GeoOOA: Object-Oriented Analysis for Geographic Information Systems*

Georg Kösters, Bernd-Uwe Pagel, Hans-Werner Six
University of Hagen
Department of Computer Science
D-58084 Hagen, Germany
E-mail: {georg.koesters,bernd-uwe.pagel,hw.six}@fernuni-hagen.de

Abstract

We stress the need of a domain-tailored requirements engineering method for the development of GIS-applications. To this end, we discuss three GIS-application scenarios, the so-called topological scenario, the network scenario and the (spatio-) temporal scenario, we frequently encountered in real life projects. Our investigations reveal that conventional RE-methods like e.g. Coad/Yourdon's Object-Oriented Analysis (OOA) are not capable to model GIS-specific requirements appropriately. To overcome the deficiencies of conventional OOA we introduce GeoOOA which complements OOA by suitable domain-tailored primitives and allows for a more adequate treatment of GIS-specific requirements. We provide GeoOOA-models of a broadcasting corporation, an electric power supply system and a land registry as first validation steps for the usefulness of GeoOOA.

1. Introduction

Over the last years, a rapidly growing interest in *Geographic Information Systems* (*GIS*) supporting storage and processing of spatially related data from geographic applications has been observed. Indicators are the increasing number of commercial GIS-products and installations as well as the great demand for user-specific applications in a broad range of problem domains. The development of such an application comprises the elaboration and implementation of a suitable domain model and the realization of a graphical user interface with sophisticated graphical facilities which is user-friendly and reflects the characteristics of the problem domain the user is accustomed to.

Consequently, the development of a GIS-application requires a systematic, software engineering based development approach. In this paper, we address the requirements engineering (RE) phase of the development process. We discuss RE for three GIS-application scenarios which frequently occurred in real life projects we carried out in cooperation with industrial companies. These scenarios span a wide range of GIS problem domains and deal with features like topological (whole-part) relationships, networks and object versions (see e.g. [4, 6, 9, 2, 11]). The investigation reveals that conventional RE-methods like e.g. Coad/Yourdon's Object-Oriented Analysis (OOA) [3] are not capable to model GIS-specific requirements appropriately and especially do not support the user's view sufficiently. Similar observations have been made in [7] and [10], where the lack of expressiveness and adequacy of conventional modeling approaches have also been complained. Concrete recommendations, however, have not been provided.

To overcome the deficiencies of conventional OOA we introduce GeoOOA which complements OOA by suitable geo-primitives and allows for a more adequate treatment of GIS-specific requirements. Concerning the typical trade-off between simplicity and understandability on the one hand, and preciseness and completeness on the other hand, GeoOOA is biased towards the user who prefers a simple and natural model of the problem domain. GeoOOA-models of a broadcasting corporation, an electric power supply system and a land registry serve as first validation steps for the usefulness of GeoOOA.

*Due to space limitations, only an excerpt of GeoOOA can be presented in this paper, namely the major primitives and their graphical representations. For further aspects of GeoOOA like, for instance, textual specifications and methodology the reader is referred to the paper's full version available as Technical Report No. 1-96 via www (http://voss.fernuni-hagen.de/gebiete/pi3/pi3.publications.html).

For the rest of the paper we assume the reader to be familiar with the object-oriented paradigm in general and OOA [3] in particular, s.t. we abstain from recalling the fundamental concepts.

2. Topological Relationships

Our first scenario concerns an application domain where objects are arranged according to *topological whole-part relationships* with certain topological constraints. A typical example is the partition relationship where the geometries of the parts are disjoint and cover the entire geometry of the whole. (Nested) partitions are, for instance, administrative districts like countries, states and counties. Querying for the parts of a given whole or updating a border line are classical operations associated with partitions.

2.1. The Broadcasting Example

As simplified example from a broadcasting domain we choose a public radio corporation which supplies a country with radio programs. The country is partitioned into local broadcasting districts each of them managed by a local department of the radio corporation. The radio corporation as well as the local departments broadcast various radio programs. For each district, transmitters guarantee that a program can be received at any location without interference phenomena.

Fig. 1 shows an OOA-model of the broadcasting domain, more precisely, the model's graphical representation. The abstract class *POINT*, resp. *REGION*, generalizes common geometric attributes and services of objects with point-, resp. 2-dimensional, geometry.

