Ontology-Driven Information Integration

Frederico Fonseca **Max Egenhofer**
National Center for Geographic Information and Analysis
Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469-5711
USA

Clodoveu Davis
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Prodabel
Empresa de Informática e Informação
do Município de Belo Horizonte MG
Brazil

Abstract

The integration of information of different kinds, such as spatial and alphanumeric, 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, within 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 systems 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, 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 presents an architecture for an ontology-driven geographic information system (ODGIS) as a solution for geographic information integration. The system uses an object-oriented mapping of ontologies. A strongly typed mapping of classes from multiple ontologies provides a high level of integration. It is necessary to find adequate structures to represent ontologies related to spatial information. This work also shows the potential for extraction of different levels of information, within the framework of ontology-driven geographic information systems. A navigation method inside an ontology-derived class hierarchy is used as a guide to generalization operations. 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.

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). Gruber (1991) suggested that ontologies play a software specification role. 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 & 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).

This paper describes work in development, and the expected results are a working prototype of an ontology-driven geographic information system, a geo-ontology editor, a 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 & Egenhofer 1999). ODGIS are built using software components derived from various ontologies. These software components are classes that can be used to develop new GIS 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 (Egenhofer & Frank 1992; Gahegan & Roberts 1988). 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 next, focusing on the aspects of knowledge generation and use.

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, such as 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 & McGilton 1995). These interfaces need to be complemented with Java code by independent class developers, thus generating Java classes.

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.

Hornsby (1999) points out the difference between resolution and granularity. While resolution refers to the amount of detail and representation, granularity refers to the cognitive aspects involved in selection of features. 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 relate to concepts of a lower level by establishing methods that help to implement rules and constraints.

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, so 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. Guarino (1997) classifies ontologies according to their dependence on a specific task or point of view: *top-level ontologies* describe very general concepts, *domain ontologies* describe the vocabulary related to a generic domain, *task ontologies* describe a task or activity, and *application ontologies* describe concepts depending on both a particular domain and a task, and are usually a specialization of them. Our solution allows the user to incrementally go from coarse to fine-grained ontologies on-line, eliminating thus the division between on-line and off-line ontologies presented in (Guarino 1997).

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.

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 & 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*.

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 & Laender 1999).

The idea of a basic class, such as the *Object* class of ODGIS, is suggested both in object-oriented frameworks such as Java (Gosling & 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 (Pernici 1990). 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 geographic information integration inside a framework of ontology-driven geographic 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 semantic 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 1997) 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 Chrisman, N., ed., *AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography*, 211–220.

Bishr, Y. 1997. *Semantic Aspect of Interoperable GIS*. Ph.D. Dissertation, Wageningen Agricultural University.

Cardelli, L. 1984. A semantics of multiple inheritance. In Kahn, G.; McQueen, D.; and Plotkin, G., eds., *Semantics of Data Types*, volume 173 of *Lecture Notes in Computer Science*. New York: Springer-Verlag. 51–67.

Davis, C., and Laender, A. 1999. Multiple representations in GIS: Materialization through map generalization, geometric and spatial analysis operations. In Medeiros, C. B., ed., *7th ACM Symposium on Advances in Geographic Information Systems*, 60–65. Kansas City, MO: ACM Press, N.Y.

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 Medeiros, C. B., ed., *7th ACM Symposium on Advances in Geographic Information Systems*, 14–19. Kansas City, MO: ACM Press, N.Y.

Frank, A. 1997. Spatial ontology: A geographical point of view. In Stock, O., ed., *Spatial and Temporal Reasoning*. Dordrecht, The Netherlands: Kluwer Academic Publishers. 135–153.

Gahegan, M., and Roberts, S. 1988. An intelligent, object-oriented geographical information system. *International Journal Geographical Information Systems*, Vol. 2:101–110. In Surveying Engineering Library.

Gosling, J., and McGilton, H. 1995. The Java language environment: a white paper. Technical report, Sun Microsystems.

Gruber, T. 1991. The role of common ontology in achieving sharable, reusable knowledge bases. In *Principles of Knowledge Representation and Reasoning*, 601–602. Cambridge, MA: Morgan Kaufmann.

Guarino, N. 1997. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In Pazienza, M., ed., *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97*, volume 1299 of *Lecture Notes in Computer Science*, 139–170.

Guarino, N. 1998. Formal ontology and information systems. In Guarino, N., ed., *Formal Ontology in Information Systems*. Amsterdam, Netherlands: IOS Press. 3–15.

Hornsby, K. 1999. *Identity-Based Reasoning about Spatio-Temporal Change*. Ph.D. Dissertation, University of Maine.

Kuhn, W. 1991. Are displays maps or views? In Mark, D., and White, D., eds., *AUTO-CARTO 10, Tenth International Symposium on Computer-Assisted Cartography*, volume 6, 261–274.

Kuhn, W. 1993. Metaphors create theories for users. In Frank, A., and Campari, I., eds., *Spatial Information Theory*, volume 716 of *Lectures Notes in Computer Science*. Berlin: Springer-Verlag. 366–376.

Nunes, J. 1991. Geographic space as a set of concrete geographical entities. In Mark, D., and Frank, A., eds., *Cognitive and Linguistic Aspects of Geographic Space*. Norwell, MA: Kluwer Academic Publishers. 9–33.

OMG., ed. 1991. *The Common Object Request Broker: Architecture and Specification, Revision1.1*. OMG Document No. 91.12.1.

Pernici, B. 1990. Objects with roles. In *IEEE/ACM Conference on Office Information Systems*, 205–215.

Sheth, A. 1999. Changing focus on interoperability in information systems. In Goodchild, M.; Egenhofer, M.; Fegeas, R.; and Kottman, C., eds., *Interoperating Geographic Information Systems*. Norwell, MA: Kluwer Academic Publishers. 165–180.

Smith, B., and Mark, D. 1998. Ontology and geographic kinds. In *International Symposium on Spatial Data Handling*, 308–320.

Tempero, E., and Biddle, R. 1998. Simulating multiple inheritance in Java. Technical Report CS-TR-98/1, Victoria University of Wellington, School of Mathematical and Computing Sciences.

Worboys, M. 1994. Object-oriented approaches to geo-referenced information. *International Journal of Geographical Information Systems* 8(4):385–399.