Issues on Interoperability and Integration of Heterogeneous Geographical Data*

RENATO FILETO

[1] Institute of Computing, University of Campinas - Caixa Postal 6176, 13081-970, Campinas, SP, Brazil
fileto@ic.unicamp.br

Abstract. The interoperability of information systems has been pursued for a long time by researchers and practitioners. It involves the exchange of information among different systems, and requires agreement on formats and application domain concepts. Interoperability may also encompass commonality of user interaction and system behavior. One of the interoperability problems which has been investigated for over 20 years on different contexts is that of integration of heterogeneous data. However, even this fundamental and old problem is very hard to solve. The data integration problem may be considered from many perspectives and on increasing levels of complexity or abstraction. This paper reviews the literature about interoperability in general, and integration of heterogeneous geographical data in particular, presents several facets of the data integration problem, and some approaches to deal with them.

1 Introduction

The traditional paradigm for the development of databases and applications using their data is based on the cycle modelling-design-implementation, and considers a single database framework under one data model and with one schema. The advent of heterogeneous systems and, more recently, the Web, is changing this picture. Large amounts of data are available in distinct formats and systems, varying from structured DBMS repositories to unstructured files and home pages. Some data are structured according to well established data modeling techniques, such as the relational or object-oriented data models. Other data, such as data maintained in various information systems, spreadsheets, or Internet repositories, are in proprietary formats, semi-structured or unstructured. This situation of multiple models and schemas, combined with the intrinsic difficulties for communication and establishment of agreement for data representation in the application domains, makes the interoperability problem very complex.

The demand for interoperability has boosted the development of standards and tools to facilitate data transformation and integration. Nevertheless, there are still many challenges to be met, especially those concerned with data semantics and behavior of interoperating systems.

This work surveys some results from the literature related to interoperability and, more specifically, integration of heterogeneous geographical data. Our goal is the construction of data warehouses integrating several kinds of data sources, and in special geographical data. We believe that data warehouses constitute a suitable starting point for research and experiments on data integration. The maintenance of consolidated data at the warehouse, as independently handled copies of data available in data sources, confers greater versatility to the representation and manipulation of consolidated data. Furthermore, the one way flow of data from the sources to the warehouse, as well as the warehouse update policy that does not require on-line access to data sources, simplifies data processing a lot. The problem can be decomposed into that of extracting data from the sources to feed the warehouse, and integrating these multiple source data into the warehouse. The emphasis of this work is on the second step. This decomposition allow us to focus on representational and semantic issues, and the fundamental data integration problems. Afterwards, results from the research on data integration in warehouses may be useful to achieve interoperability in a wider sense.

Distinct data sources may be developed independently. In fact, autonomous management of databases is frequently a prerequisite for information systems, either because they are related to different applications or because they pertain to different organizations. Autonomy may be required even when applications are slightly different, there is considerable overlapping among the databases or the roles of the organizations are very similar. However, valuable information may be extracted when collections of data obtained from different data sources are analyzed as whole. The integrated analysis of data from different sources potentially triggers a wide variety of data heterogeneity problems, from simple data representation discrepancies to mismatches involving different data models, semi-structured data and even differences in data interpretation. Furthermore, connection of autonomous heterogeneous databases complicates classical database problems such as consistency maintenance, concurrency control, transactions and distributed query processing, and optimization. This work is not

*This work has been done with partial support of the Brazilian Agricultural Research Corporation (Embrapa) and the Brazilian Finep/Pronex/IC/SAI95/97 project.

concerned with any of these problems. Only consistency maintenance is considered in some degree. The core of our research is semantic data heterogeneity, specially when geographical data are involved.

The remainder of this paper is organized in the following way. Section 2 presents basic concepts related to interoperability. Section 3 focuses on data heterogeneity conflicts and data integration, establishing a framework to analyze related problems and proposed solutions. Section 4 presents the most typical apparatus for data integration. Section 5 discusses some results from the literature and future trends, under the framework proposed in section 3. Finally, section 6 gives the conclusions so far obtained from our study.

2 Interoperability of Information Systems

The term interoperability, in a general sense, refers to the ability of two systems to exchange information [16, 3]. It requires some degree of compatibility between the interoperating systems, in order to minimize transformations required for data exchange and to enable correct interpretation of transferred data. Ideally, the interoperating systems should be rigidly compliant with computational and application domain standards. However, it may be impossible in practice, due to the rate of technological changes, the lack of universally accepted standards, the existence of legacy systems, or just for reasons of autonomy of each information system. Sometimes, the required compatibility can only be achieved using abstractions to hide complexity and implementation details. The publication of formats and interfaces, as prescribed by the methodology for building open systems, can also contribute for reaching interoperability.

