Bidirectional Graph Transformation Infrastructure and its Applications

Soichiro Hidaka*
National Institute of Informatics

EASSy 2013
September 10th, 2013

*and collaborators

Overview of works in NII

Req. analysis

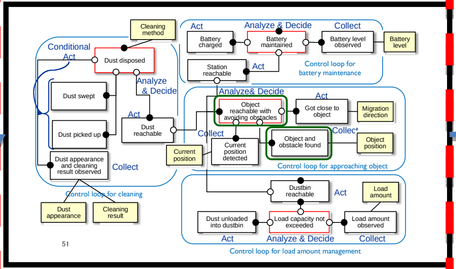
Architecture Design

Detailed Design

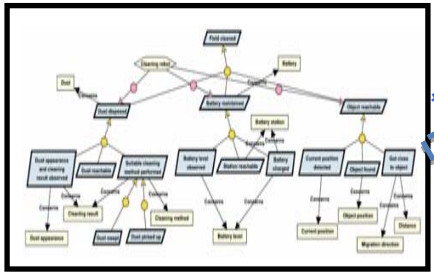
Impl.

・Exploration of adaptation space
Fuyuki Ishikawa

Adaptation space analysis



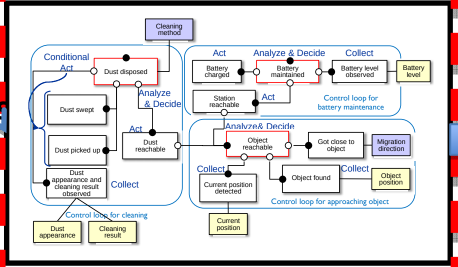
traceability link



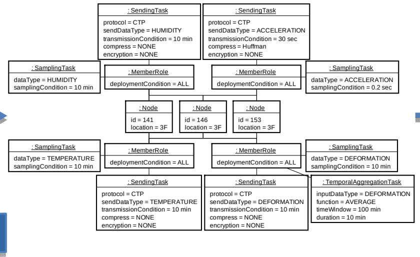
Traceability maintenance to localize changes

・Designing Self-adaptive System using Control Loops
Shinichi Honiden

adaptation



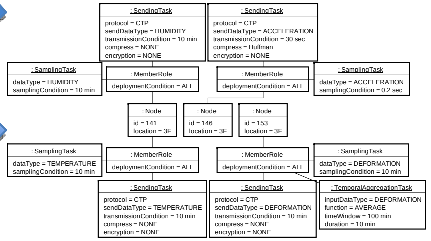
・Composition-based interaction design for adaptable distributed software systems
Kenji Tei



・Putback-based Bidirectional Programming
Zhenjiang Hu

・Bidirectional Graph Transformation Infrastructur
and its Applications
Soichiro Hidaka

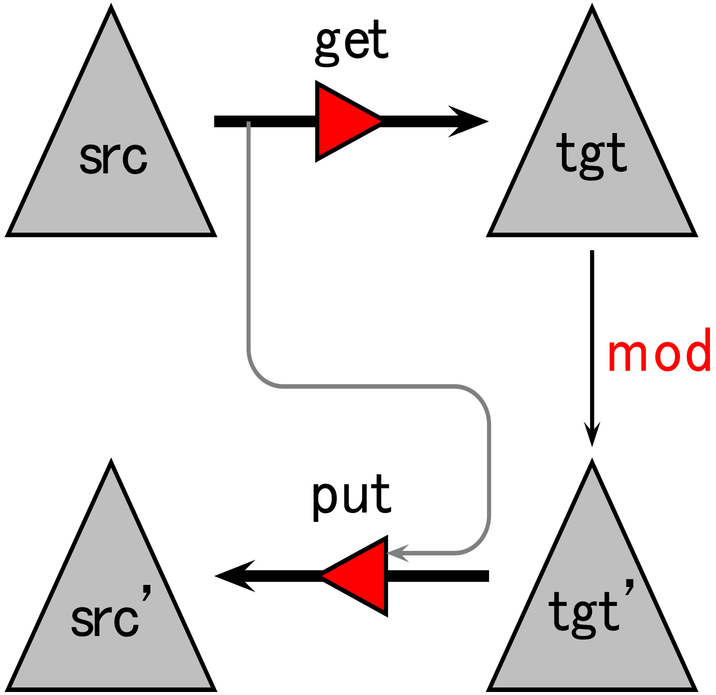
Change propagation



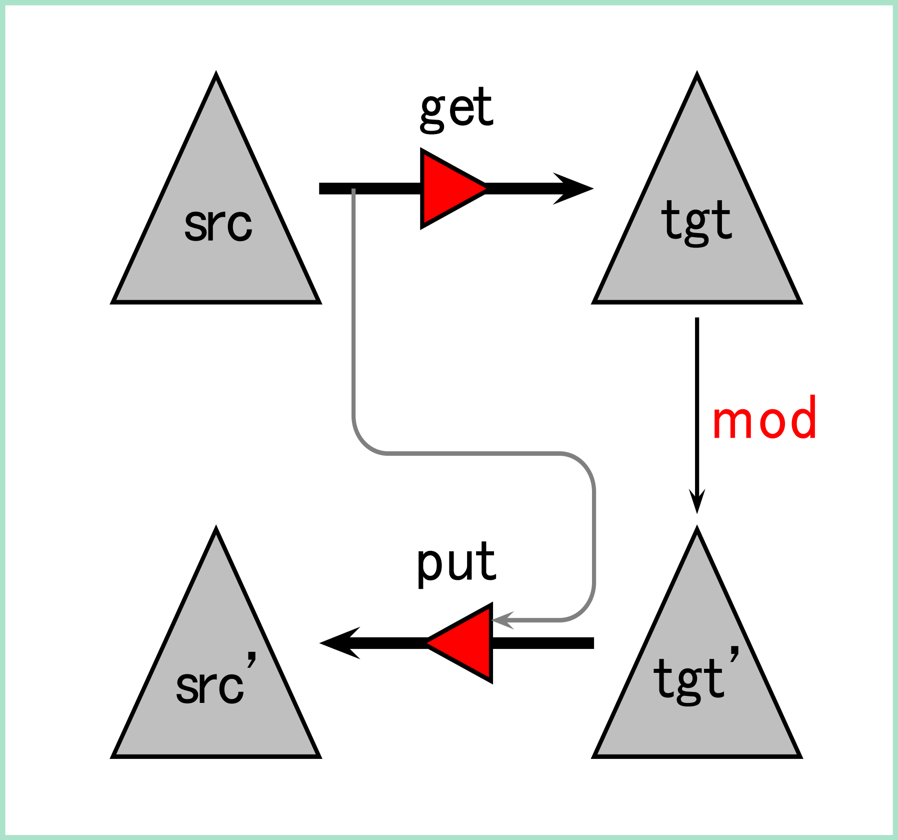
Bidirectional Transformation

Bidirectional Transformation (BX)

[Nate Foster, et al: POPL 2005]



Roundtrip Properties



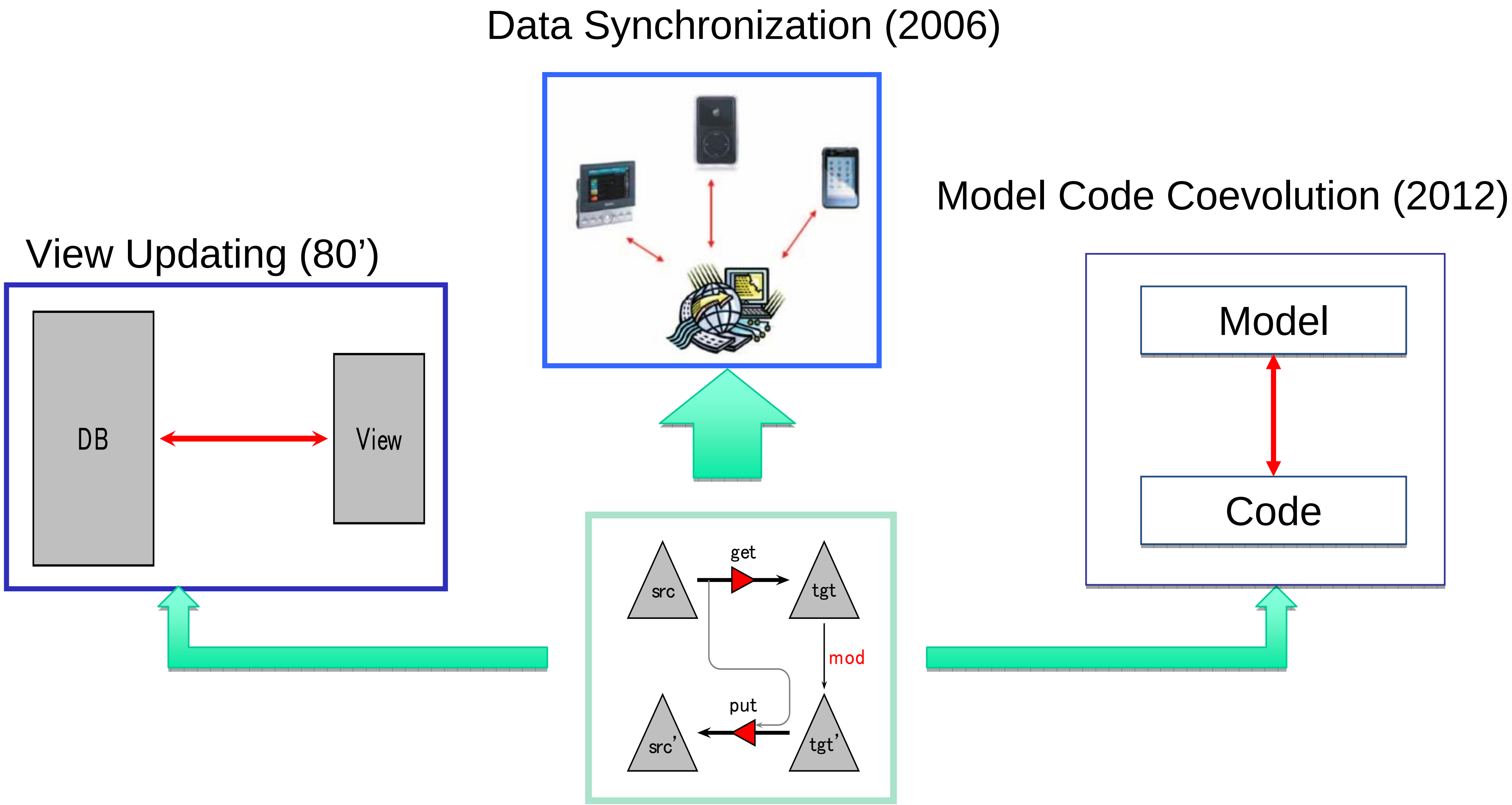
Get-Put:

put s (get s) ≡ s

Put-Get:

get (put s t) ≡ t

Pervasive Bidirectional Transformation



Data Synchronization (2006)

View Updating (80')

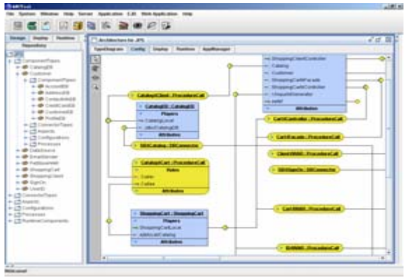
Model Code Coevolution (2012)

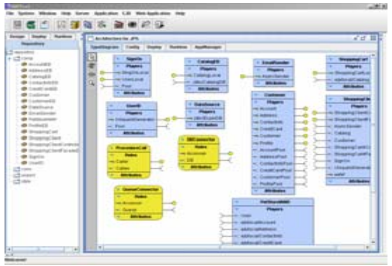
Bidirectional *Graph* Transformation

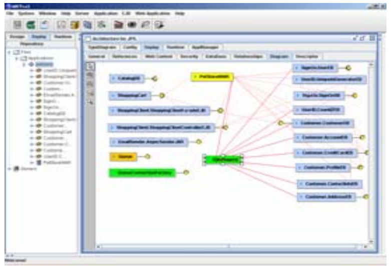
BX on Graphs is Wanted!

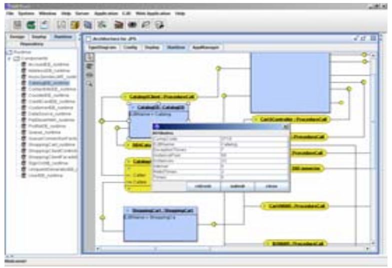
Models: Graphs

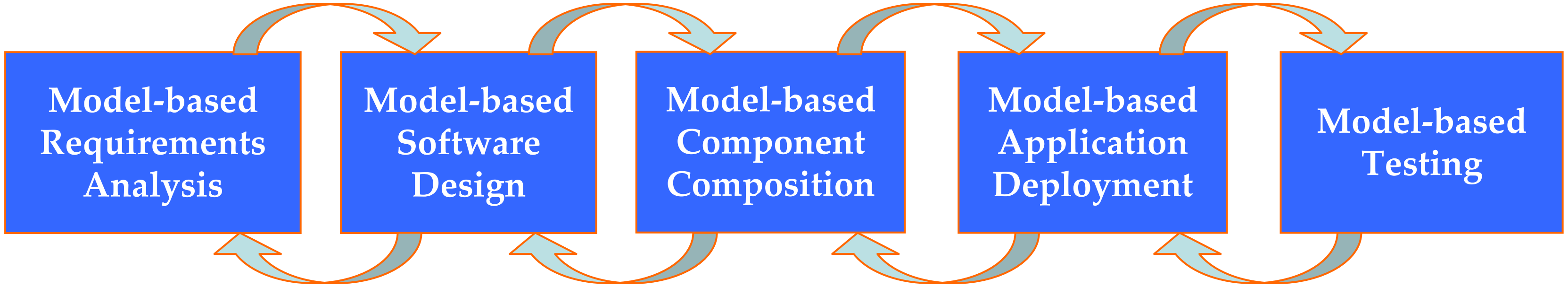
Bidirectional Model Transformation: Bidirectional Graph Transformation











