# A Knowledge-Based Approach to Querying Heterogeneous Databases

M. Andrea Rodríguez and Marcela Varas

Department of Information Engineering and Computer Science
University of Concepción,
Edmundo Larenas 215, Concepción, Chile.
andrea@udec.cl,mvaras@inf.udec.cl

**Abstract.** Query processing plays a fundamental role in current information systems that need to access independent and heterogeneous databases. This paper presents a new approach to querying heterogeneous databases that maps the semantics of query objects onto database schemas. The sematics is captured by the definitions of classes in an ontology, and a similarity function identifies not only equivalent but also semantically similar classes associated with a user's request. These similar classes are then mapped onto a database schema, which is compared with schemas of heterogeneous databases to obtain entities in the databases that answer the query.

## 1 Introduction

New approaches to knowledge-based retrieval have highlighted the use of ontologies and semantic similarity functions as a mechanism for comparing objects that can be retrieved from heterogeneous databases [1,2,3,4]. Ontologies aim to capture the semantics of a domain through concept definitions [5], which are used as primitives of a query specification and as primitives of resource descriptions. In current knowledge-based information systems, accessing information involves a semantic matching between users' requests and stored data. In environments with multiple and heterogeneous databases, this semantic matching is predicated on the assumption that independent databases share the same ontology or agree to adopt an ontology derived from the integration of existing ones [1,2,4]. But, given the need to query heterogeneous databases that use different conceptualizations (i.e., different ontologies), we need to modify the single-ontology paradigm of semantic matching for information access.

We present an approach to querying heterogeneous databases based on ontologies and similarity evaluations. We start at the top-level with users' requests expressed by terms defined in a *user ontology*. In this context, a user ontology provides terms definitions concerning a given domain [6]. By using such an ontology we can capture a richer semantics in the users' requests, and we allow users to express their queries without the need to know the schemas of data representation.

The scope of this work is the retrieval of information described by classes of objects. For example, consider the following query to a Geographic Information System (GIS): “retrieve *utilities* in Atlanta, Georgia.” This work concentrates on whether or not heterogeneous databases contain such an entity as *utility* or conceptually similar entities, such as *power plant* and *pipeline*. We leave for future work the treatment of query constraints given by, for example, attribute values or spatial constraints.

Unlike other approaches to knowledge-based retrieval that map the local terms of a database onto a shared ontology [4,7,8], we map the user ontology onto a database schema and subsequently compare this schema with each of the schemas of the heterogeneous databases. Our approach does not force heterogeneous databases to commit to a common single ontology, it just retrieves from these databases entities that are most likely similar according to our similarity measurement to the conceptual classes requested by the user. The strategy of this work is to map ontological descriptions of query objects onto database schemas, since extracting semantics from logical representation of data is a much harder process than mapping semantic definitions onto logical structures. Thus, it combines ontologies and database schemas with the goal of leading to intelligent database systems.

The organization of this paper is as follows. Section 2 describes our main approach to querying heterogeneous databases. Section 3 describes components of the ontology specification and the similarity model to compare entity classes in this ontology. Section 4 describes the mapping process to the database schema and the similarity evaluation between heterogeneous database schemas. A study case in the spatial domain is presented in Section 5. Conclusions and future research directions are described in Section 6.

## 2 Components of the Knowledge-Based Query Process

The general *query process* is described as follows. A user query is pre-processed to extract terms identifying entity classes in a *user ontology*. Using a *semantic similarity model* (Section 3), we compare entity classes in this ontology to determine all classes that are semantically similar to the ones that we extract from the user's request. In this way, even if the databases do not contain exactly what the user is searching for, they may still be able to provide some semantically similar answers.

Once the set of classes associated with concepts requested by the user has been determined, the definitions of these classes are mapped onto a database schema. To do this mapping, a set of transformations tied to the type of database schema (e.g., relational or object-oriented databases) is defined and applied over the classes’ definitions to create a *query schema*, i.e., the schema of entity classes that models the user’s request. For this paper we have used the traditional relational database schema [9] and we provide a summary of transformations that map entity classes onto this database schema. The generated *query schema* is then compared to each heterogeneous database (See Section 4).

