

Wrappers and legacy systems

Presentation
Michal Oglodek
2007



Research papers

- IBM Almaden Research Center: „A Wrapper Architecture for Legacy Data Sources” Mary Tork Roth

- „Software Agents”

Michael R. Genesereth
Logic Group
Computer Science Department
Stanford University

Steven P. Ketchpel
Computer Science Department
Stanford University



Wrappers for legacy systems

- Demand: Interoperability
- Problem: Heterogeneity
- How to deal with legacy systems? –
Wrappers
 - Agents
 - Web Service
 - Other ?



Agent Wrappers

- Wrapper as translator between agents and legacy system
- This ensures:
 - Agent communication protocols respected
 - Legacy systems decoupled from agents



Possible interactions with legacy software

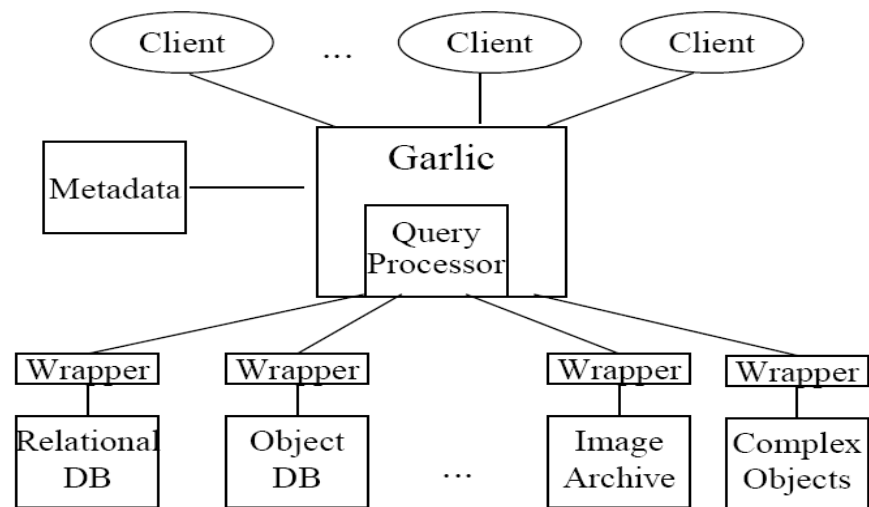
- Transducer
- Wrapper injecting code
- Rewriting of the legacy software



IBM Almaden Research Center

- „Garlic” – middleware system that provides an integrated view of a variety of legacy data sources
- 10 datasources wrapped already
- Architecture for quick „wrapping” of other legacy systems

