7th ACM Symposium on Advances in Geographic Information Systems
Kansas City, MO
C. Bauzer Medeiros (ed.), November 1999.

Ontology-Driven Geographic Information Systems

Frederico T. Fonseca[12] and Max J. Egenhofer[123]
[1]National Center for Geographic Information and Analysis
[2]Department of Spatial Information Science and Engineering
[3]Department of Computer Science
University of Maine, Orono, ME 04469-5711, USA
fred@spatial.maine.edu max@spatial.maine.edu

ABSTRACT

This paper introduces a geographic information system architecture based on ontologies. Ontology plays a central role in the definition of all aspects and components of an information system in the so-called ontology-driven information systems. The system presented here uses a container of interoperable geographic objects. The objects are extracted from multiple independent data sources and are derived from a strongly typed mapping of classes from multiple ontologies. This approach provides a great level of interoperability and allows partial integration of information when completeness is impossible.

Keywords

GIS architecture, interoperability, ontology, object orientation.

1. INTRODUCTION

Ontology is an ancient discipline dating back to Aristotle's Categories and Metaphysics. After being used initially for Artificial Intelligence systems, it has been proposed to play a central role driving all aspects and components of an information system, leading to ontology-driven information systems [17]. This paper presents an architecture for an ontology-driven geographic information system (ODGIS). The system uses an object-oriented mapping of ontologies. A strongly typed mapping of classes from multiple ontologies provides a high level of interoperability. The need to share geographic information is well documented [16, 27, 37]. In the past, exchanging geographic information was simply sending paper maps or raw data tapes through mail. Today, there is a huge amount of data gathered about the Earth, computers throughout the world are connected, and the use of GIS has become widespread. Although spatial information systems have been characterized as an integration tool, GIS interoperability is far from being fully operational [35].

Imagine a scenario where there is a need to build a larger road over an existing one. The road is located at the border of two cities. A state representative needs to prepare a draft of a bill allocating funds to the project. In order to do this, the state representative needs an estimate of the value of the properties that have to be purchased to build the new road (Figure 1).

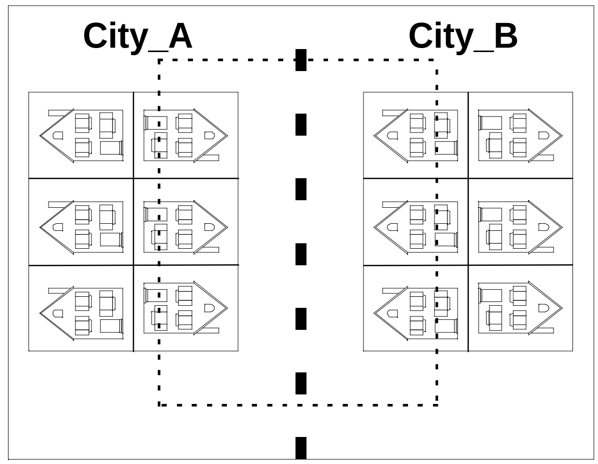

Figure 1: An interoperability problem

In order to solve this problem, it is necessary to (1) generate a buffer of 150 ft along the centerline of the existing road; (2) select and group the parcels touched by the buffer; and (3) add up the values of the properties. To do this with a geographic information system it is necessary to access the geographic databases of the two cities over a network, select and retrieve the desired features, and then apply the buffer operation (Table 1).

Type	Operation
Semantic search	Locate the two cities' geographic databases.
	Browse the databases looking for parcel and centerline features, or equivalents of these features.
	Define the geographic scope to do the search.
Objects browsing	Extract the features available in the chosen area.
	Display the retrieved features.
	Identify the centerline.
	Ask for a buffer operation.
	Select the parcels inside the buffer.
	Add up their values.

Table 1: The necessary steps to solve the problem

Research about semantic interoperability is today widespread both among the computer science community and the GIS community [5, 7, 13, 18, 20]. At the same time object extraction through geographic querying has been a well-studied theme [3, 8, 9, 19, 38]. In this paper we focus on the last five steps of Table 1, i.e., the direct manipulation, query, and use of extracted geographic objects.

