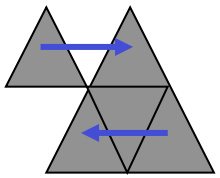
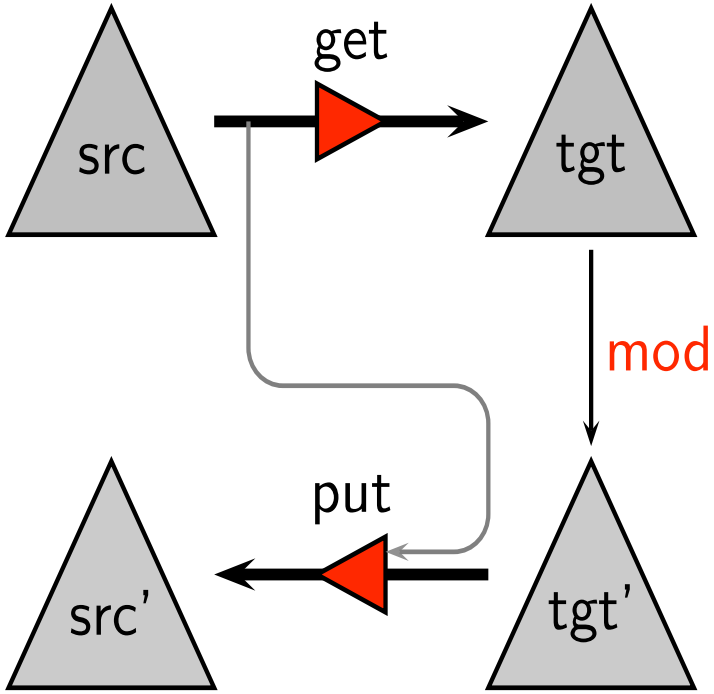
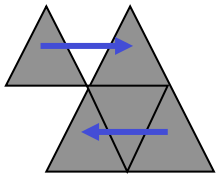
Bidirectional Model Transformations for High-Confidence Software Evolution

Zhenjiang Hu
National Institute of Informatics
December 17, 2011

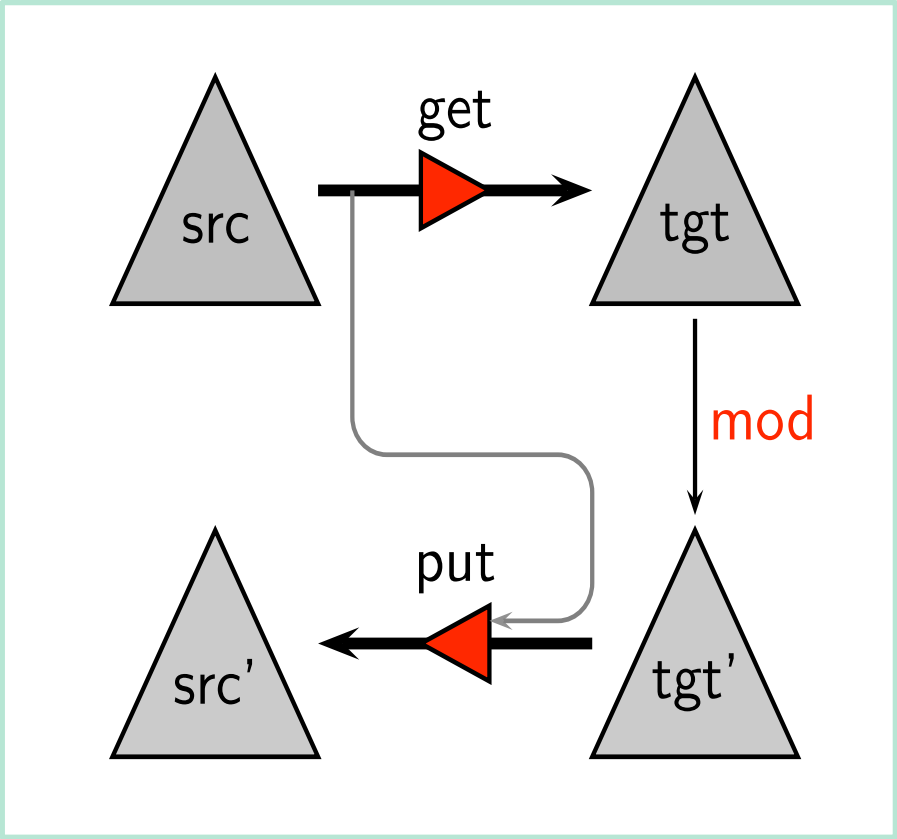


Bidirectional Transformation





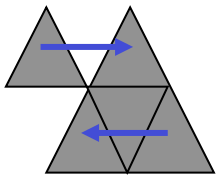
Roundtrip Properties



The result is not READ-ONLY.
It is CHANGEABLE!

Get-Put:
put (s, get (s)) = s

Put-Get:
get (put(s,t)) = t

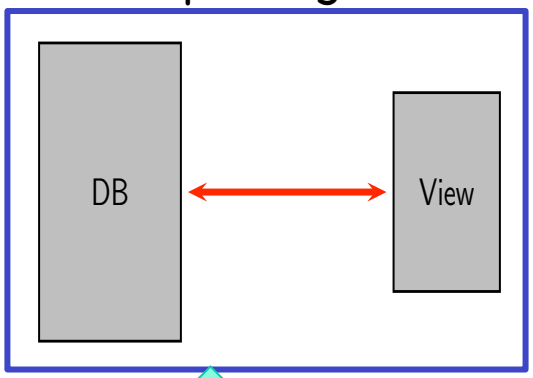


Pervasive Bidirectional Transformation

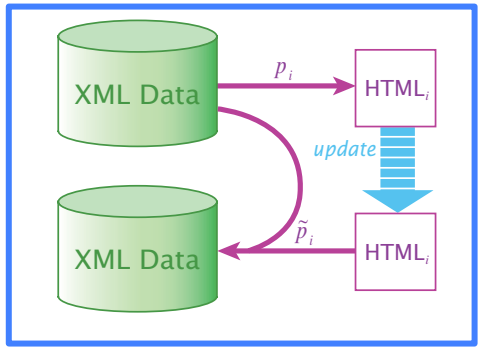
Data Synchronization (2006)

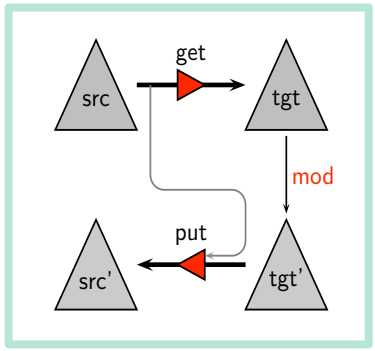
Roundtrip Engineering

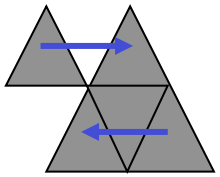
View Updating (80')



Web Maintainer (2008)

