SEMANTIC MODELING FOR GEOGRAPHIC INFORMATION SYSTEMS

Christine Parent
HEC ISI, University of Lausanne, http://lbd.epfl.ch/~cparent/christine.html

Stefano Spaccapietra
Ecole Polytechnique Fédérale de Lausanne, http://lbd.epfl.ch/~stefano/e-index.html

Esteban Zimányi
CoDE, Université Libre de Bruxelles, ezimanyi@ulb.ac.be

SYNONYMS

Conceptual Modeling, Geographical Databases, Spatiotemporal Databases, GIS

DEFINITION

Semantic modeling denotes the activity of designing and describing the structure of a data set using a *semantic data model*. Semantic data models (also known as *conceptual data models*) are data models whose aim is to provide designers with modeling constructs and rules that are well suited for representing the user's perception of data in the application world, abstracting from implementation concerns. They contrast with *logical* and *physical data models*, whose aim is to organize data in a way that is easily manageable by a computer. Most popular semantic data models are UML, a de facto standard, and ER (Entity-Relationship), still widely used in many design methodologies and favored by the academic community.

Semantic models were first created in the database community in the 1980s. They started to be developed for Geographical Information Systems (GIS) in the 1990s. Their aim is the same as for traditional databases, to free GIS users from the specificities of system-oriented data models and proprietary file formats (e.g., spaghetti and topological data models, triangulated irregular network (TIN) models, shape files, and raster models). While a number of semantic data models for geographic data have been developed, rapidly the focus has shifted from supporting spatial data to supporting data with both spatial and temporal features, leading to the development of several spatiotemporal semantic data models. Despite the fact that current GIS and Database Management Systems (DBMS) provide poor support for temporal features, semantic modeling advocates that space and time aspects are intrinsically correlated in the application world.

HISTORICAL BACKGROUND

Most people consider the 1976 paper by Peter Chen [5], defining the basic ideas of Entity-Relationship modeling, as the foundational milestone for semantic modeling. The paper had indeed an enormous effect on the database design community, leading to considerable developments to further extend the semantic capabilities of the approach. It took more than 15 years to see the same idea spreading in the academic GIS community with, for example, the 1993 MODUL-R formalism [4], which extended with spatial data the ER approach used in the leading French design methodology, Merise. Further work on MODUL-R eventually resulted in the Perceptory UML-based approach and tool [3]. Semantic models for GIS bloomed in the 1990s, basically splitting into approaches stemming from the object-oriented paradigm (e.g., [16], [7]) and approaches following the ER or the UML paradigm (e.g., MADS [11], STER [15], GeoUML [2]). A survey of many spatial data models may be found in [13]. The industrial and application world has also developed GIS data modeling specifications to help promoting interoperability between different systems and different applications. The Open Geospatial Consortium (OGC) and the International Standards Organization (ISO) have produced specifications supporting conceptual modeling for data with spatial (and some temporal) features (http://www.opengeospatial.org/).

Thanks to the development of ubiquitous and mobile computing on the one hand, and of sensors and GPS technologies on the other hand, large-scale capture of the evolving position of mobile objects has become technically and economically feasible. This opened new perspectives for a large number of applications (e.g., from transportation and logistics to ecology and anthropology) built on the knowledge of objects' movements. Typical examples of moving objects include cars, persons, and planes equipped with a GPS device, animals bearing a transmitter whose signals are captured by satellites, and parcels tagged with RFIDs. This fostered the interest in spatiotemporal models, rather than purely spatial or purely temporal models at the logical and semantic levels. Güting's approach [9] defined a set of data types and associated operators for moving objects (points and surfaces), which allows one to record, for example, the changing geometry of pollution clouds and flooding waters. At the semantic level, examples of spatiotemporal models include MADS [11], Perceptory [3], STUML [15], STER [15], and ST USM [10]. Extending the limited capabilities of commercial data management systems, some research prototype systems [1, 12] do provide nowadays support for storing and querying the position of a moving object all along the lifespan of the object. The latest developments in this domain are the management of trajectories, which adds a semantic interpretation to the movement of objects of kind moving point [6]. Trajectory management is important in many application domains, e.g., for addressing traffic management issues, building social models of people's movements within a city, and optimizing the localization of resources (e.g., communication antennas, shops, advertisement panels) that have to be available to moving customers.

