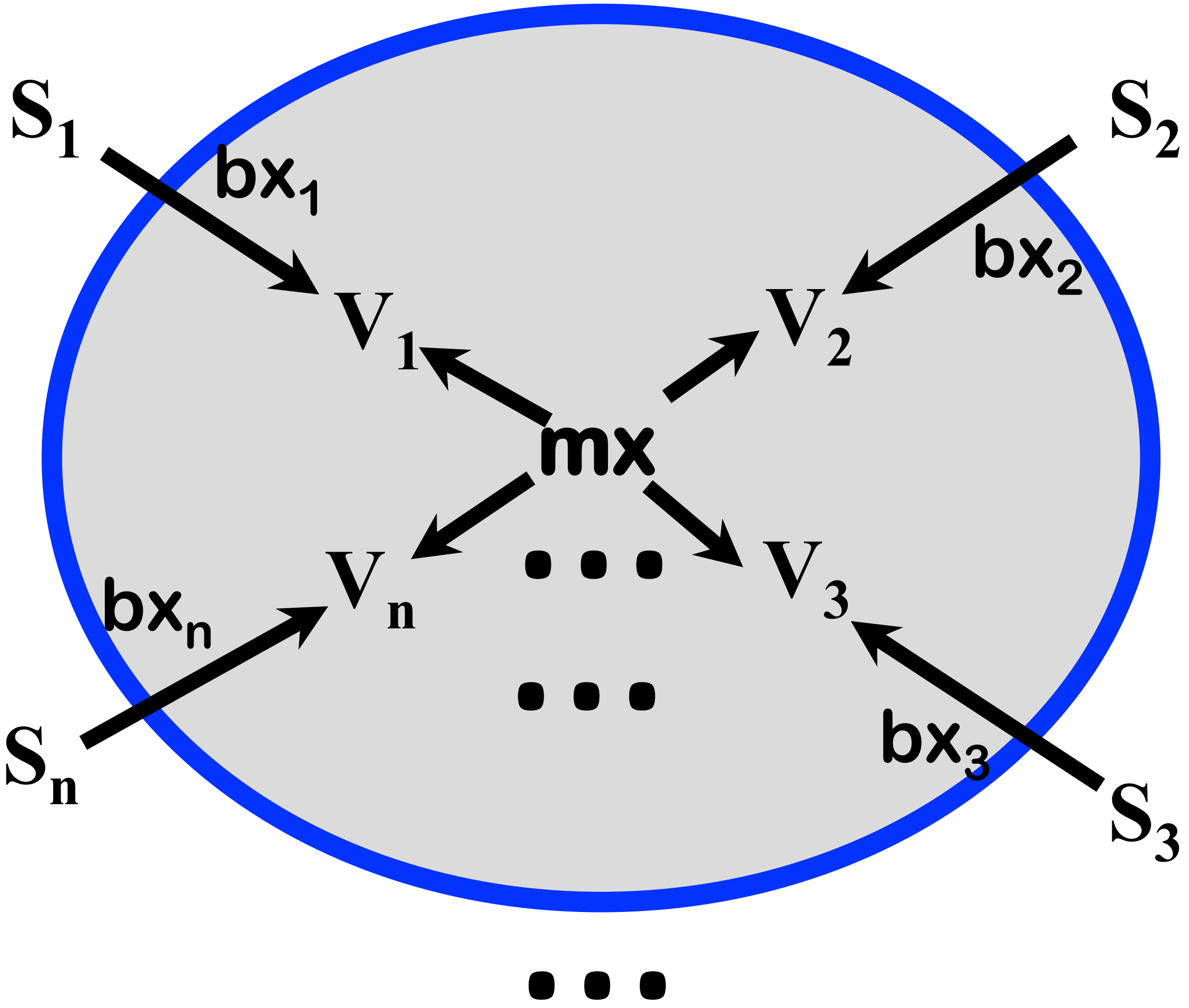
See: Diskin, König, Lawford: Multiple Model Synchronization with Multiary Delta Lenses with Amendment and K-Putput @ FAOC(2019)

