Evolution of Modelling Languages

Bart Meyers
Hans Vangheluwe

McGill
School of Computer Science
Universiteit Antwerpen
Department of Mathematics and Computer Science
MSDL
Modelling, Simulation and Design Lab

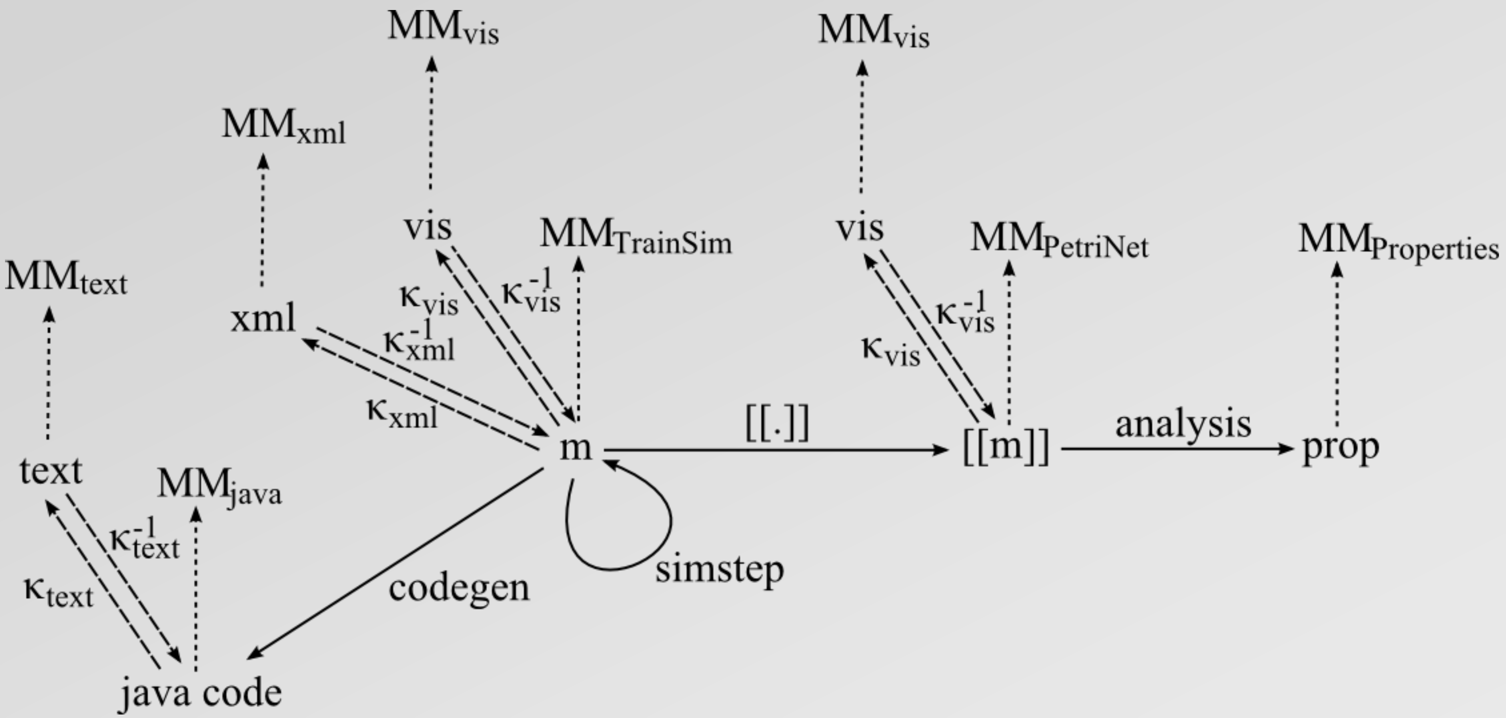
A Foundation for Modelling Language Evolution

