Querying Heterogeneous Spatial Databases: Combining an Ontology with Similarity Functions

Mariella Gutiérrez[1] and Andrea Rodríguez[2]

[1] School of Engineering, Universidad Católica de la Santísima Concepción, Caupolicán 490, Concepción,Chile
marielag@ucsc.cl
[2] Department of Computer Science, Universidad de Concepción, Edmundo Larenas 215, Concepción, Chile
andrea@udec.cl

Abstract. This paper uses a knowledge-based approach to querying heterogeneous spatial databases based on an ontology and conceptual and attribute similarities. The ontology, which may be independent of the databases, expands and filters a user query. Then, queries are translated into a formal specification of entity classes, which are compared against definitions in databases. This process is carried out by determining the conceptual similarity between entities in a user ontology and by comparing these entities in the ontology with entities in the conceptual models of databases. In addition, the specification of a query is done not only by identifying entity classes but also by considering constraints based on attribute values. The paper describes the system architecture and presents a case study with data from a forestry information system.

1 Introduction

This paper presents a system architecture for accessing information across heterogeneous spatial databases based on a user ontology and similarity functions. The focus of the paper is at the semantic level, where the ontological definitions of geographic features are independent of their geometric representations.

Studies that use an ontology for data integration require that databases subscribe to a common ontology, which is similar to subscribing to a shared schema at the schematic level. This common ontology is obtained by a single ontology or by the integration of multiple and independent ontologies [2, 17–19, 25, 29]. This work, in contrast, relaxes this strategy of using a common ontology, since it does not force databases either to subscribe to a common ontology or to have a complete semantic description of their information content. The approach of this work is to use semantic similarity measures to associate dynamically entities from different conceptualizations while maintening these conceptualizations independent [12].

This work follows and extends ideas from [12–14] that define similarity functions between ontologies and between ontologies and databases. Unlike these

previous works, in this paper we define a mechanism that retrieves data from heterogeneous databases based on the identification not only of entity classes, but also of instances that are similar to a user request. This work assumes that each database has a conceptual schema. The use of the logical schema was explored in [14], but this approach has strong limitations respect to the description of the information content of a database. Conceptual schemas and the user ontology are expressed in OWL, a standard language for the definition of Ontologies in the Semantic Web [3, 15].

The organization of this paper is as follows. Section 2 reviews related work about querying heterogeneous databases. Section 3 describes the system architecture followed by Section 4 that addresses the description of the user ontology and conceptual schemas of databases. Section 5 adapts similarity functions of previous works [12–14] to evaluate similarity within the user ontology and between the ontology and conceptual schemas. A case study in the area of a forestry information system illustrates the access to databases in Section 6. Conclusions and future work are presented in Section 7.

2 Related Work on Querying Heterogeneous Data Repositories

Many studies have treated the problem of accessing independent databases as a problem of solving heterogeneities among these databases. Focusing on semantic heterogeneities, studies have proposed the use of ontologies to specify queries and describe the content information of databases [2, 8, 18, 19]. In current ontology-based information systems, semantic matching has meant the agreement on the vocabulary used by different agents. This implies sharing the same conceptualization or agreeing to adopt a common conceptualization, which is usually the intersection of the original conceptualizations [10, 11]. Consequently, the general approach to handling semantic heterogeneity has been to map the local terms in a database onto a shared or common ontology. Most of these approaches use the terms interrelationships to determine semantic similarity between concepts [4–6]. Other approaches are measures based on graph matches and probabilistic measures that predict the probability that an instance of a concept in a differentiated ontology will satisfy a request [30].

In environments with multiple and independent information systems, however, each system may have its own conceptualization and, therefore, its own intended model or ontology. Nonetheless, if existing ontologies are well defined, their integration may reduce the cost of building a global ontology from scratch [2, 16]. Ontology integration is a complex problem, because concepts can overlap or definitions of concepts may be inconsistent across ontologies [27]. Some systematic approaches to handling ontology integration are composition algebras [21], lexical interrelations [2, 18, 19], mappings with mediator agents [22], inheritance from top-level ontologies [8], and semantic correspondence that relies on a common vocabulary for defining concepts across different concepts [25, 29]. All

of these approaches are manual or semi-automatic, requiring some input from domain experts.

