

Dataspaces: The Tutorial

Day 2

Alon Halevy, David Maier
VLDB 2008
Auckland, New Zealand

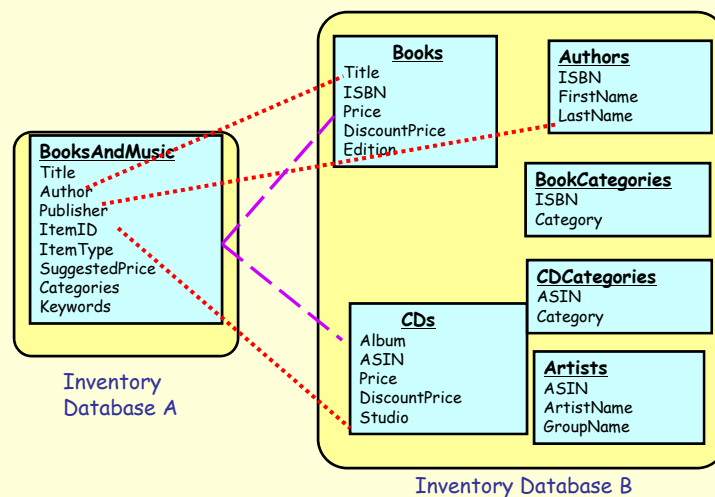
Outline

- ✓ Dataspaces: why? What are they?
 - Examples and motivation
- Dataspace techniques:
 - ✓ Locating and understanding data sources
 - Creating mappings and mediated schemas
 - Pay-as-you-go: improving with time
 - Querying dataspaces
- Research challenges on specific dataspaces:
 - Science, the desktop, the Web

Sub-Outline

- What are schema matches and mappings?
 - Why is it so hard to create them?
- Automatic techniques for creating them
- Probabilistic schema mappings
- Probabilistic mediated schemas
- Trails: mapping hints

Schema Matching and Mapping



Tabular structure, attribute names, synonyms, hypernyms, Coverage, level of detail, ...

Why is it so Hard?

- Schemas were developed in different contexts for different purposes
- Schemas **never fully capture** their intended meaning:
 - They're just symbols and structures.
 - Descriptions are:
 - Often missing,
 - In plain text, or wrong, or,
 - Don't capture all the semantics

Schema Mapping Overview

- Step 1: schema matching:
 - Generate *correspondences* between elements of the two schemas
 - Easier to elicit from designers
 - May actually be all that's needed
- Step 2: create mappings:
 - Decide on joins, unions, filters, ...

User in the loop in both steps

See Chapter 5 of upcoming book

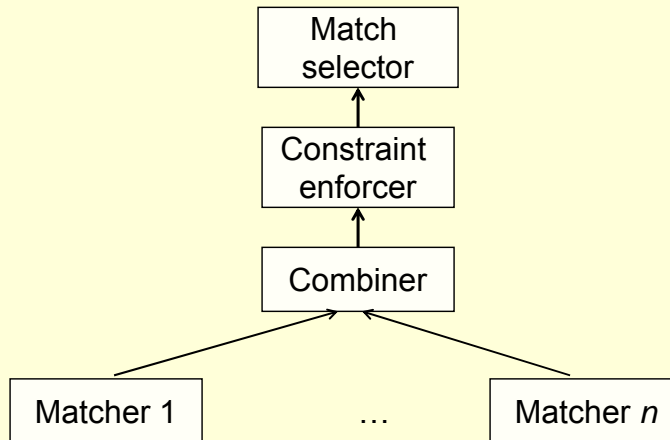
Sub-Outline

- ✓ What are schema matches and mappings?
 - ✓ Why is it so hard to create them?
- Automatic techniques for creating them
- Probabilistic schema mappings
- Probabilistic mediated schemas
- Trails: mapping hints

Schema Matching overview

- One trick won't do it all
- Hence:
 - Consider several *base* matchers
 - And then combine them
- Exploit domain constraints when possible
- We focus on 1-1 matching here
- [See Survey by Rahm & Bernstein, 2001]

Schema Matching Architecture



Basic Matchers

- Schema level:
 - Name, description, data type,
 - Constraints (keys, foreign keys, is-a)
 - Schema structure
- Instance level:
 - Look for common patterns in the data
 - Often more meaningful than the schema

Example: Edit Distance

Levenshtein Distance:

Number of operations needed to transform one name to the other.

$$edSim(s_1, s_2) = 1 - \frac{edit_distance(s_1, s_2)}{\max(length(s_1), length(s_2))}$$

edSim(discountPrice, discountedPrice)?

Instance-Based Matchers

- Formatting patterns in the data can reveal type:
 - E.g., dates, phone numbers, prices, addresses, names, ...
- What other attribute names were used elsewhere for such values?
 - Additional clues to name matcher
- Consider similarity in values & type between two columns
 - E.g., house price versus # of rooms

Sub-Outline

- ✓ What are schema matches and mappings?
 - ✓ Why is it so hard to create them?
- ✓ Automatic techniques for creating them
 - Probabilistic schema mappings
 - Probabilistic mediated schemas
 - Trails: mapping hints

Probabilistic Schema Mappings

- In a dataspace, we may rely on automatically created schema mappings
--> uncertainty
- How do we model uncertain mappings?
- How do we answer queries in their presence?

Probabilistic Mappings

[Dong, H., Yu, VLDB 2007]

- $S=(pname, email-addr, home-addr, office-addr)$
- $T=(name, mailing-addr)$

Possible Mapping	Probability
$\{(pname,name),(home-addr, mailing-addr)\}$	0.5
$\{(pname,name),(office-addr, mailing-addr)\}$	0.4
$\{(pname,name),(email-addr, mailing-addr)\}$	0.1

Semantics? by table or by tuple?

By-Table v.s. By-Tuple Semantics

Possible Mapping	Probability
$\{(pname,name),(home-addr, mailing-addr)\}$	0.5
$\{(pname,name),(office-addr, mailing-addr)\}$	0.4
$\{(pname,name),(email-addr, mailing-addr)\}$	0.1

$D_S=$

pname	email-addr	home-addr	office-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

$D_T=$

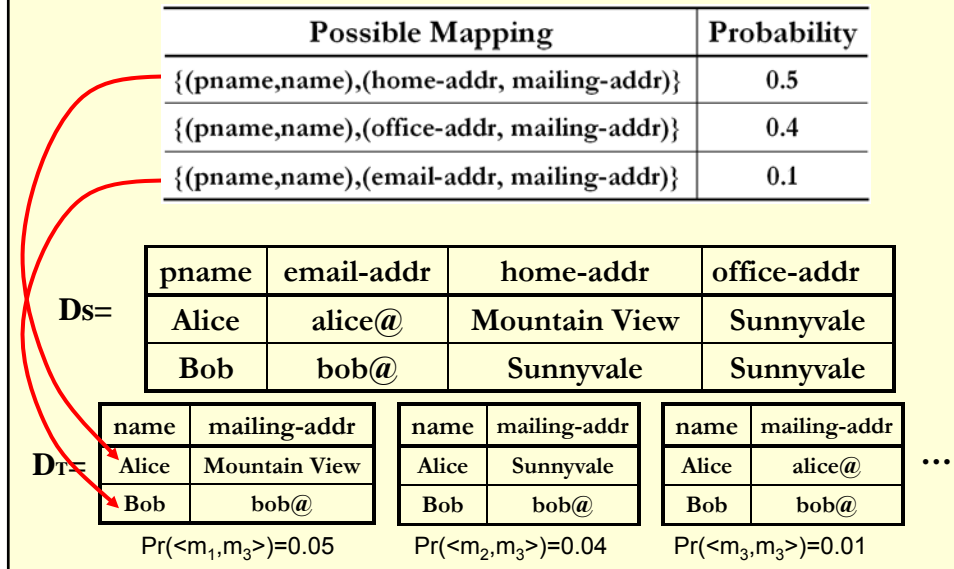
name	mailing-addr	name	mailing-addr	name	mailing-addr
Alice	Mountain View	Alice	Sunnyvale	Alice	alice@
Bob	Sunnyvale	Bob	Sunnyvale	Bob	bob@

$Pr(m_1)=0.5$

$Pr(m_2)=0.4$

$Pr(m_3)=0.1$

By-Table v.s. **By-Tuple** Semantics



Complexity of Query Answering

	By-table	By-tuple
Data Complexity	P _{TIME}	#P-complete
Mapping Complexity	P _{TIME}	P _{TIME}

Works for compressed representations of mappings too.

