

Analysis Patterns for GIS Data Schema Reuse on Urban Management Applications

JUGURTA LISBOA FILHO

Universidade Federal de Viçosa, Departamento de Informática,
Viçosa/MG, Brazil, 36571-000
jugurta@dpi.ufv.br

CIRANO IOCHPE

Universidade Federal do Rio Grande do Sul, Instituto de Informática,
Porto Alegre/RS, Brazil, 91501-970
ciochpe@inf.ufrgs.br

KARLA A.V. BORGES

PRODABEL
Belo Horizonte/MG, Brazil, 31230-000
karla@pbh.gov.br

ABSTRACT

An analysis pattern is any part of a requirement analysis specification that can be reused in the design of other information systems as well. Urban management systems (e.g.: Tax Control Systems, Urban Transportation System) are implemented in a similar way for many counties. This paper proposes three analysis patterns that make possible the reuse of geographic database design for urban area planning and management applications, developed in Geographic Information System (GIS).

Keywords: Analysis Patterns, GIS, Conceptual Model, Reuse.

1. INTRODUCTION

The key points behind large information systems' success are the precise requirement analysis and design through well-known representation forms. Experiences have shown that requirement analysis and database conceptual design stages are complex activities that take a lot of time. According to Johannesson [13], the reasons for this are that the application knowledge domain and the system requirements process are done almost always from the beginning of each new system under development.

Geographic Information System (GIS) applications may have some special requirements (e.g.: geographic referenced data manipulation), but must be developed using the same processes as any other information system. One of the many processes that are receiving special attention is the use of instruments that allow software components reuse through pattern definitions, which is mainly the object oriented systems community.

A pattern is a recurring combination of modeling elements that occurs in some context [7]. Patterns may be used in any of the several software development stages, therefore receiving different names as analysis patterns, design patterns, architecture patterns, languages (implementation patterns), and others.

Reuse of software components is being done informally since the very first software implementation. Design patterns use in Computer Science however is more recent [10]. Pattern use in software development is increasing daily due to reuse and software components quality contribution. Design patterns allow knowledge availability and designers' experience exchange, as well as making communication among project members easier [24].

Another pattern category is analysis pattern, which has been treated as a reuse instrument solution for requirement analysis and database conceptual modeling [7], [9], [12], [13] and [24].

A comparison of the several existing patterns may be found in [3]. According to Fernandez [6], among the reasons that differentiate analysis and design patterns are:

- analysis patterns depend on the application therefore its semantics describe some domain or application specific aspects;
- design patterns are closer to implementation because it focuses mainly in typical design aspects such as man-machine interactions, object creation, basic structural properties;
- design patterns may be applied to a larger number of applications. For an instance most applications have the man-machine interaction.

One particular GIS applications property is that usually the data handled by these applications have a strong relationship among each other, because they describe geographic phenomena about one specific geographic region. For an example many spatial data (e.g.: theme data) are generated from basic data (e.g.: topographic data). The data type set that usually creates the cartographic data for one GIS application has a conceptual structure alike other GIS applications. This particular feature makes GIS applications strong candidates to benefit from reuse of existing database designs [14], as happens to geo-spatial data sharing in digital media [23].

This paper proposes three analysis patterns applicable to requirement analysis and conceptual geographic database modeling stages. Moreover a specific notation for GIS analysis patterns description is shown.

Section 2 shows some analysis patterns theories detailing creation processes and documentations. Section 3 presents three analysis patterns identified in geographical database applications in urban management. Section 4 presents some final considerations and perspectives over future work.

2. ANALYSIS PATTERNS

An analysis pattern is any requirements specification part that begins in a project and may be reused by others [9]. The analysis pattern describes a set of real world objects, its relationships and rules that define its behavior and states in an arbitrary level of abstraction [13].

Analysis patterns have been developed to make possible reuse different solutions in system analysis. Some patterns are less generic and may be reused in different applications inside a certain domain while others may be used in different domains. As specific domain analysis pattern examples we have insurance application patterns [21] and banking application patterns [17]. Among the most generic patterns there are the reservation and renting reuse entities (e.g.: hotel rooms reservation, vehicle renting reservation) [7], the Measure and Observed patterns [9] and the Contract and Documents patterns [12].

Johannesson [13] identifies two quality requirements that a pattern must have. The first one it must be generic enough, i.e. an analysis pattern must be applicable in different systems. The second is that it must be easier to understand and easier to

model a new application from the beginning. Certainly these two requirements have conflicts once highly generic patterns may be hard to fit into new projects. That is a great challenge to patterns documenting designers, which is the task to find the balance between describing an abstract pattern in such a way it can be reused, but also that they can be easily understandable to future users.

Although reuse approach has designer productivity increase as one of the main advantages, pattern use requires some additional work, for an example patterns description and availability. For this, some stages must be fulfilled. The first stage is to choose a language for the analysis pattern description. After that there is the proper pattern documentation and finally the effort to make this pattern's reuse desirable and effective. The following subsections describe these aspects in a higher detail.

Recognizing new analysis patterns

The process of discovering or capture a new analysis pattern requires enough analyst's skills and experience to realize existing analogies inside design parts of a studied system, that may be happening in other systems. According to Fernandez [8] the patterns creating process is made through identifying design parts that are possible reuse candidates. After that this sub-diagram must be generalized. Its specifics must be eliminated and the pattern must be documented according a chosen pattern description language.

In turn, patterns use requires an analogy feeling between a current problem and the available patterns with the following identified pattern adaptation to the new project. Figure 1 shows the creation process and the analysis patterns use.