Applications that use an ontology-based access to information require associations between concepts in an ontology with data stored in information sources. Ontologies may relate to database schemas or single terms. A simple strategy for mapping ontologies onto databases is to translate the database structure into a language in which automatic reasoning is possible [1]. Another approach uses the ontology to further refine terms in the databases or database schema [25]. A structure enrichment combines the translation of data structure with the use of an ontology for enriching the definition of terms [16]. In the World Wide Web domain, the use of metadata adds semantics to an information source or databases. For these metadata, efforts have been made concerning the use of standards for expressing content information of data repositiories, such as the Resource Description Framework (RDF) and the Web Ontology Language (OWL) [28].

3 System Architecture

Main components of the proposed system include a *user ontology*, *conceptual schemas*, and *similarity functions* (Figure 1). An ontology describes concepts of terms in a user query; conceptual schemas describe the content of databases; and similarity functions compare concepts or descriptions at two different levels: (1) comparing entities within the user ontology for query validation and expansion, and (2) comparing entities in the ontology with entities in the conceptual schemas of databases.

An ontology allows users to express queries in their own terms according to their own conceptualizations without having to know the underlying modeling and representation of data in heterogeneous databases. Concepts used by the user in a query can be then compared in order to search not only for what the user has explicitly requested, but also for semantically similar terms (i.e., query expansion). These concepts are compared at the ontological level where there is a more complete description of the semantics of terms. The user can also select attributes of entities classes to constrain answers.

Since databases may have been designed without assuming the same user ontology, our system compares definitions in the user ontology with the content description of databases. This type of comparison differs from the one within a single ontology, since different levels of explicitness and formalization may affect the way definitions may be compared. Therefore, a second similarity evaluation compares ontological definitions with available components of conceptual schemas of databases. This comparison reduces the search space in each heterogeneous database to the set of entities that are semantically similar to the terms that belong to a user query.

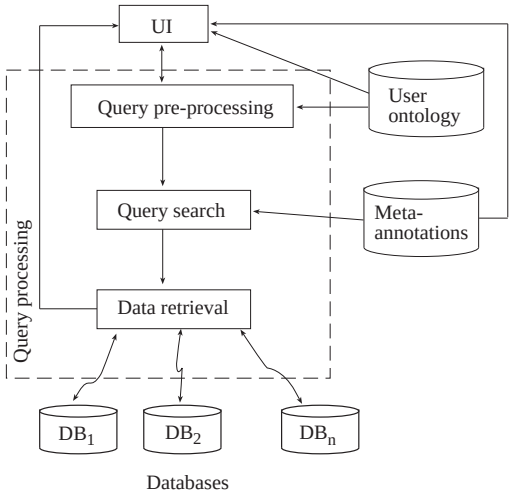

Fig. 1. System architecture

4 Ontology and Conceptual Schemas

This work uses three basic components that define entity classes in an ontology of a spatial domain [12, 13]: (1) a set of synonym words (synset) that denotes an entity class, (2) a set of semantic interrelations among these entity classes, and (3) a set of distinguishing features that characterize entity classes. This ontology, similar to a terminological ontology, supports information retrieval rather than query answering, which is typically done with an axiomatized ontology [24]. In this ontology, the use of a set of words to denote entity classes addresses polysemy and synonymy in the process of linking words to meaning. Polysemy occurs when the same word denotes more than one meaning, and synonymy occurs when different words denote the same or very similar entity classes [20]. Synonym sets attempt to capture more semantics than a single word that denotes an entity class.

