Information Integration: the MOMIS Project Demonstration

D. Beneventano[1,2], **S. Bergamaschi**[1,2], **A. Corni**[1,2], **R. Guidetti**[1], **G. Malvezzi**[1] and **M. Vincini**[1]
(1) Università di Modena e Reggio Emilia
DSI - Via Campi 213/B, 41100 Modena
(2) CSITE-CNR Bologna
V.le Risorgimento 2, 40136 Bologna
e-mail : {domenico.beneventano,sonia.bergamaschi,corni.alberto}@unimo.it
{guidetti,malvezzi,vincini}@dsi.unimo.it

1 Overview

The goal of this demonstration is to present the main features of a Mediator component, *Global Schema Builder* of an I3 system, called MOMIS (Mediator envirOnment for Multiple Information Sources) [1]. MOMIS has been conceived to provide an integrated access to heterogeneous information stored in traditional databases (e.g., relational, object-oriented) or file systems, as well as in semistructured sources. The demonstration is based on the integration of two simple sources of different kind, structured and semi-structured, wich will be described in Section 2.
Like other integration projects [2, 3], MOMIS follows a "semantic approach" to information integration based on the conceptual schema, or metadata, of the information sources, and on the following functional elements:

1. a common data model, ODM_{I^3}, which is defined according to the ODL_{I^3} language, to describe source schemas for integration purposes. ODM_{I^3} and ODL_{I^3} have been defined in MOMIS as subset of the corresponding ones in ODMG, following the proposal for a standard mediator language developed by the I^3/POB working group [4]. In addition, ODL_{I^3} introduces new constructors to support the semantic integration process;

2. one or more wrappers, to translate metadata descriptions into the common ODL_{I^3} representation;

3. a mediator which is composed of two modules: the *Global Schema Builder* (GSB) and the *Query Manager* (QM). The GSB module processes and integrates ODL_{I^3} descriptions received from wrappers to derive the integrated representation of the information sources. The QM module performs query processing and optimization. In particular, it generates the OQL_{I^3}[1] queries for wrappers, starting from a global OQL_{I^3} query formulated by the user on the global schema. Using Description Logics techniques, the QM component can generate in an automatic way the translation of the global OQL_{I^3} query into different sub-queries, one for each involved local source.

2 Demonstration

2.1 Running example

In order to illustrate the way our approach works, we will use the following example of integration in the Restaurant Guide domain. Consider two different datasources that collect information about restaurants. The `Eating Datasource` guidebook (`ED`) contains semistructured objects about restaurants of the west coast and their menu, quality, ... Fig. 1 illustrates a portion of the data (we use a notation similar to the one of the OEM model [5, 6]). We use the notion of *object pattern* to represent all different objects that describe the same concept in a given semistructured source. Object patterns for all the objects in our semistructured source are shown in Fig. 2 (the symbol "*" denotes "optional" labels). Three object patterns are defined: `Restaurant` containing information about restaurants; `Owner` containing information about people involved and `Address`. Each `Restaurant` has an atomic `name`, `category` and `specialty`. Furthermore, some `Restaurant` have an atomic `address` and some other a complex `address`, a `phone`, a complex object `nearby`, that specifies the nearest restaurant, and `owner`, that indicates the `name`, the `address` and the `job` of the restaurant's owner.

[1] OQL_{I^3} is a subset of OQL-ODMG.

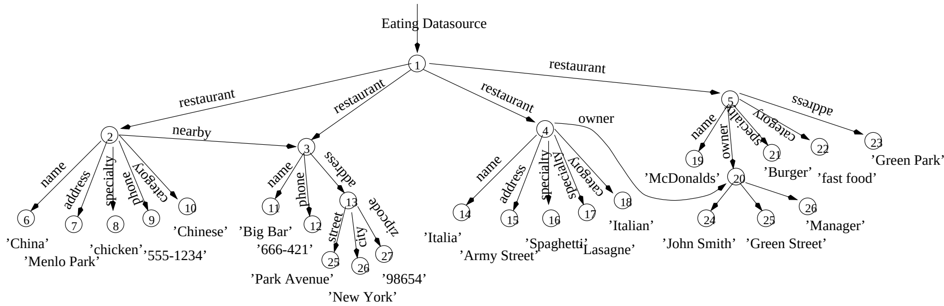


Figure 1: Eating Datasource (ED)

Restaurant-pattern = (`Restaurant`,{`name`,`address`, `phone`*,`specialty`,`category`,`nearby`*,`owner`*})
Owner-pattern = (`Owner`,{`name`,`address`,`job`})
Address-pattern = (`Address`,{`street`,`city`,`zipcode`})

Figure 2: The object patterns for the ED source

The `Food Guide` Database (`FD`) is a relational database containing information about USA restaurants. There are four relations: `Steakhouse`, `Bistro`, `Person`, and `Brasserie` (see Fig. 3). Information related to restaurant is maintained into the `Steakhouse` relation. `Bistro` instance is a subset of `Steakhouse` instance and contains information about the small informal restaurants that serve wine. Each `Steakhouse` and `Bistro` is managed by a `Person`. Information about places where drinks and snacks are served on, are stored in `Brasserie` relation.

