Ontology-Driven Information Integration

Frederico Fonseca[1] **Max Egenhofer**[1] **Clodoveu Davis**[2]

[1]National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469-5711, USA

[2]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
and
Prodabel - Empresa de Informática e Informação do Município de Belo Horizonte
Belo Horizonte MG Brazil

Abstract

The integration of information of different kinds, such as spatial and alphanumeric information, at different levels of detail is a challenge. While a solution is not reached, it is widely recognized that the need to integrate information is so pressing that it does not matter if detail is lost, as long as integration is achieved. This paper shows the potential for extraction of different levels of information inside the framework of ontology-driven geographic information systems.

Introduction

Today, there is a huge amount of data gathered about the Earth, not only from new spatial information systems, but also from new and more sophisticated satellites. At the same time, the continuous expansion of the global network and new application domains have introduced important changes to application development. Contemporary information systems are distributed and heterogeneous, which leads to a number of interesting research challenges regarding spatial information systems. One of them is on how to integrate information of different kinds, such as spatial and alphanumeric information, at different levels of detail. It is widely recognized that the need to integrate information is so pressing that it does not matter if detail is lost, as long as integration is achieved. This paper shows the potential for extraction of different levels of information inside the framework of ontology-driven geographic information systems. We also show the use of a navigation method inside an ontology-derived class hierarchy as a guide to generalization operations, and how the use of ontologies in GIS development can enable knowledge sharing and information integration. The proposed approach provides dynamic and flexible information exchange and allows partial integration of information when completeness is impossible.

Ontologies and IS development

The next generation of information systems should be able to solve semantic heterogeneity to make use of the amount of information available with the arrival of the Internet and distributed computing. Ontologies play a key role in enabling semantic interoperability, and Sheth (1999) believes that research should focus on a specific domain, such as geographic information systems (GIS), before more general architectures can be developed.

Ontology-driven information systems (ODIS) (Guarino 1998) are based on the explicit use of ontologies at development time or at run time. The use of ontologies in GIS development has been discussed by Frank (1997) and Smith and Mark (1998). Ontology playing a software specification role was suggested as early as 1991 (Gruber 1991). Also Nunes (1991) pointed out that the first step in building a next-generation GIS would be the creation of a systematic collection and specification of geographic entities, their properties, and relations. Ontology plays an essential role in the construction of GIS, since it allows the establishment of correspondences and interrelations among the different domains of spatial entities and relations (Smith and Mark 1998). Frank (1997) believes that the use of ontologies will contribute to better information systems by avoiding problems such as inconsistencies between ontologies built in GIS, conflicts between the ontological concepts and the implementation, and conflicts between the common-sense ontology of the user and the mathematical concepts in the software. Kuhn (1993) asks for spatial information theories that look toward GIS users instead of focusing on implementation issues. Ontology use can also help GIS to move beyond the map metaphor, which sees the geographic world as layers of independent information that can be overlaid. Several inadequacies of the map metaphor have been pointed out (Kuhn 1991).

Philosophers and software engineers, however, have a different perspective of ontologies. 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 that reflects a specific view of the world is the philosophical meaning of ontology (Guarino 1998). Smith (1998) notes that since ontology for a philosopher is the science of being, of what is, it is inappropriate to talk about multiple ontologies as engineers do. To solve this problem Smith suggests a terminological distinction between referent or reality-based ontology (R-ontology) and elicited or epistemological ontology (E-ontology). R-ontology is a theory about how the whole universe is organized, and corresponds to the philosopher's point of view. An E-ontology, on the other hand, fits the purposes of software engineers and information scientists, and is defined as a theory about how a given individual, group, language, or science conceptualizes a given domain.

This paper describes work in development and the expected results are a working prototype of an ontology-driven geographic information system, an geo-ontology editor, an geo-ontology translator with some basic geographic classes.

