SINGAPORE: Towards Flexible Querying of Heterogeneous Data Sources

Klaus R. Dittrich
(joint work with Ruxandra Domenig)

Institut für Informatik
der Universität Zürich
Database Technology
Research Group

Winterthurerstrasse 190, CH-8057 Zürich
e-mail: dittrich@ifi.unizh.ch, http://www.ifi.unizh.ch
Tel.: ++41-1-635 4312, Fax: ++41-1-635 6809
Outline

- Motivation
- Integration Approaches
- SINGAPORE
  - Data Space
  - Query Language
  - SINGAPORE Architecture
  - Metadata
  - Query (Pre)processing
  - Source Registration
- Conclusions
Motivation

- ever-increasing volume of data available online
- organized in heterogeneous data sources (of many kinds)
- users searching for information have various aims, but also different knowledge levels

---

we look for flexible integrated access to heterogeneous data sources
**Motivation (cont.)**

**flexible** querying:

- **extensibility**: data sources can join and leave
- allow **various** user groups to satisfy their information needs: **exact** and **fuzzy** queries
  - exact: \~ database queries
  - fuzzy: \~ keyword search (information retrieval)
- **structural heterogeneity**: structured, semistructured and unstructured data

**integrated** search:

- **single access point**
- combine data from various sources
Heterogeneous Systems: Data Transformation

- **homogenization**: translation of the local data models and query languages into the global data model and query language.
- **integration**
Integration Approaches

- **materialized**
  - universal DBMS — data imported from the underlying sources, users work with a new integrated full-fledged database

- **virtual**
  - federated and multi-DBMS — a global (partial) schema is defined
  - mediated query systems — integration implemented by specialized software components (mediators), the integration is more “flexible”

- **integrated schema approach**
  - 😊 database view
  - 😞 full extensibility difficult to achieve

- **mediated query systems: sample approaches**
  - TSIMMIS (tools for generation of mediators and wrappers)
  - Garlic (support for heterogeneous data offering a database view)
  - DIOM (users can define a personal view of the data)
  - DISCO (global view, also solve problems like unavailable data sources)
The SINGAPORE Approach

SINGAPORE
(SINGle Access POint for heterogeneous data REpositories)

- general approach: no a-priori integration
  - no integrated schema, but a unique query interface
  - direct access to sources

- full extensibility
- suitable for any kind of users
  - users can decide where and how to search (exact queries)
  - users with little knowledge about the sources can still formulate queries (fuzzy queries)

- however: integration task shifted towards user
Data Space

- defines which (sets of) data can be queried
  - data stored in a source regarded as a class extension
  - unions of extensions
  - specified on four levels:
    - source schemas
    - source location
    - data or content type
    - classification roots
  
  to be done (dynamically) by administrator
Query Language

- supports querying of structured, semistructured and unstructured data sources
  - take a structured query language as a starting point (OQL)
  - extend it with features for unstructured and semistructured data
  - for unstructured data:
    - CONTAINS ( ... )
      
      ```
      SELECT * FROM unstructured
      WHERE CONTAINS ("database")
      ```
    - CONTAINSRELATED ( ... ) for “similar” information
Query Language (cont.)

- for semistructured data (examples):
  - navigation in the „semi“-structure (regular path expressions)
    
    \[
    \text{address.*.zip\_code} \quad \longrightarrow \quad \text{address.zip\_code} \\
    \qquad \quad \text{address.location.zip\_code}
    \]
  - type coercion
    
    \[
    \text{SELECT * FROM addr.address WHERE zip\_code = '8001'} \\
    (\text{zip\_code is declared as INTEGER in Addr})
    \]
  - use of non-rigid structure when querying: LIKE ( ... )
    
    \[
    \text{SELECT LIKE (name) FROM addr} \\
    \qquad \downarrow \\
    \quad \text{SELECT address.location.loc\_name FROM addr}
    \]
  - all these operators apply for the entire data space
SINGAPORE Architecture

- **User Interface**
  - User level
  - User level

- **Metadata Repository**
  - Core query processing
  - Query preprocessing
  - Source registration
  - Result combination