2.2 Demonstration Architecture

Global Schema Builder (GSB) is the Mediator component which processes and integrates ODL_{I^3} descriptions received from wrappers to derive the integrated representation of the information sources, i.e. the Global Virtual Schema. It is composed mainly by a GUI (the SI-Designer module), a data repository and coordination module (GlobalSchema module) and a set of services (service level) used during the integration (see figure 4). All such modules are available as CORBA objects and interact using established `idl` interfaces. Data sources to be integrated are reachable by *wrapper modules* that are also CORBA object (with a very simple common interface).
The *Designer* performs the integration process in a semi-automatic way, following the steps suggested by the (**SI-Designer**). Each step is characterized by a graphical form (see figure 5) and each form "talk directly" with the *GlobalSchema* object (the `idl` interface between *GlobalSchema* and *SI-Designer* is strictly modular) retrieving data and saving new information provided by the Designer in the Common Thesaurus, a common ontology among sources.
For the integration phase, GSB uses the following services:

- SIM (*Source Integrator Module*): extracts intra-schema intensional relationships on the basis of the source structures;
- SLIM (*Schemata Lessical Integrator Module*): extracts inter-schema intensional relationships between attribute and class names, exploiting the Wordnet lexical system [7]; In this case, synonyms, hypernyms/hyponyms, and related terms can be automatically proposed to the designer, by selecting them according to relationships

Food Guide Database (FD)

```
Steakhouse(s_code, name, street, pers_id, special_dish)
Bistro(s_code, type, pers_id)
Person(pers_id, first_name, last_name, qualification)
Brasserie(b_code, name, address)
```

Figure 3: Food Guide Database (FD)

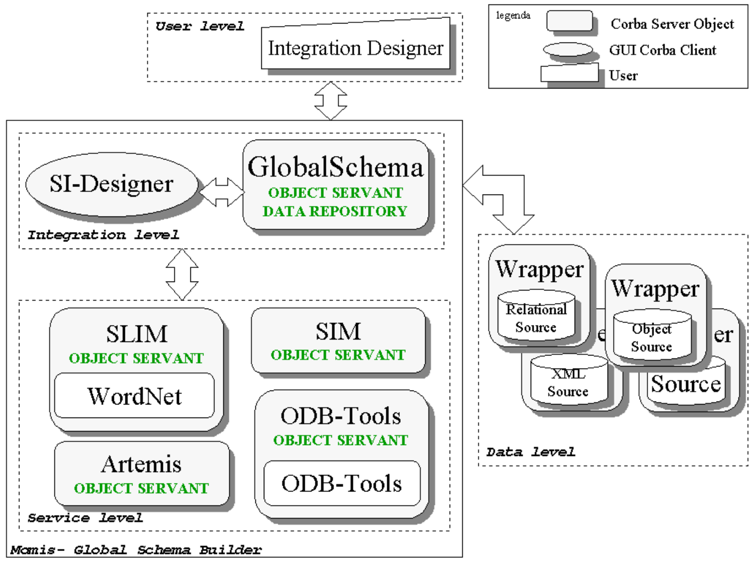


Figure 4: Demonstration architecture

predefined in the lexical system.

- ARTEMIS, Terminological relationships in the Common Thesaurus are used by ARTEMIS to assess the level of *affinity* between ODL_{I^3} classes by interactively computing the affinity coefficients. ODL_{I^3} classes with affinity are automatically classified using hierarchical clustering techniques [8].
- ODB-Tools, a tool based on the OLCD Description Logics [9] inference techniques, such as *incoherence* detection and *subsumption* computation, which performs ODL_{I^3} schema validation and evaluates implicit inter-schema *isa* relationships;

The integration process is subdivided in two phases (**1**) *Common Thesaurus* generation, (**2**) global classes generation. The sequence of interactions to build the global schema is the following:

- SIM(extracts intra-schema relationships);
- SLIM (extracts inter-schema intensional relationships between attribute and class names, exploiting the Wordnet lexical system [7]);
- ARTEMIS computes *affinity* coefficients between ODL_{I^3} classes.

