Dataspaces: The Tutorial

Day 2

Alon Halevy, David Maier
VLDB 2008
Auckland, New Zealand

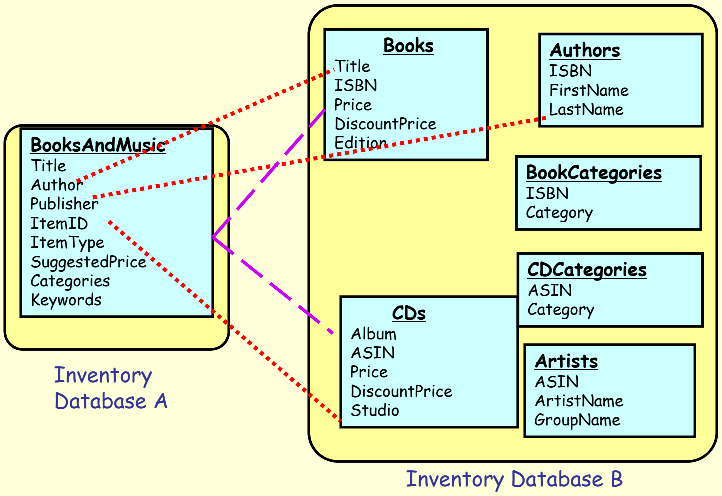
Outline

- ✓ 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



BooksAndMusic
Title
Author
Publisher
ItemID
ItemType
SuggestedPrice
Categories
Keywords

Inventory Database A

Books
Title
ISBN
Price
DiscountPrice
Edition

Authors
ISBN
FirstName
LastName

BookCategories
ISBN
Category

CDCategories
ASIN
Category

CDs
Album
ASIN
Price
DiscountPrice
Studio

Artists
ASIN
ArtistName
GroupName

Inventory Database B

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

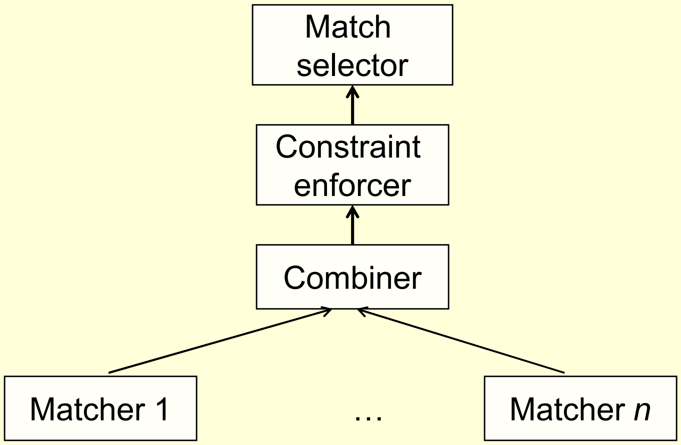
Sub-Outline

- ✓ 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

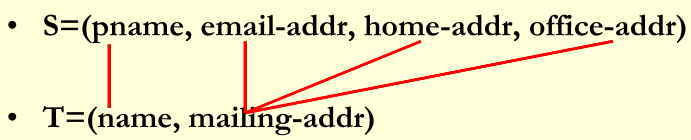
- ✓ What are schema matches and mappings?
  - ✓ Why is it so hard to create them?
- ✓ Automatic techniques for creating them
- Probabilistic schema mappings
- Probabilistic mediated schemas
- Trails: mapping hints

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]

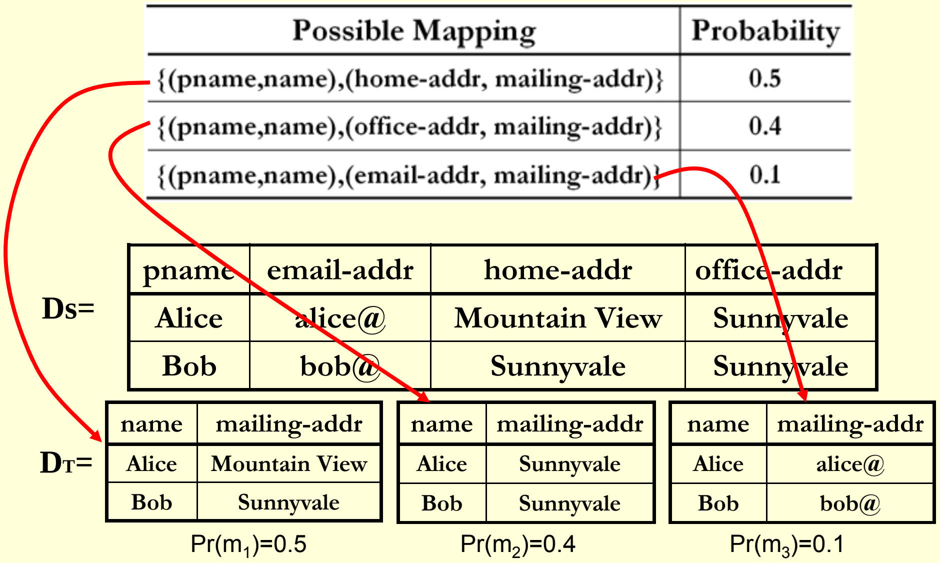


- S=(pname, email-addr, home-addr, office-addr)
- T=(name, mailing-addr)

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

Ds=

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

DT=

name	mailing-addr
Alice	Mountain View
Bob	Sunnyvale

$Pr(m_1)=0.5$

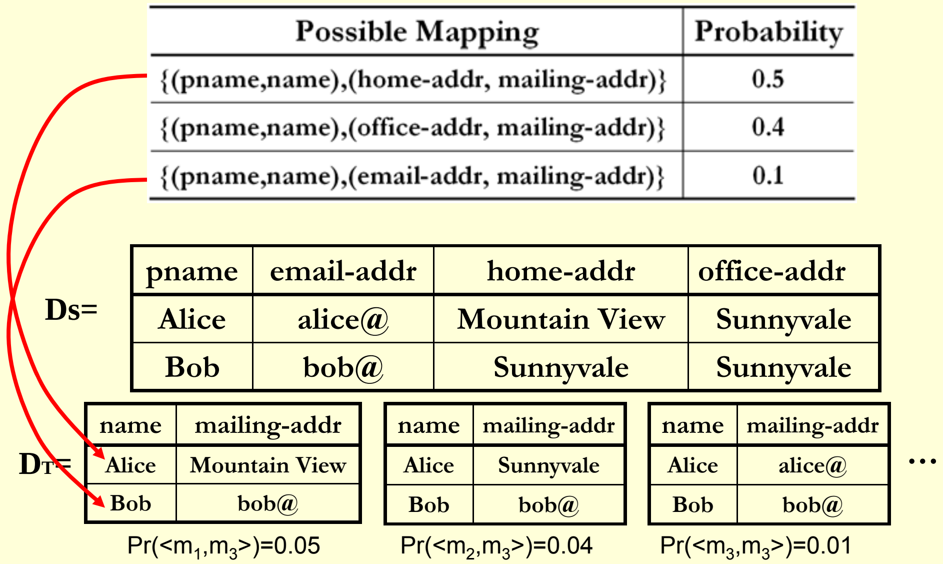
name	mailing-addr
Alice	Sunnyvale
Bob	Sunnyvale

$Pr(m_2)=0.4$

name	mailing-addr
Alice	alice@
Bob	bob@

$Pr(m_3)=0.1$

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

Ds=

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

DT=

name	mailing-addr
Alice	Mountain View
Bob	bob@

Pr(<m_1,m_3>)=0.05

name	mailing-addr
Alice	Sunnyvale
Bob	bob@

Pr(<m_2,m_3>)=0.04

name	mailing-addr
Alice	alice@
Bob	bob@

Pr(<m_3,m_3>)=0.01

…

Complexity of Query Answering

	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

Works for compressed representations of mappings too.

Results extend to more complex mapping languages.

