Integrating Theory and Practice for Scientific Data Sharing

Zachary G. Ives and Todd J. Green

Work done in collaboration with N. Taylor, G. Karvounarakis, V. Tannen

University of Pennsylvania

Funded by NSF IIS-0477972, 0513778, 0629846
http://www.cis.upenn.edu/~zives/orchestra/

InfInt Workshop

2 October 2007

We Haven't Built a Channel from Theory ➔ Practice in Academia

Proposition 1

Conjecture 1

Theory gets mapped to practice if(f):

1. It provides a **solution** to problems difficult to solve using alternative strategies
2. A **system** implements the theory in a non-commercial setting first
3. The system is applied and **evaluated** in real(istic) settings

Let's Consider Sharing among Large Numbers of Scientific DBs

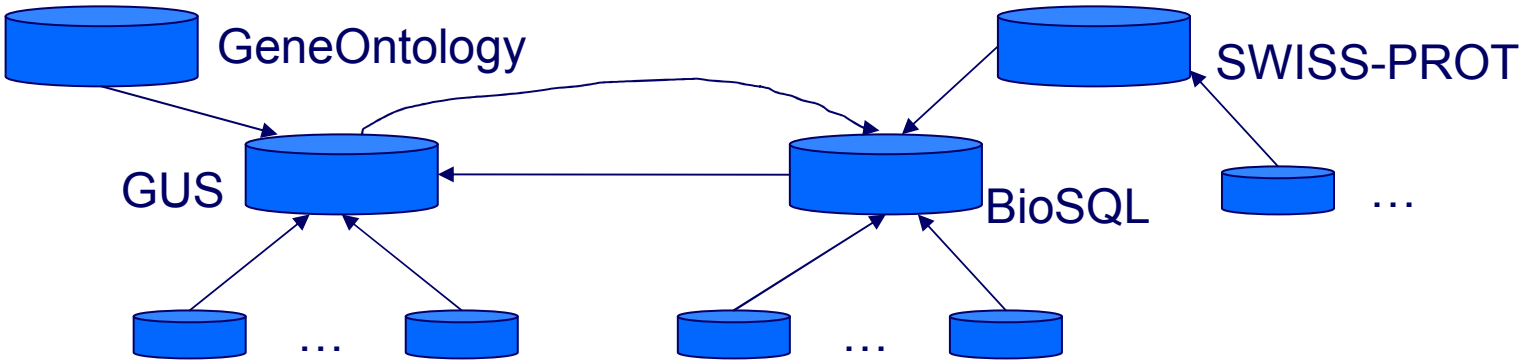
Scientific data sharing is a great domain for data integration research:

- Data and schemas publicly available – **much to share**
- Great **diversity**, overlap
- Many efforts to bring the data together
- Huge potential (scientific) payoff, eager users
  - Often best-effort is sufficient – much exploratory work

➢It provides an eager user community, real problems that we can get access to. It also gives lessons that transfer to other areas.

- But we currently do a very poor job of supporting it!

Scientific Data Sharing Today (Life Sciences, e.g., Genomics)



- Many labs produce raw data and analyze / curate it
- Large community / organization warehouses (e.g., SWISS-PROT, GenBank, FlyBase, …) **standardize some data**
- But there's a lot of overlapping and complementary info among warehouses – sites **import others' data**
- Almost all based on Perl, Python scripts as "mappings"!

➢ Why aren't they using declarative mappings and tools?

Many Unfulfilled Scientific Data Sharing Desiderata

Autonomy and **local control** of data are critical

- Control which data is imported / integrated
- Modify any data, even data from elsewhere!

Many different **points of view** !!!

- Disagreements about data, mappings, schemas...
- Which sources are trusted / distrusted

Data **provenance** tracking

Support frequent **updates**, to data, schemas, mappings

➢ Data integration & exchange don't provide a unified solution

Conventional Data Integration Architectures Don't Work Here

The traditional virtual integration model is too controlled:

- Global standardization, strictly hierarchical

  Doesn't scale, especially to diverse user bases or evolving domains

- Assumes globally consistent view of data is possible and desirable

Data exchange [Fagin+03] is one-time only

Peer data management (e.g., Hyperion, Piazza, [Bernstein+02], [Calvanese+04], peer data exchange, …) helps somewhat:

- Supports many schemas for many different needs
- Can incrementally add schemas, mappings
- But what about changes to the data? Disagreements among sites? Curation of imported data? Control over what data is imported?