This talk

Complex sync of multiple systems



(co)Star composition of lenses



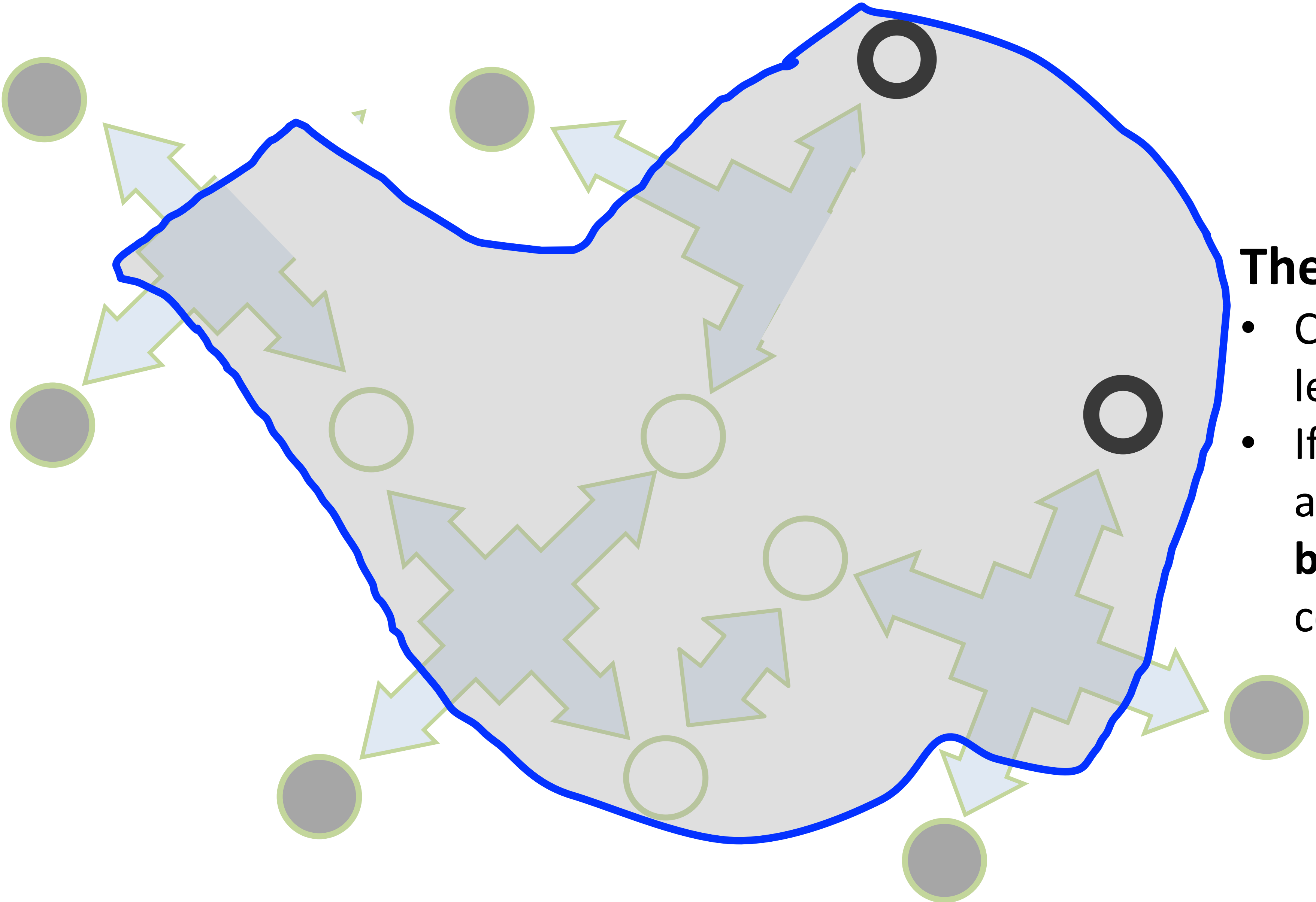
Theorem (coStar)

- Composition **MX** = **mx***(**bx**$_1$+… +**bx**$_n$) is an mx-lens.
- If some (junction) conditions are satisfied, **MX** is **well-behaved** as soon as lenses **mx** and **bx**$_i$ are such

Theorem (Star—reverse all bx)

- Composition **MX** is an mx-lens. But well-behavedness is only weakly preserved (see the paper for details).

Complex sync of multiple systems



6-ary MX built from five mx

Theorem.

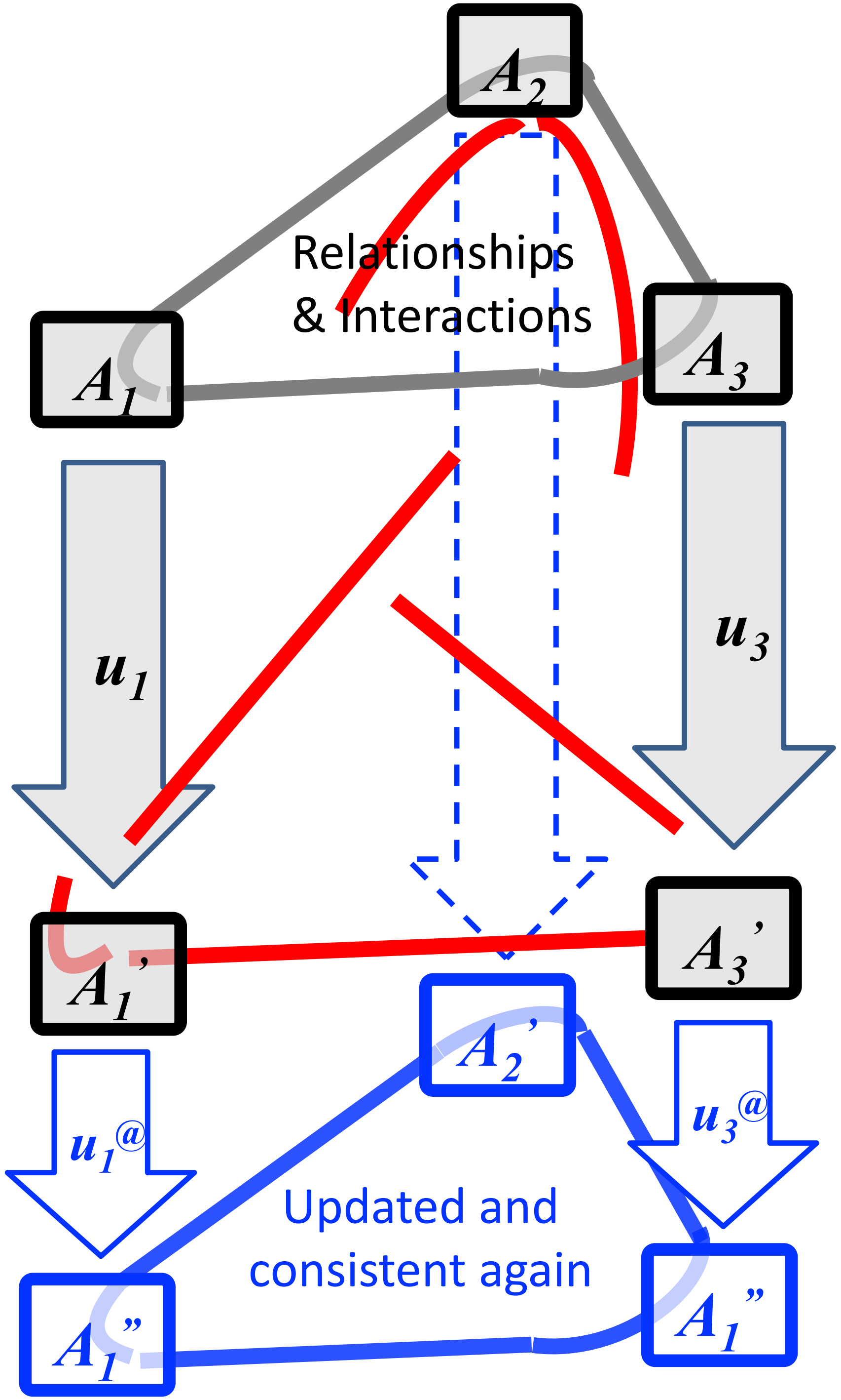
- Composed **MX** is a legal mx-lens.
- If some (junction) conditions are satisfied, **MX** is **well-behaved** as soon as component lenses are such