Results extend to more complex mapping languages.

P_{TIME} for important special cases

Sub-Outline

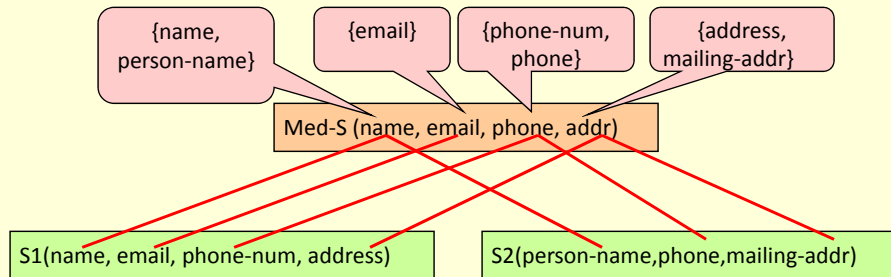
- ✓ What are schema matches and mappings?
 - ✓ Why is it so hard to create them?
- ✓ Automatic techniques for creating them
- ✓ Probabilistic schema mappings
 - Probabilistic mediated schemas
 - Trails: mapping hints

Creating the Mediated Schema

[Das Sarma, Dong, H., SIGMOD 2008]

- Mediated schema creation: up front effort.
- Can we create it automatically?
 - If we can, then we can completely bootstrap data integration.
- Probabilistic mediated schemas:
 - manage the uncertainty involved.

Example Mediated Schema



- A *mediated schema* is a clustering of a subset of the set of all attributes appearing in source schemas.

21

Why *Probabilistic* Mediated Schema?

Med1 ({name}, {phone, hPhone, oPhone}, {address, hAddr, oAddr})

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name, phone, address)

Q: SELECT name, hPhone, oPhone FROM Med

22

Why *Probabilistic* Mediated Schema?

Med1 ({name}, {phone, hPhone, oPhone}, {address, hAddr, oAddr})

Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name, phone, address)

Q: SELECT name, phone, address FROM Med

23

Why *Probabilistic* Mediated Schema?

Med1 ({name}, {phone, hPhone, oPhone}, {address, hAddr, oAddr})

Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

Med3 ({name}, {phone, hPhone}, {oPhone}, {address, hAddr}, {oAddr})

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name, phone, address)

Q: SELECT name, phone, address FROM Med

24

Why *Probabilistic* Mediated Schema?

Med1 ({name}, {phone, hPhone, oPhone}, {address, hAddr, oAddr})

Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

Med3 ({name}, {phone, hPhone}, {oPhone}, {address, hAddr}, {oAddr})

Med4 ({name}, {phone, oPhone}, {hPhone}, {address, oAddr}, {hAddr})

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name,phone,address)

Q: SELECT name, phone, address FROM Med

25

Why *Probabilistic* Mediated Schema?

Med1 ({name}, {phone, hPhone, oPhone}, {address, hAddr, oAddr})

Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

Med3 ({name}, {phone, hPhone}, {oPhone}, {address, hAddr}, {oAddr})

Med4 ({name}, {phone, oPhone}, {hPhone}, {address, oAddr}, {hAddr})

Med5 ({name}, {phone}, {hPhone}, {oPhone}, {address}, {hAddr}, {oAddr})

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name,phone,address)

Q: SELECT name, phone, address FROM Med

26

Why *Probabilistic* Mediated Schema?

Med1 ({name}, {phone, hPhone, oPhone}, {address, hAddr, oAddr})

Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

Med3 ({name}, {phone, hPhone}, {oPhone}, {address, hAddr}, {oAddr})

Med4 ({name}, {phone, oPhone}, {hPhone}, {address, oAddr}, {hAddr})

Med5 ({name}, {phone}, {hPhone}, {oPhone}, {address}, {hAddr}, {oAddr})

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name, phone, address)

Q: SELECT name, phone, address FROM Med

27

Probabilistic Mediated Schema

Med3 ({name}, {phone, hPhone}, {oPhone}, {address, hAddr}, {oAddr})

Pr=.5

Med4 ({name}, {phone, oPhone}, {hPhone}, {address, oAddr}, {hAddr})

Pr=.5

S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name, phone, address)

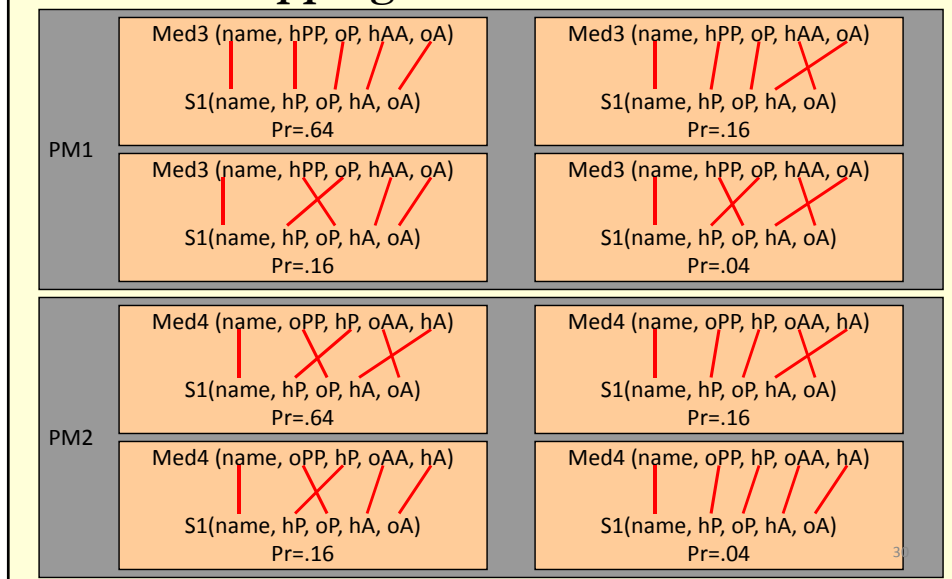
28

Probabilistic Mediated Schema

- A p-med-schema is a set
 - $\mathbf{M} = \{ (M_1, Pr(M_1)), \dots, (M_n, Pr(M_n)) \}$ where
 - M_i is a med-schema; $i \neq j \Rightarrow M_i \neq M_j$
 - $Pr(M_i) \in (0, 1]$; $\sum Pr(M_i) = 1$

29

P-Mappings w.r.t. P-Med-Schema



Bootstrapping Data Integration

- Need to choose a mapping based on the correspondences:
 - One that minimizes entropy
- Consolidate probabilistic med schemas into one -- for the user.
- Between 0.85 and 0.95 P/R for queries on collections of 50-800 tables from the Web.

