Managing the When-provenance of Data: Opportunities and Challenges

Wang-Chiew Tan
University of California, Santa Cruz

(Mainly based on work with Mary Roth, CIDR 2013)

Data provenance

- Provenance: [Merriam-Webster online dictionary, cited Apr 1, 2013]

  1. origin, source.

Data provenance

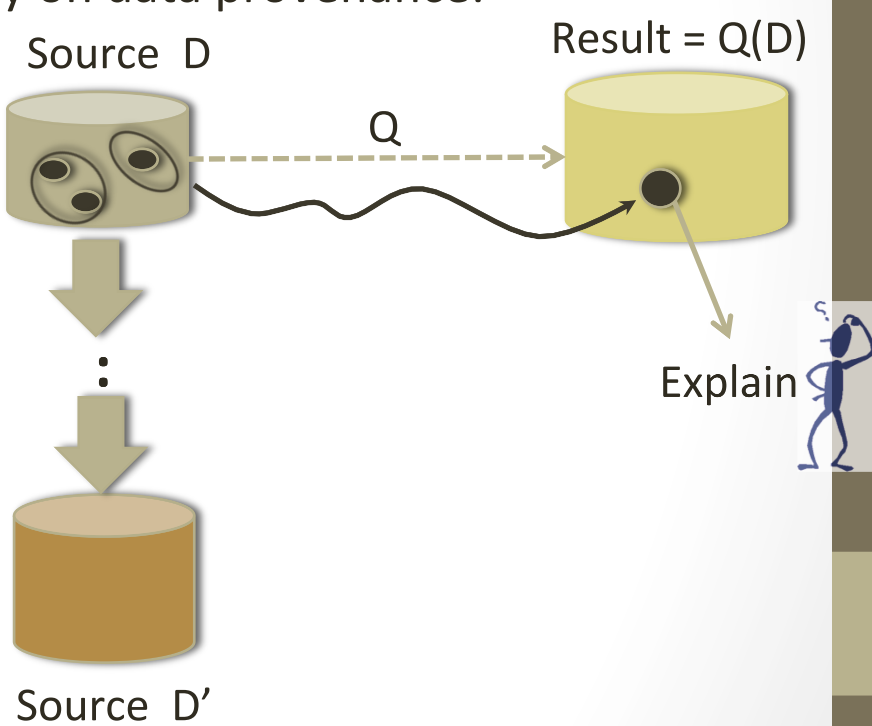
- Provenance: [Merriam-Webster online dictionary, cited Apr 1, 2013]

  1. origin, source.

  2. ***the history of ownership of a valued object or work of art or literature.***

Past work by the database community on data provenance:

- Lineage [Cui,Widom,Wiener 00]
- Why and where-provenance [Buneman,Khanna,T. 01,02]
- Provenance semirings [Green,Karvournarakis,Tannen 07]
  - aka how-provenance
- Causes and degree of responsibility [Meliou *et al*. 09,10]

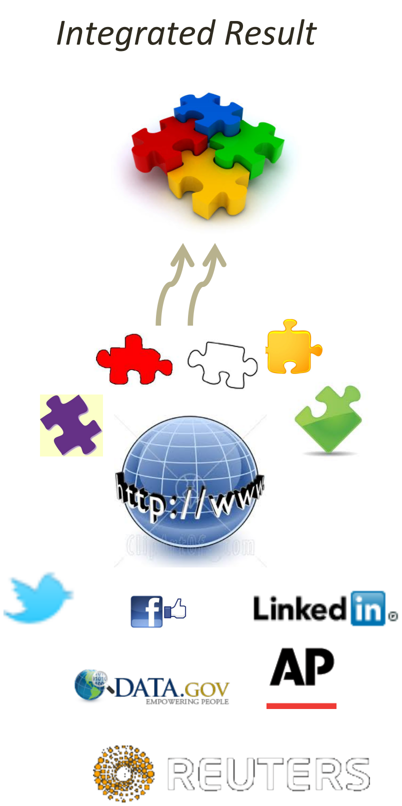


Keep all versions, keep all changes – is this it?

- Can we easily answer questions such as:
  - How has the Jane's salary changed over the decade 2000-2010?
  - Did Jane work in the same company as John and when?
  - Compute the average number of days Jane spends in Chicago per year.

- Difficult in general.
  - Need to reconcile different data sources, imprecise and conflicting information across time from different evolving data sources.

The Opportunity:
Create a Whole Greater than the Sum of its Parts



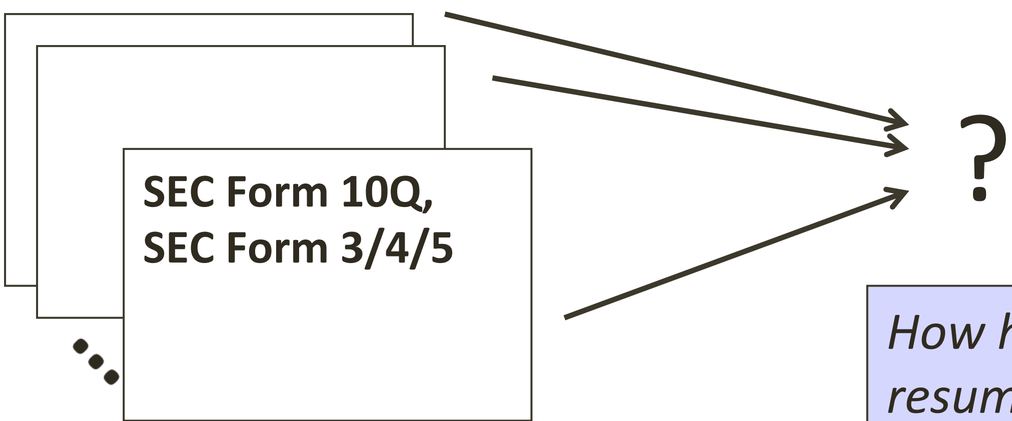
- Electronically available data is growing at a record pace
  - Enterprise (personnel records, business transactions)
  - Public (web sites, blogs, tweets)
  - Required by regulation (financial filings, real estate transactions, …)

"When-provenance"

- It is possible to build and maintain a ***historical account*** of just about anything and everything
  - People: corporate officers, public officials, job applicants, …
  - Places: countries, cities, properties, …
  - Things: proteins, genes, …

The Challenge: *How can we derive and maintain a temporally consistent view from…*

- **A lot** of information.
- Example:
  - US Security and Exchange Commission (SEC).



200,000 forms continuously filed every year

How has each insider's resume and stock portfolio changed over multiple years and multiple corporations?

The Challenge: *How can we derive and maintain a temporally consistent view from…*

- **Different (distributed) heterogeneous** sources.
- Example:
  - A patient may visit different physicians over the course of her lifetime, sometimes simultaneously.

Betsy's patient record at the Marshfield Clinic.

Betsy's patient record at Riverview Clinic.

?

Was Betsy taking Coumadin and Septra during March 2002, which are known to have adverse interactions, at the same time?

The Challenge: *How can we derive and maintain a temporally consistent view from…*

- **Conflicting** and **imprecise** information.

Resume for Wil Riker
Title: First Officer
2008-present
…

Resume for Wil Riker
Title: Captain
2009-present
…

Resume for Wil Riker
Title: First Officer
2009-present
…

?

Was Wil promoted, demoted, or does one version of his resume correct another? What does 'present' mean?

Yet another example – Social Media Data

Anna's Tweet

March 28, 11.01am. Anna: I am at SFMOMA …

Anna's Tweet

March 28, 1.15pm. Anna: Enjoying the Matisse!

Jessica's blog

I met Anna at Fleur De Lys on March 28 at 12 noon for lunch… enjoyed a delicious Chocolate Tartlet!

