Information Integration on the Web

AAAI Tutorial (MA1)

Monday July 29th 2002. 9am-1pm

Craig Knoblock & Subbarao Kambhampati
Tutorial Objectives

• Motivate the need for Information Integration
• Survey the current work
  – With emphasis on the many roles AI technology can play in supporting information integration
Caveats

• The subject falls in the middle of Databases and AI
  – Need background in both
    • This tutorial will assume AI background, and will provide rudimentary review of relevant database material
  – Clash of cultures
    • DB approaches tend to be bottom-up, while AI approaches tend to be top-down
      – Both are needed…
Acknowledgements

• “Slide Integration”
  – Thanks to
    • Alon (Ha)Levy, Jim Hendler, Eric Lambrecht, Zaiqing Nie, Ullas Nambiar, Sreelakshmi Vaddi
    • Greg Barish, Steve Minton, Ion Muslea, Kristina Lerman, Sheila Tejada

for permission to use/mutilate some of their slides
Preamble & Platitudes

• Internet is growing at an enormous rate
  – Even after the bubble-burst
• All kinds of information sources are online
  – Web pages, databases masquerading as web pages, Services, Sensors
• Promise of unprecedented information access to every Tom, Dick and Mary..
  – But, right now, they still need to “know” where to go, and be willing to manually put together bits and pieces of information gleaned from various sources and services

“Information Integration” aims to do this automatically.
The Problem

Source Trust

Source Fusion/
Query Planning

Executor

Monitor

Services

Webpages

Structured data

Sensors (streaming Data)

Query

Answers

Mediator

Kambhampati & Knoblock

Information Integration on the Web (MA-1)
• User queries refer to the **mediated schema**.
• Data is stored in the sources in a **local schema**.
• Content descriptions provide the semantic mappings between the different schemas.
• Mediator uses the descriptions to translate user queries into queries on the sources.

**Source Fusion/Query Planning**
Needs to handle:
- Multiple objectives,
- Service composition,
- Source quality & overlap

**Source Trust**
Ontologies;
Source/Service Descriptions

**Executor**
Needs to handle
Source/network Interruptions,
Runtime uncertainty, replanning

**Monitor**
Updating Statistics
Replanning Requests
Source Calls
Probing Queries

**Preference/Utility Model**

**Answers**

**Query**

**DWIM**

Kambhampati & Knoblock
Isn’t web mostly text?

– The invisible web is mostly structured…
  • Most web servers have back end database servers
  • They dynamically convert (wrap) the structured data into readable english
    – <India, New Delhi>  => The capital of India is New Delhi.
    – So, if we can “unwrap” the text, we have structured data!
      » (un)wrappers, learning wrappers etc…
    – Note also that such dynamic pages cannot be crawled...

– The (coming) Semi-structured web
  • Most pages are at least “semi”-structured
  • XML standard is expected to ease the presentation/on-the-wire transfer of
    such pages. (BUT…..)

– The Services
  • Travel services, mapping services

– The Sensors
  Stock quotes, current temperatures, ticket prices…
Why isn’t this just

• Search engines do text-based retrieval of URLs
  – Works reasonably well for single document texts, or for finding sites based on single document text
    • Cannot integrate information from multiple documents
    • Cannot do effective “query relaxation” or generalization
    • Cannot link documents and databases

• The aim of Information integration is to support query processing over structured and semi-structured sources as well as services.
Are we talking “comparison shopping” agents?

- Certainly closer to the aims of these
- But:
  - Wider focus
  - Consider larger range of databases
  - Consider services
  - Implies more challenges
  - “warehousing” may not work
  - Manual source characterization/integration won’t scale-up
Why isn’t this just Distributed Databases

- **No common schema**
  - Sources with heterogeneous schemas (and ontologies)
  - Semi-structured sources
- **Legacy Sources**
  - Not relational-complete
  - Variety of access/process limitations
- **Autonomous sources**
  - No central administration
  - Uncontrolled source content overlap
- **Unpredictable run-time behavior**
  - Makes query execution hard
- **Presence of “services”**
  - Need to “compose” services
Who is dying to have it? (Applications)

• **WWW:**
  – Comparison shopping
  – Portals integrating data from multiple sources
  – B2B, electronic marketplaces

• **Science and culture:**
  – Medical genetics: integrating genomic data
  – Astrophysics: monitoring events in the sky.
  – Culture: uniform access to all cultural databases produced by countries in Europe provinces in Canada

• **Enterprise data integration**
  – An average company has 49 different databases and spends 35% of its IT dollars on integration efforts
What’s AI got to do with it?

Automated Planning
- Plan languages
- Service composition
- Reactive planning/
  Plan monitoring
- Plan recognition

Learning/Mining
- Source discovery
- Source statistics
- Wrapper Learning

Knowledge Rep
- Ontologies
- Meta-data/Inference
  Query languages

Source Trust
Ontologies;
Source/Service Descriptions

Source Fusion/
Query Planning
Needs to handle:
Multiple objectives,
Service composition,
Source quality & overlap

Executor
Needs to handle
Source/network Interruptions,
Runtime uncertainty, replanning

Monitor

Skeptic’s corner
The Problem

Source Calls

Probing Queries

Structured data

Services

Web pages

Sensors
(streaming Data)

Kambhampati & Knoblock
Information Integration on the Web (MA-1)
Issues in Information Integration
- User queries refer to the mediated schema.
- Data is stored in the sources in a local schema.
- Content descriptions provide the semantic mappings between the different schemas.
- Mediator uses the descriptions to translate user queries into queries on the sources.
Overview

- User queries refer to the *mediated schema*.
- Data is stored in the sources in a *local schema*.
- Content descriptions provide the semantic mappings between the different schemas.
- Mediator uses the descriptions to translate user queries into queries on the sources.

*Schema*: Template for the stored data
Source Descriptions

- Contains all meta-information about the sources:
  - Logical source contents (books, new cars).
  - Source capabilities (can answer SQL queries)
  - Source completeness (has all books).
  - Physical properties of source and network.
  - Statistics about the data (like in an RDBMS)
  - Source reliability
  - Mirror sources
  - Update frequency.
Source Fusion/Query Planning

• Accepts user query and generates a plan for accessing sources to answer the query
  – Needs to handle tradeoffs between cost and coverage
  – Needs to handle source access limitations
  – Needs to reason about the source quality/reputation
Monitoring/Execution

- Takes the query plan and executes it on the sources
  - Needs to handle source latency
  - Needs to handle transient/short-term network outages
  - Needs to handle source access limitations
  - May need to re-schedule or re-plan
Dimensions to Consider

• How many sources are we accessing?
• How autonomous are they?
• Can we get meta-data about sources?
• Is the data structured?
• Supporting just queries or also updates?
• Requirements: accuracy, completeness, performance, handling inconsistencies.
• Closed world assumption vs. open world?
Overview

- Motivation for Information Integration [Rao]
- Accessing Information Sources [Craig]
- Models for Integration [Rao]
- Query Planning & Optimization [Rao]
- Plan Execution [Craig]
- Standards for Integration/Mediation [Rao]
- Ontology & Data Integration [Craig]
- Future Directions [Craig]

We will play tag, so you won’t get to sleep 😊
Acessing Information Sources

Wrapper Learning
Wrapper Induction

Problem description:

- Web sources present data in *human-readable format*
  - take user query
  - apply it to data base
  - present results in “template” HTML page

- To integrate data from multiple sources, one must first *extract relevant information* from Web pages

- Task: learn extraction rules based on labeled examples
  - Hand-writing rules is tedious, error prone, and time consuming
Example of Extraction Task

Casablanca Restaurant
220 Lincoln Boulevard, Venice, CA 90291
(310) 392-5751

NAME Casablanca Restaurant
STREET 220 Lincoln Boulevard
CITY Venice
PHONE (310) 392-5751
• Assumes items are always in **fixed, known order**
  ... Name: J. Doe; **Address:** 1 Main; **Phone:** 111-1111. <p>
  Name: E. Poe; **Address:** 10 Pico; **Phone:** 777-1111. <p> ...