2.1 Viewpoints of systems interoperation

Hasselbring [18] shows that interoperability of information systems must be considered from three distinct viewpoints: application domain, conceptual design and software systems technology. These interoperability viewpoints can be outlined as illustrated in figure 1. They first arise during the development of each information system and are important to understand the system behavior and the data it manages.

Applications' viewpoint

The systems analyst is responsible for eliciting the requirements of the system to be developed. From this viewpoint, application domain experts have their own approaches, concepts, terminology, and even data representation choices to model their problems. Sometimes, even in restricted application domains, the same problem may have many alternative approaches for its modeling and solution. During the specification of an information system, a team of users, domain experts and software developers must establish an agreement on the characteristics chosen for the intended system. However, because of this diversity, the terms used to express concepts and the data representation can be very different among developed applications.

Conceptual viewpoint

The system designer formalizes the collected requirements by means of a conceptual design. The objective is to better understand the application's scope and characteristics, in order to find out the most suitable ways to model things and facts in a database. There are countless models the designer can choose (e.g. the entity-relationship model and its extensions, and the object-oriented model). Each model has its basic constructs and paradigms. The outcome of the design phase is a "blueprint" for the information system, expressing, among other things, data structuring and representation, from an abstraction level, independent from the implementation platform.

Implementation viewpoint

The computer programmer builds the application software according to the conceptual design and the technological resources available. The rapid technological evolution and the wide variety of choices to implement each system, added to differences derived from the previous viewpoints, can lead to a high degree of heterogeneity among information systems, even if they have the same application domain and user universe in mind.

Consequently, data stored in repositories maintained by different systems may present many discrepancies in their representation, engendering difficulties when there is a need for integrated information analysis. Interoperability may be desirable in all the three major viewpoints, i.e., it must be as natural and transparent as possible to the users, the conceptual design must be compatible, and interoperation between application programs must be automatic. A key point to achieve data interoperability is compatibility in the conceptual viewpoint. However, agreement on the conceptual design sometimes requires the resolution of conflicts from the applications' viewpoint. The use of technologies that facilitate interoperability, such as CORBA and XML is another factor to achieve interoperability.

Note that each viewpoint has the instance level (solutions, projects, application programs), the meta-level (approaches, models, technologies), where the general characteristics of instances are defined, and, maybe, the meta-meta level, where the models are defined, and so on. Hence, heterogeneity can also be considered from each of these successive levels of abstraction.

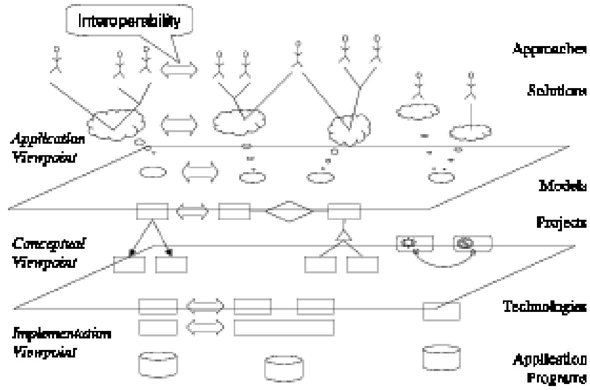

Figure 1: The viewpoints of systems interoperation

3 Integration of Heterogeneous Data

Heterogeneous data are those data presenting differences in their representation or interpretation, although referring to the same reality [23]. Data heterogeneity conflicts are the incompatibilities that may occur among distinct datasets. Heterogeneous data may be originated from different information systems, which can easily result in conflicts as outlined above.

Today, data is made available in several structured (e.g. databases) and semi-structured formats (HTML, XML, etc.), as well as in tables, spreadsheets and statistical tools. The integrated analysis of data collected from these kinds of heterogeneous sources may be important in many situations, notably for decision making purposes, when analysis of some situation can take advantage from the processing of these data as a whole.

The interoperability problem considered herein is the integration of heterogeneous data in a single database view, with unified syntax and semantics. It involves the resolution of heterogeneity conflicts and transformations of source data to accommodate them in the integrated database view. Conflicts make the integration process laborious and complex, especially if there are large volumes of highly incompatible data.