Category theory of lens composition

- **Bx Workshops series Bx'12-21**
  - Michael Johnson, Robert D. Rosebrugh:
    - **Multicategories of Multiary Lenses.** Bx@PLW 2019: 30-44 [2]
    - **Cospans and symmetric lenses.** Programming 2018;
    - **Spans of Delta Lenses.** Bx@STAF 2015: 1-15
- **Applied Category Theory conference series, ACT'19-21**
  - *Bryce Clark:* **A diagrammatic approach to symmetric lenses** @ ACT'20 [1]
  - Emma Chollet, Bryce Clarke, Michael Johnson, Maurine Songa, *Vincent Wang*, *Gioele Zardini:* **Limits and colimits in a category of lenses**, ACT'21
  - *Sophie Libkind*, Andrew Baas, Evan Patterson, James Fairbanks, **Operadic Modeling of Dynamical Systems: Mathematics and Computation** @ ACT'21 (state-based lenses are employed)
- The result on lens composition presented above are from [1], [2], and **Multiple Model Synchronization ….** @ FAOC(2019) by Zinovy Diskin, Harald König, Mark Lawford

Content

- Background
- Our contribution: Why deltas
- Sync in the large: Multi-ary delta lenses (mx) and lens composition
  - Why Mx (with amendment)
  - Lens composition: Building sync systems from components
  - Concurrent updates
- Applications in DB, SE, PL
- Applications in ML and AI: looking forward



Concurrent updates

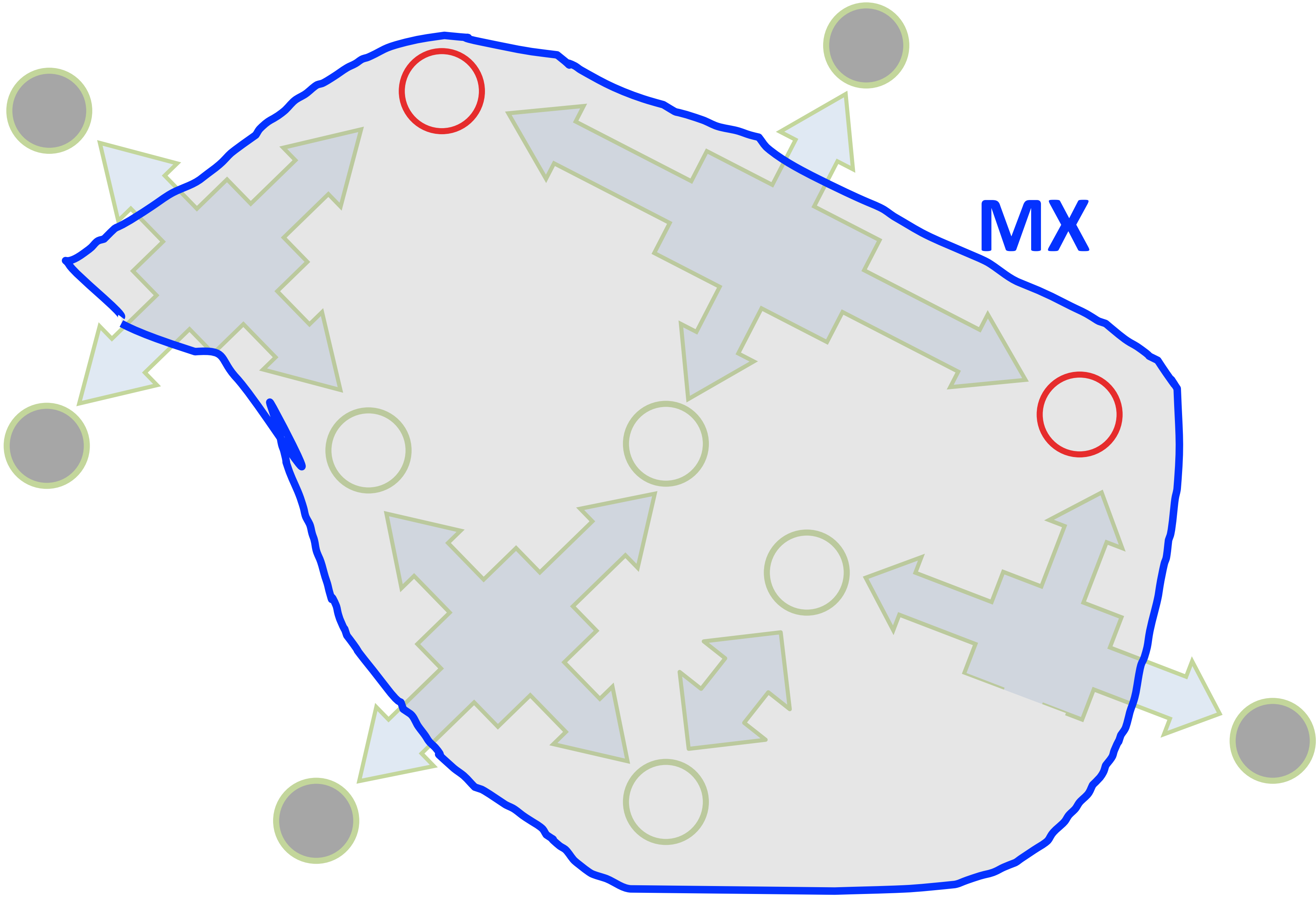
	source		
target	ppg_{11}	ppg_{21}	ppg_{31}
	ppg_{12}	ppg_{22}	ppg_{32}
	ppg_{13}	ppg_{23}	ppg_{33}

