

Diagrammatic algebra for model management
(versioning, sync, reconciliation, merge...)
Work in Progress

Zinovy Diskin

Joint work with Michał Antkiewicz and Krzysztof Czarnecki

Generative Software Development Lab

University of Waterloo, Canada

Motivation

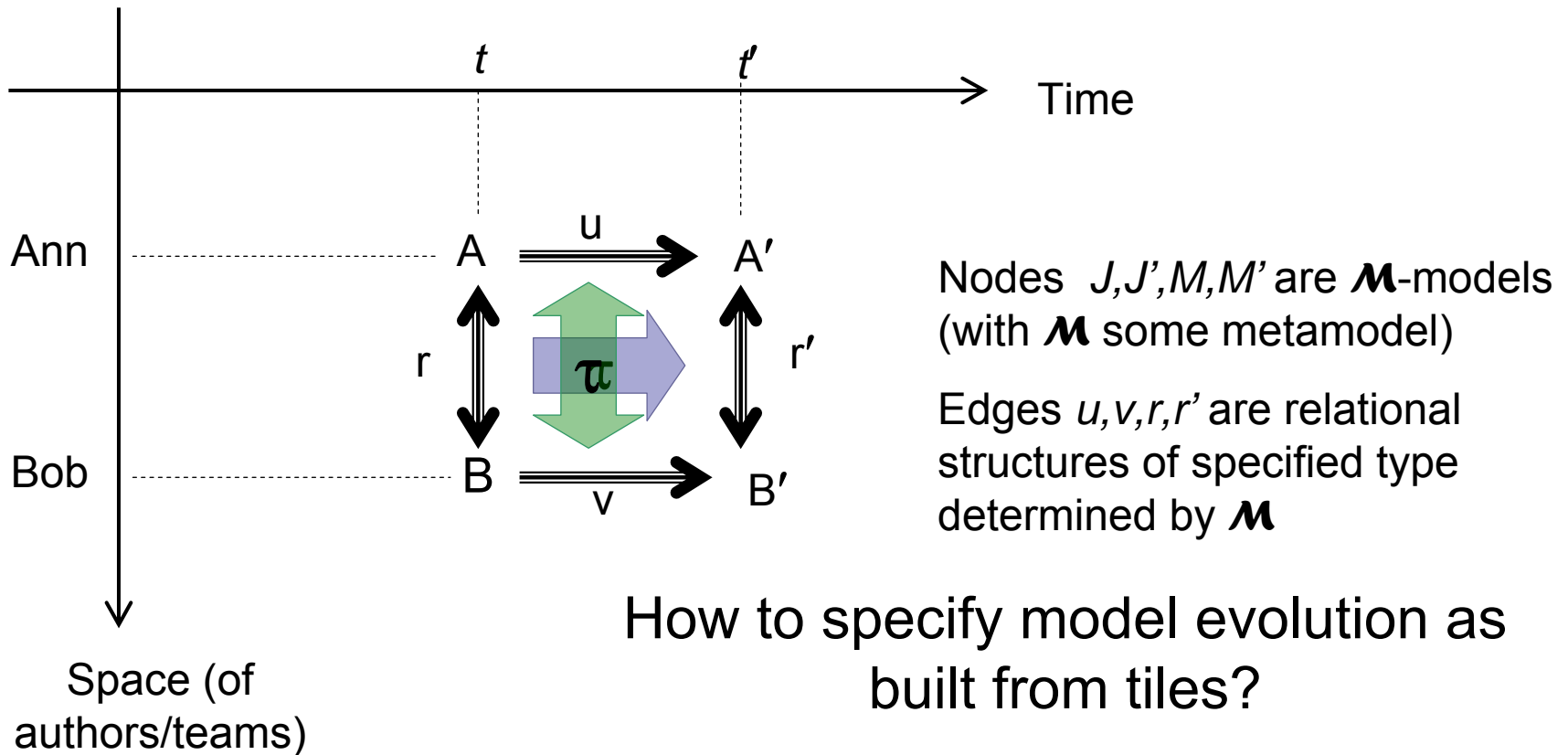
Specifying modelware is, simplistically, this:

- Picture/diagrams are good but informal (no formal semantics)
- Formulas are bad (“mind boggling” :) but precise
- The best of the two worlds: precise diagrams with formal semantics