Model-based Requirements Analysis

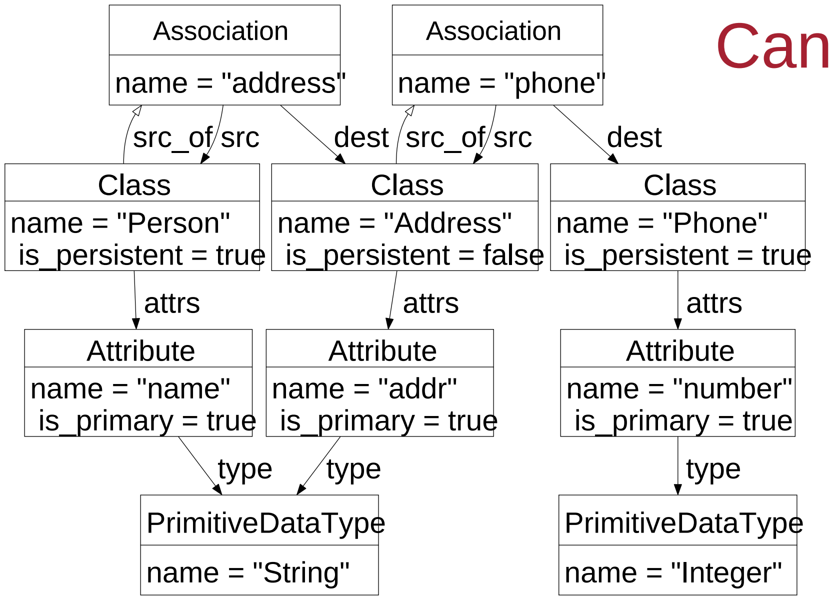
Model-based Software Design

Model-based Component Composition

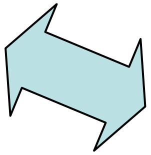
Model-based Application Deployment

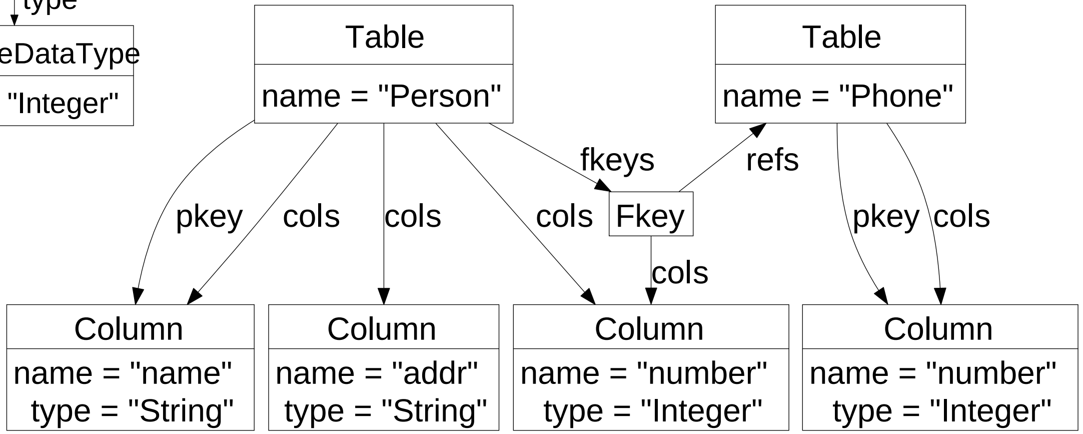
Model-based Testing

Bidirectional Model Transformation Example



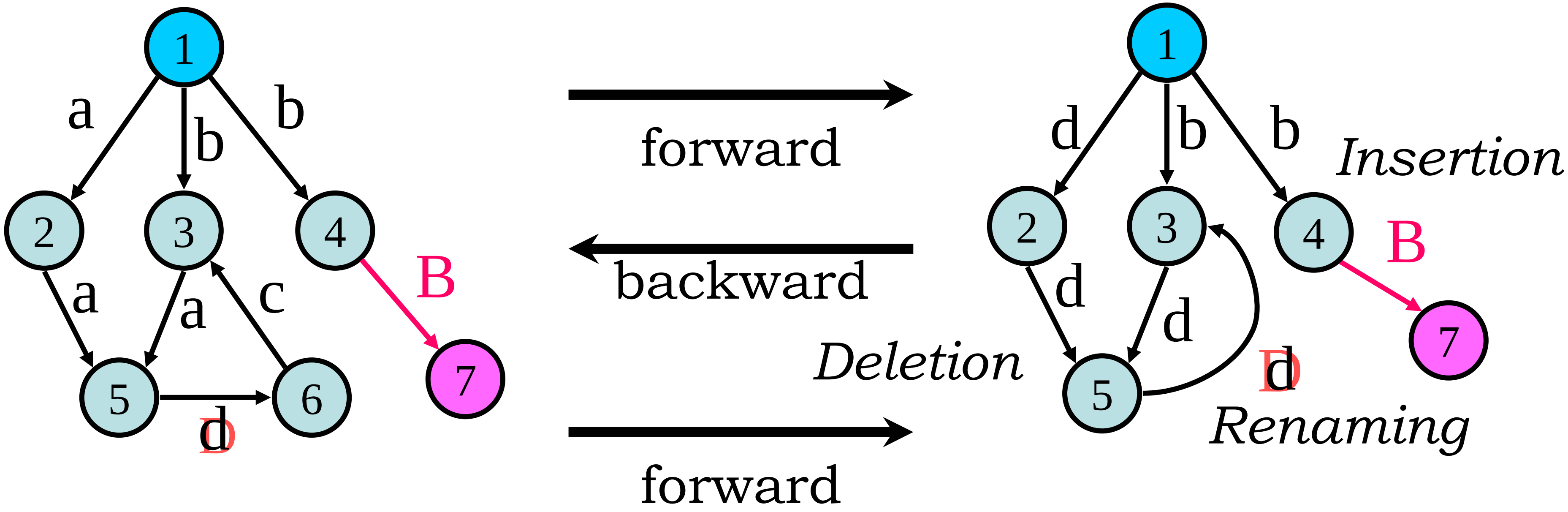
Can We Design a Language for BiG?





Example of BX on Graphs

- Replace ‘a’ by ‘d’ and removes ‘c’



forward

backward

forward

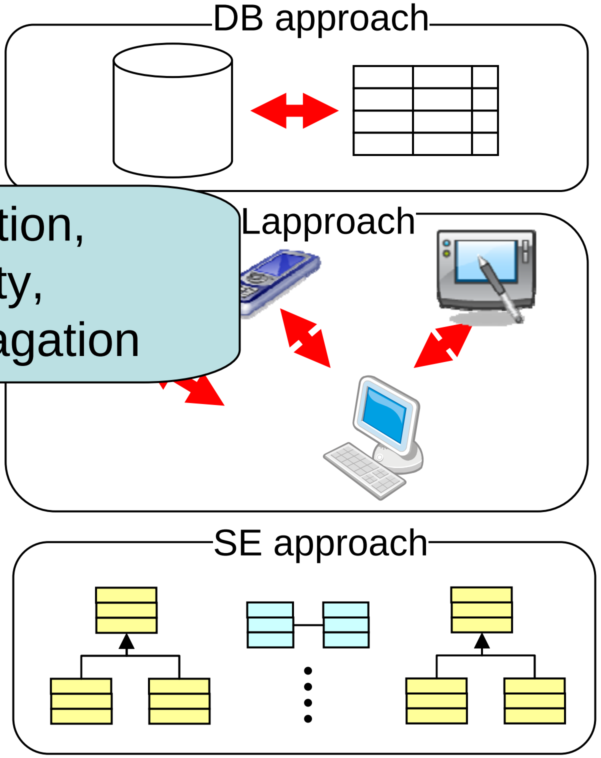
Insertion

Deletion

Renaming

Limitations in Existing Approaches

- Database/programming languages field
  - Targeted to tables, list
  - Graphs have not been addressed directly
  - Why? Graphs have: <u>cycles and sharing</u>
- Software engineering field (models = graphs)
  - Difficult to guarantee *well-behavedness (round-trip)* in compositional settings



Termination,
Equality,
Back-propagation

Towards a General Solution …

- How to deal with termination of graph transformation?

  ➔Structural Recursion and its bulk semantics

- How to deal with equality of two graphs?

  ➔ Bisimulation (graphs as regular trees)

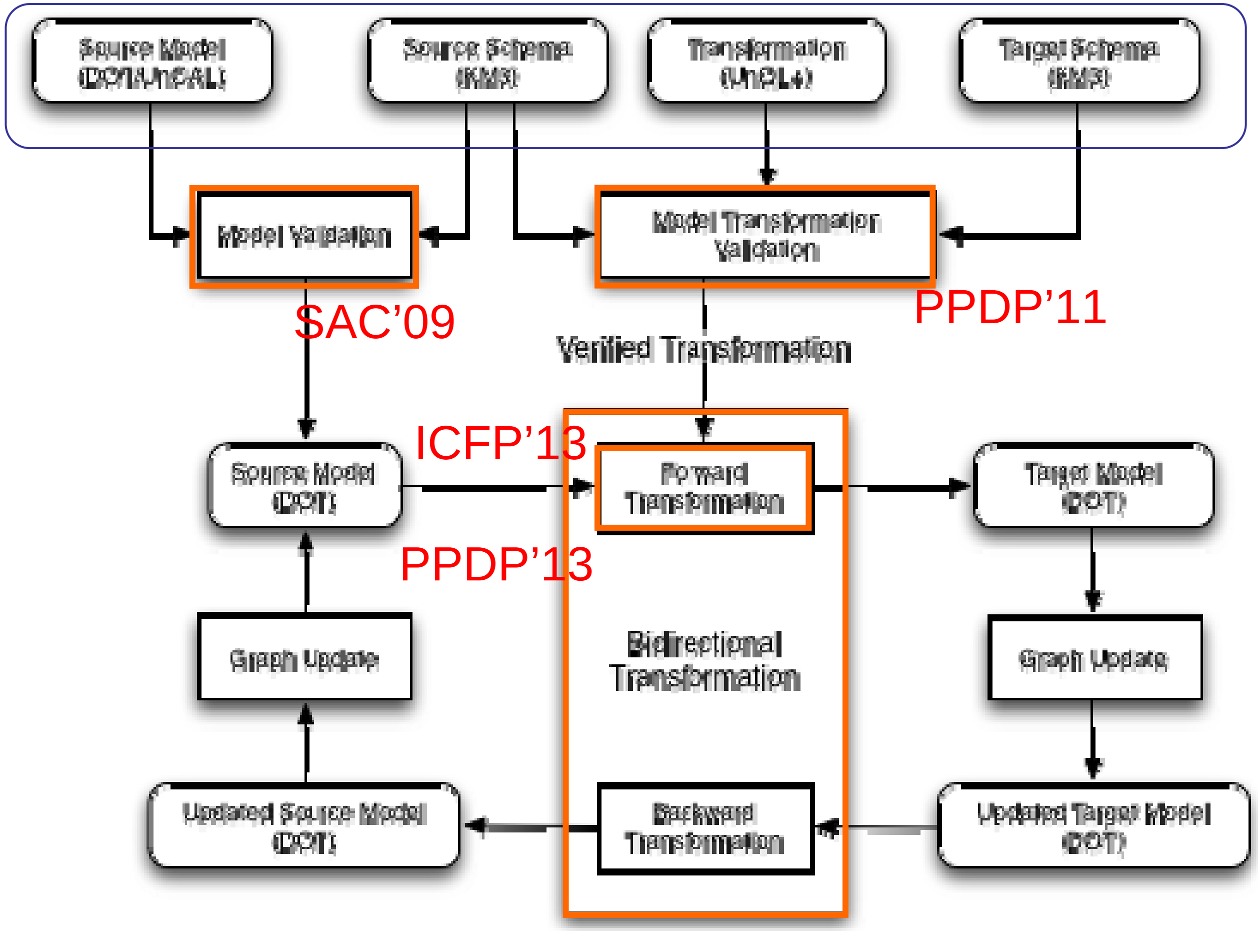
- How to correctly reflect changes on the view to the source?

  ➔ Traceability based on Bulk Semantics

GRoundTram: A General Functional Framework

- It is compositional (functional)
  - Based on the existing graph query language UnQL
- 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