• Introduces several types of wrappers
  ![Diagram](image)

• Advantages:
  • Fast to learn & extract

• Drawbacks:
  • Cannot handle permutations and missing items
  • Must label entire page
Rule Learning

- Machine learning:
  - Use past experiences to improve performance

- Rule learning:
  - INPUT:
    - Labeled examples: training & testing data
    - Admissible rules (hypotheses space)
    - Search strategy

  - Desired output:
    - Rule that performs well both on training and testing data
Learning LR extraction rules

- Admissible rules:
  - prefixes & suffixes of items of interest

- Search strategy:
  - start with shortest prefix & suffix, and expand until correct
SoftMealy [Hsu & Dung, ‘98]

- Learns a transducer
- Advantages:
  - Also learns order of items
  - Allows item permutations & missing items
  - Uses wildcards (eg, Number, AllCaps, etc)
- Drawback:
  - Must “see” all possible permutations
Whizbang! Site Wrapper
[Cohen & Jensen ’01, IJCAI ATEM Workshop]

- Uses Inductive Logic Programming techniques to create and compose “builders”
- Exploits the Document Object Model to determine hierarchy and lists
- Extracts data from either the DOM or sequences of tokens

**Advantages**
- Can exploit hierarchical structure without having to specify it

**Disadvantages**
- Dependent on table and list structures to extract repeated elements
**RoadRunner Wrappers**

[Crescenzi, Mecca, & Merialdo, 2001]

- Automatically generates wrappers web pages
- Supports nested structures and lists
- Applies to large, complex pages with regular structure

**Approach**
- Start with the first page and create a union-free regular expression that defines the wrapper
- Match each successive sample against the wrapper
- Mismatches result in generalizations of the regular expression
Example Matching

- Wrapper (initially Page 1):
  01:  <HTML>
  02:  Books of:  
  03:  <B>       
  04:  John Smith 
  05:  </B>     
  06:  <UL>     
  07:  <LI>     
  08-10: <I>Title: </I>
  11:  DB Primer  
  12:  </LI>    
  13:  <LI>     
  14-16: <I>Title: </I>
  17:  Comp. Sys. 
  18:  </LI>    
  19:  </UL>   
  20:  </HTML>

- Sample (Page 2):
  01:  <HTML>
  02:  Books of:  
  03:  <B>       
  04:  Paul Jones 
  05:  </B>     
  06:  <IMG src=.../> 
  07:  <UL>     
  08:  <LI>     
  09-11: <I>Title: </I>
  12:  XML at Work
  13:  </LI>    
  14:  <LI>     
  15-17: <I>Title: </I>
  18:  HTML Scripts
  19:  </LI>    
  20:  <LI>     
  21-23: <I>Title: </I>
  24:  Javascript
  25:  </LI>    
  26:  </UL>   
  27:  </HTML>

- Wrapper after solving mismatches:

  <HTML>Books of:<B>#PCDATA</B>  
  ( <IMG src=.../> )
  <UL>  
  ( <LI><I>Title: </I>#PCDATA</LI> ) +
  </UL></HTML>
Types of Mismatches

- String mismatches are used to discover fields of the document
- Tag mismatches can indicate either optional elements or iterators
- For iterations, mismatch is caused by repeated elements in a list
  - End of the list corresponds to last matching token
  - Beginning of list corresponds to one of the mismatched tokens
  - These create possible “squares”
Limitations

• Assumptions:
  • Pages are well-structured
  • Want to extract at the level of entire fields
  • Structure can be modeled without disjunctions

• Search space for explaining mismatches is huge
  • Uses a number of heuristics to prune space
    • Limited backtracking
    • Limit on number of choices to explore
    • Patterns cannot be delimited by optionals
  • Will result in pruning possible wrappers
Hierarchical wrapper induction
- Decomposes a hard problem in several easier ones
- Extracts items independently of each other
- Each rule is a finite automaton

Advantages:
- Powerful extraction language (e.g., embedded list)
- One hard-to-extract item does not affect others

Disadvantage:
- Does not exploit item order (sometimes may help)
STALKER: The Wrapper Architecture

Query Data

Information Extractor

EC Tree

Extraction Rules
Extraction rule: sequence of *landmarks*

SkipTo(Phone) SkipTo(<i>) SkipTo(</i>)

Name: Joel’s  Phone: <i>(310) 777-1111</i> Review: …
The Embedded Catalog Tree (ECT)

Name: KFC
Cuisine: Fast Food
Locations:

- Venice: (310) 123-4567, (800) 888-4412.
- L.A.: (213) 987-6543.
- Encino: (818) 999-4567, (888) 727-3131.
Learning the Extraction Rules

GUI → Labeled Pages → EC Tree → Inductive Learning System → Extraction Rules
Example of Rule Induction

Training Examples:

Name: Del Taco <p> Phone (toll free): <b>( 800 ) 123-4567 </b><p>Cuisine ...

Name: Burger King <p> Phone: ( 310 ) 987-9876 <p> Cuisine: …

Initial candidate:

SkipTo( ()

SkipTo( <b> ( ) … SkipTo(Phone) SkipTo( () … SkipTo( :) SkipTo( ()

… SkipTo(Phone) SkipTo( :) SkipTo( () …
Active Learning & Information Agents

- **Active Learning**
  - **Idea:** system selects most informative exs. to label
  - **Advantage:** fewer examples to reach same accuracy

- **Information Agents**
  - One agent may use hundreds of extraction rules
    - Small reduction of *examples per rule* => big impact on user
  - Why stop at 95-99% accuracy?
    - Select most informative examples to get to 100% accuracy
Which example should be labeled next?

Training Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joel’s</td>
<td>(310) 777-1111</td>
<td>The chef...</td>
</tr>
<tr>
<td>Kim’s</td>
<td>(213) 757-1111</td>
<td>Korean...</td>
</tr>
</tbody>
</table>

Unlabeled Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Jean</td>
<td>(310) 666-1111</td>
<td>...</td>
</tr>
<tr>
<td>Burger King</td>
<td>(818) 789-1211</td>
<td>...</td>
</tr>
<tr>
<td>Café del Rey</td>
<td>(310) 111-1111</td>
<td>...</td>
</tr>
<tr>
<td>KFC</td>
<td>(800) 111-7171</td>
<td>...</td>
</tr>
</tbody>
</table>
Two ways to find start of the phone number:

SkipTo( Phone: )

BackTo( Number )

Name: KFC  Phone: (310) 111-1111  Review: Fried chicken ...
<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joel’s</td>
<td>(310) 777-1111</td>
<td>...</td>
</tr>
<tr>
<td>Kim’s</td>
<td>(213) 757-1111</td>
<td>...</td>
</tr>
<tr>
<td>Chez Jean</td>
<td>666-1111</td>
<td>...</td>
</tr>
<tr>
<td>Burger King</td>
<td>789-1211</td>
<td>...</td>
</tr>
<tr>
<td>Café del Rey</td>
<td>111-1111</td>
<td>...</td>
</tr>
<tr>
<td>KFC</td>
<td>111-7171</td>
<td>...</td>
</tr>
</tbody>
</table>
Accessing Information Sources

Wrapper Maintenance
Wrapper Maintenance

Problem

- Landmark-based extraction rules are fast and efficient...but they rely on stable page layout
- If the page layout changes, the wrapper fails!
- Average site on the Web changes layout more than twice a year
- Need to detect changes and automatically re-induce extraction rules when layout changes
Learning Regular Expressions
[Goan, Benson, & Etzioni, 1996]

- Character level description of extracted data
- Based on ALERGIA [Carrasco and Oncina, 1994]
  - Stochastic grammar induction algorithm
  - Merges too many states resulting in over-general grammar
- WIL reduced faulty merges by imposing syntactic categories:
  - Number, lower upper, and delim
- Only merges when nodes contain the same syntactic category
- Requires large number of examples to learn
- Computationally expensive
Learning Global Properties for Wrapper Verification [Kushmerick, 1999]

• Each data field described by global numeric features
  • Word count, average word length, HTML density, alphabetic density

• Computationally efficient learning

• HTML density alone could account for almost all changes on test set

• Large number of false negatives on real changes to web sources [Lerman, Knoblock, Minton, 2002]
Learning Data Prototypes
[Lerman & Minton, 2000]

- Approach to learning the structure of data
- Token level syntactic description
  - descriptive but compact
  - computationally efficient
- Structure is described by a sequence (pattern) of general and specific tokens.
- Data prototype = starting & ending patterns

```
STREET_ADDRESS
start with: START _NUM _CAPS
220 Lincoln Blvd
2040 Sawtelle Blvd
420 S Fairview Ave
end with: _CAPS Blvd _CAPS _CAPS
```
Wrapper Verification

Data prototypes can be used for web wrapper maintenance applications.

