Query Set Specification Language (QSSL)

Michalis Petropoulos
Alin Deutsch
Yannis Papakonstantinou

UNIVERSITY OF CALIFORNIA, SAN DIEGO

June 2003
Exporting DBMSs on the Web

- Exporting Query Capabilities on the Web
  - Web Services (Function Signatures)
- Integrating Web Applications
  - Use Web Services
  - Export Query Capabilities Themselves
Overview

- **Query Set Specification Language (QSSL)**
  - Describes Parameterized Tree Pattern (TP) Queries
- **Data Services**
  - Web Services for Query Capabilities
- **Authoring Interface**
Motivation for QSSL (I)

- Any combination of the following conditions on:
  - the name of the airline company
  - the origin and destination of one or more flights (optional)
  - a day of the week
  - the origin of zero or more legs (optional)
  - the destination of zero or more legs (optional)
  - the aircraft used for zero or more legs (optional)

- The queries may return “airline” or “flight” elements
Motivation for QSSL (II)

- Web Services published as function signatures:
  - Fixed number of input and output parameters
  - Do not capture the functionality of databases
  - Large number of web services needed
    - One function signature for every parameterized query
  - Do not capture the semantic connections the available functions have with each other and with the underlying databases

- JDBC Interfaces
  - All possible queries
Query Set Specification Language

Query Language

- Tree Pattern Queries:
  - Acyclic XPath expressions consisting of:
    - node tests
    - child axis ‘/’
    - descendant axis ‘//’
    - predicates ‘[]’
  - Widely used in current applications
  - Building blocks of XQuery
  - Excellent visual paradigm for GUIs

flights/airline[name='Delta']/flight[from='JFK'][to='LAX'][day='MON'][leg[to='LAS']]
Query Set Specification Language

Query Set Specification

\[ f_1 \rightarrow \text{flights} \rightarrow \text{airline} \]
\[ f_2 \rightarrow \text{name} = \#1 \]
\[ f_3 \rightarrow \text{flight} \]
\[ f_4 \rightarrow \text{from} = \#2 \rightarrow \text{to} = \#3 \]
\[ f_5 \rightarrow \text{day} = \#4 \]
\[ f_6 \rightarrow \text{leg} \]
\[ f_8 \rightarrow \text{to} = \#6 \]

Query:
- flights
- airline
- flight
- name = \#1
- from = \#2
- to = \#3
- day = \#4
- leg
- to = \#6

Airline: Delta
- from = LAX
- to = JFK
- day = MON

Day: MON
- leg
- to = LAS
Query Set Specification Language

Query Set Specification

- Similar to extended context-free grammars
Query Set Specification Language

Recursive XML Schemas
Query Set Specification Language

Recursive XML Schemas

• QSS of fixed size
Data Services

- **Data Service** = WSDL + QSS
- A QSS deployed as a web service
- Exports the XML Schema of an XML view
- Connects the WSDL calls with the underlying database
- Receives queries that are encoded by QSS
- Explicit relationship between input and output
Data Services

- QSS is translated to XML Schema (QSSX)
- TP queries that are encoded as XML (TPX)
- Query result is described by an XML Schema
Data Services

Reasoning

1. Membership of a query in a data service
2. Subsumption of data services
3. Totality of a data service
4. Overlap of data services

- QSS can be translated to an equivalent top-down nondeterministic unranked tree automaton
- Problems are decidable
Data Services

Authoring Interface
Future Work

Capability-Based Rewriting

User Query
TPX

Rewriting Processor

Candidate Queries

Plan Processor

Containing Result Remainder Query

Execution Engine

Query Result
XML

Data Source

flights
airline
name=\#1

flight +

from \#3
to

? day

flights
airline
name = ‘Delta’

flight

from = ‘JFK’
to = ‘LAX’
day = ‘MON’

flights
airline
name = ‘Delta’

flight

from = ‘LAX’
to = ‘JFK’
day = ‘MON’
Questions and Answers
Query Set Specification Language

Example Derivation

\[ f_1 \Rightarrow \text{flights} \quad \Rightarrow \quad \text{flights} \quad \Rightarrow \quad \text{flights} \]

\begin{align*}
&f_2 \\
&f_3 + \\
&\text{airline} \\
&\text{name}=#1 \\
&f_4 ? \\
&f_5 \\
&f_6 * \\
&\text{flight} \\
&\text{from}=#2 \quad \text{to}=#3 \quad \text{day}=#4 \\
&\text{leg} \\
&f_7 ? \\
&f_8 ? \\
&f_9 ?
\end{align*}
Related Work

Capability-Based Rewriting

- Capabilities described as binding patterns
  - Adornments on view attributes
  - Negative approach
- Expansions of Datalog programs
  - Recursive programs \(\rightarrow\) Infinite queries
  - Positive approach