- Possible conflicts between:
  - ppg_{13}(u1) and u3
  - u1 and ppg_{31}(u3)
  - ppg_{12}(u1) and ppg_{32}(u3)
- Sync needs:
  - Conflict resolution
  - Update merging

Approached in:

- Hermann, Ehrig, Ermel, Orejas @ FASE'12
- Xiong, Song, Hu, Takeichi @ SoSym (2013)
- Trollmann & Albayrak (MGG) @ ICMT'15-17
- Orejas, Pino, Navarro @ FASE'20
- Fritsche, Kosiol, Möller, Schürr, Taentzer @SLE'20
- **Future work:** employ mx lenses with amendment

Concurrency again: Lens composition with cycling

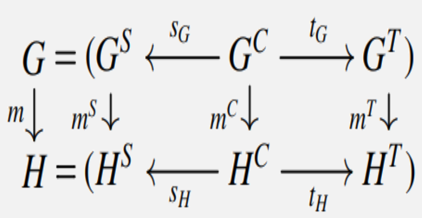


Building MX needs conflict resolution and update merging

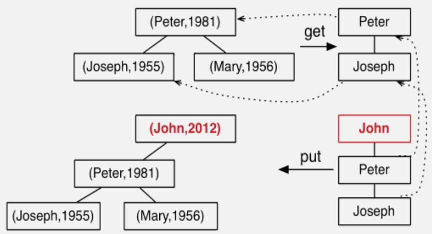
Content

- Background
- Our contribution: Why deltas
- Sync in the large: Multi-ary delta lenses (mx) and lens composition
- Applications in DB, SE, PL
- Applications in ML and AI: looking forward

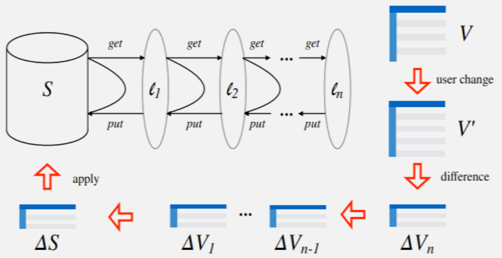
Delta Lenses & Deltas have been widely used



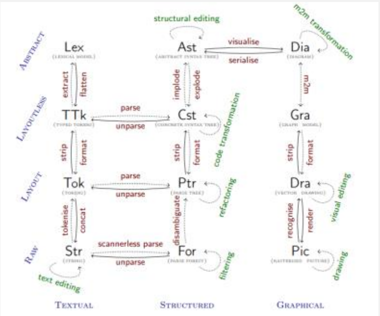
TGG-based Model Synch [MODELS11, FASE12, ICGT12, ICMT14, JOT21, …]



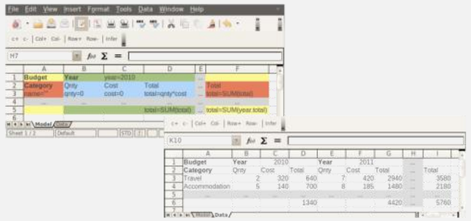
Prog. Lang. for Inductive Data Struct. [BX12]



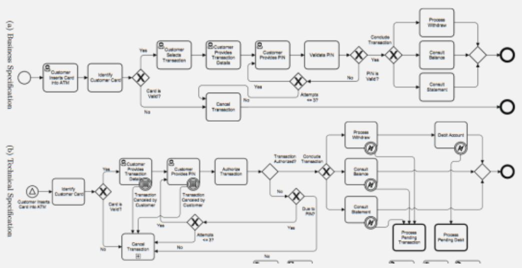
Database View Update [ICFP18]



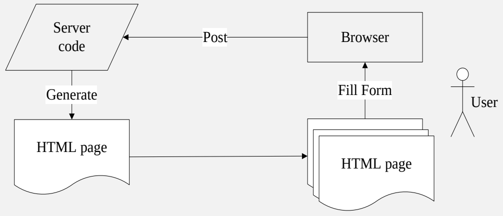
Incremental Parsing and Tokenizing [MODELS14]



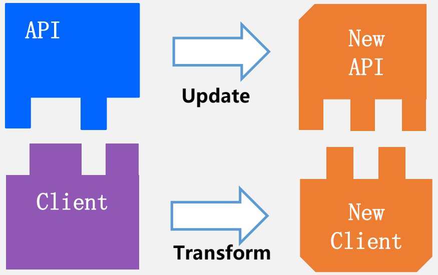
Spreadsheet Synch [ICMT12]