The remainder of this paper is organized as follows. Section 2 gives an overview of interoperability, object orientation, and ontologies using a GIS perspective. Section 3 describes the architecture for an ODGIS. Section 4 shows the use of the proposed architecture. Section 5 presents conclusions and future work

2. GIS INTEROPERABILITY, OBJECTS, AND ONTOLOGIES

Research in interoperability is motivated by the ever-increasing heterogeneity of the computer world. Research in the integration of databases can be tracked back to the mid-80s [4] and now interoperability is turning into an integration science [41]. Heterogeneity in GIS is not an exception, but the complexity and richness of geographic data and the difficulty of their representation raise specific issues for GIS interoperability.

The literature shows many proposals for integration of data, ranging from federated databases with schema integration [32] and the use of object orientation [21, 28] to mediators and ontologies [39]. In this section we review work that deals with interoperability, objects, and ontologies

2.1 GIS and Objects

The object view of the geographic world is thoroughly discussed and established in [10, 14]. An outline of the main concepts behind the object-oriented approach and its application to geo-referenced information handling appears in [43].

The object view of the spatial world avoids problems like the horizontal and vertical partitioning of data. Furthermore, an object representation of the geographic world offers many views of an object through the use of multiple inheritance. Objects are also useful in zooming operations, because when we get closer to a scene instead of seeing enlarged objects we see different kinds of objects. This is achieved through aggregation as in the case of a house that consists of walls, or a block formed by land parcels [22].

2.2 Interoperability and Objects

The concept of object orientation to provide interoperability is used either in the implementation or in the modeling aspect. Soley and Kent [33] stress the ability of representing complex data structures and behavioral specifications as a reason for using object technology in interoperation. Kent [21] considers that object orientation has some features that are useful to enhance information compatibility as the use of object identity to link different sources and reconciliation of different levels of abstraction through subtyping.

Clients prefer to receive information in an object-oriented format when integrating multiple heterogeneous sources because objects enable aggregation of information into meaningful units. These units can have hierarchical linkages to other classes providing a valid model even for a complex world [28, 40].

2.3 GIS Interoperability

The first attempts to obtain GIS interoperability involved the direct translation of geographic data from one vendor format into another. A variation of this practice is the use of a standard file format. These formats can lead to information loss, as is often the case with the popular CAD-based format DXF. Alternatives that avoid this problem are usually more complex, like the Spatial Data Transfer Standard (SDTS) and the Spatial Archive and Interchange Format (SAIF). A modernization proposal for SDTS using an object profile that integrates a dynamic schema structure, an OpenGIS interface, and the Common Object Request Broker Architecture Interface Definition Language (CORBA IDL) is presented in [2]. A broader discussion of geographic information exchange formats can be found in [30].

One important initiative to achieve GIS interoperability is the OpenGIS Consortium. This is an association of government agencies, research organizations, software developers, and systems integrators, looking to define a set of requirements, standards, and specifications that will support GIS interoperability. According to the OpenGIS guide [24] "the objective is technology that will enable an application developer to use any geodata and any geoprocessing function or process available on 'the net' within a single environment and a single work flow." An OpenGIS product would have a structure much more complex than today's GIS applications framework. However, the complexity will be hidden from the user. Geographic data will be more readily available, understandable, and flexible. But to reach this point a great deal of commitment to information communities is needed. For instance, reliable translators require a lot of work. That commitment has a positive aspect with the improvement of the quality of data inside the information community. Usually the necessity of sharing occurs at a very low level, and it can be upgraded step by step to reach other communities [15].

Despite initiatives like SDTS, SAIF, and OpenGIS, the use of standards as the only worthwhile effort to achieve interoperability is not widely accepted. Since widespread heterogeneity arises naturally from a free market of ideas and products there is no way for standards to banish heterogeneity by decree [11]. The use of semantic translators in dynamic approaches is a more powerful solution for interoperability than the current approaches, which promote standards [5].