In summary, our main approach includes two types of similarity assessments: (1) a semantic similarity assessment that aims at capturing classes that are semantically similar to the user query and (2) a database similarity measure that compares representations of entities in database schemas. Instead of making all similarity evaluations at the database schema level or at the ontological level, we combine these two similarity assessments for the following reasons:

- by using a user ontology we allow users to express queries in their own terms according to their own ontology without having to know the underlying modeling and representation of data in heterogeneous databases.
- we extract from the specified query and a semantic similarity model entity classes in a user ontology that are semantically associated with the user's request. We compare these classes at the ontological level where we have a more complete description of their semantics and we can obtain a set of possible answers.
- we assume that commonly existing databases have no ontological descriptions of their stored entities so, we are not provided with the full semantic description of entities stored in heterogeneous databases. Therefore, we use available components of the schema representation to compare entities through a database similarity model.

## 3 Ontology and Semantic Similarity

In a previous work [10, 11], we introduced an ontology defined with retrieval purposes whose basic specification components are described as follows.

### 3.1 Components of the entity classes' definitions

Components of our ontology are entity class definitions in terms of the classes' semantic interrelations and distinguishing features. We refer to entity classes by words or sets of synonyms, which are interrelated by hyponymy or is-a relations and by meronymy relations or part-whole relations. We use the distinguishing features of classes to capture details among descriptions of classes that otherwise are missed in the classes' semantic interrelations. For example, we can say that a *hospital* and an *apartment building* have a common superclass *building*; however, this information falls short when trying to differentiate a *hospital* from an *apartment building*. We suggest a finer identification of distinguishing features and classify them into functions, parts, and attributes. Function features are intended to represent what is done to or with a class. Parts are structural elements of a class, such as the *roof* and *floor* of a *building*, that may have not be defined as entity classes. Finally, attributes can correspond to additional characteristics of a class that are not considered by either the set of parts or functions. This classification of distinguishing features into parts, functions, and attributes attempts to facilitate the implementation of the entity class representation, as well as it enables the separate manipulation of each type of distinguishing feature [11].

### 3.2 Semantic Similarity

We define a computational model that assesses similarity by combining a feature-matching process with a semantic-distance measurement [11]. The global similarity function $S_C(c_1,c_2)$ is a weighted sum of the similarity values for parts, functions, and attributes. For each type of distinguishing feature we use a similarity function $S_t(c_1,c_2)$ (Equation 1), which is based on the *ratio model* of a feature-matching process [12]. In $S_t(c_1,c_2)$, $c_1$ and $c_2$ are two entity classes, $t$ symbolizes the type of features, and $C_1$ and $C_2$ are the respective sets of features of type $t$ for $c_1$ and $c_2$. The matching process determines the cardinality (| |) of the set intersection ($C_1 \cap C_2$) and the set difference ($C_1 - C_2$).

$$S_t(c_1,c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha(c_1,c_2) \cdot |C_1 - C_2| + (1-\alpha(c_1,c_2)) \cdot |C_2 - C_1|} \qquad (1)$$

The function $\alpha$ determines the relative importance of different features between entity classes. This function $\alpha$ is defined in terms of the degree of generalization of entity classes in the hierarchical structure, which is determined by a semantic-distance measurement. This definition assumes that a prototype is generally a superclass for a variant and that the concept used as a reference (i.e., the second argument) should be more relevant in the evaluation [12,13].

Our similarity model has two advantages over semantic similarity models based on semantic distances and their variations [14,15,16]. First, it allows us to discriminate among closely related classes. For example, we could distinguish similarity between pairs of subclasses (between *hospita*l and *house* an between *hospital* and *apartment building*, which are all subclasses of *building*) and between classes that are indirectly connected in the hierarchical structural (*stadium* as a subclass of *construction* and *athletic field* as a subclass of *field*). Second, our model does not assume a symmetric evaluation of similarity and allows us to consider context dependencies associated with the relative importance of distinguishing features [10,11].