- Automatically detect when the wrapper is no longer correctly extracting data from an information source
  - (Kushmerick 1999)
Wrapper Reinduction

- Rebuild the wrapper automatically if it is not extracting data correctly from new pages
- Data extraction step
  - Identify correct examples of data on new pages
- Wrapper induction step
  - Feed the examples, along with the new pages, to the wrapper induction algorithm to learn new extraction rules
The Lifecycle of A Wrapper

GUI → Wrapper Induction System → To be labeled

Web pages → Wrapper → Extracted data

Automatic Re-labeling → Wrapper Verification
(An exceedingly brief) Database Refresher

Overview

- Motivation for Information Integration [Rao]
- Accessing Information Sources [Craig]
- Models for Integration [Rao]
- Query Planning & Optimization [Rao]
- Plan Execution [Craig]
- Standards for Integration/Mediation [Rao]
- Ontology & Data Integration [Craig]
- Future Directions [Craig]
Traditional Database Architecture

Query (SQL) -> Database Manager (DBMS)
- Storage mgmt
- Query processing
- View management
- (Transaction processing)

Answer (relation) -> Database (relational)

Source Trust
- Ontologies
- Source/Service Descriptions

Source Fusion/Query Planning
- Needs to handle: multiple objectives, Source quality & overlap

Source Calls

Preference/Utility Model

Replanning Requests

Monitor

Updating Statistics

Executor
- Needs to handle: Interruptions, Runtime uncertainty, replanning

Probing Queries

Source Calls
Database Outline

• What we care about
  – Structured data representations
    • Relational databases
    • Deductive databases
  – Structured query languages
    • SQL
      – Views (& materialized views)
  – Query optimization overview
Relational Data: Terminology

Attribute names

Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>gizmo</td>
<td>$19.99</td>
<td>gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Power gizmo</td>
<td>$29.99</td>
<td>gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

tuples

schema

Product(name: string, Price: real, category: enum, Manufacturer: string)

(Arity=4)
Relational Algebra

- **Operators**
  - tuple sets as input, new set as output
- **Operations**
  - Union, Intersection, difference, ..
  - Selection (σ)
  - Projection (Π)
  - Cartesian product (X)
- **Join (⋈)**
SQL: A query language for Relational Algebra

Many standards out there: SQL92, SQL2, SQL3, SQL99

- **Select** attributes
- **From** relations (possibly multiple, joined)
- **Where** conditions (selections)

Other features:
- aggregation, group-by etc.

```
“Find companies that manufacture products bought by Joe Blow”
SELECT Company.name
FROM Company, Product
WHERE Company.name=Product.maker
AND Product.name IN
  (SELECT product
   FROM Purchase
   WHERE buyer = “Joe Blow”);
```
Deductive Databases

- Relations viewed as predicates.
- Interrelations between relations expressed as “datalog” rules
  - (Horn clauses, without function symbols)
  - Queries correspond to datalog programs
    - “Conjunctive queries” are datalog programs with a single non-recursive rule [Correspond to SPJ queries in SQL]

\[
\text{Emprelated}(\text{Name}, \text{Dname}) :\text{Empdep}(\text{Name}, \text{Dname})
\]

\[
\text{Emprelated}(\text{Name}, \text{Dname}) :\text{Empdep}(\text{Name}, \text{D1}), \text{Emprelated}(\text{D1}, \text{Dname})
\]
CREATE VIEW Seattle-view AS

SELECT buyer, seller, product, store
FROM Person, Purchase
WHERE Person.city = "Seattle" AND
Person.name = Purchase.buyer

We can later use the views:

SELECT name, store
FROM Seattle-view, Product
WHERE Seattle-view.product = Product.name AND
Product.category = "shoes"

What’s really happening when we query a view??
**Query Optimization**

*Declarative SQL query* → *Imperative query execution plan:*

```
SELECT S.buyer
FROM Purchase P, Person Q
WHERE P.buyer=Q.name AND
  Q.city='seattle' AND
  Q.phone > '5430000'
```

---

**Inputs:**
- the query
- statistics about the data (indexes, cardinalities, selectivity factors)
- available memory

---

**Ideally:** Want to find best plan. **Practically:** Avoid worst plans!

(different join orders)

(different ways of scanning tables) "shortest execution time"

(different placements of selections w.r.t joins)
Integrator vs. DBMS

- No common schema
  - Sources with heterogeneous schemas
  - Semi-structured sources
- Legacy Sources
  - Not relational-complete
  - Variety of access/process limitations
- Autonomous sources
  - No central administration
  - Uncontrolled source content overlap
  - Lack of source statistics
- Tradeoffs between query plan cost, coverage, quality etc.
  - Multi-objective cost models
- Unpredictable run-time behavior
  - Makes query execution hard
- Presence of “services”
  - Need to “compose” services
Models for Integration

Overview

- Motivation for Information Integration [Rao]
- Accessing Information Sources [Craig]
- Models for Integration [Rao]
- Query Planning & Optimization [Rao]
- Plan Execution [Craig]
- Standards for Integration/Mediation [Rao]
- Ontology & Data Integration [Craig]
- Future Directions [Craig]
Solutions for small-scale integration

- Mostly ad-hoc programming: create a special solution for every case; pay consultants a lot of money.
- Data warehousing: load all the data periodically into a warehouse.
  - 6-18 months lead time
  - Separates operational DBMS from decision support DBMS. (not only a solution to data integration).
  - Performance is good; data may not be fresh.
  - Need to clean, scrub you data.

Junglee did this, for employment classifieds
The Virtual Integration Architecture

- Leave the data in the sources.
- When a query comes in:
  - Determine the relevant sources to the query
  - Break down the query into sub-queries for the sources.
  - Get the answers from the sources, and combine them appropriately.
- Data is fresh. Approach scalable
- Issues:
  - Relating Sources & Mediator
  - Reformulating the query
  - Efficient planning & execution

Garlic [IBM], Hermes[UMD]; Tsimmis, InfoMaster[Stanford]; DISCO[INRIA]; Information Manifold [AT&T]; SIMS/Ariadne[USC]; Emerac/Havasu[ASU]
Desiderata for Relating Source-Mediator Schemas

• **Expressive power**: distinguish between sources with closely related data. Hence, be able to prune access to irrelevant sources.

• **Easy addition**: make it easy to add new data sources.

• **Reformulation**: be able to reformulate a user query into a query on the sources efficiently and effectively.

• **Nonlossy**: be able to handle all queries that can be answered by directly accessing the sources.