Overview of GRoundTram



User Input

SAC'09

PPDP'11

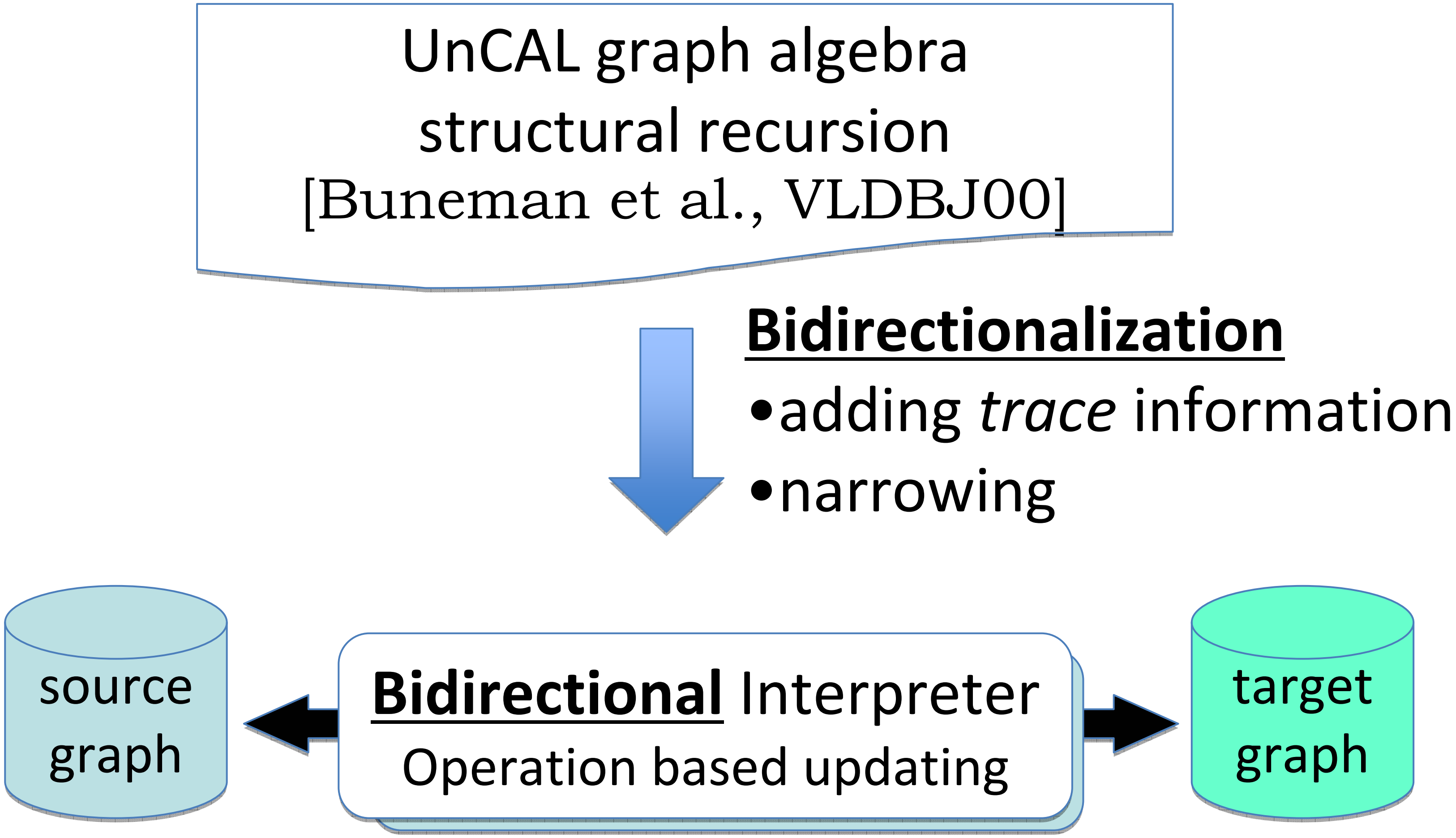
ICFP'13

PPDP'13

ICFP'10, LOPSTR'11

[ASE] S. Hidaka, Z. Hu, K. Inaba, H. Kato and K. Nakano, *GRoundTram: An Integrated Framework for Developing Well-Behaved Bidirectional Model Transformations* (short paper), ASE 2011

Our Approach:Bidirectionalization



UnCAL graph algebra
structural recursion
[Buneman et al., VLDBJ00]

<u>Bidirectionalization</u>

- adding *trace* information
- narrowing

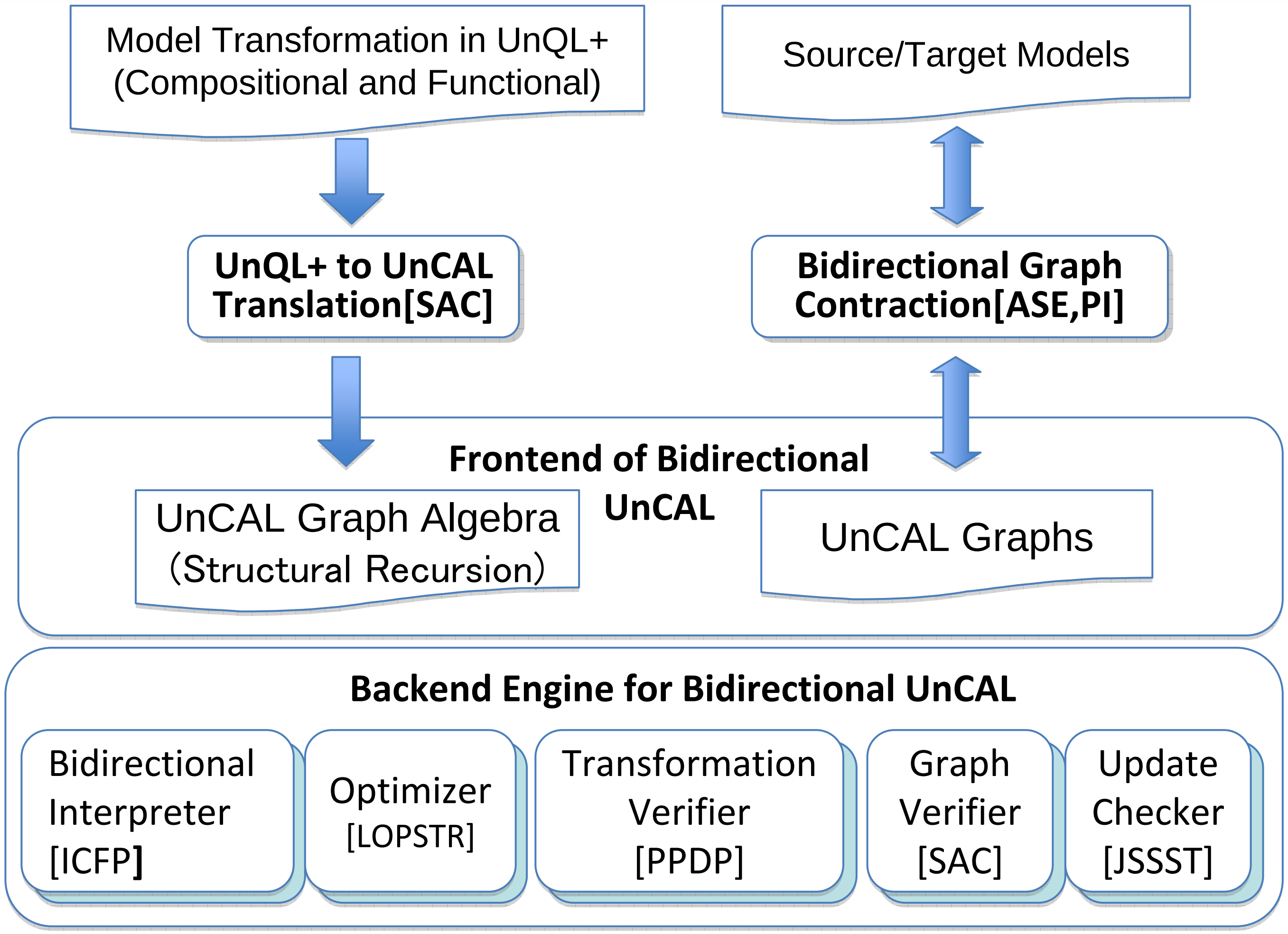
source graph

<u>Bidirectional</u> Interpreter
Operation based updating

target graph

[ICFP'10] <u>S. Hidaka</u>, Z. Hu, K. Inaba, H. Kato, K. Matsuda, K. Nakano, Bidirectionalizing Graph Transformations, 15th ACM SIGPLAN International Conference on Functional Programming, pp.205-216 Sep 2010

Architecture of GRoundTram


Model Transformation in UnQL+ (Compositional and Functional)

Source/Target Models

UnQL+ to UnCAL Translation[SAC]

Bidirectional Graph Contraction[ASE,PI]

Frontend of Bidirectional UnCAL

UnCAL Graph Algebra (Structural Recursion)

UnCAL Graphs

Backend Engine for Bidirectional UnCAL

Bidirectional Interpreter [ICFP**]**

Optimizer [LOPSTR]

Transformation Verifier [PPDP]

Graph Verifier [SAC]