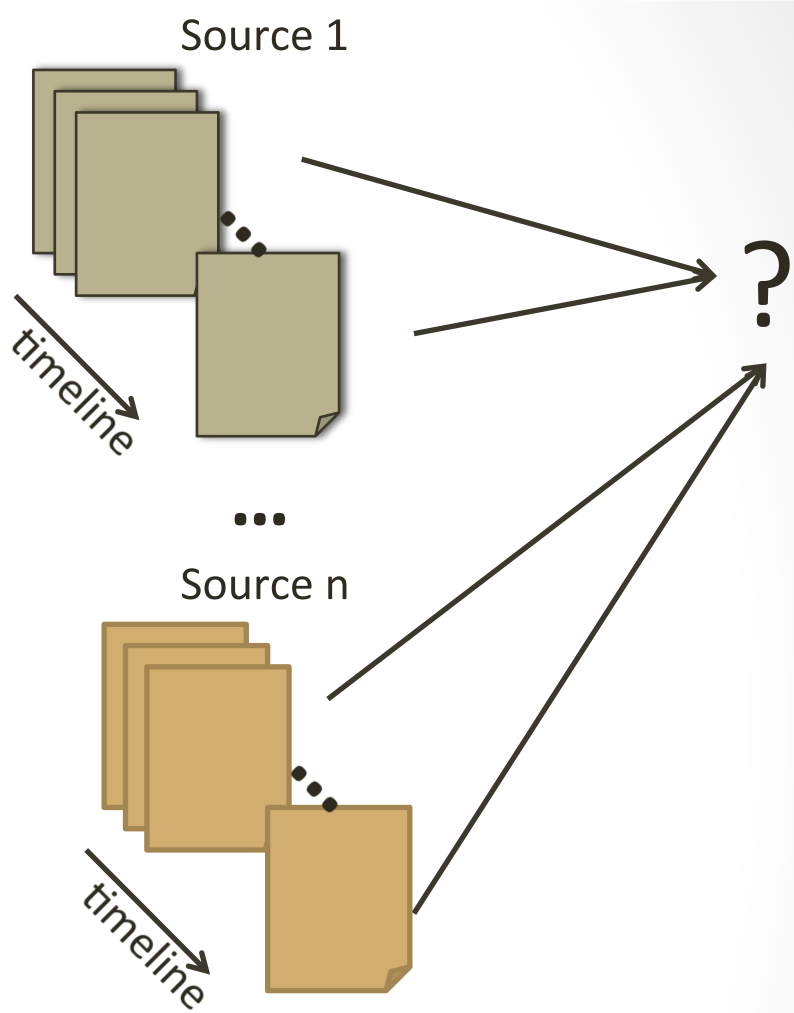
Anna's timeline and location on March 28:

11.01-12noon: @SFMoma.
12noon-1.15pm: @Fleur De Lys.
1.15pm onwards: @SFMoma.

The Challenge

- Heterogeneous and possibly large *time-aware data*:
  - Contain time information as part of data.
    - Implicit as part of data
    - Explicit e.g., version number.

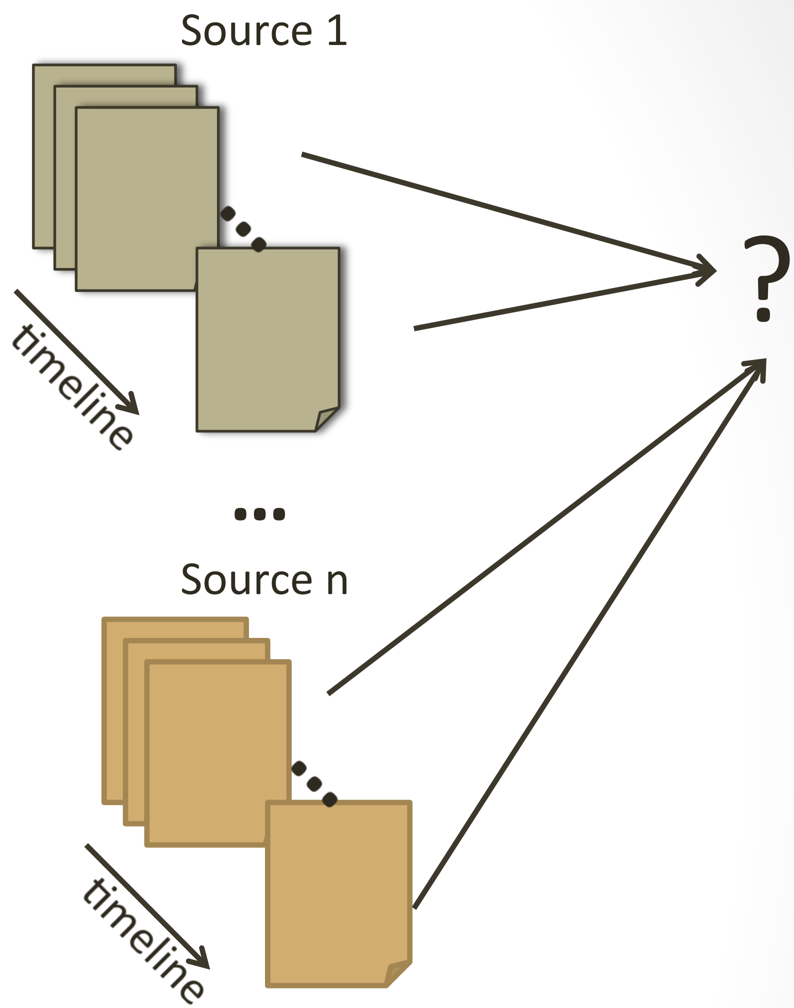
- **How can we uniformly manipulate and access (conflicting) data from these data sources?**



The integrated result should provide a complete description of the when-provenance of an entity w.r.t. the data sources.

The Challenge

- Heterogeneous and possibly large time-aware data:
  - Contain time information as part of data.
    - Implicit as part of data
    - Explicit e.g., version number.



- **How can we uniformly manipulate and access (conflicting) data from these data sources?**

What is needed is a foundation for consistent, scalable, and efficient integration of time-aware data.

When-provenance: The truth of a fact over time

- Time is a linear structure (**T**, <), also called a *time dimension*. [Chomicki, Toman 05]
  - < is the precedence relation. Transitive, irreflexive, asymmetric.

- A *time point* may involve multiple time dimensions, e.g., $(t_1, \ldots, t_k)$, where $k>=1$, and t_i is a time dimension.

- The **when-provenance of a fact f** is the set of all time points when f is true.

Representation of time

Representation for a time point:

- Write a time point as a record $[l_1{:}t_1, \ldots, l_k{:}t_k]$,

  l_i are labels. Sometimes, t_i is represented as dates. E.g.,

  [asof: 10/1/02, reported:10/2/03]

Succinct representation of multiple time points:

- A *time interval* [s,e) is often a compact representation of multiple time points.
  - s denotes the start time, e denotes the end time, and s <= e. Default semantics: closed on s, open on e.
  - (s,*) * denotes the end time is "now".

Representation of when-provenance

- The **when-provenance of a fact** can be represented as a set of records, called a *temporal vector*, where each record in the vector is of the form [l_1: v_1, …, l_n:v_n].
  - l_is are label names and v_is are time intervals.

- Such representation is akin to *temporally grouped models* [Clifford *et al.* 93], where an additional attribute on a N1NF relation is used to keep all time points where the tuple is true.

Example: Time-aware data sources about Freddy Gold

SEC Forms

Homepages

SEC filings (Forms 3/4/5, 10K)

Asof	Reported	Ticker	Shares
7/01/10	7/01/10	OLP	396043
8/25/10	8/26/10	OLP	13415
8/23/10	8/24/10	OLP	141
8/20/10	8/30/10 C	OLP	1322179
8/26/10	8/30/10 C	OLP	396043
7/09/10	8/22/10	BRT	1820
7/14/10	8/02/10	BRT	0

Versions of resume

Asof	Reported	School	Degree
1960	2000	NYL	JD

Asof	Reported	Corp	Title
1996	2000	BRT	CEO

News articles

Asof	[illegible]	[illegible]	[illegible]
1996-	[illegible]	[illegible]	[illegible]
1984	[illegible]	[illegible]	[illegible]
2005-2007	2012	OLP	CEO

Each row corresponds to information extracted from a version of the data source.