3.1 Characterizing Data Heterogeneity

Data integration is a very general problem, but it can be unfolded in some interrelated sub-problems, making it more tractable. This section defines the major categories of data heterogeneity conflicts, establishing a basis for discussions throughout this work. The taxonomy presented here, generalizes and reorganizes the classes of data heterogeneity conflicts described in [21] in the context of relational databases. We take into account additional discrepancies that can arise between heterogeneous schemas and different data models and categorize conflicts using two orthogonal classifications. On the one hand, conflicts are classified according to the abstraction level where they are detected and handled (data instance, schema or data model). On the other hand, conflicts may refer to representation (syntactics) or the meaning of the data (semantics). Though simple, this taxonomy exhibits a variety of facets of the data integration problem and clarifies the interrelationships among them.

3.1.1 Abstraction Level Concerns

Data Value Conflicts

Data value conflicts are those conflicts that arise at the instance level. They are related to the representation or the interpretation of the data values. Examples of these conflicts are discrepancies of type, unit, precision, allowed values (i.e. enumerated values for user defined atomic types), and spelling (abbreviations, typing mistakes, etc.).

To solve data value conflicts it is necessary first to establish correspondences between heterogeneous database elements. Some data value conflicts may be handled by defining appropriate metadata associated with the database schema (e.g. to describe units, allowed values, quality attributes, or the meaning of the data), and using this information to support data conversion and integration. There are also conflicts whose solution requires more sophisticated techniques. Examples of these hard problems are defining the correspondences between incompatible sets of allowed values for the same measure, the reconciliation of some nomenclature discordances, and the resolution of typing conflicts. Many of these problems could be avoided by using data entry standards.

Schema Conflicts

Schema conflicts are due to different alternatives provided by one data model to develop schemas for the same reality. For example, what is modeled as an attribute in one relational schema may be modeled as a relation in another relational schema for the same application domain. Another example is the use of different names to designate elements referring to the same real world object in different schemas. There may also be data versus schema conflicts, i.e., what is data in one schema can be metadata in another one. For example, a data value in one relational schema can be the label of an attribute or a relation in another relational schema.

Modeling choices have impact on the expressiveness, versatility and the quality of schemas. However, schema conflicts may arise even if the best modeling guidelines are employed in the development of the autonomous databases. The resolution of schema conflicts require more powerful techniques than those available in typical data definition and manipulation languages such as SQL. The identification of correspondences may require additional knowledge (semantics) about the database schema, the data model employed and the application domain.

Data Model Conflicts

Data model conflicts occur when databases use different data models, e.g., one database designed according to the relational model, and another one object-oriented. This is reflected on database schemas, with distinct basic constructs, paradigm and expressive power. The resolution of these conflicts consists in converting each of the schemas to a common data model. The target data model must allow the representation of everything (semantics) that can be expressed by any of the source data models, otherwise information can be lost in the conversion process. On the other hand, the richer the common data model, the more complex the integration process is liable to be.

The conversion of source schemas to the common data model may be done by shifting focus to the meta-meta-level of abstraction, i.e., to the definition of the data models themselves. In a high level of abstraction sometimes it is possible to define a small set of constructs sufficient to express all basic constructs of the heterogeneous data models, and a small set of operations to make interoperability possible. Hence, the use of semi-structured data models with high level of abstraction for data integration is considered in several works [26, 8, 7]. However, if source schemas carry some knowledge that cannot be expressed at the meta-meta abstraction level, some information may be lost in the integration process.

3.1.2 Representation and Interpretation Concerns

Syntactic Conflicts

Syntactic conflicts refer to discrepancies in the representation of data. They can arise in the representation of instantiated data (data values), at the schema or at the data model level. Syntactic conflicts must be solved from the higher to the lower abstraction levels, taking advantage of abstraction to obtain systematic solutions. Conflicts may also involve databases with no explicit or well defined schema, such as semi-structured and unstructured data. We consider integration of data only if some schema can be devised to support resolution of conflicts at proper abstraction levels.

Of course, syntactic conflicts can ever have a semantic counterpart. Nevertheless, we prefer to separate syntactic from semantic concerns because we believe that doing so contributes to better understand and modularize problems and solutions related with data integration. The different approaches found in the literature for solving syntactic and semantic conflicts corroborates this assertion.