Two semantic relations play an important role in the specification of ontologies: hyponymy, also called the is-a relation, and meronymy, which is a partial ordering of concept types by the part-whole relation [9, 23]. Properties that distinguish entity classes from the same superclass are called distinguishing features [24]. Usually, attributes describe different types of distinguishing features of a class. They provide the opportunity to capture details about classes, and their values describe the properties of individual objects (i.e., instances of a class). Unlike our previous work [12], this work does not distinguish between types of features, since such a distinction could only work at the ontological level, but not when comparing the ontological description of a user request with the description of entities in a traditional databases.

In order to be able to specify constraints in terms of attribute values, we complement the ontological definitions of entities' attributes by the description of the attributes' values, that is, values' domain and, if necessary, values' units. The definition of the ontology as a RDF graph model is presented in Figure 2, which was then expressed in the OWL language [3, 7, 15].

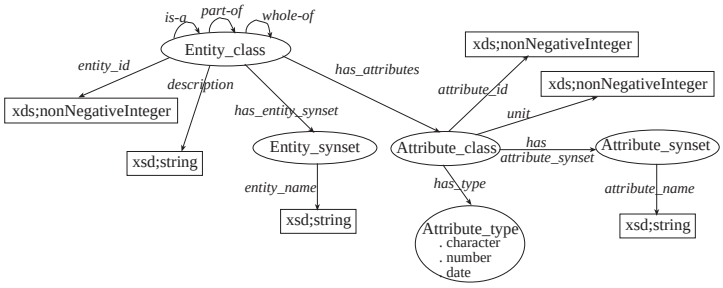

Fig. 2. The RDF graph model of the ontology

Conceptual schemas are rich enough to establish attributes and semantic interrelations between entity classes. Unlike the ontological definitions of entity classes, databases are created for a particular application in a domain. In this context, we consider schemas as simplified views of ontological descriptions, which are expressed in the same way than the user ontology in the RDF graph model.

5 Similarity Functions

5.1 Query Expansion

In this work, the expansion of a query translates the query into a set of entity classes that are semantically related to terms in a user request. This expansion is accomplished by applying matching terms of a query with entity classes in an ontology and, then, by determining similarity between entity classes in the ontology.

Similarity is calculated as a function of common and different features $S_{exp}(a, b)$ (Equation 1) [12], which is based on the ratio model of a feature-matching process [26]. In $S(a, b)$, a and b are two entity classes, A and B correspond to the description sets of a and b (i.e., synonym set of features of entity classes)), and c is the the first class that subsumes a and b by the is-a or whole-of relation. The

matching process determines the cardinality (||) of the set intersection ($A \bigcap B$) and the set difference (A/B).

$$S_{exp}(a,b) = \frac{|A \bigcap B|}{|A \bigcap B| + \alpha(a,b)|A/B| + (1-\alpha(a,b))|B/A|}$$

where

$$\alpha(a,b) = \begin{cases} \frac{distance(a,c)}{distance(a,b)} & \text{if } distance(a,c) \leq distance(b,c) \\ 1 - \frac{distance(a,c)}{distance(a,b)} & \text{if } distance(a,c) > distance(b,c) \end{cases} \quad (1)$$

5.2 Mapping Ontology onto Databases

Mapping ontological definitions onto descriptions of a database can be achieved if the ontology representation and the conceptual schema of the database share some components. A natural way to exploit the full expressiveness of concept representations for a similarity evaluation is to compare each component in those representations. Thus, two different descriptions (i.e., ontologies, conceptual schemas) that have at least one common specification component can still be compared.

In this study we follow some of the results from the work by Rodríguez and Egenhofer [12] where they compare definitions in different ontologies. In our case, we make comparisons between a user ontology and databases' schemas, whereas in their work Rodríguez and Egenhofer compare entity classes across ontologies and use three independent similarity assessments: name-based similarity, neigborhood-based similarity, and attribute-based similarity.