PTIME for important special cases

Sub-Outline

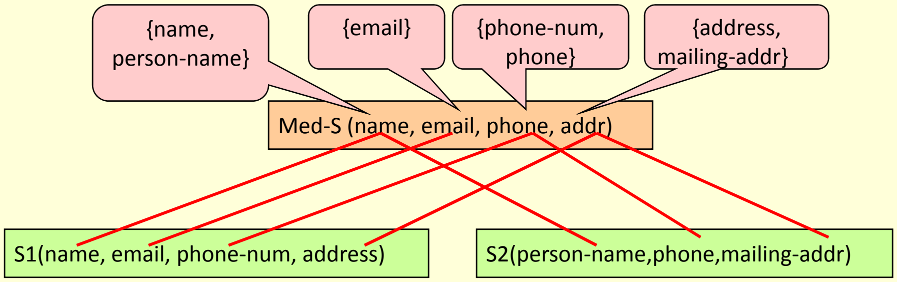
- ✓ What are schema matches and mappings?
  - ✓ Why is it so hard to create them?
- ✓ Automatic techniques for creating them
- ✓ Probabilistic schema mappings
- Probabilistic mediated schemas
- Trails: mapping hints

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



{name, person-name}

{email}

{phone-num, phone}

{address, mailing-addr}

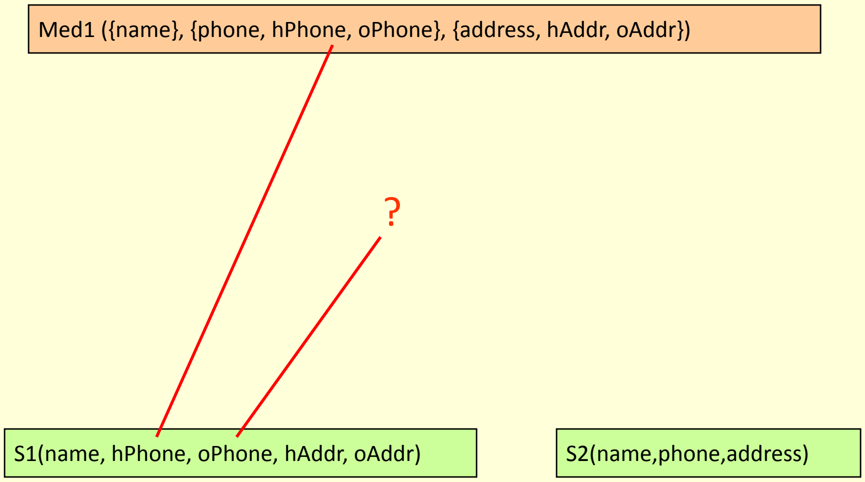
Med-S (name, email, phone, addr)

S1(name, email, phone-num, address)

S2(person-name,phone,mailing-addr)

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

Why *Probabilistic* Mediated Schema?



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

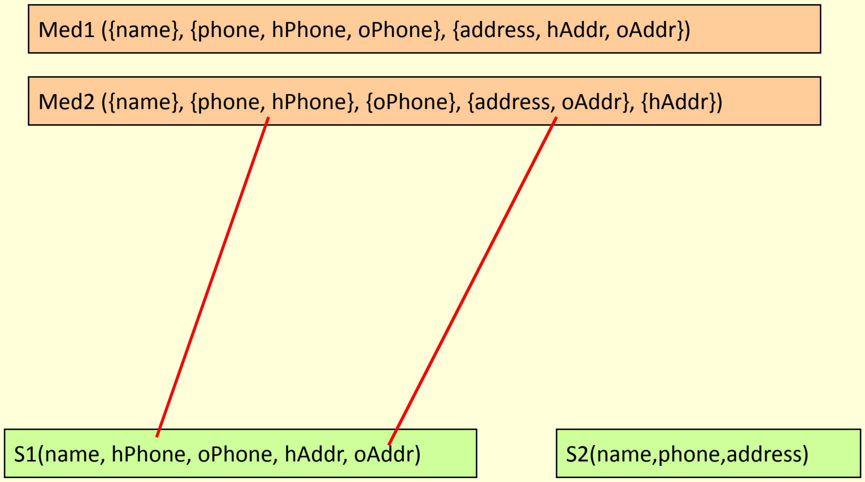
?

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

S2(name,phone,address)

Q: SELECT name, hPhone, oPhone FROM Med

Why *Probabilistic* Mediated Schema?



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

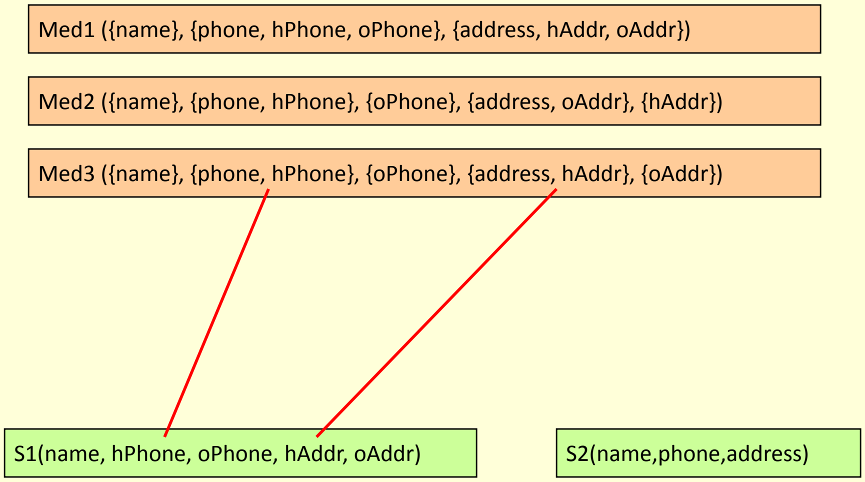
Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

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

S2(name,phone,address)

Q: SELECT name, phone, address FROM Med

Why *Probabilistic* Mediated Schema?



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

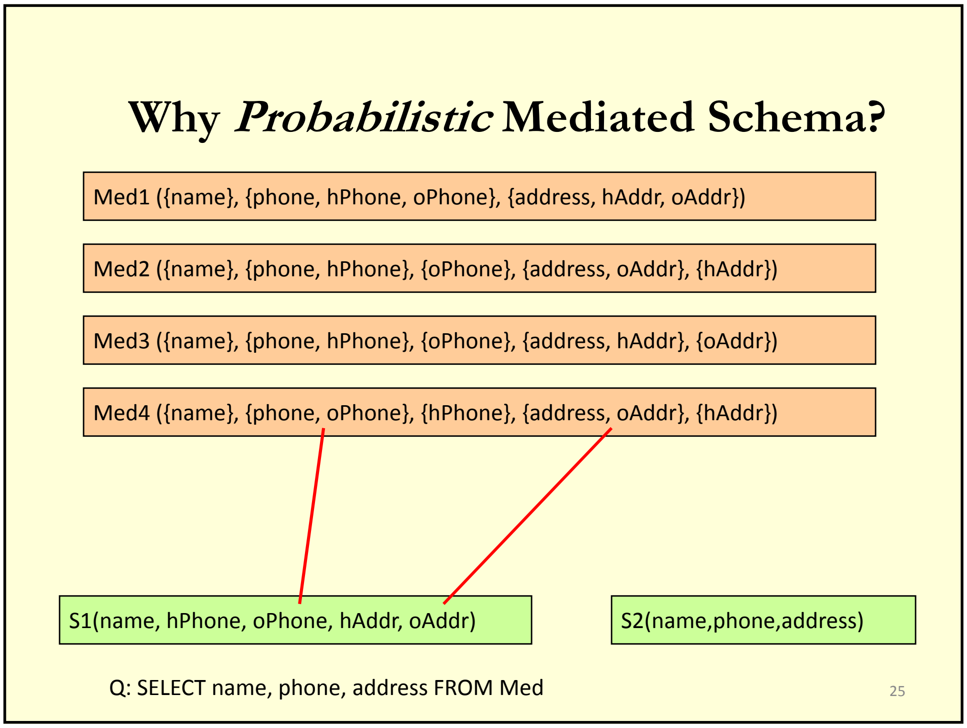
Med2 ({name}, {phone, hPhone}, {oPhone}, {address, oAddr}, {hAddr})

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



Why *Probabilistic* Mediated Schema?