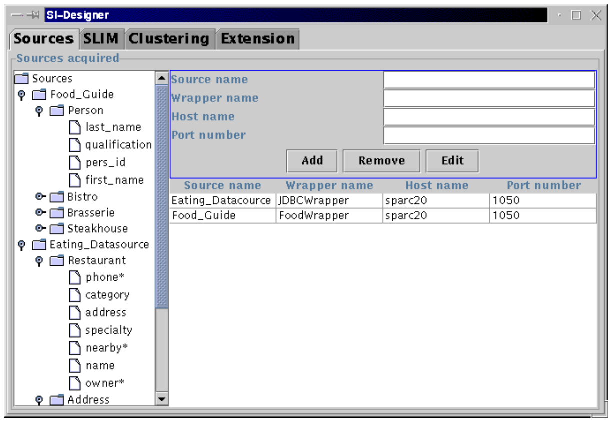
At each interaction the extracted relationships are shown to the designer which can confirm or not them and provide further information.

When the *Common Thesaurus* has been built, SI-Designer uses again the ARTEMIS module to individuate, by a clustering algorithm, the disjuncted set of the classes with an*affinity* threshold value: i.e. the *clusters*. Affinity clusters of ODL_{I^3} classes are interactively selected in ARTEMIS and passed to ODB-Tools to construct the Global Virtual Schema of the Mediator; an integrated global ODL_{I^3} class is interactively defined for each selected cluster. ODB-Tools is exploited for a semi-automatic generation of the global ODL_{I^3} classes. The set of global ODL_{I^3} classes defined constitutes the global schema of the Mediator to be used for posing queries against the sources.

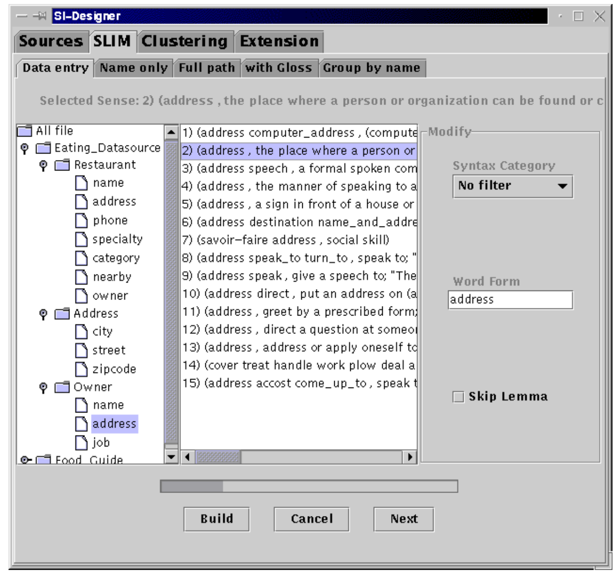
The **GlobalSchema** is the information repository and act as coordination object for a integration session, for each *integration* exists a GlobalSchema object. Such object is characterized by a *status* that spaces between the value *uninitialized* to the value *complete* when the global schema is completely modeled. A GlobalSchema object will be supplied as input to a future *Query Manager* object that will manage queries on the integrated schema.

References

[1] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Records*, 28(1), March 1999.



(a)



(b)

Figure 5: Example: (a) Source binding interface and (b) SLIM interface

[2] Y. Arens, C.Y. Chee, C. Hsu, and C. A. Knoblock. Retrieving and integrating data from multiple information sources. *International Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.

[3] M.T. Roth and P. Scharz. Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proc. of the 23rd Int. Conf. on Very Large Databases*, Athens, Greece, 1997.

[4] P. Buneman, L. Raschid, and J. Ullman. Mediator languages - a proposal for a standard, April 1996. Available at ftp://ftp.umiacs.umd.edu/pub/ONRrept/medmodel96.ps.

[5] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. The lorel query language for semistructured data. *Journal of Digital Libraries*, 1(1), 1996.

[6] Y.Papakonstantinou, H.Garcia-Molina, and J.Widom. Object exchange across heterogeneous information sources. In *Proc. of ICDE95*, Taipei, Taiwan, 1995.

[7] A.G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[8] S. Castano and V. De Antonellis. A schema analysis and reconciliation tool environment for heterogeneous databases. In *IEEE Proc. of IDEAS'99 Int. Database Engineering and Applications Symposium*, Montreal, 1989.

[9] D. Beneventano, S. Bergamaschi, S. Lodi, and C. Sartori. Consistency checking in complex object database schemata with integrity constraints. *IEEE Transactions on Knowledge and Data Engineering*, 10:576–598, July/August 1998.