---

**Reformulation**

- **Given:**
  - A query $Q$ posed over the mediated schema
  - Descriptions of the data sources

- **Find:**
  - A query $Q'$ over the data source relations, such that:
    - $Q'$ provides only *correct answers* to $Q$, and
    - $Q'$ provides *all* possible answers to $Q$ given the sources.
Approaches for relating source & Mediator Schemas

- **Global-as-view (GAV):** express the mediated schema relations as a set of views over the data source relations
- **Local-as-view (LAV):** express the source relations as views over the mediated schema.
- Can be combined…?

```
CREATE VIEW Seattle-view AS
    SELECT buyer, seller, product, store
    FROM Person, Purchase
    WHERE Person.city = "Seattle" AND Person.name = Purchase.buyer

We can later use the views:
```

```
SELECT name, store
FROM Seattle-view, Product
WHERE Seattle-view.product = Product.name AND Product.category = "shoes"
```

“View” Refresher

Let’s compare them in a movie Database integration scenario..
Global-as-View

Mediated schema:
Movie(title, dir, year, genre),
Schedule(cinema, title, time).

Express mediator schema relations as views over source relations

[S1(title, dir, year, genre)]

[S2(title, dir, year, genre)]

[S3(title, dir), S4(title, year, genre)]
Global-as-View

Mediated schema:

Movie(title, dir, year, genre),
Schedule(cinema, title, time).

Create View Movie AS

select * from S1  
[\text{S1}(title, dir, year, genre)]
union
select * from S2  
[\text{S2}(title, dir, year, genre)]
union  
[\text{S3}(title, dir), \text{S4}(title, year, genre)]
select S3.title, S3.dir, S4.year, S4.genre
from S3, S4
where S3.title=S4.title

Express mediator schema relations as views over source relations

Mediator schema relations are Virtual views on source relations
Local-as-View: example 1

Mediated schema:

Movie(title, dir, year, genre),
Schedule(cinema, title, time).

Create Source S1 AS
select * from Movie

Create Source S3 AS
select title, dir from Movie

Create Source S5 AS
select title, dir, year
from Movie
where year > 1960 AND genre="Comedy"

Sources are “materialized views” of mediator schema

Express source schema relations as views over mediator relations
GAV vs. LAV

Mediated schema:
Movie(title, dir, year, genre),
Schedule(cinema, title, time).

Source S4:  S4(cinema, genre)

Create View Movie AS
select NULL, NULL, NULL, genre
from S4
Create View Schedule AS
select cinema, NULL, NULL
from S4.

But what if we want to find which cinemas are playing comedies?

Create Source S4
select cinema, genre
from Movie m, Schedule s
where m.title=s.title

Now if we want to find which cinemas are playing comedies, there is hope!

Lossy mediation
<table>
<thead>
<tr>
<th><strong>GAV</strong></th>
<th><strong>vs.</strong></th>
<th><strong>LAV</strong></th>
</tr>
</thead>
</table>
| • Not modular  
  – Addition of new sources changes the mediated schema  
• Can be awkward to write mediated schema without loss of information  
• Query reformulation easy  
  – *reduces to view unfolding* (*polynomial*)  
  – Can build hierarchies of mediated schemas |  | • Modular--adding new sources is easy  
• Very flexible--power of the entire query language available to describe sources  
• Reformulation is hard  
  – Involves answering queries only using views (can be intractable—see below) |
| • Best when  
  – Few, stable, data sources  
  – well-known to the mediator (e.g. corporate integration)  
• Garlic, TSIMMIS, HERMES |  | • Best when  
  – Many, relatively unknown data sources  
  – possibility of addition/deletion of sources  
• Information Manifold, InfoMaster, Emerac, Havasu |
Reformulation in LAV: The issues

Query: Find all the years in which Zhang Yimou released movies.

Select year
from movie M
where M.dir=yimou

Q(y) :- movie(T,D,Y,G), D=yimou

Which is the better plan?
What are we looking for?
--equivalence?
--containment?
--Maximal Containment
--Smallest plan?
Reformulation Algorithms

Bucket Algorithm

\[ \text{Q(\cdot)} \leftarrow \text{V1(\cdot) \& V2(\cdot)} \]

- Bucket algorithm
  - Cartesian product of buckets
  - Followed by “containment” check

[Levy]

\[ P_1 \text{ contains } P_2 \text{ if } P_2 \models P_1 \]

Inverse Rules

\[ \text{S11(\cdot)} \leftarrow \text{V1(\cdot)} \]
\[ \text{S12(\cdot)} \leftarrow \text{V1(\cdot)} \]
\[ \text{S21(\cdot)} \leftarrow \text{V2(\cdot)} \]
\[ \text{S22(\cdot)} \leftarrow \text{V2(\cdot)} \]
\[ \text{S00(\cdot)} \leftarrow \text{V1(\cdot), V2(\cdot)} \]

[Levy]

\[ \text{Q(\cdot)} \leftarrow \text{V1(\cdot) \& V2(\cdot)} \]

[Levy]

\[ \text{V1(\cdot)} \leftarrow \text{S11(\cdot)} \]
\[ \text{V1(\cdot)} \leftarrow \text{S12(\cdot)} \]
\[ \text{V1(\cdot)} \leftarrow \text{S00(\cdot)} \]
\[ \text{V2(\cdot)} \leftarrow \text{S21(\cdot)} \]
\[ \text{V2(\cdot)} \leftarrow \text{S22(\cdot)} \]
\[ \text{V2(\cdot)} \leftarrow \text{S00(\cdot)} \]

[Duschka]

\[ \text{Inverse Rules} \]
- plan fragments for mediator relations

[Duschka]
Complexity of finding maximally-contained plans in LAV

- Complexity does change if the sources are not “conjunctive queries”
  - Sources as unions of conjunctive queries \( \text{NP-hard} \)
    - Disjunctive descriptions
  - Sources as recursive queries \( \text{Undecidable} \)
    - Comparison predicates
- Complexity is less dependent on the query
  - Recursion okay; but inequality constraints lead to \text{NP-hardness}
- Complexity also changes based on Open vs. Closed world assumption

[Abiteboul & Duschka, 98]
Practical issues complicating Reformulation

- Sources may have access limitations
  - Access restrictions can lead to recursive rewritings even when the queries are non-recursive!
- Sources may have overlap
  - Non-minimal rewritings may result when overlap information is ignored
Source Limitations

• **Sources are not really fully-relational databases**
  – Legacy systems
  – Limited access patterns
    • (Can’s ask a white-pages source for the list of all numbers)
  – Limited local processing power
    • Typically only selections (on certain attributes) are supported

• Access limitations modeled in terms of allowed (“feasible”) binding patterns with which the source can be accessed
  – E.g. $S(X,Y,Z)$ with feasible patterns $f,f,b$ or $b,b,f$
Access Restrictions & Recursive Reformulations

Create Source S1 as

\[
\text{select * from Cites given \text{paper1}}
\]

Create Source S2 as

\[
\text{select \text{paper from ASU-Papers}}
\]

Create Source S3 as

\[
\text{select \text{paper from AwardPapers given \text{paper}}}
\]

Query: select * from AwardPapers

\[
\begin{align*}
S1^{bf}(p_1,p_2) & :\text{ites}(p_1,p_2) \\
S2(p) & :\text{Asp}(p) \\
S3^{b}(p) & :\text{Awp}(p)
\end{align*}
\]

\[
\begin{align*}
Q(p) & :\text{Awp}(p) \\
\text{Awp}(p) & :\text{Dom}(p), S3^{b}(p) \\
\text{Asp}(p) & :S2(p) \\
\text{Cites}(p_1,p_2) & :\text{Dom}(p), S1^{bf}(p)
\end{align*}
\]

\[
\begin{align*}
\text{Dom}(p) & :S2(p) \\
\text{Dom}(p) & :\text{Dom}(p_1), S1(p_1,p)
\end{align*}
\]

Recursive plan!!