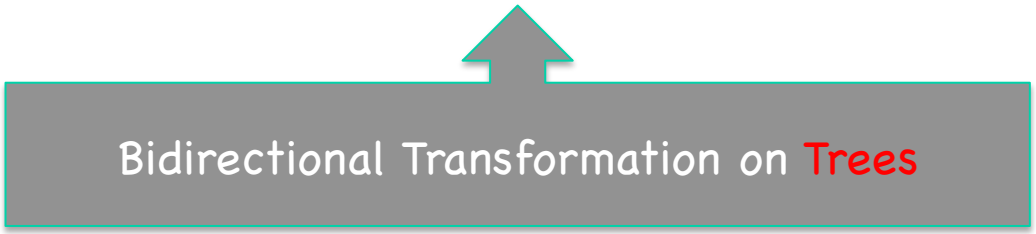


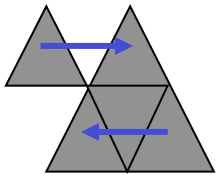




Bidirectional Transformation Languages

- We need languages to support development of software with bidirectional computation
  - Lens (for data sync) Univ. of Pennsylvania
    [POPL'06, PODS'06, ICFP'08, ICFP'10, POPL'11,12]
  - X/Inv/BiX (for doc construction) Univ. of Tokyo / NII
    [PEPM'04, ICFP'07, ESOP'10, ICFP'10]

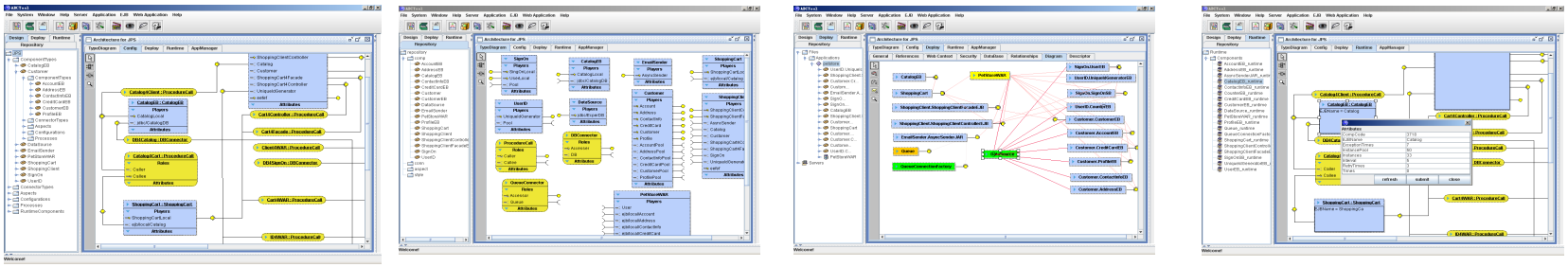


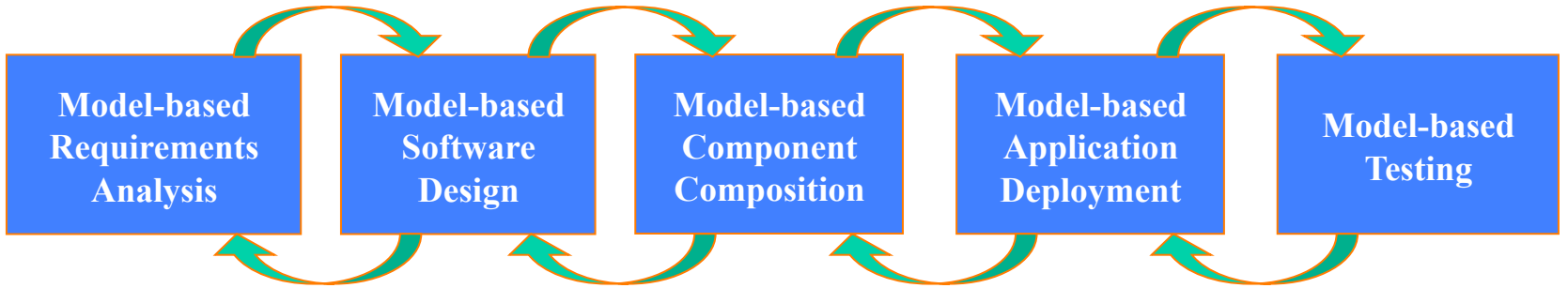


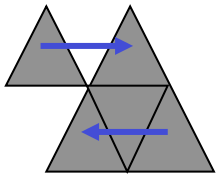
Bidirectional Model Transformation

Models: Graphs

Bidirectional Model Transformation: Bidirectional Graph Transformation

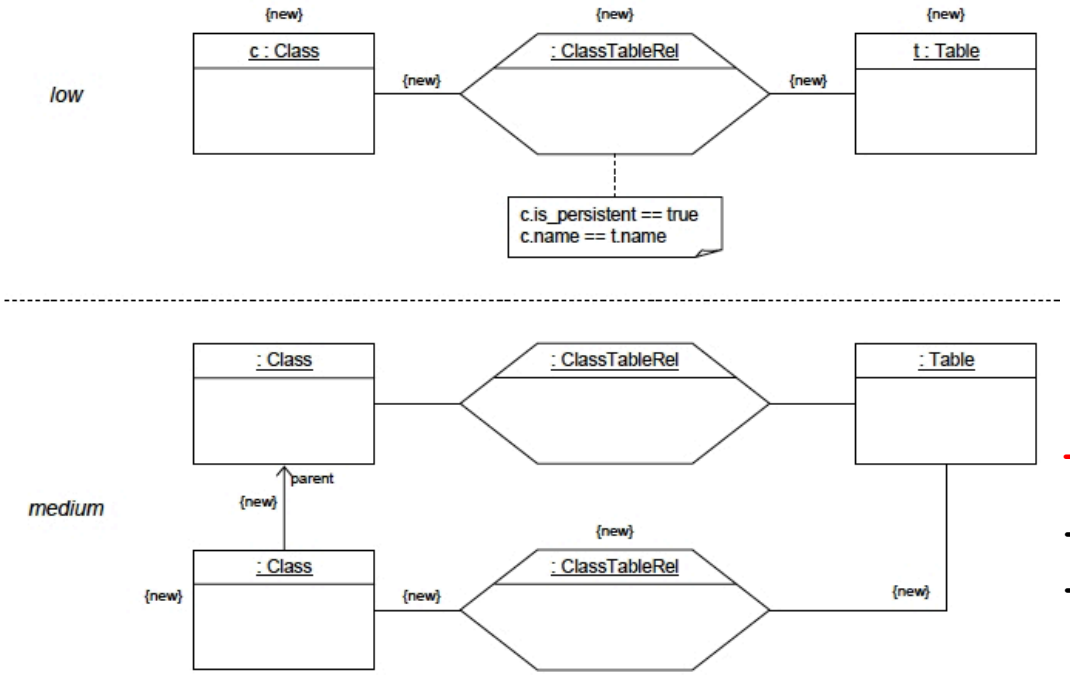




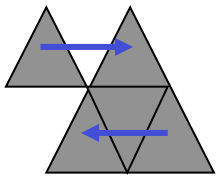


TGG Framework [Schurr et al.]