Semantic Conflicts

Semantic conflicts refer to disagreement about the meaning, interpretation or intended use of the same or related data [32]. Like syntactic conflicts, semantic conflicts may occur in any level of abstraction: instance, schema, data model. The resolution of syntactic conflicts, at any abstraction level, usually require the analysis of their semantic counterpart.

The difference between syntactic and semantic conflicts may seem subtle, because they can arise between the same portions of heterogeneous datasets. In order to understand the this difference, it is necessary to remember that data, including structure, are just (inaccurate) representations of reality. Semantics tries to attach a meaning to data. Thus, semantic conflicts refer to discrepancies in the association of representations with real world objects and phenomena.

Therefore, the resolution of semantic conflicts consists in explaining the matching (perfect or not) between items in heterogeneous datasets, with respect to the correspondence established between these items and reality. Detecting and solving semantic heterogeneity are difficult problems. Typically, source data and schemas do not provide enough semantics to interpret data consistently [32]. Heterogeneity due to differences in data models also makes it difficulty to identify and solve semantic conflicts. Dealing with semantic heterogeneity often requires knowledge about the application domain and some agreement on concepts and terminology (e.g. via ontologies or dictionaries, as explained in section 5.2).

3.2 The General Data Integration Process

Parent and Spaccapietra [27] present a general data integration process in their survey on database integration. Figure 2, adapted from [27], illustrates the information flow and the main steps of this process.

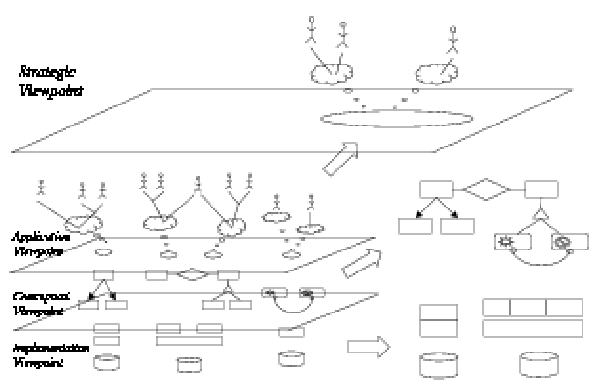

Figure 2: The data integration process

Heterogeneous database schemas are first converted to an homogenized representation, using transformation rules that explain how to transform constructs from the source data models to the corresponding ones in the target common data model. The transformation specification produced by this step specifies how to transform instance data from the source schema to the corresponding target schema. As the schemas of the data sources are semantically poor, it is always necessary to enrich the translated schemas with semantic information to support the determination of correspondence among heterogeneous data elements. Then, correspondences among objects of the heterogeneous databases are investigated, using the semantic descriptions of the database contents, standards and similarity rules, to produce a collection of correspondence assertions. Finally, correspondence assertions and integration rules are used to integrate the enriched schemas into an integrated schema, where corresponding elements of heterogeneous schemas are unified. Details of the tasks performed on each step, an approach to describe some semantic conflicts, and some solutions to these conflicts can be found in [27].

It is crucial to automate or at least assist some laborious tasks of the data integration process, in order to make data integration practicable. On the other hand, as there may be complex syntax and semantic concerns, data integration without any human intervention is often unfeasible. Thus, it is necessary at least to assist the data integration activities, as well as making complex processes of data integration repeatable.

4 General Apparatus for Data Integration

4.1 Gateways

A Gateway is some middleware that allows an application running in one DBMS to access data maintained by another DBMS. It is necessary to develop one gateway for each DBMS pair and configure it for each database to be connected. Usually, gateways are available only for DBMS that employ the same data modeling paradigm and do not provide location or interface transparency. Hence, gateways are not versatile and do not offer support to establish an homogeneous view of heterogeneous data.

4.2 Wrappers and Mediators

Wrappers and mediators [35, 14] provide data manipulation services over a unified view of heterogeneous data. Wrappers encapsulate details of each data source, allowing data access using a common data model. Mediators offer an integrated view of the data supplied by a collection of sources and through wrappers. A mediator transforms requests posed according to the integrated view into requests to the data sources, integrating the results. Some systems adopt many levels of mediators in order to modularize the data transformation process along successive levels of abstraction.

Wrapper generators and data mapping specification languages [14] enable the specification of data transformation and integration in a more intelligible manner than using conventional programming languages to hard code them in wrappers and mediators. They also provide lower development costs.