- completeness
- automation

Challenges

- consistency
- continuity

Language evolution in an MDE system



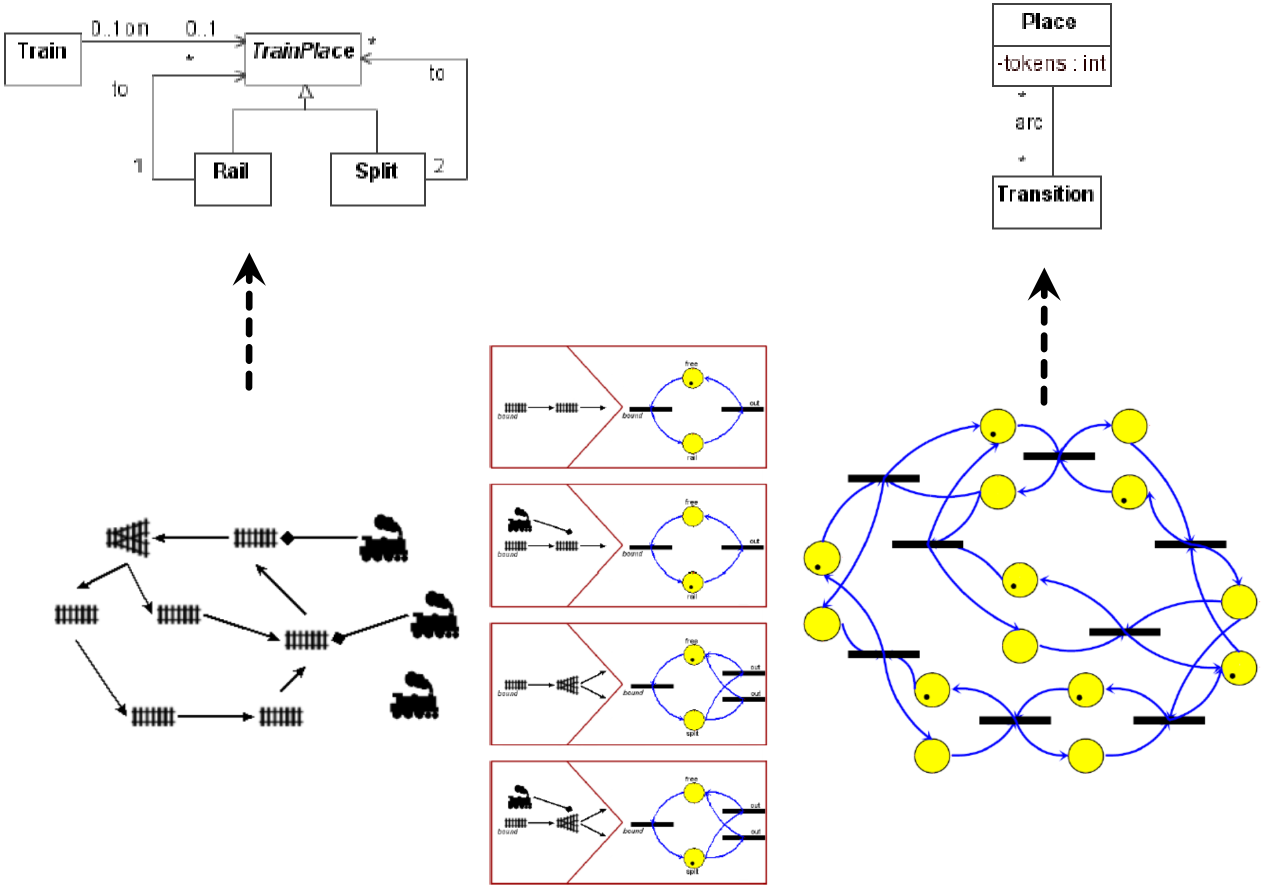
Automating co-evolution

$$\begin{tikzcd}
MM_D & MM \\
m \arrow[u, dotted] \arrow[r, "T"] & T(m) \arrow[u, dotted]
\end{tikzcd}$$

basic co-evolution architecture

complete

Example: TrainSim



Automating co-evolution

$$\begin{tikzcd}
MM_D & MM_I \arrow[dr, "\Delta MM_I"] & \\
& & MM_{I'} \\
m \arrow[uu, dotted] \arrow[r, "T"] & T(m) \arrow[uu, dotted] &
\end{tikzcd}$$

image evolution

difference model

Automating co-evolution

$$\begin{tikzcd} MM_D & MM_I \arrow[dr, "\Delta MM_I"] & \\ & & MM_{I'} \\ m \arrow[uu, dashed] \arrow[r, "T"] & T(m) \arrow[uu, dashed] & \end{tikzcd}$$