Another important question in GIS interoperability is semantics. The complex issue of the meaning of data and its description is discussed in [6]. Three types of heterogeneity are distinguished: (1) semantic heterogeneity, in which a fact can have more than one description; (2) schematic heterogeneity, in which the same object in the real word is represented using different concepts in a database; and (3) syntactic heterogeneity, in which the databases use different paradigms.

The idea of a Virtual GIS as an integrated system is discussed in [1] and [36].Abel *et al*. [1] propose an architecture to integrate and retrieve information from multiple component systems. It distributes the processing load through the global front end and the components. It is based on an object-oriented model and uses the schema integration ideas presented in [32]. Vckovsky [36] proposes the use of a well-defined canonical interface to access multiple spatial databases. The Virtual Data Set (VDS) corresponds to a protocol between the data consumer and the data producer. VDS is based on the object orientation paradigm.

2.4 Ontology-Driven Information System

The difference between ontology in the philosophical sense and in the way the term is used by the Artificial Intelligence community is given in [17]. Ontology as an engineering artifact describes a certain reality with a specific vocabulary using a set of assumptions regarding the intended meaning of the vocabulary words. A particular system of categories reflecting a specific view of the world is the philosophical meaning of ontology. Guarino [17] presents a refined definition of the difference between an ontology and a conceptualization. "An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e., its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models."

We can speak of ontology-driven information systems (ODIS) when an explicit ontology plays a central role in the system's life cycle. In this case the ontology drives all aspects and components of the system. Ontologies can be used at development time or at run time. In ODIS the ontology is called application ontology and it is a specialization of a domain ontology and a task ontology [17].

The use of ontologies as a support for interoperability is discussed in the next section, where the proposed architecture for an ODGIS is presented. A detailed analysis of ontologies and interoperability can be found in [25, 42].

3. ONTOLOGY-DRIVEN GEOGRAPHIC INFORMATION SYSTEM

An object-oriented view of the world is useful to represent geographic entities. Object technology can be used as a tool for promoting interoperability. Interoperable geographic objects can be held in a container. Since objects carry both operations and attributes, the container can extract its functionality from them and make it available to the end user. The architecture of the system is basically a container of interoperable objects. The container communicates with the end user and the data warehouses, commanding the extraction of information and serving the results to the end user.

This idea presents two kinds of questions. First, in relation to the objects, what kind of objects are these? What are their interfaces? How are they generated? How do they communicate with the container? Second, how can the container hold objects from different sources?

To answer the objects questions we propose an object-oriented mapping of the objects from multiple ontologies. Using this mapping, the objects inside the container are derived from application ontologies. The result is an object factory, in which objects are generated. It is open to discussion if integration of data from multiple sources should be done at the server, at the client, or at the middleware [40]. In the proposed architecture the generation of objects can take place either at the data warehouses, at the query processor, or at the container, as long as objects are generated from application ontologies. The use of multiple ontologies to generate objects is possible through the use of multiple inheritance. This provides more flexibility to the system allowing partial integration of information when complete integration is impossible. The combined use of multiple ontologies and multiple inheritance within the proposed approach overcomes the weakness in type hierarchy and conflicts in operation's signature, as indicated by [26]. The use of multiple ontologies is fundamental in developing a system with multiple information producers and consumers [31]. Multiple inheritance also leads to multiple views of each object.

To answer the container question we propose a distributed object framework. This kind of framework allows objects written in different languages, and compiled by different compilers, to communicate seamlessly via standardized messaging protocols. These frameworks implement higher levels of interoperability between distributed objects [23]. Implementation of the container can be made using the Object Management Group (OMG) object-oriented framework. A Java end-to-end or Java/CORBA solution can be used to implement the objects. However, the implementation description is beyond the scope of this paper.