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

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

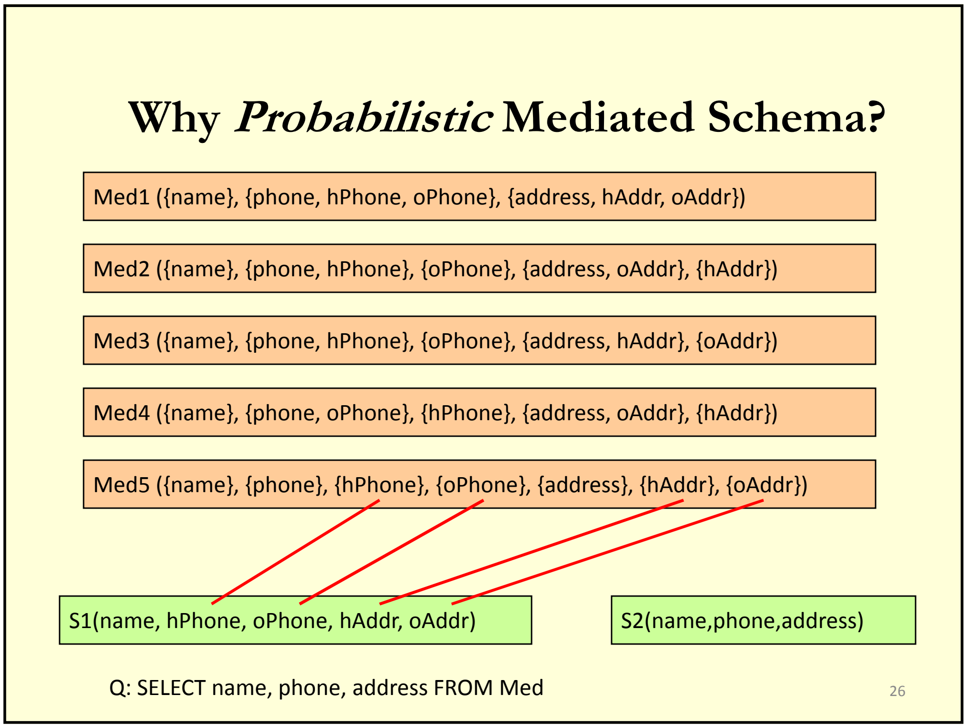
Med3 ({name}, {phone, hPhone}, {oPhone}, {address, hAddr}, {oAddr})

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



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})

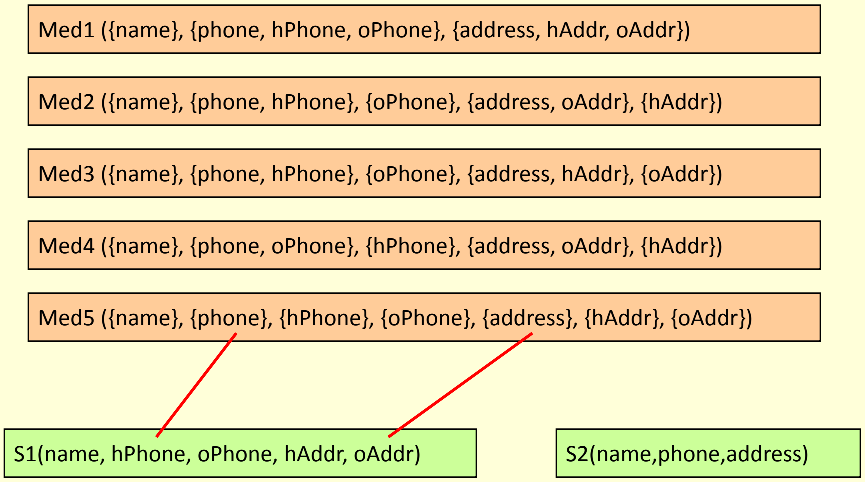
Med5 ({name}, {phone}, {hPhone}, {oPhone}, {address}, {hAddr}, {oAddr})

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

S2(name,phone,address)

Q: SELECT name, phone, address FROM Med

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

Probabilistic Mediated Schema

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

Pr=.5

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

Pr=.5

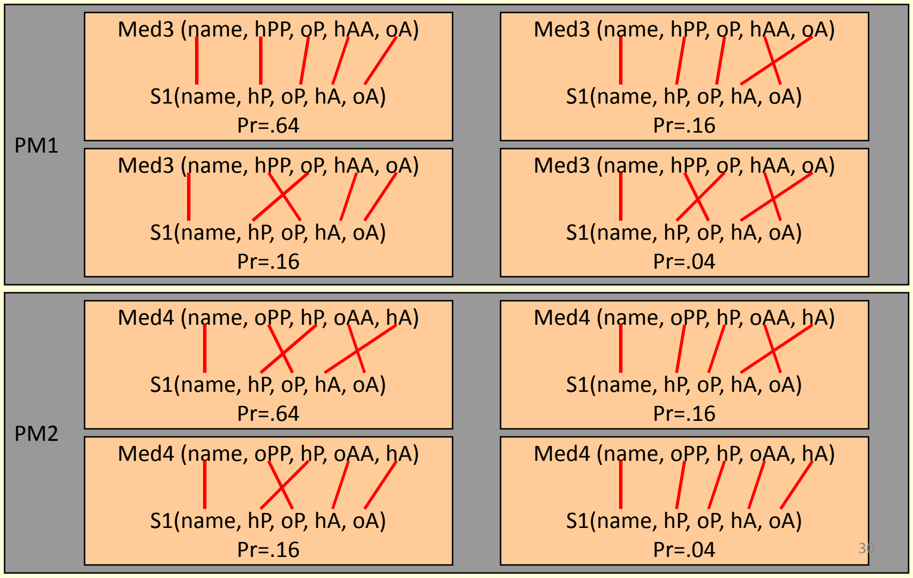
S1(name, hPhone, oPhone, hAddr, oAddr)

S2(name,phone,address)

Probabilistic Mediated Schema

- A p-med-schema is a set
  M = { $(M_1, Pr(M_1))$, …, $(M_l, Pr(M_l))$} where
  - M_i is a med-schema; i≠j => $M_i \neq M_j$
  - $Pr(M_i) \in (0,1]$; $\Sigma Pr(M_i) = 1$

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



PM1

Med3 (name, hPP, oP, hAA, oA)
S1(name, hP, oP, hA, oA)
Pr=.64

Med3 (name, hPP, oP, hAA, oA)
S1(name, hP, oP, hA, oA)
Pr=.16

Med3 (name, hPP, oP, hAA, oA)
S1(name, hP, oP, hA, oA)
Pr=.16

Med3 (name, hPP, oP, hAA, oA)
S1(name, hP, oP, hA, oA)
Pr=.04

PM2

Med4 (name, oPP, hP, oAA, hA)
S1(name, hP, oP, hA, oA)
Pr=.64

Med4 (name, oPP, hP, oAA, hA)
S1(name, hP, oP, hA, oA)
Pr=.16

Med4 (name, oPP, hP, oAA, hA)
S1(name, hP, oP, hA, oA)
Pr=.16

Med4 (name, oPP, hP, oAA, hA)
S1(name, hP, oP, hA, oA)
Pr=.04

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

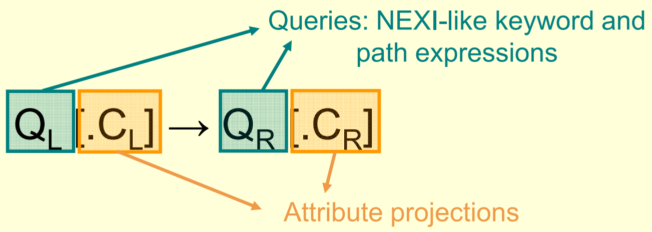
✓ What are schema matches and mappings?
  ✓ Why is it so hard to create them?
✓ Automatic techniques for creating them
✓ Probabilistic schema mappings
✓ Probabilistic mediated schemas
- Trails: mapping hints

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



Queries: NEXI-like keyword and path expressions

$Q_L\ [.C_L] \rightarrow Q_R\ [.C_R]$

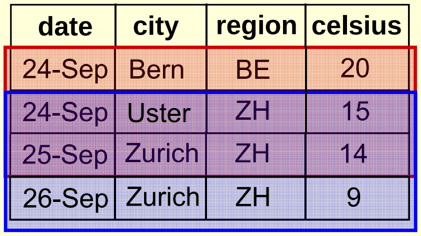
Attribute projections

- 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"

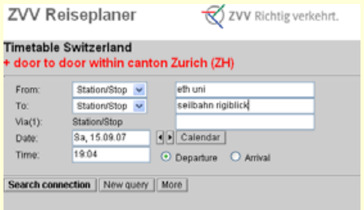
```
global warming →
   //Temperatures/*[celsius > 10]
```

- **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`



Timetable Switzerland

+ door to door within canton Zurich (ZH)