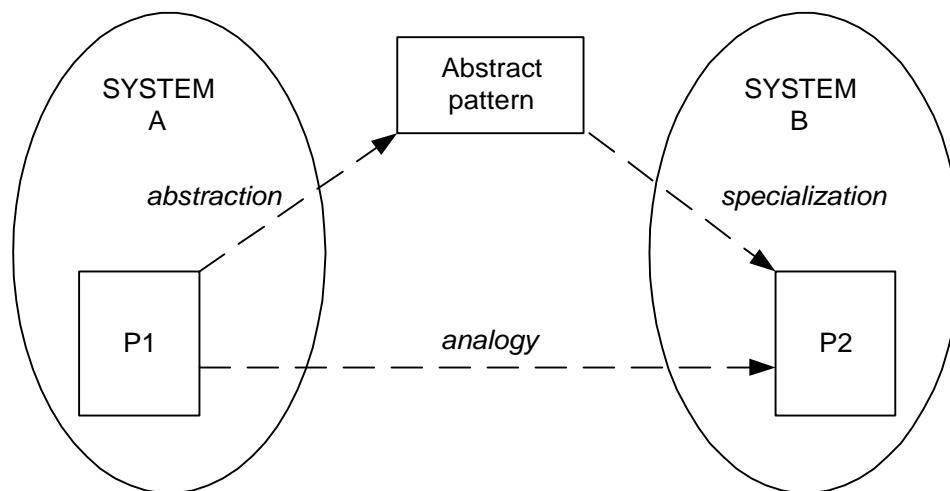


FIGURE 1 – Analysis patterns capture process [8]

The analyst's most important skill to identify new analysis patterns is abstraction ability. The process of discovering new reuse patterns is a much harder task than its use. When defining new patterns details specific to a single system or organization details for which the system was developed, must be eliminated from pattern description. A pattern must include only generic enough elements so that it can be reused. During analysis stages new patterns candidates must be specified and made available.

Rawsthorne [20] presents a pattern language for requirements analysis in an object oriented systems project context. A new proposed pattern (Pattern Scavenging) offers directives for an analyst that looks for new pattern candidates. That is made through examining existing analytical models, searching for structure and behavior

similarities, which is made from similar “shapes” fetching in existing diagrams. Silva et al [22] propose a method of analysis patterns identification applying Knowledge Discovery in Databases (KDD).

Specifying analysis patterns

Initially many analysis patterns were described in a free form narrative (e.g.: patterns described in [9] and [12]), i.e. they did not follow a standard structure such as happens to design patterns [10]. Recently analysis patterns are described in a similar way to design patterns, which use an established topic structure (template). In this paper the opted structure was that of Meszaros [18], in which an analysis pattern description must have at least the following items: Problem-Context-Strengths-Solution. However when desired, patterns also include use example.

- *Problem* – supplies a brief declaration of the problem that needs to be solved;
- *Context* – describes the context in which the problem was identified and for which the solution was presented;
- *Forces* – set of restrictions that were considered when solving the problem;
- *Solution* – supplies a classes diagram with the proposed solution.

To document the presented solution by analysis patterns in geographic database design it is necessary to create the conceptual model, what forces to choose an adequate conceptual data model. To attend to the necessary GIS application's modeling requirements many conceptual models have been proposed. Examples include Perceptory, OMT-G, and MADS. A detailed survey of geographic database conceptual models is shown on [16].

A common feature in these models is that they are actually extensions of an already consecrated conceptual model (e.g.: E-R, OMT). These extensions introduce some new abstract constructors that make possible related space-time aspects for capturing and representing its geographic information semantics. A second common feature in these models is that if they offer many different new semantic constructors making them high expression languages, on the other hand the excessive number of new semantic constructors reduces resulting diagrams legibility and increases understanding difficulty, especially for GIS users that normally design their own systems.

To describe an analysis pattern the requirements that a conceptual data model should have include among others: (1) conceptual model must be from the designers' domain; (2) it has to be easy to understand to unlearned users; (3) it should have a high expression capability; (4) it must be extensible so it can do specifics imposed by features on many application domains.

In [15] is shown that Unified Modeling Language (UML) [1] does these requirements. Thus it is considered adopt the classes' diagram graphical notation of the UML language (Figure 2) to present an analysis pattern solution. As modeling geographic information spatial aspects may be done by the Geoframe framework [15] with its stereotypes set (Figure 3).

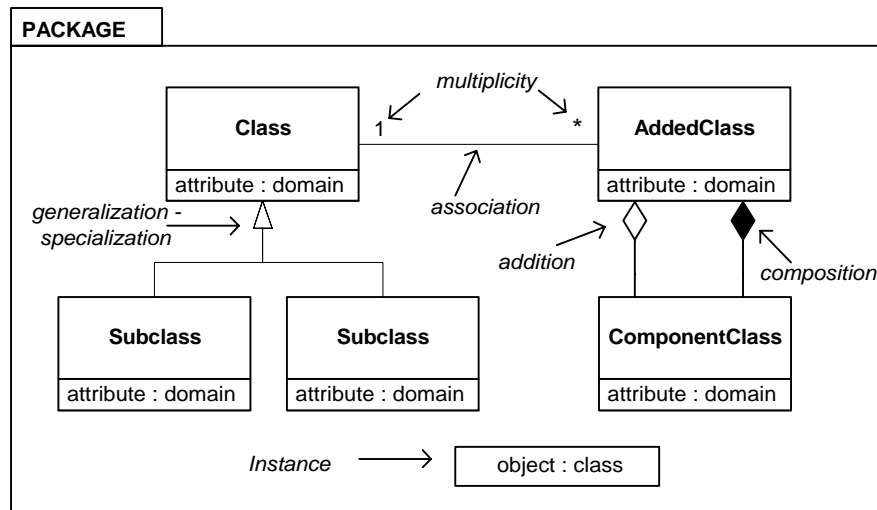


FIGURE 2 – UML classes' diagram notation (resumed)

In Figure 3 the first stereotype set (*Geographic phenomenon* and *Conventional object*) is used to differentiate two main object types that belong to a geographic database. *Geographic phenomenon* is specialized in *Geographic object* (\triangle) and *Geographic field* (\triangle) as the two ways of understanding geographic phenomenon described by Goodchild [11]. Non-geographic objects or conventional objects, i.e. those not geographically referenced are modeled in a traditional way identified by the stereotype (\triangle).