Sub-Outline

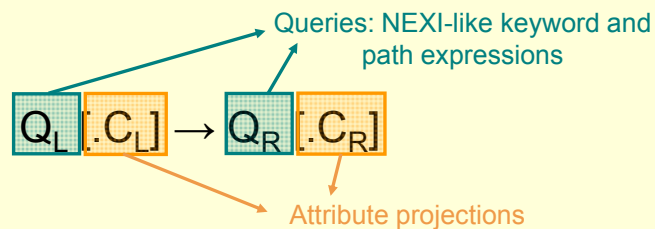
- ✓ What are schema matches and mappings?
 - ✓ Why is it so hard to create them?
- ✓ Automatic techniques for creating them
- ✓ Probabilistic schema mappings
- ✓ Probabilistic mediated schemas
- Trails: mapping hints

iTrails: Add Integration Hints Incrementally [Vas Salles et al., VLDB 06, 07]

- **Step 1:** Provide a search service over **all** the data
 - Use a general graph data model (see VLDB 2006)
 - Works for unstructured documents, XML, and relations
- **Step 2:** Add integration semantics via hints (**trails**) on top of the graph
 - Works across data sources, not only between sources
- **Step 3:** If more semantics needed, go back to step 2
- **Impact:**
 - Smooth transition between **search** and **data integration**
 - Semantics added incrementally improve **precision / recall**

Defining Trails

- Basic form of a Trail



- Intuition: When I query for $Q_L [.C_L]$, you should also query for $Q_R [.C_R]$

Trail Examples: Global Warming Zurich

global warming zurich



Temperatures

date	city	region	celsius
24-Sep	Bern	BE	20
24-Sep	Uster	ZH	15
25-Sep	Zurich	ZH	14
26-Sep	Zurich	ZH	9

- **Trail for Implicit Meaning:** “When I query for global warming, you should also query for Temperature data above 10 degrees”

```
global warming →
//Temperatures/*[celsius > 10]
```

- **Trail for an Entity:** “When I query for zurich, you should also query for references of zurich as a region”

```
zurich → //*[region = "ZH"]
```

Trail Example: Deep-Web Bookmarks

train home



- **Trail for a Bookmark:** “When I query for train home, you should also query for the TrainCompany’s website with origin at ETH Uni and destination at Seilbahn Rigiblick”

```
train home →
//trainCompany.com/*[origin="ETH Uni"
and dest = "Seilbahn Rigiblick"]
```

Station/Stop	Date	Time	Platform	Products	Comments
Zürich, ETH/Universitätsspital	15.09.07	dep 19:05		Trm 9	Trm Direction: Zürich, Hirzenbach
Zürich, Seilbahn Rigiblick		arr 19:08			

Duration: 0:03; runs Sa
Hint: Departure/Arrival replaced by an equivalent station
Tariff level: 9; Zones: 10; Short distance

Trail Examples: Schema Equivalences

Employee		
empID	empName	salary

Person			
SSN	name	age	income

- **Trail for schema match on names:**
“When I query for Employee.empName, you should also query for Person.name”

```
//Employee//*.tuple.empName →
//Person//*.tuple.name
```

- **Trail for schema match on salaries:**
“When I query for Employee.salary, you should also query for Person.income”

```
//Employee//*.tuple.salary →
//Person//*.tuple.income
```

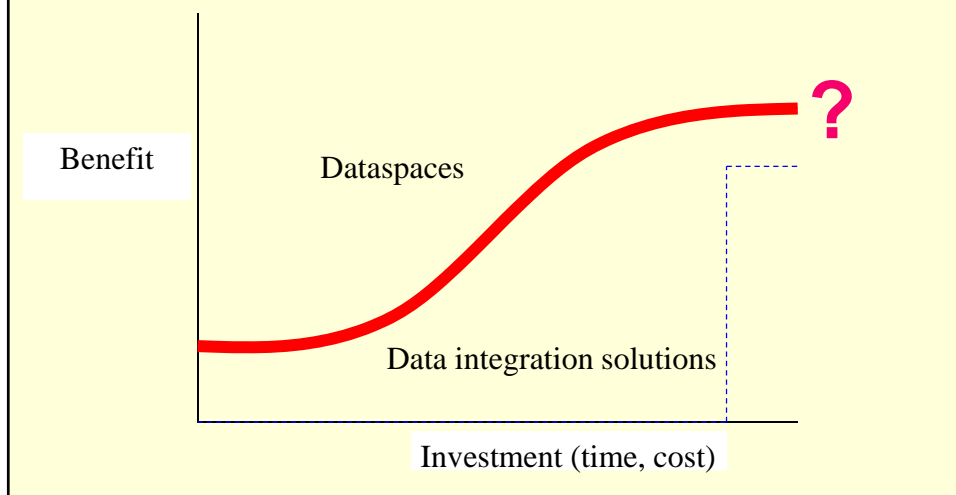
More on Trails

- Creation:
 - Given by the user explicitly or by relevance feedback.
 - (Semi-)Automatically: information extraction, schema matching, user communities, ontologies.
- Uncertainty on trails: some paths are better than others.
- Query reformulation: avoid cycles. (see paper)

Outline

- ✓ Dataspaces: why? What are they?
 - Examples and motivation
- Dataspace techniques:
 - ✓ Locating and understanding data sources
 - ✓ Creating mappings and mediated schemas
 - Pay-as-you-go: improving with time
 - Querying dataspace
- Research challenges on specific dataspace:
 - Science, the desktop, the Web

Getting the Red Curve

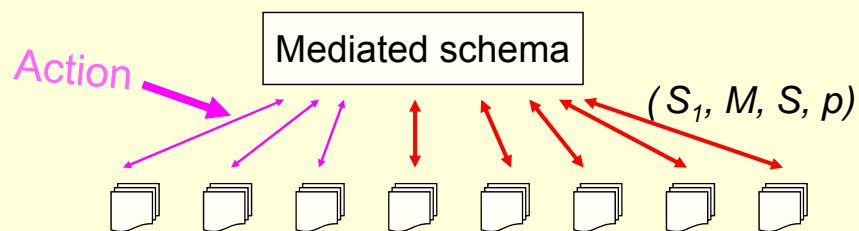


Reusing Human Attention

- Principle:
 - *User action = statement of semantic relationship*
 - *Leverage actions to infer other semantic relationships*
- Examples
 - Providing a semantic mapping
 - *Infer other mappings*
 - Writing a query
 - *Infer content of sources, relationships between sources*
 - Creating a “digital workspace”
 - *Infer “relatedness” of documents/sources*
 - *Infer co-reference between objects in the dataspace*
 - Annotating, cutting & pasting, browsing among docs
- ESP [von Ahn], mass collaboration [Doan+], active learning for record matching [Sarawagi et al.]

Learning Schema Mappings

[Doan et al., 2001]



- Classifiers for mediated schema
- Training examples: manually created schema matches
- Technique: multi-strategy learning. Use different learners and combine their predictions.
- Used in Transformic Inc. to create thousands of mappings.

Soliciting User Feedback

[Jeffrey, Franklin, H., SIGMOD 2008]

- After bootstrapping, we need help from users to improve.
 - Reference reconciliation
 - Schema matches
 - Extractions from text
- What questions should we ask the users?

The Most Beneficial Match

Decision theory to the rescue!

→ *Value of Perfect Information (VPI)*

“What is the benefit of resolving an unknown?”

Intuition:

$$\text{Benefit}(\text{match } m_j) = \text{Utility}(m_j \text{ confirmed})(p_{\text{correct}}) + \text{Utility}(m_j \text{ disconfirmed})(1 - p_{\text{correct}}) - \text{Utility}(\text{without asking})$$

Utility of a Dataspace

- Focus on queries!
- 2 components:
 - Result quality
 - Query importance

$$U(D, M) = \sum_{(Q_i, w_i) \in W} r(Q_i, D, M) w_i$$

Look at all queries in the workload
 Query result quality (precision/recall)
 Query importance

Challenges

- How to estimate benefit without computing all queries?
- Don't want to check all possible resulting dataspace when a match is confirmed.
- Result: much faster dataspace improvement
 - Experiments on GoogleBase data.