[Kwok&Weld, 96; Duschka &Levy, 97]
Managing Source Overlap

• Often, sources on the Internet have overlapping contents
  – The overlap is not centrally managed (unlike DDBMS—data replication etc.)

• Reasoning about overlap is important for plan optimality
  – We cannot possibly call all potentially relevant sources!

• Qns: How do we characterize, get and exploit source overlap?
  – Qualitative approaches (LCW statements)
  – Quantitative approaches (Coverage/Overlap statistics)
Local Completeness Information

- If sources are incomplete, we need to look at each one of them.
- Often, sources are *locally complete*.
- Movie(title, director, year) complete for years after 1960, or for American directors.
- **Question**: given a set of local completeness statements, is a query Q’ a complete answer to Q?

![Diagram showing Advertised description, True source contents, and Guarantees (LCW; Inter-source comparisons)]
Using LCW rules to minimize plans

Basic Idea:
--If reformulation of Q leads to a union of conjunctive plans
  \[ P_1 \cup P_2 \cup \ldots \cup P_k \]
--then, if \( P_1 \) is “complete” for Q (under the given LCW information), then
we can minimize the reformulation by pruning \( P_2 \ldots P_k \)
  -- [\( P_1 \rightarrow \text{LCW} \)] contains \( P_1 \cup P_2 \cup \ldots \cup P_k \)

[Duschka, AAAI-97]

For Recursive Plans (obtained when the sources have access restrictions)
--We are allowed to remove a rule \( r \) from a plan \( P \), if the “complete”
version of \( r \) is already contained in \( P-r \)

Emerac [Lambrecht & Kambhampati, 99]
Example

S_1 • Movie(title, director, year) (complete after 1960).

S_2/3 • Show(title, theater, city, hour)
  • Query: find movies (and directors) playing in Seattle:
    Select m.title, m.director
    From Movie m, Show s
    Where m.title=s.title AND city="Seattle"
  • Complete or not?

Q(t,d) :- M(T,D,Y) & Sh(T,Th,C,H) & C="Seattle" ==Query

Q’(t,d) :- S_1(T,D,Y) & S_2(T,Th,C,H) & C="Seattle" ==Plan1

Q’’(t,d) :- M(T,D,Y) & S_2(T,Th,C,H) & C="Seattle" ==Plan2

[Levy, 96; Duschka, 97; Lambrecht & Kambhampati, 99]
Quantitative ways of modeling inter-source overlap

- Coverage & Overlap statistics [Koller et. al., 97]
  - $S_1$ has 80% of the movies made after 1960; while $S_2$ has 60% of the movies
  - $S_1$ has 98% of the movies stored in $S_2$

- Computing cardinalities of unions given intersections

Who gives these statistics?
- Third party
- Probing
Query Optimization Challenges

-- Deciding what to optimize

--Getting the statistics on sources

--Doing the optimization
What to Optimize

• Traditional DB optimizers compare candidate plans purely in terms of the time they take to produce all answers to a query.
• In Integration scenarios, the optimization is “multi-objective”
  – Total time of execution
  – Cost to first few tuples
    • Often, the users are happier with plans that give first tuples faster
  – Coverage of the plan
    • Full coverage is no longer an iron-clad requirement
      – Too many relevant sources, Uncontrolled overlap between the sources
    • Can’t call them all!
  – (Robustness,
  – Access premiums…)
Source Statistics Needed

- The size of the source relation and attributes;
  - The length and cardinality of the attributes;
  - The cardinality of the source relation;
- The feasible access patterns for the source;
- The network bandwidth and latency between the source and the integration system
- Coverage of the source S for a relation R denoted by P(S|R)
  - Overlap between sources P(S₁..Sₖ | R)
Getting the Statistics

• Since the sources are autonomous, the mediator needs to actively gather the relevant statistics
  – Learning bandwidth and latency statistics
    • [Gruser et. al. 2000] use neural networks to learn the response time patterns of web sources
      – Can learn the variation of response times across the days of the week and across the hours of the day.
    – Learning coverages and overlaps
      • [Nie et. al. 2002] use itemset mining techniques to learn compact statistics about the spread of the mediator schema relations across the accessible sources
        – Can trade quality of the statistics for reduced space consumption
Learning Coverage/Overlap Statistics

Challenge: Impractical to learn and store all the statistics for every query.

StatMiner: A threshold based hierarchical association rule mining approach

- Learns statistics with respect to “query classes” rather than specific queries
  - Defines query classes in terms of attribute-value hierarchies
  - Discovers frequent query classes and limits statistics to them
- Maps a user’s query into it’s closest ancestor class, and uses the statistics of the mapped class to estimate the statistics of the query.
- Handles the efficiency and accuracy tradeoffs by adjusting the thresholds.

Havasu [Nie et. al. 2002]
Approaches for handling multiple objectives

- Do staged optimization
  - [Information Manifold] Optimize for coverage, and then for cost
- Do joint optimization
  - Generate all the non-dominated solutions (Pareto-Set)
  - Combine the objectives into a single metric
    - e.g. [Havasu/Multi-R]
      » Cost increases additively
      » Coverage decreases multiplicatively

\[
\text{utility}(p) = w \cdot \log(\text{coverage}(p)) - (1-w) \cdot \text{cost}(p)
\]

» The logarithm ensures coverage additive[Candan 01]
Staged Optimization of Cost & Coverage

- Information Manifold ([IM])
  - [Levy et. al; VLDB96]
  - Cartesian product of buckets followed by “containment” check
  - There can be $m^n$ distinct plans
- Ranking and choosing top N plans [Doan et. al; ICDE02]
- Finding a physical plan for each selected plan

Problem: Cost and Coverage are interrelated..
Joint Optimization of Cost & Coverage

- Havasu/Multi-R [Nie et. al. 2001]
  - Search in the space of “parallel” plans
    - Each subplan in the parallel plan contains (a subset of) sources relevant for a subgoal
  - Dynamic programming is used to search among the subgoal orders
  - Greedy approach is used to create a subplan for a particular subgoal
    - Keep adding sources until the utility (defined in terms of cost and coverage) starts to worsen
  - Capable of generating plans for a variety of cost/coverage tradeoffs

Increasing relative weight of coverage

- utility(p) = w*\log(\text{coverage}(p)) - (1-w)\text{cost}(p)
Techniques for optimizing response time for first tuples

- Staged approach: Generate plans based on other objectives and post-process them to improve their response time for first-k tuples
  - Typical idea is to replace asymmetric operators with symmetric ones
    - e.g. replace nested-loop join with symmetric hash join
      - e.g. Telegraph, Tukwila, Niagara
  - Problem: Access limitations between sources may disallow symmetric operations
  - Solution: Use joint optimization approach (e.g. Havasu) and consider the cost of first tuples as a component of plan utility
    - [Viglas & Naughton, 2002] describe approaches for characterizing the “rate” of answer delivery offered by various query plans.
Integrating Services

- Source can be “services” rather than “data repositories”
  - Eg. Amazon as a composite service for book buying
  - Separating line is somewhat thin
- Handling services
  - Description (API;I/O spec)
    - WSDL
  - Composition
    - Planning in general
  - Execution
    - Data-flow architectures
      - See next part
Plan Execution
Executing Plans

• Problem
  • Information gathering plans can be slow (seconds to execute)

• Why?
  • Unpredictable network latencies
  • Varying remote source capabilities
  • Thus, execution is I/O-bound

• Complicating factor: binding patterns
  • During execution, many sources cannot be queried until a previous source query has been answered
Streaming Dataflow for Efficient Plan Execution

Dataflow language & execution

- Dataflow computers go back to the late 60's
- An alternative to the Von-Neumann model
  - Von-Neumann: instruction counter drives execution
  - Dataflow: presence of data drives execution