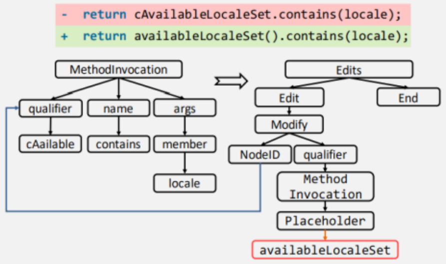
Business Process Model Synch [Sosym13]



PHP Page Change [FSE12]



API Updating [ECOOP16]

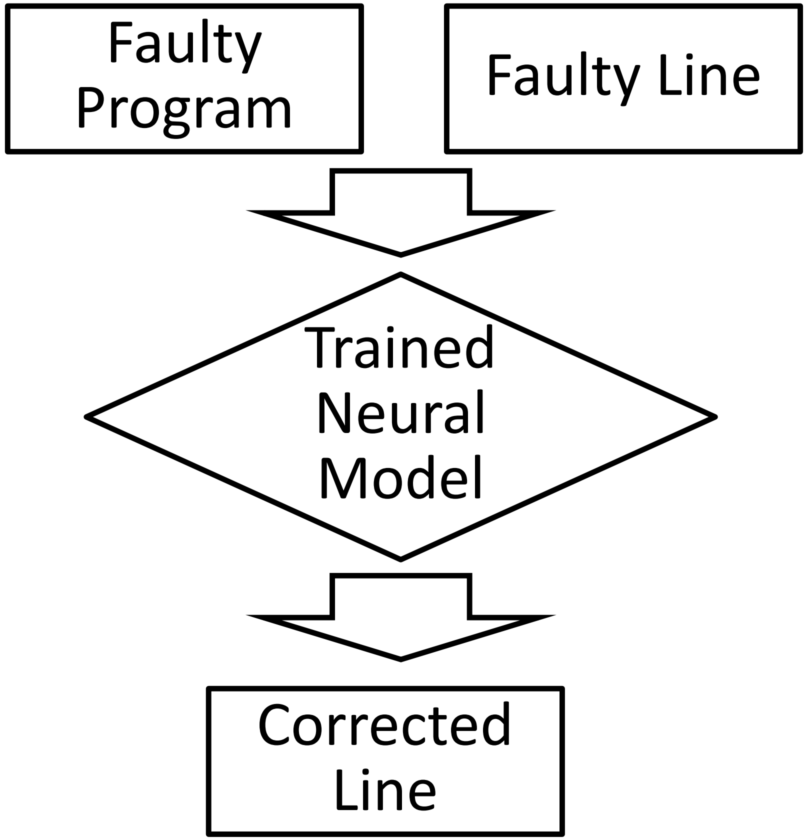


Program Repair [ESECFSE21]

[1] Diskin, Gholizadeh, Wider, Czarnecki: A three-dimensional taxonomy for bidirectional model synchronization. J. Syst. Softw. 111: 298-322 (2016)

Application Example: Program Repair

- Traditional neural program repair approaches use state-based representation of changes



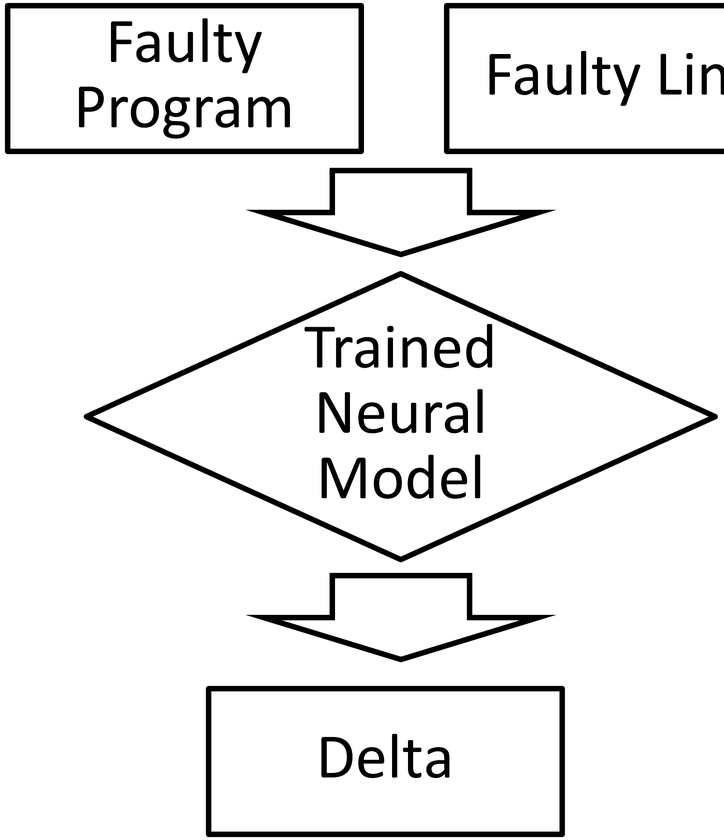

```
cfa.createEdge(fromNode, Branch.UNCOND, finallyNode);
```

1. Need to learn delta discovery during training
2. Need to generate 13 tokens correctly

```
cfa.createEdge(fromNode, Branch.ON_EX, finallyNode);
```

Application Example: Program Repair

- Delta-based change representation is simpler and more concise




```
cfa.createEdge(fromNode, Branch.UNCOND, finallyNode);
```