The second and third stereotypes sets (*Geographic object's spatial component* and *Geographic field's spatial component*) are used to model the phenomena spatial component as the object and field views respectively. Multiple representations occurrence is specified matching two or more stereotypes. For an example a *City* class may have two spatial component abstraction forms, points and/or polygons, which is specified by its stereotypes' pair ($\square \square$).

Finally the stereotype `<<function>>` is used to describe a special association type that occurs when modeling categorical functions. According to Chrisman [4] in a covered categorical structure space is classified in mutual exclusive categories, i.e. a variable may have a category type in all points inside a region (e.g.: soil types).

<i>Geographic phenomenon and Conventional object</i>	<i>Geographic object's spatial component</i>	<i>Geographic field's spatial component</i>
\triangle Geographic object	$\square \bullet$ Point	$\square \bullet \bullet$ Irregular points
\triangle Geographic field	$\square \text{---}$ Line	$\square \bullet \bullet \bullet$ Grid of points
\triangle Non-geographic object	$\square \square$ Polygon	$\square \text{---}$ Adjacent polygons
	$\square *$ Complex spatial obj.	$\square \text{---}$ Isolines
<code><<function>></code> <i>categorical function</i>		$\square \text{---}$ Grid of cells
		$\square \text{---}$ TIN

FIGURE 3 – Geoframe stereotypes

Although stereotypes set in Figure 3 makes possible modeling geographical phenomena perceived in the field vision an urban applications' feature is that most

identified phenomena fits only in object vision as will be seen in analysis patterns' examples presented in the next section.

As an UML-GeoFrame approach use example, Figure 4 illustrates a conceptual model extract using the UML-GeoFrame notation where is seen many themes (e.g.: COAL_ACTIV, SOIL_USE), modeled as packages. Each theme congregates linked classes that may be subclasses of conventional object [△] (e.g.: CoalCompany and SoilUseType) of geographic phenomena perceived in object views [△] (e.g.: City, Deposit) or geographic phenomena perceived in field view [△] (e.g.: CobSoilUse, Topography). Multiple representation examples are shown in CoalMine [□▪□], HydroResources [□▪□] and Topography [□▪□▪□▪].

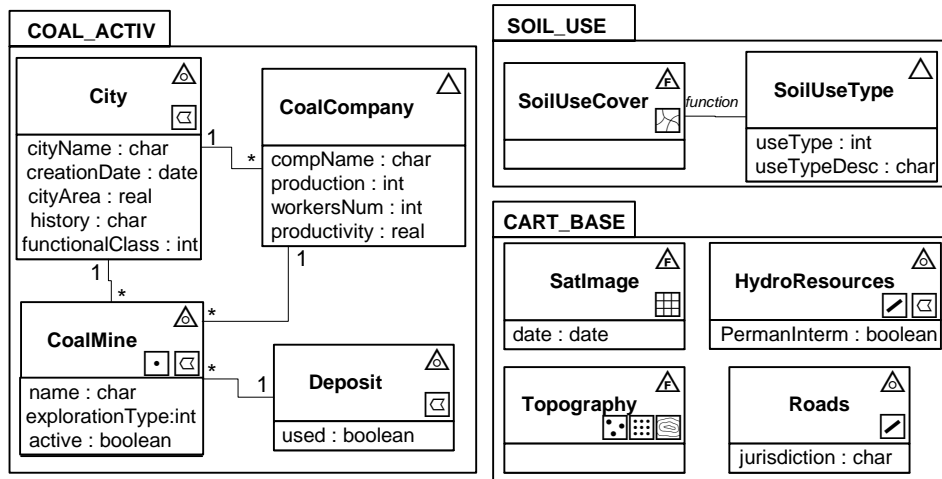


FIGURE 4 – An UML-GeoFrame data model example

3. ANALYSIS PATTERN IN URBAN APPLICATIONS

GIS are used in many different areas such as Natural Environment, Telecommunications, Business and Marketing, Fleets Monitoring, Public Administration, and others. On each of these application areas it is indispensable to create a specific analysis model for the given problem so that the observed objects can be related with a determined geographical region.

The concerned area within the urban management is a city, which is shaped by its own natural and built environment. It has road routes, constructions, open areas, climate, vegetation, population, and others.

A city is a mutant, dynamic, living body where deep differences exist and need to be managed on behalf of its population's quality of life [2]. Traditional representation systems, as the maps, are statics even if made by a computer (CAD systems) because they represent the situation in the time they were produced. A GIS can generate dynamic maps keeping the reality evolution record from data collected in administrative tasks. To do so, management needs to see the city as a whole, putting aside its different visions and actions over the city. It is unique and sensitive to time changes [2].

Public management has grown interest into GIS use due to the needs of integrated city management and to increase urban quality of life [2]. However, the first challenge is getting human resources with technical skills to design, implement and maintain systems using GIS technology. This problem is even more serious with medium and small cities.

By authors' experience acquired in development of urban management applications the first observed feature is its great potential to reuse previous solutions, which may be either from different agencies inside the same administration, or even from different administrations. Analysis patterns provide a considerable potential to reduce these difficulties since it:

- makes possible for a less experienced designer to reuse knowledge that has been tested and validated previously;
- in urban management the basic environment that compose the digital cartographic base (e.g.: streets, blocks, open areas and neighborhoods) may be reused by several applications.