Versions of corporate websites

Asof	Reported	Corp	Title
2006	2006	OLP	CEO
2001	2007	BRT	Chair

Each row represents a distinct filing or version of the source instance.

Corporate webapges, wikipedia, …

Bloomberg, Business Times, …

Integrated profile of Freddy Gold

Freddy Gold 1960-now

Education 1960-now

Asof	Reported	School	Degree
1960	2000	NYL	JD

Positions 1984-now

Asof	Reported	Corp	Title
1984-now	2012	OLP	Chair
1996-2001	2000	BRT	CEO
2001–now	2007	BRT	Chair
2005-2007	2012	OLP	CEO

Stocks held 7/1/2010 – 9/30/2010

OLP

Asof	Reported	Shares held
7/1-8/20 8/26-now	7/01-now, 8/30-now	396043
8/20-8/23	8/30-now	1322179
8/23-8/25	8/24-now	141
8/25-8/26	8/26-now	13415

BRT

Asof	Reported	Shares held
7/09-7/14	8/22-now	1820
7/14-now	8/02-now	0

Integrated profile of Freddy Gold based on all reported information.

Extract, entity resolution

SEC filings (Forms 3/4/5, 10K)

Asof	Reported	Ticker	Shares
7/01/10	7/01/10	OLP	396043
8/25/10	8/26/10	OLP	13415
8/23/10	8/24/10	OLP	141
8/20/10	8/30/10 C	OLP	1322179
8/26/10	8/30/10 C	OLP	396043
7/09/10	8/22/10	BRT	1820
7/14/10	8/02/10	BRT	0

Versions of resume

Asof	Reported	School	Degree
1960	2000	NYL	JD

Asof	Reported	Corp	Title
1996	2000	BRT	CEO

News articles

Asof	Reported	Corp	Title
1996-2001	2012	BRT	CEO
1984	2012	OLP	Chair
2005-2007	2012	OLP	CEO

Versions of corporate websites

Asof	Reported	Corp	Title
2006	2006	OLP	CEO
2001	2007	BRT	Chair

Each row represents a distinct filing or version of the source instance.

Transform

Freddy Gold 1960-now

Education 1960-now

Asof	Reported	School	Degree
1960	2000	NYL	JD

Positions 1984-now

Asof	Reported	Corp	Title
1984-now	2012	OLP	Chair
1996-2001	2000	BRT	CEO
2001–now	2007	BRT	Chair
2005-2007	2012	OLP	CEO

Stocks held 7/1/2010 – 9/30/2010

OLP

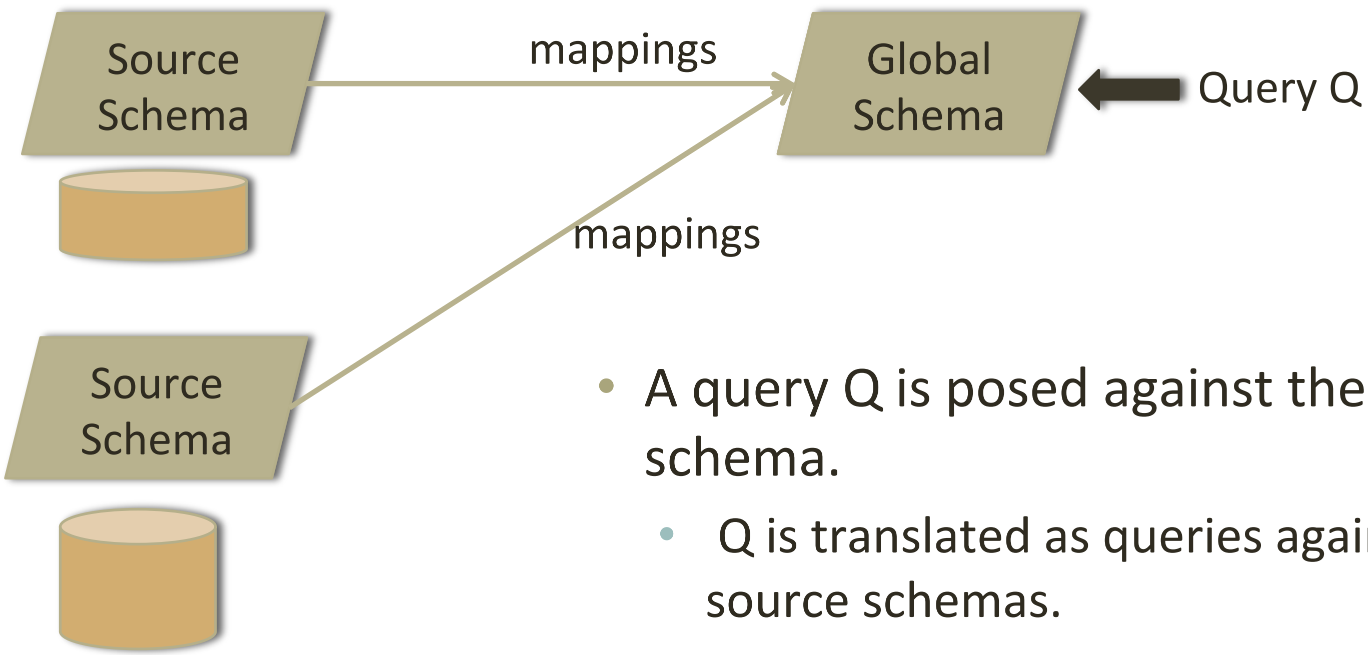
Asof	Reported	Shares held
7/1-8/20	7/01-now,	
8/26-now	8/30-now	396043
8/20-8/23	8/30-now	1322179
8/23-8/25	8/24-now	141
8/25-8/26	8/26-now	13415

BRT

Asof	Reported	Shares held
7/09-7/14	8/22-now	1820
7/14-now	8/02-now	0

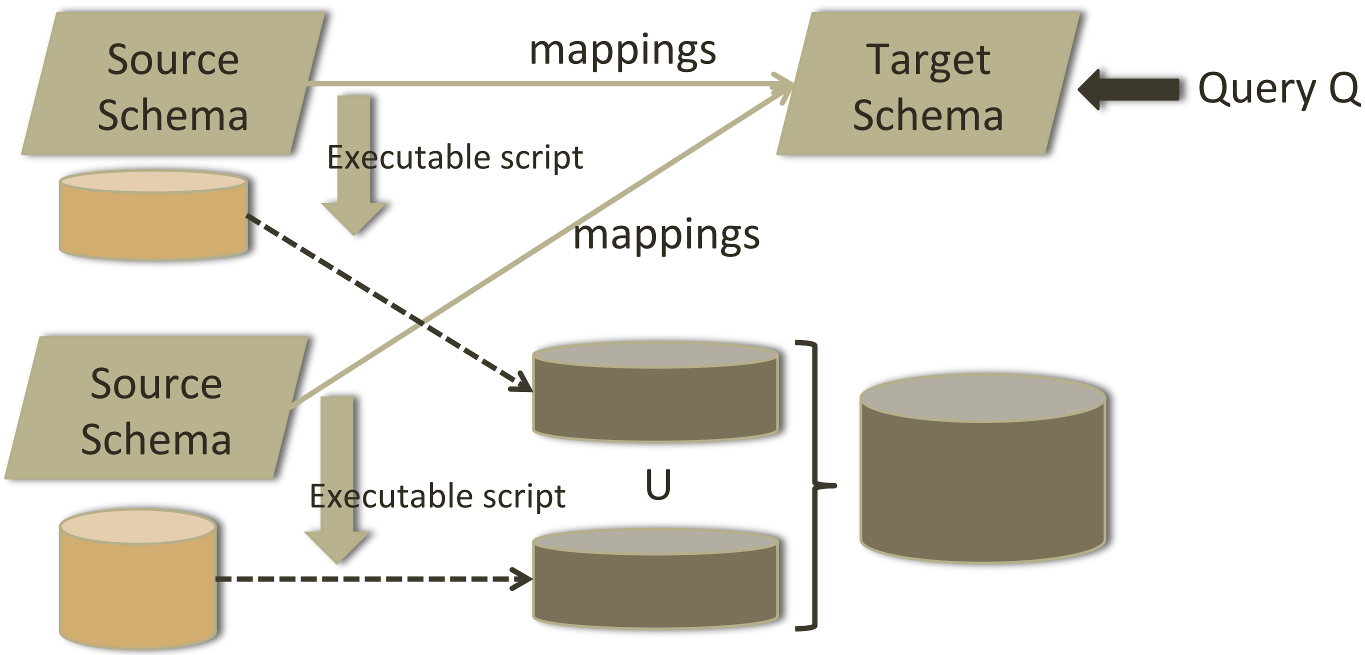
Integrated profile of Freddy Gold based on all reported information.