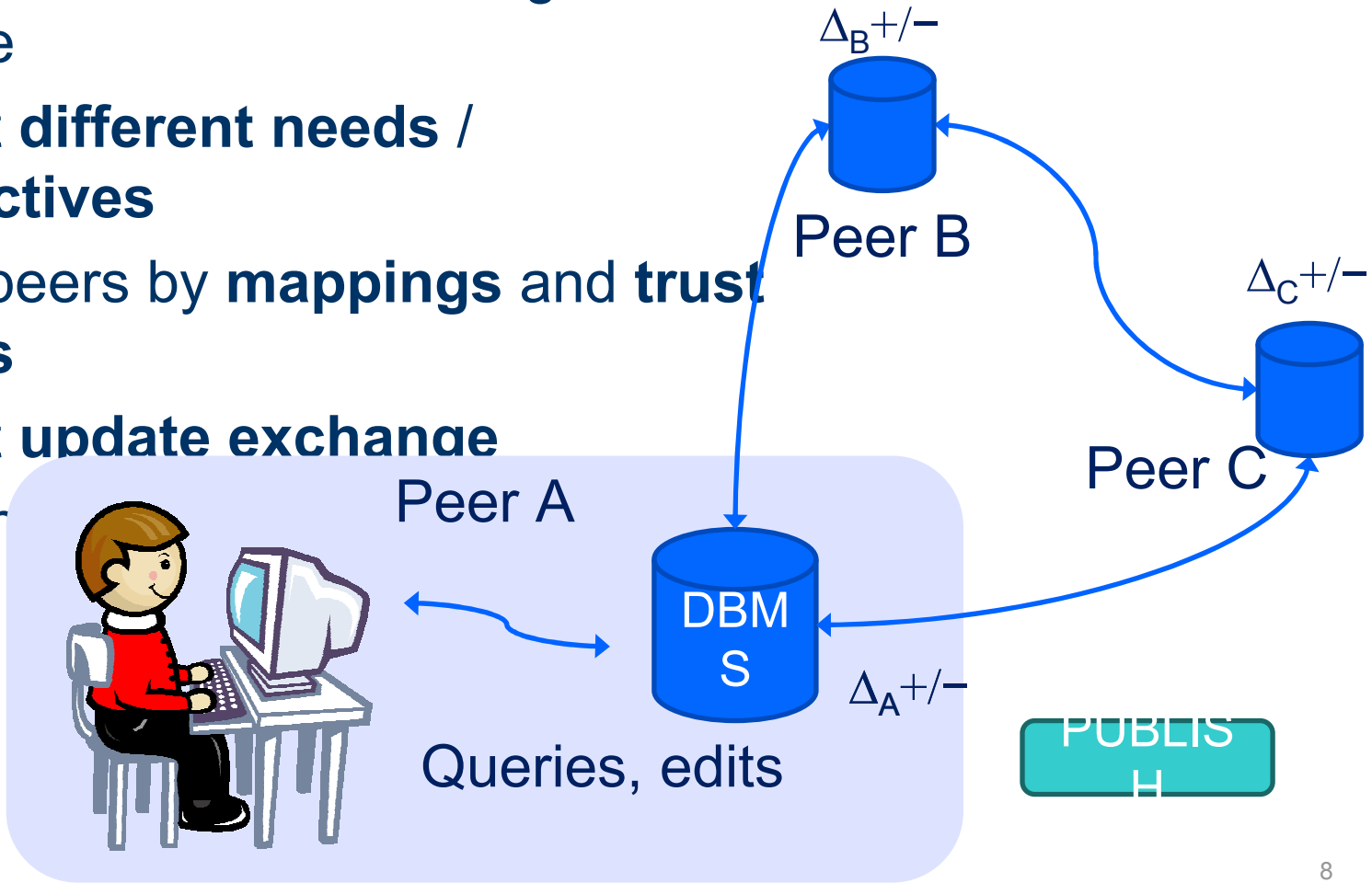
Much as semi-structured data
made it easier to handle occasionally irregular
information, we need a more
flexible, **dynamic, incremental** scheme for
integrating / sharing structured data!

