

Issues of GIS data management

Tomáš Richta

Department of Computer Science and Engineering
Faculty of Electrical Engineering, Czech Technical University in Prague
E-mail: richtt1@fel.cvut.cz

Keywords: GIS, CAD, object-orientation, data modelling, data management, database

Abstract

The paper deals with current issues of spatial data modelling and management used by spatial management applications. As a case study for explaining the problem, we use comparison of two main groups of software tools covering this area GIS and CAD systems - and the possibilities of their integration. Studying its functionality, we have found two main problematic issues. The first of them is the density distribution characteristics of stored data according to described area. CAD systems are oriented towards modeling individual man-made objects and structures with relatively high level of detail, so the data stored covers small areas with huge amount of information. On the other side GIS applications maintain large-scale models of real world with significantly lower amount of detail. Here the density distribution of data coverage is better balanced. So the combination of described different densities is the first problem. The second watched issue is the way of storing spatial data. While CAD data are usually stored in individual files (like DXF, IGES), GIS data tend to be stored in files or relational databases. The question we see is, if it is possible to store CAD data along with GIS data in the same database in spite of different distribution densities and different data models. Our paper describes ways of solving this problem.

Motivation

At the beginning of our work we started thinking about an information system capturing real world with the highest achievable level of detail. That means system with 1:1 model of real world entities. Such a system must be able to describe visible and invisible properties of captured objects in the manner to be easily achievable and logically compounded both in computer memory and data warehouse. Mainly we want to describe the data management background needed for such a system.

Two main categories of systems that partly solves our problem are GIS and CAD applications. The aim of both is to model the real world, but each one does this by its own way. GIS applications are constructed for maintaining information in connection with its geographical location. With regard to world scaling factor, GIS data is relatively well distributed in the space. CAD data on the other side relate to only very small areas consuming megabytes of data. Data density distribution in both systems is different. So the question is, which data structures and data management approaches use to obtain global-oriented information system like a GIS with local-detailed information description like a CAD systems.

Previous work

We tried to investigate whether someone is solving the problem of such a spatial management system design. Few relevant papers concerning our topic could be thematically separated into these areas of interest: CAD/GIS integration, 3D GIS data modelling, an object-oriented approach and 3D GIS data management. In following chapters we try to explain the problems mentioned above introducing the ideas of some experts followed by our comments.

GIS/CAD integration

Observing the situation in the area of CAD and GIS integration, we found it almost untouched. That could be partly caused by the complexity of solving such a problem, partly by the lack of interest on the side of GIS and CAD vendors. Both of them have very expensive and broadly used systems, that ensure them good living. Usually it is exactly because the data formats used in CAD are so different and hard to transform, that using one system is the only choice. So all interoperability tendencies are against the profit of vendors, thus unwelcomed. Maybe some development is done behind the curtain, which could be learned from recent Google activities in the geospatial area. But we can only speculate about that. Scientific papers are only describing many differences between those two worlds, which of course are doubtless.

GIS aspects [2,3]

- landscape-level analysis and mapping
- advanced information tools
- mostly 2D modelling
- database based
- optimized for data retrieval
- 1:5000 scale and below
- constrained editing environment

CAD aspects [2,3]

- object-level design and drafting
- advanced drawing tools
- 3D modelling
- file based
- optimized for data design
- 1:40-5000 scale
- unconstrained editing environment

CAD systems deal with large-scale models without maintenance of attributes and geographical coordinate systems. GIS are on the other side are able to manage small-scale models with

attributes and a variety of different geographic coordinate systems. CAD and GIS share one major characteristic both deal with geometry [1].

CADs usually store data in file format, GISs more often maintain data permanently in databases. CAD systems generally assume an orthogonal world, while GIS systems deal with data sources based on the model of spherical world. Also, all these different pieces of information are created and maintained by totally different organizations with different tools and different utilization goals [2].

As could be seen the integration process probably leads to use the CAD for data capture, design and modelling, while GIS for data management, analysis and visualizing. Solution for an integrated CAD/GIS framework must start with the design of final data warehouse. The solution could be found in mapping both data into the neutral data model [2, 5]. So the first issue is to develop a proper data model that could then be integrated into designed spatial management system. Now let us know the stage where the solving of this problem could be found. Because of the common feature of CAD and GIS that is 3D representation, we will show only data models that deal with 3D representation of spatial data.