Data Integration: Basic Framework



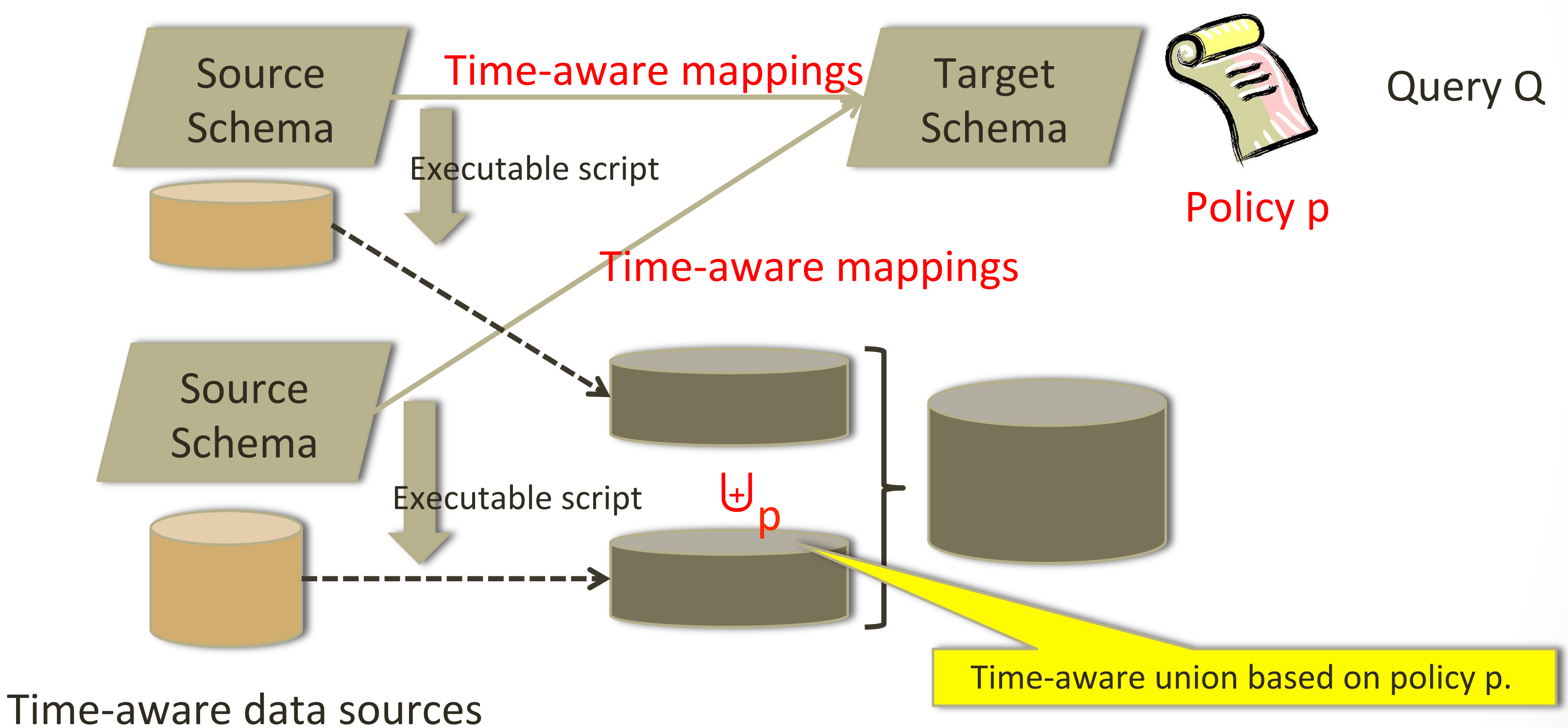
- A query Q is posed against the global schema.
  - Q is translated as queries against the source schemas.

Data Exchange: Basic Framework



- A query Q is posed against the target schema.

Time-aware Data Integration/Data exchange: Basic Framework



What's needed: A foundational framework for time-aware data integration/data exchange

- Time-aware data model
  - Model time as first-class construct.
  - Formalize time-sensitive schema constraints.
- Time-aware mapping rules
  - High-level language for specifying time-specific transformations.
- Data Integration and Data Exchange across time
  - Time-aware union under different policies.
- Others
  - Query Answering, Managing Changes

Basic time-aware data model

τ ::= Str | Int | now | (τ,τ) | SetOf τ | SetOf* τ |

Rcd[l_1:τ_1, …, l_n:τ_n] | Pair[l_1:τ_1, l_2:τ_2]

- Can define tree-like structures with set types, records and pairs.
  - Set types must be of the form SetOf Rcd.
  - SetOf Rcd must have *keys* defined.
- Time and data are both modeled as first class citizens.
- Essentially, every node can be associated with a *temporal vector,* through the Pair type.

Time-aware nested data model

Why nested data model?

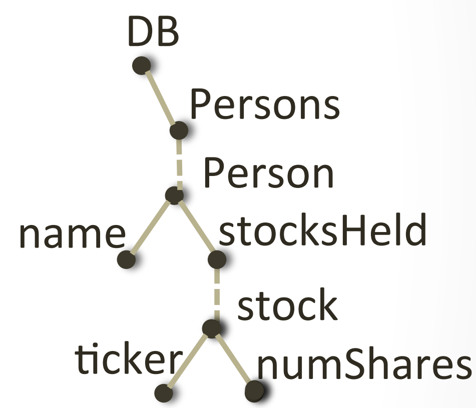
- Many data sources are hierarchical.
  - Modeled in JSON, XML, or proprietary formats.
  - E.g., SEC, Biological databases, Social Media Data.

- Easier to model the association of time information with data.

Example: Schema and Constraints

Definition of a bi-temporal vector.

```
starttime ::= Int;
endtime ::= Int | now;
ActRep ::= SetOf Rcd[asof: (starttime,endtime), reported:(starttime,endtime)];

DB ::=

    Persons: SetOf

        Person:
          Rcd[ name*: Str,
               stocksHeld: SetOf

                   stock: Rcd [ ticker*: Str,
                                numShares:
                                                 :Int]
                   ]]]
```



Example: Schema and Constraints

Temporal dependencies [Jensen *et al.* 96]

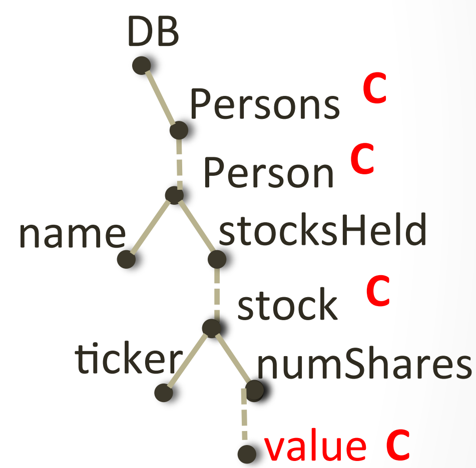
```
starttime ::= Int;
endtime ::= Int | now;
ActRep ::= SetOf Rcd[asof: (starttime,endtime), reported:(starttime,endtime)];

DB ::=
Pair[C:ActRep,
     Persons: SetOf
        Pair [C:ActRep,
           Person:
             Rcd[ name*: Str,
                  stocksHeld: SetOf
                   Pair[ C:ActRep,
                        stock: Rcd [ ticker*: Str,
                                numShares: SetOf*
                                  Pair[C: ActRep, value*:Int]
                  ]]]
```

Pair types are only used to associate a temporal vector to a label.