need for **co-evolution** of instance models and transformation models

Automating co-evolution

$$\begin{tikzcd}
MM_D & MM_I \arrow[dr, "\Delta MM_I", ""{name=D}] & \\
 & & MM_{I'} \\
m \arrow[uu, dotted] \arrow[r, "T"] & T(m) \arrow[uu, dotted] \arrow[dr, "E"', ""{name=E}] & \\
 & & T(m)_E \arrow[uu, dotted]
\arrow[from=D, to=E, dashed]
\end{tikzcd}$$

migration of instance models

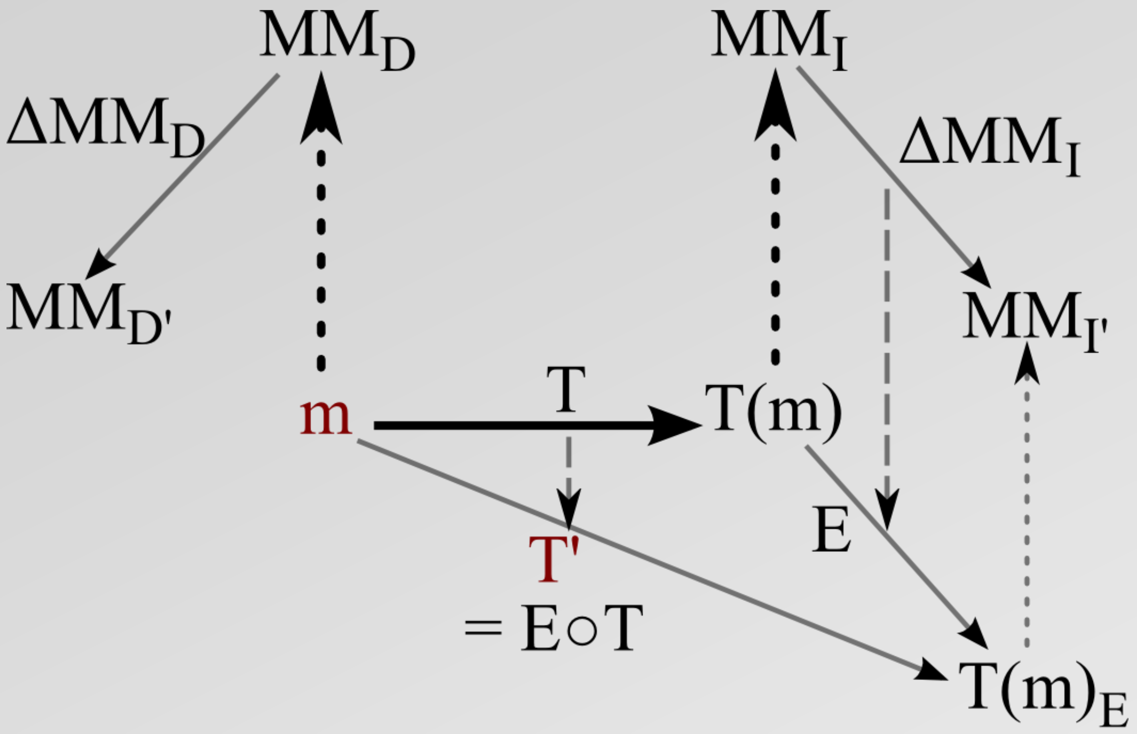
Automating co-evolution

$$\begin{tikzcd}
MM_D & & MM_I \arrow[dr, "\Delta MM_I", ""{name=D, below}] & \\
& & & MM_{I'} \\
m \arrow[uu, dotted] \arrow[rr, "T", ""{name=T, below}] \arrow[drrr, "{T' = E \circ T}"', ""{name=TP}] & & T(m) \arrow[uu, dotted] \arrow[dr, "E"', ""{name=E}] & \\
& & & T(m)_E \arrow[uu, dotted]
\arrow[from=D, to=E, dashed]
\arrow[from=T, to=TP, dashed]
\end{tikzcd}$$

migration of transformation models

transformation models remain **untouched**!