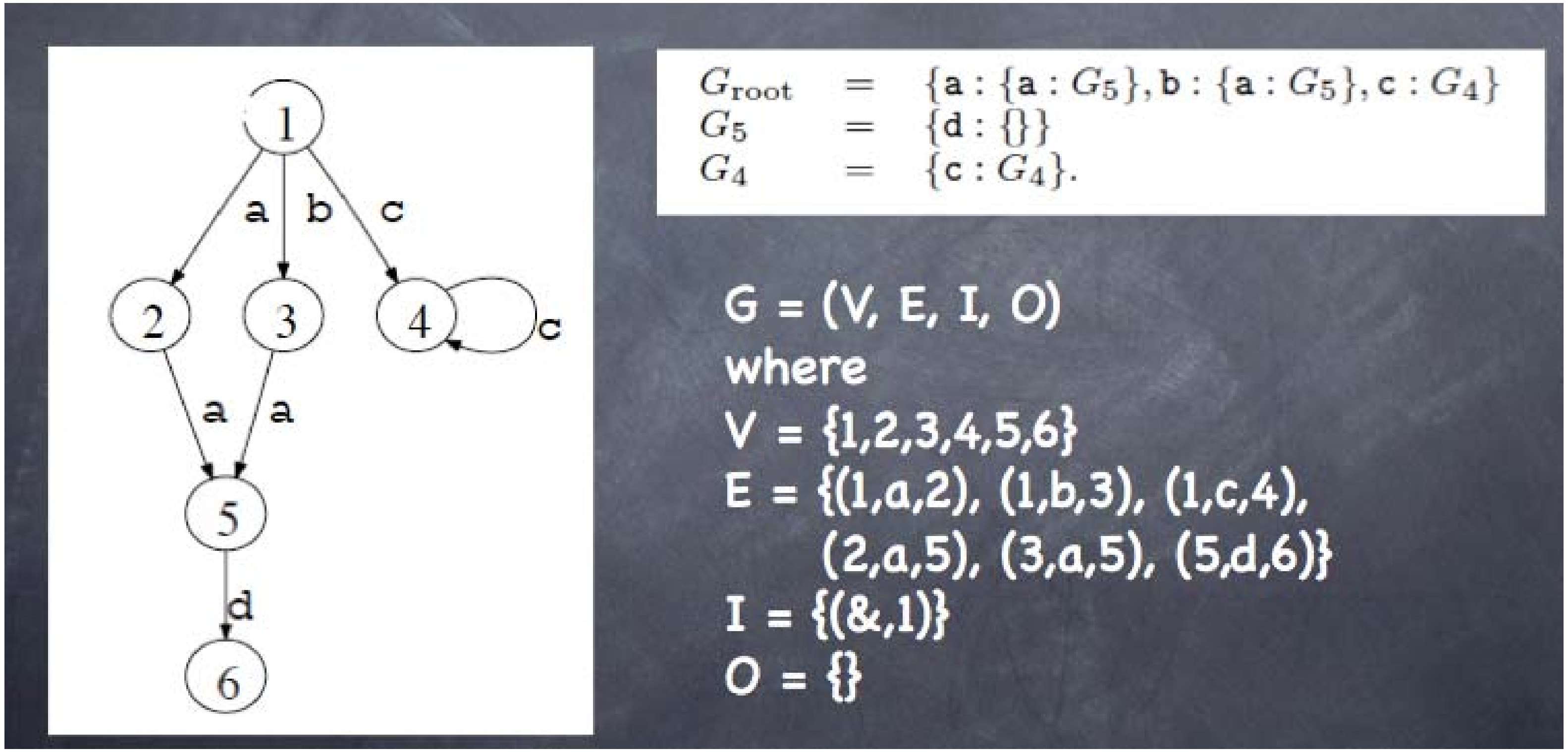
Update Checker [JSSST]

Application

[ICSE,BX]

Graph Model

- Rooted Edge-labeled Graph

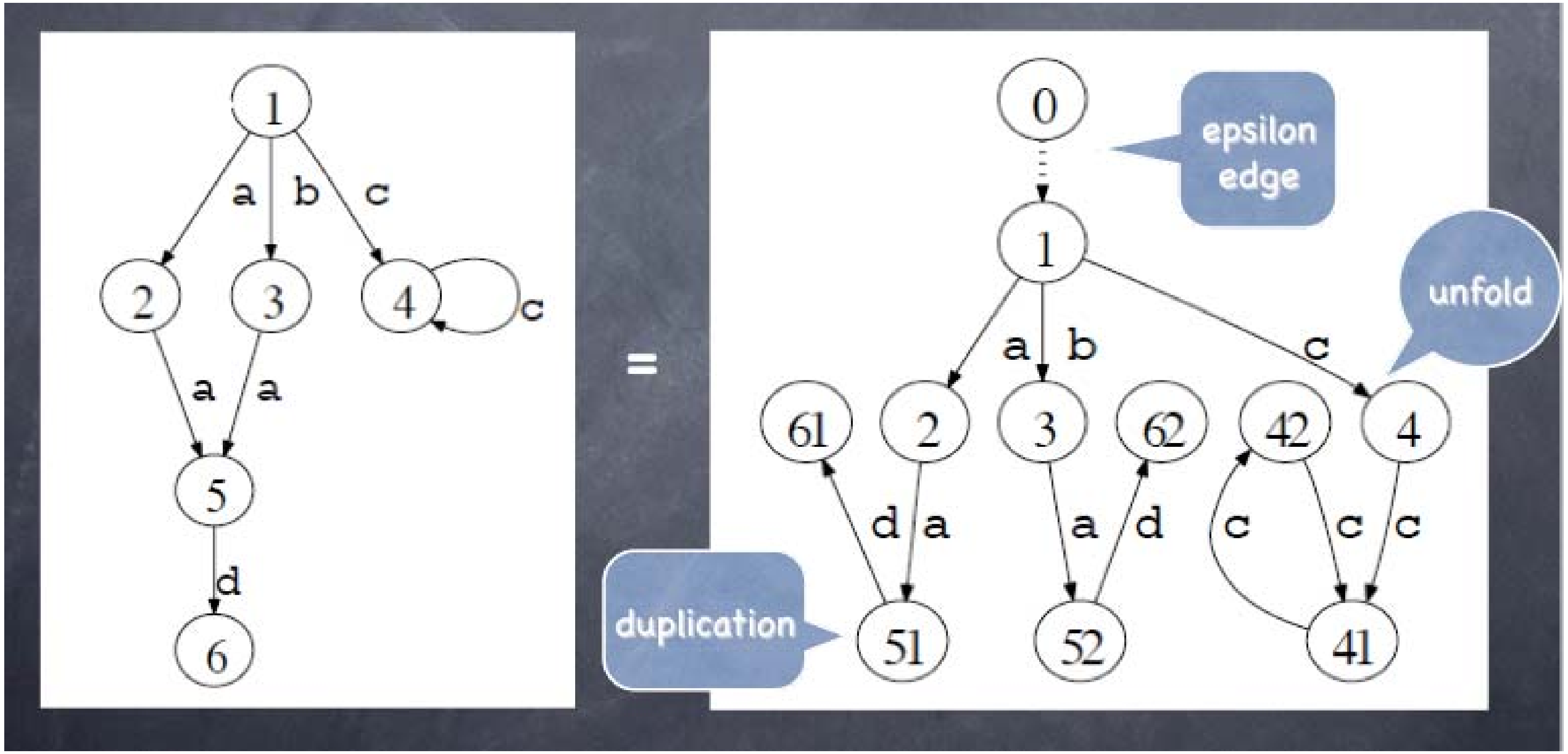


$$
\begin{aligned}
G_{\text{root}} &= \{\mathtt{a}:\{\mathtt{a}:G_5\},\mathtt{b}:\{\mathtt{a}:G_5\},\mathtt{c}:G_4\}\\
G_5 &= \{\mathtt{d}:\{\}\}\\
G_4 &= \{\mathtt{c}:G_4\}.
\end{aligned}
$$

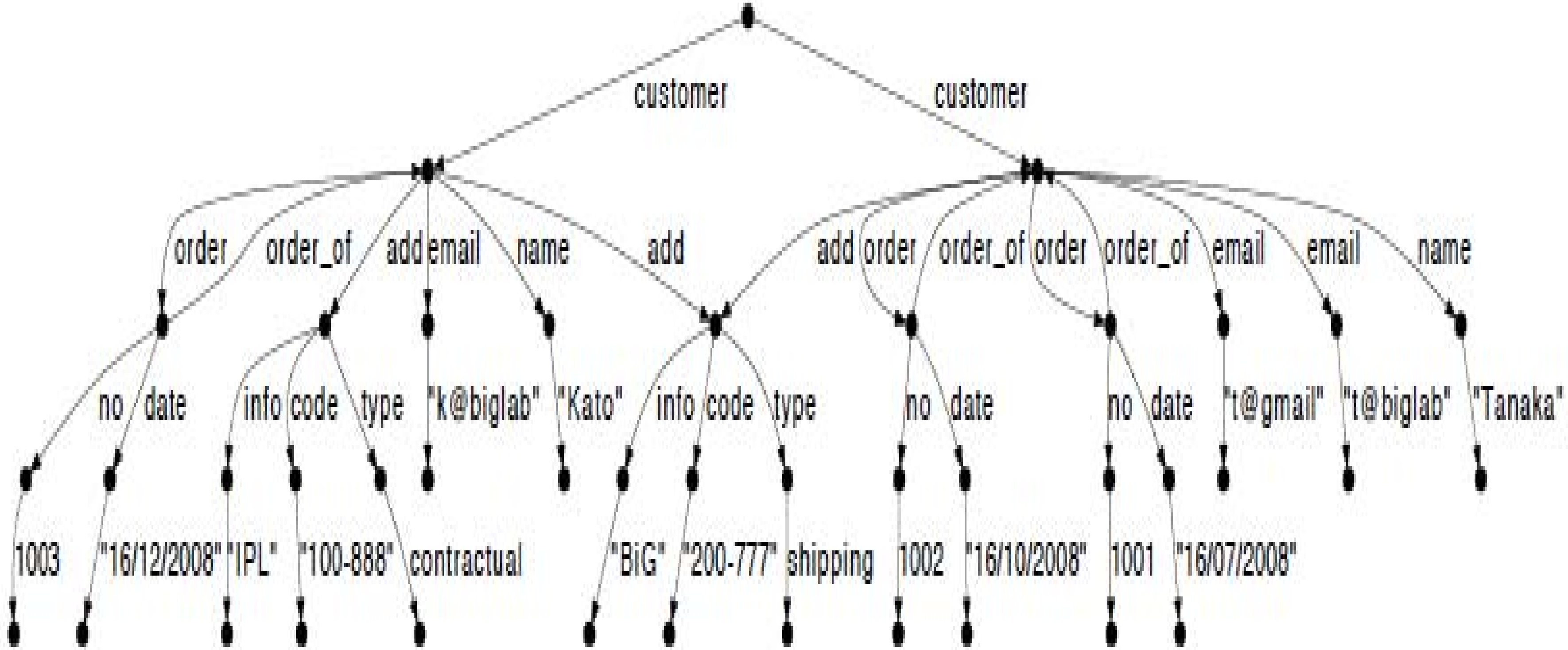
G = (V, E, I, O)
where
V = {1,2,3,4,5,6}
E = {(1,a,2), (1,b,3), (1,c,4),
(2,a,5), (3,a,5), (5,d,6)}
I = {(&,1)}
O = {}