## 4 Mapping and Comparison of Database Schemas

Once we have the desired entity classes, the next step in processing the query is to map the entities classes of our ontology onto database schemas, which are then compared with schemas of heterogeneous databases. We describe our approach to mapping with databases that are modeled with the relational database schema [9]; however, we could have used another type of database schema, such as an object-oriented schema, in which case new mapping transformations should be defined.

### 4.1 From Ontology to Database Schema

We assume that the existing database schemas (target schemas) are represented in the relational model with the following constructors:

- Entities: names, attributes, primary key, and foreign keys.
- Attributes: names.
- Foreign keys (FK): relations that they belong and refer to.

Prior to transforming the entity classes' definitions into a relational schema, we apply preprocessing to these definitions in order to keep only those components that can be mapped onto a relational schema:

- *Semantic relations extraction:* semantic relations are considered while descriptions and distinguishing features are eliminated in the subsequent mapping process. As we will explain in Section 4.2, we do not compare attributes (i.e., distinguishing features) since this would give misleading results due to the strong application dependences of attribute definitions in existing databases.
- *Synonym extraction:* Since synonyms are important to managing the multiple ways that people can refer to the same entity class, and since synonyms are not directly handled in the relation schema, we define an additional structure to deal with synonym sets of entity classes. This structure includes the set of synonym sets and an index as unique key.

Then, we take the simplified entity classes' definitions and we map them onto a relational schema. There is a direct mapping of entity classes onto relational schema; however, we also need to define transformations for mapping is-a, is-part- and whole-of semantic relations. Since there are several alternatives to mapping semantic relations onto relational schemas, we define a subset that considers only relational tables or entities' interrelations mapped through foreign keys (Table1).

**Table 1.** Mapping transformations from the entity class definition onto a relational schema

| **Semantic Relation** | **Transformation** |
|---|---|
| **Is-A** | • *$Isa_1$:* Create an entity for each of the *children* entities with a foreign key pointing to the parent entity. |
| **Part-Of** | • *$Part_1$:* Define new structures (relations) that associate *whole* entities with *part* entities.<br>• *$Part_2$*: Create a foreign key in a *part* entity for each of its *whole* entities. |
| **Whole-Of** | • *$Whole_1$*: Define new structures (relations) that associate *whole* entities with *part* entities.<br>• *$Whole_2$* Create a foreign key in a *whole* entity for each its *part* entities. |

The combination of the alternative transformations (i.e., 1 alternative for is-a relations, 2 alternatives for whole-of and part-of relations) gives us 4 possible mapping transformations.

### 4.2 Comparing Database Schemas

Comparing databases schemas is difficult due to the intrinsic differences in the database design and modeling. To deal with this problem in the most general case, we consider all possible mapping transformations over the reduced number of entity classes obtained from the semantic similarity assessment.

At the bottom line, we compare character strings of entities' names, attributes domains, and foreign keys' references (Equation 2). This string matching is over all synonyms that refer to entities in the query schema, which are defined in the complementary structure *synSet*. In Equation 2, $e_j^o$ corresponds to an entity *j* in the query schema (user ontology), $e_i^p$ corresponds to an entity *i* in a database schema *p*, $t_i$ represents a term (e.g. *building*) or composite term (e.g., *building_complex*) that refers to an entity, attribute domain, or foreign key's reference.

$$S_w(e_i^p, e_j^o) = \underset{t_j \in synSet_{e_j^o}}{Max} \left[ \frac{|t_{e_i^p} \cap t_j|}{|t_{e_i^p} \cap t_j| + |t_{e_i^p} - t_j| + |t_j - t_{e_i^p}|} \right] \tag{2}$$