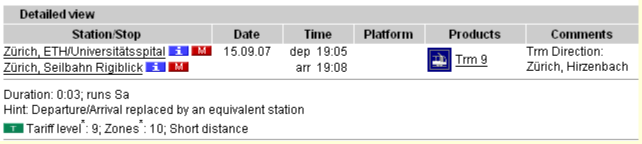
From: Station/Stop eth uni

To: Station/Stop seilbahn rigiblick

Via(1): Station/Stop

Date: Sa, 15.09.07

Time: 19:04 ● Departure ○ Arrival



Detailed view

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

Duration: 0:03; runs Sa

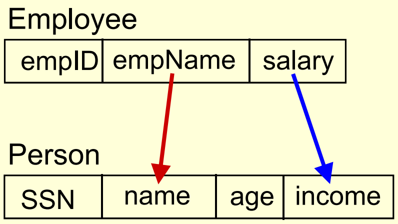
Hint: Departure/Arrival replaced by an equivalent station

T Tariff level*: 9; Zones*: 10; Short distance

- **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"]
```

Trail Examples: Schema Equivalences



- **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
```

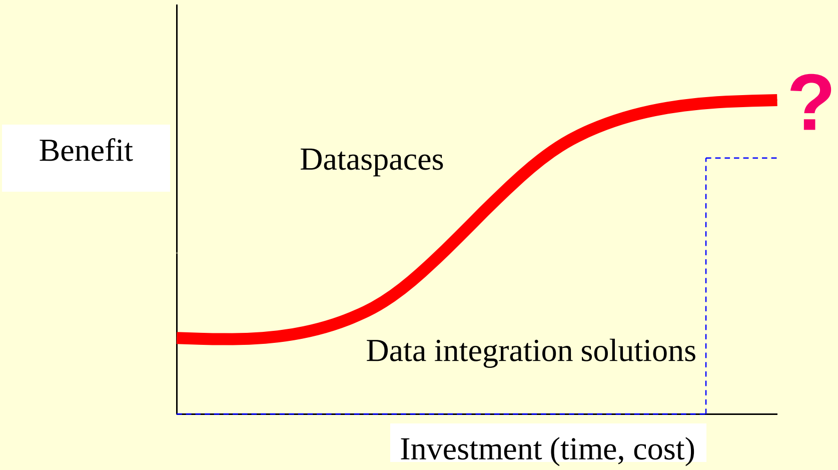
More on Trails

- 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 dataspaces
- Research challenges on specific dataspaces:
  - Science, the desktop, the Web

Getting the Red Curve

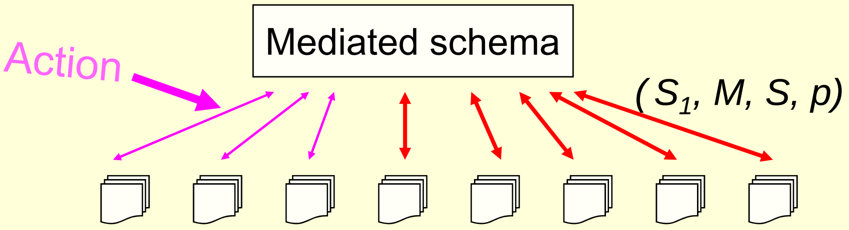


Reusing Human Attention

- 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:

$$\begin{aligned} \mathit{Benefit}(\text{match } m_j) = {} & \boxed{\begin{aligned} & \mathit{Utility}(m_j \text{ confirmed})(p_{\text{correct}}) + \\ & \mathit{Utility}(m_j \text{ disconfirmed})(1 - p_{\text{correct}}) \end{aligned}} - \\ & \boxed{\mathit{Utility}(\text{without asking})} \end{aligned}$$

Utility of a Dataspace

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

$$\boxed{U(D,M) = \underbrace{\sum_{(Q_i,w_i)\in W}}_{\text{Look at all queries in the workload}} \underbrace{r(Q_i,D,M)}_{\text{Query result quality (e.g. precision/recall)}} \underbrace{w_i}_{\text{Query importance}}}$$

Challenges

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

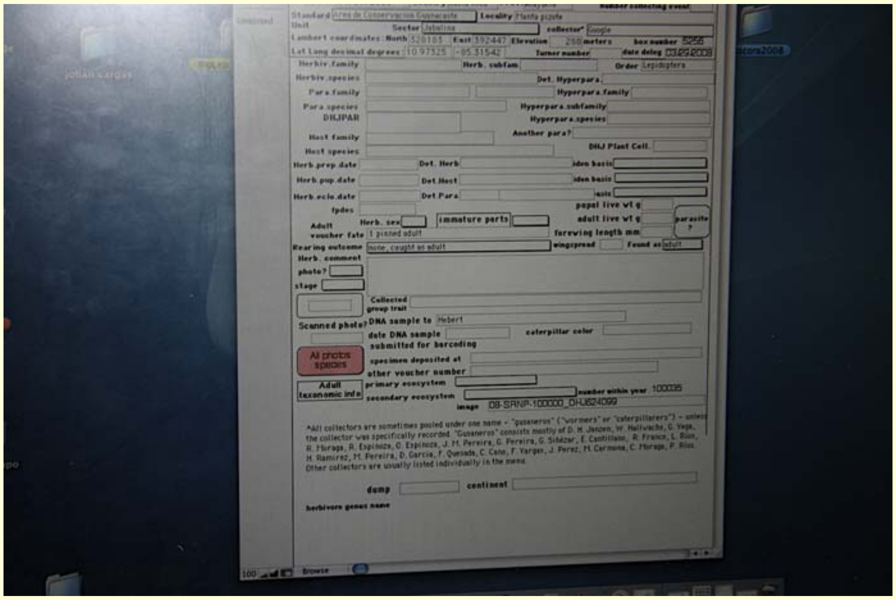
Outline

- ✓ 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 dataspaces:
  - Science, the desktop, the Web

Looking for data management problems
in the rainforest in Costa Rica

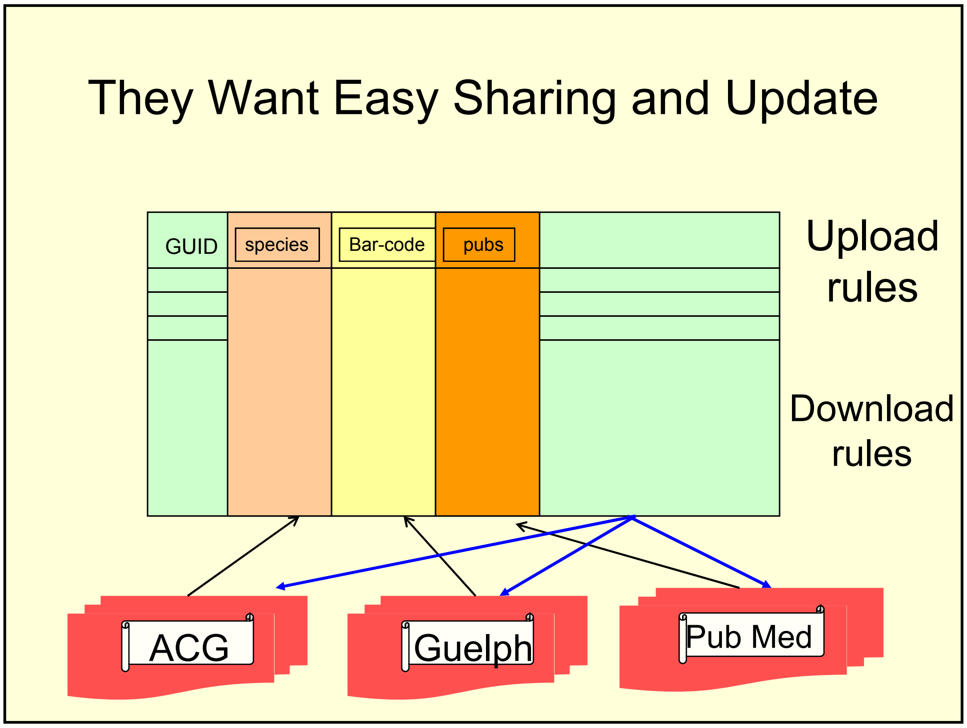
Cutting Off a Leg

I Created a Row in a Database!



