Mediator-based Information Systems: Garlic

Garlic is a middleware system that provides an integrated view of a variety of legacy data sources, without changing how or where data is stored\(^1\). Key components of the Garlic architecture are wrappers that encapsulate foreign data sources. Wrappers mediate between the data source and the Garlic middleware and participate in query planning and execution.

### Schema of the geographic data source

```java
interface Country {
    attribute string name;
    attribute string airlines_served;
    attribute boolean visa_required;
}

interface City {
    attribute string name;
    attribute long population;
    attribute boolean airport;
    attribute Country country;
}
```

### Schema of the hotel data source

```java
interface Hotel {
    attribute readonly string name;
    attribute readonly short category;
    attribute readonly double daily_rate;
    attribute readonly string location;
    attribute readonly string city;
}
```

Table 1: Sample Garlic Schema Definitions

Table 1 shows sample Garlic schema definitions for two data sources. Assume that each of these sources is encapsulated by a Garlic wrapper. Say, Garlic is asked to retrieve five star hotels close to the beach in Greek towns with less than 500 inhabitants. Describe the query planning process, the resulting query plan, and the query execution process for the following wrapper capabilities!

```sql
SELECT h.name, h.daily_rate
FROM hotel h, country c, cities i
WHERE h.class = 5 AND h.location = 'beach' AND c.name = 'Greece' AND i.population < 500 AND h.city = i.name AND i.country = c.oid
```

---

\(^1\) M. T. Roth, P. Schwarz, Don’t Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, VLDB 97
1) The geographic wrapper and the hotel wrapper are unable to evaluate predicates. The geographic wrapper does not support joins, i.e. neither the `plan_join()` method nor the `plan_bind()` method are implemented.

**Work Request for the hotel wrapper**

<table>
<thead>
<tr>
<th>Work Request</th>
<th>Wrapper Access Plan for hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: <code>h.oid</code>, <code>h.name</code>, <code>h.city</code>, <code>h.daily_rate</code>, <code>h.class</code>, <code>h.location</code></td>
<td>Project: <code>h.oid</code>, <code>h.name</code>, <code>h.city</code>, <code>h.daily_rate</code>, <code>h.class</code>, <code>h.location</code></td>
</tr>
<tr>
<td>Predicates: <code>h.class = 5</code>, <code>h.location = 'beach'</code></td>
<td>Predicates:</td>
</tr>
</tbody>
</table>
2) The hotel wrapper cannot handle equality predicates on strings because it does not adhere to SQL semantics for string equality. However, it treats the predicate `location = 'beach'` as `location LIKE '%beach%'`, which provides a superset of the results of the equality predicate. The geographical wrapper supports local joins, i.e. the `plan_join()` method is implemented while the `plan_bind()` method is not.

**Work Request for the hotel wrapper**

<table>
<thead>
<tr>
<th>Work Request</th>
<th>Wrapper Access Plan for hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: h.oid, h.name, h.city, h.daily_rate, h.class, h.location</td>
<td>Project: h.oid, h.name, h.city, h.daily_rate, h.class, h.location</td>
</tr>
<tr>
<td>Predicates: h.class = 5, h.location = 'beach'</td>
<td>Predicates: h.class = 5</td>
</tr>
</tbody>
</table>

**Work Request for the geographic wrapper**

<table>
<thead>
<tr>
<th>Work Request</th>
<th>Wrapper Access Plan for countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: c.oid, c.name</td>
<td>Project: c.oid, c.name</td>
</tr>
<tr>
<td>Predicates: c.name = 'Greece'</td>
<td>Predicates: c.name = 'Greece'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Work Request</th>
<th>Wrapper Access Plan for cities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: i.oid, i.name, i.population, i.country</td>
<td>Project: i.oid, i.name, i.population, i.country</td>
</tr>
<tr>
<td>Predicates: i.population &lt; 500</td>
<td>Predicates: i.population &lt; 500</td>
</tr>
</tbody>
</table>
Work Request

Input Plans

Wrapper Access Plan for countries

Project: c.oid, c.name

Predicates: c.name = 'Greece'

Wrapper Access Plan for cities

Project: i.oid, i.name, i.population, i.country

Predicates: i.population < 500

Join Predicates: i.country = c.oid

Wrapper Join Plan for countries and cities

Project: c.oid, c.name, i.oid, i.name, i.population, i.country

Predicates: c.name = 'Greece', i.population < 500, i.country = c.oid
3) The hotel wrapper is able to evaluate equality predicates and the geographic wrapper supports both, joins and bind joins.

<table>
<thead>
<tr>
<th>Work Request for the hotel wrapper</th>
<th>Wrapper Access Plan for hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong> h.oid, h.name, h.city, h.daily_rate, h.class, h.location</td>
<td><strong>Project:</strong> h.oid, h.name, h.city, h.daily_rate, h.class, h.location</td>
</tr>
<tr>
<td><strong>Predicates:</strong> h.class = 5, h.location = ‘beach’</td>
<td><strong>Predicates:</strong> h.class = 5, location = ‘beach’</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Work Request for the geographic wrapper</th>
<th>Wrapper Access Plan for countries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong> c.oid, c.name</td>
<td><strong>Project:</strong> c.oid, c.name</td>
</tr>
<tr>
<td><strong>Predicates:</strong> c.name = ‘Greece’</td>
<td><strong>Predicates:</strong> c.name = ‘Greece’</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Work Request for cities</th>
<th>Wrapper Access Plan for cities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong> i.oid, i.name, i.population, i.country</td>
<td><strong>Project:</strong> i.oid, i.name, i.population, i.country</td>
</tr>
<tr>
<td><strong>Predicates:</strong> i.population &lt; 500</td>
<td><strong>Predicates:</strong> i.population &lt; 500</td>
</tr>
</tbody>
</table>
Middleware for Heterogeneous and Distributed Information Systems

Work Request

Input Plans

Wrapper Access Plan for countries
Project: c.oid, c.name
Predicates: c.name = 'Greece'

Wrapper Access Plan for cities
Project: i.oid, i.name, i.population, i.country
Predicates: i.population < 500
Join Predicates: i.country = c.oid

Wrapper Join Plan for countries and cities
Project: c.oid, c.name, i.oid, i.name, i.population, i.country
Predicates: c.name = 'Greece', i.population < 500, i.country = c.oid

Work Request

Input Plan

Wrapper Join Plan for countries and cities
Project: c.oid, c.name, i.oid, i.name, i.population, i.country
Predicates: c.name = 'Greece', i.population < 500, i.country = c.oid
Bind Predicates: i.name = $BIND_1

Wrapper Bind Plan for countries, cities, and hotles
Project: c.oid, c.name, i.oid, i.name, i.population, i.country
Predicates: c.name = 'Greece', i.population < 500, i.country = c.oid, i.name = $BIND_1
Filter
- h.class = 5 AND location = 'beach'

Project
- h.name, h.daily_rate

Bind Join
- h.city $\Rightarrow$ BIND_1

City
- i.population < 500

Country
- c.name = 'Greece'

Join
- i.country = c.oid

Filter
- h.class = 5 AND location = 'beach'

Hotel
- h.name, h.daily_rate