Benefits

- Parallelism
  - Dataflow
  - Streaming
- Asynchronous execution
Dataflow vs Von-Neumann

\[((a + b) \times (c + d))\]
Dataflow for Information Gathering

- Information Gathering Plans
  - Plan is a dataflow graph (nodes and edges)
    - Operator nodes = dataflow actors
  - Operators produce and consume data
    - Producer/consumer relationships = dataflow arcs
  - Operators "fire" when information becomes available

![Diagram showing dataflow for information gathering](attachment:diagram.png)
Streaming Dataflow

Dataflow-style execution

- Operators execute when inputs become available
  - In contrast, von Neumann style machines use an instruction counter to schedule execution
- Optimizes **horizontal parallelism**
  - Plan is as parallel as its data dependencies allow

Data pipelining

- Data in the system represented as *relations*
  - Operators pipeline *tuples* to consumer
- Optimizes **vertical parallelism**
  - Multiple operators can work on same relation concurrently
Theseus

- A **plan language** and **execution system** for Web-based information integration
  - Expressive enough for monitoring a variety of sources
  - Efficient enough for near-real-time monitoring

![Diagram of Theseus Executor with input data and a plan]

```plaintext
PLAN myplan {
  INPUT: x
  OUTPUT: y
  BODY {
    Op (x : y)
  }
}
```
TheaterLoc Application

Client Web Browser

HTTP

Information Agent

Showtimes Wrapper

Restaurant Wrapper

Theaters Wrapper

Trailer Wrapper

Map Wrapper

Geocoder Wrapper
EXAMPLE: TheaterLoc

- For a given city
  - Locate restaurants and theaters
  - List them & plot on a map
- Web sources
  - Yahoo, CuisineNet, ETAK, USGS
Example: TheaterLoc

- For a given city
  - Combine restaurants + theaters
  - Geocode + plot them on a map

- Web sources:
  - Yahoo, CuisineNet, ETAK, USGS

```
Example: TheaterLoc

- For a given city
  - Combine restaurants + theaters
  - Geocode + plot them on a map

- Web sources:
  - Yahoo, CuisineNet, ETAK, USGS
```
TheaterLoc Plan (Data flow)

<table>
<thead>
<tr>
<th>NAME</th>
<th>ADDRESS</th>
<th>CITY</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>187 Maxella</td>
<td>Venice</td>
<td>CA</td>
</tr>
<tr>
<td>AMC Movies</td>
<td>191 Maxella</td>
<td>Venice</td>
<td>CA</td>
</tr>
</tbody>
</table>

EOS

```sql
SELECT street, city, state FROM Restaurants
UNION
SELECT street, city, state FROM Theaters
PROJECT
street, city, state
JOIN
street=street
city=city
state=state
```

WRAPPER Geocoder

JOIN
street=street
city=city
state=state

WRAPPER TigerMap
Plan Definition

PLAN theaterloc
{
  INPUT: city
  OUTPUT: geolocations, map_url

  BODY
  {
    wrapper ("cuisinenet", "name, addr", city : restaurants)
    wrapper ("yahoo_movies", "name, addr" city : theaters)
    union (restaurants, theaters : places)
    project(places, "street, city, state" : addresses)
    wrapper ("geocoder", "name, lat, lon", addresses : latlons)
    join (places, latlons, "name=name" : geolocations)
    wrapper ("tigermap", geolocations : map_url)
  }
}
Expressivity

- Basic relational-style operators
  - Select, Project, Join, Union, etc.

- Operators for gathering Web data
  - Wrapper
    - Database-like access to a Web source
  - Xquery, Rel2Xml, and Xml2Rel
    - Enables better integration with XML sources

- Operators for monitoring Web data
  - DbExport, DbQuery, DbAppend, DbUpdate
    - Facilitates the tracking of online data
  - Email, Phone, Fax
    - Facilitates asynchronous notification
Expressivity

- Operators for extensibility
  - **Apply**: single-row functions (e.g., UPPER)
  - **Aggregate**: multi-row functions (e.g., SUM)

- Operators for conditional plan execution
  - **Null**: Tests and routes data accordingly

- Subplans and recursion
  - Plans are named and have INPUT & OUTPUT
    - We can use them as operators (subplans) in other plans
  - Subplans make recursion possible
    - Makes it easy to follow arbitrarily long list of result pages that are each separated by a NEXT page link
  - Subplans encourage modularity & reuse
Adaptive Query Execution

- **Network Query Engines**
  - Tukwila
    - Operator reordering
    - Optimized operators
  - Telegraph
    - Tuple-level adaptivity
  - Niagara
    - Partial results for blocking operators

- **Agent Execution Language**
  - Theseus
    - Speculative execution
Tukwila – Interleaved Planning and Execution

- Generates initial plan
- Can generate partial plans and expand them later
- Uses rules to decide when to reoptimize

From Ives et al., SIGMOD’99

```
WHEN end_of_fragment(0)
IF card(result) > 100,000
THEN re-optimize
```
**Tukwila – Adaptive Double Pipelined Hash Join**

From Ives et al., SIGMOD’99

**Hybrid Hash Join**
- No output until inner read
- Asymmetric (inner vs. outer)

**Double Pipelined Hash Join**
- Outputs data immediately
- Symmetric
- More memory
Tukwila – Dynamic Collector Op

- Smart union operator
- Supports
  - Timeouts
  - Slow sources
  - Overlapping sources

From Ives et al., SIGMOD’99

```
WHEN timeout(CustReviews) DO activate(NYTimes), activate(alt.books)
```
Telegraph (Hellerstein et al. 2000)

- Tuple-level adaptivity
- **Rivers** (optimize horizontal parallelism)
  - Adaptive dataflow on clusters (i.e., data partitioning)
- **Eddies** (optimize vertical parallelism)
  - Leverage commutative property of query operators to dynamically route tuples for processing
Telegraph – When can processing order be changed?

- **Moment of symmetry:**
  - Inputs can be swapped without state management
  - Nested Loops: at the end of each inner loop
  - Merge Join: any time
  - Hybrid Hash Join: never!

From Avnur & Hellerstein, SIGMOD 2000
**Telegraph – Beyond Reordering Joins**

**Eddy**
- A pipelining tuple-routing iterator (just like join or sort)
- Adjusts flow adaptively
  - Tuples flow in different orders
  - Visit each op once before output
- Naïve routing policy:
  - All ops fetch from eddy as fast as possible
  - Previously-seen tuples precede new tuples

From Avnur & Hellerstein, SIGMOD 2000
Execution with partial results
[Shanmugasundaram et al. 2000]

- Niagara uses partial results to reduce the effects of blocking operators
  - Reduces blocking nature of aggregation or joins

- Basic idea
  - Execute future operators as data streams in, refine as slow operators catch up
  - Execution is still driven by availability of real data
  - Notion of refinement is similar to "correction" in speculative execution
Speculative Execution in Theseus (Barish & Knoblock 2002)

- Augment plan with two special operators
  - **Speculate**: Issues predictions upon hints
  - **SpecGuard**: Enforces correctness

```
Retrieve restaurants

Retrieve theaters

Union

Retrieve geocodes

Join

Retrieve map

locations

city

hint

answer

Speculate

confirmation

SpecGuard
```
Speculative Execution

- Perform likely computation tasks in advance
  - Even though **committal** of instructions must be in order, **execution** can be out of order...
    - Makes better use of idle CPU
    - Guessing is better than doing nothing at all
      - Often, there are reasonable guesses that could be made

![Graph showing optimized elapsed time (seconds) for different operations: fetch map, fetch latitude and longitude, combine them, fetch theaters & restaurants.](image)
Speedups beyond 2

- Cascading speculation
  - Speculation on speculation

- Functional dependencies
  - Enable early confirmation because subsequent FD processing is deterministic
Learning What to Speculate On (Barish & Knoblock 2002)

- **Decision trees** (novel hints)
  - Identify key features of hints that predict data
- **Transducers** (novel hints, novel predictions)
  - Describe how can hint be translated into prediction

```plaintext
Classification
(Knuth, Art Vol 1)
(Knuth, Art Vol 2)

Transduction
CSCO
http://x.com?sym=cSCO
```
Classification

- Decision Trees
  - Identifies most useful hint

**Computer Science?**

```
Author
  /    \
Donald Knuth  Stephen King

Publisher
  /    \
MIT Press  Bantam

Yes       No
```
Transduction

Views prediction as **translation**

- **INPUT** = Venice CA
- **OUTPUT** = http://example.com?city=VENICE&state=CA

- Determines **alignment** between hint & prediction

- Uses a **transducer** to convert hint into prediction
Discussion

- Theseus, Tukwila, Telegraph, Niagara are all:
  - Streaming dataflow systems
  - Targeting network-based query processing
    - Large source latencies
    - Unknown characteristics of sources
  - Proposed various techniques for improving the efficiency of processing data
    - More efficient operators (e.g., double-pipelined join)
    - Tuple-level adaptivity
    - Partial results for blocking operators
    - Speculative execution
The X-standards...