Standard Unit: Area de Conservacion Guanacaste
Locality: Planta pizote
Sector: Jabalina
collector*: Google
Lambert coordinates: North 320183 East 392447 Elevation 288 meters
box number 5256
Lat Long decimal degrees: 10.97325 -85.31542
date delog 03292008
Order: Lepidoptera
Adult voucher fate: 1 pinned adult
Rearing outcome: none, caught as adult
Found as: adult
DNA sample to: Hebert
number within year: 100035
image: 08-SRNP-100000_DHJ624099

*All collectors are sometimes pooled under one name - "gusaneros" ("wormers" or "caterpillarers") - unless the collector was specifically recorded. "Gusaneros" consists mostly of D. H. Janzen, W. Hallwachs, G. Vega, R. Moraga, R. Espinoza, O. Espinoza, J. M. Pereira, G. Pereira, G. Sihézar, E. Cantillano, R. Franco, L. Rios, H. Ramirez, M. Pereira, D. Garcia, F. Quesada, C. Cano, F. Vargas, J. Perez, M. Carmona, C. Moraga, F. Rios. Other collectors are usually listed individually in the menu.



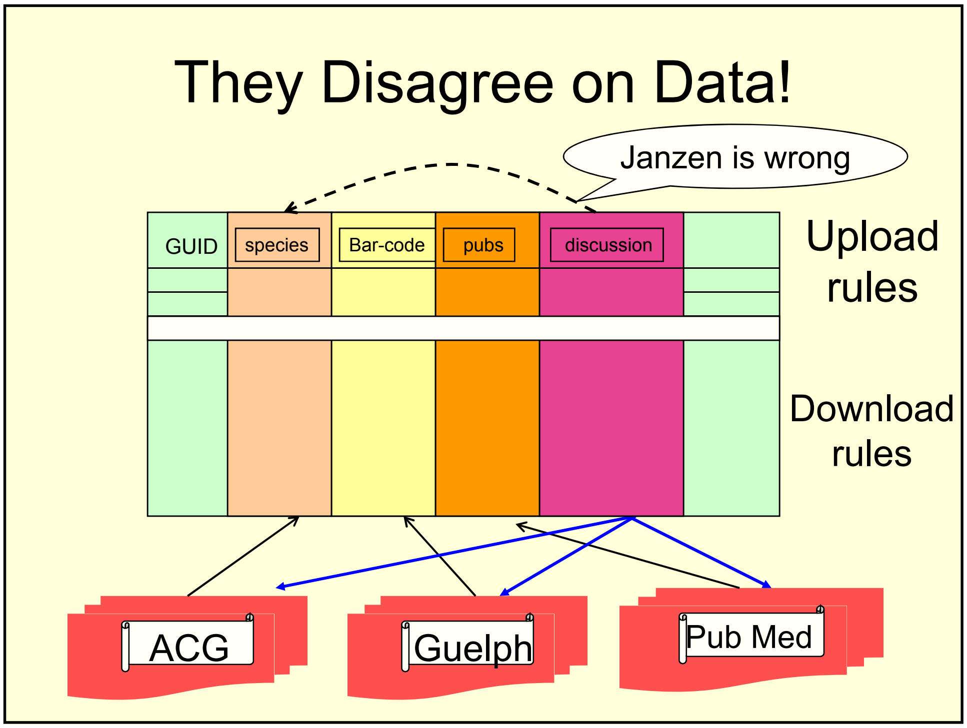
They Want Easy Sharing and Update

GUID
species
Bar-code
pubs

Upload rules

Download rules

ACG
Guelph
Pub Med



They Disagree on Data!

Janzen is wrong

GUID
species
Bar-code
pubs
discussion

Upload rules

Download rules

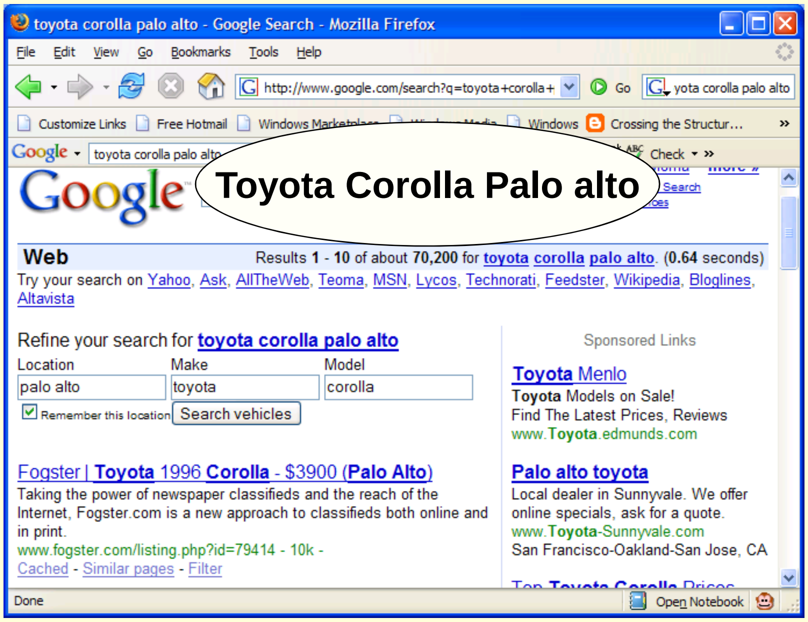
ACG
Guelph
Pub Med

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 (Kasneci 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



Toyota Corolla Palo alto

Web Results **1** - **10** of about **70,200** for toyota corolla palo alto. (**0.64** seconds)

Try your search on Yahoo, Ask, AllTheWeb, Teoma, MSN, Lycos, Technorati, Feedster, Wikipedia, Bloglines, Altavista

Refine your search for **toyota corolla palo alto**

Location	Make	Model
palo alto	toyota	corolla

☑ Remember this location Search vehicles

Fogster | **Toyota** 1996 **Corolla** - $3900 (**Palo Alto**)
Taking the power of newspaper classifieds and the reach of the Internet, Fogster.com is a new approach to classifieds both online and in print.
www.fogster.com/listing.php?id=79414 - 10k -
Cached - Similar pages - Filter

Sponsored Links

Toyota Menlo
Toyota Models on Sale!
Find The Latest Prices, Reviews
www.**Toyota**.edmunds.com

Palo alto toyota
Local dealer in Sunnyvale. We offer online specials, ask for a quote.
www.**Toyota**-Sunnyvale.com
San Francisco-Oakland-San Jose, CA

Honda Palo alto

Web ,000 for **honda** palo alto. (**0.23** seconds)

Try your search on Yahoo, Ask, AllTheWeb, Teoma, MSN, Lycos, Technorati, Feedster, Wikipedia, Bloglines, Altavista

Sponsored Links

Honda Palo Alto
www.AndersonDirect.com/ Low prices, no hassle, great deals at Anderson **Honda** in **Palo Alto**, CA.

Honda of Stevens Creek
www.**honda**ofstevenscreek.com Our Prices & Service Can't Be Beat. Your Discount **Honda** Store & More!

Local results for **honda** near **Palo Alto, CA**
Anderson **Honda** - 1.4 miles NE - 1766 Embarcadero Rd, Palo Alto, 94303 - (650) 856-6000
Honda Service - 0.9 miles W - 930 Emerson St, Palo Alto, 94301 - (650) 327-3061
Honda Small Car Shop - 1.0 miles W - 111 Homer Ave, Palo Alto, 94301 - (650) 329-0666

Anderson **Honda** - New & Certified Used **Honda** Cars Dealer at Palo

Sponsored Links

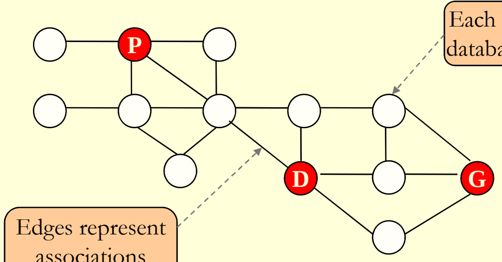
Honda In **Palo Alto**
Instant Price Quote Directly From Anderson **Honda** - (888) 898-4309
Honda.Quote-Form.com

Larry's AutoWorks
When you want it right, you go to the experts. Larry's knows Hondas.
www.autoworks.com
San Francisco-Oakland-San Jose, CA

Palo Alto Honda
Radical Approach to New Car Buying. Competing Bids from Dealers in Area
DealersCompeteYouWin.com

System Q

[Talukdar et al., VLDB 2008]



Each node is a database/table.

Edges represent associations (e.g. cross-ref/mapping)

Query Keywords

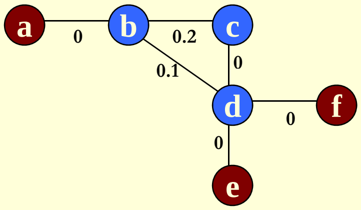
Protein, **G**ene, **D**isease = "AIDS"

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 ?