Our work takes two steps to evaluation similarity between a query and stored data. The first step compares entity classes in the ontology with entities in databases based on name and neighborhood similarities. The similarity of names between an entity class in the ontology and an entity in a database's conceptual schema aims at exploiting the general agreement in the use of words and detects equivalent words that likely refer to the same entity class. This evaluation takes the maximum similarity between names, which are composed of one or more words. This similarity is evaluated as a simple matching process $Name(a,b)$ (Equation 2), where a_n is a name in the entity's synonym set in the user ontology and b_n is a name of an entity in a database's schema. Consider that a name may be composed of one or more words.

$$Name(a,b) = \frac{|a_n \bigcap b_n|}{|a_n \bigcap b_n| + |a_n/b_n| + |b_n/a_n|} \quad (2)$$

The similarity of neighborhoods involves semantic relations themselves as the subject of comparison. Since the types of semantic relations are known (e.g., is-a or part-whole relations), the interesting aspect of comparing semantic relations is whether target entities (i.e., entity classes that are the subject of comparison) are related to the same set of entity classes. If so, the entities may be semantically similar. Comparing semantic relations becomes a comparison between the

semantic neighborhoods of entities $SN(a, b)$. The semantic neighborhood (N) of an entity a consists of those entities that are at a minimum distance from a, i.e., those entities that have a direct relationship with a (Equation 3).

$$SN(a,b) = \frac{|N(a) \bigcap_n N(b)|}{|N(a) \bigcap_n N(b)| + \delta(N(a), N(b)) + \delta(N(b), N(a))} \tag{3}$$

where

$$\delta(N(a), N(b)) = \begin{cases} |N(a)| - |N(a) \bigcap_n N(b)| \text{ if } |N(a)| > |N(a) \bigcap N(b)| \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

The similarity evaluation of entities in semantic neighborhoods compares all entities in one neighborhood with entities in a second neighborhood in such a way that yields the maximum similarity between neighborhoods. The comparison between entities in semantic neighborhoods is based on name matching (Equation 4). In a extreme case, it is possible that the comparison between neighborhoods gives a larger value than the number of elements in one of the neighborhoods. This can happen when more than one entity in a neighborhood is similar to a single entity in another neighborhood. In such a case, $\delta()$ is considered to be equal to zero.

$$|N(a) \bigcap_n N(b)| = \left[\sum_{c \in N(a)} max_{d \in N(b)} Name(c,d) \right] \tag{4}$$

Combining name and neighborhood similarities, a global entity similarity is given by Equation 5, where ω_w and ω_n are the relative importance of name similarity and neighborhood similarity, respectively.

$$S_{entity}(a,b) = \omega_w \cdot Name(a,b) + \omega_n SN(a,b) \tag{5}$$

Unlike [12], this work uses the similarity evaluation between attributes as a subsequent evaluation that is only applied when entities are similar. In this sense, attribute similarity can be seen as a second filter and final discriminator between entities in the database, a discriminator that is only useful when the system has detected that the entities in the databases are similar to the entities in the user ontology. We do so, because feature similarity was found to be more useful for comparing definitions within a single ontology or for comparing semantically similar entities [12].

In our system two query cases are when the user has not specified attribute values as constraints or when the user has filtered the entities to be retrieved by attribute values. In the first case, comparing attributes at a semantic level could discriminate between similar entities by considering an attribute matching at the conceptual level, that is, matching of the class of attribute rather than between attribute values (Equation 6). In Equation 6, a_t is the set of synonym sets that refer to attributes in the entity class a of the user ontology and b_t is the

set of terms that refers to attributes in the entity b of a database, respectively. Attribute correspondence is determined by considering a strict matching between terms, that is, a common attribute means that the term that refers to this attribute in the database was found as one of the terms in a synonym set of the attributes in the ontology.

$$S_{query}(a,b) = \begin{cases} \frac{|a_t \bigcap b_t|}{|a_t \bigcap b_t| + |b_t / a_t|} & \text{if } S_{entity}(a,b) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We assume that the ontology is semantically richer than the conceptual schemas of the databases, so that entities in the ontology may have a more complete set of attributes. In this case, the set of attributes in the database is likely a subset of the set of attributes of the ontology, and attributes that are present in the database and not in the ontology reflect potential differences in the semantics of entities. Therefore, only attributes that are included in the databases' schemas and not in the ontology will affect the similarity assessment.