Even in our small example, it takes a second glance to find out that the classes *CONTRACT* and *PROGRAM* are non-spatial classes. Moreover, only by tracking the inheritance structures it will turn out that *TRANSMITTER* and *DEPARTMENT* are specialized *POINT* classes. Analogous problems occur with the specialized *REGION* classes *PROPAGATION AREA* and *DISTRICT*. For real life OOA-models with more than hundred classes and relationships such a modeling approach will encounter severe problems. The specification of important domain-inherent properties in terms of inheritance structures instead of, for example, in terms of domain-tailored primitives, increases the unavoidable complexity of the model and decreases its understandability.

Concerning topological relationships and constraints the situation is even worse. For instance, the model's

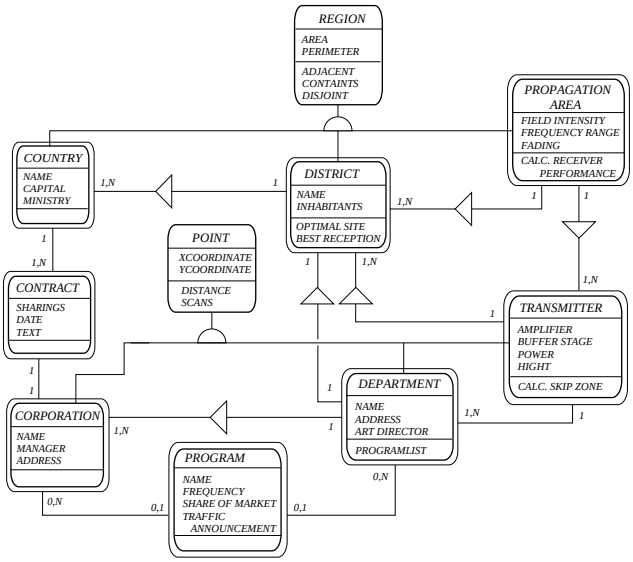

Figure 1. OOA: broadcasting example

graphical representation does not provide any information about the topological inclusion constraints of the whole-part relationship between a district and its transmitters. The partition constraints of the country and the districts as well as the topological relationship between districts and propagation areas are missing. All these essential informations cannot be expressed within the graphical representation and must be specified in the supplementary textual part. The latter will get lengthy because frequently occurring common topological constraints lead to redundant specifications. For example, whole-part structures of districts and transmitters as well as districts and departments must be completed by identical specifications of the geometric containment structures.

We summarize the major deficiencies of the OOA-model in Fig. 1 as follows:

- An easy distinction between spatial and non-spatial classes is not provided.
- The "type" of a spatial class (e.g. point, line, region) is not obvious.
- A distinction between topological and conventional relationships is missing.
- Important topological constraints are "hidden" in the textual specifications.
- Frequently occurring common topological constraints lead to redundant textual specifications.

2.2. GeoOOA: Preliminary Remarks

The deficiencies of the OOA-model of the broadcasting domain mark the starting point of the development of *GeoOOA* which enhances conventional OOA by some GIS-tailored primitives. In general, the identification of suitable primitives starts with a careful analysis of the problem domain guided by the following questions:

- What are the special characteristics (e.g. basic data types, relationships, constraints, aggregates/structures) of the problem domain?
- For each of the identified characteristics, is it relevant and fundamental enough to be incorporated as model primitive? Here it must be taken into account that too many primitives decrease simplicity and eventually readability of the model.

The GeoOOA-primitives support abstractions of spatial class types, topological whole-part structures, network structures, and temporal classes. For the specification of the semantics of spatial objects, operations, and their topological relationships we refer to e.g. [4, 6, 9, 2].

2.3. Conventional Classes and Geoclasses

From our real life projects we have learned that GIS-users as well as GIS-developers accept a domain model as useful only if it allows for an easy distinction between spatial and non-spatial classes, between different kinds of spatial objects, namely points, lines and regions, and between basic topological relationships. To this end, GeoOOA distinguishes between a *conventional class type* and four *geoclass types*. Objects without geometric properties are modeled in *CONVENTIONAL CLASSES*, point objects belong to *POINT CLASSES*, 1-dimensional objects to *LINE CLASSES*, 2-dimensional objects to *REGION CLASSES*, and scanned objects to *RASTER CLASSES*.[1] Each geoclass type provides geometric standard services similar to the operations of the corresp. spatial data type of a spatial DBMS. GeoOOA also offers predefined topological constraints, e.g. if the objects of a class may intersect or not, to improve the solidity and preciseness of the associated textual class specification.

