D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

A CONTEXT SENSITIVE MODEL FOR SHARING DISTRIBUTED GEOSPATIAL INFORMATION

Bishr Y.[1], Molenaar M.[2], and Radwan M[2].
[1]Institut für GeoInformatik, Universität Münster, Germany
[2]Department of Geoinformatics
International Institute for Aerospace Surevey and
Earth Scienses (ITC), Enschede, The Netherlands

ABSTRACT

Research on Interoperability of geospatial information systems is becoming widespread in GI research community. We have identified three types of heterogeneity: semantic, schematic, and semantic. We argue that the current technological development is focused on resolving the syntactical problems, with little to know attention to the schematic and semantic differences, respectively. In this paper we introduce a three-layered model to provide interoperability at the semantic level. Architecturally the model is built on a proven syntactical model known as the formal data structure. We will show how this model can be extended to incorporate solutions to the schematic and semantic differences.

1 INTRODUCTION

During the last decade, we have witnessed increasing possibilities to access information via the Internet. The necessity of suitable tools for information organization, extraction and integration has become more and more evident. In the “global information” perspective, the added value of digital information is no more only bounded to the particular application which motivated its acquisition, but tend to increase in dependence of its reusability. In this situation, the crucial characteristic of a piece of information is what it is about, i.e., the real world features it refers to [**Guarino, 1997**]. One way to achieve this semantic transparency is to present reusable information in the user’s native database model as if it was collected and stored in the first place with the user’s application in mind.

Achieving this goal requires, among other things, data models that hold information about the meaning of its objects and what they refer to in the real world. We call this type of information *Context Information*. *Context* is the domain where the process of abstracting the real world to a data model occurs. In previous publications we have introduced the syntactic, schematic and semantic differences as three types of heterogeneity which arise when two more or more information systems share their data [Bishr, 1996; Bishr, et al., 1996; Bishr 1997; and Bishr, 1998]. Figure 1 exemplifies these differences and shows how they can be appended to the theory of spatial information.

The foundation of spatial objects representation is formalized in a model called the Formal Data Structure, or FDS [Molenaar, 1996 (b)]. This foundation forms the GIS syntax. At the lowest level of the syntactic definition, as shown in Figure 1, we find the classic data structures, i.e., field and object based approaches. The GIS theory formalizes the topologic relationships amongst fields and objects, uncertainty aspects, and the handling of geometry and topology of fuzzy objects [Molenaar, 1996 (a)]. The theory introduces a consistent framework for object hierarchies, such as generalization and aggregation, which form the building blocks for schema definitions.

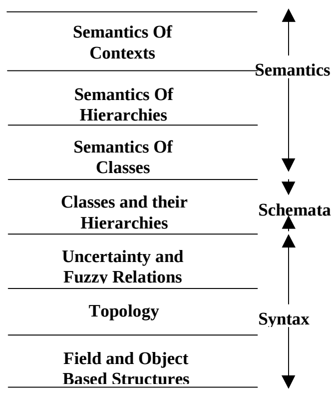

Figure 1 Syntactic and semantic definition.

Semantics in our case refers to the relationship between the database objects and the real world features they refer to. Elsewhere we have discussed this issue extensively [Bishr, 1997]. Figure 1 shows the semantics to be built onto the syntactic and schematic definitions. Class intension, as well as the relationship between instances of the classes and the real world features are considered a semantic problem. Finally, the relationship between contexts is the third semantic level. Defining the semantics of contexts establishes a relationship between different GISs.

The purpose of this paper is to present a model that attempts to resolve these types of heterogeneity and eventually achieve semantic interoperability. The model is called Semantic Formal Data Structure, SFDS. It has three main layers, namely, syntactic, schematic, and semantic layer.

2 A PROTOTYPE FOR SHARING INFORMATION

To provide a framework for information sharing at the semantic level a prototype was developed. Only the general architecture of the prototype is briefly described. Detailed description can be found in [Bishr, 1997]. The prototype is built using the available technologies and the semantic formal data structure, which is the focus of this paper, as shown in Figure 2.

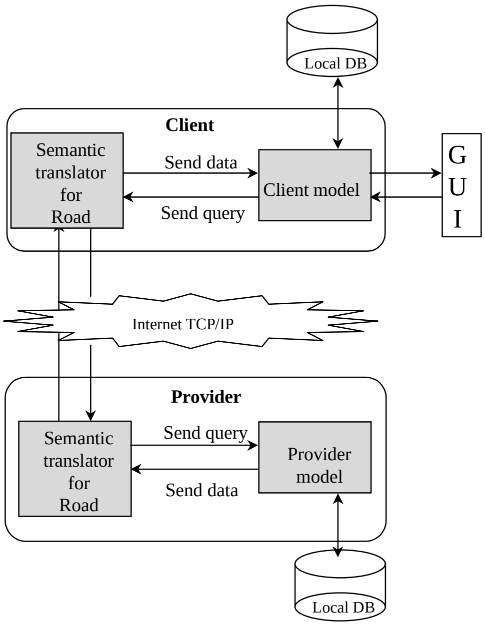

Figure 2 : Architecture of the prototype.