3.1 Architecture

The main components of the architecture are the ontologies, the container, the data warehouses, and the user interface. The architecture includes a coordinator that integrates all other components. The coordinator also is in charge of finding services on the network, and redirecting them to the components. These services can include an ontology-based search engine for geographic information [12], a query language server, or a framework that can function as a user interface with other embedded services. The focus of this paper is to provide an outline of the architecture of an ODGIS with an emphasis on the container and the ontologies. The user interface, the data warehouses, and the cooperating services are beyond the scope of this paper.

3.1.1 Ontologies

Ontologies here are seen as dynamic, object-oriented structures that can be navigated. An ontology server that holds a standard catalog of ontologies is available for the user to search and browse. Kashayp and Sheth [20] present a similar solution for ontology browsing from a software-engineering point of view. The basic service of this component is to find ontologies, let the user browse them, and then select the necessary nodes of the ontology to perform the required task. Once a geographic scope is delimited and the classes chosen, the ontology server provides pointers to data warehouses. The relationship between the ontologies and the data warehouses is considered to be previously existent.

3.1.2 Container

The objects are held inside a container. The idea of the map space as a container is presented in [34] considering that a container holds other containers. Here, a container is considered to accommodate just objects. Views can be derived from this container and represent the different interfaces of the objects achieved through multiple inheritance. Each parent class corresponds to a role played by the new object. The model for implementing objects with roles has been suggested in [29]. For example, an instance of the class parcel of city B can be perceived as a parcel or as a polygon. Although an instance of a class can be viewed through multiple interfaces, it never loses its identity. The container holds the unique object and makes it available through its interfaces. An object has to adapt itself to various views and relationships through changes of classes, usually from a more detailed to a less detailed class. For instance, if a polygonal object is asked to merge with another polygonal object in its superclass, this object has to adapt itself by generating an instance of the superclass. Data are never lost in conversions, because they never occur in the original object, but only in its representations.

3.1.3 Application Objects

The geographic objects implement or inherit all methods of the classes from which they are derived. Since we are proposing a distributed object framework, the implementation of the objects can rely upon services of external objects to perform the methods. For instance, when implementing a census tract class, a software developer should write a method to extract the population of a subarea of a census tract. Instead of developing this method, the developer can call a service that has a more advanced implementation. From the point of view of the caller, the object method is executed locally, but actually it is performed by another method using services provided elsewhere.

As in CORBA and Java, where all classes are derived from a basic class, we propose here that all classes should be derived from a basic class, called Object. This class has two methods that are fundamental for the whole system, the methods Up() and Create_From(). The method Up(), when applied to an object returns an object of the immediate superclass. The method Create_From() instantiates a version of the class from an instance of the immediate superclass. These two methods provide the means to navigate through the whole ontology tree.

Once a set of objects are retrieved from a data warehouse the user needs methods to manipulate it. The methods from the Object class are always available. The other methods depend on the selection of objects. If the user selects only one, then all the public methods of the object's class are available. But the proposed system becomes more interesting when objects from multiple classes are selected. Then an upward navigation through each parent class can be performed. The points of intersection are the classes that have their methods made available to the user.

3.2 Views

Since the system has an object-centered design and also supports multiple inheritance it leads to multiple views of the data. An object plays many roles. These roles represent many views to the user. The views are derived from the multiple ontologies and are mapped one by one from them. These views can also be combined, generating new views. We can have a geometric view, a network view, or an alphanumeric relational-like.

All views of one object are mapped directly onto the original object in the data warehouse. There is only one object, but it can be seen with many faces. We apply here the concept of object identity defined by OMG, in which each object has a unique identity that is distinct from and independent of any of its characteristics. The identity of the object is constant, although its characteristics may vary during its lifetime.

The views can be combined enabling the user to have a geometric/alphanumeric view. An example of the use of this combined view is a "point-and-click" operation over a parcel that highlights its shape and shows its alphanumeric data.