- TGG (triple graph grammar) + Rule-based
  - MOFLON
  - QVT/ATL: Query / View / Transformation



Two issues:
- compositional
- Well-behavedness

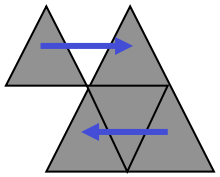


Domain Specific Frameworks

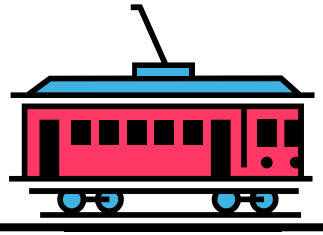
- Bidirectionalization of a subset of ATL (Xiong et al. ASE'07)

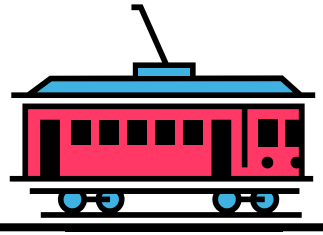
+ BX

- Beanbag: model synchronization (Xiong et al. FSE'09)

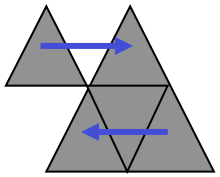


GRoundTram: Graph Roundtrip Transformation

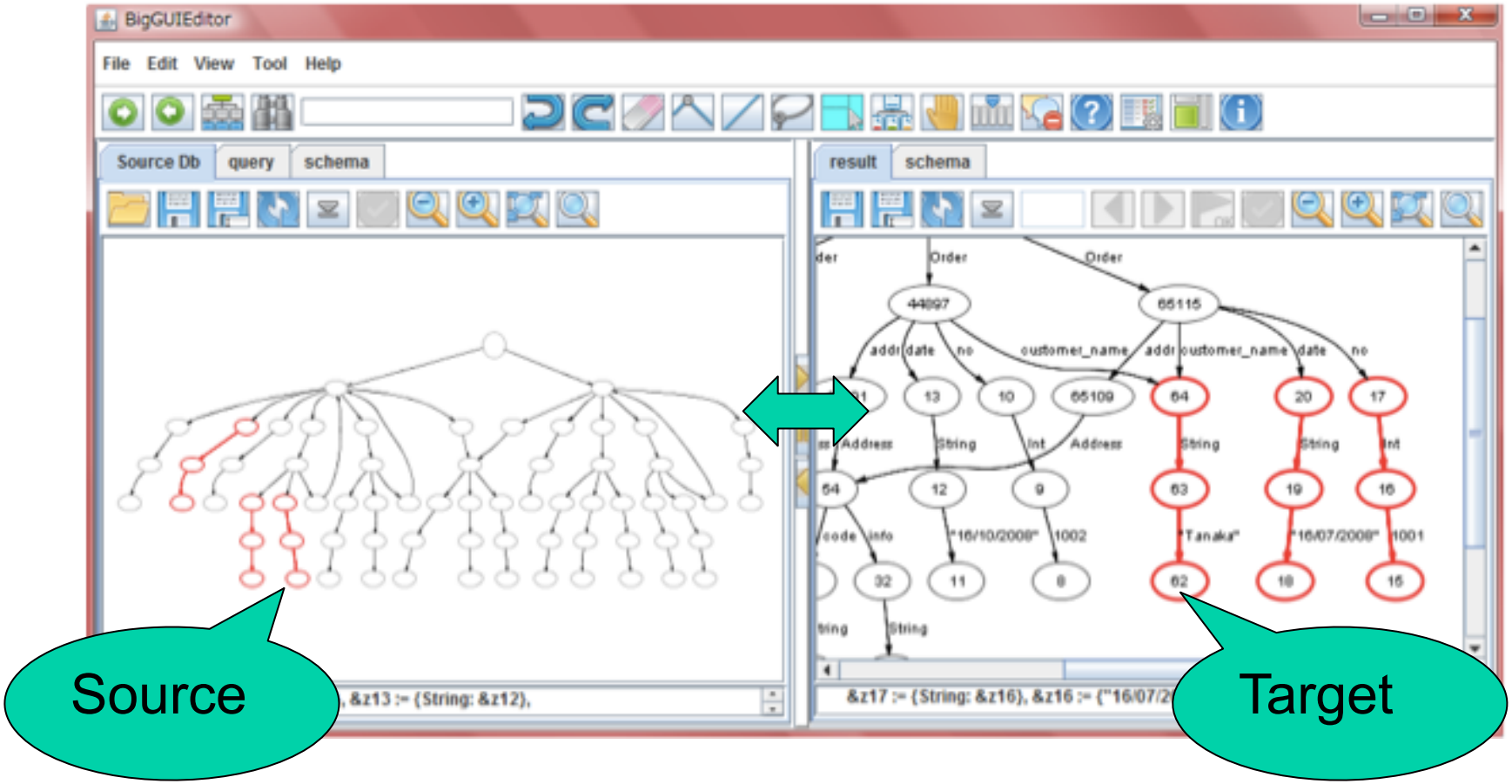




- It is compositional
  - Based on a practical (functional) graph query language
  - Allowing intermediate models
- It is well-behaved
  - Built upon bidirectional UnCAL: a graph algebra with clear bidirectional semantics
- It is an integrated development environment
  - Graph editor, graph validation, graph transformation checking, visualizations of bidirectional behavior