In the second query case, in order to be able to answer the query, attributes in the query specification should be part of the entities in databases. So, these attributes in the query specification q_t are necessary conditions of entities (Equation 7).

$$S_{query}(a,b) = \begin{cases} \frac{|a_t \bigcap b_t|}{|a_t \bigcap b_t| + |b_t / a_t|} & \text{if } (S_{entity}(a,b) \geq \tau) \wedge (q_t \subseteq b_t) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

6 Case Study: Forestry Management System

As a case study, we consider a forestry management domain. For such domain, an ad-hoc ontology in Spanish was created with definitions derived from available dictionary, thesaurus and glossaries used in agencies of resource management in Chile. A portion of the ontology translated to English is shown in Figure 3.

The case study uses a real forestry database. This database contains 100 entities, 27 of them related to forest management. As examples of how the system works, consider the following queries. The first case is a query without constraints based on attribute values (Table 1).

The original query is expanded to include seven different entity classes based on a threshold of $S_{exp} \geq 0.5$. For each of these entity classes, the system finds the most similar entity in the database based on the similarity S_{entity}. Only entities in the database with a similarity $S_{entity} \geq 0.5$ are considered in the final evaluation of query similarity S_{query}. Since this query does not consider attribute values as search criteria, S_{query} between an entity class in the ontology and an entity in the database is given by their common attributes (Equation 6).

A second case is a query with the specification of an attribute value that exists in a database (Table 2).

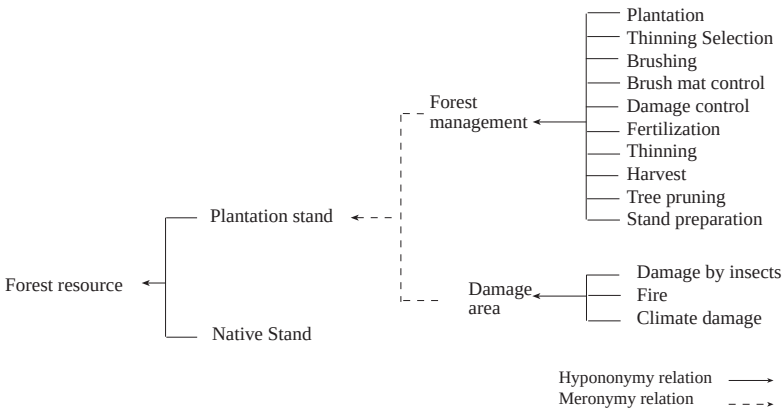

Fig. 3. A portion of an ontology for forest management

Table 1. A query without an attribute value specification

Query	Expansion	S_{exp}	Database Entity	S_{entity}	S_{query}
Select *stand*	stand	1	plantation stand	0.58	0.63
	forest resource	1	nonexistent	-	-
	damage	0.67	stand damage	0.62	0.28
	climate damage	0.62	nonexistent	-	-
	fire	0.62	nonexistent	-	-
	damage by insects	0.59	nonexistent	-	-
	native stand	0.56	native stand	0.5	0.5

Table 2. A query with an attribute specification that exists in the database

Query	Expansion	S_{exp}	Database Entity	S_{entity}	S_{query}
Select *stand* where *plantation year* $>$ 2000	stand	1	plantation stand	0.58	0.63
	forest resource	1	nonexistent	-	-
	damage	0.67	stand damage	0.62	0
	climate damage	0.62	nonexistent	-	-
	fire	0.62	nonexistent	-	-
	damage by insects	0.59	nonexistent	-	-
	native stand	0.56	native stand	0.5	0

Unlike the first query, the second query uses a search criterion that is defined by an attribute value (i.e., *plantation year* $>$ 2000). In this case, the system will check if this attribute exists in each of the entities that were found to be most similar to the entity classes in the expanded query. Only in the case that this

attribute exists in the entity of the database, S_{query} is calculated by common attributes between the entity class in the ontology and the entity in the database; otherwise, S_{query} is equal to zero (Equation 7).