It is possible to provide a direct mapping between the schemes of two databases. Although this is possible in a limited number of databases, the direct mapping becomes impractical when a large number of databases is involved. We then require an intermediate context which databases can map to, if they need to share their data. If a database is to exchange information with another, it means that they will share interest in part of the real world, although they have modeled it differently. In this case it is necessary to develop a model which is an abstraction of the common part of the real world. We call this model, proxy context.

The basic idea is to provide a level of autonomy to clients and providers without enforcing a fixed data model. The concept of semantic translator is introduced for this purpose [Wiederhold, 1992]. A semantic translator is a middleware which can map between spatial database schemas while preserving their semantics. A client, or a provider, presents their data model, or part of it for sharing. A semantic translator is developed such that it maps between the client's data model and provider's data model and is known as the proxy context. In our approach semantic translators implements the semantic formal data structure, as will be shown in the next section.

It is important to stress here that a semantic translator supports only one certain application domain. For example, in case that clients or providers want to share road and cadastre information, two semantic translators are developed, one for each application domain.

3 CHARACTERISTICS OF SFDS

The semantic formal data structure, SFDS, has three layers: the syntactic layer, which takes the formal data structure as its foundation; the schematic layer where we propose a reference model, which any federated schema can be attached to; and the semantic layer, which includes the context information. The general characteristics of SFDS are:

- At the first layer, the syntactic heterogeneity is resolved. SFDS adopts the formal data structure, FDS.
- At the second layer, the schematic heterogeneity is resolved. SFDS adopts the concept of federated schemas, to which each database schema should map [references on federated databases]. The design of a federated schema is specific for a particular application domain.
- The context information is needed to map between the heterogeneous database schemas. For this reason, the semantic layer provides a mechanism to associate this information with the federated schema.

3.1 The Syntactic Layer of SFDS

In FDS an object that belongs to a class has an identifier, for a unique identification in the database, as well as geometric and thematic descriptors. As shown in Figure 3, a class has a label and a list of attributes, which characterizes that class. A database object is a member of some class and has the attribute structure of the class it belongs to. For example, a house can be defined by a unique identifier in the database (e.g., a house number and address), has a geometric description (e.g., bounding rectangle), and has thematic descriptors (e.g., number of rooms and owner).

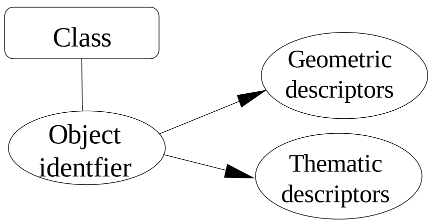

Figure 3 : Representation of spatial objects in FDS.

FDS defines a syntax for both the geometric and the thematic descriptors of spatial objects. A brief discussion of the geometric and the thematic formalism is provided in sections 3.1.1, 3.1.2, respectively. The formalism of the geometric syntax is given in a 2-D planar graph. Only a summary of the syntax of the vector maps is presented here. More details can be found in [Molenaar, 1989; Molenaar, 1991; Molenaar, 1994; Molenaar, 1996 a, b, c, d].

3.1.1 The Geometric Syntax of FDS

The formal data structure formalizes the syntax for spatial objects in a vector database. In FDS the same syntax can be applied to raster structure and other tessellations by considering them as faces in a planar graph. The spatial structure is expressed in terms of nodes, edges, and faces. The general characteristics are listed, and shown in Figure 4:

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

- The model has three geometric primitives. These are nodes, edges, and faces.
- Three types of complex objects: point objects which are defined by nodes, lines which are collections of edges, and area objects which are geometrically described by a collections of faces.
- An edge has a begin and an end node.
- An edge has one left and one right face.
- Two edges join at no more than one node.
- Two line objects can cross, or intersect, each other at a node, one line object is upper and the other is lower.

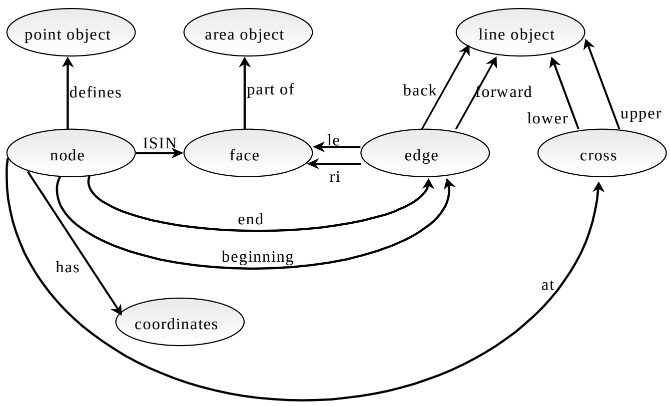

Figure 4 : Formal data structure.