At any time, a person is uniquely identified by her name.

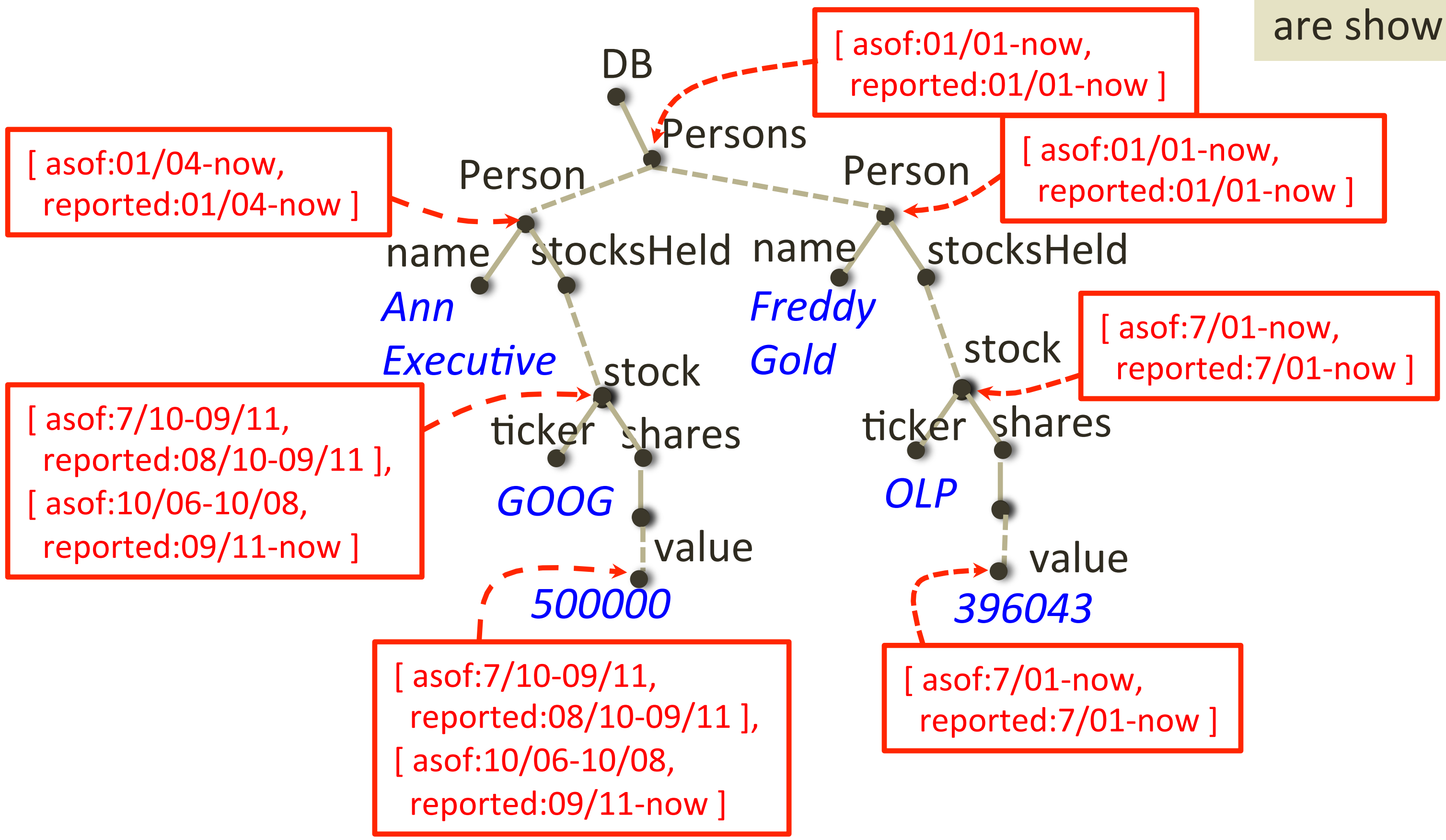
At any time, the stocks held by a person is uniquely identified by its ticker symbol.

At any time, number of shares of a ticker is unique.



Example: An instance

Temporal vectors are shown in red.



When-provenance of an entity?

- **Fact**: Every node (or *entities*) of an instance can be uniquely identified by a sequence of labels and key values.

- Examples of identifiers for entities:
  - DB/Persons
  - DB/Persons/Person(name="Freddy Gold")
  - DB/Persons/Person(name="Freddy Gold")/stocksHeld/stock(ticker="OLP")

- Goal is to have the when-provenance of an entity described by the temporal vector that is associated with that entity.
- So how can we construct the when-provenance of an entity from heterogeneous time-aware data sources?

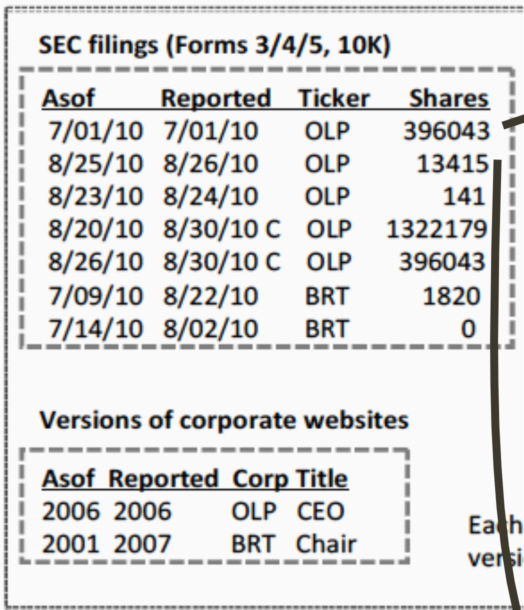
Time-aware mapping rules

- A *data model* is a mathematical formalism that consists of two parts:
    1. A notation for describing data and mathematical objects for representing data.
    2. A set of operations for manipulating data.

We now have a notation for describing time-aware data.

- **Next**: What is an appropriate language or set of operations for manipulating time-aware data?

- **Desiderata**: The proposed framework must embrace existing data integration and data exchange framework as a special case.