The remainder of this paper is organized as follows. Section 2 describes the framework for an ontology-driven geographic information system. Section 3 shows the GIS and ODGIS perspectives of information granularity. Section 4 presents the proposed navigation mechanism. Section 5 presents conclusions and future work.

Ontology-Driven Geographic Information Systems

The use of an ontology, translated into an active information system component, leads to Ontology-Driven Information Systems (ODIS) (Guarino 1998) and, in the specific case of GIS, it leads to what we call Ontology-Driven Geographic Information Systems (ODGIS) (Fonseca and Egenhofer 1999). ODGIS are built using software components derived from various ontologies. These software components are classes that can be used to develop new applications. Being ontology-derived, these classes embed knowledge extracted from ontologies. The object-oriented approach for describing the geographic world is thoroughly discussed and established (Gahegan and Roberts 1988; Egenhofer and Frank 1992). The use of the object data model as the basic conceptualization of space has been discussed by Nunes (1991), where he argues that the issue of defining geographic space is actually the issue of defining and studying the geographic objects, their attributes, and relations. Discussing the main concepts behind the object-oriented approach and its application to geo-referenced information, Worboys (1994) considers that this approach can describe both field-based and object-based spatial models.

The ODGIS framework is presented in the next two sections focusing on the aspects of knowledge generation and use (Figure 1).

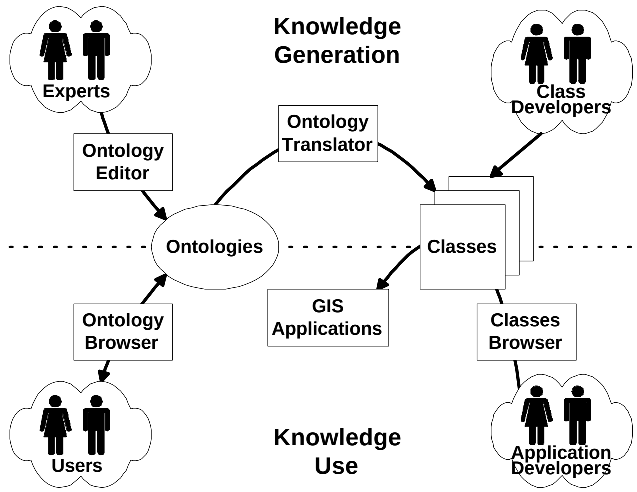

Figure 1: ODGIS framework

Knowledge Generation

The first step to build an ODGIS is to specify the ontologies using an ontology editor. The editor is able to translate the ontologies into a formal language to be used in a computer implementation. The ODGIS ontology editor is able to import and export ontologies written in XOL (Karp *et al.* 1999), an ontology exchange language used in object-oriented knowledge representation systems, which is based on the extensible markup language (XML) syntax (Graham and Quin 1999). The ontology editor also translates ontologies into Java interfaces. A Java interface describes the set of public methods that must be supported by a class that implements the interface, and also their calling conventions. However, a Java interface does not implement those methods. Each descendant class has to provide the code for each existing interface method. Since new classes can implement more than one interface, multiple inheritance can be achieved in Java (Gosling and McGilton 1995). These interfaces need to be complemented with Java code by independent class developers, thus generating Java classes.

Knowledge Use

The result from the knowledge generation phase of an ODGIS is a set of ontologies specified in a formal language and a set of classes. The ontologies are available to be browsed by the end user and they provide metadata information about the available data. The result from the translation is a set of classes that contain data and operations that constitute the system's functionality. These classes contain the knowledge available to be included in the new ontology-based systems.