Fig. 2 depicts the class symbols of GeoOOA. Note that we use rectangular class symbols instead of oval ones in order to distinguish between GeoOOA and conventional OOA. For sake of brevity, we omit the attributes and services of the classes in the graphical representations.

[1] If cartographic generalization is neglected the most accurate geometric representation of an object determines its class type.

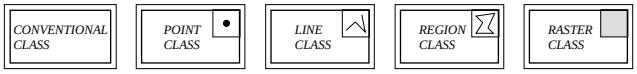

Figure 2. Class symbols

2.4. Topological Whole-Part Structures

GeoOOA offers *topological whole-part structure* primitives to model topological relationships between a whole and its parts typically occurring in GIS-application domains. Note that conventional objects must not participate in topological whole-part structures. We divide whole-part structures into *covering*, *containment*, and *partition structures*. A common property of all these structures is the non-empty intersection of the geometry of each part with the geometry of the whole.

- For a *covering structure*, the whole and its parts belong to the same geoclass type and the geometry of the whole is covered by (the union of) the geometries of its parts.
- For a *containment structure*, the geometry of the whole contains the geometry of all its parts.
- The *partition structure* is a containment structure with two additional conditions: the whole and its parts belong to the same geoclass type and the geometries of the parts form a partition of the geometry of the whole.

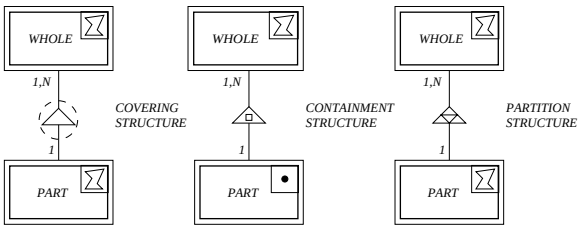

Figure 3. Topological whole-part structures

The symbols of the three primitives are depicted in Fig. 3. The annotation with cardinalities is similar to conventional whole-part structures. Keywords are provided for the textual specification of topological whole-part structures. As with conventional relationships, topological whole-part structures provide standard services like accessing or (dis-)connecting objects as well as querying for neighbored objects or objects on

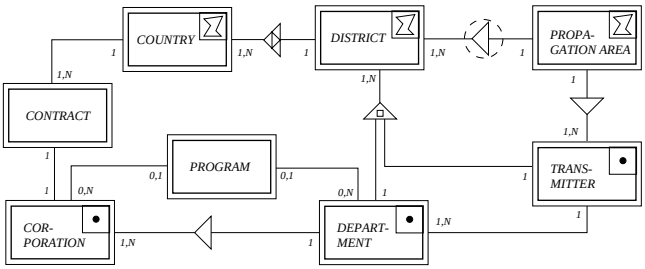

Figure 4. GeoOOA: broadcasting example

the border of a partition. Clearly, a standard service respects the topological constraints of the associated structure.

Comparing the GeoOOA-model's representation of the broadcasting example in Fig. 4 with the representation of the OOA-model in Fig. 1 reveals that the GeoOOA-representation gains clarity as well as expressiveness from the new primitives. In contrast to the OOA-model, relevant spatial information is directly expressed within the model. Besides an improved graphical representation the textual specification benefits from keywords referring to predefined topological relationships and constraints.

3. Networks

The second scenario deals with *networks* where objects of node classes are connected via objects of link classes. If node and link objects are spatial objects, a network is called *spatially embedded*. Examples are road maps or power, water and gas supply networks. Typical operations are connectivity checks, browsing, and flow analysis.

3.1. The Electric Power Supply Example

We take an simplified example from the electric power supply domain. Central entities are lines for high and low voltage. The lines link node entities together, for instance, power plants in the high voltage network and poles in the low voltage network. The H.V.-network and L.V.-network are connected via well-defined transition points, so-called transformers, which belong to both networks. Important operations on these networks are, for instance, checking that the capacities of the network units meet the capacity demand of the consumers, the supervision of maintenance intervals including the generation of a maintenance plan, and reports of various kinds (e.g. to support inventory).