3D GIS data model

Among the important 3D data models for GIS applications belongs: Molenaar's FDS, Wang and Gruen's V3D [14], Zlatanova's SSS [13], Pilouk's TEN, Shi et al.s OO3D [16], Coors's UDM [15] and Balovnev et als GeoToolKit [17]. Here we introduce a brief summary of their features. We don't want to describe them in detail or compare each other, because it has been done previously. We just want to get an idea, how the problem is solved.

All those mentioned models are vector and boundary based, which partition spatial objects as points, lines faces and bodies or similar geometric primitives. First one was the FDS, which partitioned the space into non-overlapping objects and thus tried to ensure 1:1 relationships between the primitives and the spatial objects of same dimension, e.g. surface and face. FDS describes the basic geometric elements: node, edge, arc and face and four spatial objects: point, line, surface and volume. As a first model it was broadly discussed and frequently extended like in V3D. The distinct feature of V3D model is, that the geometric information is combined with attribute information [14]. Then the TEN was introduced which includes node, arc, triangle and tetrahedron as the basic elements. The most important part in constructing a spatial object in the TEN model is to decompose the object into a set of the composed tetrahedrons. The SSS model is a further development of the FDS model. Compared to FDS, the SSS keeps the explicit relationships between body and face and eliminates the edge and arc object. On the other hand, the SSS keeps the relationship of a geometric objects and attribute data such as texture. OO3D uses node, segment and triangle as basic elements [16]. UDM and SSS are quite similar. Both don't support the arc and the edge elements. Because of this restriction only polyhedrons can be represented in the UDM [15].

Particular attention requires the GeoToolKit, because it is not only a data model, but also an object-oriented geo-database kernel system for the support of 3D geological applications, which demonstrates the potential of object-oriented concept in 3D database. GeoToolKit deals primarily with two basic notions: a SpatialObject and a Space (a collection of spatial objects). A spatial object is defined as a point set in the 3D Euclidean space. Object-oriented

method is a key nature of GeoToolKit, which helps to construct the object hierarchy. The Group gathers spatial objects of different types into a collection and then is treated as a single object. GeoToolKit backbone is a class library for the efficient storage and retrieval of spatial objects within an object-oriented database. Currently GeoToolKit offers classes for representation and manipulation of simple (point, segment, triangle, tetrahedron) and complex (curve, surface, solid) 3D spatial objects. The application developed with GeoToolKit simply inherits geometric functionality from GeoToolKit, extending it with application-specific semantics [13, 17, 18]. Surveying 3D data models, the paradigm of object-orientation frequently appeared. Because it seems to be very important for GIS researchers, we have to explain it briefly.

An object-oriented approach

MDittrich's definition says that there are three types of object/orientation: structural object orientation - any entity, independent of whatever complexity and structure, may be represented by exactly one object, no artificial decomposition into simpler parts due to technical restrictions should be necessary, operational object orientation operations on complex objects are possible without having to decompose the objects into a number of simple objects, behavioral object orientation a system must allow its objects to be accessed and modified only through a set of operations specific to an object type [7].

There are four main concepts covering object orientation. The first concept is the encapsulation. It means that the object or some group of objects (class) and the procedures (methods) defined on it are stored and managed together. To activate a procedure the program sends a message to an encapsulated data-procedures set, in the consequence of the procedure's activity the set can send another message to another set, etc. The second concept is the inheritance. The inheritance is related to the class hierarchy. If we have a subdivision of a class, then the subclasses inherit from the data and methods. The third concept is the object identity. It means that despite different transformations the object's identification should not change. There is a fourth concept, the so-called polymorphism. The word polymorphism we can interpret as different responses to the same message depending on the object in the address of the message. For example we can send a message: plot to the address a,b,c. If into the object a is encapsulated a procedure of a circle, in the object b that of an ellipse and in the object c that of a square, then depending on the addresses, the command plot will result a circle, an ellipse or a square [9]. As would be seen the object-oriented approach provide a number of mechanisms to model real objects in a natural way.

3D GIS data management

As a last issue we have to describe the 3D GIS data management. Going through the ideas of many experts, we have to notice, that the discussion about the 3D GIS data management is concerned in comparing the relational and object database management systems. The relational data model is the most common one. It seems to be suitable for modelling commercial data for which humans may have the mental model of tables, such as bank accounts, telephone calls, etc. But it is not proper for modelling data that describe spatial phenomena. Relational database management systems are suitable and successful for applications dealing

with weakly structured data, but they fail when they are used for applications of data with a complex structure. Since the relational data model does not match the natural concepts humans have about spatial data, users must artificially transform their mental models into a restrictive set of non-spatial concepts. The object-oriented data model is built on the four basic concepts of abstraction: classification, generalization, association, and aggregation [7].