The application developer can derive new classes, more specific to the application, called user classes, which are different from more generic ontology classes. User classes belong to the application ontologies level, while ontology classes belong to top-level, domain, or task ontologies (Guarino 1998). An application developer can build an application using either ontology classes or user classes, although they are separated here for clarification purposes. After building ontology classes, it is important to have a mechanism that enables the application developer to create user classes. User classes represent objects that have diverse characteristics. For instance, geographic features usually have geometric characteristics along with alphanumeric attributes. We propose here the use of multiple inheritance (Cardelli 1984) to define these kinds of classes. The use of multiple inheritance allows the developer to make use of the existing ontologies to build new classes. A geographic object should, for instance, descend from both geometric classes and feature-oriented classes. In the first group, all necessary representational and locational data can be handled by inherited methods, while in the other information on the semantics and behavior of the feature are inherited from specific ontology-derived classes.

Information Granularity

The abstraction of concepts and notions about real-world objects is an important part of the creation of information systems. In the abstraction process, certain characteristics of the objects are identified and coded in a database in such a way that the set of characteristics is representative of the much more complex real-world object. Depending on the user's interest, however, this set of characteristics can be defined to be more or less detailed. Furthermore, the increase in the level of detail can be directed towards a particular set of characteristics, as required by the user. Object-oriented modeling suggests that it is convenient to define a more generic set of concepts about an object, and then define a number of its specializations. In each specialization, a number of additional characteristics is recorded, thereby increasing the level of detail of the original object. Additionally, each specialization inherits the characteristics of the more generic object class. Different specializations can then be compared and mixed if the additional detail is disregarded, and only the common characteristics are used.

The GIS perspective

The notion of granularity applied to GIS leads to studies of the variation of the representation of geographic objects and phenomena across a wide range of scales. Certain phenomena are scale-dependent, i.e., their representation varies across the scales. For instance, if an urban settlement is perceived at a small scale, the level of detail is usually small, enough for an entire city, with all its complex internal structure, to be represented as a point or as a simple polygon on a map. If the same city is perceived at a larger scale, it becomes necessary to represent its internal structure with more detail, for instance depicting blocks, squares, major streets, and buildings. If both representations have to coexist in a geographic database, how could it be possible to maintain and update only the most detailed version of the objects, and then filter out unwanted detail to produce the less-detailed version ? (Beard 1987) Ontologies offer a possible solution to this problem, by specifying exactly how high-level abstractions such as `Urban Settlement` relate (conceptually) to concepts of a lower level, such as `Blocks`, by establishing methods that, say, determine that if a block has less than 50% of its parcels occupied by houses, then it is not to be considered a part of the urban settlement. This kind of rule is highly dependent on the set of concepts the users have about blocks and settled areas and, therefore, must be conveniently included in the Urban Settlement ontology.

The same set of notions applies to the reverse case. Suppose that some state organization, such as a property records office, defines a set of concepts about parcels. These concepts are very generic in the sense that the particular interest of the property records office is not on fine details regarding the parcels, but on higher-level information, such as ownership, surface, and value. In terms of location, it is only necessary to identify to which city the parcel belongs. This set of concepts is then used to create a very generic `Parcel` ontology. From that ontology, each interested city can define a new ontology in which further concepts are incorporated. There may be city-specific concepts, defined by municipal law, such as land use constraints, or common concepts, such as the representation of the parcel with a polygon. If the parcels that belong to a city need to be compared or analyzed together with some other city's parcels, both sets of parcels will have to be taken back one step in the ontological hierarchy, by considering them as members of the more generic ontology-derived `Parcel` class.

The ODGIS perspective

In the ODGIS architecture there are different levels of ontologies. Accordingly there are also different levels of information detail. Low-level ontologies correspond to very detailed information and high-level ontologies correspond to more general information. Thus, if a user is browsing high-level ontologies he or she should expect to find less detailed information. We propose that the creation of more detailed ontologies should be based on the high-level ontologies, such that each new ontology level incorporates the knowledge present in the higher level. These new ontologies are more detailed, because they refine general descriptions of the level from which they inherit (Figure 2).

