Exporting and Interactively Querying Web Service-Accessed Sources: The CLIDE System

Michalis Petropoulos

Database Seminar, February 2010

Large-Scale Data Integration Systems

What queries can the mediator answer for me? CLIDE

Running Example

Parameterized Views

Conjunctive Queries CQ
  + Equality & Comparison Conditions
  + Parameters
Running Example

Integrated Schema

- Integrated schema puts together the Dell and Cisco schemas

Attribute Associations
- (Computers.cid, NetCards.cid)
- (NetCards.rate, Routers.rate)
- (NetCards.standard, Routers.standard)

Sophisticated Mediators Make Feasible Queries Hard to Predict

Feasible Queries FQ
- Equivalent CQ query rewritings using the views
- Might involve more than one views
- Order might matter

Query: Feasible
Get all {Wireless Routers}, together with their NetCards and their compatible 'Wireless' Routers

The CLIDE Solution

- A query formulation interface, which interactively guides the developer toward feasible queries by employing a coloring scheme
QBE-Like Interfaces

Microsoft SQL-Server

CLIDE Interface

• Table, selection, projection and join actions
• Feasibility Flag
• Color-based suggestions

Example Interaction

Snapshot 1

Yellow ➔ required action
- All feasible queries require this action

White ➔ optional action
- Feasible queries can be formulated w/ or w/o these actions

Snapshot 2

Blue ➔ required choice of action
- At least one feasible query cannot be formulated unless this action is performed

ComByCpu ('P4')
cid
cpu
ram
price

A123
P4
512
400

B123
P4
1024
550

C

ram
price
512
400
1024
550
**Example Interaction**

**Join Lines:**
- Only yellow and blue are displayed
- Must appear in Attribute Associations

**Example Interaction**

- \( \Rightarrow \) any other constant
- \( \textbf{Red} \) prohibited action
  - Does not appear in any feasible query
  - Lead to "Dead End" state

**Example Interaction**

**Demo**
CLIDE Properties

- **Completeness of Suggestions**
  - Every feasible query can be formulated by performing yellow and blue actions at every step

- **Summarization of Suggestions**
  - At every step, only a minimal number of actions is suggested, i.e., the ones that are needed to preserve completeness

- **Rapid Convergence By Following Suggestions**
  - The shortest sequence of actions from a query to any feasible query consists of suggested actions

Interaction Graph

- Nodes are queries: One for each $q \in \text{CQ}$
- Edges are actions: Table, selection, projection and join actions
- Green nodes are feasible queries
- Infinitely big structure
  - All CQ queries
  - All possible combinations of actions formulating them

Interaction Graph: Colorable Actions

- Colorable actions $\alpha_c$ label outgoing edges of the current node

Interaction Graph: Colors

- **Yellow action $\alpha$**
  - Every path from current node $n$ to a feasible node contains $\alpha$

- **Blue action $\alpha$**
  - At least one feasible query cannot be formulated unless this action is performed (summarization)

- **Red action $\alpha$**
  - No path to a feasible node contains $\alpha$
CLIDE Architecture

- Back-End invoked every time the user performs an action
  - i.e., the user arrives at a new node in the interactions graph

Closest Feasible Queries Algorithm

- Color Algorithm
- Parameters Algorithm
- Closest Feasible Queries Algorithm
  - Aliases Collapse Rule
  - Maximally-Contained Rewriter

Maximally Contained Queries $FQ_{MC}$

- Assuming fixed SELECT clause (projection list)
- Covered extensively in literature
  - MiniCon, Bucket, InverseRules Algorithms
- $FQ_{MC}$ is finite

Closest Feasible Queries $FQ_C$ Algorithm

- Compute maximally contained queries $FQ_{MC}$
- **Theorem:** All $FQ_C$ queries are reachable via a path of length $p \leq p_L$
  - The radius $p_L$ is the longest path to a maximally contained query
Closest Feasible Queries $FQ_C$ Algorithm

**Challenge:** Find the Closest Feasible Queries

- **Theorem:** All queries in $FQ_{MC}$ are in $FQ_C$.
- But not all queries in $FQ_C$ are in $FQ_{MC}$.

Solution: Collapse Aliases

- Collapse Aliases to compute $FQ_C \setminus FQ_{MC}$.
- Check satisfiability.

Color Algorithm

**Yellow and Blue**
- An action $\alpha$ is colored based on which closest feasible queries it appear in.
  - Yellow, if $\alpha$ appears in all queries in $FQ_C$.
  - Blue, if $\alpha$ appears in at least one (but not all) query in $FQ_C$.

**White and Red**
- Attach Maximum Projection Lists to Closest Feasible Queries
  - Projections that can be added to a feasible query, without compromising feasibility.
- Projection $\alpha$ is white if in the maximum projection list.
- Color selections based on projections.

CLIDE Implementation & Optimizations

- Views expansion introduce redundancy.
  - Affects CLIDE's rapid convergence and summarization.
- Efficient containment test crucial to redundancy removal.
CLIDE Performance

Chains of Stars – No Parameters

- **Queries**
  - A-span = 7
  - B-span = 3
  - Selections = 4, 6, 8, 10

- **Schema**

- **Views**

Chains of Stars – With Parameters

- **Queries**
  - A-span = 7
  - B-span = 3
  - Selections = 4, 6, 8, 10

- **Schema**

- **Views**

**Chains of Stars – No Parameters**

**Chains of Stars – With Parameters**
CLIDE Summary

First interactive query formulation interface based on source and mediator capabilities

Applicability
• Service-Oriented Architectures
• Privacy-Preserving Services

Contributions
• Interaction Guarantees: Rapid Convergence, Completeness, Summarization of Suggestions
• Interaction Graph
• Back-End Algorithms
  - Closest Feasible Queries, Colors, Parameters
• Modular, Customizable Architecture

http://www.clide.info