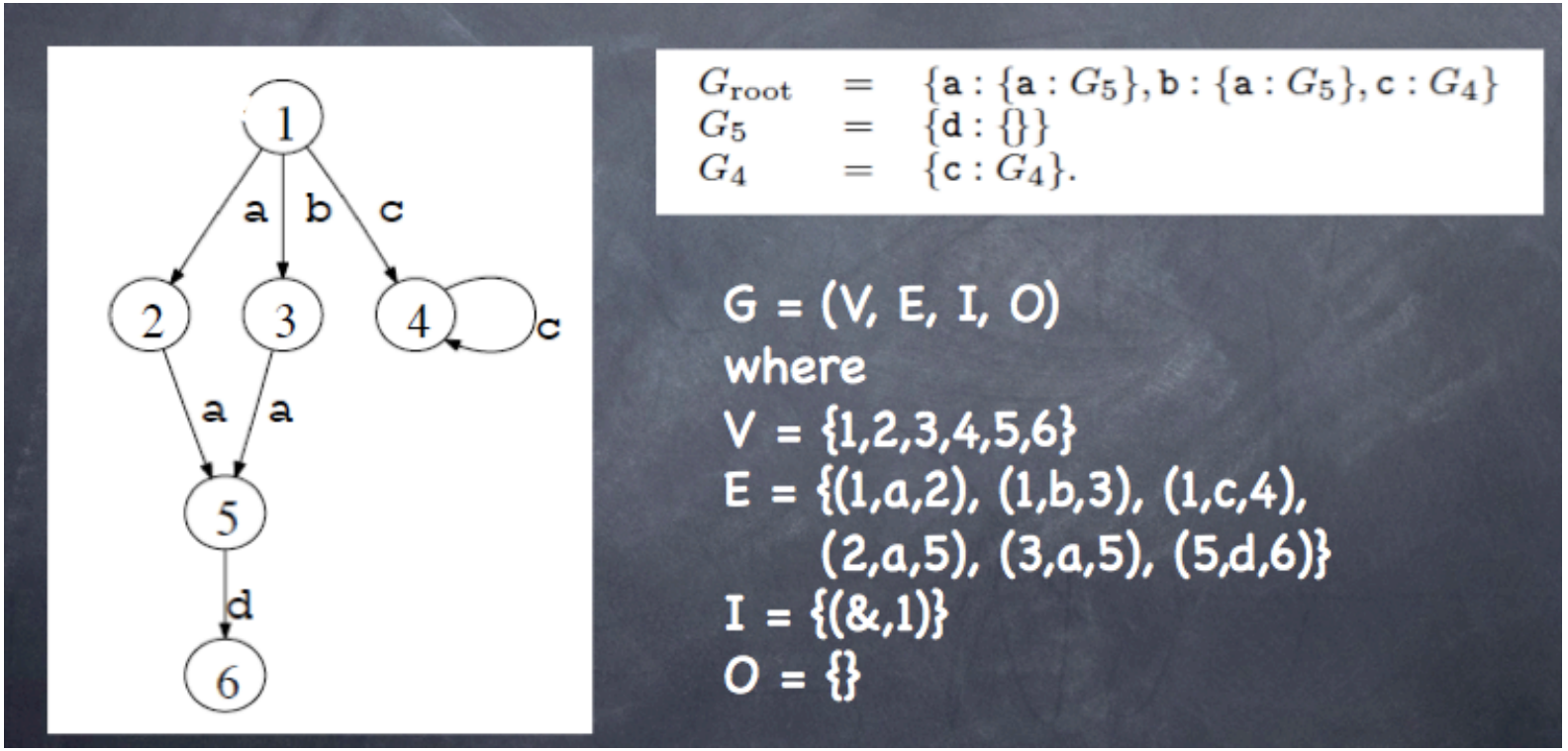
GRoundTram: Snapshot

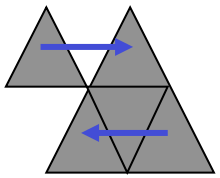


Graph Roundtrip Transformation of Models

Graph Model

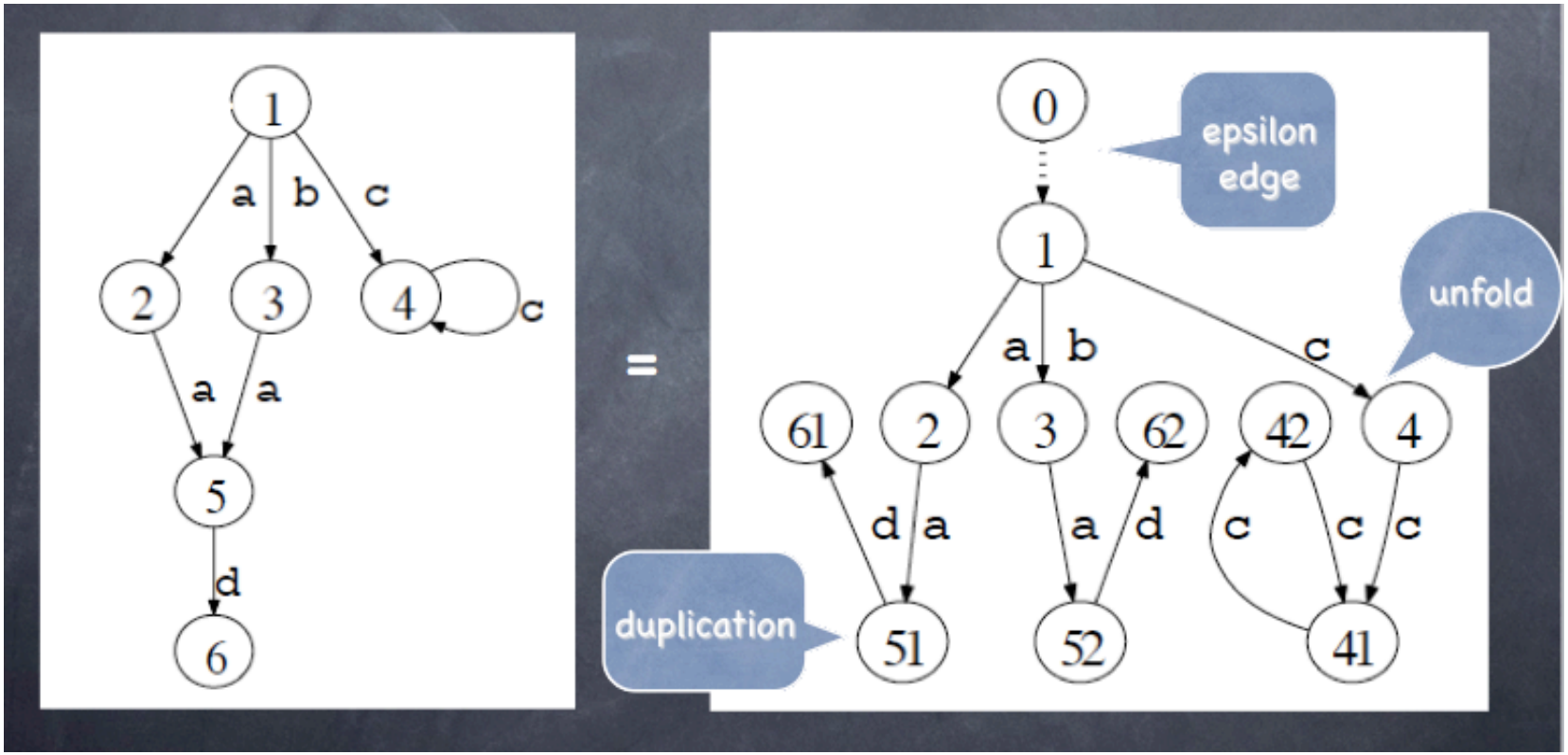
- Rooted Edge-labeled Graph





Graph Model

- Graph Equivalence based on Bisimulation



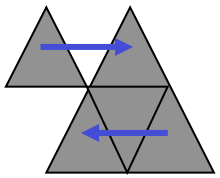
General enough, tree-based graphs

Forward Graph Transformation in UnQL+

An XQuery-like language

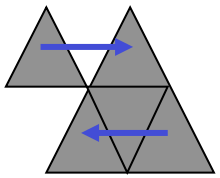
$$\texttt{select}\ \{\texttt{result}:\$G_1\}\ \texttt{where}\ \{\_*.(\texttt{a}|\texttt{b}):\$G_1\}\ \texttt{in}\ \$db,\ \{\$l:\$G_2\}\ \texttt{in}\ \$G_1,\ \$l \neq \texttt{a}$$

```
replace _*.Class.name.String -> $u
by {("class_" ^ $name):{}} in $db
where {$name: $v} in $u
```