- **XML**: an on-the-wire representation for data
  - **Xquery**: a query language for XML
  - **Xschema/DTD**: a schema description language for XML data
- **RDF**: a language for meta-data description
- **WSDL/SOAP/UDDI**: languages for describing services
HTML vs. XML

<h1> Bibliography </h1>
<p> <i> Foundations of Databases </i>  
Abiteboul, Hull, Vianu  
Addison Wesley, 1995  
</p>
<p> <i> Data on the Web </i>  
Abiteoul, Buneman, Suciu  
Morgan Kaufmann, 1999  
</p>

<biblization>
<book>  
<title> Foundations… </title>  
<author> Abiteboul </author>  
<author> Hull </author>  
<author> Vianu </author>  
<publisher> Addison Wesley </publisher>  
<year> 1995 </year>  
</book>  
...
</biblization>

“Self-describing”  
-Schema info part of the data  
-Good for data exchange  
(albeit baroque for storage)
XML Terminology

- tags: book, title, author, ...
- elements are nested
- empty element: `<red></red>` abbrv. `<red/>`
- an XML document: single root element

well formed XML document: if it has matching tags
Why are Database folks so excited about XML?

- XML is just a syntax for (self-describing) data
- This is still exciting because
  - No standard syntax for relational data
  - With XML, we can
    - Translate any legacy data to XML
    - Can exchange data in XML format
      - Ship over the web, input to any application
XML vs. Relational Data

- XML is meant as a language that supports both Text and Structured Data
  - Conflicting demands...
- XML supports *semi-structured data*
  - In essence, the schema can be union of multiple schemas
    - Easy to represent books with or without prices, books with any number of authors etc.
- XML supports free mixing of text and data
  - using the #PCDATA type
- XML is *ordered* (while relational data is *unordered*)
Querying XML

• Requirements:
  – Need to handle lack of schema.
    • We may not know much about the data, so we need to navigate the XML.
  – Need to support both “information retrieval” and “SQL-style” queries.
    • Ordered vs. un-ordered XML
  – “Human readable”
    • like SQL? 😊

• Candidates
  – Many… based on conflicting requirements
    • XSL: Makes IR folks happy
    • XML-QL: Makes DB folks happy
    • Xquery: W3C’s attempt to make everybody (un)happy
Example Query

Query

```
<bib>
  { for $b in /bib/book
      where $b/publisher = "Addison-Wesley"
      and $b/@year > 1991
      return <book year={ $b/@year }>
        { $b/title }
      </book> }
</bib>
```

“For all books after 1991, return with Year changed from a tag to an attribute”

Result

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
  </book>
  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
  </book>
</bib>
```
Example Query (2)

• Return the books that cost more at amazon than fatbrain

Let $amazon :=
  document(http://www.amazon.com/books.xml),
Let $fatbrain :=
  document(http://www.fatbrain.com/books.xml)
For $am in $amazon/books/book,
  $fat in $fatbrain/books/book
Where $am/isbn = $fat/isbn
  and $am/price > $fat/price
Return <book>{ $am/title, $am/price, $fat/price }
  </book>
Impact of XML on Integration

If and when all sources accept Xqueries and exchange data in XML format, then
- Mediator can accept user queries in Xquery
- Access sources using Xquery
- Get data back in XML format
- Merge results and send to user in XML format

• How about now?
- Sources can use XML adapters (middle-ware)
XML middleware for Databases

- XML adapters (middle-ware) received significant attention in DB community
  - SilkRoute (AT&T)
  - Xperanto (IBM)
- Issues:
  - Need to convert relational data into XML
    - Tagging (easy)
  - Need to convert Xquery queries into equivalent SQL queries
    - Trickier as Xquery supports schema querying
    - A single query may be mapped into a union of SQL queries
Is XML standardization a magical solution for Integration?

If all WEB sources standardize into XML format

– Source access (wrapper generation issues) become easier to manage

– BUT all other problems remain

  • Still need to relate source (XML)schemas to mediator (XML)schema
  • Still need to reason about source overlap, source access limitations etc.
  • Still need to manage execution in the presence of source/network uncertainties
“Semantic Web”

• The LAV/GAV approaches assume that some human expert will do the actual schema mapping
• The “semantic-web” initiative attempts to automate schema mapping
  – Idea: Allow pages to write logical axioms relating their vocabulary (tags) to other external tags
  – Support automatic inference of relations between source and mediator schema using these rules
    • DAML+OIL
XML ≠ machine accessible meaning

This is what a web-page in natural language looks like for a machine
XML ≠ machine accessible meaning

XML allows “meaningful tags” to be added to parts of the text

<education>

<work>

<private>

林克昌 拒留台湾 可能増高

在受惠及熱心奔走之下，華裔名指揮家林克昌拒留台灣的可能性又提升了很多。兩廳院總監李天、國家音樂廳副局長黃奕明日前親赴林克昌、石靈芳寓所拜會，並提出多項善待條件。此間，台灣省立交響樂團團長陳維詠也早早「下書」，邀請林克昌赴台中聖會，從八月十日起到九月廿，為期長達一個月。

在台灣諸多公家樂團中，陳維詠是以實際行動表達對林克昌肯定的樂界人士之一，曾多次公開表示對林克昌指揮才華的欣賞，而且幾乎每個樂季都邀請林克昌客席演出。

此間，林克昌上個月赴俄羅斯與頂尖的「俄羅斯國家管弦樂團」灌錄了柴可夫斯基現代總計三首交響曲以及「羅密歐與朱麗葉」、「斯拉夫進行曲」、「義大利隨想曲」，最後的DAT母帶也在前兩天寄回台灣。製作人楊忠衡與林克昌試聽之後，都對錄音效果一音樂表現非常感到十分滿意。楊忠衡估計呈現了三分林克昌指揮神韻。

俄羅斯國家管弦樂團首席布魯尼日前也指稱林克昌的指揮藝術有三大特色：一是控制自如的彈性速度；二是強烈的動態對比；三是宛如呼吸歌唱的旋律處理。這些對錄音師而言都構成很大挑戰。俄羅斯錄音師雖然採用多軌混音，但定位、場面都有可觀之處。
XML ≠ machine accessible meaning

But to your machine, the tags look like this....
XML ≠ machine accessible meaning

Schemas help....

...by relating common terms between documents
But other people use other schemas

Someone else has one like this…. 
But other people use other schemas

...which don’t fit in

Moral: There is still need for ontology mapping..
--More on this from Craig Knoblock
Ontology and Data Integration

Integrating Ontologies/Schemas from Different Sources
Schema/Ontology Integration

- Integration of data at the schema (or ontology) level
- Requires resolving differences in the organization and naming of the ontologies
- Today problem largely solved with manual tools
  - Tools exist for building data warehouses
- Automatic schema integration tools have focused on using meta information
  - (e.g., attribute names)
- Recent work has begun to explore the combination of meta information and source data
- Rich ontologies and languages are becoming available to support this type of integration
  - (e.g., Cyc, DAML, XML Schema)
Problem: Automated techniques and tools for mapping a source and its corresponding ontology into an existing ontology

**Existing Ontology:**