Garlic architecture





Garlic interface

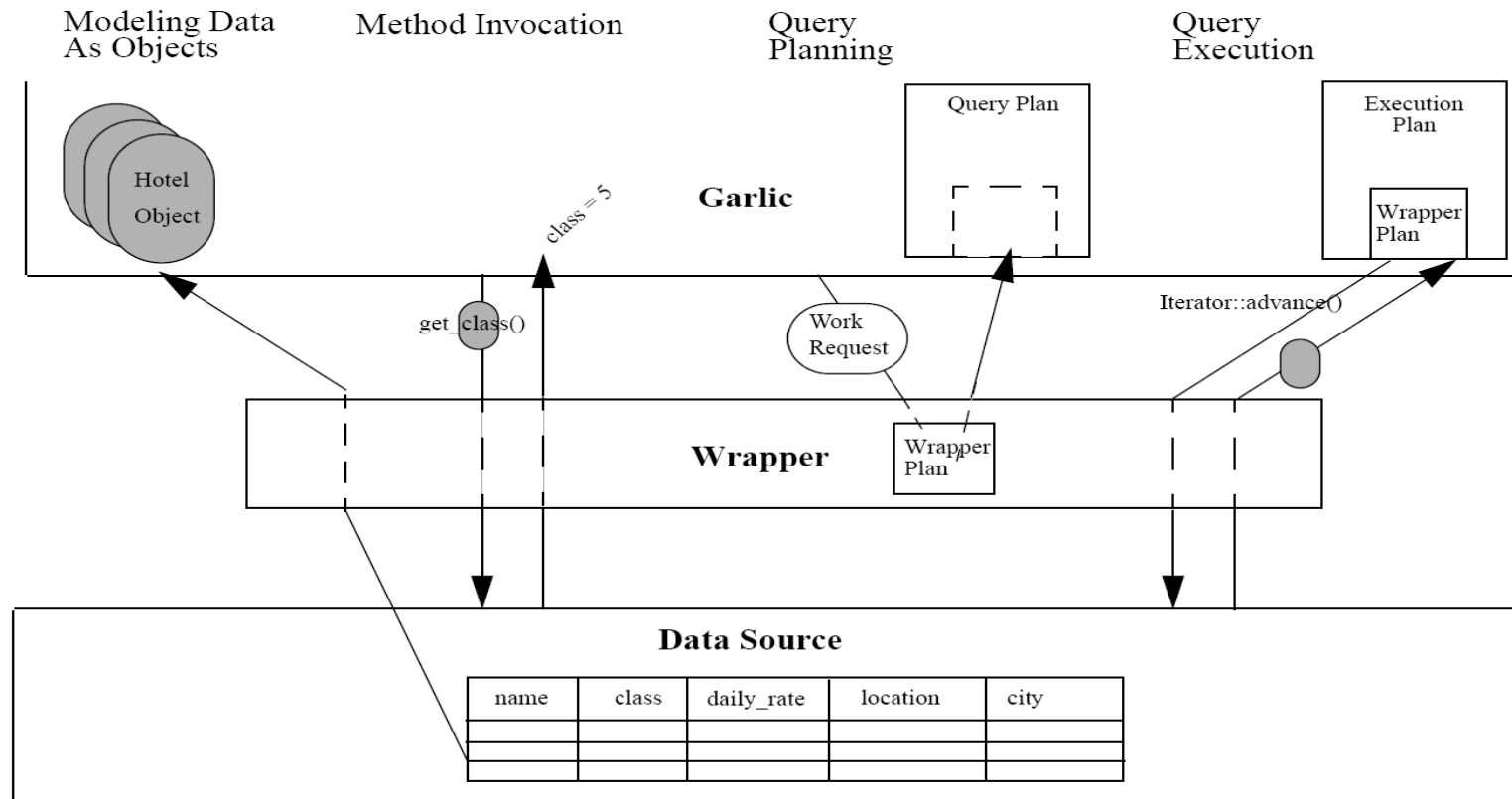
- C++ programming interface
- Garlic's query language (extension of SQL)
 - Path expressions
 - Nested collections
 - methods



Goals for the Wrapper architecture

- *The start-up cost to write a wrapper should be small*
- *Wrappers should be able to evolve.*
- *The architecture should be flexible and allow for graceful growth.*
- *The architecture should readily lend itself to query optimization.*

Services provided by Wrapper





Example Wrappers for Simple Travel Agency

■ 1. Modelling Data as Objects

- Wrapper turns data underlying of underlying into objects accessible by Garlic (interface – abstract objects behaviour, implementation – realization of interface)
- Garlic Object ID (OID)
 - IID – implementation identifier
 - Key – uninterpreted by Garlic



Example – cont.

Data used by Travel Agency

- Our Travel agency:
 - Countries / Cities info – relational DB
 - Access to website with booking information
 - Images of destinations in Image Server



Example – cont

Modelling Data as Objects

Travel Agency Application schema
Interfaces provided by wrappers

<p>Relational Repository Schema</p> <pre>interface Country { attribute string name; attribute string airlines_served; attribute boolean visa_required; attribute Image scene; }</pre>	<p>Web Repository Schema</p> <pre>interface Hotel { attribute readonly string name; attribute readonly short class; attribute readonly double daily_rate; attribute readonly string location; attribute readonly string city; }</pre>
<pre>interface City { attribute string name; attribute long population; attribute boolean airport; attribute Country country; attribute Image scene; }</pre>	<p>Image Server Repository Schema</p> <pre>interface Image { attribute readonly string file_name; double matches(in string file_name); void display(in string device_name); }</pre>



