Querying Heterogeneous Information Sources Using Source Descriptions

Alon Y. Levy – AT&T Laboratories
Anand Rajaraman – Stanford University
Joann J. Ordille – Bell Labs

Presentation By: Mirza Beg

Outline

Problem Description
Proposed System
System Architecture
Description of System Modules
Algorithms
Experiments & Results
Discussion

Problem Statement

- Increasing number of structured data sources
- Interrelated data
- The user interacts with each information source separately and combine data

Alternatively:
- How do we extract the relevant data for a given query?
Solution

A System that:

- Provides a uniform query interface to distributed structured sources
- Uses source descriptions to describe data sources
- Generates executable query plans
- Returns the merged result set to the user

Information Manifold

Architecture

Information Manifold

World View

- A virtual global schema on which the user can pose queries

Product (Model)
Automobile (Model, Year, Category)
    Car (Model, Year, Category)
        NewCar (Model, Year, Category)
        UsedCar (Model, Year, Category)
    CarForSale (Model, Year, Category, SellerContact)
Motorcycle (Model, Year)
Information Manifold

Source Descriptions

Source Descriptions for Auto Sources

<table>
<thead>
<tr>
<th>Source 1: Used cars for sale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exterior: View front, rear, and side views of car, and exterior photos. Check for paint issues, scratches, and dents.</td>
</tr>
<tr>
<td>Interior: View inside, check for wear and tear, and note any malfunctions.</td>
</tr>
<tr>
<td>Source 2: Luxury cars for sale</td>
</tr>
<tr>
<td>All cars in this database are priced $50,000 or less.</td>
</tr>
<tr>
<td>Source 3: Premium cars for sale</td>
</tr>
<tr>
<td>All cars in this database are priced $100,000 or less.</td>
</tr>
<tr>
<td>Source 4: Luxury cars for sale</td>
</tr>
<tr>
<td>Cars are listed by model and year, with prices starting at $100,000.</td>
</tr>
<tr>
<td>Source 5: Car reviews database</td>
</tr>
<tr>
<td>Includes reviews for cars manufactured after 1990.</td>
</tr>
</tbody>
</table>

Content Records of Auto Sources

<table>
<thead>
<tr>
<th>Source 6: Used cars for sale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exterior: View front, rear, and side views of car, and exterior photos. Check for paint issues, scratches, and dents.</td>
</tr>
<tr>
<td>Interior: View inside, check for wear and tear, and note any malfunctions.</td>
</tr>
<tr>
<td>Source 7: Luxury cars for sale</td>
</tr>
<tr>
<td>All cars in this database are priced $75,000 or less.</td>
</tr>
<tr>
<td>Source 8: Premium cars for sale</td>
</tr>
<tr>
<td>All cars in this database are priced $125,000 or less.</td>
</tr>
<tr>
<td>Source 9: Luxury cars for sale</td>
</tr>
<tr>
<td>Cars are listed by model and year, with prices starting at $75,000.</td>
</tr>
<tr>
<td>Source 10: Car reviews database</td>
</tr>
<tr>
<td>Includes reviews for cars manufactured after 1990.</td>
</tr>
</tbody>
</table>
Capability Records of Auto Sources

Desired Inputs

Possible Outputs

Selection Set

Information Manifold

Plan Generator

Query Reformulation Steps

1. Prune irrelevant sources
2. Split query into sub goals
3. Generate conjunctive query plans
4. Find an executable ordering of sub goals
Step 1. Bucket Algorithm

Given a query Q:
- Find a relevant source
- Create a bucket for this sub-goal
- Check source for Satisfiability
- Add information source to bucket for this sub-goal

Example: Contents and Capabilities

<table>
<thead>
<tr>
<th>Source</th>
<th>Type</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabular</td>
<td>PDF</td>
<td>Contents [C], Capabilities [C], Source [S]</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td>Source [S]</td>
</tr>
<tr>
<td>Database</td>
<td>SQL</td>
<td></td>
</tr>
</tbody>
</table>
Bucket Algorithm: Example

\[ p(x, y, z) = \text{Confidential}(x), \text{Anonymous}(y), \text{Secret}(z), q \geq 1992 \]

\[ \text{Print}(x, y), \text{Modify}(x, y), \text{Delete}(x, y), \text{Confidential}(z), q \geq 1992 \]

Find the mapping \( x \to y \)

\[ \text{Confidential}(x), \text{Anonymous}(y), \text{Secret}(z), q \geq 1992 \]

\[ \text{Print}(x, y), \text{Modify}(x, y), \text{Delete}(x, y), \text{Confidential}(z), q \geq 1992 \]

Source 1 is added to bucket1.

Source 2 is added to bucket1.

Source 3 does not get added because \( q \leq 1992, q \geq 1992 \) is not satisfiable.

Step 2. Finding an Executable Ordering

- Considering all possible combinations of information sources, enumerate semantically correct plans.

Step 2. Algorithm for finding an Executable Ordering

- Maintain a list of available parameters.
- At every point add to the ordering any sub-goal whose input requirements are satisfied.
- Push as many selections as possible to the sources.
Step 3. Checking Containment

- Minimize each plan by removing redundant sub-goals

Experimental Results

<table>
<thead>
<tr>
<th>Query</th>
<th>Number of sources</th>
<th>Max. factor size</th>
<th>Plan generated</th>
<th>Plan generated</th>
<th>Time per plan (sec.)</th>
<th>Total time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0.10</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>3</td>
<td>3</td>
<td>0.10</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>7</td>
<td>3</td>
<td>0.10</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Query 1: Find titles and years of movies featuring Tom Hanks
Query 2: Find titles and reviews of movies featuring Tom Hanks
Query 3: Find telephone number(s) for Alaska Airlines

Experimental Results (cont.)
Conclusions

- A novel system that provides a DB-like query interface to distributed structured information sources
- Frees the user from interacting with each information source individually
- Integrates data from multiple sources and filters information
- Information Manifold applicable to WWW and company-wide d-DB’s

Open Questions

- How to automatically extract contents and capabilities from sources?
- Are there better algorithms to determine the relevant sources?
- Scalability?
- Overall Performance issues?

Discussion Points

- A foundational paper in web-data mining.
- Substantial impact on current integration systems.
- Contents & capabilities at the core of the system yet no proposed generation algorithm.
- Experiments carried out on a very small set of queries.
Questions ?