Outline

- ✓ Dataspaces: why? What are they?
 - Examples and motivation
- Dataspace techniques:
 - ✓ Locating and understanding data sources
 - ✓ Creating mappings and mediated schemas
 - ✓ Pay-as-you-go: improving with time
 - Querying dataspaces
- Research challenges on specific dataspace:
 - Science, the desktop, the Web



Looking for data
management
problems
in the rainforest in
Costa Rica

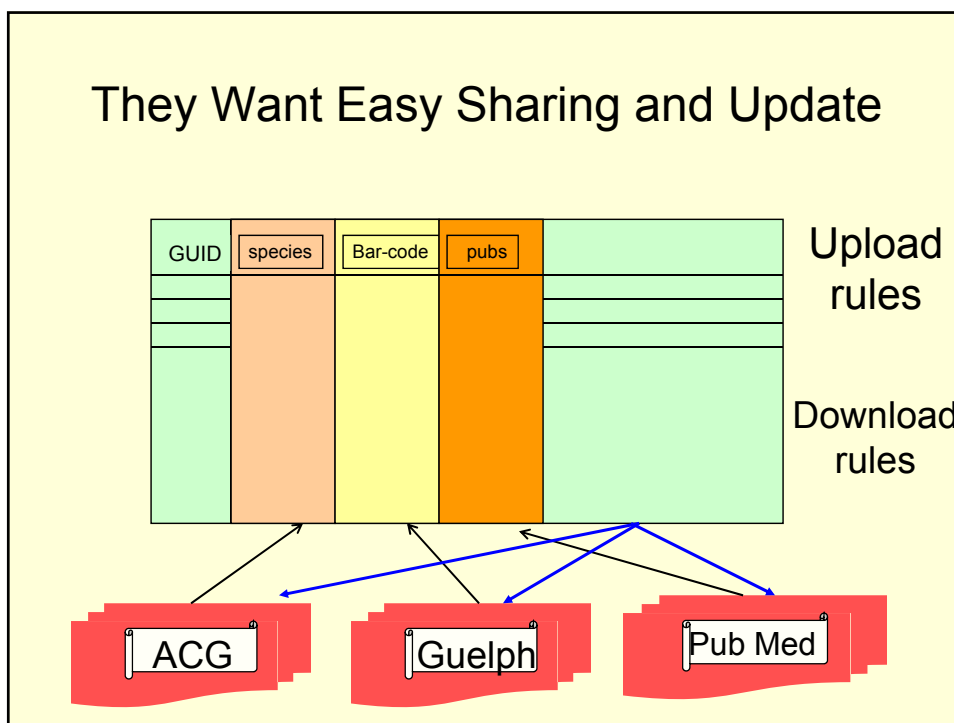
Cutting Off a Leg



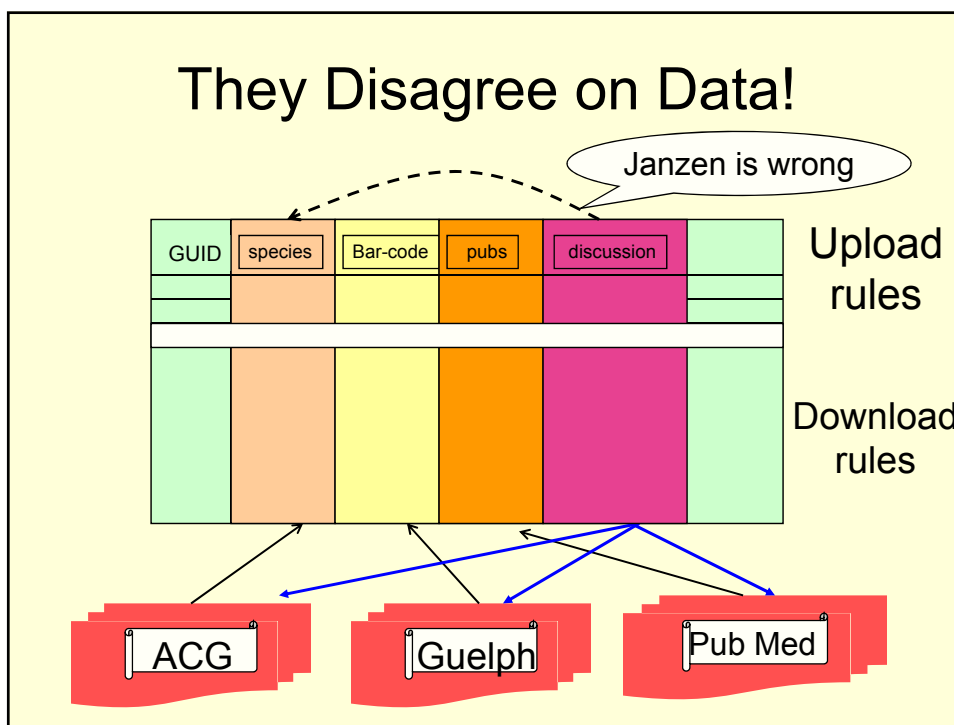
I Created a Row in a Database!



They Want Easy Sharing and Update



They Disagree on Data!

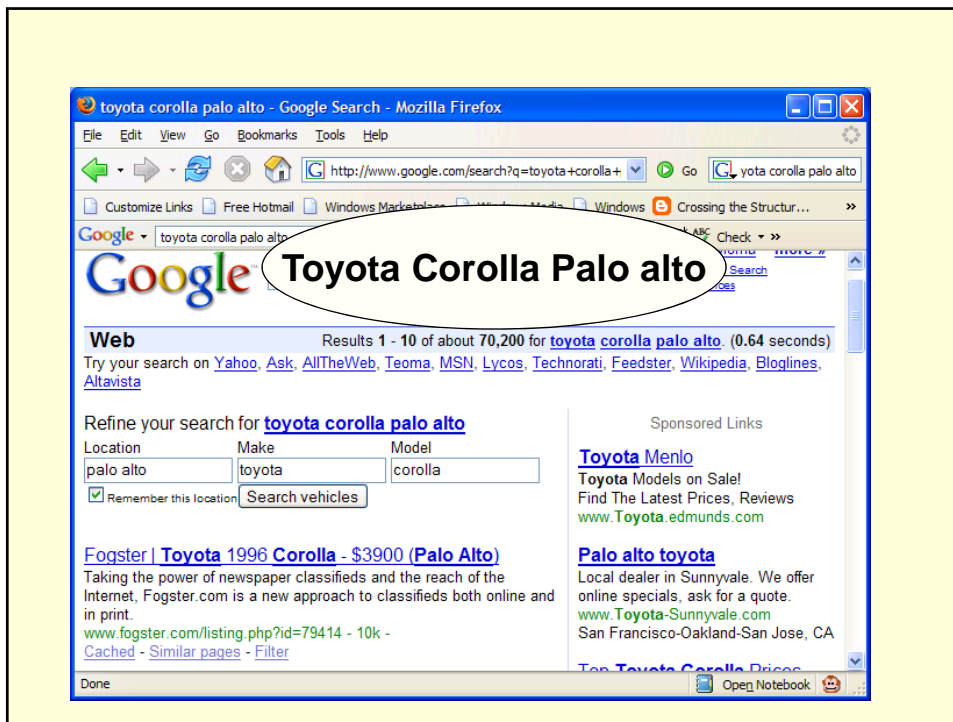


Querying Dataspaces

- We'll talk about the 'how' in a moment, but let's set expectations first.
- Recall that uncertainty is everywhere:
 - Data, mappings, query formulation
- Hence, results need:
 - To be ranked
 - Come with their provenance & explanation
 - See tutorial by Tan & Buneman, SIGMOD 2007.
 - They won't be sets of tuples necessarily.

