

Model (Graph) transformation using UnQL⁺

Hiroyuki Kato

National Institute of Informatics

Grace Meeting
December 15, 2008

Hello, my name is ...

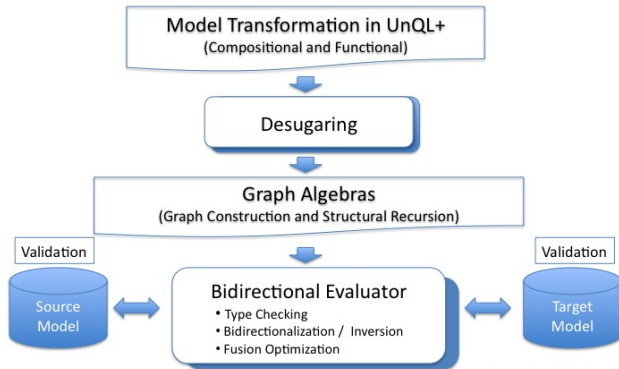
- Hiroyuki Kato from National Institute of Informatics, Japan.
- I'm a member of BiG Project led by Prof. Zhenjiang Hu.

Hello, my name is ...

- Hiroyuki Kato from National Institute of Informatics, Japan.
- I'm a member of BiG Project led by Prof. Zhenjiang Hu.

BiG

A Functional Approach to Bidirectional Model Transformation
S. Hidaka, Z. Hu, H. Kato, S.C. Mu, K. Nakano



A Grand Challenge Project at National Institute of Informatics in Japan



国立情報学研究所
National Institute of Informatics

My background:Databases

DBPL like XQuery and UnQL.

My background:Databases

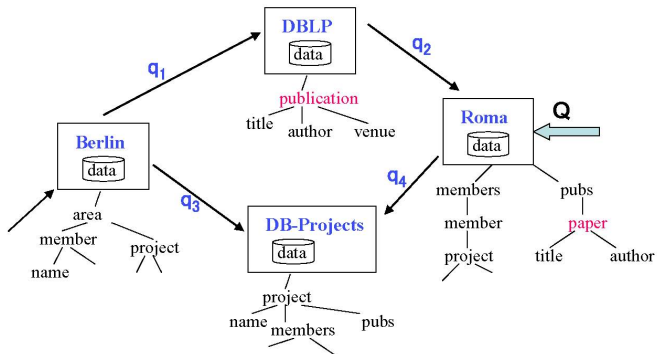
DBPL like XQuery and UnQL.

XQuery fusion(PLAN-X 2009)

My background: Databases

DBPL like XQuery and UnQL.

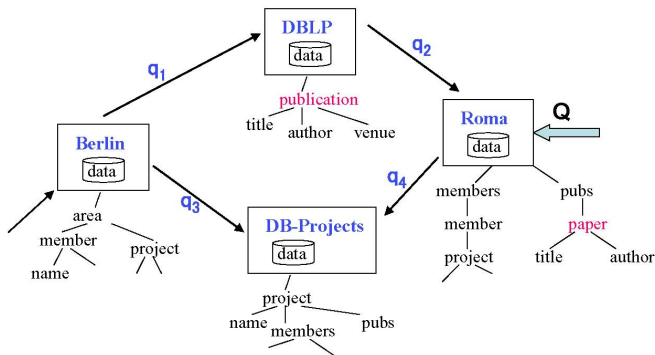
XQuery fusion(PLAN-X 2009)



My background: Databases

DBPL like XQuery and UnQL.

XQuery fusion(PLAN-X 2009)

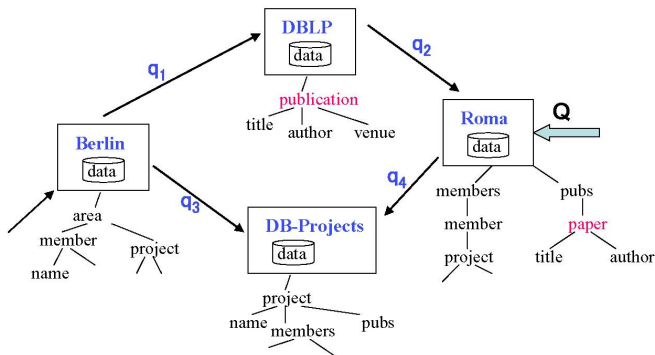


- Result of $Q := Q(\text{Roma})$

My background:Databases

DBPL like XQuery and UnQL.

XQuery fusion(PLAN-X 2009)

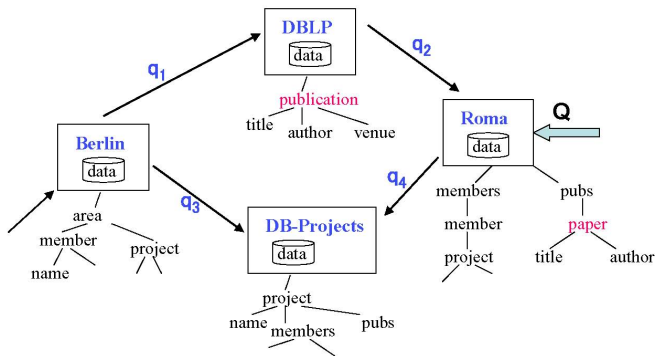


- Result of $Q := Q(\text{Roma}) \cup \underline{Q(q_2(\text{DBLP}))}$

My background:Databases

DBPL like XQuery and UnQL.