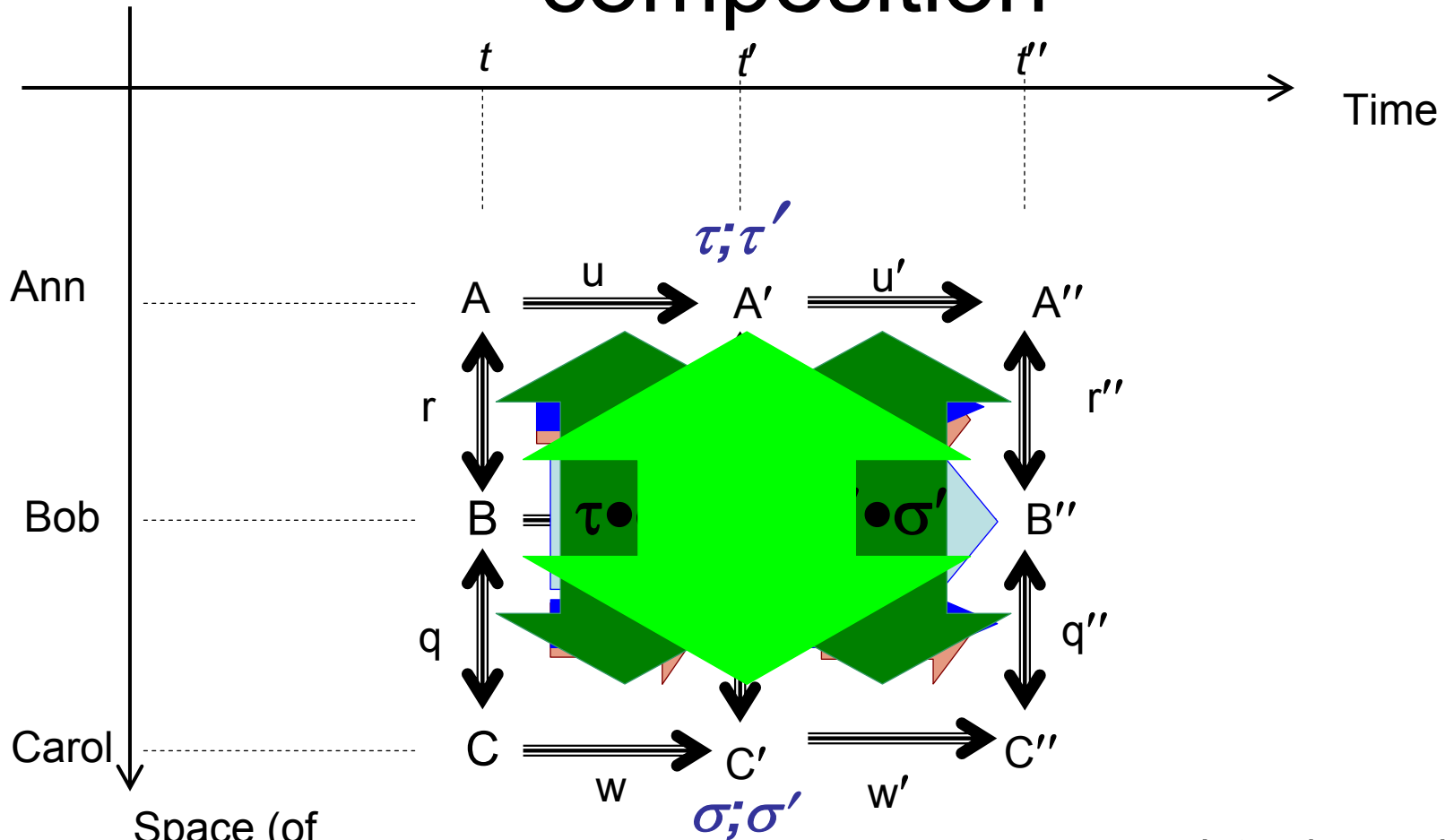
An example: Diagrammatic algebra of model versioning

Synchronization in the small -- *tiles*

Model versions vary in time and space of authors/teams.



Synchronization in the large – tile composition



$$AC'' = (\tau; \tau) \bullet (\sigma; \sigma)$$

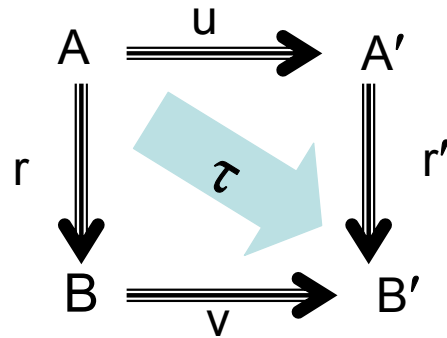
$$AC'' = (\tau \bullet \sigma); (\tau' \bullet \sigma')$$

Interchange law:

$$AC'' = AC''$$

“Right” categorical view

Double-
categories:



- Grand Slogan: Model versioning and sync is, structurally, a phenomenon of higher-dimensional category theory
- Tile diagram chasing: Towards calculus of updates and update reconciliation

Abstractness vs. engineering

- Physics
- Math
- Category theory (abstract nonsense)
- Higher-dimensional CT

Software
Engineering



Plans for the workshop

- Talking about model sync
- Semi-technical talk about syntax vs. semantics in model transforms. The Thesis:
Schema/model gives us much more than its set of instances (Tech.Report)
- Tech talk on algebraic semantics for model translation (ATEM'06)
- Whiteboard talk: Model= Graph + Constraints;
How to manage constraints in a graph-based way (ENTCS, to appear)
- Graphical algebras and queries, declaratively (Bull.Symb. Logic, 1997, DKE'03)
- Model sync in pictures, formally