Query Mechanisms

- Keyword search over structured data
 - BANKS (Mumbai), Xrank (Cornell), Discover (Hristidis and Papakonstantinou), Naga (Kasneji et al.)
- Keywords as a starting point:
 - Find the relevant data source and reformulate the query
 - Examples below
 - Find appropriate structured queries over multiple sources
 - System Q



System Q

[Talukdar et al., VLDB 2008]

Query Keywords
**Protein, Gene,
Disease = "AIDS"**

The Big Question

How do we point a user to the right data when multiple databases, tables are involved and not all databases and tables are of equal value/relevance/quality/authority ?

Learn the Queries to Integrate Data

Schema Graph

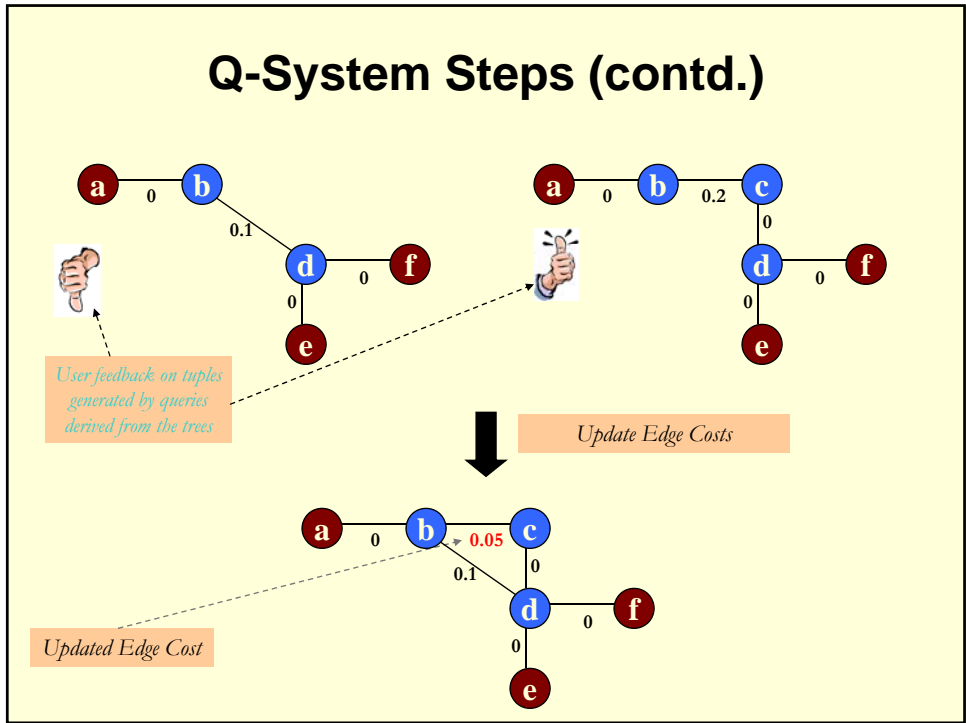
Query Keywords
a, e, f

↓ Find trees connecting red nodes

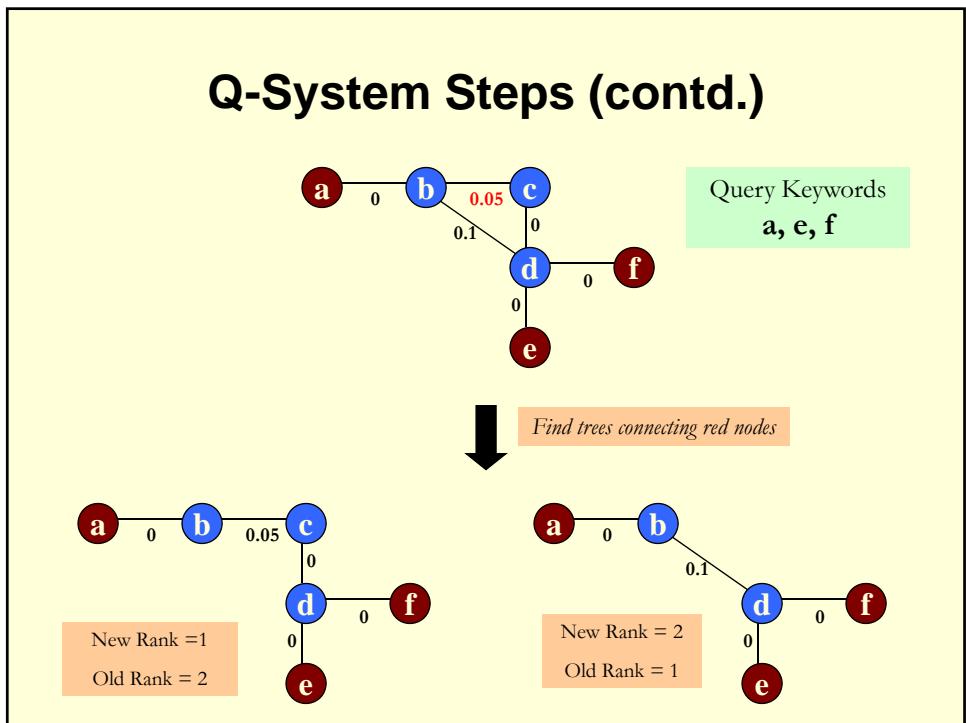
Rank = 1
Cost = 0.1

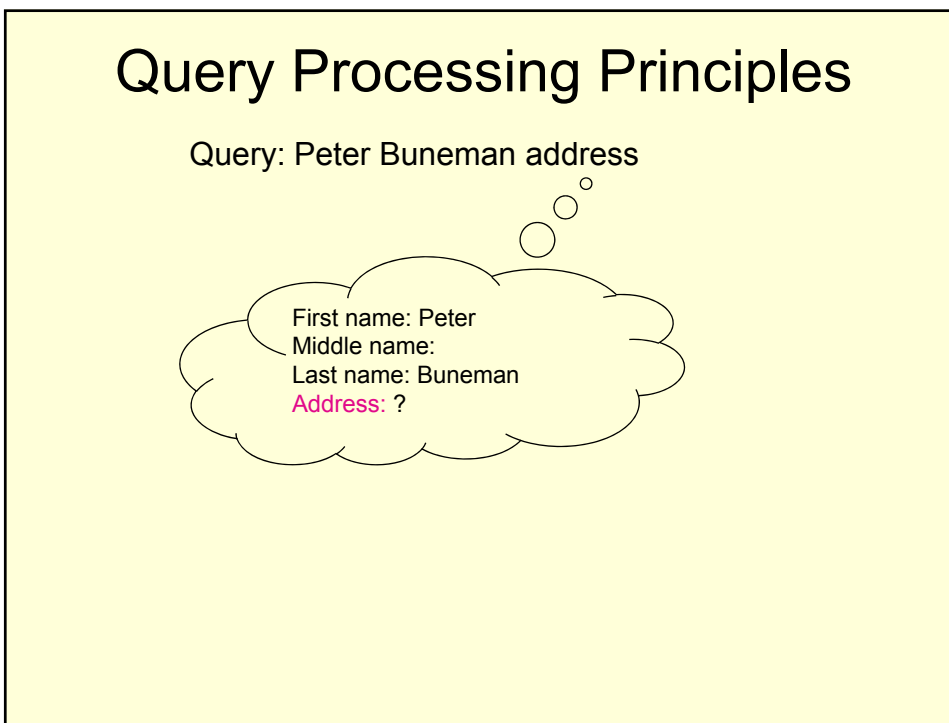
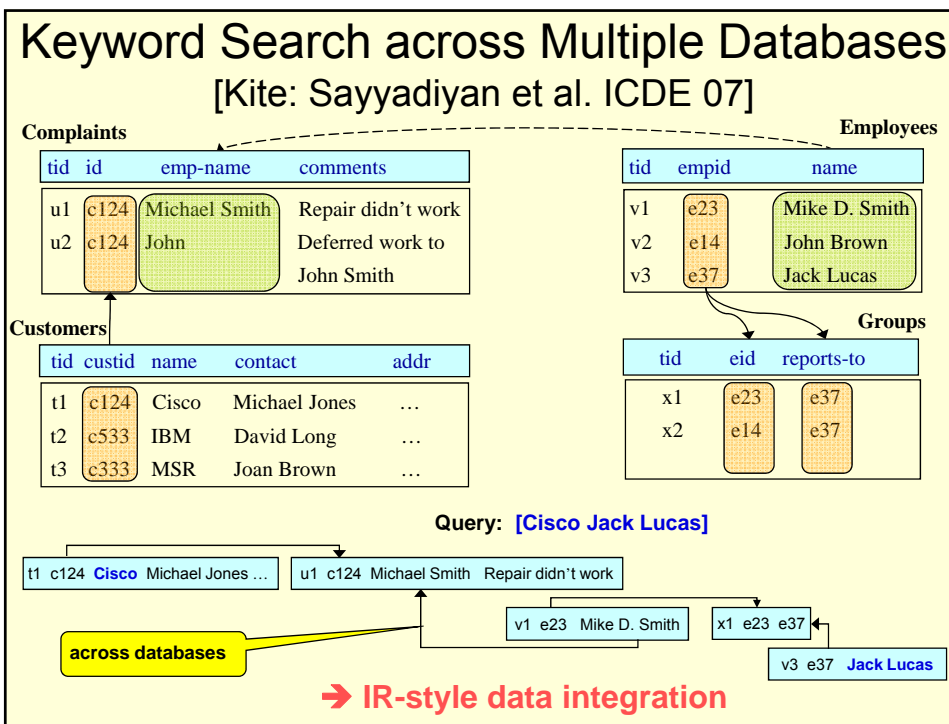
Rank = 2
Cost = 0.2

Q-System Steps (contd.)

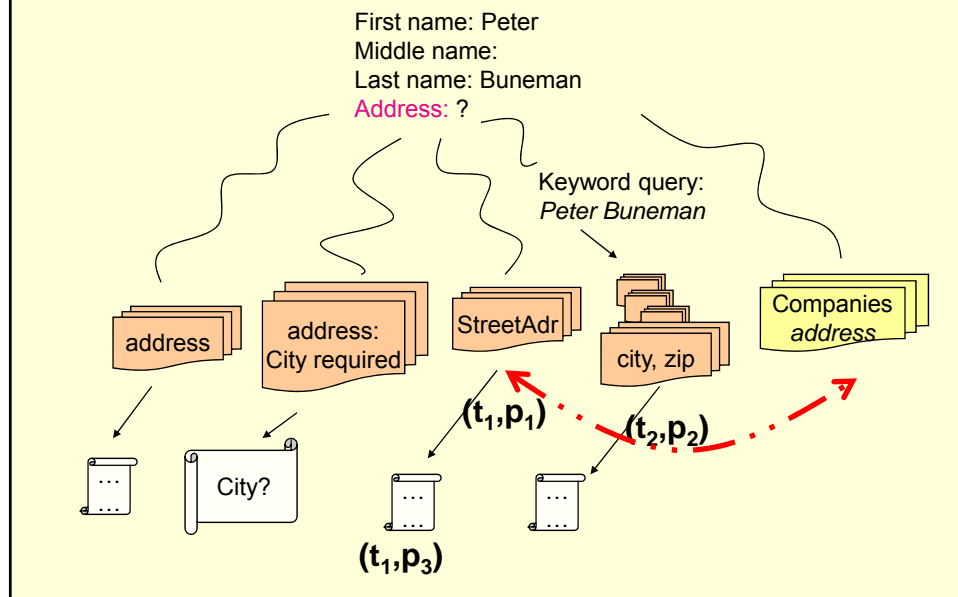


Q-System Steps (contd.)





Query processing as fact gathering



Outline

- ✓ Introduction
- ✓ Dataspace principles through data integration
- **Research challenges on specific dataspace:**
 - Dataspaces on the Web,
 - in Science, and
 - for Personal Information Management

the Presidents of the USA - Enchanted Learning.com - Mozilla Firefox

http://www.enchantedlearning.com/history/us/pres/list.shtml

As a thank-you bonus, site members have access to a banner-ad-free version of the site, with print-friendly pages.

(Already a member? [Click here.](#))

US History

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

African-Americans Artists Explorers of the US Inventors US Presidents US Symbols US States

Enchanted Learning.com

The Presidents of the United States of America

[In the order in which they served](#) [Alphabetical order](#) [Short table of Data](#)

[President's Day Activities](#) [Abraham Lincoln](#)