… we have been defining its theory and
implementation, in a system called
ORCHESTRA…

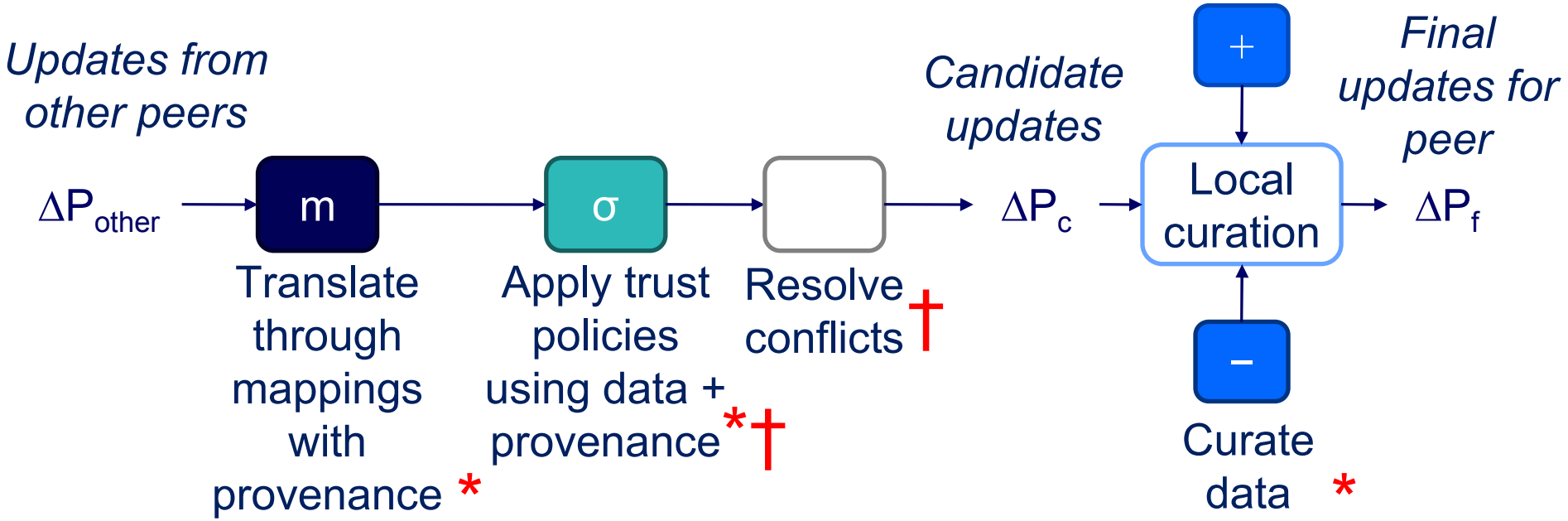
The Collaborative Data Sharing System (CDSS): Loosely Coupled, Highly Dynamic

[Ives+05]

- Give peers **full control** using local instance
- Support **different needs / perspectives**
- Relate peers by **mappings** and **trust policies**
- Support **update exchange**
- Maintain



The CDSS "Pipeline" (One Peer's Perspective)



* [Green+07]

† [TaylorIves06]

Technical Overview – Roadmap

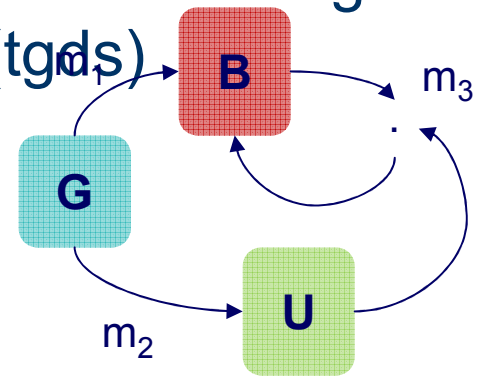
Update exchange in a CDSS:

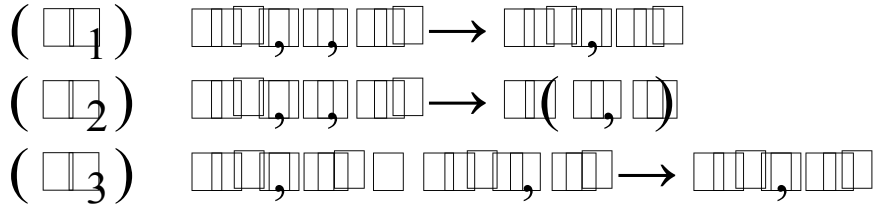
- **Schema mappings**
- Tracking of **data provenance**
- **Incremental propagation** of updates
- Provenance-based **trust policies**
- **Local curation** via insertions / deletions

Brief overview of prototype (ORCHESTRA)