4.3 Data Warehouses

A data warehouse [28, 20, 6] is a separated database built specifically for decision support. It is useful when decisions depend on heavy analysis of large amounts of data, collected from a variety of possibly heterogeneous data sources. A data warehouse replicates and integrates data from sources such as databases maintained by on-line transaction processing systems, spreadsheets, and textual data. These data sources typically run at the operational level of organizations, while data warehouses are aimed to the strategic level.

The processes of collecting, transforming and integrating data for consolidated analysis is called data warehousing. This process may be realized off-line with periodical updates, perhaps overnight. The separation between the data warehouse and the data sources prevents interference between the data processing for decision support and the functioning of the systems at the operational level. This separation also confers greater flexibility for data organization and data processing in the warehouse. Possibly, the data from the sources is first processed before being stored at the warehouse. Data may be filtered, relations may be joined or aggregated, etc. This processing is performed for data integration and according to the requirements of data analysis.

There are specific methods for modeling and organizing data in a warehouse – e.g. multidimensional, star, and snowflake style schemas [19] – and also for data processing and user interaction – e.g. multidimensional data analysis or on-line analytical processing (OLAP) [15, 6, 17]. Powerful data analysis tools, i.e. computer programs implementing techniques to process existing data in order to (help to) derive new information from these data, can be attached on the top of a data warehouse. These tools can either read data as represented in the warehouse or the warehouse can export data to the format required for each specific tool.

5 Future Directions

5.1 XML as a Syntactic Standard

XML [38] has recently emerged as a new standard format for data representation and exchange in the Internet [1]. It is posing many challenges for database academia and industry and is a very promising direction [5, 34]. There are many technologies being developed in conjunction with the basic standard. Leading software vendors are committed to XML and moving toward adopting it, either as an internal data representation model for their software or just for data exportation and importation among different applications and platforms.

Like HTML, XML employs tags to structure data, but in XML tags are defined for each application and can be used to identify the meaning of the data. Web servers and applications encoding their data in XML can quickly make their information available in a simple and generally accepted format. Information content is separated from information rendering, making it easy to provide multiple views of the same data.

Though simple in its core definition, XML is much more general and flexible to represent data than other standards, including the relational data model. Powerful resources for manipulation of XML data, such as XML query languages [1], have been proposed and many tools are expected to be available in the future. Furthermore, the publication data in XML format can make the Web a huge data source for all sorts of information.

Therefore, using XML as the data representation standard can bring many benefits for data integration [1, 22]. As XML is a semi-structured data model it can lend versatility and openness to data representation and integration. If consensual semantics are associated to XML data and their markup, the activities involved in the data integration process can be simplified and distributed. The sources can export data in a format suitable for data integration, and a variety of views from the integrated data can be supplied, according to the peculiarities of each target system or tool.

However, XML without agreed upon semantics associated to data and tags does nothing to support integration, except for providing a common syntax. Thus, it is necessary to establish multiple agreements on application domain terminologies, taxonomies, and representations, to achieve true information systems interoperability with XML.

5.2 Semantic Heterogeneity

Data exists independently of the computer programs. Each area of knowledge use some kind of data to represent information. Data may differ among application domains, even when referring to the same objects of the real world, because people usually have different perceptions and interpretations of reality, according to the activities they do.

Semantics can be defined as a consensual meaning associated to data shared by some community, called an information community. Hence, the resolution of semantic conflicts relies on the determination and standardization of the meaning of the concepts, terms, and structuring constructs found in source data. It involves metadata enrichment to support the investigation of semantic matching among data items from distinct datasets.

The first step is to semantically describe data, by associating consensual descriptions to published and exchanged data. At this stage, the establishment of accord is usually possible only for small communities [16]. Common semantics can be expanded to wider communities, as information is better understood and appropriate levels of abstraction are devised to make possible data exchange with minimal loss of meaning.

Syntactical standards to represent data and addressing semantic heterogeneity at higher abstraction levels, as is done for solving syntactical conflicts, can help to elucidate semantic correspondence. Consensual semantics is sometimes easier to obtain (or only possible) at high abstraction levels, and in this case integration can occur with some loss of information. However, addressing semantic conflicts to higher abstraction levels just shifts the problem to a context more prone to consensus. Standard terminologies and classifications are necessary to break the abstraction process somewhere and establish agreement.