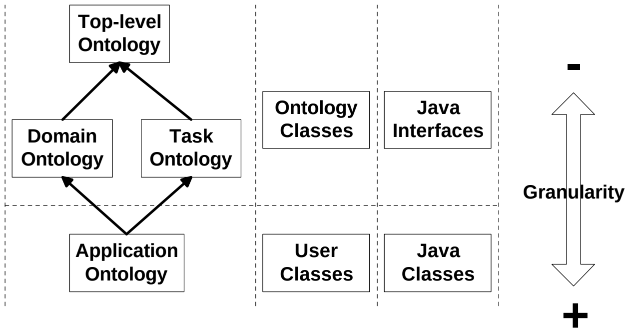

Figure 2: Granularity levels, extended from Guarino(1997)

Guarino (1997) classifies ontologies according to their dependence on a specific task or point of view.

- *Top-level ontologies* describe very general concepts. In ODGIS a top-level ontology describes a general concept of space. A theory describing parts and wholes, and their relation to topology, called mereotopology (Smith 1995), is at this level.
- *Domain ontologies* describe the vocabulary related to a generic domain, which in ODGIS can be remote sensing or the urban environment.
- *Task ontologies* describe a task or activity, such as image interpretation or noise pollution assessment in ODGIS.
- *Application ontologies* describe concepts depending on both a particular domain and a task, and are usually a specialization of them. In ODGIS these ontologies are created from the combination of high-level ontologies. They represent the user needs regarding a specific application such as an assessment of lobster abundance on the Gulf of Maine.

Another distinction that Guarino (1998) proposes is between coarse and fine-grained ontologies, or on-line and off-line ontologies. A coarse ontology consists of a minimal number of axioms and is intended to be shared by users that already agree on a conceptualization of the world. A fine-grained ontology needs a very expressive language and has a large number of axioms. Therefore, Guarino (1998) concludes that coarse ontologies are more likely to be shareable and should be used on-line to support the system's functionality. On the other hand, fine-grained ontologies should be used off-line, because they are accessed eventually for reference purposes. Our solution allows the user to incrementally go from coarse to fine-grained ontologies on-line, eliminating thus the division between on- and off-line ontologies.

Objects and Navigation

In ODGIS, ontologies are translated into classes. All classes derive from an initial class, called Object. This class has special operations for navigation in the ontology tree. The combined use of objects and ontologies provides a rich model to represent geographic entities, avoiding the problems of poor representation (Nunes 1991). In this section we describe the basic structure of an object in ODGIS and the mechanism for navigation.

Inheritance Mechanism

When one class inherits directly from only one class, it is called single inheritance, and when a class inherits from more than one class, it is called multiple inheritance (Cardelli 1984). In general, user classes have to inherit from more than one ontology class, therefore, the ability to implement multiple inheritance is necessary to build new classes from various ontologies. Multiple inheritance is a controversial concept, with benefits and drawbacks (Tempero and Biddle 1998) and we decided to use it because we believe that it is the best way to represent such complex features as geographic objects.

As an example, consider a class that is used to represent land parcels. This class needs to include information on the geometric shape of the parcel, as well as associated information about ownership, land use limitations, and so on. The `geographic parcel` class should descend from both a `Parcel` class and a `Polygon` class. Therefore, instead of having a single class that contains its own geometry and the methods necessary to handle it, there is a class that inherits such geometric characteristics and methods from a more generic `Polygon` class, and inherits application-specific characteristics and methods from a more generic `Parcel` class. This approach allows methods that deal with the geometry of the representations to have direct access to the geometry of the parcel, instead of relying on the `Parcel` class to handle its geometry to other class for appropriate treatment. This solution is achieved through interface conformance. A class should conform to every parent class, so the parcel of the example can be seen and treated as both an instance of `Polygon` and an instance of `Parcel` (Figure 3).