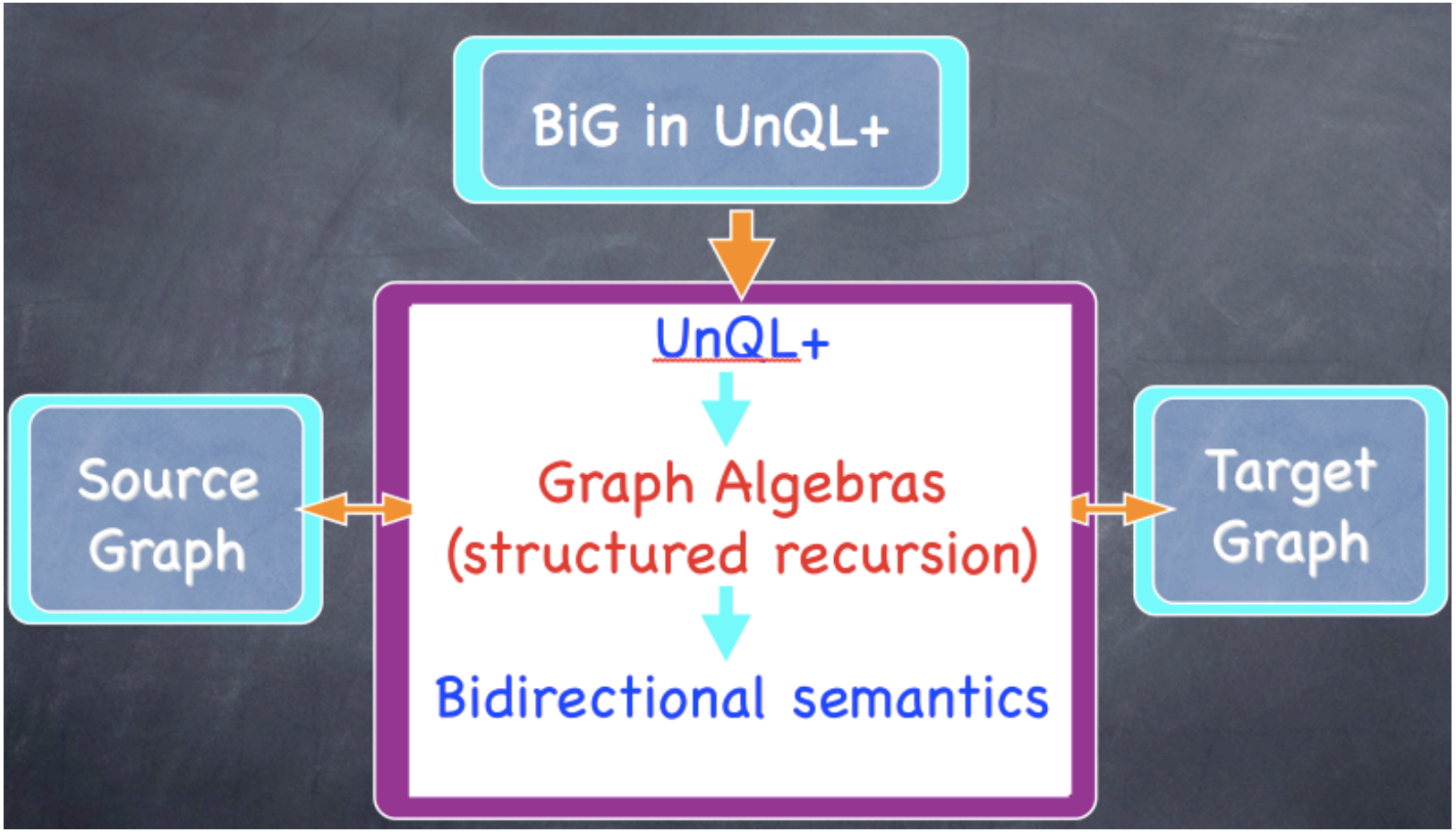



Main Functions of GRoundTram

- Automatic Graph Validation (ACM SAC 2009)
- Automatic verification of correctness of forward graph transformation (ACM PPDP 2011)
- Automatic construction of backward graph transformation (ACM ICFP 2010)
- Automatic Updatability Checking
- Efficient Implementation (ICSE 2010 (short), ASE 2011 (short), LOPSTR 2011)



Bidirectionalization of UnQL+ (ICFP'10)



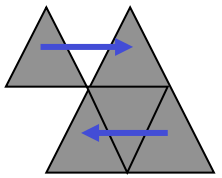
Structured Recursion

Structural Recursion:

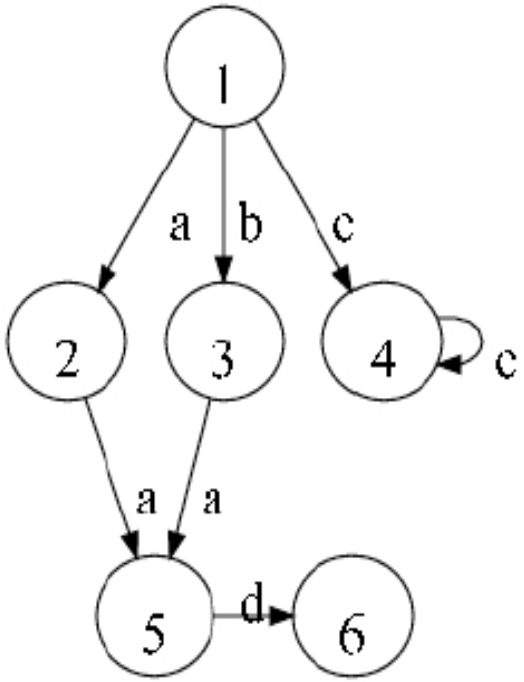
$$
\begin{aligned}
f\ (\{\}) &= \{\} \\
f\ (\{l : g\}) &= l \odot f(g) \\
f\ (g_1 \cup g_2) &= f(g_1) \cup f(g_2)
\end{aligned}
$$

Or written as:

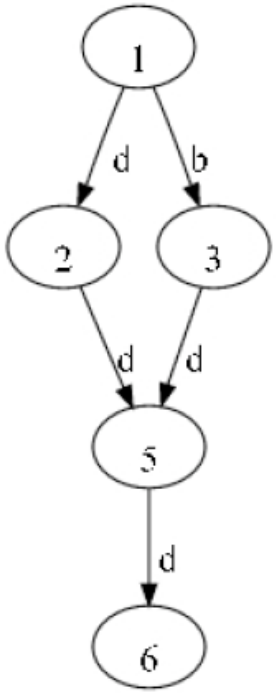
$$\text{sfun}\ \ f\ (\{l : g\}) = l \odot f(g)$$