We assume the OO-analyst to be familiar with the GeoOOA-primitives introduced, i.e. with geoclasses and topological structures. Then his attempt will rather probably result in a model similar to that presented in Fig. 5. The model focuses on a "structural" view: The graph structure is modeled by organizing the basic graph entities in node classes and line classes while inheritance is used to extract common properties and to avoid redundant information. Relevant information, e.g. which kinds of lines are incident to which kinds of nodes or which kinds of nodes and lines belong to a H.V.-network or L.V.-network, respectively, is not depicted because their illustration would substantially increase the complexity of the representation (see Fig. 6). Hence, domain-inherent information is not reflected in the graphical model and must be specified within the supplementary textual part.

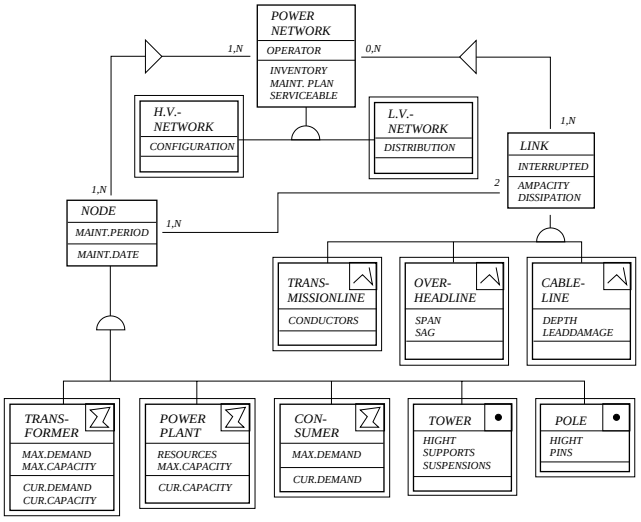

Figure 5. Enhanced OOA: power supply example (first approach)

We summarize the deficiencies of the (graphical) model in Fig. 5 as follows:

- The graph structure, i.e. which kinds of lines are incident to which kinds of nodes, is not reflected.
- Information about which kinds of nodes and lines belong to the H.V.-network, resp. L.V.-network, is not provided.
- Important domain-inherent information must be specified within the textual documents.

Concentrating on the information missing in the first attempt yields as possible result the variant shown in

Fig. 6.[2] The information about structural properties is traded for a detailed modeling of whole-part relations and instance connections. Taking into account that the example covers only a small subset of the real problem, one can imagine the complexity of the entire model. In our projects, the users were not willing, resp. able, to understand such a model in detail. Despite of its complexity common properties of node, resp. link, classes, which are expressed in terms of inheritance structures in the first model, are not worked out and lead to redundant textual specifications. Since node, link and network classes are distinguishable only by names, the graph structure is not sufficiently reflected. As in the first model's representation, relevant information must be specified in the textual part.

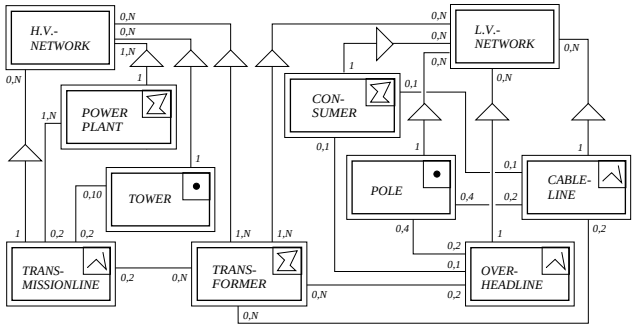

Figure 6. Enhanced OOA: power supply example (second approach)

We summarize the deficiencies of the attempt in Fig. 6 as follows:

- The graph structure is not sufficiently reflected, because the distinction between node classes and link classes is indicated only by suitable class names.
- Common properties of node classes and link classes cannot be identified.
- Important domain-inherent information must be specified within the textual documents.

Roughly speaking, the disadvantages of the first approach are the advantages of the second and vice versa. However, the simple idea of combining both models into one (as far as possible) to overcome the deficiencies of the partial models would not yield the desired result. From the discussion above it should be clear, that the resulting graphical representation would be of tremendous (and inadequate) complexity and would be accepted neither by users nor by system developers.

The electric power supply example has demonstrated that the enhanced OOA-models of network scenarios fall short of basic requirements and exhibit for a network-tailored primitive.