Structuring Graphs as Regular Trees

- Graph Equivalence based on Bisimulation



Example: A Customer-Order Graph

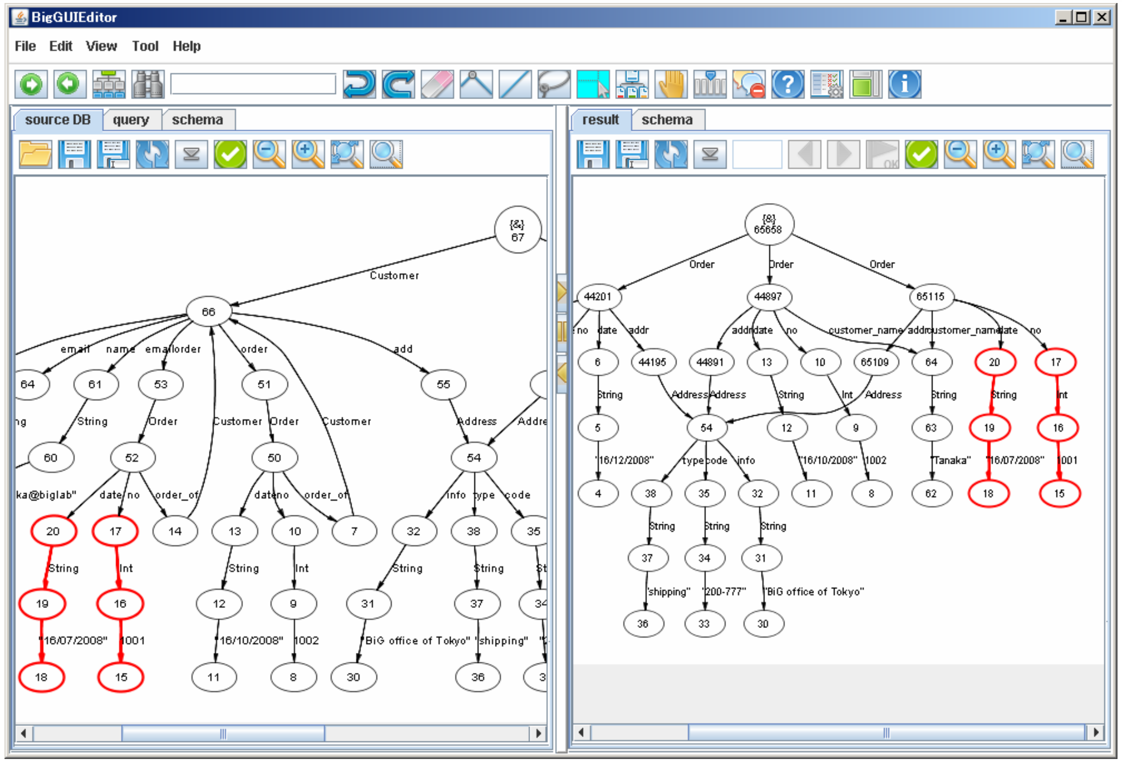


Compositional User Language

- SQL-like graph query language

```
select {tables : $table} where
  $persistentClass in
    (* select classes *)
    (select $class where
      {Association.(src|dest).Class : $class} in $db,
      {is_persistent : {Boolean : true}} in $class),
  $table in
    (* replace Attribute *)
    (replace attrs -> $g
      by (select {Column : $a} where
        {attrs.Attribute : $a} in $persistentClass)
      in $persistentClass)
```

GUI of the GRoundTRam Implementation



- Fwd/bwd transformation by 1 click
- Invalid modification on the target is reported.
- Corresponding source and target nodes are highlighted

GRoundTram

Applications

Application to Software Engineering

Co-evolution of models and codes

Background：Codes generated from models are often modified by the programmers. Models are also updated.

⇩

Problem：**<u>Changes by the programmar are lost</u>** when the codes are regenerated from the updated model ⚠

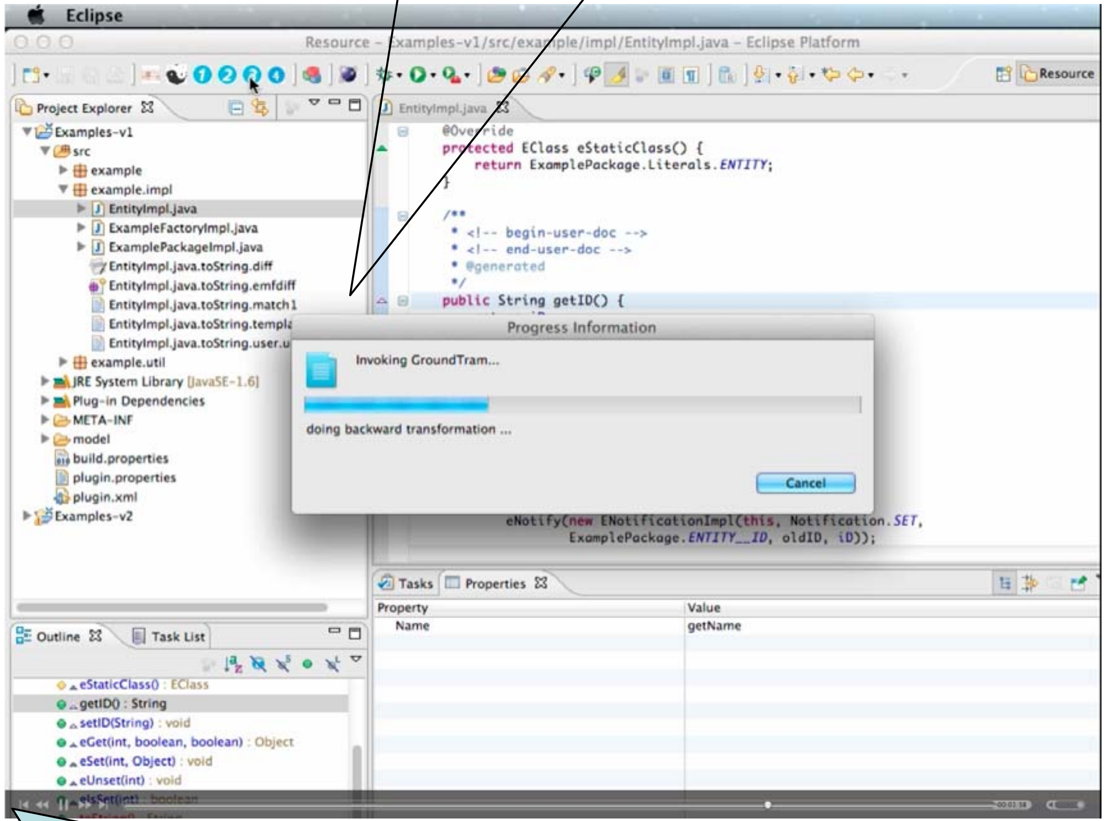
⇩

Proposal：’Undo’ of the programmer’s change is represented by the forward transformation of BX. Model updates are propagated using backward transformation.

⇩

Contribution：<u>Consistent evolution</u> of models and codes ✓

“Invoking GrondTram...
doing backward transformation ...”



Resource – Examples-v1/src/example/impl/EntityImpl.java – Eclipse Platform

```
    @Override
    protected EClass eStaticClass() {
        return ExamplePackage.Literals.ENTITY;
    }

    /**
     * <!-- begin-user-doc -->
     * <!-- end-user-doc -->
     * @generated
     */
    public String getID() {
```

```
            eNotify(new ENotificationImpl(this, Notification.SET,
                    ExamplePackage.ENTITY__ID, oldID, iD));
```

Progress Information

Invoking GroundTram...

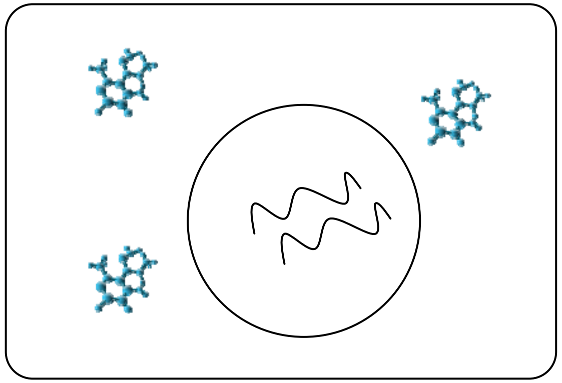
doing backward transformation ...