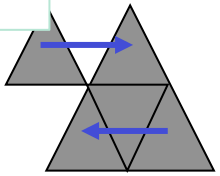
Structured Recursion



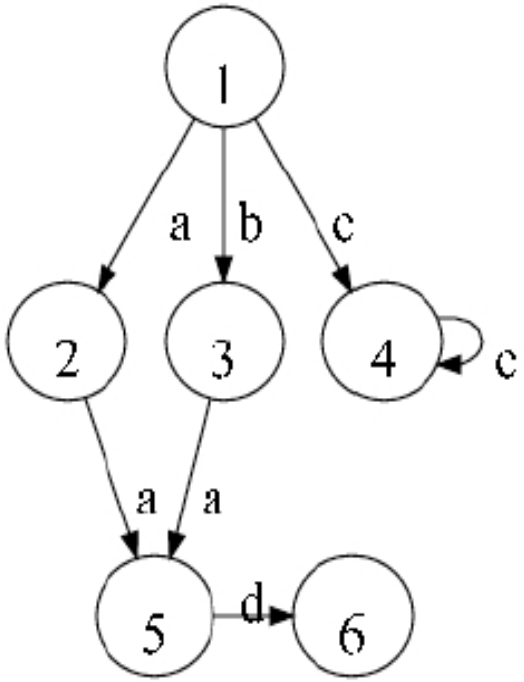
a2d_xc

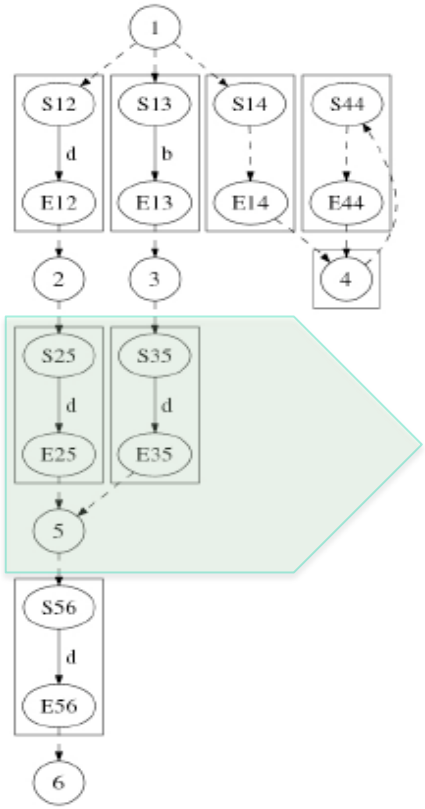


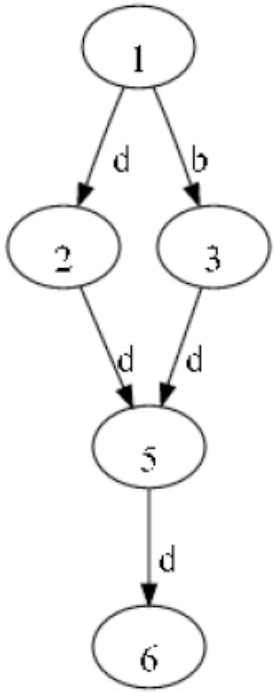
$$\text{sfun } a2d\_xc\ (\{l : g\}) = \begin{array}{l} \text{if } l = a \text{ then } \{d : a2d\_xc(g)\} \\ \text{else if } l = c \text{ then } a2d\_xc(g) \\ \text{else } \{l : a2d\_xc(g)\} \end{array}$$



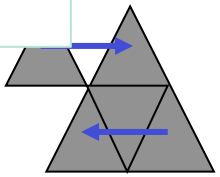
Structured Recursion



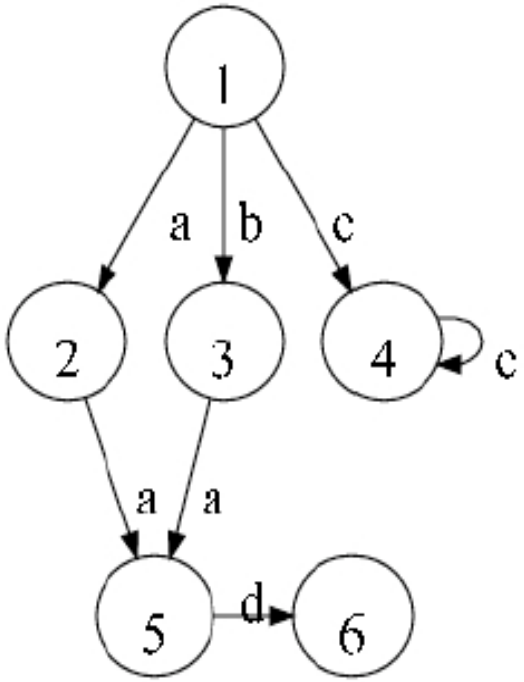


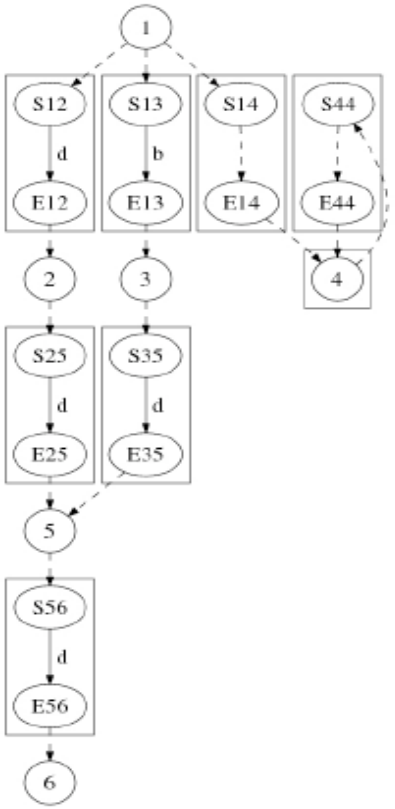


$$\text{sfun } a2d\_xc\ (\{l : g\}) = \text{if } l = a \text{ then } \{d : a2d\_xc(g)\} \text{ else if } l = c \text{ then } a2d\_xc(g) \text{ else } \{l : a2d\_xc(g)\}$$