Three analysis patterns are proposed in the following sections. They were identified from several database conceptual model analyses in urban applications. Only essential operations and attributes in the given examples are shown to resume this work.

3.1 Urban Street Mesh Pattern

Problem:

Which elements belong to a city's street mesh?

Context:

Almost every city in Brazil has shown the same organization pattern, which is structured by their pathways organization (e.g.: streets, avenues, drives). The set of pathways stretches generates an urban street network.

Forces:

- Each drive way stretch is considered a road instance and should have an identification code and a name. It normally should be divided into several segments as well.
- A road stretch is a pathway segment between two connections.
- The set formed by the connections (or terminal points) and road stretches create an urban street mesh.

Solution:

Figure 5 shows the belonging classes diagram to the pattern.

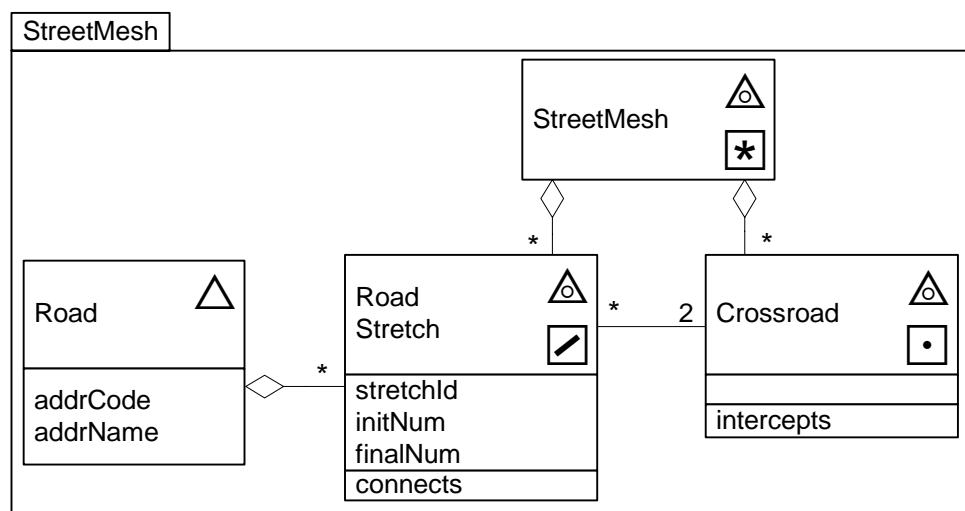




FIGURE 5 – “Urban Street Mesh” pattern classes’ diagram.

Each geographic phenomenon's pattern specifies the most general properties (attributes and operations), which must be extended and specialized for their specific application. Consequently their spatial abstraction components are specified. For an instance, the class RoadStretch spatial component is specified as linear .

Participants:

The StreetMesh class is a geographic phenomenon represented by a complex spatial object (represented by the  symbol). In this class many attributes may be defined relating to the network as a whole. Road is a conventional class implemented normally as a table in a relational DBMS. Each road is made of several road stretches, which corresponds to a network arc. A road stretch may be connected to other stretches but this connection is represented by the Crossroad class' instances, which are the network nodes. The network elements' manipulation operations may be implemented as classes' methods from StreetMesh, RoadStretch and Crossroad depending on their functionality.

Related Patterns:

The *Urban Street Mesh* uses the "State Across a Collection" pattern [5] when modeling the Road and Road Stretch phenomena. Moreover a new pattern project may be abstracted to create any network structure model made by nodes and arcs, whose topology's relationship among its elements is kept to make possible common network operations such as the shortest path calculation (needs a weight for each arc), network navigation, distance between nodes, etc.

3.2 Urban Street Traffic Network Pattern

Problem:

How to model urban street traffic network elements?

Context:

Vehicle traffic in a city is done over an urban street mesh. The street traffic network offers traffic direction and the street mesh offers the street structure. Some pathways have one direction and other have two directions. Each stretch is classified according their street system's importance. For an instance if it is a collecting road, a regional link or a local street (many applications use this kind of information).

Forces:

- Each route stretch has information about the vehicle allowed traffic. And may comprise many road stretches.
- Some stretches do not allow vehicle traffic (e.g.: pedestrian walkways).

Solution:

Figure 6 illustrates the pattern classes' diagram. The street traffic network overlaps the street mesh network so that the *Urban Street Traffic Network* extends the *Urban Street Mesh* pattern described in the last section.

Participants:

The StreetTrafficNetwork class has a complex spatial representation formed by traffic stretches and conversion nodes representation. Operations involving the entire network are defined as this class' method. Each traffic stretch may be associated with many road stretches meaning a more compact network overlapping another more complete. On the other hand not every road stretch is part of a traffic stretch as happens to pedestrian walkways. In the same way there are crosses of road stretches

that may not necessarily be a conversion node. According to the application, a traffic stretch may be specialized on many ways. In traffic systems context is common to have the traffic stretches classification by the street size (e.g.: arterial roads, collecting roads, regional links, local streets).