- **Wrapper Manager**
  - Source level
  - Source level

- **Wrappers**
  - Query
  - Source data
  - Metadata
Metadata

- supports users when formulating queries
  - learning about the data space

- supports processing of queries
  - processing special operators and inconsistencies in queries
  - integration of data

- contains
  - structure, query capabilities and result type of data sources
  - descriptive information about the sources
Metadata (Integration Information)

- Integration information about similarities

- For each source and element in a source, define a Description and a Related_To field ("soft" metadata):

```
<DataSource><Name>D1</Name>
  <Description>Database about employees of the CS department since 1980</Description>
  <Related_To>D2</Related_To>
</DataSource>
```
Metadata (Integration Information)

- integration information about conflicts

  - structural conflicts
    - for table PERSON: Name and Surname
    - for class AUTHOR: AuthName
      - use the “soft” metadata

  - descriptive conflicts
    - naming conflicts
      - for table PERSON: Name, for table EMPLOYEE: EmpName
      - use the “soft” metadata and LIKE (…)
    - scaling conflicts
      - `<Scale><From>km</From><To>m</To>
        <Through>1 km = 1000 m</Through>`
      - `</Scale>`
**Query Preprocessing**

- **why?**
  - for a flexible way of querying and for (partly) solving the heterogeneity problem
  - maps “fuzzy” queries into “exact” ones

- comprises the following steps:
  1. processing of special operators in SOQL
  2. generation of paths for structure specifications in the query
  3. generation of *consistent* queries
Query Preprocessing (cont.)

- **step 1:**
  - e.g. processing of `LIKE(path)`
    - keyword search in the data space
    - the data space is indexed (Porter algorithm, Wordnet)
    - special algorithm for searching paths

- **step 2:** path generation

- **step 3:**
  - *consistent queries*: queries for which the SELECT, FROM and WHERE parts correspond to each other
  - each newly generated consistent query is semantically equivalent to the initial query
### Query Preprocessing (Example)

<table>
<thead>
<tr>
<th>input</th>
<th><code>SELECT photo FROM structured WHERE LIKE(name) = &quot;Meier&quot;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>step 1: operator processing</td>
<td><code>SELECT photo FROM structured WHERE &lt;name, fname, iname&gt; = &quot;Meier&quot;</code></td>
</tr>
<tr>
<td>step 2: path generation</td>
<td><code>SELECT D1.person.photo FROM &lt;D1.person, D2.institute&gt; WHERE &lt;D1.person.name, D1.person.fname, D2.institute.iname&gt; = &quot;Meier&quot;</code></td>
</tr>
<tr>
<td>step 3: consistent queries</td>
<td><code>SELECT D1.person.photo FROM D1.person WHERE D1.person.name = &quot;Meier&quot; OR D1.person.fname = &quot;Meier&quot;</code></td>
</tr>
</tbody>
</table>
Source Registration

- each wrapper is started as a process which communicates with the server
- a system administrator enters metadata for a given source into the metadata repository
- the server connects the wrappers (using TCP/IP communication) and submits appropriate queries, waiting for results
- the wrappers return results in XML format back to the server
Conclusions

- SINGAPORE implements a flexible approach for integrated access to structurally heterogeneous data sources
  - no a-priori, “beforehand” integration
  - exact and fuzzy queries

- challenges:
  - query model and language
  - metadata (for query formulation and processing)
  - query processing (combination of database and information retrieval techniques, optimization)
Current and recent projects at DBTG

general remarks

- basic research (sponsored by own resources, SNF etc.)
- applied research (sponsored by national special programs, EU programs, companies; cooperation with e.g. SMEs, UBS, Swiss Life, ABB, ...)
- carried out by Ph.D. assistants and postdocs
- prototypes
- mostly middleware-oriented
Current and recent projects at DBTG

advanced DBMS features & architecture

- active (object-oriented) DBMS
- component DBMS
- generalized event engine
Current and recent projects at DBTG

information integration technology

- federated DBMS
  - access control/version management/integrity management
  - schema integration using ontologies
  - dynamic component systems

- data warehousing
  - DWH refreshment
  - metadata management

- integrated querying

- workflow management

- enterprise content management