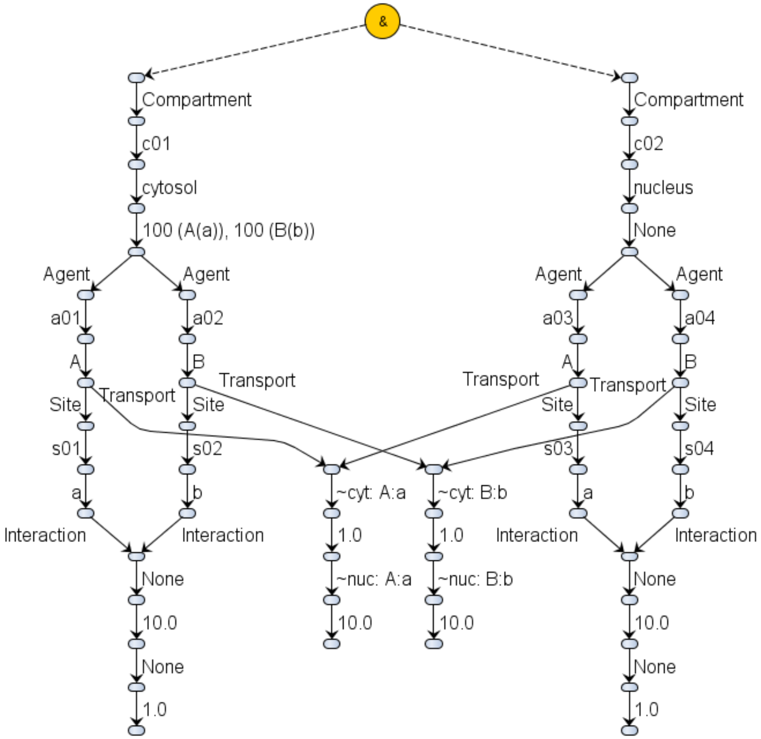
Property	Value
Name	getName

Demo at ICSE’12

Integrated into popular development environment

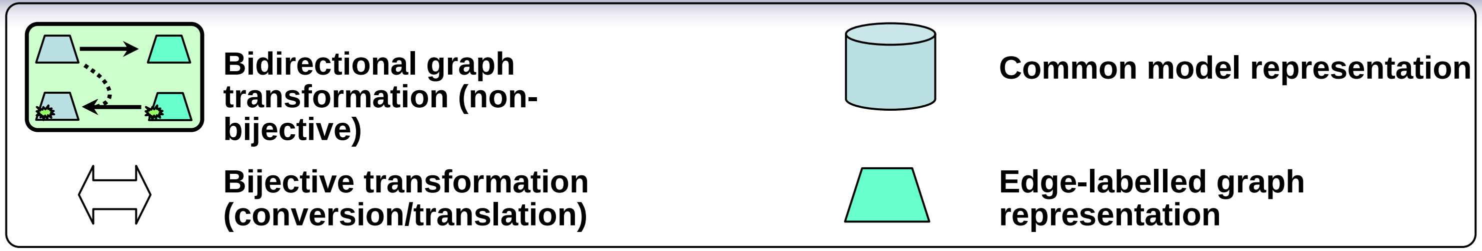
Collaborative development of Bio-models





J. R. Wilson-Kanamori, S. Hidaka, A Bidirectional Collaboration Framework for Bio-Model Development, 2nd International Workshop on Bidirectional Transformations, Mar 2013

Collaborative development of Bio-models

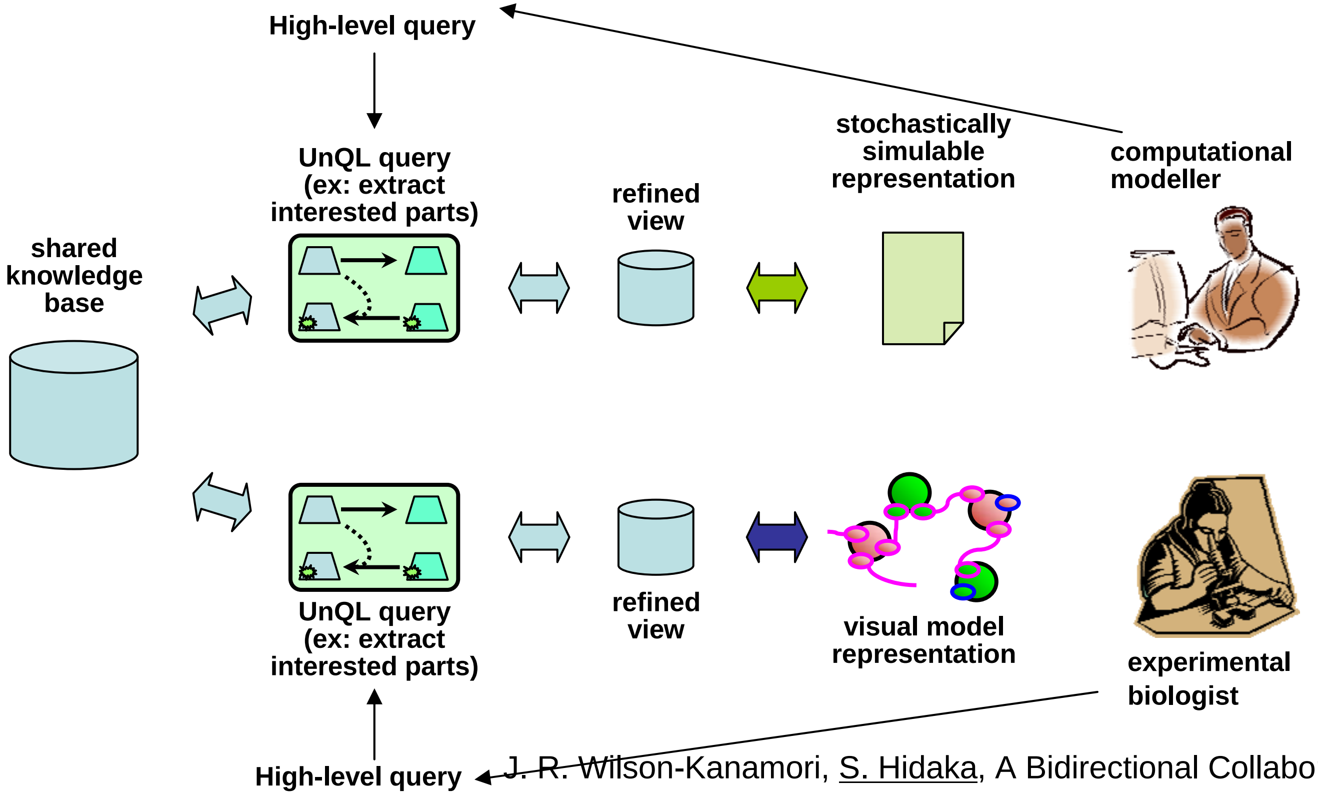


Bidirectional graph transformation (non-bijective)

Common model representation

Bijective transformation (conversion/translation)

Edge-labelled graph representation



High-level query

UnQL query (ex: extract interested parts)

refined view

stochastically simulable representation

computational modeller

shared knowledge base

UnQL query (ex: extract interested parts)