SCIENTIFIC FUNDAMENTALS

Requirements for Semantic Modeling of Spatial Data

Semantic modeling of spatial data requires concepts for the description of both the discrete and the continuous view of space, in a seamlessly integrated way. The *discrete view* (or *object-based view*) is the one that sees space as filled by objects with a defined location and shape. Parts of space where no object is located are considered as empty. This view typically serves application requests asking where certain objects are located, or which objects are located in a given surface. On the other hand, the *continuous view* (or *field-based view*) is the one that sees space as a continuum, holding properties whose values depend on the location in space but not on any specific object (i.e., the value for the property is given by a function whose domain is a spatial extent). Typical examples where this view applies are the recording of continuous phenomena such as temperature, altitude, soil coverage, etc. Both views are important for applications, which may use one or the other, or both simultaneously.

Assuming the discrete view, any traditional database schema can be enriched to become a spatiotemporal database schema by including the description of the spatial and/or temporal properties of the real-world phenomena represented in the schema. Consider, for instance, a Building object type, with properties name, address, usage, architect, and owner. Adding positional information on the geographic location of the building (e.g., its coordinates in some spatial reference system) turns Building into a spatial object type. If we add information characterizing the existence of the building in time (e.g., when construction was first decided, when construction started, when it was completed, when it was abandoned, and when it was demolished), Building becomes a temporal object type. Space and time are independent dimensions. Some data may have spatial features, some may have temporal features, some may have both, and some may have none.

Objects, be they spatial or not, can have spatial properties, i.e., properties whose value domain is composed of spatial values rather than alphanumeric values. Spatial values conform to spatial data types (see the entry in this encyclopedia), e.g., point, line, polyline, surface. For example, a Building object type can have a property nearestFireStation whose value for each building is the geographic location of the nearest fire station, e.g., a spatial value composed of two spatial coordinates defining a point.

Most basic types for space are Point, Line, and Surface (and Volume for 3D databases). However, applications may require more than simple spatial data types. Some spatial objects have extents (the term "extent" denotes the set of points that an object occupies in space) that are made up of a set of elementary extents. For example, an archipelago is a set of surfaces; many coastal countries do have islands too; and facility networks may be represented by connected sets of lines. Moreover, some spatial objects have complex extents made up of a heterogeneous set of spatial values. For example, an avalanche zone is described by a surface and a set of oriented lines describing, respectively, its maximal extent and the usual avalanche paths. Similarly, a river may be described by lines when its bed is narrow and by surfaces when it is broad. Therefore, the set of spatial data types should include types for homogeneous or heterogeneous collections, like PointSet, LineSet, SurfaceSet, or SpatialHeterogeneousSet. The whole set of spatial data types is organized into a generalization hierarchy with generic data types, in order to support spatial object types whose extent may be of different types depending on the instance. For example, the object type City may contain larges cities represented by a surface and small ones represented by a point. The spatial extent of City could then be described by a generic spatial data type that would contain points and surfaces. The Open Geospatial Consortium (OGC) has defined such a hierarchy of spatial data types.

Geographical applications often need to enforce spatial or temporal constraints between spatial or temporal features. For example, harbors should be located along water bodies and bridges on roads or railways. Therefore, a spatial data model should support constructs allowing designers to specify constraints that will be automatically enforced by the system. A first kind of construct is the spatial (or temporal) relationship type. They link two spatial (and/or temporal) object types and bear a spatial (and/or temporal) condition that the linked objects must obey. Typically, conditions express topological relationships (e.g., inclusion, disjointedness, overlapping), metric relationships (e.g., based on distance), orientation relationships (e.g., North of), or the temporal predicates defined by Allen (e.g., during). Applications may need two different kinds of these spatial and temporal relationships:

- Spatially/temporally constraining relationship types: Users can link two spatial objects by a spatially constraining relationship only if their spatial/temporal extents abide by the condition.
- Derived spatial/temporal relationship types: The system automatically creates the instances of the relationship for all couples of objects that satisfy the condition.

Moving and deforming objects may also be linked by spatial relationships. For instance, an aeronautic database may need recording the trajectories of planes when they cross storms, the two being moving objects. In these cases, the condition of the relationship type is spatiotemporal: It bears on the location and the time.

Applications may also need constraints between composite and component elements. For example, a spatial aggregation relationship may enforce that the extent of the composite object is made up by the union of the extents of the component objects, as in a spatial aggregation linking the spatial object types Country and District. Another example is restricting the spatial (or temporal) values of attributes to be within the spatial (or temporal) extent of the object to which they belong. For example, the values of the spatial attribute majorCities (a multivalued attribute of type Point) of the object type Country should be within the spatial extent of the country. This kind of constraint may be frequent, but it is not always the case. Refer for example to the Building spatial object type with the spatial attribute nearestFireStation. Therefore, the data model should not automatically and implicitly enforce these constraints. It should provide designers with a means for explicitly specifying which constraints should be enforced.