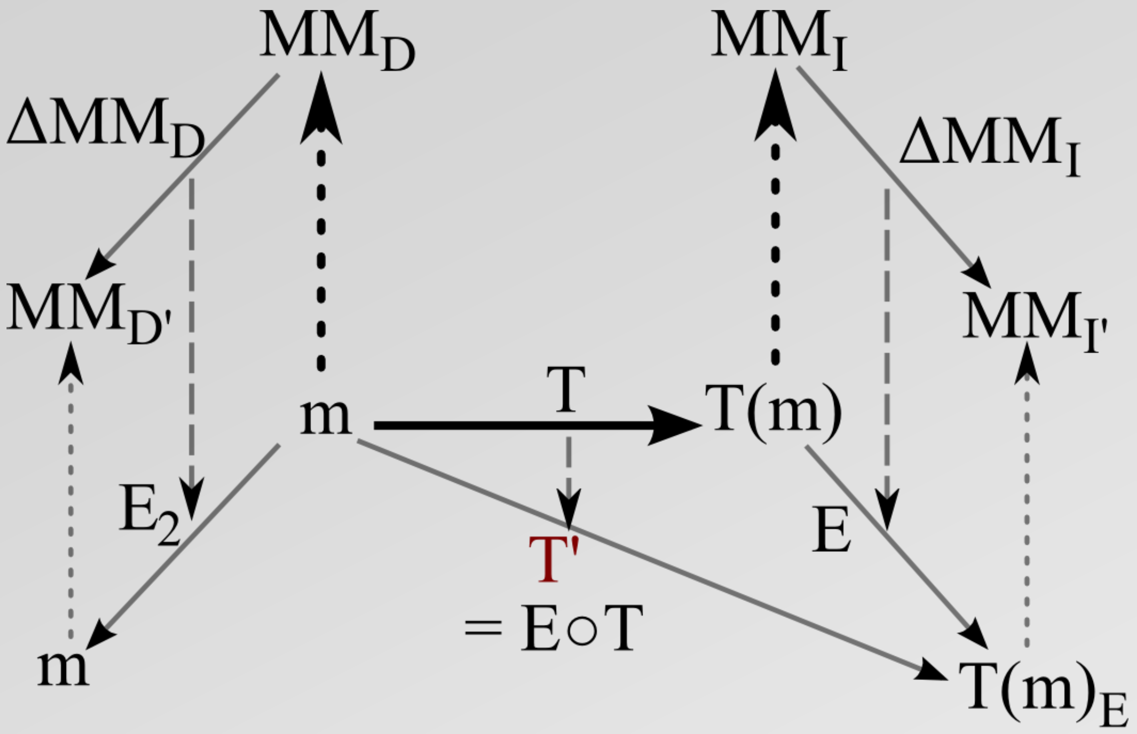
Automating co-evolution



domain evolution

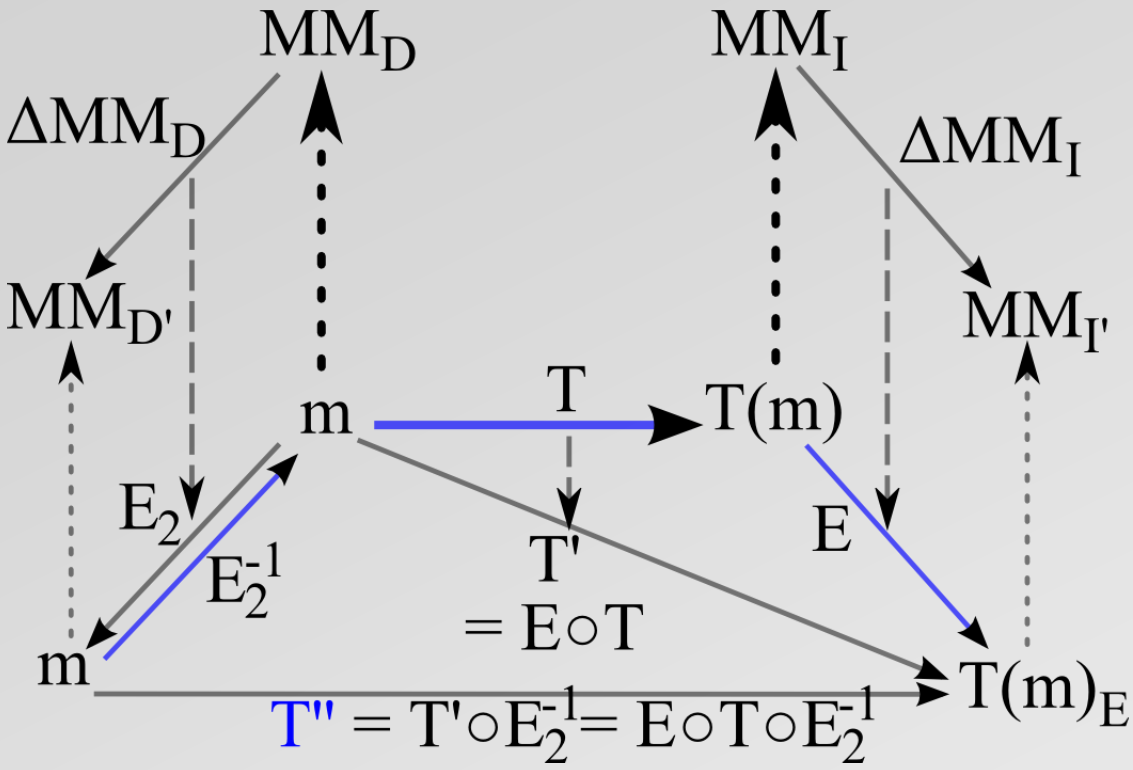
need for **co-evolution** of instance models and transformation models