refined view

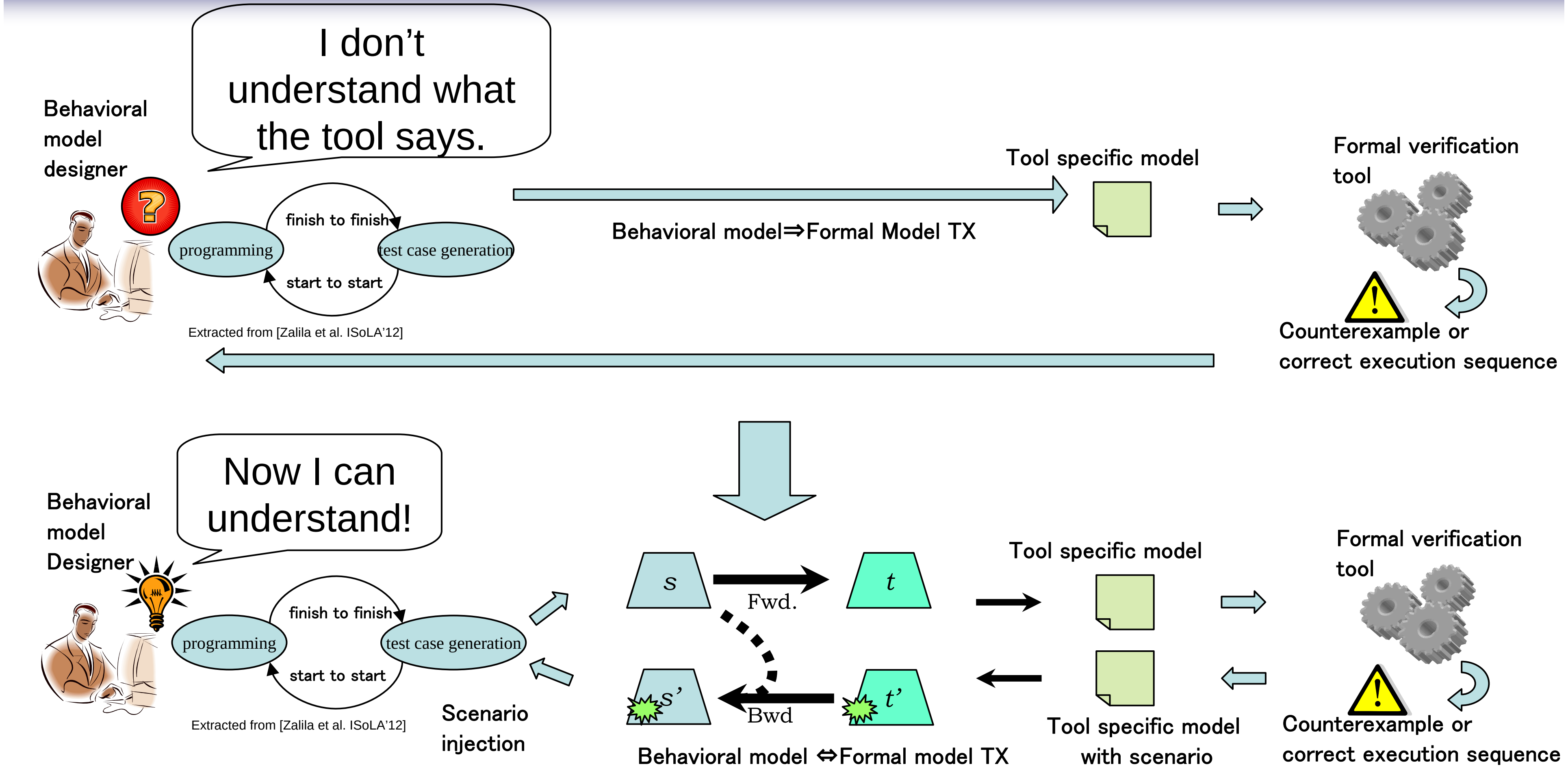
visual model representation

experimental biologist

High-level query

J. R. Wilson-Kanamori, S. Hidaka, A Bidirectional Collaboration Framework for Bio-Model Development,
2nd International Workshop on Bidirectional Transformations, Mar 2013

Feedback of verification results to designed activity



Behavioral model designer

I don't understand what the tool says.

programming

finish to finish

test case generation

start to start

Extracted from [Zalila et al. ISoLA'12]

Behavioral model⇒Formal Model TX

Tool specific model

Formal verification tool

Counterexample or correct execution sequence

Behavioral model Designer

Now I can understand!

programming

finish to finish

test case generation

start to start

Extracted from [Zalila et al. ISoLA'12]

Scenario injection

s

t

Fwd.

s'

t'

Bwd

Behavioral model ⇔Formal model TX

Tool specific model

Tool specific model with scenario

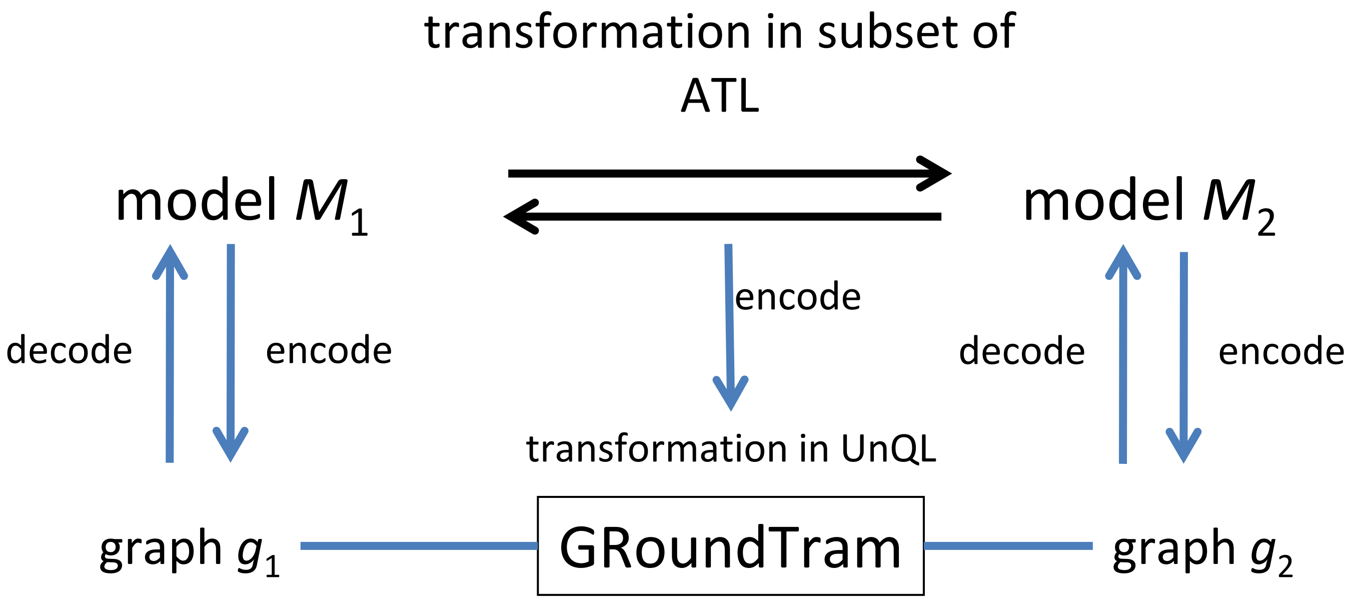
Formal verification tool

Counterexample or correct execution sequence

F. Zalila, S. Hidaka, Facilitating Verification Results Feedback on DSM Verification Context Using Bidirectional Model Transformation, submitted for publication, May 2013

Integration with Unidirectional Transformation

- BXing de-facto model transformation language ATL (INRIA, Shibaura Institute of Technology/NII)



I. Sasano, Z. Hu, S. Hidaka, K. Inaba, H. Kato, K. Nakano, Toward bidirectionalization of ATL with GRoundTram, Proc. of the 4th International Conference on Model Transformation (ICMT 2011) LNCS 6707 pp.138-151 Jun 2011

Cf. BiQuery/iGRT by Zan Tao et al.

Conclusion

- *First* framework of compositional and well-bahaved BX of *graphs*
  - Can be considered as infrastructure of adaptation by propagating changes forward and backaward
- Implemented *BX platform for graphs* GRoundTram
  - Used by research groups beside us
- *Applied to model-code co-evolution*, *synthetic biology*, and other research projects

Thank you very much for your kind attentions.

More information can be found at

http://research.nii.ac.jp/~hidaka/ and

http://www.biglab.org/

Appendix

Publications - *Foundations*

[ICFP'10] S. Hidaka, Z. Hu, K. Inaba, H. Kato, K. Matsuda, K. Nakano, Bidirectionalizing Graph Transformations, 15th ACM SIGPLAN International Conference on Functional Programming, pp.205-216 Sep 2010

[PPDP'11] K. Inaba, S. Hidaka, Z. Hu, H. Kato, K. Nakano, Graph-Transformation Verification using Monadic Second-Order Logic, 13th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming, pp.17-28 Jul 2011

[LOPSTR] S. Hidaka, Z. Hu, K. Inaba, H. Kato, K. Matsuda, K. Nakano and I. Sasano, Marker-directed optimization of UnCAL graph transformations, 21st International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR 2011) LNCS vol. 7225, pp.123-138 Jul 2011

[ICFP'13] S. Hidaka, K. Asada, Z. Hu, H. Kato, K. Nakano, Structural Recursion for Querying Ordered Graphs, 18th ACM SIGPLAN International Conference on Functional Programming, to appear, Sep 2013

[PPDP'13] K. Asada, S. Hidaka, H. Kato, Z. Hu, K. Nakano, A Parameterized Graph Transformation Calculus for Finite Graphs with Monadic Branches,15th International Symposium on Principles and Practice of Declarative Programming, to appear, Sep 2013

[JSSST.J] K. Nakano, S. Hidaka, Z. Hu, K. Inaba, H. Kato, View Updatability Checking with Simulation-based Graph Schema,, JSSST Computer Software 29(2) pp.174-192 Apr 2012

Publications - *Framework*

[PI] S. Hidaka, Z. Hu, K. Inaba, H. Kato, K. Nakano: GRoundTram: An Integrated Framework for Developing Well-Behaved Bidirectional Model Transformations, Progress in Informatics, No. 10, Apr 2013

[ASE] S. Hidaka, Z. Hu, K. Inaba, H. Kato and K. Nakano, GRoundTram: An Integrated Framework for Developing Well-Behaved Bidirectional Model Transformations (short paper), 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011) pp.480-483 Nov 2011

[ICSE NIER] S. Hidaka, Z. Hu, H. Kato, K. Nakano, A compositional approach to bidirectional model transformation, 31st International Conference on Software Engineering Companion Volume pp.235-238 May 2009

[SAC] S. Hidaka, Z. Hu, H. Kato, K. Nakano, Towards a Compositional Approach to Model Transformation for Software Development, ACM symposium on Applied Computing pp.468-475 Mar 2009

Publications - *Applications*

[ICSE] Y. Yu, Y. Lin, Z. Hu, S. Hidaka, H. Kato, L. Montrieux, Maintaining Invariant Traceability through Bidirectional Transformations, 34th International Conference on Software Engineering (ICSE 2012) pp.540-550 Jun 2012

[BX] J. Wilson-Kanamori and S. Hidaka, A Bidirectional Collaboration Framework for Bio-Model Development, Second International Workshop on Bidirectional Transformations (BX 2013), Mar 2013, Rome, Italy, colocated with ETAPS 2013

[ICMT] I. Sasano, Z. Hu, S. Hidaka, K. Inaba, H. Kato, K. Nakano, Toward bidirectionalization of ATL with GRoundTram, Proc. of the 4th International Conference on Model Transformation (ICMT 2011) LNCS 6707 pp.138-151 Jun 2011