4. USING THE SYSTEM

In order to use the system the user starts the coordinator, which provides a user interface. Through the interface the user defines the scope of the area and browses through the available ontologies to choose the classes to work with, although not necessarily in this order. First the user can browse and select classes and afterwards apply the scope as a constraint. A second option is to choose first the scope and then to browse only the available classes for that area. The result of browsing the ontologies is as set of classes that the user is interested in and a defined spatial area to work with. The browsing can be done through a semantic search engine and the definition of the scope can be provided by a scope defining service through interactive navigation of key maps. The result of these operations should be a pointer to data warehouses in a network. Now the user can query the data warehouses for geographic objects. The resulting objects are stored in the object container. The coordinator queries the objects to retrieve their classes. Using these classes as input the coordinator queries the ontologies to get the methods available for theses classes and offers them to the user through the interface.

In the motivating example (Figure 1), City_A and City_B have the necessary geographic data to solve the problem. Both have parcels and centerlines stored in two different GISs. The chosen models to represent parcels are different. City_A did not have enough funds to digitize the boundaries of the parcels. So it decided to represent parcels as symbols using only one X,Y location with all the alphanumeric data associated to it. City_B represented the parcels as polygons. Both cities have lines representing the central axis of the streets through centerlines (Figure 2).

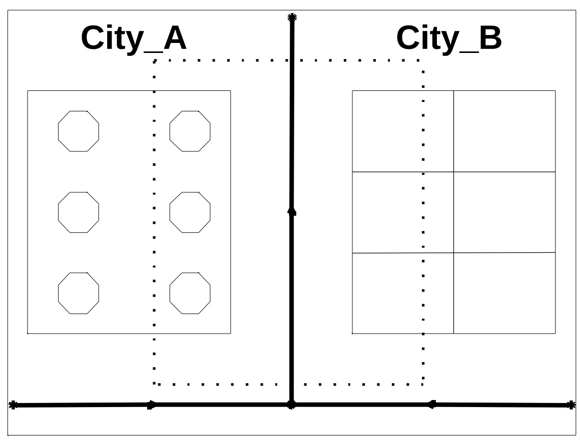

Figure 2: Different representations of land parcels

In order to create the classes necessary to represent this information two ontologies are used: an Urban ontology and Geometry ontology. The parcel class for City_A is derived from two classes, Parcel and Symbol, generating Parcel_A class and for City_B, Parcel_B came from Parcel and Polygon. In this example two characteristics of retrieved objects are used. First, the geometry role is necessary to answer the "touched by" question posed by the buffer. Second, the parcel role is used to answer the "value of property" question.

In this case the usefulness of the upward navigation in the inheritance tree is clear. This problem can only be solved using methods of the intersection point in both trees: value in Parcel and location in Geometry. Location is used to answer the question if the geometry of the object is touched by or inside the buffer polygon. Value is used by the sum function.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have reviewed issues of the relation between geographic information systems and object-oriented techniques, interoperability, and ontologies. Through this review this paper presented an outline of an ontology-driven geographic information system (ODGIS). Such systems are characterized by an extensive use of ontologies in the development phase and in the use of the system. The mapping of multiple ontologies onto the system classes was done through object-oriented techniques using multiple inheritance. This kind of mapping allowed partial integration of information when completeness was impossible. This paper has demonstrated that ODGIS can play an important role in the world of interoperable and distributed systems.

We presented a GIS characterized by its objects rather than by its interface or database, since it is in the objects that all the functionality of the system is embedded.

A variety of issues remain to be resolved. One of them is to explain in more detail the architecture and its elements and how it can be implemented using current distributed object technology, like OMG, CORBA, and Java. The study of how the currently available ontologies for geographic domains can be applied in ODGIS should be also subject of further study.

6. ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under NSF grant number SBR-9700465. Max Egenhofer's research is further supported through NSF grants IRI-9613646, BDI-9723873 and IIS-9970123; the National Imagery and Mapping Agency under grant number NMA202-97-1-1023; the National Institute of Environmental Health Sciences, NIH, under grant number 1 R 01 ES09816-01; and Bangor Hydro-Electric Co.; and a Massive Digital Data Systems contract sponsored by the Advanced Research and Development Committee of the Community Management Staff.