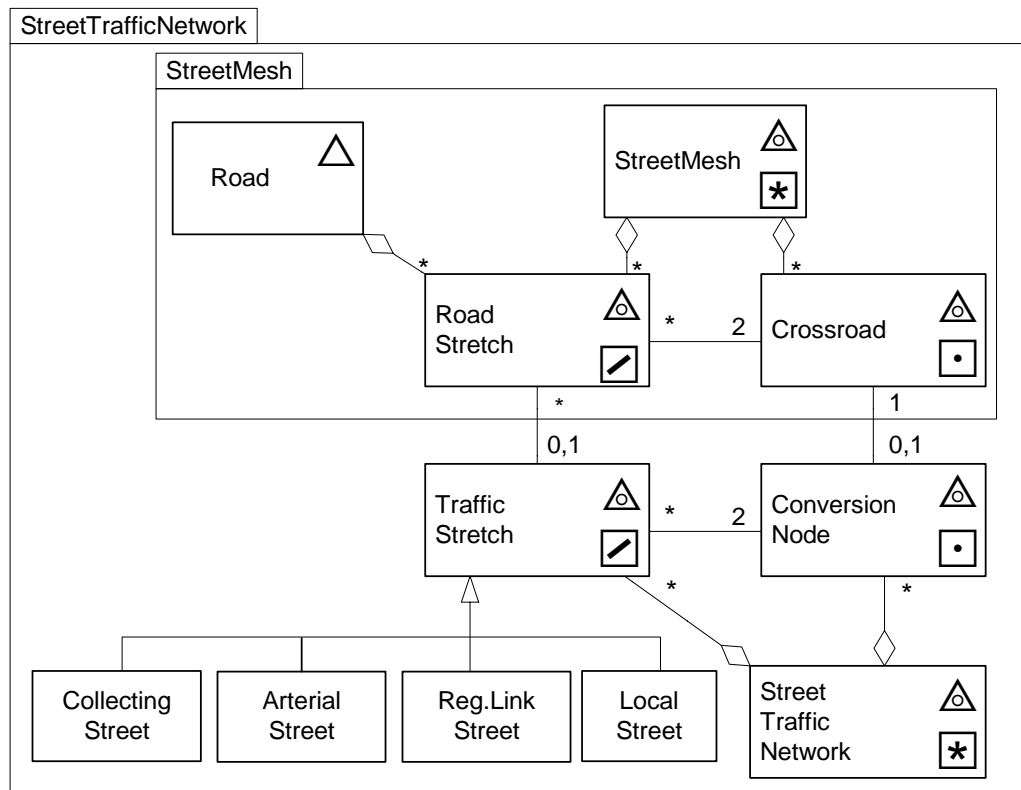


FIGURE 6 – “Urban Street Traffic Network” classes’ diagram.

Related patterns:

The *Urban Street Traffic Network* is based on the *Urban Street Mesh* pattern.

Example:

A bus route system uses the street traffic network that in turn is on the street network, which even so may be used to other purposes. Figure 7 extracted from Belo Horizonte’s urban transportation system’s database conceptual schema, illustrates the *Urban Street Traffic Mesh* pattern use.

Observe the modeling reuse of the themes *Street Mesh* and *Street Traffic Network*. The designer needs to model only a system’s part related to the application, i.e. the urban bus transportation system. Emergency attendance vehicle routing systems (e.g.: ambulances, police, firemen) are other examples to this analysis pattern as well as products delivery.