3.2. Network Structures

A suitable model for network scenarios with usually numerous information is always a compromise between completeness and preciseness on the one hand and readability and understandability on the other hand. To meet the user's need the model must allow for an immediate answer of at least the following questions:

- Does a certain class belong to a network and, if so, to which one?
- Is a certain class a *NODE* or a *LINK* class?
- Does a certain *NODE* or *LINK* class connect different network structures?

Detailed information about, for example, which *NODE* classes are connected by a link and how many links of a certain *LINK* class are incident to a certain node must not hide basic information.

The *network structure* primitive of GeoOOA can be regarded as a generic pattern for three roles with the usual graph semantics, namely the node role, the link role and the network role. An instantiated network structure consists of one *NETWORK class* and at least one but possibly more *NODE*, resp. *LINK*, *classes*. A geoclass may occur in more than one network structure and may even play a node role in one and a link role in another network structure. A network structure provides several standard services, e.g. test on connectivity or cycles, determination of sources and sinks.

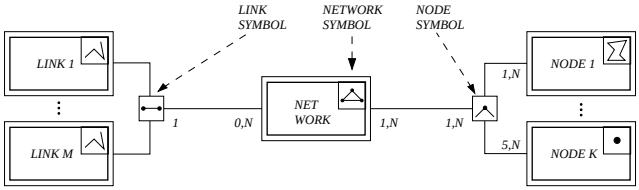

Figure 7. Network structure

The graphical representation of a network structure is depicted in Fig. 7. It consists of one network class

[2] For sake of brevity, we sometimes omit the attributes and services of the classes in the graphical representations.

connected to all *NODE*, resp. *LINK*, classes. We abstain from an exact picture of the up to $n \cdot m$ (annotated) instance connections (assuming n *NODE* classes and m *LINK* classes) and provide only condensed information about these details. The exact specification is shifted to the textual documents. For sake of clarity, we characterize the *NETWORK* class by a network symbol while the edges to *NODE*, resp. *LINK*, classes are enhanced by symbols denoting the role of the associated classes. For modeling common properties of the *NODE* classes an abstract *NODE* class is associated with the node symbol. Analogously, an abstract *LINK* class is associated with the link symbol. The annotation of the edge from the node symbol to a *NODE* class denotes the minimal/maximal number of links incident to the *NODE* class. The edge between the *NETWORK* class and the *NODE*, resp. *LINK* symbol, are annotated by the minimal/maximal numbers of nodes, resp. links, the network consists of and, vice versa, by the minimal/maximal number of different networks a node, resp. a link, may belong to.

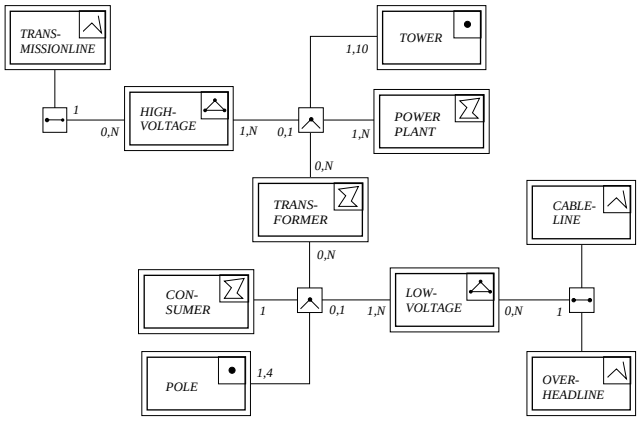

Figure 8. GeoOOA: power supply example

Fig. 8 shows a GeoOOA-model of the electric power company example giving an impression of a concrete network scenario including its annotations. Compared to the corresp. models in Fig. 5 and 6, both expressiveness and clarity have – as we feel – significantly gained from the network primitive. However, a compromise between simplicity and preciseness is always subject to personal taste.

4. Temporal Information

Over the last decade, GIS-activities are mainly characterized by the implementation of basic data models and applications on the one hand and digitalization of spatial data ("building systems and filling them with data") on the other hand. The next 10 years will rather probably be the decade of more sophisticated, customized applications and data maintenance ("refining systems and keeping data up to date"). During maintenance new objects come up, attribute values including the geometry of objects change, and objects disappear. To cope with these requirements the versions of objects and relationships must be determined and ways to restore former versions must be identified.[3] Our third example reflects this problem domain and deals with versioned (spatial) objects, so-called *(spatio-) temporal objects*. Examples are opinion research, land registry and weather forecasting. Typical temporal operations are restoration and version comparison.