In order to complement the name-matching evaluation, we take the semantic relations (is-a, part-of, and whole-of relations) as the subject of comparison. The idea is to compare whether compared entities are related to the same set of entities. Thus, comparing semantic relations becomes a comparison between the semantic neighborhoods of entities, where the semantic neighborhood of an entity is the set of entities related through the is-a, part-of, and whole-of relations. The general approach to compare semantic neighborhoods is to use name matching over their components in a database schema. In the case of relational schema, entities in semantic neighborhoods are represented by *references* in the description of foreign keys or by values in the domain of an attribute type in an entity. Thus, we define a similarity function based on alternatives of the semantic relation representation assuming that the query schema will always represent entities in a semantic neighborhood by foreign keys in the corresponding relational tables (Equation 3). In Equations 3, *n* is the number of foreign keys in the *i*th entity of the database *p* ($e_i^p$), *m* is the number of attributes domain available in entity $e_i^p$, $FK_{i,e}$ corresponds to the *reference-to* specification of the *i*th foreign key in entity *e*, $D_{l,j,e}$ is the *l*th domain value in attribute *j* of entity *e*, and β is the number of domain attributes in $e_i^p$ with similarity greater than zero to any of the foreign keys in $e_j^o$. The variable β is defined during the similarity process, since we cannot consider all attributes as attribute types that represent a semantic relation.

$$S_n(e_i^p, e_j^o) = \frac{\sum\limits_{ii=1}^{n} \underset{jj}{Max}\, S_w(FK_{ii,e_i^p}, FK_{jj,e_j^o}) + \sum\limits_{ii=1}^{m} \underset{l,j}{Max}\, S_w(D_{l,ii,e_i^p}, FK_{jj,e_j^o})}{n + \beta} \tag{3}$$

This comparison is asymmetric. Specifically, the base element of the comparison (i.e., the second argument) comes from an ontology definition, and the target entity (i.e., the first element) is an entity of an existing database, which is likely to have a subset of the full semantic description of the concept. In our previous work [10] we showed that in comparing concepts from different ontologies, a good indication of whether or not these entity classes are similar across ontologies is obtained by matching entities' names and matching entities in a semantic neighborhood. Although we explored attribute matching, our previous work showed that attributes are application-dependent components of entities’ representations, and so there would be less chance that two databases would have many attributes in common.

In order to integrate the information obtained from the similarity assessments of name matching and semantic neighborhoods, we use a similarity function that is defined by the weighted sum of the similarity of each specification component.

## 5 Example

We have implemented our approach in a prototype that includes an ontology definition and the similarity models. We applied our approach in the spatial domain and we created a user ontology derived from a subset of two already available information sources: WordNet [17] and The Spatial Data Transfer Standard [18]. We created this ontology with 260 definitions to exploit a more complete definition of entity classes (i.e., semantic relations from WordNet and distinguishing features from SDTS). As an existing spatial database, we consider a relational schema derived from the specification of the Vector Smart Map (VMAP) level 0 of the National Imagery Mapping Agency (NIMA).

As an example, we consider the simple query to spatial databases to retrieve information about “*utilities*.” Then we took the entity *utility* in our user ontology and we applied a semantic similarity evaluation that results in a set of three semantically similar entity classes: *electrical system*, *heating system*, and *plumbing system*. In this example we considered as candidate answers all entities whose similarity to the entity class *utility* is larger than 0.5. We then mapped the definitions of each of these candidate entity classes onto a relational schema (Table 2). In Table 2 we show only the mapping schema that leads to the best results of similarity, using the transformations $is_1$ and $whole_1$, and transformations for *part-of* relations being unnecessary for this case.