Data and Metadata Standards

The integration of semantically heterogeneous data requires specific domain knowledge. How to gather and represent this knowledge? To start answering this question it is useful to observe that semantic heterogeneity also involves several facets: some of them more related to implementation and others to the specific application domain. There are also semantics related to data instances, schemas, and data models. An environment aiming to support data integration must separate these concerns in order to factoring problems.

There are many metadata standards proposed on the literature, ranging from general standards for data representation to standards devoted to specific fields such as biology, medicine, agriculture, as well as some categories of applications such as digital libraries and electronic commerce. Among the most prominent efforts on developing metadata standards are [12, 29, 30]. Nowadays, many of the available metadata standards are being aligned with XML.

Ontologies

Ontologies are a means to structure knowledge to support interoperability. They capture semantics of diverse sources, by taking terms of the universe of discourse and arranging then according to their relationships [36]. When data sources are to be integrated, their ontologies can be used to drive the investigation of matching among their elements. There are many works in the literature investigating the use of ontologies for integration of heterogeneous geographical data [37, 13].

5.3 Geographical Information Systems

Geographical information systems (GIS) [2, 24, 9] are data integration and information analysis systems by their own, because they process data from many kinds of sources such as remote sensing, GPS, legacy systems, and digitalized maps. The GIS provide basic facilities to process all this data and present the results in suitable ways for information extraction and decision support.

The GIS market has been characterized by proprietary formats that make interoperability hard to achieve. Many standards have been proposed for exchanging geographical data among systems [31, 3], but scientists and practitioners have progressively found out that standard formats are not enough to enable interoperability [16]. The conversion of data through these formats often results on some kind of information loss. GIS interoperability requires additional levels of integration such as commonality of systems behavior and system-user interaction. The adoption of a common geographical data model [33, 4] or at least a framework to unify heterogeneous models [10] constitutes one ingredient to achieve this goal. Some endeavors are being conducted on developing standards for GIS interoperability in the wide sense, notably by the Open GIS Consortium [25, 16, 11].

6 Conclusions

The integration of heterogeneous data has been one of the greatest challenges in database research, and the advent of the Web is pushing the demand for solutions. Users do not want to cope with idiosyncrasies of data organization and processing to get the information they need, from large volumes of data spread among many data sources which can be connect via Internet. In this context, managing semi-structured data and answering informally specified queries (e.g. keyword searches) are problems very related to data integration. Solutions for these problems rely on relaxing data structuring, and enriching data with metadata to describe structure and meaning.

We believe that an environment to support data integration must be based on a common data model and the management of the data transformation and integration workflow. The common data model for data integration must be versatile and self-explanatory. Basic constructs and operations must allow the representation and manipulation of data in a wide range of abstraction levels. Data access must be driven by well established semantics, to propitiate minimal information loss when transferring data among information communities. Ideologically, a common data model should support exploratory navigation on heterogeneous database structure and contents, without expertise on databases in general or any specific data model or schema, but based on skills related to particular application domains.

Documenting the data transformation and integration workflow is very important to better understand the data interrelationships and data semantics. The management of this workflow could keep track of data genealogy, what is important to measure data quality (e.g. the accuracy of a particular dataset depends on the process employed to produce it). Thus, data processing and semantics cannot be dissociated. Additionally, the management of the information flow must transcend particular information systems boundaries, in order to empower interoperability of diverse tools.

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web – from relations to semistructured data and XML*. Morgan Kaufann Publishers, San Francisco, CA, 2000.

[2] T. Abraham and J. F. Roddick. Survey of spatio-temporal databases. *GeoInformatica*, 3(1):61–995, 1999.

[3] J. Albrecht. Geospatial information standards a comparative study of approaches in the standardisation of geospatial information. *Computers & Geosciences*, 25:9–24, 1999.

[4] K. A. V. Borges. Geographical data modeling: Extension of OMT for spatial applications. Master's thesis, Department of Computer Science, Federal University of Minas Gerais, Brazil, 1997. (in portuguese).

[5] S. Ceri, P. Fraternali, and S. Paraboschi. XML: Current developments and future challenges for the database community. In *Advances in Database Technology - 7th Int'l Conf. on Extending Database Technology (EDBT)*, volume 1777 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2000.

[6] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *ACM SIGMOD Record*, 26(1):65–74, 1997.