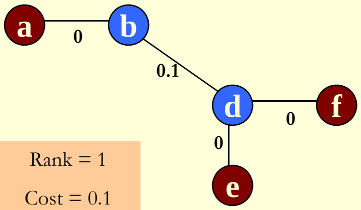
Learn the Queries to Integrate Data

Schema Graph



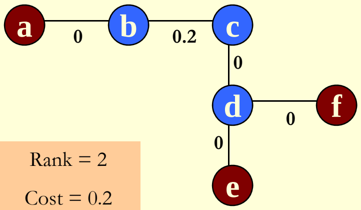
Query Keywords
a, e, f

Find trees connecting red nodes



Rank = 1

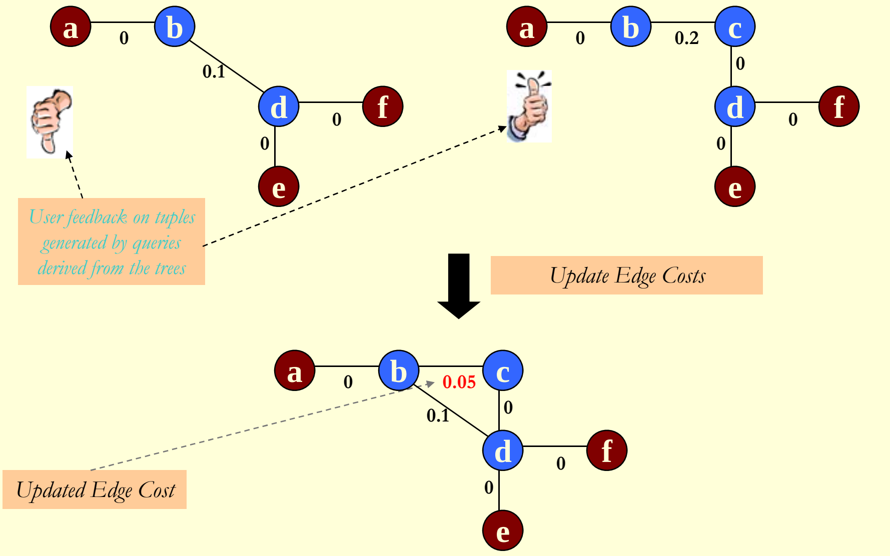
Cost = 0.1



Rank = 2

Cost = 0.2

Q-System Steps (contd.)

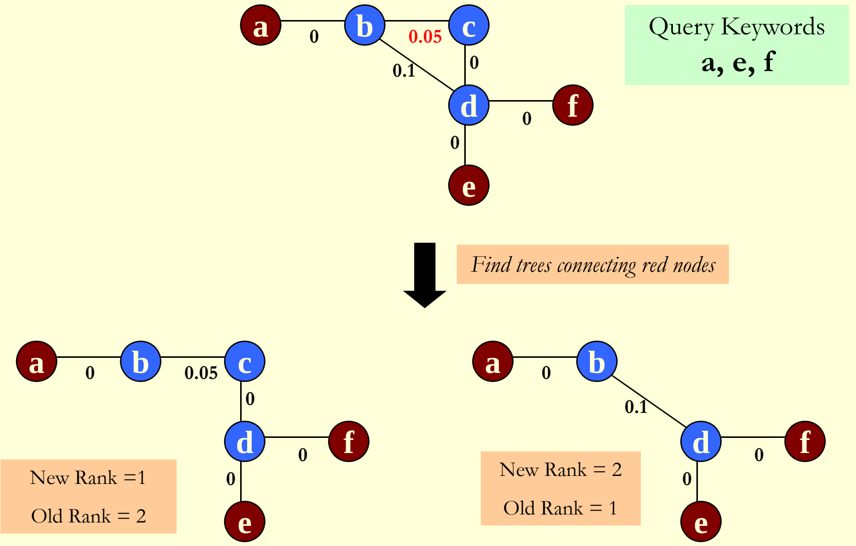


User feedback on tuples generated by queries derived from the trees

Update Edge Costs

Updated Edge Cost

Q-System Steps (contd.)



Query Keywords
a, e, f

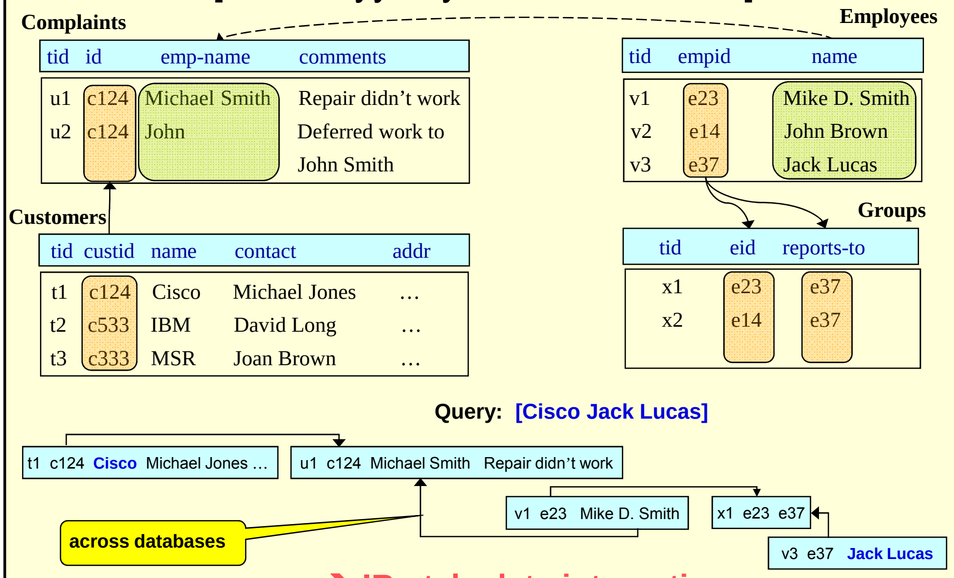
Find trees connecting red nodes

New Rank =1
Old Rank = 2

New Rank = 2
Old Rank = 1

Keyword Search across Multiple Databases

[Kite: Sayyadiyan et al. ICDE 07]



Complaints

tid	id	emp-name	comments
u1	c124	Michael Smith	Repair didn't work
u2	c124	John	Deferred work to John Smith

Employees

tid	empid	name
v1	e23	Mike D. Smith
v2	e14	John Brown
v3	e37	Jack Lucas

Customers

tid	custid	name	contact	addr
t1	c124	Cisco	Michael Jones	…
t2	c533	IBM	David Long	…
t3	c333	MSR	Joan Brown	…

Groups

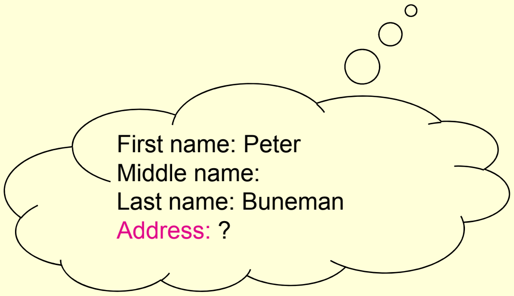
tid	eid	reports-to
x1	e23	e37
x2	e14	e37

Query: **[Cisco Jack Lucas]**

➔ **IR-style data integration**

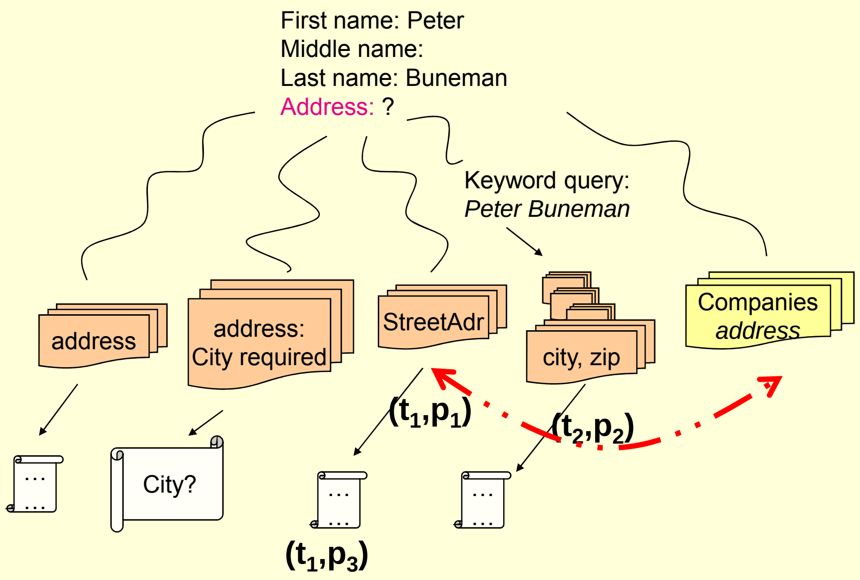
Query Processing Principles

Query: Peter Buneman address



First name: Peter
Middle name:
Last name: Buneman
Address: ?