The President and Vice-President are elected every four years. They must be at least 35 years of age, they must be native-born citizens of the United States, and they must have been residents of the U.S. for at least 14 years. (Also, a person cannot be elected to a third term as President.)

President	Party	Term as President	Vice-President
1. George Washington (1732-1799)	None, Federalist	1789-1797	John Adams
2. John Adams (1735-1826)	Federalist	1797-1801	Thomas Jefferson
3. Thomas Jefferson (1743-1826)	Democratic-Republican	1801-1809	Aaron Burr, George Clinton
4. James Madison (1751-1836)	Democratic-Republican	1809-1817	George Clinton, Elbridge Gerry
5. James Monroe (1758-1831)	Democratic-Republican	1817-1825	Daniel Tompkins
6. John Quincy Adams (1767-1848)	Democratic-Republican	1825-1829	John Calhoun
7. Andrew Jackson (1767-1845)	Democrat	1829-1837	John Calhoun, Martin van Buren
8. Martin van Buren (1782-1862)	Democrat	1837-1841	Richard Johnson
9. William H. Harrison (1773-1841)	Whig	1841	John Tyler
10. John Tyler (1790-1862)	Whig	1841-1845	
11. James K. Polk (1795-1849)	Democrat	1845-1849	George Dallas
12. Zachary Taylor (1784-1850)			
13. Millard Fillmore (1800-1874)			
14. Franklin Pierce (1804-1869)			
15. James Buchanan (1791-1868)	Democrat	1857-1861	John Breckinridge

See next session -- Cafarella et al.

Dataspaces on the Web

- The Deep Web (yesterday, Madhavan et al.):
 - Millions of forms.
- Main challenges:
 - The domain of everything
 - The context of the data carries semantics
 - Need to live with the rest of web data
- Opportunities:
 - Scale: stuff you can do with millions of schemas, forms

Issues in Science Dataspaces

- Concepts are still gelling, or have multiple abstractions
 - E.g., Gene
 - Coding region of a chromosome
 - Particular transcription and splicing of a region
 - Particular variant of the region
 - Product (usu. protein) coded by the region
- Whether they should be treated the same can depend on task or even query
- Makes schemas complex

Science DS Issues, Cont.