The modeling of the continuous view of space requires another construct for properties that are defined on a spatial extent and whose value depends on the exact location (point) of the spatial extent. The spatial extent may be the whole space covered by the database or a specific extent. For example, the water quality of a river exists only in the spatial extent of the river course. On the other hand, temperature, soil, and land coverage are information that exist and may be measured (if relevant) at any point of the geographical space covered by the database. *Field-based* models (see the entry in this encyclopedia) are well suited for applications that perceive the real world exclusively through varying properties. For the many applications that use both the discrete and continuous views, several spatial data models provide a predominant discrete view (i.e., based on spatial objects) in addition to a special construct for representing varying properties, the *space-varying attribute*, which is a function from a spatial extent to a range of values. Any object and relationship, be it spatial or not, should be able to bear space-varying attributes. Moreover, the range of space-varying attributes may be simple (e.g., elevation) or complex (e.g., weather composed of temperature, pressure, and rainfall), monovalued (e.g., altitude) or multivalued (e.g., insects in forests, assuming this information is captured using a space unit large enough to be the home of several kinds of insect, e.g., using cells of 1 sq.m.).

Another important requirement for space modeling is the ability to describe data at different granularity or resolution, for example to be able to support applications working with maps at different scales. How to support this requirement is the topic of the “Multiple Representation Modeling” entry in this encyclopedia.

Finally, an essential requirement is the ability to model spatial features of a phenomenon irrespectively of the fact that the phenomenon has been modeled as an object, a relationship, or an attribute. This orthogonality of the space modeling dimension with the data structure modeling dimension is what avoids making the designs in the two dimensions dependent on each other.

Survey of Current Semantic Modeling Approaches

Semantic models are typically developed in the academic world. For example, MADS [11] has been purposely developed to match all the requirements discussed in the previous section. MADS belongs to the extended ER family of models. Its distinguishing feature is the full support for multiple perceptions and multiple representations of the same real-world objects. Another distinguishing feature of MADS is its support of explicit relationships equipped with topological and synchronization constraints. Multiple perceptions and representations are also supported, to a more limited extent, by Perceptory [3], an UML extension targeted to support spatiotemporal analysis in a data-warehousing framework. STUML [15] and GeoUML [2] are other UML-based approaches, although without multi-perception support. Other spatiotemporal approaches include ST USM [10], very similar to MADS but emphasizing support of multi-granularity, and STER [15], another extended ER formalism which supports both valid and transaction time but, compared to MADS, is weaker in data structures. Most of these academic proposals have been implemented in prototypes, but, with the exception of Perceptory, they have not yet turned in commercial products. These proposals deal with 2D data. For 3D data management, please refer to the corresponding entry in this encyclopedia.

A very different approach, known as spatial constraint database modeling, relies on mathematical equations to define spatial extents. Some existing prototype systems (e.g., DEDALE [8]) use this approach.

KEY APPLICATIONS

Semantic modeling is an essential capability for organizations that need to develop a database that provides different applications and different categories of users with different sets of data, possibly organized in different ways. Designing a database in such a complex environment is a very challenging task, as has been extensively proven in traditional data management. Adding spatial features makes the design task even more complex, in particular since this inevitably leads to adding also temporal features. Indeed, what most applications in the geographical domain need to analyze is the temporal evolution of the spatial features of interest. Cartographic applications are the most traditional ones, but today the focus is rather on all kinds of planning and forecasting services to citizens and the society at the municipal, regional, and statewide levels. Examples of such services include environmental control management and global warming. Given the cost of developing databases for these applications and the need for people in charge (politicians and managers) to be successful, it is of the highest importance that the design of an operational database is carried out using the most suitable tools. Semantic modeling is the key to a successful design that determines what data is needed, to be complemented afterwards in the implementation phase by addressing performance aspects in order to guarantee that the data can be used effectively.

Semantic modeling is also the key to all data exchange, reuse, and integration efforts. Whether in database terms, as discussed here, or in ontological terms, semantic modeling is the kernel of the semantic web.

FUTURE DIRECTIONS

The economic trend towards worldwide enterprise operation and the technical trend towards web-based interoperability will significantly increase the complexity of GIS and the challenges designers will have to overcome. A key help in this context will come from ontologies about spatiotemporal application domains. These ontologies provide a common semantic basis to build repositories of domain knowledge that go beyond traditional enterprise boundaries. In this perspective, the current focus on ontology-assisted semantic modeling and ontology-assisted data integration is leading research into a fruitful direction [14].