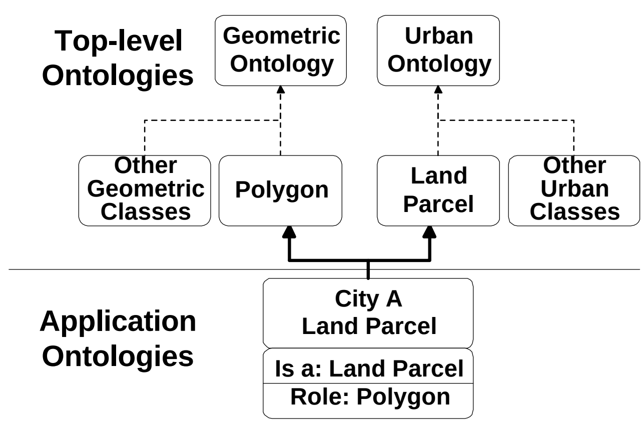

Figure 3: Classes with roles

The usual alternative for the multiple inheritance mechanism is to define the `Parcel` class as a descendant of the `Polygon` class. This succeeds in giving the `Parcel` class direct access to both its geometry and its application-specific characteristics, but makes it harder to implement different representations for the parcel, as demanded by other users or applications. For instance, another application might represent parcels using symbols or the parcel's front line. The proposed mechanism avoids this problem, and allows the existence of multiple representations for the same class, each of them adequate for a range of scales or a specific use. In many cases, the most detailed representation can be considered a primary representation, and the alternative representations can be obtained from it, using a set of transformation operators (Davis and Laender 1999).

Special Classes

The idea of a basic class, such as the `Object` class of ODGIS, is suggested both in object-oriented frameworks such as Java (Gosling and McGilton 1995) and CORBA (OMG 1991). This idea is also present in ontology literature, where many ontologies start with a concept named `Thing`. The `Object` class contains the basic methods used for navigation, called `Up()` and `Create_From()`. All other functionality is implemented using roles. An object has at least a basic role and a variable number of additional ones.

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. Since each class in the ontology tree is derived from the basic class, each interface inherits the necessary navigation tools.

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 copies or instances.

Conclusions

We presented a solution for information integration inside a framework of ontology-driven information systems. The solution allows for different levels of information sharing using a navigation system inside a hierarchy of classes derived from ontologies.

We presented a navigation system in which objects can generate new instances with different levels of information detail, and proposed the use of a special parent class that allows navigation from application ontologies to top-level ontologies, passing through domain and task ontologies. This navigation capability shortens the gap between generic and specialized ontologies, enabling the sharing of software components and information. ODGIS employs user classes that are derived through multiple inheritance from various ontologies to solve the problem of schematic heterogeneity (Bishr 1997).

We also presented here how ontology-driven geographic information systems can deal with different levels of information. The solution presented here tries to shorten the gap between coarse and fine-grained ontologies in ontology-driven information systems (Guarino 1998) by allowing navigation through ontologies of all specialization levels.

Acknowledgments

This work was partially supported by the National Science Foundation under NSF grant numbers SBR-9700465 and IIS-9970123, NASA/EPSCoR fellowship grant number 99-58 and an ESRI graduate fellowship. Max Egenhofer's research is further supported by NSF grants IRI-9613646, BDI-9723873, and EIA-9876707; by the National Imagery and Mapping Agency under grant number NMA202-97-1-1023; by the National Institute of Environmental Health Sciences, NIH, under grant number 1 R 01 ES09816-01 and by a contract with Lockheed Martin.

References

Beard, K. (1987) How To Survive A Single Detailed Database. in: N. R. Chrisman, (Ed.) *AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography*, Baltimore, MD.

Bishr, Y. (1997) *Semantic Aspect of Interoperable GIS*. Ph.D. thesis, Wageningen Agricultural University, The Netherlands.

Cardelli, L. (1984) A Semantics of Multiple Inheritance. in: G. Kahn, D. McQueen, and G. Plotkin, (Eds.), *Semantics of Data Types*. *Lecture Notes in Computer Science* 173, pp. 51-67, Springer-Verlag, New York.