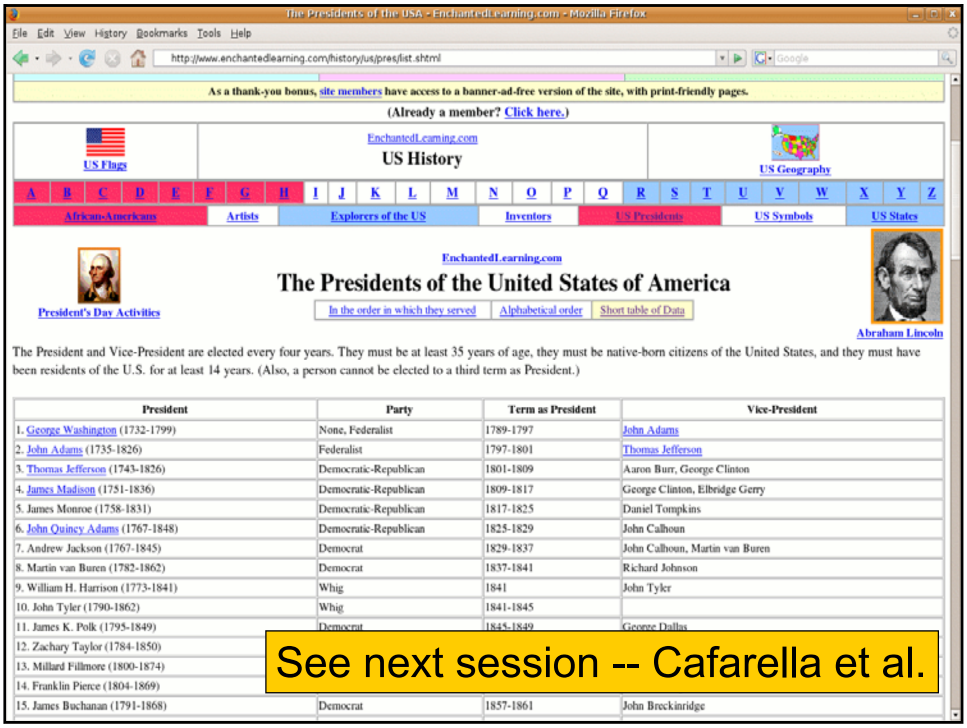
Query processing as fact gathering



First name: Peter
Middle name:
Last name: Buneman
Address: ?

Outline

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



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.)

EnchantedLearning.com

US History

EnchantedLearning.com

The Presidents of the United States of America

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

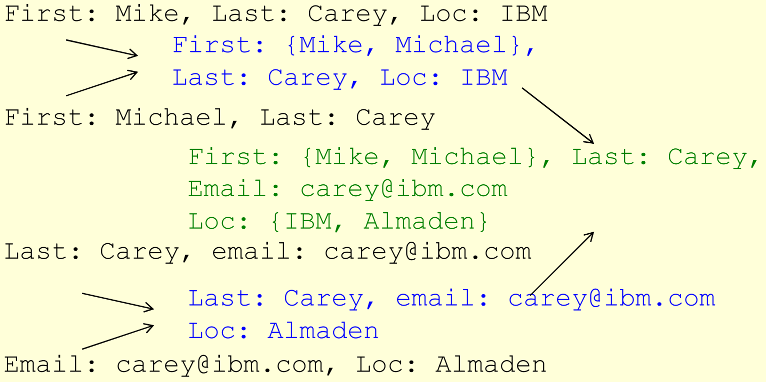
“What exam questions do I have that weren’t in the HW, weren’t on the practice exam, weren’t used in class, aren’t in the back of the book, aren’t examples in the book?”

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: {Mike, Michael},
Last: Carey, Loc: IBM

First: Michael, Last: Carey

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

Last: Carey, email: carey@ibm.com

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

Email: carey@ibm.com, Loc: Almaden

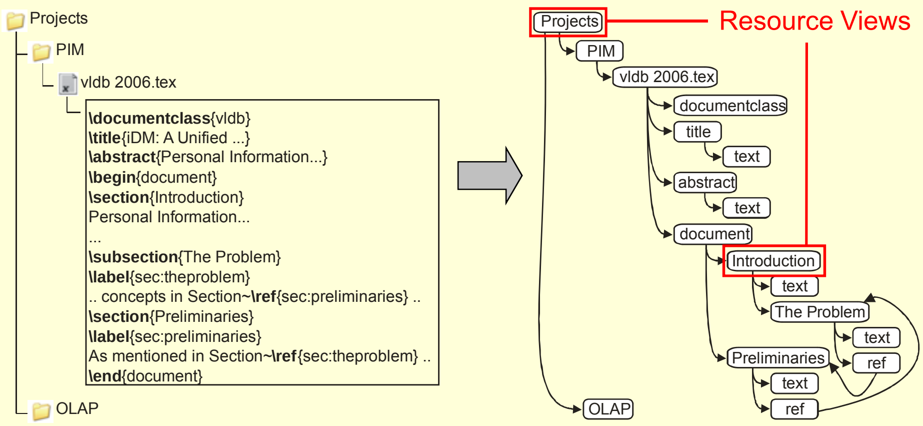
Evolving Objects

- 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 over come 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



```
\documentclass{vldb}
\title{iDM: A Unified ...}
\abstract{Personal Information...}
\begin{document}
\section{Introduction}
Personal Information...
...
\subsection{The Problem}
\label{sec:theproblem}
.. concepts in Section~\ref{sec:preliminaries} ..
\section{Preliminaries}
\label{sec:preliminaries}
As mentioned in Section~\ref{sec:theproblem} ..
\end{document}
```

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 **M** = { $(M_1,Pr(M_1)), \ldots, (M_l,Pr(M_l))$}
  - P-mappings **pM** = {$pM(M_1), \ldots, pM(M_l)$ }
- Output probability of tuple **t:**
  - $p = \Sigma Pr(t|M_i) * Pr(M_i)$