Many researchers comply that the spatial information systems will benefit from the use of object-oriented database management systems in various ways: The architecture of a GIS will become clearer such that the maintenance of GIS software will be easier and its life cycle will be longer, programmers should not worry about aspects of the physical implementation of data [7, 8, 10, 11, 4].

Object oriented databases have been portrayed as being the solution for complex applications such as Geographical Information Systems. A further motivation for the use of an object-oriented approach to the production of such a system is therefore the expectation that the approach will result in a system which has a clean interface and is easier to maintain than an equivalent system built using conventional programming techniques [10].

Developers of object-oriented GIS systems comply that a new object-oriented database is to be developed, because there's need to reengineer the database up against performance and others problems that will arise, when such a system will be tested [4].

GIS and CAD problem

Resulting from mentioned papers, we tried to make our own comments for described issues. Each of current GIS application is able to manage and display points, lines and polygons. That's because GIS data are usually stored as point, lines and polygons. Let's call them geometrical primitives. There's no standard among GIS data formats, but the most common file format is ESRI shape file. Each shapefile contains information about locations of one type of geometrical primitives covering one thematical area and is called layer. This file is succeeded with database file, which contains attribute information about geomtrival primitives from shapefile. There is also file with information about connection between the shapefile primitives and database records, index file. GIS applications are constructed to load these files and pile up the layers to produce final map. This approach is sometimes turned into storing all described information into relational database which structure is similar to the structure of shapefiles. The problem here lies in the fragmentation of stored information. Points, lines and polygons representing objects are stored separatelly. And of course used data types are much simpler than modeled real world objects. Definition of new data types is not supported in GIS applications, because it is very problematic to change for example relational data structure.

Almost the same problem we could find in CAD systems, buy here the situation starts to change. Usual CAD data are stored as points, lines and polygons. But some newest applications, especially in machine building industry, are able to let user define his own objects and compose complex structures from the simpler ones. But there's no data model supporting storage of defined objects, so they must be saved as points, lines and polygons.

Conclusion and future work

Now we can summarize the problem of the GIS and CAD integration. Because of the different characteristics of the GIS/CAD worlds, firstly there's need to decide for some suitable 3D data model, which could maintain complex and structured data types. This model also must be able to maintain the large-scale 3D models produced by CAD as well as low-scale objects used by GIS. Object-oriented approach seems to be the proper way, by which this model could be developed, because it offers richer data structures and more intuitive representation of the real world objects. An example of basic data model could be GeoToolKit.

Secondly, there's need to prepare a system for maintaining the 3D data model. This system must be able to exchange data with different data sources including CAD files. We suppose that this system has to be closely connected with an object-oriented database, which will maintain persistent objects.

As a third part of such a system we have to develop application-specific data models for describing the real world objects of particular interest. In example, when we are concerned in modelling the cities, our data model has to describe buildings, streets and other city components. Some of them are usually designed using CAD systems and so it would be a great deal to be able use CAD data for city modelling. Here we introduce an example of data models describing the city problem. It describes the relations between the city parts.