Example – cont

Method Invocation

■ Stub dispatch

```
Iterator<Gstruct> *result_iter;
Gstruct result_tuple;
...
oql(db, result_iter,
    "select C.scene from Countries C where C.airlines_served LIKE '%American%'");
while (result_iter->next(result_tuple)) {
    Image *i = result_tuple[0].get_reference();
    i->display(myScreen);
}
```

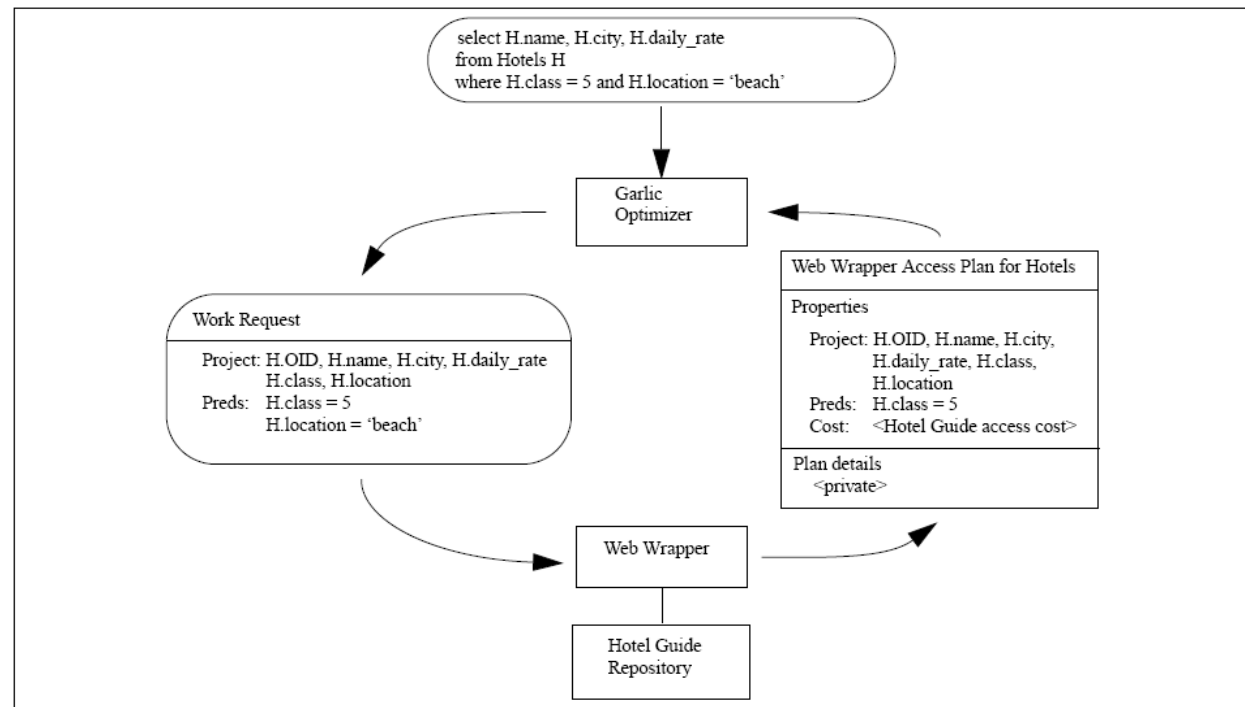
■ Generic dispatch

```
select population
from Cities
where name = <OID key value for name> and country = <OID key value for country>
```