The last case is a query with the specification of an attribute value that do not exist in the database (Table 3).

Table 3. A query with an attribute specification that does not exist in the database

Query	Expansion	S_{exp}	Database Entity	S_{entity}	S_{query}
Select *stand* where *resource type* = "artificial"	stand	1	plantation stand	0.58	0
	forest resource	1	nonexistent	-	-
	damage	0.67	stand damage	0.62	0
	climate damage	0.62	nonexistent	-	-
	fire	0.62	nonexistent	-	-
	damage by insects	0.59	nonexistent	-	-
	native Stand	0.56	native stand	0.5	0

The last query is an example where, although there are entities in the database that are similar to entity classes in the expanded query, none of these entities includes the attribute that is used as the search criterion. Consequently, S_{query} is always equal to zero.

7 Conclusions and Future Work

This paper describes a systems that queries heterogeneous spatial databases by using a user ontology and similarity functions that compare entities and instances. The advantages of this system are that databases can be independent of user ontologies and updates in both the user ontology and the databases will not affect the system. By using similarity functions, the system can dynamically associate user requests with entities stored in databases. Requirements of the system are a user ontology and conceptual schemas of databases described in OWL with two basic components: semantic relations (i.e., generalization and aggregation) and distinguishing features or attributes.

As future work, we plan to incorporate constraints or query conditions that combine types of entity classes (e.g., joins). Such types of queries impact the way the final similarity (S_{query}) between a query and a stored data is determined. We also expect to be able to express queries that use spatial criteria such as geographic windows and spatial relations. From the implementation point of view, we expect to have a fully running system where ontologies and conceptual schemas can be modified, and with a user friendly visualization of results.

Acknowledgment This work has been partially funded by CONICYT, Chile, under grant Fondecyt 1030301. Mariella Gutierrez' research is also funded by Universidad Católica de la Santísima Concepción under grant DIN 03/2004.

We would like to thank FORESTAL BIOBIO S.A. for providing the real database for our case study. We also thank Dr. Max Egenhofer for his contribution to previous works published in [12, 13].

References

1. Y. Arens, C.-N. Hsu, and C. Knoblock. *Readings in Agents*, chapter Query Preprocessing in the SIMS Information Mediator, pages 82–90. Morgan Kaufmann, San Francisco, CA, 1997.
2. B. Bergamaschi, S. Castano, S. De Capitani di Vermercati, S. Montanari, and M. Vicini. An intelligent approach to information integration. In N. Guarino, editor, *First International Conference on Formal Ontology in Information Systems*, pages 253–268, Pisa, Italy, 1998. IOS Press.
3. J. Berner-Lee, J. Handler, and O. Lassila. The semantic web. *Scientific American*, 184(5):34–43, 2001.
4. Y. Bishr. Overcoming the semantic and other barriers to gis interoperability. *Int. J. Geographical Information Science*, 12(4):299–314, 1998.
5. M. Bright, A. Hurson, and S. Pakzad. Automated resolution of semantic heterogeneity in multidatabases. *ACM Transactions on Database Systems*, 19(2):212–253, 1994.
6. P. Fankhauser and E. Neuhold. Knowledge based integration of heterogeneous databases. In H. Hsiao, E. Neuhold, and R. Sacks-Davis, editors, *Database Semantics Conference on Interoperable Database Systems*, pages 155–175, Victoria, Australia, 1992. Elsevier Science Publishers.
7. D. Fensel and M. Musen. The semantic web: A new brain for humanity. *IEEE Intelligent Systems*, 16(1):24–25, 2001.
8. F. Fonseca, M. Egenhofer, P. Agouris, and C. Camara. Using ontologies for integrated information systems. *Transactions in GIS*, 6(3):231–257, 2002.
9. N. Guarino. Fornal ontology, conceptual analysis, and knowledge representation. *Int. Jounal on Human Computers Studies*, 43:625–640, 1995.
10. N. Guarino. *Information Extraction: A Multidisciplinary Approach to an Engineering Information Technology*, chapter Semantic Matching: Formal Ontological Distinctions for Information Organization, Extractionm and Integration, pages 139–170. Springer-Verlag, Francasi, Italy, 1997.
11. N. Guarino. *Formal Ontology in Information Systems*, chapter Formal Ontology in Information Systems, pages 3–15. IOS Press, Trento, Italy, 1998.
12. A. Rodríguez and M. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, 2003.
13. A. Rodríguez and M. Egenhofer. Comparing geospatial entity classes: An asymmetric and context dependent similarity measure. *Int. Journal of Geographical Information Science*, 18(3):229–256, 2004.
14. A. Rodríguez and M. Varas. A knowledge-based approach to querying heterogeneous databases. In Z. Rás M.-S. Hacid, D. Zighed, and Y. Kodratoff, editors, *Foundations of Intelligent Systems. LNAI 2366*, pages 213–222. Springer-Verlag, 2002.