**Table 2.** Definitions of entity class *utility* and its semantically similar entity classes

| Entity class | Relational Schema |
|---|---|
| **entity_class {**<br>**name:** {utility}<br>**description:** A Unit composed of one or more pieces of equipment connected to a structure and designed to provide service such as heat, light, water, or sewage disposal.<br>**is_a:** {facility} **part_of:** {} **whole_of:** {}<br>**parts:** {}<br>**functions:** {{transmit,conduct,carry}}<br>**attributes:** {{name},{condition}, {support_type},{location}**}** | **Utility(**$\mathrm{FK}_{\mathrm{facility}}$**)**<br>Foreign key: $\mathrm{FK}_{\mathrm{facility}}$ references to Facility |
| **entity_class {**<br>**name:** {electrical_system}<br>**description:** Equipments that provide electricity or light.<br>**is_a:** {utility}<br>**part_of:** {}<br>**whole_of:** {{power_plant}, {cable,wire,line,transmission line}}<br>**parts:** {{power_plant}, {cable,wire,line,transmission line}}<br>**functions:** {{transmit,conduct,carry}}<br>**attributes:** {{name},{condition}, {support_type}, {location}, {signal_type},{single_multiple_wires}**}** | **Electrical system (**$\mathrm{FK}_{\mathrm{utility}}$**,** $\mathrm{FK}_{\mathrm{power\_plant}}$, $\mathrm{FK}_{\mathrm{cable}}$**)**<br>Foreign key: $\mathrm{FK}_{\mathrm{utility}}$ reference to Utility<br>Foreign key: $\mathrm{FK}_{\mathrm{power\_plant}}$ reference to Power_Plant<br>Foreign key: $\mathrm{FK}_{\mathrm{cable}}$ reference to Cable<br>**Power plant(**$\mathrm{FK}_{\mathrm{electrical\ system}}$**)**<br>Foreign key: $\mathrm{FK}_{\mathrm{electrical\ system}}$ references to Electrical system<br>**Cable(**$\mathrm{FK}_{\mathrm{electrical\ system}}$, **)**<br>Foreign key: $\mathrm{FK}_{\mathrm{electrical\ system}}$ references to Electrical system |
| **entity_class {**<br>**name:** {heating_system}<br>**description:**<br>**is_a:** {utility}<br>**part_of:** {}<br>**whole_of:** {{pipeline,piping,pipage,pipe}}<br>**parts:** {{pipeline,piping,pipage,pipe}}<br>**functions:** {{transmit,conduct,carry}, {warm,heat}}<br>**attributes:** {{name},{condition}, {support_type},{location}**}** | **Heating system (**$\mathrm{FK}_{\mathrm{utility}}$**,** $\mathrm{FK}_{\mathrm{pipeline}}$**)**<br>Foreign key: $\mathrm{FK}_{\mathrm{utility}}$ references to Utility<br>Foreign key: $\mathrm{FK}_{\mathrm{pipeline}}$ reference to Pipeline<br>**Pipeline (**$\mathrm{FK}_{\mathrm{Heating\ system}}$, FK $\mathrm{P}_{\mathrm{lumbing\ system}}$**)**<br>Foreign key: $\mathrm{FK}_{\mathrm{Heating\ system}}$ references to Heating system<br>Foreign key: FK $\mathrm{P}_{\mathrm{lumbing\ system}}$ references to Plumbing System |
| **entity_class {**<br>**name:** { plumbing_system}<br>**description:**<br>**is_a:** {utility}<br>**part_of:** {}<br>**whole_of:** {{pipeline,piping,pipage,pipe}}<br>**parts:** {{pipeline,piping,pipage,pipe}}<br>**functions:** {{transmit,conduct,carry}, {dispose,throw out,throw away}}<br>**attributes:** {{name},{condition}, {support_type}, {location}**}** | **Plumbing system (**$\mathrm{FK}_{\mathrm{utility}}$**,** $\mathrm{FK}_{\mathrm{pipeline}}$**)**<br>Foreign key: $\mathrm{FK}_{\mathrm{utility}}$ references to Utility<br>Foreign key: $\mathrm{FK}_{\mathrm{pipeline}}$ reference to Pipeline |

The final similarity values between entities in the database that best match entities in the query schema are shown in Table 3.

**Table 3.** Results of similarity evaluation between the DB_VMAP and QS

| Entity in the DB_VMAP | Entity in the QS | $S_w$ | $S_n$ | Similarity Total |
|---|---|---|---|---|
| Utility Point Feature | Electrical System | 0 | 0.65 | 0.31 |
| Pipeline Line Feature | Heating System | 0 | 0.65 | 0.31 |
| Pipeline Line Feature | Plumbing System | 0 | 0.65 | 0.31 |
| Utility Line Feature | Electrical System | 0 | 0.45 | 0.23 |

As the results in Table 3 show, when we deal with heterogeneous databases, we cannot expect high values of similarity, but at least we are able to offer entities that have a strong chance of being associated with the concepts specified in the query.