```
house
  /   \
/     \  
address contact num-baths amenities
  |     |
  |     |
  name  phone
```

**Ontology for new source:**

```
house
  /   \
/     \  
location contact-info full-baths half-baths handicap-equipped
  |     |
  |     |
  agent-name agent-phone
```
Multi-Strategy Learning
Doan, Domingos, Levy, SIGMOD 2000

• Use a set of **base learners**:
  • Name learner, Naïve Bayes, Whirl, XML learner

• And a set of **recognizers**:
  • County name, zip code, phone numbers.

• Each base learner produces a prediction weighted by **confidence score**

• Combine base learners with a **meta-learner**, using **stacking**.
Applying the Learners

**Schema of homes.com**
- area
- day-phone
- extra-info

**Mediated schema**
- address
- price
- agent-phone
- description

Example from [Doan, Domingos, Levy, SIGMOD 2000]

```
<area>Seattle, WA</area>
<area>Kent, WA</area>
<area>Austin, TX</area>
<day-phone>(278) 345 7215</day-phone>
<day-phone>(617) 335 2315</day-phone>
<day-phone>(512) 427 1115</day-phone>
<extra-info>Beautiful yard</extra-info>
<extra-info>Great beach</extra-info>
<extra-info>Close to Seattle</extra-info>
```

```
Name Learner
Naive Bayes
......
Name Learner
Naive Bayes
Meta-Learner
......
Meta-Learner
......
```

```
(address,0.8), (description,0.2)
(address,0.6), (description,0.4)
(address,0.7), (description,0.3)
(address,0.7), (description,0.3)
(agent-phone,0.9), (description,0.1)
(description,0.8), (address,0.2)
```
Ontology and Data Integration

Integrating Data Across Sources
Object Identification (aka Record Linkage)

Problem

• Different sources typically represent and format information differently.
• As a result, determining if two sources are referring to the same object can be difficult.
• Example
  • Is “Joe Cool” the same person as “Joseph B. Cool”?
  • What if they have the same telephone number?
  • What if Joe Cool’s number is 310-322-0730 and Joseph B. Cool’s number is 310-640-2973?
Example Data Integration Problem

- How to align (or join) the objects across different sources

Zagat’s Restaurant Guide Source
- Art’s Deli
- California Pizza Kitchen
- Campanile
- Citrus
- Grill, The
- Philippe The Original
- Spago

Department of Health Restaurant Source
- Art’s Delicatessen
- Ca’ Brea
- CPK
- The Grill
- Patina
- Philippe’s The Original
- The Tillerman
Information Retrieval Approach [Cohen, 1998]

- Follows the same approach used by classical IR algorithms (including web search engines).
- First, “stemming” is applied to each entry.
  - E.g. “Joe’s Diner” -> “Joe [‘s] Diner”
- Then, entries are compared by counting the number of words in common.
- Note: Infrequent words weighted more heavily by TFIDF metric = Term Frequency Inverse Document Frequency
Unsupervised Record Linkage

- Idea: Analyze data and automatically cluster pairs into three groups:
  - Let $R = P(\text{obs} | \text{Same}) / P(\text{obs} | \text{Different})$
  - Matched if $R > \text{threshold } T_U$
  - Unmatched if $R < \text{threshold } T_L$
  - Ambiguous if $T_L < R < T_U$

- This model for computing decision rules was introduced by Felligi & Sunter in 1969

- Particularly useful for statistically linking large sets of data, e.g., by US Census Bureau
Unsupervised Record Linkage (cont.)

- Winkler (1998) used EM algorithm to estimate $P(\text{obs} \mid \text{Same})$ and $P(\text{obs} \mid \text{Different})$
- EM computes the *maximum likelihood estimate*
  - The algorithm iteratively determines the parameters most likely to generate the observed data.
- Additional mathematical techniques must be used to adjust for “relative frequencies”
  - I.e. last name of “Smith” is much more frequent than “Knoblock”.
Supervised Active Learning Approach
[Tejada, Knoblock & Minton, 2001]

• Supervised learning. System learns:
  • Which attributes to weight more heavily:

<table>
<thead>
<tr>
<th>Name</th>
<th>Street</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zagat’s</td>
<td>Art’s Deli</td>
<td>818-756-4124</td>
</tr>
<tr>
<td>Dept of Health</td>
<td>Art’s Delicatessen</td>
<td>818/755-4100</td>
</tr>
</tbody>
</table>

• Transformation rules

Zagat’s       Transformations       Dept of Health

Art’s Deli California Pizza Kitchen Philippe The Original
Prefix Acronym Stemming CPK Philippe’s The Original
Observations:

- Mapping objects can be application dependent
- Example:

<table>
<thead>
<tr>
<th>Steakhouse The</th>
<th>128 Fremont Street</th>
<th>702-382-1600</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binion's Coffee Shop</td>
<td>128 Fremont St.</td>
<td>702/382-1600</td>
</tr>
</tbody>
</table>

- The mapping is in the application, not the data
- User input is needed to increase accuracy of the mapping
### Mapping Rules

#### Set of Similarity Scores

<table>
<thead>
<tr>
<th>Name</th>
<th>Street</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>.967</td>
<td>.973</td>
<td>.3</td>
</tr>
<tr>
<td>.17</td>
<td>.3</td>
<td>.74</td>
</tr>
<tr>
<td>.8</td>
<td>.542</td>
<td>.49</td>
</tr>
<tr>
<td>.95</td>
<td>.97</td>
<td>.67</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Mapping Rules

- Name > .8 & Street > .79 => mapped
- Name > .89 => mapped
- Street < .57 => not mapped
Transformation Weights

• Appropriate transformations depend on the application domain
  • Restaurants, companies, airports...
• ...and on the different attributes within an application
  • Acronym is more appropriate for restaurant name than phone number
• Learn likelihood that if a transformation is applied then two object match

Transformation Weight = \( P(\text{match} | \text{transformation}) \)
Learning Object Mappings

Active Atlas

- **Candidate Generator:**
  - Judge textual similarity of mappings
  - Reduce number of mappings considered for classification

- **Mapping Learner:**
  - Active learning technique to learn mapping rules and transformation weights
  - System chooses most informative example for the user to label
  - Minimize the amount of user interaction
Mapping Rule Learner

Choose initial examples

Generate committee of learners

Learn Rules
Classify Examples
Votes

Learn Rules
Classify Examples
Votes

Learn Rules
Classify Examples
Votes

Choose Example

Set of Mapped Objects

Label

USER

Label
Committee Disagreement

- Chooses an example based on the disagreement of the query committee

<table>
<thead>
<tr>
<th>Examples</th>
<th>Committee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Art’s Deli, Art’s Delicatessen</td>
<td>Yes  Yes  Yes</td>
</tr>
<tr>
<td>CPK, California Pizza Kitchen</td>
<td>Yes  No   Yes</td>
</tr>
<tr>
<td>Ca’Brea, La Brea Bakery</td>
<td>No   No   No</td>
</tr>
</tbody>
</table>

- In this case CPK, California Pizza Kitchen is the most informative example based on disagreement
Review

- Motivation for Information Integration
- Accessing Information Sources
- Models for Integration
- Query Planning & Optimization
- Plan Execution
- Standards for Integration/Mediation
- Ontology & Data Integration
Future Directions

• Promising areas for AI in information integration
  • Planning to Compose Web Services
  • Data mining for aligning ontologies and data
  • Machine learning for automatic wrapper generation
  • Machine learning and natural language processing for extracting and integrating text
  • Constraint satisfaction for information integration
  • ...

Future Directions