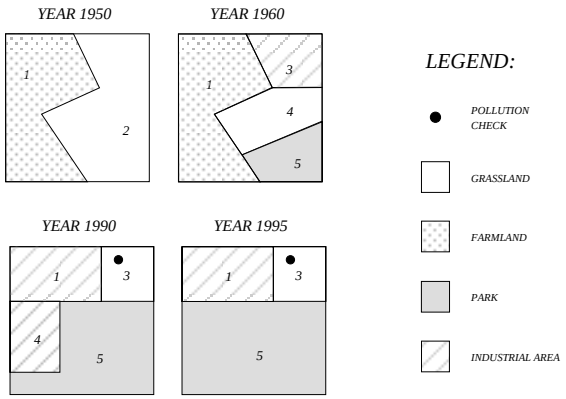

Figure 9. Spatio-temporal domain

4.1. The Land Registry Example

We take a simplified example from the domain of a land registry with parcels whose boundaries, usages and owners change over the years. To illustrate the problem domain, we trace the evolution of a region over a period of time (see Fig. 9). In 1950 the region is subdivided into two parcels. One parcel is used as farmland, the other is uncultivated grassland. In 1960 the owner of parcel 2 died and parcel 2 is split up into three new parcels. Between 1960 and 1990 we observe a "shifting" of parcel 4 due to buying and selling of subareas. After closing the chemical factory in parcel 3 in 1990 a pollution check exhibited the contamination

[3] In recent years, much research has been devoted to temporal or spatio-temporal databases in general and database models and index structures in particular (see e.g. [8, 1, 11, 5]).

of the parcel, s.t. any usage will be prohibited until the year 2000. Finally, in 1995 the owner of parcel 5 bought parcel 4 and both parcels are merged into one.

If we neglect all temporal aspects, GeoOOA introduced so far can be applied to the land registry example in a straightforward manner and results in the model depicted in Fig. 10. This skeleton misses the following temporal informations:

- Changes of a parcel's area and usage induce a new version of that parcel. (Note that the identity of the parcel remains the same.)
- Pollution checks are worthless without the date of examination.
- Buying and selling of parcels are recorded in the land register.
- Whenever parcels are split up or merged, some parcels are "born" and others "die". Each parcel's journey through life must be traceable with respect to all its parents and sons.

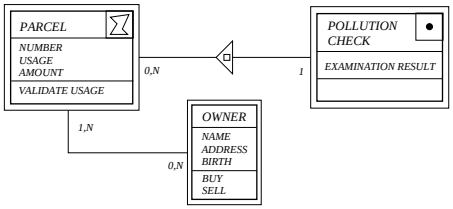

Figure 10. Enhanced OOA: land registry example (temporal informations excluded)

Parcels are time-dependent objects in a strong sense: After a parcel's birth new versions are created by changes and sometimes the object dies. Pollution checks are time-dependent in a weak sense: A pollution check object is born at the time the examination is performed, records this event and remains unchanged forever. Hence, only information about a certain event is provided. While weak time-dependency can be modeled sufficiently by simply adding a timestamp attribute (e.g. an attribute *DATE*), strong time-dependency cannot appropriately be covered by GeoOOA introduced so far.

The attempt in Fig. 11 is guided by the fact that object identity must not be affected by object versioning: Time-dependent attributes that are affected by the same event and hence are modified at the same

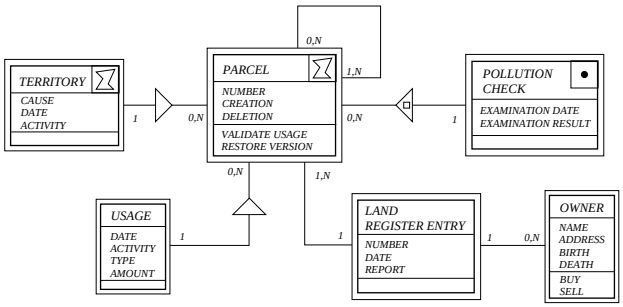

Figure 11. Enhanced OOA: land registry example (temporal informations included)

time (e.g. attributes *USAGE* and *AMOUNT*) are combined into a class and supplemented by a timestamp attribute. As a consequence, each *PARCEL*-object is decomposed into a time-independent kernel and several time-dependent parts, each reflecting an elementary versioning event. For instance, changing a parcel's usage forces the creation of a new *USAGE*-object which records this event. Service *RESTORE VERSION* of class *PARCEL* collects all time-dependent parts valid at a certain point in time (in the past) and provides an integrated view of a parcel object at that time.