## 6 Conclusions and Future Work

We have defined a new approach to querying heterogeneous databases based on similarity functions at an ontological level corresponding to a user's query and, at the logical level, between database schemas. The main characteristics of our approach are that we do not assume that databases share some level of the same conceptualization and we search for possible common components within the entities' representations.

The results of our experiment indicate that our approach detects correspondences between entities that are most likely similar; however, it may not detect all cases of similarity. In particular, further research needs to be done to recognize in the similarity evaluation when relational tables are just structures that represent semantic relations (e.g., structures created by transformation *part1* and *whole1*), as opposed to structures representing entities. In addition, we have not considered attributes in our comparison, but if we wish to process the whole query, we need to treat query constraints, which are usually described by attributes values.

## 7 Acknowledgement

This work has been partially funded by Fundación Andes, Chile. Andrea Rodríguez's research is further funded by Conicyt under grant Fondecyt 1010897. Marcela Varas's research is funded by the Dirección de Investigación of Universidad the Concepción under grant number 99.093.003-1. We also want to thanks to Max Egenhofer's contribution to the previous work upon which this paper is based, and to Jim Farrugia for his valuable comments.

## References

1. Guarino, N., Masolo, C., Verete, G.: OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems **14** (1999) 70-80.
2. Bergamaschi, B., Castano, S., de Vermercati, S., Montanari, S., Vicini, M.: An Intelligent Approach to Information Integration. In: N. Guarino (ed.): First International Conference on Formal Ontology in Information Systems. IOS Press, Trento Italy (1998) 253-268
3. Voorhees, E.: Using WordNet for Text Retrieval. In: Fellbaum C. (ed.): WordNet: An Electronic Lexical DatabaseCambridge. The MIT Press, MA (1998) 285-303
4. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.: OBSERVER: An Approach for Query Processing In Global Information Systems Based on Interoperation across Pre-existing Ontologies. Distributed and Parallel Databases **8** (2000) 223-271
5. Bertino, E., Catania, B., Zarri, G.: Intelligent Database Systems. ACM Press, London UK ( 2001)
6. Guarino, N.: Formal Ontology in Information Systems. In: Guarino, N. (ed.): Formal Ontology in Information System*s*. IOS Press, Trento, Italy (1998) 3-15
7. Bright, M., Hurson, A., Pakzad, S.: Automated Resolution of Semantic Heterogeneity in Multidatabases. ACM Transactions on Database Systems **19** (1994) 212-253
8. Fankhause, P., Neuhold, E.: Knowledge Based Integration of Heterogeneous Databases. In: Hsiao, H., Neuhold, E., Sacks-Davis, R. (eds.): Database Semantics Conference on Interoperable Database Systems IFIP WG2.6. Elsevier Science Publishers, North-Holland (1992) 155-175
9. Codd, E.: A Relational Model of Data for Large Shared Data Banks. Communications of the ACM **13** (1970) 377-387
10. Rodríguez, A., Egenhofer, M.: Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineerin*g* (in press)
11. Rodríguez, A. Egenhofer, M.: Putting Similarity Assessment into Context: Matching Functions with the User's Intended Operations. In: Sefarini, L., Brezillon, O., Castellano, F., Bouquet, P. (eds): Modeling and Using Context CONTEXT99. Lecture Notes in Computer Science *V*ol. 1688. Springer-Verlag, Berlin (1999) 310-323
12. Tversky, A.: Features of Similarity. Psychological Review **84** (1977) 327-352
13. Rosch, E.: Cognitive Representations of Semantic Categories. Journal of Experimental Psychology **104** (1975) 192-233
14. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and Application of a Metric on Semantic Nets. IEEE Transactions on System, Man, and Cybernetics **19** (1989) 17-30
15. Sussna, M.: Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network. In: Second International Conference on Information Knowledge Management, CIKM'93 (1993)
16. Resnik, O.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity and Natural Language. Journal of Artificial Intelligence Research **11** (1999) 95-130
17. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-Line Lexical Database. International Journal of Lexicography **3** (1990) 235-244.
18. USGS: View of the Spatial Data Transfer Standard (SDTS) Document (1998)