Bidirectional Bulk Semantics

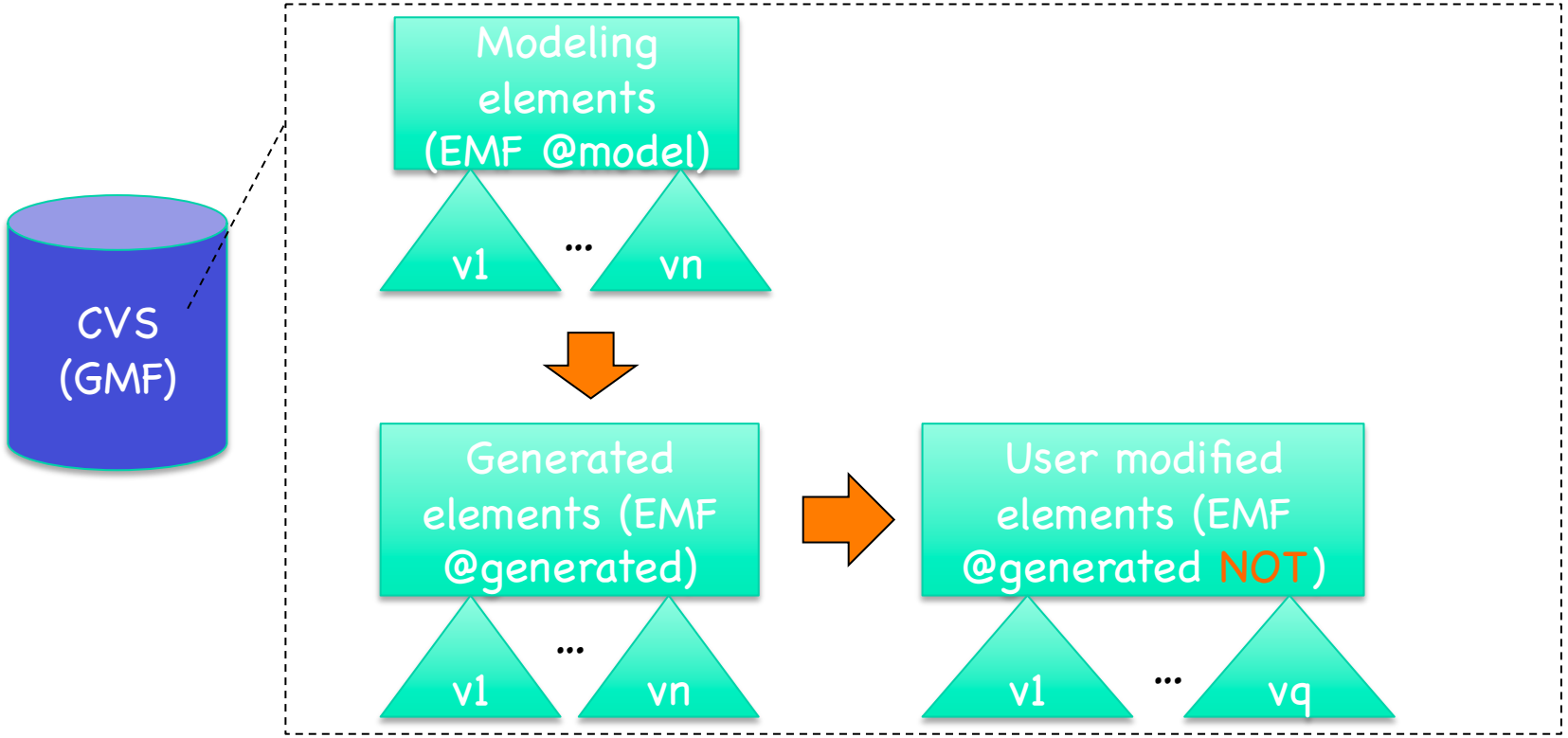




$$\begin{aligned}\text{sfun } a2d\_xc\ (\{l : g\}) \;=\; & \text{if } l = a \text{ then } \{d : a2d\_xc(g)\} \\ & \text{else if } l = c \text{ then } a2d\_xc(g) \\ & \text{else } \{l : a2d\_xc(g)\}\end{aligned}$$

An Application: blinkit

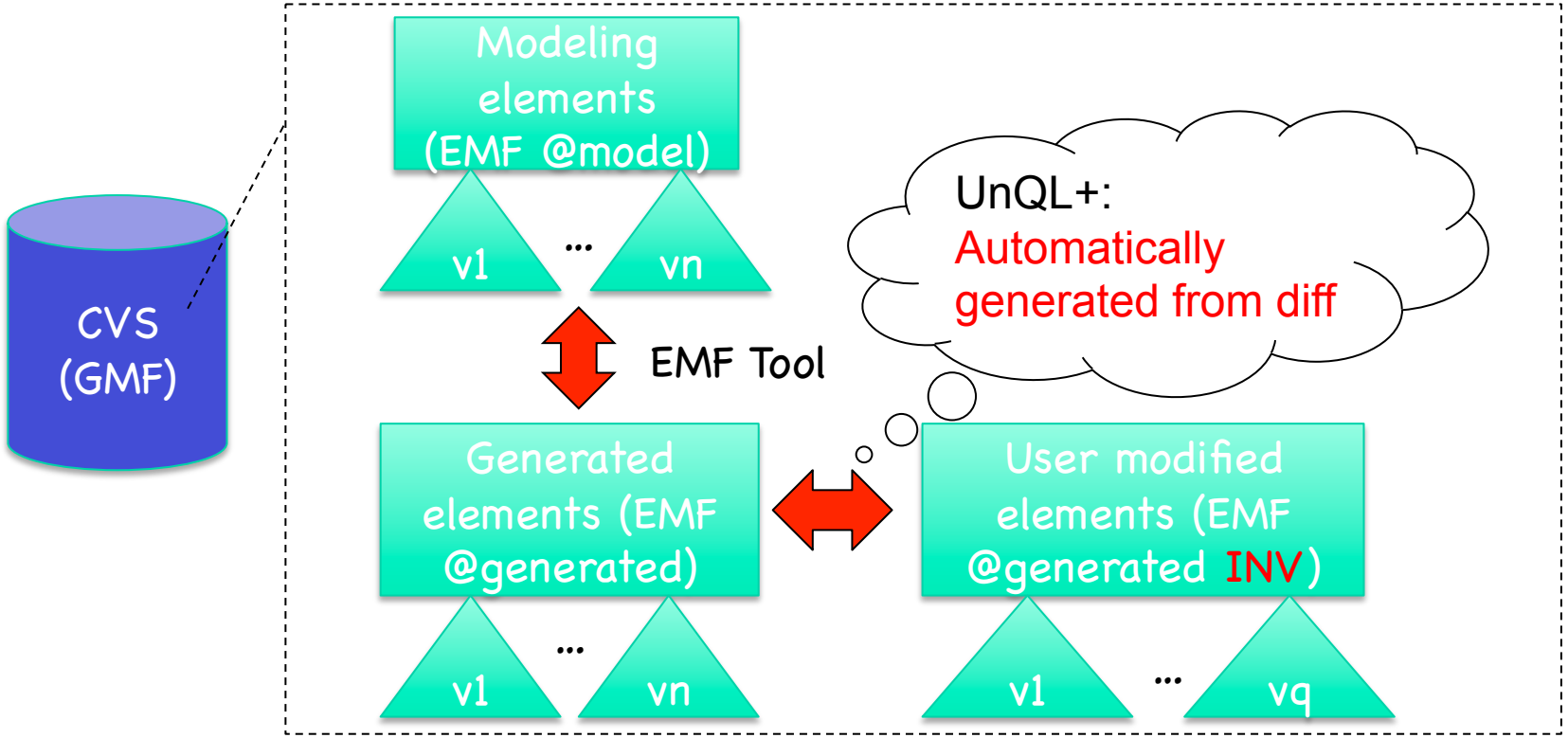
- A tool for supporting invariant traceability in model-driven development



(Joint work with Yijun Yu at Open Univ.)