1. Delta is directly shown
2. Need to generate 3 tokens correctly

```
modify(9, ON_EX)
```

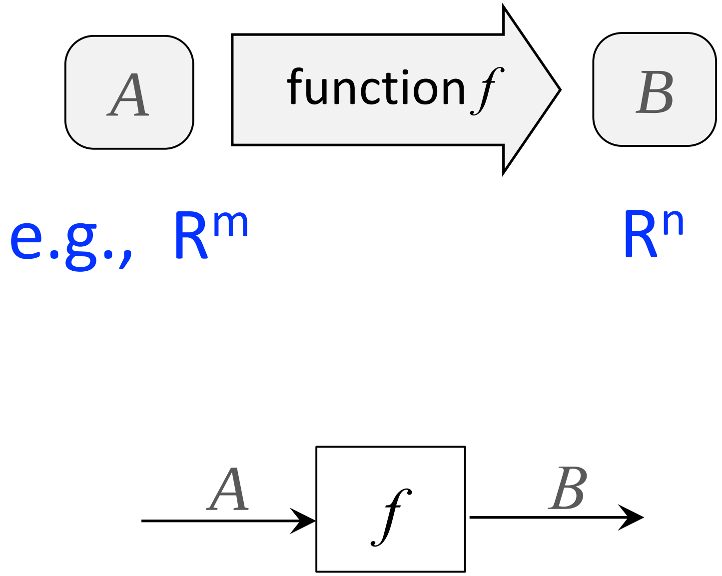
Delta-based repair approach repairs 76% more bugs than SOTA [1]

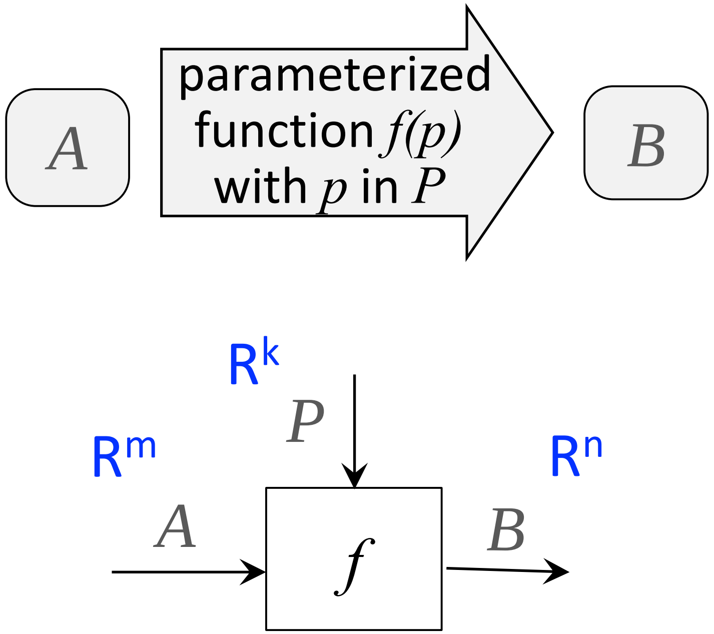
[1] Q Zhu, Z Sun, Y Xiao, W Zhang, K Yuan, Y Xiong, L Zhang. A Syntax-Guided Edit Decoder for Neural Program Repair. ESEC/FSE'21.

Content

- Background
- Our contribution: Why deltas
- Sync in the large: Multi-ary delta lenses (mx) and lens composition
- Applications in DB, SE, PL
- Applications in ML and AI: looking forward
  - Reverse gradient descent as a lens
  - Deep learning and lens composition
  - Open games, cybernetics

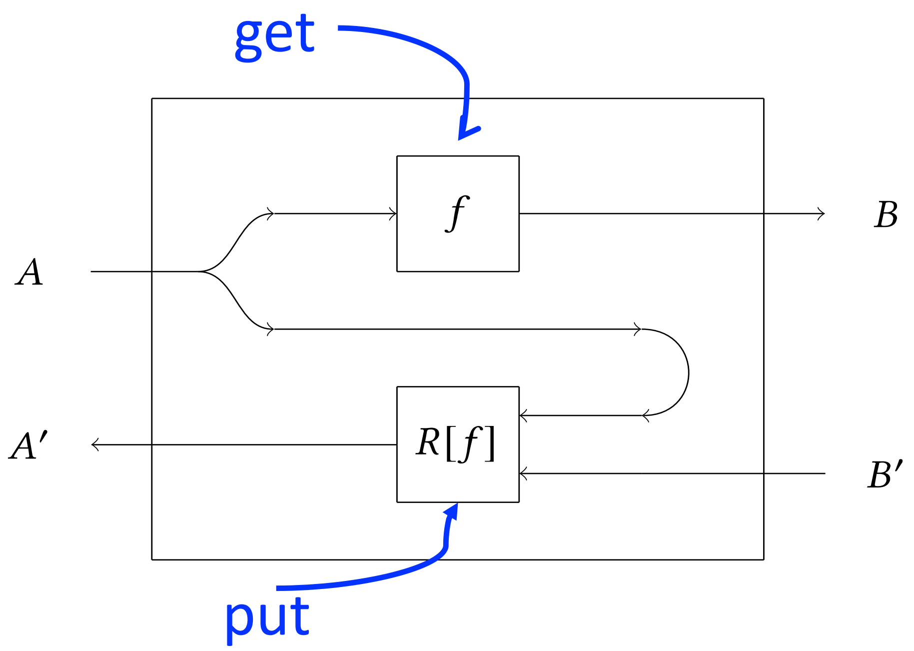
Background: String/block diagram notation for functions