Mappings and updates

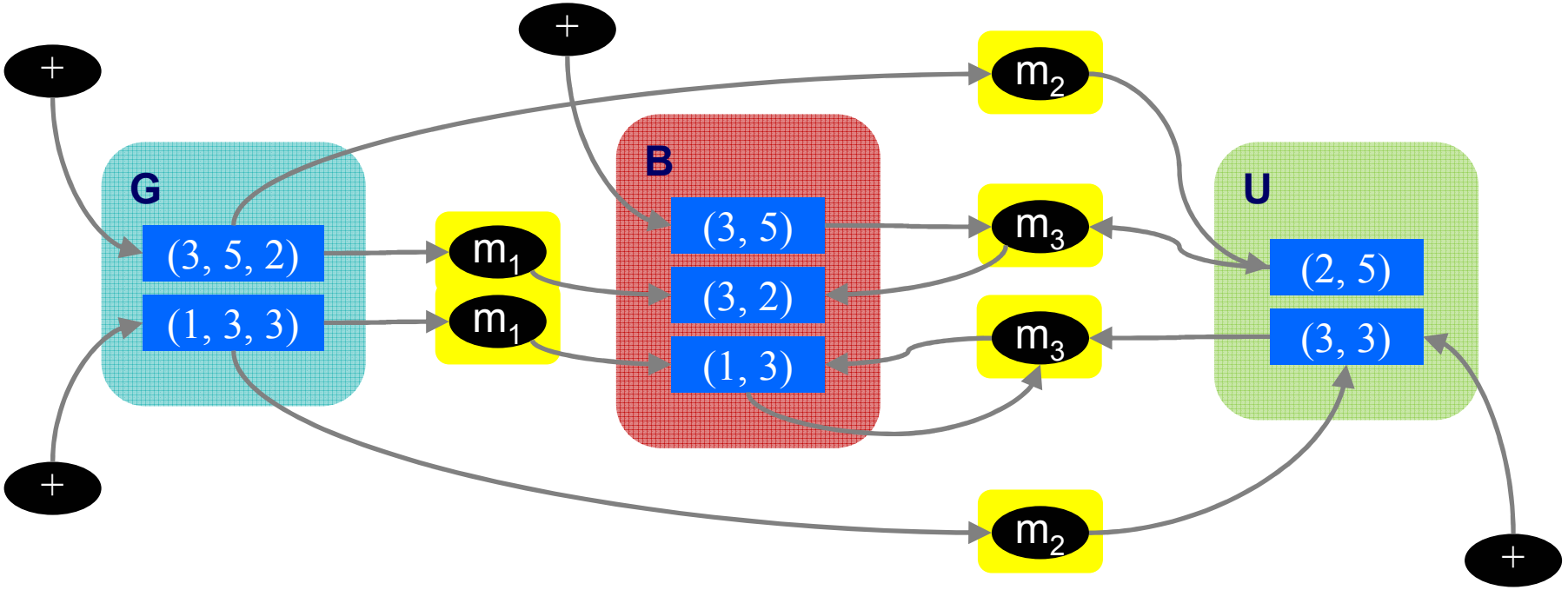
- CDSS setting: set of peers; set of declarative mappings (tgds)

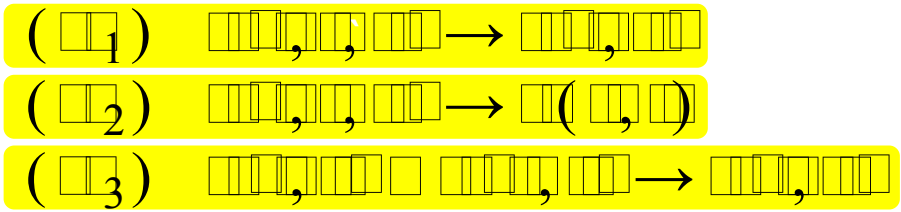




- Given: setting, base data, updates
- Goal: local instance at each peer cf. **data exchange** paradigm [Fagin+03]
  - **Universal solution** yields the "**certain answers**" to queries
  - Can be computed using the **chase**
- Our contribution: how to do it **incrementally**, with **provenance**...