Example –cont

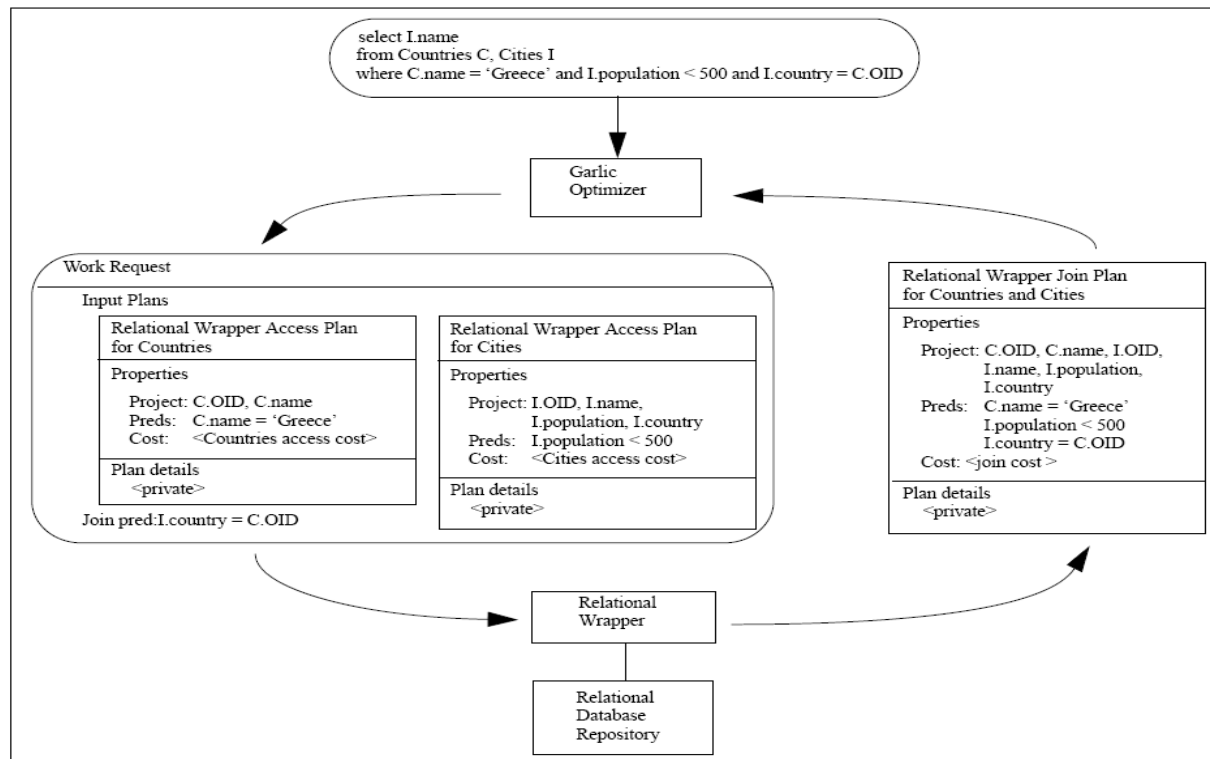
Query Planning

- Construction of a Wrapper Access Plan
- „Garlic user wants to find 5-star hotels near beach