Position information is given by the coordinates of the nodes and the geometry is described by means of the geometric primitives. Points and lines are dealt with as nodes and edges, respectively. Therefore, the underlying mathematics for the geometric description of spatial objects in a vector map is provided by graph theory.

The geometry of a vector database is represented in a planar graph *G(N,E)* where

- $N = \{n_1, \ldots, n_l\}$ is the set of nodes in a context
- $E = \{e_1, \ldots, e_m\}$ is the set of edges in a context, where
- $\forall e_j = \{e_j = (n_p, n_q)\}$ is an ordered pair where $n_p, n_q \in N$, and
- We say that $\mathcal{F}$ is the set of faces generated by (N, E) in a planar graph.

FDS identifies six types of topologic relationships between pairs of elementary objects (only a list is provided):

- Point object - point object
- Line object - point object
- Area object - point object
- Line object - line object
- Area object - line object
- Area object - area object

3.1.2 The Thematic Syntax of FDS

Objects in FDS have thematic and geometric descriptors. The geometric descriptors are briefly shown in the previous section. Objects in a context are distinguished on the basis of their different characteristics. In most GIS applications these differences are thematic. Two objects are distinct, if their thematic descriptors are not equal. Objects with similar characteristics are collected in classes. Criteria are formulated for each class to specify when an object is a member of that class. The following formalism assumes that classes and objects are defined within a context.

Let *C* be a class which has a set of attributes *A*. We use the national convention

$LIST(C) = \{A_1, \ldots, A_r, \ldots, A_n\}$ in *Cont*

If an object *O* passes a test formulated in a decision function for a class *C,* it will be a member of that class and that will be expressed by the membership function:

M(O, C) = *True* if *O* is a member of *C*
= *False* otherwisein *Cont*

The attribute structure of objects is determined by the class to which they belong, so that each object has a list containing one value for every attribute of its class. An object then takes the attribute structure of its class.

M(O, C) = True implies that
$VLIST(O) = \{a_1, \ldots, a_r, \ldots, a_n\}$ in *Cont*
where $a_r = A_n(O)$ is the value of A_n for object *O*
and $A_r \in LIST(C)$
and $a_r \in DOMAIN(A_r)$

The extension of a class is the set of all the objects that belong to it, hence
$Ext(C) = \{O \mid M[O, C] = True\}$ in *Cont*

Classes C_i and C_j are semantically distinct, if they have different attribute structures, i.e.,
$LIST(C_i) \neq LIST(C_j)$ in *Cont* where $i \neq j$

Classes within a context are exhaustive, which means that all objects identified in a context must belong to some class. Classes which are semantically distinct within a context, are disjunct. The notion of disjunction implies that an object can belong to only one class within a context. The two conditions can then be defined respectively as follow:

- $(\forall O \mid O \in Cont)\ (\exists C \mid C \in Cont) \Rightarrow M[O, C] = True$
- Let P be the set of all classes in Cont, i.e.,
- $P = \{C_1, \ldots, C_c\}$ and
- $(C_j, C_k \in P \mid C_j \neq C_k) \Rightarrow Ext(C_j) \cap Ext(C_k) = \emptyset$

A superclass has the list of common attributes between a set of classes. The classes with the common attributes are known as the subclasses of the superclass. This implies that a subclass has the attributes of its superclass (i.e., inheritance) in addition to its own attributes. Hence, the attribute value list of an object contains the values of the attributes of its class and the superclass. The extension of a subclass is also part of the extension of its superclass but not vice versa. For example, a superclass *Transportation* can have *Roads* and *Railways* subclasses. The extension of roads is also a part of the extension of transportation but the extension of transportation is not necessarily the extension of roads. The above discussion can be formalized as follows:

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

Let C_s be a superclass of C_k, then
$O \in Ext\ (C_k) \Rightarrow O \in Ext\ (C_s)$
If $O \in$ Ext C_k then
$VLIST(O) = \{a_1, \ldots, a_r, \ldots, a_n\}$ in *Cont*
$A_r \in LIST(C_k)$
$a_r = A_r(O)$ is the value of A_r for object O
$A_r \in$ List(C_k) $\cup$ List(C_s)
Ext(C_k) $\subseteq$ Ext(C_s)

The relationship between the geometric description and the thematic description is formalized in FDS. Only the relationship between faces and the thematic description is shown here, the same can be applied to point and line objects. Based on the requirement that each face in a context is related to exactly one area object, each edge is related to at most one line object and each node to at most one point object. Therefore, the objects form the geometric partition of the underlying part of the real world being modeled. The relationship between the geometric description and classes can then be found through the extensions.