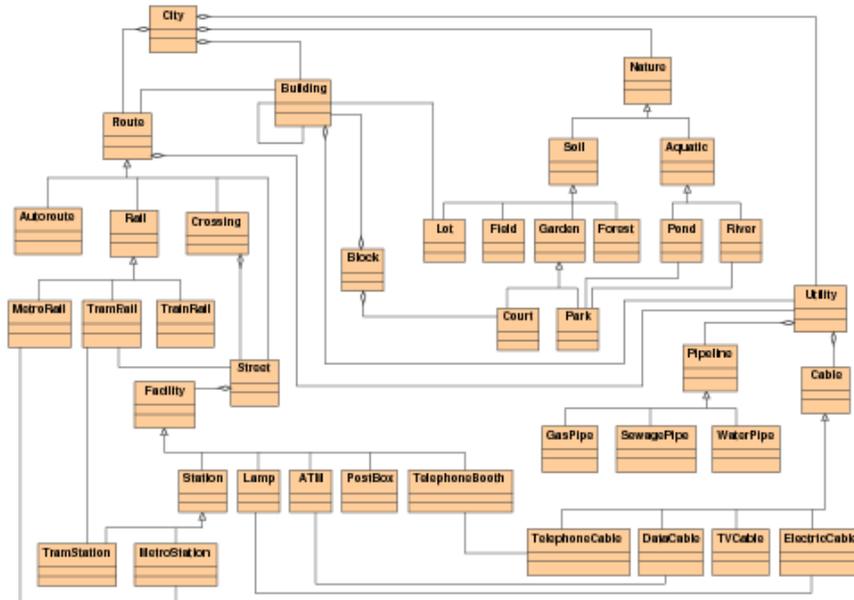


Fig 1.: An example of the city data model

The advantage of using object-oriented approach is that we can interactively modify our model, when it is necessary. Because this model is stored by an object-oriented database, the data structure might be built exactly like you can see them in the model. There's no need

to decompose this model into additional data structures because now we can store the whole object. In addition we can add a behavior to this object defining its methods. This way we can make digitalized abstractions of real world entities.

As the last problem that is to be solved when we will be storing CAD data in the 3D GIS database is how to generalize and classify the high-scaled objects from CAD when storing it into low-scaled objects in GIS. An on the other hand how to restore the details of re-scaled objects stored in GIS and be able to give them back into the CAD to be redesigned.

References

1. Zlatanova S.: Large-scale data integration An Introduction to the Challenges for CAD and GIS Integration, Directions magazine, July 10, 2004.
2. van Oosterom P.: Bridging the Worlds of CAD and GIS, Directions magazine, June 17, 2004.
3. Zlatanova S., Rahman A. A., Pilouk M.: Trends in 3D GIS Development, Journal of Geospatial Engineering, Vol. 4, No. 2, December 2002.
4. Bodum L., Sorensen E. M.: Centre for 3D GeoInformation Towards a New Concept for Handling Geoinformation, FIG Working Week 2003, Paris, April 2003.
5. Weinstein D.: Cross Platform CAD-GIS integration: Automating CAD Workflows and GIS Technologies to Support Structural Inspection and Decision Support Systems on Bostons Central Artery Project, GIS-T 2004, March 29, 2004.
6. Chance A., Newell R.G., Theriault D.G.: Smalworld GIS: An Object-Oriented GIS - Issues and Solutions, 2000, <http://www.logis.ro/downloads/>
7. Egenhofer M. J.: Object-oriented modeling for GIS, Journal of the Urban and Regional Information Systems Association 4 (2): 3-19, 1992.
8. Kofler M.: R-trees for Visualizing and Organizing large 3D GIS Databases, TU Graz, Austria, 1998.
9. Sarkozy F.: GIS FUNCTIONS, PERIODICA POLYTECHNICA SER. CIV. ENG. VOL. 43, NO. 1, PP.87-106, 1999.
10. Garvey M., Jackson M., Roberts M.: An Object-Oriented GIS, Net.ObjectDays 2000.
11. Li J., Jing N., Sun M.: Spatial Database Techniques Oriented to Visualization in 3D GIS, Digital Earth, June 2001.
12. van Oosterom P., Stoter J., Quak W., Zlatanova S.: The Balance Between Geometry and Topology, Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, Springer-Verlag, Berlin, pp. 209-224 , April 2002.
13. Zlatanova S.: 3D GIS for urban development, TU Delft, 2000.
14. Zhou Q., Zhang W.: A Preliminary Review on 3-dimensional City Model, Asia GIS 2003 Conference, October 2003.

15. Coors V.: 3D GIS IN NETWORKING ENVIRONMENTS, Computer, Environments and Urban Systems 27/4, Special Issue 3D cadastre, Elsevier, 2003, ISSN 0198-9715, pp 345-357, April 2003.
16. Shi W., Yang B., Li Q.: An object-oriented data model for complex objects in three-dimensional geographical information systems, Int. J. of Geographical information science, vol.17, no. 5, july-august 2003, 411-430.
17. Balovnev O., Breunig M., Cremers A. B., Shumilov S.: GEOTOOLKIT: OPENING THE ACCESS TO OBJECT-ORIENTED GEO-DATA STORES, Interoperating Geographic Information Systems, Boston: Kluwer Academic Publishers, 1999.
18. Breuning M., Cremers A. B., Seidemann R., Shumilov S., Siehl A.: Integration of GOCAD with an object-oriented geo-database system, Gocad Meeting, Nancy, France, June 1999.
19. Bodum L.: Design of a 3D virtual geographic interface for access to geoinformation in real time, CORP 2004 and Geomultimedia04, February 2004.