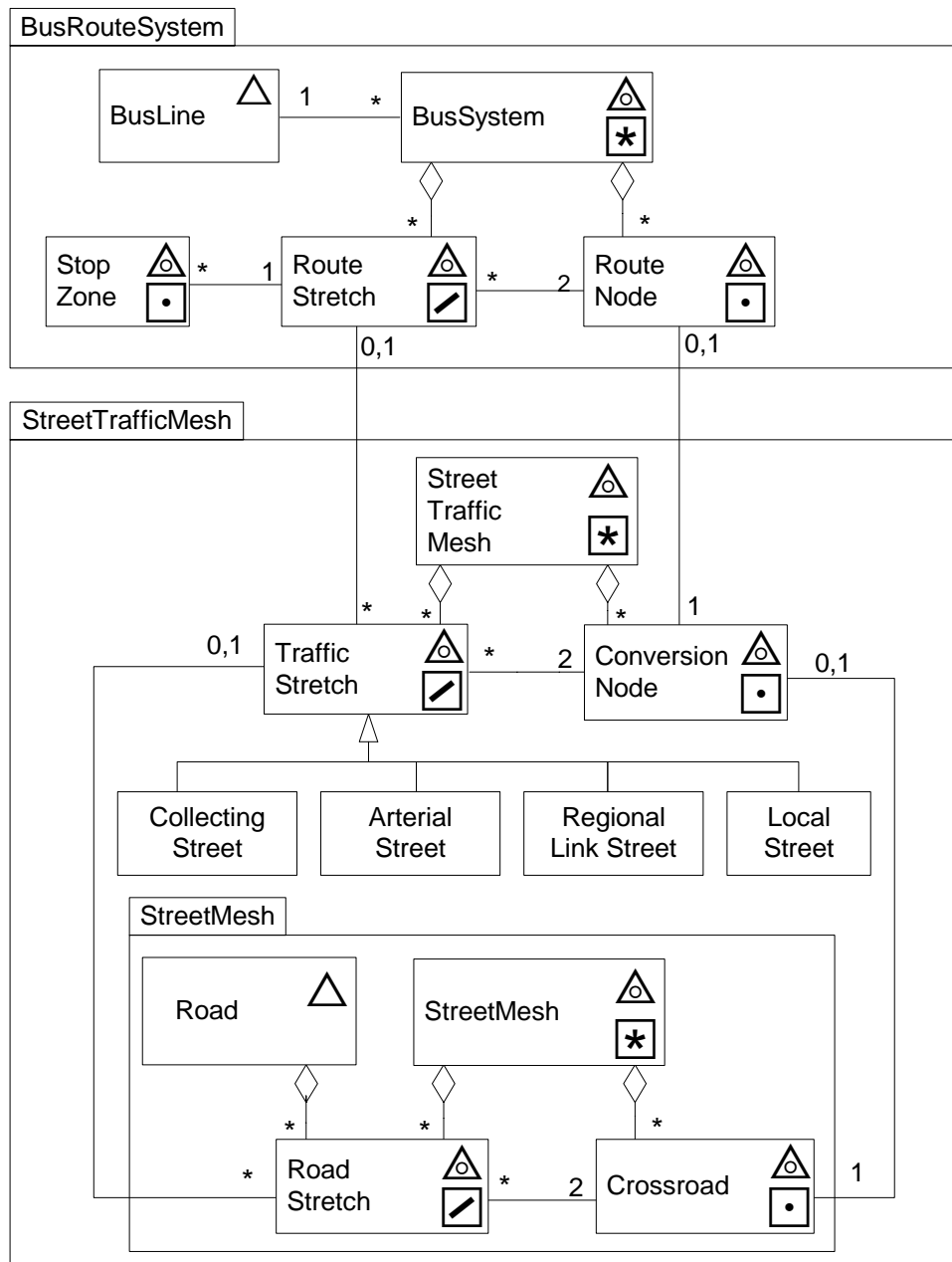


FIGURE 7 – “Urban Street Mesh” and “Urban Street Traffic Network” pattern examples

3.3 Urban Zoning Pattern

Problem:

How to structure urban cadastre data base?

Context:

The starting point for every GIS application that has a city, as the studied area is the creation of a digital cartographic base integrated to a multipurpose registration form. These two databases are used by several applications such as emergency attendance (e.g.: ambulance, firefighters, police), public school registration control, hospital distribution, tax collection, and other. These applications need information such as street structure, neighborhoods, blocks and parcel localization, and sometimes even precise information about each construction limits inside a parcel.

Forces:

- Detail level of the cadastre base depends on a previous existing spatial digital data for the modeled city. Sometimes it is not economic feasible to create the cadastre base on the wanted scale. The bigger the scale is the bigger will be the costs to generate and to maintain data.
- Many division types are used depending on the size and organization of the city. The most common types include neighborhoods and administrative divisions.
- The neighborhood concept is not unique for every city. For an instance, block not necessarily belongs to only one neighborhood.
- A parcel must have two spatial representation types: its limits representation; and the corresponding parcel's front representation also known as "block foreface".

The same may happen to squares in "square faces" representation.

Solution:

Figure 8 shows the classes' diagram that generates this pattern.

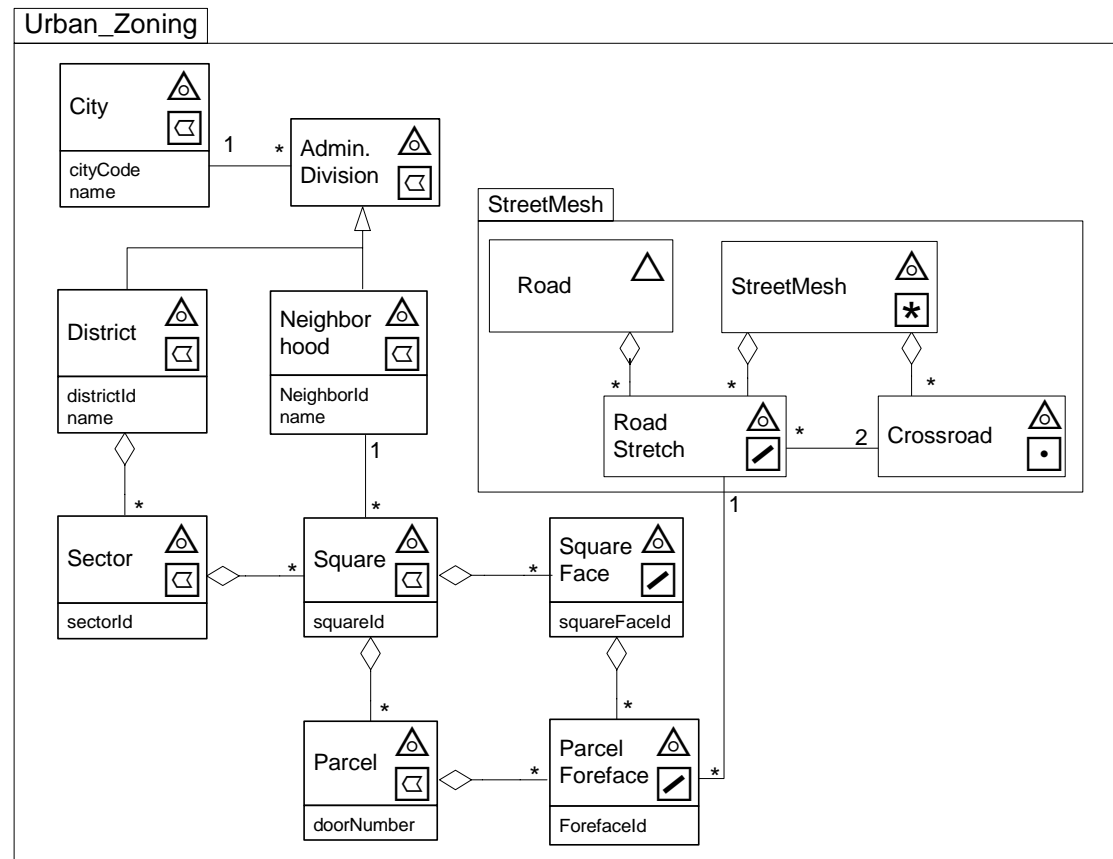


FIGURE 8 – "Urban Zoning" pattern classes' diagram

Participants:

The AdministrativeDivision class may be specialized into other city subdivisions (e.g.: census sectors, garbage collection zones, policing zones). The city or municipal site corresponds to a district.