An Application: blinkit

- A tool for supporting invariant traceability in model-driven development

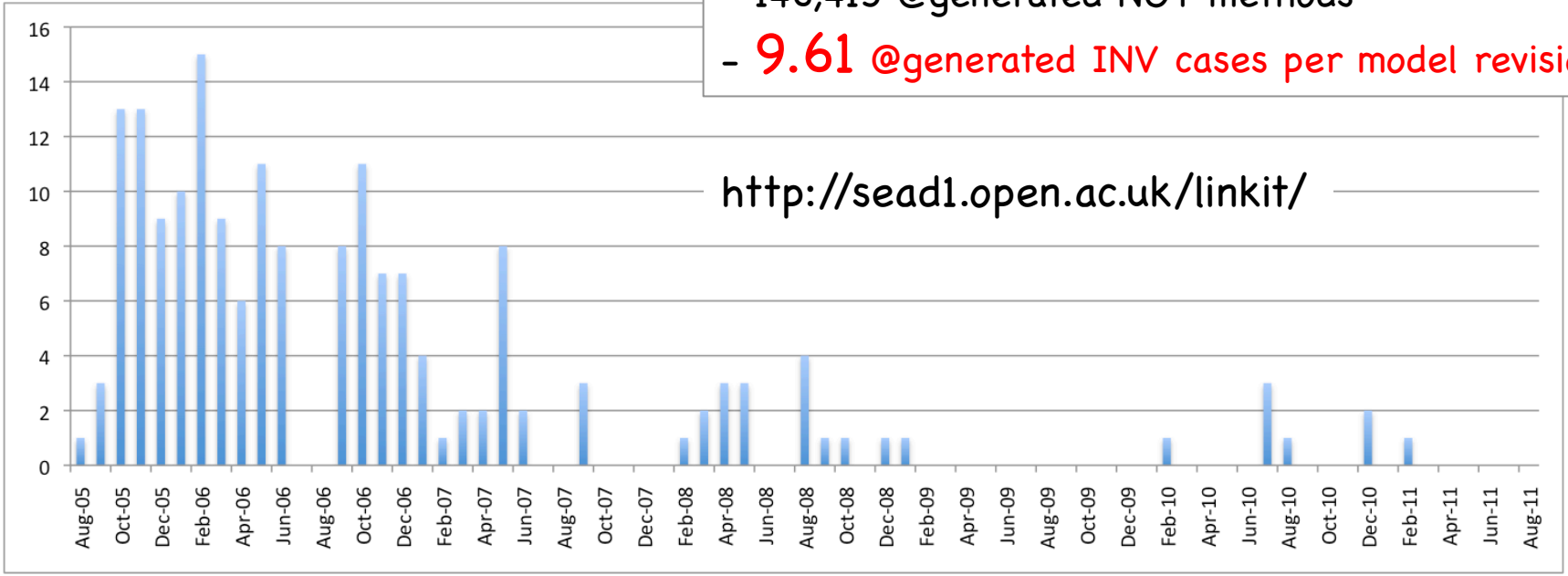


(Joint work with Yijun Yu at Open Univ.)

Evaluation

Are there realistic cases in an open-source MDD software development project where invariant traceability links can be synchronized using blinkit?

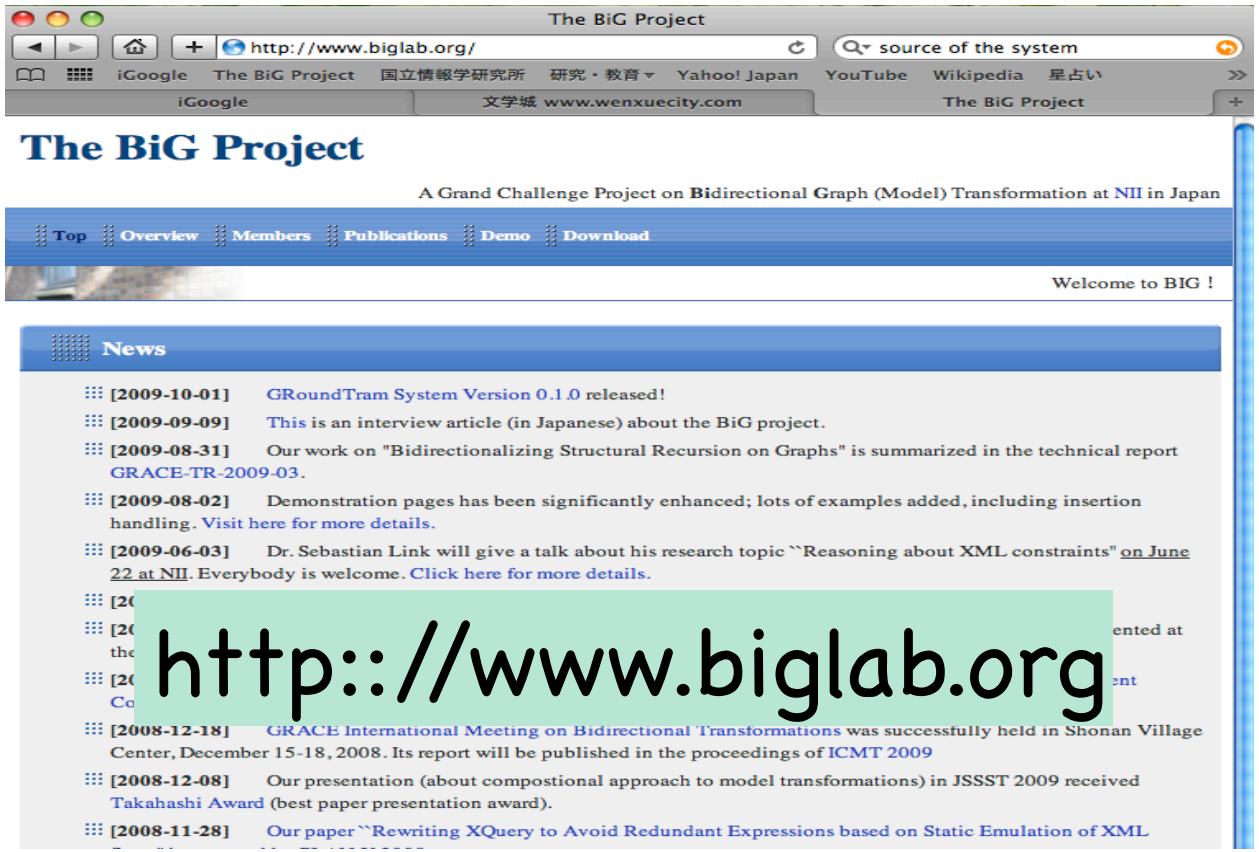
EMF/GMF
- 6 years CVS repository
- 28,070 file revisions
- 1,185 revision pairs
- 178 @model changes
- 146,415 @generated NOT methods
- 9.61 @generated INV cases per model revision



http://sead1.open.ac.uk/linkit/

BiG Project Web Site

For more information, please visit the project page which contains all published papers as well as the source codes of the GRoudTram system.



Conclusion

- Bidirectional transformations are pervasive!

- Bidirectional model (graph) transformations are challenging but promising!
  - Well-behavedness of BX
  - Scalablity
  - Killer Applications

SE + PL + DB