XQuery fusion(PLAN-X 2009)

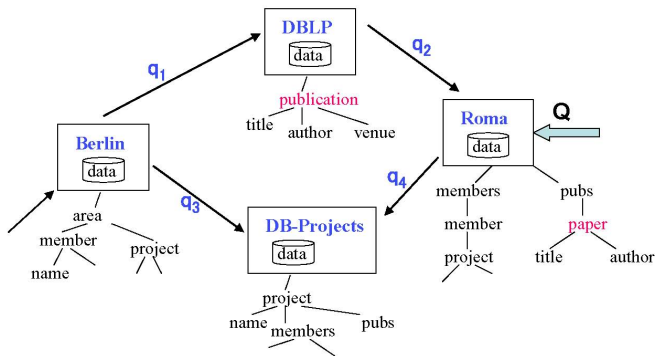


- Result of $Q := Q(\text{Roma}) \cup \underline{Q(q_2(\text{DBLP}))} \cup \dots$

My background:Databases

DBPL like XQuery and UnQL.

XQuery fusion(PLAN-X 2009)

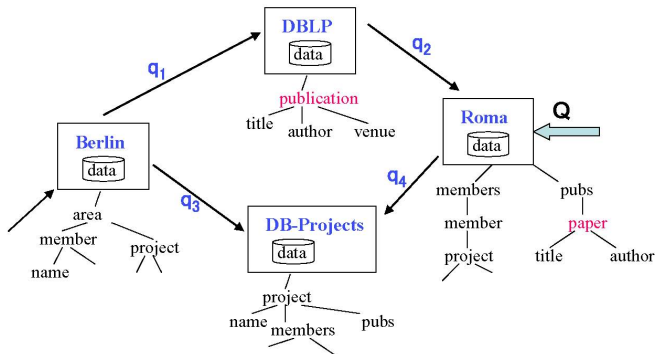


- Result of $Q := Q(\text{Roma}) \cup Q(q_2(\text{DBLP})) \cup \dots$
- $Q(q_2(\text{DBLP})) = Q \circ q_2(\text{DBLP})$

My background:Databases

DBPL like XQuery and UnQL.

XQuery fusion(PLAN-X 2009)



- Result of $Q := Q(\text{Roma}) \cup Q(q_2(\text{DBLP})) \cup \dots$
- $Q(q_2(\text{DBLP})) = Q \circ q_2(\text{DBLP}) = \text{XQueryFusion}(Q \circ q_2)(\text{DBLP})$

My research are NOT directly related to b.t.. but provide some basis for b.t..

- XQuery fusion.
- Model (Graph) transformation.
A model transformation \implies A graph transformation,
since models are basically graphs.

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: `select ... where ...`

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: **select ... where ...**
- Use **structural recursion** for manipulating graphs

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: **select ... where ...**
- Use **structural recursion** for manipulating graphs

Extending UnQL to UnQL⁺ to with three useful editing operations for graph transformation,

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: **select ... where ...**
- Use **structural recursion** for manipulating graphs

Extending UnQL to UnQL⁺ to with three useful editing operations
for graph transformation,
delete ... where ... for deletion of subgraph,

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: `select ... where ...`
- Use `structural recursion` for manipulating graphs

Extending UnQL to UnQL⁺ to with three useful editing operations for graph transformation,

`delete ... where ...` for deletion of subgraph,
`extend ... with ... where ...` for extension of input graph, and

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: **select ... where ...**
- Use **structural recursion** for manipulating graphs

Extending UnQL to UnQL⁺ to with three useful editing operations for graph transformation,

delete ... where ...	for deletion of subgraph,
extend ... with ... where ...	for extension of input graph, and
replace ... by ... where ...	for replacement of subgraph.

Graph transformation

A graph query language UnQL[Buneman et al. 2000] is a
compositional framework for graph querying

- Easy to use: **select ... where ...**
- Use **structural recursion** for manipulating graphs

Extending UnQL to UnQL⁺ to with three useful editing operations for graph transformation,

delete ... where ...	for deletion of subgraph,
extend ... with ... where ...	for extension of input graph, and
replace ... by ... where ...	for replacement of subgraph.

These can be translated into **structural recursion**.

Structural Recursion: Manipulating Graphs

Structural Recursion:

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

Structural Recursion: Manipulating Graphs

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:

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

What I can contribute to this workshop is ...

- For Model (Graph) transformation using UnQL⁺,
A short introduction to UnQL⁺.
- For XQuery fusion,
A demonstration of reformulation of XQuery expressions.