The Neighborhood class is associated to the Square class, using a "one-to-many" relationship but this association must be adapted for each specific situation. In

some counties the neighborhoods' limits may not respect the block limits, in this case the relationship would be “many-to-many”, which would create an undesired situation.

Another variation that may occur is to relate a parcel to a road stretch. The presented solution shows the road stretch related to the parcel foreface. However there could be a direct relation between the parcel and the road stretch. In situations where the square is the highest detail level Road stretch must be related to square faces.

Related Patterns:

Urban Street Mesh.

Example:

Urban Zoning pattern application may be seen on Figure 9 that illustrates a Brazilian urban taxing over building and urban land use (IPTU).

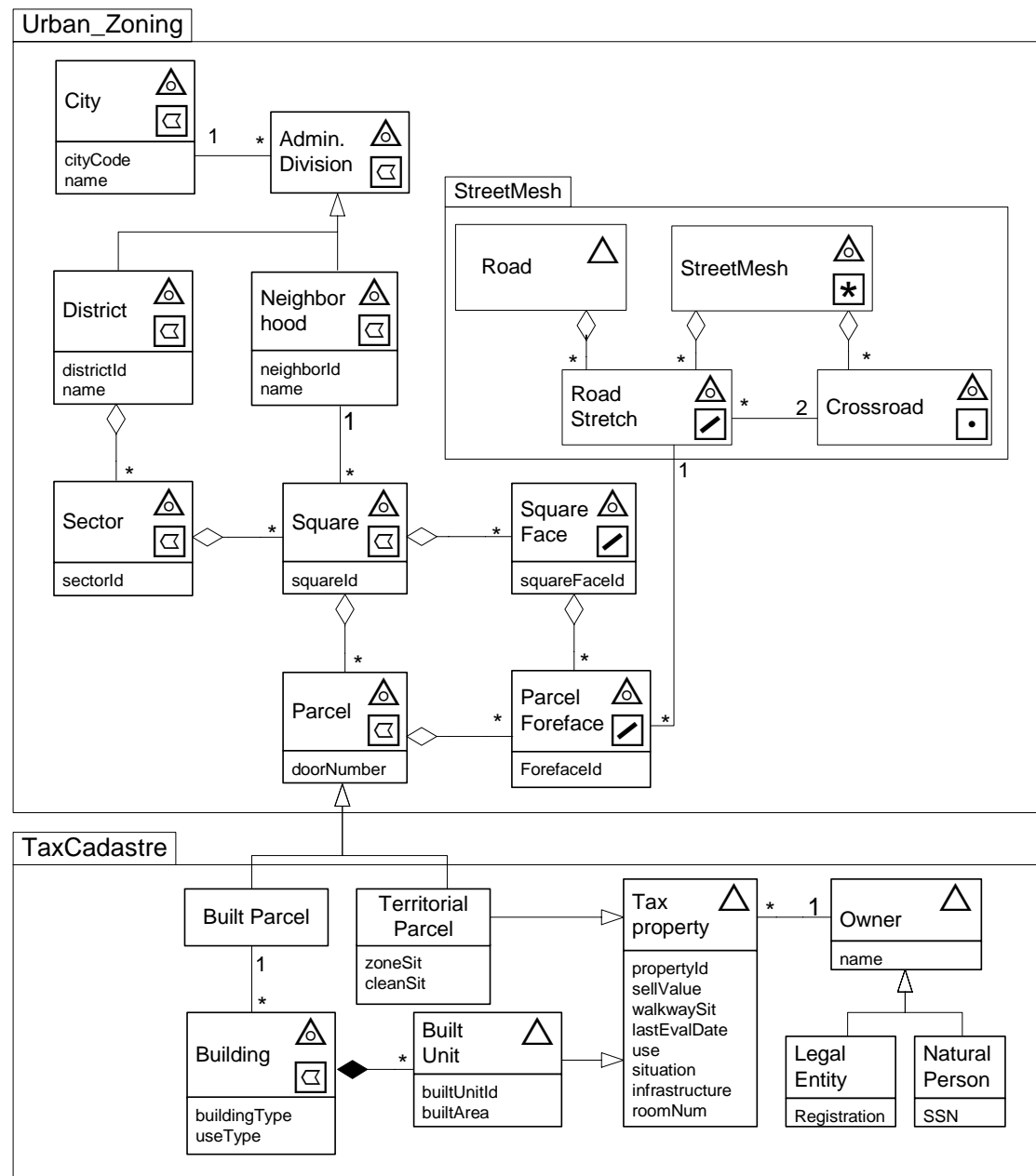


FIGURE 9 – “Urban Zoning” pattern use examples

In the shown example a built block may have many different buildings (e.g.: many buildings in a condominium, a mall). Each building may be made of many other built units (e.g.: apartments, stores). The built units and the territorial parcels (not built) are taxed as modeled by the Tax Property class, which usually is associated to a main owner. Only the most common attributes were included once attribute needs depend on system goals. The “Party” pattern [9] is used to model different parcel types and different owner types.

The example is also to show the situation where it may not be possible for one city to use georeferenced data from another city. However it is very likely that a database schema developed for one city may be largely used for others. That occurs due to legislation similarities among Brazilian cities.

4. CONCLUSIONS

The analysis patterns shown in this paper are not complete solutions as noticed previously. Patterns describe advices, initial designs and usual problem solutions. Analysis patterns need to be adapted for each specific reuse case.

The analysis patterns’ approach has an immense potential to improve municipal management applications using GIS, as well as reduce time and therefore costs in stages such as requirement analysis and database conceptual modeling.

However, for this approach’s success it is necessary to create a cooperation culture among system developers. For an example many users reuse third parties georeferenced data, but do not make their own data available [23]. Reuse a good documented solution by another designer is a very attractive idea but it is necessary that all help to the reusing approach, documenting solutions in analysis patterns form, for an instance.