Identification is hard

- No common identification scheme yet
 - Hsp10, HSP10, CPN10, Yor020p, ch10_yeast
[Jagadish, Chapman+ SIGMOD 07]
- Comparisons are on complex structures
 - Sequence, molecule, 3-D structure
- Slight variants are different entities in rw
 - Gene homologs

On-the-fly matching difficult

Want to reuse manual work

Scientific DS Issues, cont.

Complex schemas make query hard

Michigan Molecular Interactions (MiMI)

[Jayapandian, Chapman+ *Nucleic Acids Res.* 2007]

- Use abstracted schema for overview (and now query)
- Multiple query interfaces: form, XQuery, keyword, MQuery (graphical)
 - But – “same” query gives different answers in different interfaces

The Other “DataSpace”

- What’s the minimum infrastructure for initial transformation, cleaning and exploratory analysis?
- Data sets often too big to replicate, but even fast channels are hard to exploit for on-the-fly combination

[Grossman, Mazzucco *IEEE Comp in Sci & Eng* 2002]

Universal Keys

- Devise one or more domain-specific universal keys
 - Treat data as distributed columns associated with one or more UKs
 - Fast transfer and merge-join on keys; templated transform and display ops
- Later version called *Sector* with more parallelism

[Grossman, U Penn II Workshop 2006]

Supporting Analysis

Scenario: Domain experts who are unfamiliar with schema, need to make equivalence judgments

- None, <1 pack, 1-2packs, >2 packs
- Never smoked, smoker, quit

- GUAVA: GUI as View Apparatus
Query through the data-entry screen
- MultiClass: Save and reuse domain mapping decisions

[Terwilliger, Delcambre+ EDBT Workshops 2006]

Other Science DS Work

- **Multiple Genomes and Meta-genomes**
[Markowitz U Penn II Workshop 06]
Have “coarse annotation” in some components while refining annotation (perhaps even manually) in others
- **Science dataspaces on the Grid**
[Elsayed, Brezany+ DEXA 2006]
- **Ontologies in science dataspaces**
[Ning, Wang ICPCA 2007]

Personal DS Issues

Many territorial entities in your dataspace

- Device boundaries: laptop vs. PDA
- Document boundaries: directory vs. cells
- Server boundaries: files vs. email

Desktop search doesn't solve it all.

Issue: Reconciling References

- References might have small numbers of attributes
- Not a lot of data to train on or analyze
- References evolve
 - People move
 - Documents go through versions (think about your interview talk)

Issue: One-time Query

- Standard information integration often starts by listing frequent queries that are anticipated
- In a personal DS, you might want to ask a query once over a particular combination of sources

“What exam questions do I have that weren’t in the HW, weren’t on the practice exam, weren’t used in class, aren’t in the back of the book, aren’t examples in the book?”

SEMEX: Semantic Exploration

- Extract objects and relationships automatically and cast into a personal information model [Dong, Halevy CIDR 2005]
- Reference reconciliation is critical
 - First: Mike, Last: Carey, Loc: IBM
 - First: Michael, Last: Carey
 - Last: Carey, Email: carey@ibm.com
 - Email: carey@ibm.com, Loc: Almaden

Reference Merging

- Combine references, allow multivalues

```

First: Mike, Last: Carey, Loc: IBM
    ↘
    ↗
First: Michael, Last: Carey
    ↘
    ↗
Last: Carey, email: carey@ibm.com
    ↘
    ↗
Email: carey@ibm.com, Loc: Almaden
  
```

First: {Mike, Michael},
 Last: Carey, Loc: IBM
 First: {Mike, Michael}, Last: Carey,
 Email: carey@ibm.com
 Loc: {IBM, Almaden}

Last: Carey, email: carey@ibm.com
 Loc: Almaden

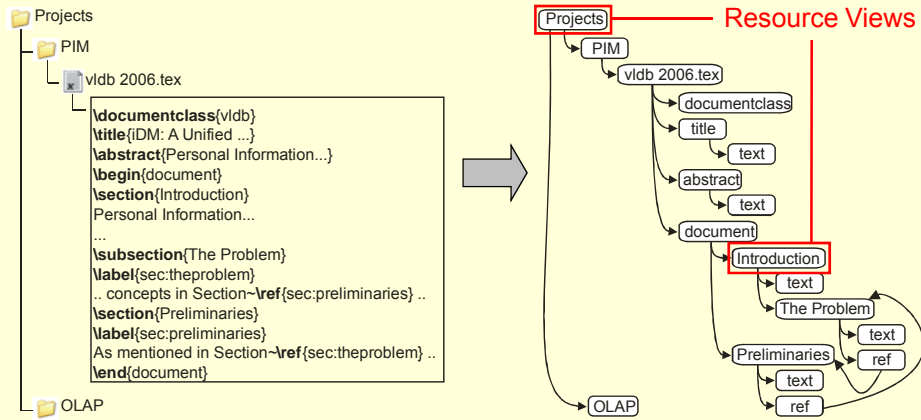
Evolving Objects

- Do fine-grained reconciliation
- Look for evidence to build chains that represent versions of objects.
Emails for Carey from ibm.com don't overlap in time with emails for Carey from bea.com

iMeMex

- You saw this previously in iTrails
[Dittrich, Vaz Salles VLDB 06]
- Try to overcome the document boundary
Why is the file-system directory hierarchy different than the element hierarchy in an XML document?
- iMeMex Data Model (iDM)

iDM Example



Courtesy Jens Dittrich and Marcos Vaz Salles

Ask Us Questions ...

... or straighten us out

Backup Slides

And Extras

Query Answering Semantics

- **Input:**
 - Source **S**, query **Q**
 - P-med-schema $\mathbf{M} = \{ (M_1, Pr(M_1)), \dots, (M_p, Pr(M_p)) \}$
 - P-mappings $\mathbf{pM} = \{ pM(M_1), \dots, pM(M_p) \}$
- Output probability of tuple **t**:
 - $p = \sum \Pr(t | M_i) * Pr(M_i)$

Query Answering

S1

name	hPhone	oPhone	hAddr	oAddr
Alice	123-4567	765-4321	123, A Ave.	456, B Ave

Q

```
SELECT name, phone, address
FROM Med-S
```

Answers

Tuple	Probability
('Alice', '123-4567', '123 A Ave.')	0.34
('Alice', '765-4321', '456 B Ave.')	0.34
('Alice', '765-4321', '123 A Ave.')	0.16
('Alice', '123-4567', '456 B Ave.')	0.16

85

Expressive Power of P-Med-Schema v.s. P-Mapping

Theorem 1. For one-to-many mappings:
 (p-med-schema + p-mappings)
 = (mediated schema + p-mapping)
 > (p-med-schema + mappings)

Theorem 2. When restricted to one-to-one mappings:
 (p-med-schema + p-mappings)
 = (p-med-schema + mappings)
 > (mediated schema + p-mapping)

86

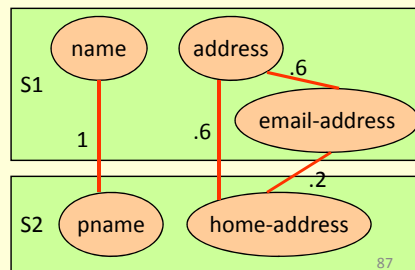
Creation: 1) Creating a Single Med-Schema

- **Input:** Single-table source schemas S_1, \dots, S_n

Output: Single-table mediated schema M

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. Remove edges with weight below τ (e.g., $\tau=.5$)
4. Each connected component is a cluster



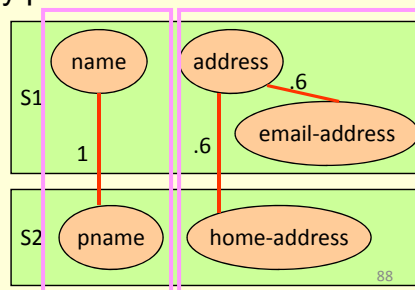
Creation: 1) Creating a Single Med-Schema