Automating co-evolution



migration of
instance models

Automating co-evolution



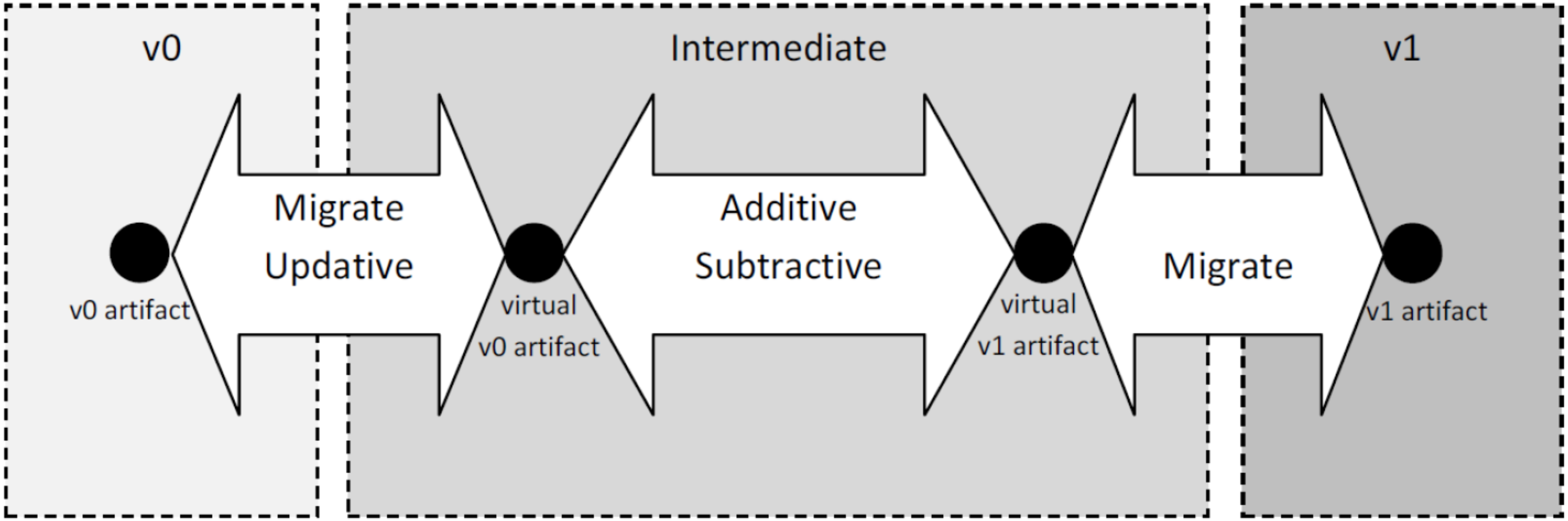
migration of transformation models

transformation models remain **untouched**!

Semantic evolution

- migrating back to previous version
  - **lose** information that was added to the new version
  - solution: “Intermediate Meta-Model” (IMM)
    - combination of both versions
    - all models of both versions can be expressed using IMM

Migrating artifacts: the migration pipeline



migration of artifacts over the migration pipeline

bi-directional

change, combine or remove parts of the migration pipeline

Challenges

- ensure consistency
- ensure continuity
- comes down to semantics
    - we need to formally represent semantics!
        - semantic mappings?
        - “properties” of a model?
    - perform compare (==), difference (-), ... operations on models of semantics
    - more automation possible?