15. I. Horrocks and P. Patel-Schneider. Three thesis of representation in the semantic web. In *Proceeding of the 12th International Conference on WWW*, pages 39–47, 2003.
16. V. Kashyap and A. Sheth. *Cooperative Information Systems: Trends and Directions*, chapter Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context, and Ontologies, pages 139–178. Academic Press, London, UK, 1998.
17. M. Kavouras and M. Kokla. A method for formalization and integration of geographic categorizations. *Internationa Journal of Geographical Information Science*, 16(5):439–453, 2002.
18. E. Mena and A. Illarramendi. *Ontology-Based Query Processing for Global Information Systems*. Kluwer Academic Publishers, Norwell, MA, 2001.
19. E. Mena, A. Illarramendi, V. Kashyap, and A. Sheth. Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.
20. G. Miller. *WordNet: An Electronic Lexical Database*, chapter Nouns in WordNet, pages 23–46. The MIT Press, 1998.
21. P. Mitra and G. Wiederhold. *Handbook on Ontologies in Information Systems*, chapter An Ontology-Composition Algebra, pages 97–119. Springer, Berlin, 2003.
22. A. Preece, K.-J. hui, W. Gray, P. Marti, T. Bench-Capon, D. Jones, and Z. Cui. The kraft architecture for knowledge fusion and transformation. *Knowledge Based Systems*, 13(2-3):113–120, 2000.
23. J. Smith and D. Smith. Database abstractions: Aggregations and generalizations. *ACM Transactions on Database Systems*, 2:105–133, 1977.
24. J. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brook/Cole, Pacific Grove, CA, 2000.
25. H. Stuckenschmidt and H. Wache. Context modelling and transformations for semantic translation. In M. Bouzeghoub, M. Klusch, and U. Sattler, editors, *Knowledge Representation Meets Databases*, pages 115–126, Berlin, Germany, 2000.
26. A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
27. Visser, D. Jones, T. Bench-Capon, and M. Shave. *Formal Ontology in Information Systems*, chapter Assessing Heterogeneity by Classifying Onotlogy Mitmaches, pages 148–162. IOS Press, Trento, Pisa, 1998.
28. W3C. Semantic web, 2001.
29. H. Wache. Towards rule-based context transformation in mediators. In S. Conrad, H. Hasselbring, and G. Saake, editors, *International Workshop on Engineering Federated International Systems*, pages 107–122, Khlungsborn, Germany, 1999. Infix-Verlag.
30. P. Weinstein and P. Birmingham. Comparing concepts in differentiated ontologies. In *12th Workshop on Knowledge Adquisition, Modeling, and Management*, Banff, Canada, 1999.