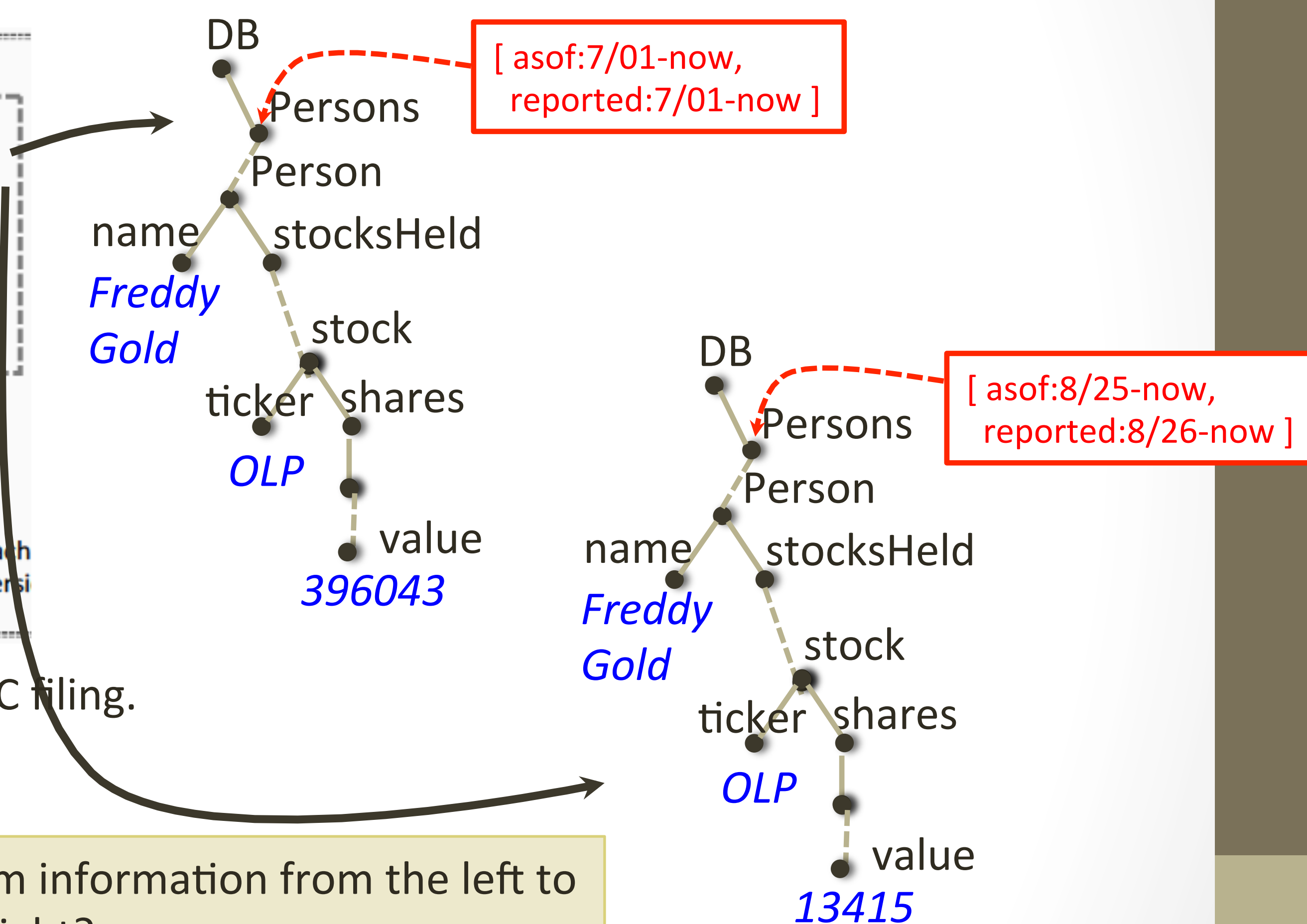


SEC filings (Forms 3/4/5, 10K)

Asof	Reported	Ticker	Shares
7/01/10	7/01/10	OLP	396043
8/25/10	8/26/10	OLP	13415
8/23/10	8/24/10	OLP	141
8/20/10	8/30/10 C	OLP	1322179
8/26/10	8/30/10 C	OLP	396043
7/09/10	8/22/10	BRT	1820
7/14/10	8/02/10	BRT	0

Versions of corporate websites

Asof	Reported	Corp	Title
2006	2006	OLP	CEO
2001	2007	BRT	Chair



Each row denotes an SEC filing.

1) How do we transform information from the left to information on the right?
2) What is the number of OLP shares held by Freddy on 8/25? Is it 396043 or 13415?

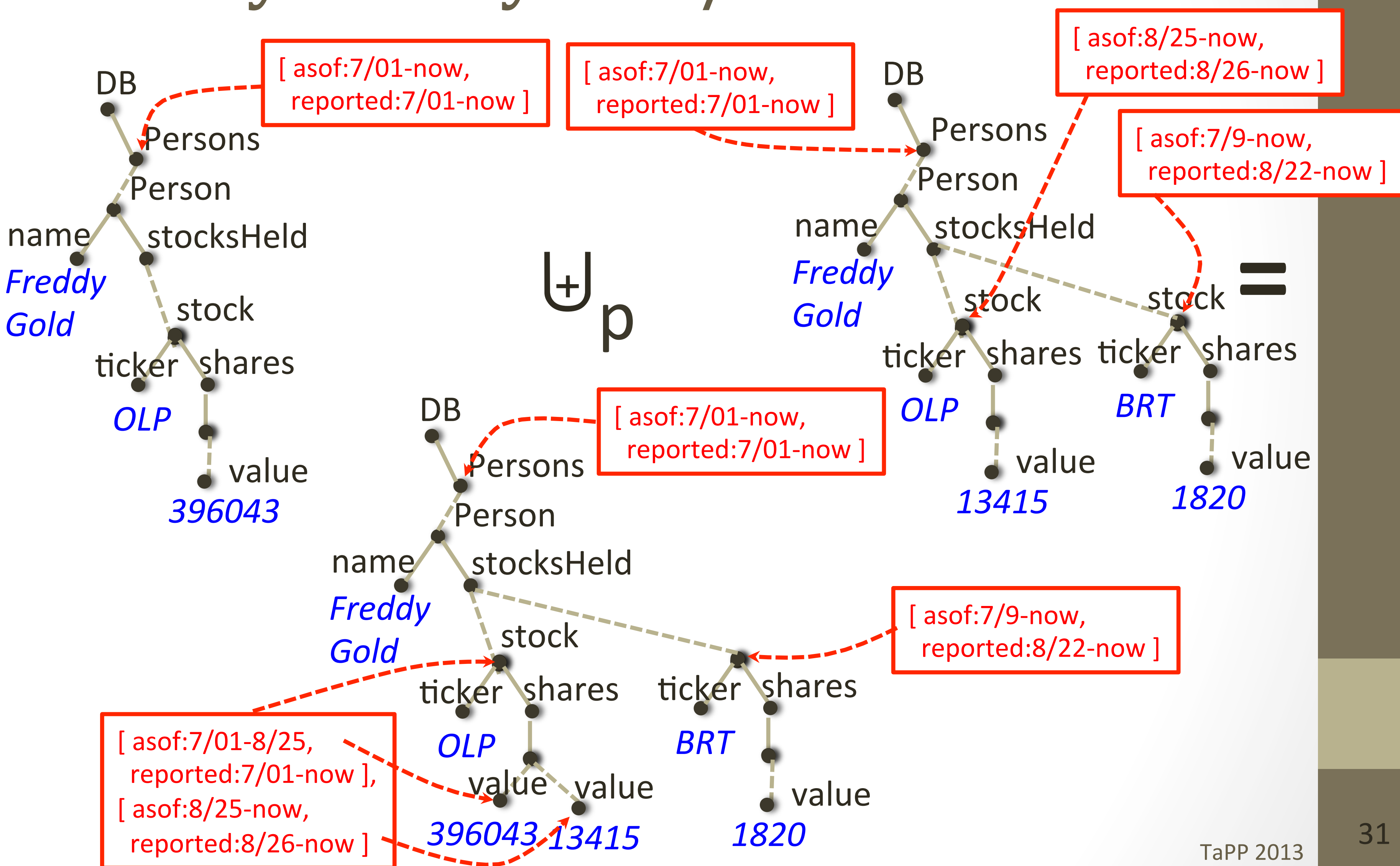
Time-aware mapping rules

```
FOR     f IN Filings
EXISTS  p IN DB.Persons, s IN p.stocksHeld,
        (t1,t2) IN TV(p)
WITH    p.name = f.name,
        s.ticker = f.ticker,
        s.numShares = f.numShares,
        t1 >= f.asof, t2 >= f.since;
```