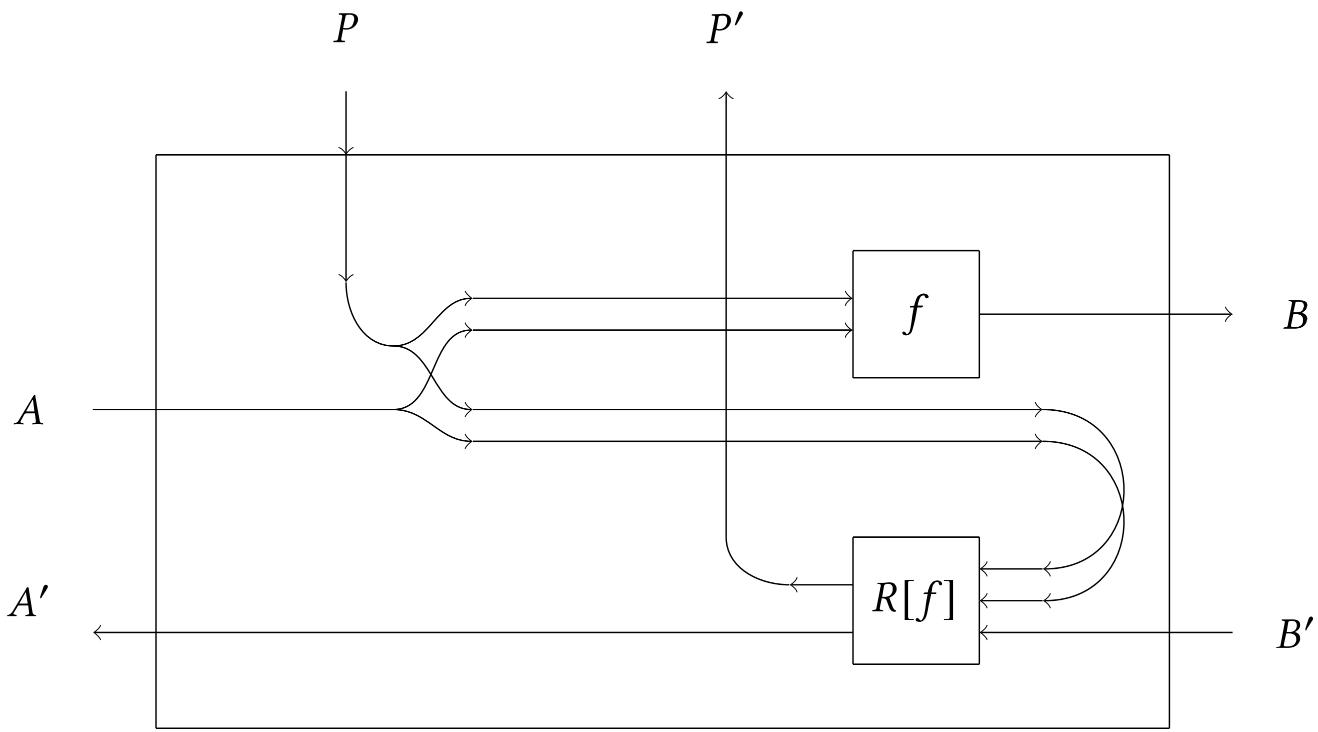


Reverse gradient descent as a lens

(from Cruttwell, Gavranović, Ghani, Wilson, Zanasi:
Categorical Foundations of Gradient-Based Learning @ ACT'21)



Asymmetric lens
(reverse gradient descent as put)



Parametric asymmetric lens
(reverse gradient as (put_A, put_P))

ML via parametric lens composition

(from Cruttwell, Gavranović, Ghani, Wilson, Zanasi:

Categorical Foundations of Gradient-Based Learning @ ACT'21)