An analysis pattern does not need to present an original solution. Patterns should document tested and validated solutions, for these are recurring problem solutions. Unique problem solutions do not need to be documented in pattern forms because they probably will not be needed again. Thus the evolution of pattern documentation should be done naturally by other designers. This paper’s analysis patterns can and should be improved. Critical contributions and comments are welcome.

Following this work there will be a search on alternatives to make existing analysis patterns available (for geographic databases) and also, in studying different GIS applications domains (e.g.: infrastructure networks, environmental applications) to mining new patterns. There are researches in developing tools that support searches in existing patterns [24]. A CASE tool that supports geographic database conceptual modeling is under development, which uses the UML-GeoFrame model. This CASE tool, called CASEGEO [19], also supports database design through analysis patterns.

Acknowledgements

The authors thank the contributions made by Rosana Teresinha Vaccare Braga, which have largely improved the described analysis patterns. We also thank Rodrigo Cipriano for his comments in the final version of this paper. This work had a partial grant from FAPEMIG and from CNPq.

REFERENCES

1. Booch, G.; Jacobson, I.; Rumbaugh, J. The Unified Modeling Language User Guide. Reading: Addison-Wesley, 1998.
2. Borges, K. A. V. Urban management and information and communication technologies. *Revista IP-Informática Pública*, Belo Horizonte, Vol.2, No.2, 2000. (in portuguese)
3. Buschmann, F. et al. *Pattern-Oriented Software Architecture: a system of patterns*. New York: John Wiley & Sons, 1996.
4. Chrisman, N. *Exploring Geographic Information Systems*. NY: J. Wiley & Sons, 1997.
5. Coad, P. *Object Models: Strategies, Patterns, and Applications*. 2.ed. New Jersey: Yourdon Press, 1997.
6. Fernandez, E. B. Building systems using analysis patterns. *Procs. of Int. Software Architecture Workshop (ISAW3)*, 1998.
7. Fernandez, E. B.; Yuan, X. An analysis pattern for reservation and use of reusable entities. *Procs. of Workshop in the Conference of Pattern Language of Programs - Plop*, 1999. Available at <http://st-www.cs.uiuc.edu/~plop/plop99/proceedings/>.
8. Fernandez, E. B.; Yuan, X. Semantic Analysis Patterns. In: A. H. F. Laender, S. W. Liddle, V. C. Storey (eds): *Procs. of ER2000 Conference*, LNCS 1920, 2000. Springer-Verlag, 2000.
9. Fowler, M. *Analysis Patterns: reusable object models*. Menlo Park, CA: Addison Wesley Longman, 1997.
10. Gamma, E. et al. *Design Patterns: elements of reusable object-oriented software*. Reading, MA: Addison Wesley, 1994.
11. Goodchild, M. F., Geographical data modeling. *Computers & Geosciences*, Vol 18, No 4, 1992, pp.401-408.
12. Hay, D. C. *Data Model Patterns: conventions of thought*. New York: Dorset House Publishing, 1995.
13. Johannesson, P.; Wohed, P. The deontic pattern – a framework for domain analysis in information systems design. *Data & Knowledge Engineering*, Vol.31, 1999.
14. Lisboa Filho, J.; Iochpe, C.; Beard, K. Applying Analysis Patterns in the GIS Domain. *Procs. of Annual Colloquium of the Spatial Information Research Centre - SIRC.*, Dunedin, NZ, 1998.
15. Lisboa Filho, J.; Iochpe, C. Specifying analysis patterns for geographic databases on the basis of a conceptual framework. *Procs. of ACM Symposium on Advances in Geographic Information Systems*, Kansas City, USA, 1999.
16. Lisboa Filho, J.; Iochpe, C. A study about conceptual data models for geographic database design. *Revista IP-Informática Pública*, Belo Horizonte, Vol.1, No.2, 1999. (in portuguese)
17. Marsura, P. *Banking patterns home page*. Available at <http://www.joeyoder.com/marsura/banking/> (03/09/1999).
18. Meszaros, G.; Doble, J. *A pattern language for pattern writing*. Available at http://hillside.net/patterns/Writing/pattern_index.html (01/12/98).

19. Pereira, M. A.; Lisboa Filho, J. CASEGEO – A CASE tool for conceptual modeling and database schemas automatic generation using GIS. In: *Procs. of GISBRASIL*, 2002. (in Portuguese)
20. Rawsthorne, D. A. A patterns language for requirements analysis. *Procs. of Workshop in the Conference of Pattern Language of Programs - PLoP*, Monticello-Illinois, 1996.
21. Robertson, S.; Strunch, K. Reusing the products of analysis. *Procs. of Int. Workshop on Software Reusability*, Lucca, Italy, 1993. Available at <http://www.atlsysguild.com/GuildSite/SQR/reusingAnalysis.html>.
22. Silva, C.; Iochpe, C.; Engel, P. Applying the process of knowledge discovery in database to identify analysis patterns for reuse in geographic database design. *Procs. of SBIA – Artificial Intelligence Brazilian Symposium*, 2002.
23. Weber, E. J.; Lisboa Filho, J.; Iochpe, C.; Hasenack, H. Geospatial metadata in Brazil: an experience in data documentation of an environmental GIS application. *Procs. of Int. Conference. Exhibition on Geographic Information (GISPlanet)*, Lisbon, Portugal, 1998.
24. Wohed, P. Tool support for reuse of analysis patterns – a case study. In: A. H. F. Laender, S. W. Liddle, V. C. Storey (eds): *ER2000 Conference*, LNCS 1920, 2000. Springer-Verlag Berlin Heidelberg, 2000.