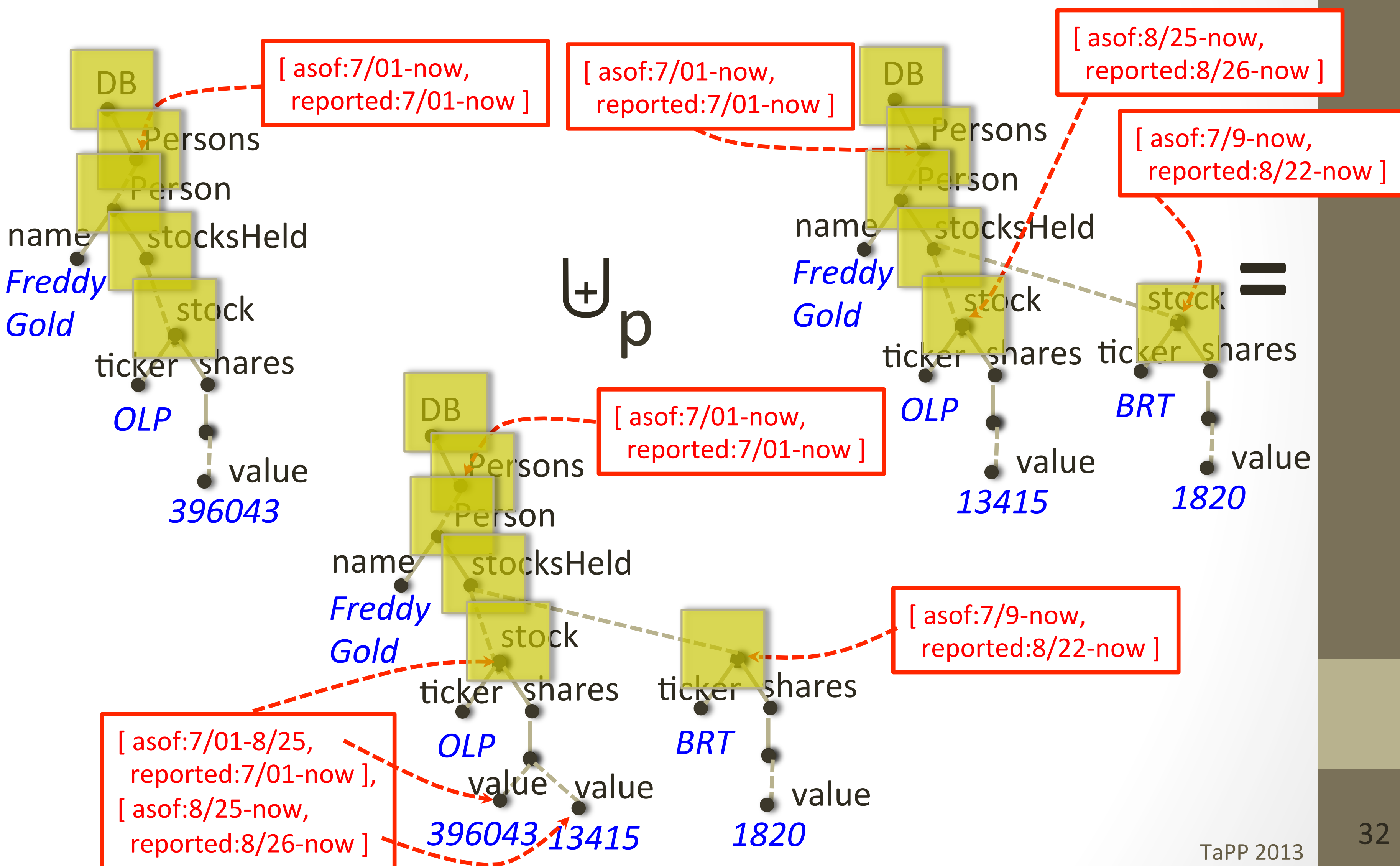
Extends well-known schema mapping language [Popa *et al.* 02, FKMP03] with syntax for manipulating temporal vectors.

For each filing, create a DB.Persons.person node, where person's name, ticker, numShares value are equal to the respective values from f, and the temporal vector of p is defined by the asof and reported times from f.

What is the number of OLP shares held by Freddy on 8/25 ?



Time-aware Union $\uplus_p$



Time-aware Union

- **Input**: A schema **S**, two instances T_1 and T_2 of **S**, and a policy **P**.
- **Output:** An integrated result of T_1 and T_2 that conforms to **S** based on policy **P**.
- One-pass recursive merge of nodes down the tree.
  - Entities are sorted and identified by their keys.
- Easy parallelization.

Policy

- Method of resolving conflicts.
- Conflicts occur when constraints imposed by the schema cannot be satisfied.
  - Example: Freddy can either own 390643 or 13415 OLP shares on 8/25 but not both.
  - Example: There can only be one Freddy Gold at any point in time.

Time-aware Union desiderata

- Important algebraic identities that should be enforced:
- Idempotence: $T \uplus_p T \approxeq T$
- Commutativity: $T_1 \uplus_p T_2 \approxeq T_2 \uplus_p T_1$
- Associativity: $(T_1 \uplus_p T_2) \uplus_p T_3 \approxeq T_1 \uplus_p (T_2 \uplus_p T_3)$

- Would guarantee equivalent result regardless of order of integration (modulo representation of time).
- If these properties hold, then time-aware union is well-suited for data integration/exchange.
  - Policy is "well-behaved".

"Well-behaved" policies

- The known truth of a fact may be adjusted as data is combined from different sources.
- Application-specific semantics through policies are required to resolve conflicts that arise during the integration.
- A policy must specify which data value to "favor", and how to adjust the "out-of-favor" value.

- Time-based policies:
  - Favor newer evidence or older evidence. Adjust by removing certain conflicting time-points.
- Source-based policies:
  - Favor evidence based on source. Discard "out-of-favor" evidence.
- Combination:
  - Favor by source, then by time.

Template for specifying a time-based policy

- Input: R_1: [l_1:(s_1,e_1), …, l_k:(s_k,e_k)], R_2: [l_1:(s_1',e_1'), …, l_k:(s_k',e_k')]
- Output: A (modified) R_1 and R_2 pair with no overlap.

- If R_1 and R_2 overlap
  - Specify which time dimension to use to decide which record to favor.
    - E.g., favor record with a larger start time for dimension i. "Out-of-favor" record will be minimally adjusted on dimension i to avoid overlap.
  - Specify how ties are broken.
    - E.g., if both records have the same start time for dimension i, keep R_1, discard R_2.
- Return R_1 and R_2.