7. REFERENCES

[1] Abel, D. J., Ooi, B. C., Tan, K.-L., and Tan, S. H., "Towards Integrated Geographical Information Processing," *International Journal of Geographical information Science*, vol. 12, pp. 353-371, 1998.

[2] Arctur, D., Hair, D., Timson, G., Martin, E. P., and Fegeas, R., "Issues and Prospects for the Next Generation of the Spatial Data Transfer Standard (SDTS)," *International Journal of Geographical information Science*, vol. 12, pp. 403-425, 1998.

[3] Aref, W. and Samet, H., "Optimization Strategies for Spatial Query Processing," 17th International Conference on Very Large Data Bases, Barcelona, Spain, 1991.

[4] Batini, C., Lenzerini, M., and Navathe, S. B., "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys*, vol. 18, pp. 323-364, 1986.

[5] Bishr, Y., "Semantic Aspect of Interoperable GIS,": Wageningen Agricultural University ITC, 1997, pp. 154.

[6] Bishr, Y., "Overcoming the Semantic and Other Barriers to GIS Interoperability," *International Journal of Geographical Information Science*, vol. 12, pp. 299-314, 1998.

[7] Breitbart, Y., Garcia-Molina, H., Litwin, W., Roussopoulos, N., Risinkiewicz, M., Thompson, G., and Wiederhold, G., "Final Report of the Workshop on Multidatabases and Semantic Interoperability, Tulsa, Oklahoma, November 2-4, 1990," University of Kentucky, Department of Computer Science, Lexington, KT 188-91, 1990.

[8] Clementini, E. and Felice, P. d., "Multiparadigm Query Strategies in Object-Oriented Geographic Databases," International Phoenix Conference on Computers and Communications, Scottsdale, AZ, 1991.

[9] Egenhofer, M., "Spatial SQL: A Query and Presentation Language," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, pp. 86-95, 1994.

[10] Egenhofer, M. and Frank, A., "Object-Oriented Modeling for GIS," *Journal of the Urban and Regional Information Systems Association*, vol. 4, pp. 3-19, 1992.

[11] Elmagarmid, A. K. and Pu, C., Introduction to the Special Issue on Heterogeneous Databases," *ACM Computing Surveys*, vol. 22, pp. 175-178, 1990.

[12] Finch, I. and Small, E., "Information Brokers for a WEB-based Geographic Information System," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 165-180.

[13] Gahegan, M., "Characterizing the Semantic Content of Geographic Data, Models, and Systems," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 71-84.

[14] Gahegan, M. N. and Roberts, S. A., "An Intelligent, Object-oriented Geographical Information System," *International Journal Geographical Information Systems*, vol. Vol. 2, pp. 101-110, 1988.

[15] Gardels, K., "The Open GIS Approach to Distributed Geodata and Geoprocessing," Third International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, NM, 1996.

[16] Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C., *Interoperating Geographic Information*

Systems. Norwell, MA: Kluwer Academic Publishers, 1999.

[17] Guarino, N., "Formal Ontology and Information Systems," in *Formal Ontology in Information Systems*, N. Guarino, Ed. Amsterdam, Netherlands: IOS Press, 1998.

[18] Harvey, F., "Designing for Interoperability: Overcoming Semantic Differences," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 85-98.

[19] Herring, J., Larsen, R., and Shivakumar, J., "Extensions to the SQL Language to Support Spatial Analysis in a Topological Data Base," GIS/LIS '88, San Antonio, TX, 1988.

[20] Kashayp, V. and Sheth, A., "Semantic Heterogeneity in Global Information System: The Role of Metadata, Context and Ontologies," in *Cooperative Information Systems: Current Trends and Directions*, M. P. Papazoglou, Schlageter, G., Eds. London: Academic Press, 1996.

