Provenance in ORCHESTRA

T.J. Green, G. Karvounarakis, Z. Ives, V. Tannen
University of Pennsylvania

Principles of Provenance (PrOPr)
Philadelphia, PA       June 26, 2007
Collaborative Data Sharing [Ives+ CIDR 05]

Schema mappings specify how data is logically related

Update exchange propagates updates and records provenance information

(1) to assess trust conditions
(2) to facilitate incremental maintenance
Deletions and provenance

+G(1,2,3) \( p_4 \)

\( G \)

(1,2,3)

(3,5,2)

+G(3,5,2) \( p_3 \)

\( B \)

(1,3)

(3,2)

(3,3)

m_1

m_2

m_3

m_4

G

m_1

B

U

(3,2)

(3,3)

(3,c_3)

(2,c_2)

(5,c_1)

(2,5)

+U(2,5) \( p_2 \)

+B(3,5) \( p_1 \)
Peer B distrusts any tuple B(i,n), if the data came from Peer G and mapping and it trusts any tuple from Peer U.
Further aspects of ORCHESTRA

- Semantics: insertions and deletions with idbs
- Handling conflicts among trusted updates (Taylor+Ives SIGMOD 06)
- Prototype implementation (demo SIGMOD 07, technical details VLDB 07):
  - Java middleware layer using database as subcomponent
  - Provenance expressions stored as tables
  - tgds become datalog rules with Skolem functions
  - Update exchange using relational query engine (recursion!)
  - Feasibility experiments
- Future work / topics for discussion
  - What else to do with rich provenance information? (ranked trust models, bag semantics, querying provenance, ...)