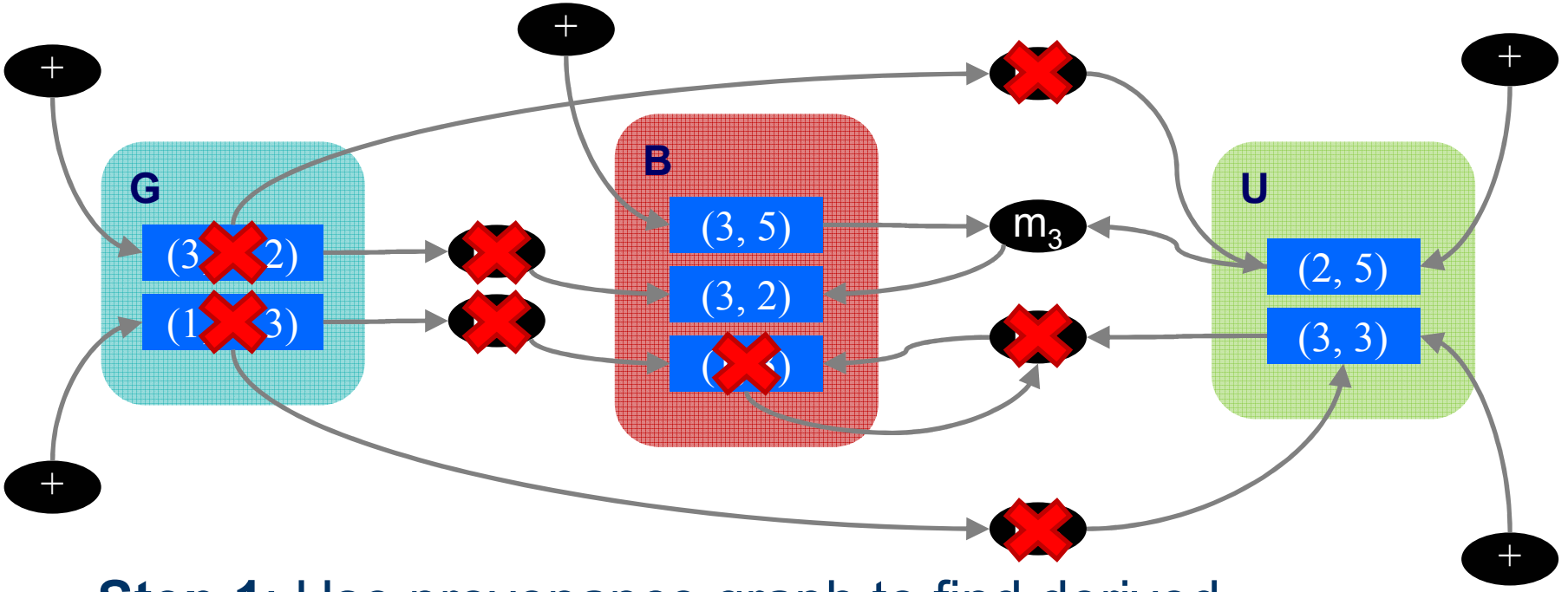
Incremental insertion





This graph represents the provenance information that ORCHESTRA maintains
(see also [Green+07])

Incremental deletion that supports recursion



- **Step 1**: Use provenance graph to find derived tuples which can also be deleted
- **Step 2**: Test other affected tuples for derivability from base insertions, and delete any not derivable
- **Step 3**: Repeat

Trust policies (not every update should be propagated)

Updates can be **filtered automatically** based on **provenance** and **content**

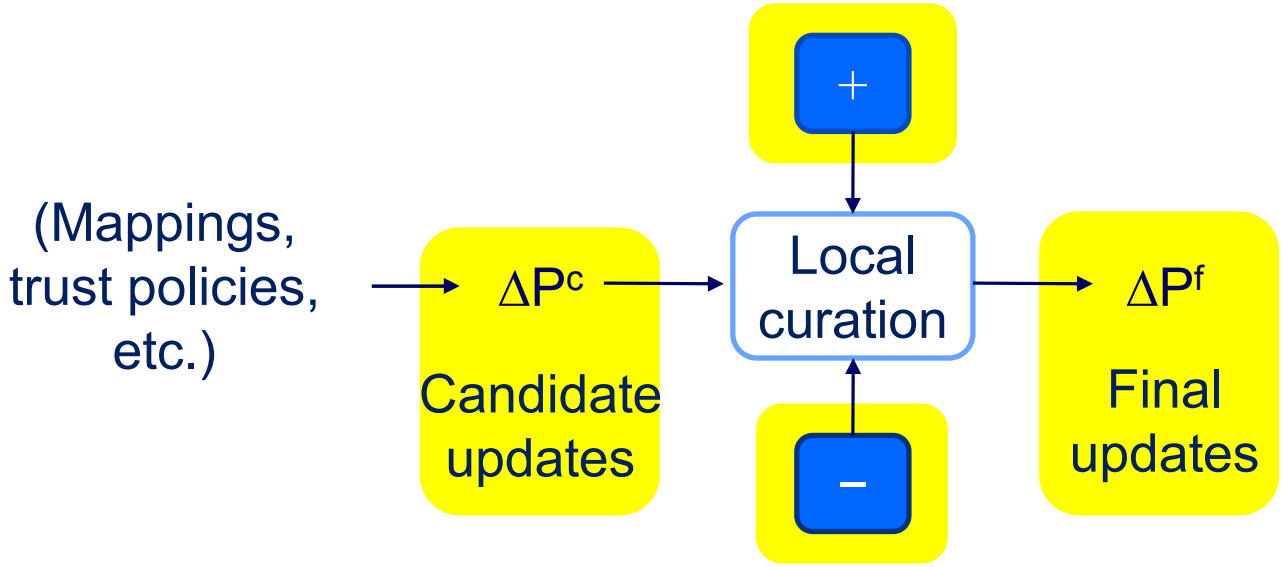
"Peer A:

- distrusts any tuple U(i,n) that came from Peer B and has value n = 3
- trusts any tuple that came from Peer C
- distrusts any tuple U(i,n) that came from mapping m4 if n = 2"

Local curation: user can **also manually** accept/reject **updates,** or introduce new ones...

Local curations

Extra tables for local **insertions** and **deletions**:



Contribution: conforms to data exchange paradigm by using **internal mappings** with local **insertions/deletions**:

$(\square_2)$ $\square_{\square}(\square, \square, \square) \rightarrow \square(\square, \square)$ $\square$

$(\square_2')$ $\square_{\square}(\square, \square, \square) \rightarrow \square(\square, \square)$

$(\square_{\square^+})$ $\square^+(\square, \square) \rightarrow \square(\square, \square)$

$(\square_{\square^-})$ $\square_{\square}(\square, \square) \wedge \neg \square^-(\square, \square) \rightarrow \square(\square, \square)$

Prototype implementation

- **Middleware** layer on top of clients' relational DBMSs
- Auxiliary tables store provenance info
- Mappings converted to **datalog rules** (as in Clio)
  - Uses either a commercial DBMS engine
    - Datalog fixpoints in Java and SQL (only linear recursion in vendor SQL)
    - Labeled nulls supported via encoding scheme; Skolem UDFs
  - Or our own engine (with custom fixpoint, labeled null types)
- 30,000 lines of Java code (growing rapidly)

  Performance validated experimentally in [Green+07]

  Demoed at SIGMOD, DILS this year

➢ Targeting open-source release in 2008

Related work

- **Peer data management systems** Piazza [Halevy+03, 04], Hyperion [Kementsietsidis+04], [Bernstein+02], [Calvanese+04], ...
- **Data exchange** [Haas+99, Miller+00, Popa+02, Fagin+03], **peer data exchange** [Fuxman+05]
- **Provenance** / **lineage** [CuiWidom01], [Buneman+01], Trio [Widom+05], Spider [ChiticariuTan06], [Green+07], ...
- **Incremental maintenance** [Blakeley+86], many others incl. commercial systems; for recursive case, [GuptaMumick95] …
- DRed ("delete and re-derive") computes **superset** of deletions, then corrects if needed
- We use provenance to compute **exact** set of deletions

The ORCHESTRA System: A Platform for Data Integration / Exchange

ORCHESTRA implements the CDSS architecture:

- Supports update exchange over tgd / GLAV mappings
  - Will also support certain-answers query answering over virtual schemas
- Tracks provenance (using *provenance semirings* [Green+07])
- Supports site-specific trust conditions, conflict resolution, local curation

Academia needs at least one "PostgreSQL of data integration"

- Postgres was very helpful in connecting research → real apps!
- We are making our prototype available by request
- Open-source release planned for 2008

With bio collaborators: build applications for phylogeny (pPOD), genomics/proteomics (SHARQ)

Discussion: Some Questions and Possible Ideas for Future Work

- Collaborative data sharing for science must support at least:
  - Loose sync of autonomous schemas, instances; possible conflicts
  - Mappings (schema, entity, …)
  - Trust / authority, related to provenance
  - Local curation (and cleansing)
- … Are there other aspects? Does business need these capabilities, or is science different? Other apps?
- Data cleaning vs. conflicts vs. trust?
- Trust aggregation: ideas from social networks, PageRank, …
- Probabilistic / ranked data or mappings? Keywords?
- Data exploration?