[21] Kent, W., "Object Orientation and Interoperability," Advances in Object-Oriented Database Systems, Izmir, Kusadasi, Turkey, 1994.

[22] Kuhn, W., "Are Displays Maps or Views?" Auto-Carto 10, Baltimore, 1991, pp. 261-274.

[23] Lewandowsky, S. M., "Frameworks for Component-Based Client/Server Computing," *ACM Computing Surveys*, vol. 30, pp. 4-27, 1998.

[24] McKee, L. and Buehler, K., "The Open GIS Guide", MA: Open GIS Consortium, Inc, 1996.

[25] Mena, E., Kashyap, V., Illarramendi, A., and Sheth, A., "Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure," in *Formal Ontology in Information Systems*, N. Guarino, Ed. Amsterdam, Netherlands: IOS Press, 1998.

[26] Nittel, S. and Muntz, R. R., "Supporting Interoperation of GIS Objects," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 165-180.

[27] OGIS, "The OpenGIS® Guide - Introduction to Interoperable Geoprocessing and the OpenGIS Specification,", L. McKee and K. Buehler, Eds., MA: Open GIS Consortium, Inc, 1996.

[28] Papakonstantinou, Y., Garcia-Molina, H., and Widom, J., "Object Exchange Across Heterogeneous Information Sources," IEEE International Conference on Data Engineering, Taipei, Taiwan, 1995, pp. 251-260.

[29] Pernici, B., "Objects with Roles," ACM Conference on Office Information Systems, 1990, pp.205-215.

[30] Salgé, F., "National and International Data Standards," in *Geographical Information Systems*, vol. 1 Principles and Technical Issues, P. Longley, M. Goodchild, D. Maguire, and D. Rhind, Eds. New York: John Wiley & Sons, 1999, pp. 693-706.

[31] Sheth, A. P., "Changing Focus on Interoperability in Information Systems," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 165-180.

[32] Sheth, A. P. and Larson, J. A., "Federated Databases Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, vol. 22, pp. 183-236, 1990.

[33] Soley, R. M. and Kent, W., "The OMG Object Model," in *Modern Database Systems: the Object Model, Interoperability and Beyond*, W. Kim, Ed. Addison-Wesley Publishing Company, 1995, pp.

[34] Timpf, S. and Frank, A., "Exploring the Life of Screen Objects," Auto-Carto 13, Seattle, USA, 1997,pp.195-203.

[35] Vckovski, A., Brassel, K., and Schek, H.-J., "Interoperating Geographic Information Systems-Second International Conference, INTEROP'99, Zurich, Switzerland," vol. 1580, *Lecture Notes in Computer Science*. Berlin: Springer-Verlag, 1999, pp. 1-16.

[36] Vckovsky, A., "Interoperable and Distributed Geoprocessing,", Zurich, 1997.

[37] Vckovsky, A., *International Journal of Geographical Information Science - Special Issue: Interoperability in GIS*, vol. 12 - Number 4, 1998.

[38] Voisard, A. and Jurgens, M., "Geospatial Information Extraction: Querying or Quarrying?" in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 165-180.

[39] Wiederhold, G., "Interoperation, Mediation and Ontologies," International Symposium on Fifth Generation Computer Systems, Tokyo, Japan, 1994.

[40] Wiederhold, G., "Value-added Middleware: Mediators," Stanford University March 1998.

[41] Wiederhold, G., "Mediation to Deal with Heterogeneous Data Sources," in *Interoperating Geographic Information Systems, INTEROP'99, Zurich, Switzerland*, vol. 1580, *Lecture Notes in Computer Science*, A. Vckovski, K. Brassel, and H.J. Schek, Eds. Berlin: Springer-Verlag, 1999, pp. 1-16.

[42] Wiederhold, G. and Jannink, J., "Composing Diverse Ontologies," Stanford University 1998.

[43] Worboys, M. F., "Object-oriented approaches to Geo-referenced Information," *International Journal of*

Geographical Information Systems, vol. 8, pp. 385-399, 1994.