In a complementary effort, ontologies and geographic markup languages facilitate the integration of geographical knowledge coming from multiple sources available through the Web. This will contribute to significantly enhance geographical knowledge, benefiting from geo-content actually hidden in Web pages.

CROSS REFERENCES

Data Models, Database Design, Geographic Information System, Spatial Data Types, Topological Relationships, Field-Based Spatial Modeling, Topological Data Models, Multiple Representation Modeling.

RECOMMENDED READING

[1] Almeida, V.T., Güting, R.H., Behr, T.: Querying Moving Objects in SECONDO. In: *Proceedings of the 7th International Conference on Mobile Data Management*, MDM 2006, pp. 47–51, Nara, Japan, May 9–13 (2006)

[2] Belussi, A., Negri, M., Pelagatti, G.: GeoUML: a Geographic Conceptual Model defined through Specialization of ISO TC211 standards. In: *Proceedings of the 10th EC GI & GIS Workshop*, ESDI State of the Art, Warsaw, Poland, June 23–25 (2004)

[3] Brodeur, J., Bédard, Y., Proulx, M.J.: Modelling Geospatial Application Database using UML-based Repositories Aligned with International Standards in Geomatics. In: Li, K.-J., Makki, K., Pissinou, N., Ravada, S. (eds.) *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems*, ACM GIS 2000, pp. 39–46, Washington DC, USA, November 10–11 (2000)

[4] Caron, C., Bédard, Y.: Extending the individual formalism for a more complete modeling of urban spatially referenced data. *Computers, environment and urban systems*, 17(4): 337–346 (1993)

[5] Chen, P.P.: The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1): 9–36 (1976)

[6] Damiani, M.L., Macedo, J., Parent, C., Porto, F., Spaccapietra, S., Vangenot, C.: A Conceptual View on Trajectories. *Data and Knowledge Engineering* (submitted for publication)

[7] Egenhofer, M.J., Frank, A.U.: Object-oriented modeling for GIS. *Journal of the Urban and Regional Information Systems Association*, 4(2): 3–19 (1992)

[8] Grumbach, S., Rigaux, P., Scholl, M., Segoufin, L.: The DEDALE Prototype. In: Kuper, G.M., Libkin, L., Paredaens, J. (eds.), *Constraint Databases*, pp. 365–382. Springer (2000)

[9] Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and quering moving objects. *ACM Transactions on Database System*, 25(1): 1–42 (2000)

[10] Khatri, V., Ram, S., Snodgrass, R.T.: On augmenting database design-support environments to capture the geo-spatio-temporal data semantics. *Information Systems*, 31(2): 98–133 (2006)

[11] Parent, C., Spaccapietra, S., Zimányi, E.: *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Springer, Berlin Heidelberg (2006)

[12] Pelekis, N., Theodoridis, Y., Vosinakis, S., Panayiotopoulos, T.: Hermes – A Framework for Location-Based Data Management. In: Ioannidis, Y.E. *et al.* (eds.), *Proceedings of the 10th*

International Conference on Extending Database Technology, EDBT 2006, pp. 1130–1134, Munich, Germany, March 26–31, Lecture Notes in Computer Science, vol. 3896, Springer, Berlin Heidelberg (2006)

[13] Rios Viqueira, J.R., Lorentzos, N.A., Brisaboa, N.R.: Survey on Spatial Data Modelling Approaches. In: Manolopoulos, Y., Papadopoulos, A., Vassilakopoulos, M. (eds.), *Spatial Databases: Technologies, Techniques and Trends*, pp. 1–22, Idea Group Inc., Hershey, Pennsylvania, USA (2005)

[14] Sugumaran, V., Storey, V.C.: The role of domain ontologies in database design: An ontology management and conceptual modeling environment. *ACM Transactions on Database Systems*, 31(3): 1064–1094 (2006)

[15] Tryfona, N., Price, R., Jensen, C.S.: Spatiotemporal Conceptual Modeling. In: Koubarakis, M. *et al.* (eds.): *Spatiotemporal Databases: The Chorochronos Approach*, chapter 3, pp. 79-116. Lecture Notes in Computer Science, vol. 2520, Springer, Berlin Heidelberg (2003)

[16] Worboys, M., Hearnshaw, H., Maguire, D.: Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, 4(4): 369–383 (1990)