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
```

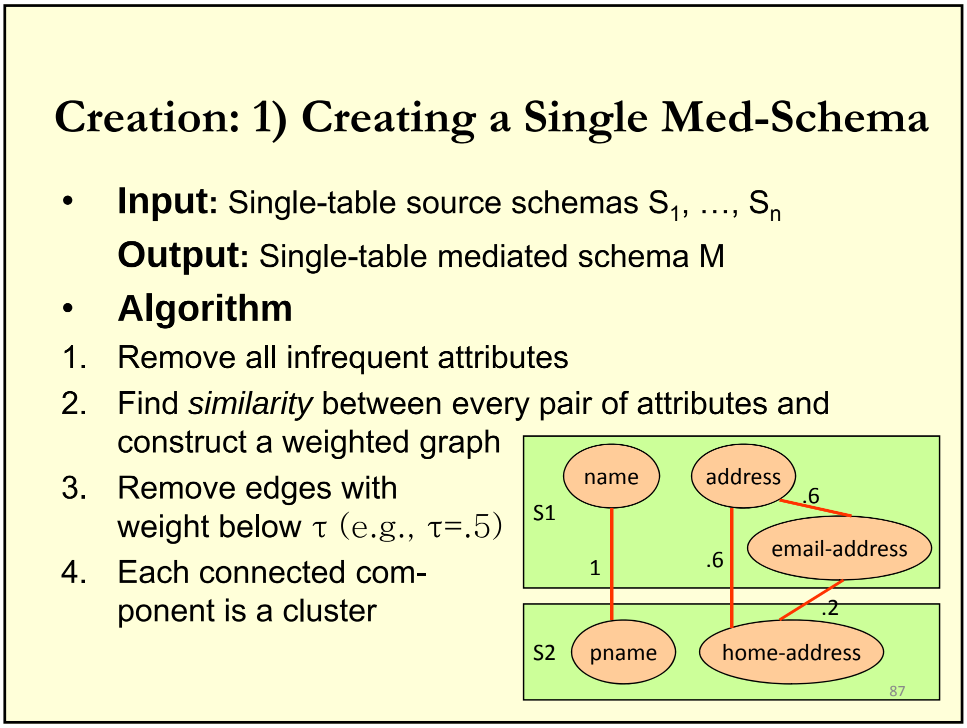
Answers

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

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)

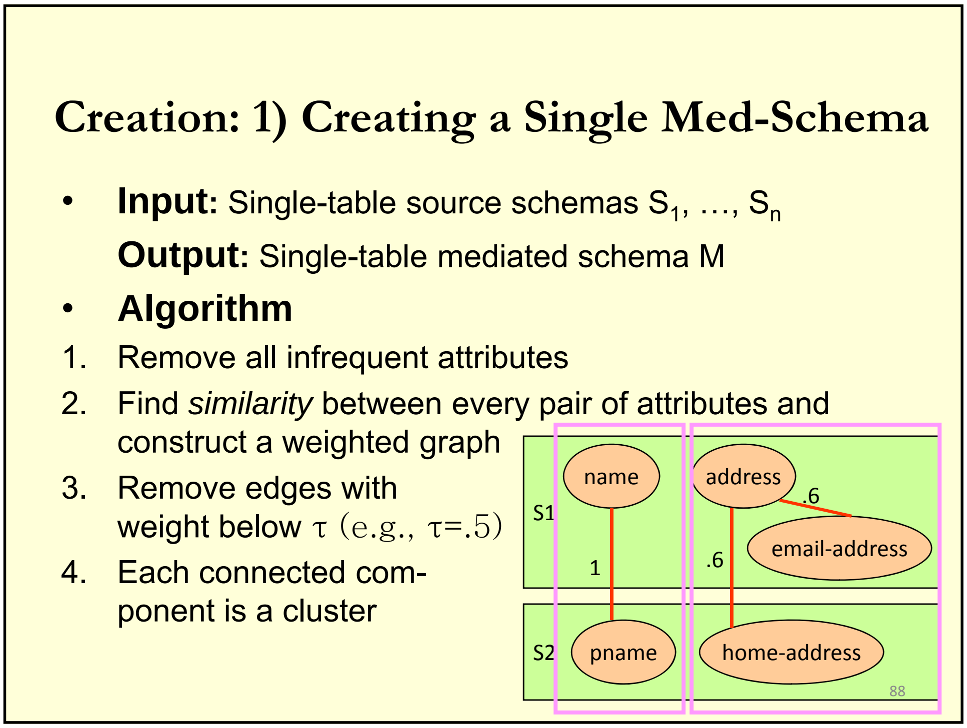


Creation: 1) Creating a Single Med-Schema

- **Input:** Single-table source schemas S_1, …, 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., τ=.5)
4. Each connected component is a cluster



Creation: 1) Creating a Single Med-Schema

- **Input:** Single-table source schemas S_1, …, S_n

  Output: Single-table mediated schema M
- **Algorithm**

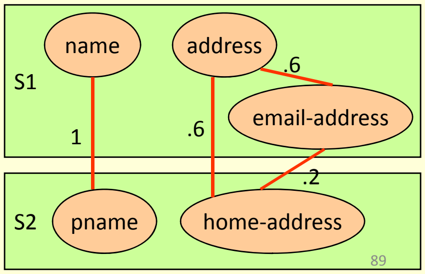
1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. Remove edges with weight below τ (e.g., τ=.5)
4. Each connected component is a cluster

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 ≥ τ+ϵ) → retain
   - (weight < τ-ϵ) → drop
   - (τ-ϵ ≤ weight < τ+ϵ) → uncertain edge

   (e.g., τ =.6, ϵ = .2)
4. Clustering for each combo of including/excluding uncertain edges

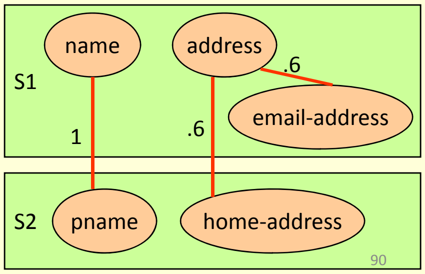


Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
   - (weight ≥ τ+ϵ) → retain
   - (weight < τ-ϵ) → drop
   - (τ-ϵ ≤ weight < τ+ϵ) → uncertain edge

   (e.g., τ =.6, ϵ = .2)
4. Clustering for each combo of including/excluding uncertain edges

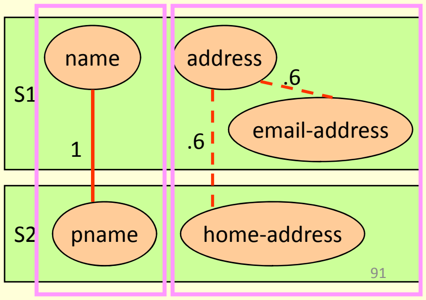


Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
   - (weight ≥ τ+ϵ) → retain
   - (weight < τ-ϵ) → drop
   - (τ-ϵ ≤ weight < τ+ϵ) → uncertain edge

   (e.g., τ =.6, ϵ = .2)
4. Clustering for each combo of including/excluding uncertain edges

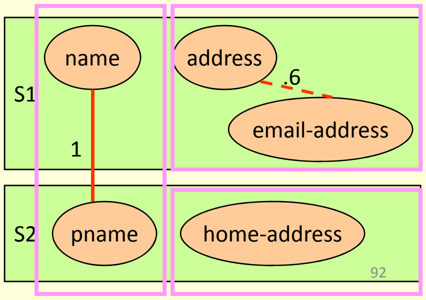


Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
   - (weight ≥ τ+ϵ) → retain
   - (weight < τ-ϵ) → drop
   - (τ-ϵ ≤ weight < τ+ϵ) → uncertain edge

   (e.g., τ =.6, ϵ = .2)
4. Clustering for each combo of including/excluding uncertain edges

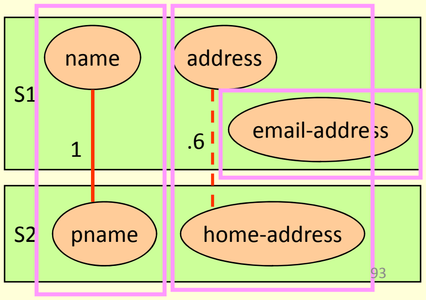


Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
   - (weight ≥ τ+ϵ) → retain
   - (weight < τ-ϵ) → drop
   - (τ-ϵ ≤ weight < τ+ϵ) → uncertain edge

   (e.g., τ =.6, ϵ = .2)
4. Clustering for each combo of including/excluding uncertain edges

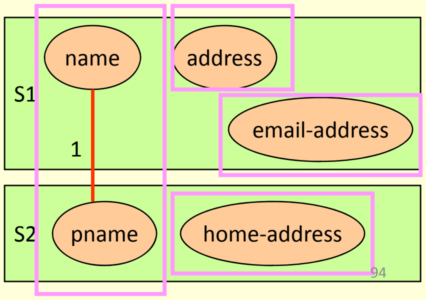


Creation: 2) Creating All Possible Med-Schemas

- **Algorithm**

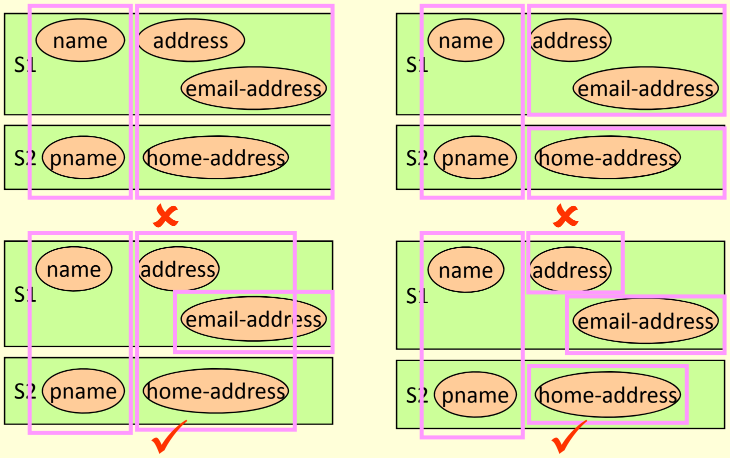
1. Remove all infrequent attributes
2. Find *similarity* between every pair of attributes and construct a weighted graph
3. For each edge
   - (weight ≥ τ+ϵ) → retain
   - (weight < τ-ϵ) → drop
   - (τ-ϵ ≤ weight < τ+ϵ) → uncertain edge

   (e.g., τ =.6, ϵ = .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*



Creation: 3) Computing Probabilities

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