[7] V. Christophides, S. Cluet, and J. Siméon. On wrapping query languages and efficient XML integration. In *ACM SIGMOD Int'l Conf. on Management of Data*, pages 141–152. ACM Press, 2000.

[8] S. Cluet, C. Delobel, J. Siméon, and K. Smaga. Your mediators need data conversion! In *ACM SIGMOD Int'l Conf. on Management of Data*, pages 177–188. ACM Press, 1998.

[9] G. Câmara, M. A. Casanova, A. Hemerly, G. C. Magalhães, and C. B. Medeiros. *Anatomy of Geographical Information systems*. State University of Campinas, Brazil, 1996. (in portuguese).

[10] G. Câmara, A. M. V. Monteiro, J. A. Paiva, J. Gomes, and L. Velho. Towards a unified framework for geographical data models. *Journal of the Brazilian Computing Society*, 7(1), 2000.

[11] G. Câmara, R. Thomé, U. Freitas, and A. Monteiro. Interoperability in practice: Problems in semantic conversion from current technology to open GIS. In *2nd Int'l Conference on Interoperable GIS*, 1999.

[12] Dublin Core Metadata Initiative. `http://www.dublincore.org/` (accessed on 2nd. july 2001).

[13] F. T. Fonseca. *Ontology-Driven Geographic Information Systems*. PhD thesis, University of Maine, 2001.

[14] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.

[15] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database System Implementation*. Prentice Hall, Upper Saddle River, NJ, 2000.

[16] M. F. Goodchild, M. J. Egenhofer, and R. Fegeas. Interoperating GIS. Report Specialist Meeting under Auspices of the Varenius Project, U.S. National Center for Geographic Information and Analysis, Santa Barbara, CA, 1997.

[17] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In *20th Int'l Conf. on Data Engineering*, pages 152–159. IEEE Computer Society, 1996.

[18] W. Hasselbring. Information system integration. *Communications of the ACM*, 43(6):33–38, 2000.

[19] B. Hüsemann, J. Lechtenbörger, and G. Vossen. Conceptual data warehouse design. In *2nd Int'l Workshop on Design and Management of Data Warehouses*, 2000.

[20] W. H. Inmon. *Building the Data Warehouse*. John Wiley and Sons, New York, 1996.

[21] Won Kim and Jungyun Seo. Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18, 1991.

[22] Alon Levy. More on data management for XML, 1999. `http://www.cs.washington.edu/homes/alon/widom-response.html` (accessed on 2nd. july 2001).

[23] W. Litwin and A. Abdellatif. Multidatabase interoperability. *IEEE Computer*, 19(12):10–18, 1986.

[24] C. B. Medeiros and F. Pires. Databases for GIS. *ACM SIGMOD Record*, 23(1):107–115, March 1994.

[25] Open GIS Consortium Specifications. `http://www.opengis.org/techno/specs.htm` (accessed on 2nd. july 2001).

[26] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *11th Int'l Conf. on Data Engineering*, pages 251–260. IEEE Computer Society, 1995.

[27] C. Parent and S. Spaccapietra. Issues and approaches of database integration. *CACM*, 41(5):166–178, 1998.

[28] V. Poe, P. Klauer, and S. Brost. *Building a Data warehouse for Decision Support*. Prentice Hall, 1998.

[29] W3C's Resource Description Framework (RDF). `http://www.w3.org/RDF/` (accessed on 2nd. july 2001).

[30] W3C's Semantic Web Activity. `http://www.w3.org/2001/sw/` (accessed on 2nd. july 2001).

[31] S. Shekthar, S. Chawla, S. Ravada, A. Fetterer, Xuan Liu, and Chang tien Lu. Spatial databases - accomplishments and reseach needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 1999.

[32] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[33] N. Tryfona and C. S. Jensen. Conceptual data modeling for spatiotemporal applications. *GeoInformatica*, 3(3):245–268, 1999.

[34] J. Widom. Data management for XML: Research directions. *IEEE Data Engineering Bulletin*, 22(3):44–52, 1999.

[35] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(1):38–49, 1992.

[36] G. Wiederhold and J. Jannik. Composing diverse ontologies. Report, Stanford University, 1998.

[37] M. F. Worboys and S. M. Deen. Semantic heterogeneity in distributed geographic databases. *SIGMOD Record*, 20(4):30–34, 1991.

[38] W3C's Extensible Markup Language (XML). `http://www.w3.org/XML` (accessed on 2nd. july 2001).