It is by no means an easy task for the user to understand that class *PARCEL* is decomposed just to model its time-dependent aspects – at least if the initial model in Fig. 10 is not at hand. The model appears unnatural, because the time-dependent parts have no counterparts in the problem domain – they are not objects in the pure sense of OOA – and the structure of the problem domain gets hidden behind numerous "artificial classes". In the worst case, we get as many time-dependent classes as attributes. Moreover, we face the well-known redundancy problem caused by semantically similar ("standardized") classes which cannot appropriately be solved using inheritance.

Time-dependent relationships can be regarded as attributed relationships with at least one timestamp attribute. In conventional OOA attributed relationships are modeled as instance connections which are split into so-called *event classes*. Correspondingly, in the revised model, parcels and owners are connected via objects of the event class *LAND REGISTER ENTRY*. Each entry reflects a sale of a parcel and records the sale's date as well as some other contract information. Unfortunately, regular classes and event classes can neither be distinguished graphically nor textually, and hence their different semantics cannot be expressed.

Since GeoOOA does not provide a suitable primitive

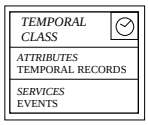

Figure 12. Temporal class symbol

for the temporal parent-son relationship yet, the ancestry of parcel objects is modeled as looped instance connection. As usual, a lengthy textual specification must compensate for the lack of expressive power.

We summarize the major deficiencies of the model in Fig. 11 as follows:

- An easy distinction between time-dependent and time-independent classes is not provided.
- Modeling strongly time-dependent classes results in numerous artificial classes inflating the model and hiding basic structural informations.
- Event classes (possibly with timestamp attributes) cannot be observed at first glance.
- Ancestry relationships are not supported appropriately.

4.2. Temporal Classes

To cover temporal requirements as well, we complement GeoOOA by some temporal primitives. To our opinion, the analysis of the non-temporal domain structure is of primary concern compared to the modeling of temporal information. Thus, our design rationale behind temporal primitives is to leave the highest degree of freedom to the non-temporal analysis. More precisely, complementing a non-temporal model by temporal primitives should not result in structural changes.

GeoOOA distinguishes between *temporal and non-temporal classes*. For an object of a temporal class, a certain service, called *event*, creates a new version. Attributes affected by the same event are grouped into a *temporal record*. A temporal record contains a *timestamp attribute* referring to the moment it was created. Temporal classes offer standard attributes recording the object's birth and death as well as standard services to create, restore, and compare versions.

The graphical representation of a temporal class is depicted in Fig. 12. The clock symbol allows for an easy distinction between temporal and non-temporal classes. A different font style is used to indicate temporal records and events.

4.3. Attributed Instance Connections and Contract Classes

Regarding *attributed instance connections* we basically adopt the OOA-idea, but slightly modify the graphical representation as well as the textual specification to emphasize the particular semantics. In addition, we rename event class in *contract class* to preclude a confusion with events of temporal classes.

Objects of contract classes, contracts for short, do not split instance connections but provide qualifying attributes. A contract either attaches attributes to a single instance connection or ties up a bundle of instance connections and attaches the same attributes to the entire bundle. Canceling a connection forces the removal of the corresp. contract and vice versa.

Fig. 13 shows the graphical representation of an attributed instance connection and its contract class in GeoOOA. The cardinality annotations at the contract class denote the number of objects which may be connected.

Both, contract classes and temporal classes do not occur as self-reliant components in a GeoOOA-model. They should rather be regarded as well-defined properties which may hold for any conventional class or geoclass. Note that contract and temporal class properties are not mutually exclusive, s.t. *temporal instance connections* are instance connections attributed with temporal contracts.

Figure 13. Attributed instance connection

4.4. Ancestry Instance Connections

Pictorially speaking, *ancestry instance connections* maintain the family tree of objects, determine to which class a descendant of an object may belong, and specify the circumstances of their birth. When a *descendant* (object) is born it takes over parts of current states of its ancestors (e.g. parts of their geometries or certain attribute values). An *ancestor* object is usually heavily modified during "birth" and may even "die in childbirth". For instance, if a parcel is split up, descendant parcels are born which "dynamically inherit" region and eventually owner of the ancestor; in this case, the ancestor parcel dies.