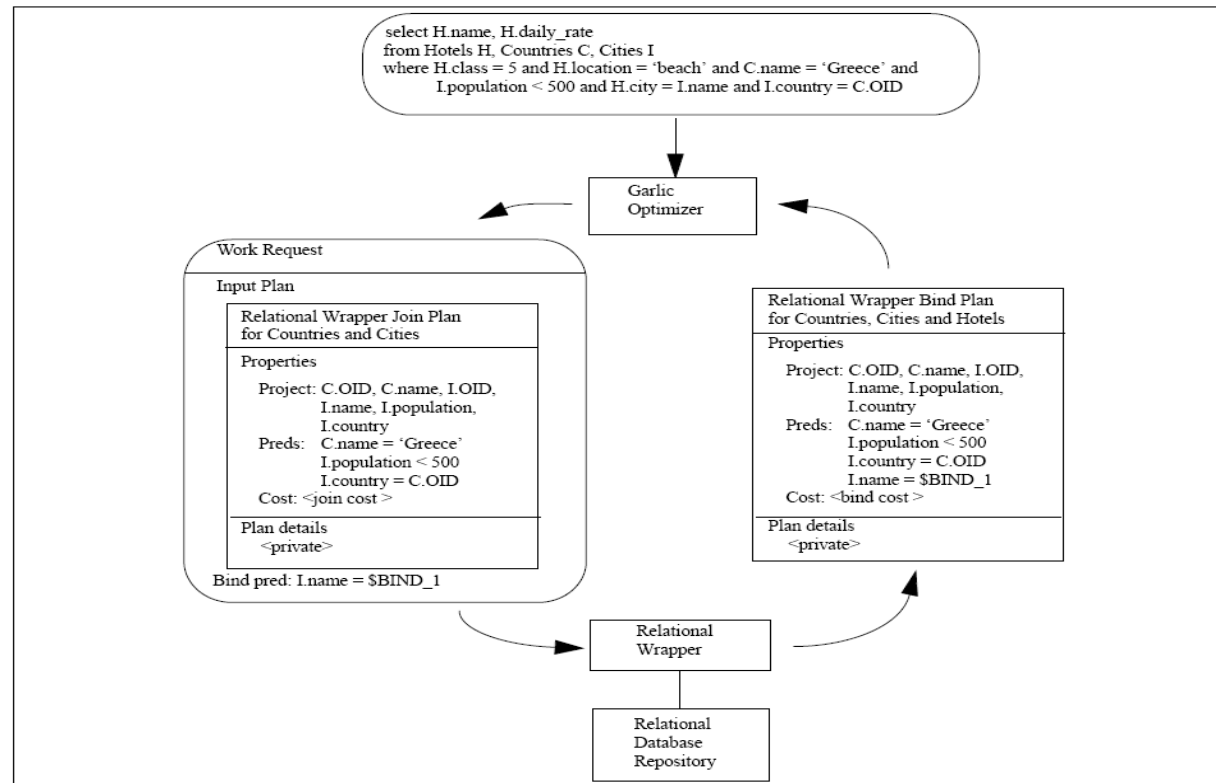
Example – cont

Constr. Of a Wrapper Join Plan



Example – cont

Construction of Bind Plan





Example – cont

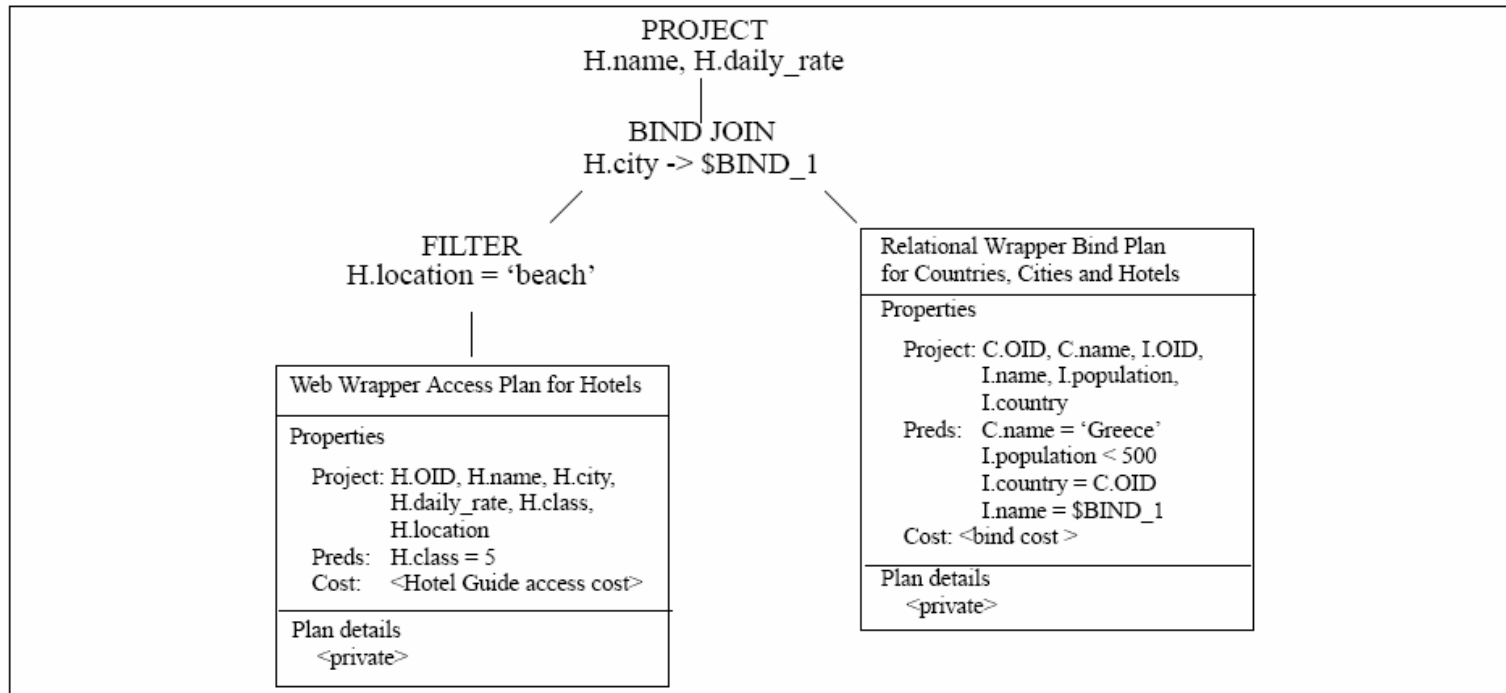
Query Execution

- Wrapper's final service:
 - Plan translation
 - Query execution
- Garlic query plan (tree of operators)
 - FILTER
 - PROJECT
 - JOIN
- Wrapper plans as leaves of the tree

Example – cont

Query Execution - cont

■ Plan for a Garlic Query



Current Status - 10 data sources wrappers implemented

Data Source	Schema description	Method invocation	Query operations handled by wrapper
DB2, Oracle	Columns of a relation map to attributes of an interface; relations become collections of objects; primary key value of a tuple is key for OID	accessor methods only; generic dispatch	general expression projections, all basic predicates, joins, bind joins, joins based on OID
Searchable web sites: http://www.hotelguide.ch , a hotel guide, and http://www.bigbook.com , a directory of U.S. business listings	Each web site exports a single collection of listing objects; HTML page data fields map to attributes of an interface; unique key for a listing provided by web site is key for OID	accessor methods only; generic dispatch	attribute projection, equality predicates on attributes, LIKE predicates of the form '%<value>%'
Proprietary database for molecular similarity search	A single collection of molecule objects; interface has contains_substructure() and similarity_to() methods to model search capability of engine; molecule l-number is key for OID	stub dispatch	attribute and method projection, predicates of the form <attr> <op> <const> and <method> <op> <const>, where <op> is a comparison operator
QBIC image server that orders images according to color, texture and shape features	Collections of image objects; interface contains matches() method to model ordering capability; image file name is key for OID	stub dispatch	ordering of image objects by image feature
Glimpse[19] text search engine that searches for specific patterns in text files	Collections of files; interface contains several methods to model text search capability and retrieve relevant text of a file; file name is key for OID	stub dispatch	projection of attributes and methods
Lotus Notes databases: Phone Directory database, Patent Server database	Notes database becomes a collection of note objects; interface defined by database Form; note NOTEID is key for Garlic OID	accessor methods only; generic dispatch	attribute projection, predicates containing logical, comparison and arithmetic operators, LIKE predicates, tests for NULL.
Complex Object Wrapper	Collections of objects; interface corresponds to interface of objects in database; database OID is key for Garlic OID	stub dispatch	attribute projection