- **Input:** Single-table source schemas S_1, \dots, S_n

Output: Single-table mediated schema M

- **Algorithm**

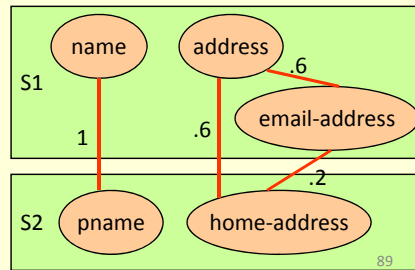
1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. Remove edges with weight below τ (e.g., $\tau=.5$)
4. Each connected component is a cluster



Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

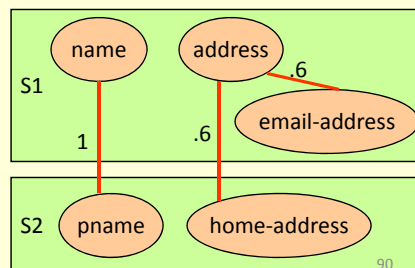
1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
 - (weight $\geq \tau + \epsilon$) \rightarrow retain
 - (weight $< \tau - \epsilon$) \rightarrow drop
 - ($\tau - \epsilon \leq \text{weight} < \tau + \epsilon$) \rightarrow uncertain edge
 (e.g., $\tau = .6$, $\epsilon = .2$)
4. Clustering for each combo of including/excluding uncertain edges



Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
 - (weight $\geq \tau + \epsilon$) \rightarrow retain
 - (weight $< \tau - \epsilon$) \rightarrow drop
 - ($\tau - \epsilon \leq \text{weight} < \tau + \epsilon$) \rightarrow uncertain edge
 (e.g., $\tau = .6$, $\epsilon = .2$)
4. Clustering for each combo of including/excluding uncertain edges



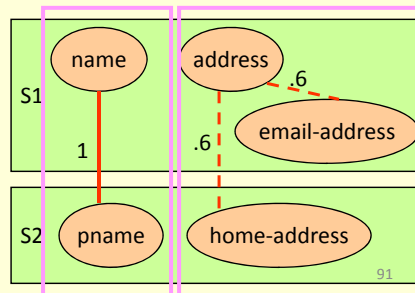
Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
 - (weight $\geq \tau + \epsilon$) \rightarrow retain
 - (weight $< \tau - \epsilon$) \rightarrow drop
 - ($\tau - \epsilon \leq \text{weight} < \tau + \epsilon$) \rightarrow uncertain edge

(e.g., $\tau = .6$, $\epsilon = .2$)

4. Clustering for each combo of including/excluding uncertain edges



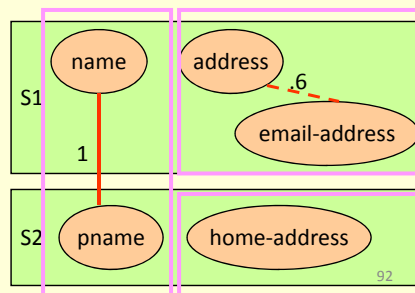
Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
 - (weight $\geq \tau + \epsilon$) \rightarrow retain
 - (weight $< \tau - \epsilon$) \rightarrow drop
 - ($\tau - \epsilon \leq \text{weight} < \tau + \epsilon$) \rightarrow uncertain edge

(e.g., $\tau = .6$, $\epsilon = .2$)

4. Clustering for each combo of including/excluding uncertain edges



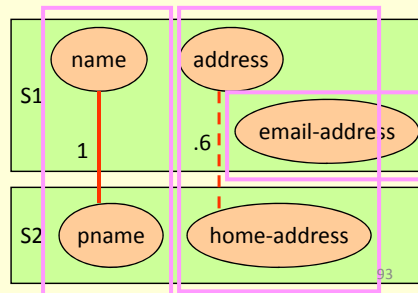
Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
 - (weight $\geq \tau + \epsilon$) \rightarrow retain
 - (weight $< \tau - \epsilon$) \rightarrow drop
 - ($\tau - \epsilon \leq \text{weight} < \tau + \epsilon$) \rightarrow uncertain edge

(e.g., $\tau = .6$, $\epsilon = .2$)

4. Clustering for each combo of including/excluding uncertain edges



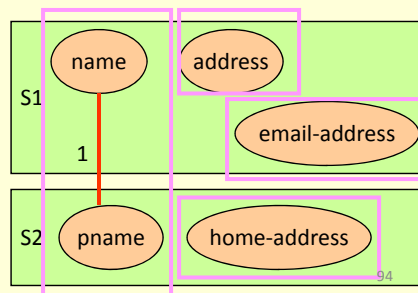
Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
 - (weight $\geq \tau + \epsilon$) \rightarrow retain
 - (weight $< \tau - \epsilon$) \rightarrow drop
 - ($\tau - \epsilon \leq \text{weight} < \tau + \epsilon$) \rightarrow uncertain edge

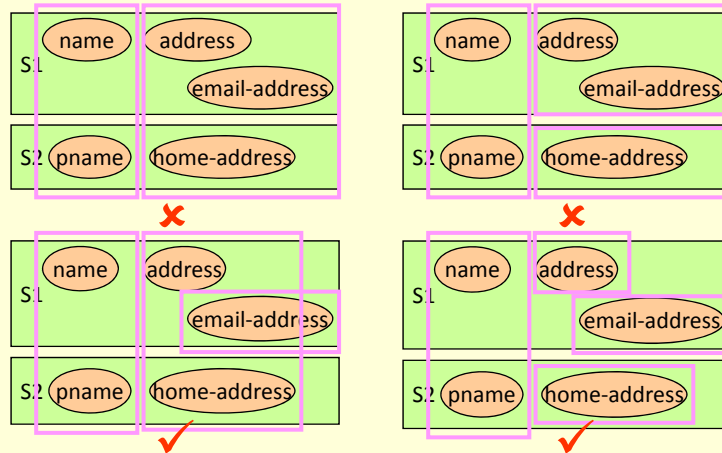
(e.g., $\tau = .6$, $\epsilon = .2$)

4. Clustering for each combo of including/excluding uncertain edges



Creation: 3) Computing Probabilities

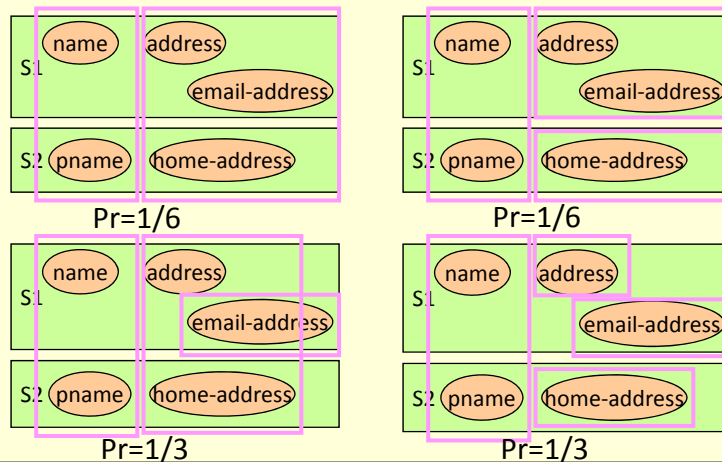
- Mediated schema M and source S are *consistent* if no two attributes of S are grouped into same cluster in M



95

Creation: 3) Computing Probabilities

- Assign probabilities to each M proportional to the number of sources it is consistent with.



96