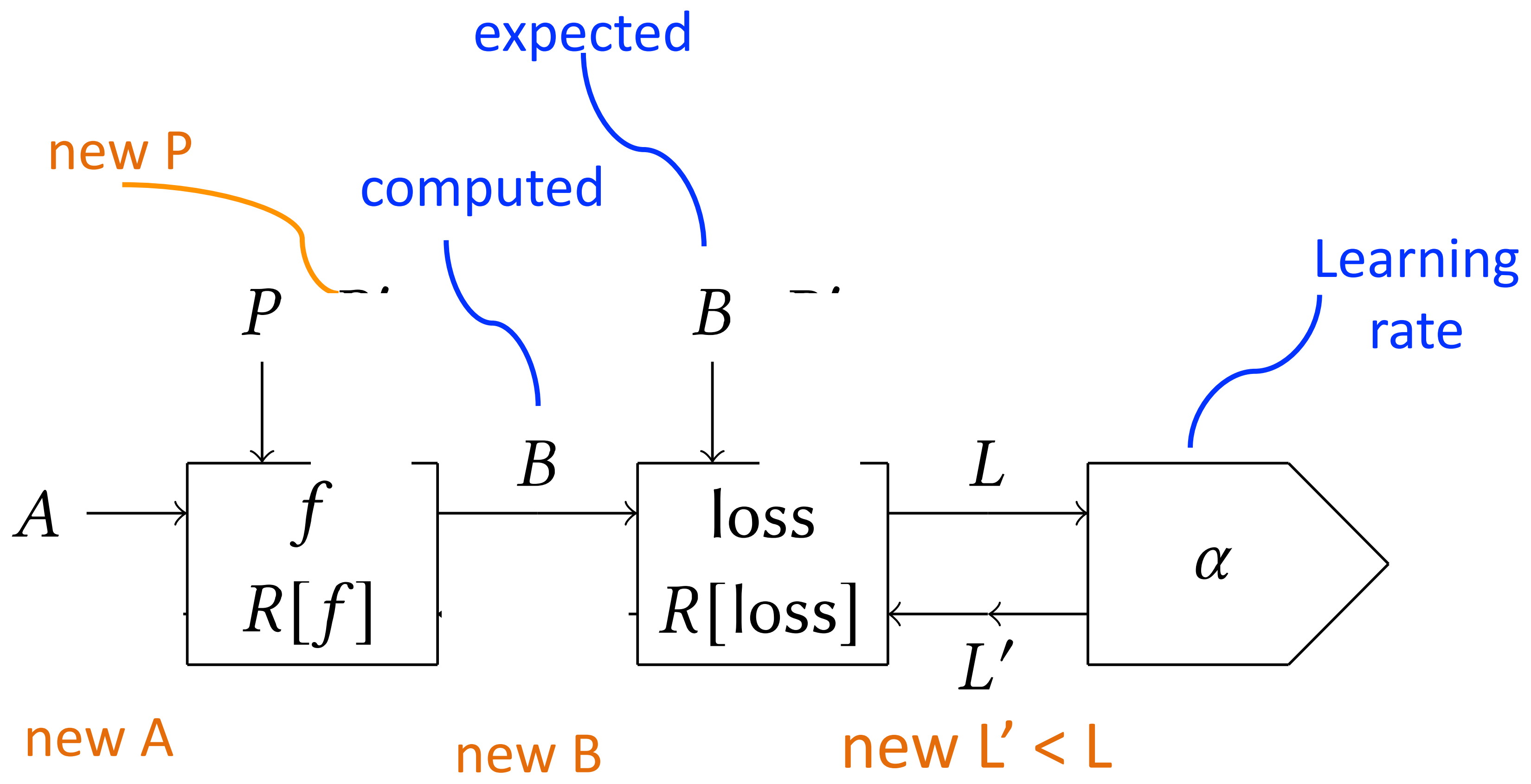


Fig. 3. Model composed with a loss function and a learning rate

Deep learning via parametric lens composition

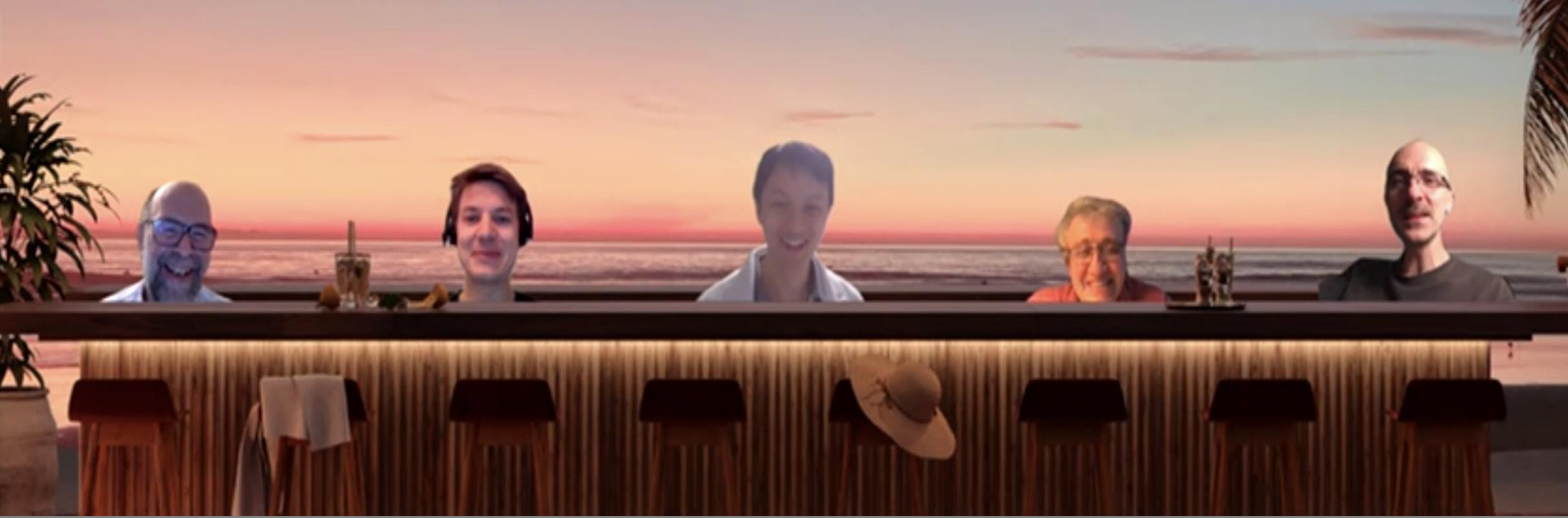
- Brendan Fong, David Spivak, Rémy Tuyéras: **Backprop as Functor: A compositional perspective on supervised learning** @LICS 2019
- Brendan Fong, Michael Johnson: **Lenses and Learners** @Bx/PLW 2019
- Zinovy Diskin: **General Supervised Learning as Change Propagation with Delta Lenses** @ FoSSaCS 2020
- David Spivak*:* **Learners' languages** @ ACT'21

Other applications of lenses in AI

- **Open games via lenses (state-based)**
  – Elena Di Lavore, Jules Hedges and Paweł Sobociński **Compositional modelling of network games** @ *Computer Science Logic 2021*
  – Neil Ghani, Jules Hedges, Viktor Winschel and Philipp Zahn **Compositional game theory** @ LICS'2018
  – Joe Bolt, Jules Hedges, Philipp Zahn: **Bayesian open games** @arxiv'19
- **Categorical cybernetics via lenses (state-based)**
  – Matteo Capucci, Bruno Gavranovic, Jules Hedges, Eigil Rischel, "Towards foundations of categorical cybernetics" @ ACT'21
- **From state- to delta-based lenses in AI (work in progress :-)**

What to take home

- **State-based sync is a composed construct**: it is a composition of delta discovery and delta propagation. To manage it better, let’s separate concerns
- **Delta lens composition** is a powerful instrument for building complex model synchronizers from components
- Delta lenses do have successful **applications in DB, SE, PL**
- Possibilities for their **applications in AI and specifically ML** is under investigation, but first results look promising



Thank you!