More formally, an ancestry instance connection is a directed instance connection between objects of tempo-

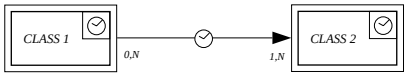

Figure 14. Ancestry instance connection

ral classes. It always points from ancestors to descendants and guarantees that the birth of a descendant is dated after the birth and before or equal to the death of the ancestor. The symbol of an ancestry relationship is depicted in Fig. 14. The annotation with cardinalities is similar to conventional instance connections.

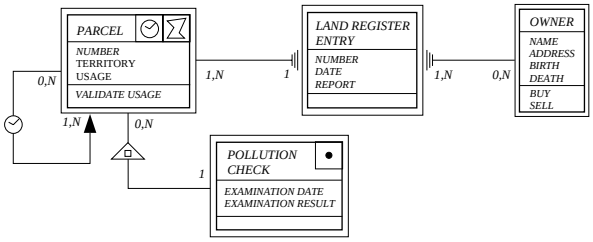

Figure 15. GeoOOA: land registry example

Comparing the GeoOOA-model of our land registry example in Fig. 15 with the model in Fig. 11 reveals that relevant temporal information is directly expressed within the model's graphical representation, while the structure of the corresp. non-temporal model is retained.

5. Future Work

Applicability and success of a method for modeling a complex application domain clearly depend on an appropriate tool support. At the moment, we are working on a development environment including a graphical editor which provides different abstraction layers and a consistency checker controlling the correct application of the method. The development environment provides the basis for a further validation of GeoOOA with large scale projects. We plan to compile a library of domain specific RE-patterns, e.g. reference networks for water, gas or power supply systems.

(Rapid) Prototyping which accompanies the RE-phase helps to identify wrong, missing or ambiguous requirements. Since GIS-applications comprise sophisticated graphical user interfaces, (user interface) prototypes are not only important for clarifying the requirements but can also be used as development basis for the user interface of the final product. We plan to incorporate a tool for incremental prototyping.

In the long term, we will address the step from requirements engineering to software design. We aim at the systematic derivation of a suitable software architecture from a GeoOOA-model. Here the underlying target system must be taken into account.

References

[1] K. Al-Taha, R. Snodgrass, and M. D. Soo. Bibliography on spatiotemporal databases. *SIGMOD RECORD*, 22(1):59–67, 1993.

[2] E. Clementini, P. D. Felice, and P. van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In D. Abel and B.-C. Ooi, editors, *Proc. 3rd Int. Symposium on Large Spatial Databases (SSD)*, pages 277–295, Singapore, June 1993. LNCS No. 692, Springer.

[3] P. Coad and E. Yourdon. *Object-Oriented Analysis*. Prentice-Hall, Englewoods Cliff, New Jersey, 2nd edition, 1991.

[4] R. Güting. Geo-relational algebra: A model and query language for geometric database systems. In J. Schmidt, S. Ceri, and M. Missikoff, editors, *Proc. EDBT*, pages 506–527, 1988.

[5] H. Raafat, Z. Yang, and D. Gauthier. Relational spatial topologies for historical geographical informations. *Int. Journal of Geographical Information Systems*, 8(8):163–173, 1994.

[6] M. Scholl and A. Voisard. Thematic map modelling. In *Proc. 1st Int. Symposium on Large Spatial Databases*, pages 167–190, Santa Barbara, 1989.

[7] T. Smith, S. Menon, L. Star, and E. Estes. Requirements and principles for the implementation and construction of large-scale geographic information systems. *Int. Journal of Geographical Information Systems*, 1(1):13–31, 1987.

[8] M. D. Soo. Bibliography on temporal databases. *SIGMOD RECORD*, 20(1):14–23, 1991.

[9] C. Tomlin. *Geographic Information Systems and Cartographic Modelling*. Prentice-Hall, Englewoods Cliff, New Jersey, 1990.

[10] M. Worboys. Object-oriented approaches to geo-referenced information. *Int. Journal of Geographical Information Systems*, 8(4):384–399, 1994.

[11] M. Worboys. A unified model for spatial and temporal information. *The Computer Journal*, 37(1):26–34, 1994.