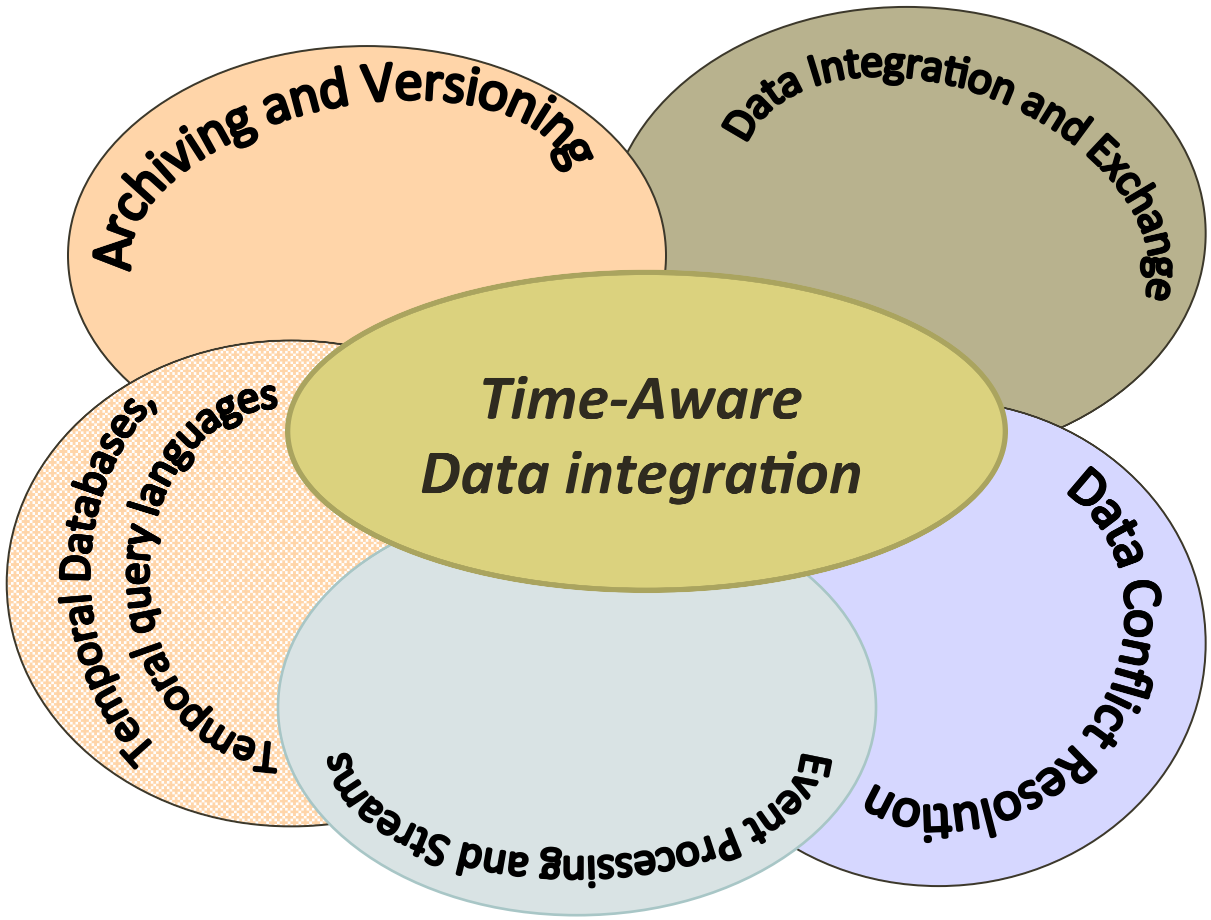
The time-based policy where adjustments are made on "asof" time applies to the SEC example.

Theorem: Let p be a time-based, source-based, or combination-based policy and let T_1, T_2, and T_3 be three instances that conform to a schema **S**. Then the following holds:

- Idempotence: $T \uplus_p T \approxeq T$
- Commutativity: $T_1 \uplus_p T_2 \approxeq T_2 \uplus_p T_1$
- Transitivity: $(T_1 \uplus_p T_2) \uplus_p T_3 \approxeq T_1 \uplus_p (T_2 \uplus_p T_3)$

Related work

- Related to temporal databases, data integration and data conflict resolution.



Bi-temporal databases

Two notions of time:

- Valid time: The time a fact is true in the real world.
- Transaction time: The time a fact is entered into a database system. Can only increase.

- Application-specific notions of time do not always match valid and transaction-time semantics of bi-temporal databases.

- See *Temporal database entries for Encyclopedia of Database Systems*, 2009.

Where was Anna?

1 — Anna: Twitter

@SFMoma
Since: Mar 28, 11.01am
Retrieved: Apr 1

2 — Anna: Twitter

@SFMoma
Since: Mar 28, 1.15pm
Retrieved: Mar 30

3 — Jessica's blog

Anna@Fleur De Lys
Since: Mar 28, 12noon
Retrieved: Mar 29

- Attempt to integrate information about Anna from Twitter and blogs using bi-temporal databases.
- No direct support for application-specific notions of time: "since" and "retrieved"
- Match "Since" to valid time (or business time), "Retrieved" to transaction time.

Where is Anna?

- DB2 syntax for inserting these records into the DBMS.

	Twitter		Twitter		Jessica's blog
1	**@SFOMoma** **Since: Mar 28, 11.01am** **Retrieved: Apr 1**	2	**@SFOMoma** **Since: Mar 28, 1.15pm** **Retrieved: Mar 30**	3	**@Fleur De Lys** **Since: Mar 28, 12noon** **Retrieved: Mar 29**

1

```
UPDATE DB FOR PORTION OF BUSINESS_TIME
FROM '03/28/13 11.01am' to CURRENT DATE
SET LOCATION= 'SFMoma',
WHERE NAME = 'Anna'
```

2

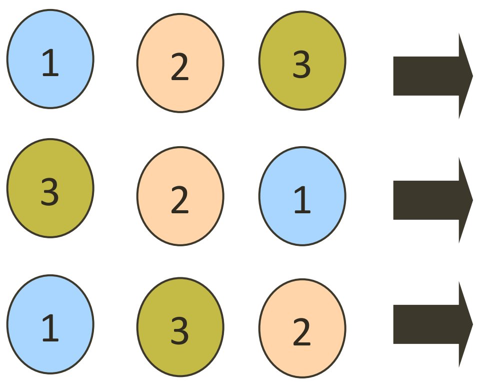
```
UPDATE DB FOR PORTION OF BUSINESS_TIME
FROM '03/28/13 1.15pm' to CURRENT DATE
SET LOCATION= 'SFMoma'
WHERE NAME = 'Anna'
```

3

```
UPDATE DB FOR PORTION OF BUSINESS_TIME
FROM '03/28/13' to CURRENT DATE
SET LOCATION='Fleur De Lys'
WHERE NAME = 'Anna'
```

Where is Anna?

- Answer to this question depends on the order in which facts are entered into the database.



Anna is at Fleur De Lys since Mar 28, 12 noon.

Anna is at SFMoma since Mar 28, 11.01am.

Anna is at SFMoma since Mar 28, 1.15pm.

- The “right” answer: On March 28, Anna was at
  - SFMoma from 11.01am to 12noon.
  - Fleur De Lys from 12noon to 1.15pm.
  - SFMoma from 1.15pm till now.

- Time-based policy by adjusting “since”.

When-provenance of Anna’s whereabouts:

(SFOMoma, {[since:11.01am-12noon, retrieved:Apr1 – now],
[since:1.15pm-now, retrieved:Mar30-now]})

(FDL, {[since: 12noon-1.15pm, retrieved: Mar29 – now]})

- Significant application-specific logic needs to be added on top of bi-temporal databases to derive the “right” answers to the when-provenance of Anna.

- Need to handle multiple dimensions of time.

- Need to handle out-of-order “updates” for data integration and data exchange.

A little more related work …

Archiving and Versioning

- Archiving Scientific Data [Buneman, Khanna, Tajima, T. 04].
  - Time-aware union draws inspiration from this work.
  - Nested merge applied on a linear evolution of data, only one dimension of time.
    - Does not manipulate time information that may exist within each version.
    - A fact that exists in a version is assumed to be true. A fact that is missing from a version is assumed to be false.

- Version, delta-based approaches. [Wang *et al.* 08, Marian *et al.* 2001; Chien *et al.* 2001; Chawathe *et al.* 1998]

A little more related work

Data Conflict Resolution

- Data Fusion [Bleiholder, Naumann 2009], Data Fusion - Resolving Data Conflicts for Integration [Dong, Naumann 2009]
  - Variants of union that implement various conflict resolution strategies. (e.g., freshness of source, prefer values over null values etc.)
  - Not clear algebraic identities would hold. No manipulation of time.

A little more related work

Event Processing and Streams

- CEDR [Barga *et al.* 07]
  - Tri-temporal model
    - valid time interval, occurrence time interval, and CEDR time interval.
  - Events can correct or retract earlier events.
- Single valid time interval and occurrence time interval. Conflict resolution is not automatic.

Immediate Challenges

- Can we develop an efficient time-aware union implementation?
  - Handle large datasets.
  - Handle partial “updates”.
- Is there a larger class of policy language for which time-aware union satisfy the algebraic identities?
- What is an appropriate time-aware mapping language?

THANK YOU