Davis, C. and Laender, A. (1999) Multiple Representations in GIS: Materialization Through Map Generalization, Geometric and Spatial Analysis Operations. in: C. B. Medeiros, (Ed.) *7th ACM Symposium on Advances in Geographic Information Systems*, Kansas City, MO, pp. 60-65.

Egenhofer, M. and Frank, A. (1992) Object-Oriented Modeling for GIS. *Journal of the Urban and Regional Information Systems Association* 4(2): 3-19.

Fonseca, F. and Egenhofer, M. (1999) Ontology-Driven Geographic Information Systems. in: C. B. Medeiros, (Ed.) *7th ACM Symposium on Advances in Geographic Information Systems*, Kansas City, MO, pp. 14-19.

Frank, A. (1997) Spatial Ontology: A Geographical Point of View. in: O. Stock, (Ed.) *Spatial and Temporal Reasoning*. pp. 135-153, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Gahegan, M. and Roberts, S. (1988) An Intelligent, Object-Oriented Geographical Information System. *International Journal Geographical Information Systems* Vol. 2: 101-110.

Gosling, J. and McGilton, H. (1995) *The Java Language Environment: a White Paper*. Sun Microsystems, Mountain View, CA, Technical Report.

Graham, I. and Quin, L. (1999) *XML Specification Guide*. John Wiley & Sons, New York.

Gruber, T. (1991) The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. in: *Principles of Knowledge Representation and Reasoning*, Cambridge, MA, pp. 601-602.

Guarino, N. (1997) Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. in: M. Pazienza, (Ed.) *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97*, Frascati, Italy, pp. 139-170.

Guarino, N. (1998) Formal Ontology and Information Systems. in: N. Guarino, (Ed.) *Formal Ontology in Information Systems*. pp. 3-15, IOS Press, Amsterdam, Netherlands.

Karp, P., Chauhdri, V. and Thomere, J. (1999) *XOL: an XML-based Ontology Exchange Language*. Artificial Intelligence Center SRI International, Menlo Park, CA, Technical Report.

Kuhn, W. (1991) Are Displays Maps or Views? in: D. Mark and D. White, (Eds.), *Auto-Carto 10*, Baltimore, pp. 261-274.

Kuhn, W. (1993) Metaphors Create Theories for Users. in: A. Frank and I. Campari, (Eds.), *Spatial Information Theory*. *Lectures Notes in Computer Science* 716, pp. 366-376, Springer-Verlag, Berlin.

Nunes, J. (1991) Geographic Space as a Set of Concrete Geographical Entities. in: D. Mark and A. Frank, (Eds.), *Cognitive and Linguistic Aspects of Geographic Space*. pp. 9-33, Kluwer Academic Publishers, Norwell, MA.

OMG, Ed. (1991) *The Common Object Request Broker: Architecture and Specification, Revision1.1*. OMG Document No. 91.12.1 Framingham, MA.

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

Smith, B. (1995) On Drawing Lines on a Map. in: A. Frank, W. Kuhn, and D. Mark, (Eds.), *Spatial Information Theory - COSIT 95*, Vienna, Austria, pp. 475-484.

Smith, B. (1998) An Introduction to Ontology. in: D. Peuquet, B. Smith, and B. Brogaard, (Eds.), *The Ontology of Fields*. pp. 10-14, NCGIA, Bar Harbor, ME.

Smith, B. and Mark, D. (1998) Ontology and Geographic Kinds. in: *International Symposium on Spatial Data Handling*, Vancouver, Canada, pp. 308-320.

Tempero, E. and Biddle, R. (1998) *Simulating Multiple Inheritance in Java*. Victoria University of Wellington, School of Mathematical and Computing Sciences, Wellington, New Zealand, Technical Report CS-TR-98/1.

Worboys, M. (1994) Object-Oriented Approaches to Geo-Referenced Information. *International Journal of Geographical Information Systems* 8(4): 385-399.