Let the set of faces related to a class C_p in a context be:
$\mathcal{F}_{cp} = U_{oi \in cp}\ \mathcal{F}_{oi}$ in *Cont* where $\mathcal{F}_{oi}$ is the set of faces of object O_i
Let the set of faces related to a class C_q in the same context be:
$\mathcal{F}_{cq} = U_{oj \in cq}\ \mathcal{F}_{oj}$ in *Cont* where $\mathcal{F}_{oj}$ is the set of faces of object O_j

The fact that two area objects are disjunct, i.e.,
$\mathcal{F}_{oi} \cap \mathcal{F}_{oj} \neq \emptyset$, and
the extensions of two semantically distinct classes are also disjunct, i.e.,
Ext(C_p) $\cap$ Ext(C_q) = $\emptyset$
implies that the faces of the two classes are disjunct, i.e.,
$\mathcal{F}_{cp} \cap \mathcal{F}_{cq} = \emptyset$

After defining the relationship between thematic and geometric characteristic of spatial objects, the model presents the concepts of generalization and aggregation hierarchies [Smith et al., 1977]. There are four strategies for generalization and aggregation:

1. *Geometry-driven generalization*: The execution of the strategy is mainly dependent on the geometric description of the spatial data. Such a type of generalization is mainly applied in cartographic generalization.
2. *Class-driven generalization and aggregation*: In this strategy the aggregation is executed on the basis of the thematic information of the spatial objects, while the generalization is based on relationship between class intension. Adjacency relationship is mostly required in aggregation hierarchies. Adjacent objects which are of the same type are generalized to more general classes. For example adjacent area objects of the classes forest and grass land are aggregated to form larger objects of natural vegetation.
3. *Function-driven aggregation*: Objects from different classes at a low aggregation level, i.e., elementary objects, defined in one context, are aggregated to form a new complex object of another class. Elementary objects have part-of relationship with complex objects from higher aggregation levels. For example homogeneous geologic structures and soil types are aggregated to form soil mapping units.
4. *Structural generalization*: In this generalization strategy the aim is to simplify the description of a spatial system, while leaving the overall structure intact. For example in a utility database, the water pipes network can be generalized into main water pipes by eliminating house connections and maintaining the main pipes.

3.2 The Schematic Layer of SFDS

The second layer of SFDS is the schematic layer. In this layer, the federated schema is defined. The federated schema is dependent on the underlying contexts that will share their data. It is proposed that the federated schema should be designed in such a way that it provides information sharing of a certain application. For example, if the purpose of a set of contexts is to share road and hydrology information, we design two federated schemas, and hence two semantic translators, one for hydrology and the other for road information. Figure 5 shows the reference model of SFDS, which is used to describe the underlying federated schema. In the sequel, the reference model and its integrity constraints are described, this is supported by an example of a federated schema attached to the reference model.

3.2.1 The Reference Model

The reference model of SFDS consists of a proxy context *Pcontext*, proxy hierarchy *Phierarchy*, proxy class *Pclass*, and a proxy attribute *Pattribute*. The relationship between these elements is an association relationship (or *member-of* relationship). A *Phierarchy* is a member-of only one *Pcontext*. A *Pcontext* can have more than one *Phierarchy*. *Pclass* is a member-of *Phierarchy*. A *Pclass* cannot be a member-of more than one *Phierarchy*, while a *Phierarchy* can have more than one *Pclass*. Similarly, *Pattribute*, which is a member of *Pclass*, belongs to only one *Pclass*.

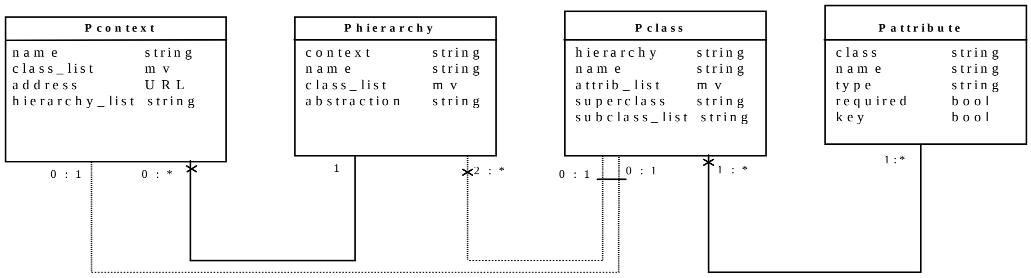

Figure 5: The reference model of the semantic translator.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

The notation shown in Figure 5 is based on the object-oriented system analysis model, OSA. The concepts of OSA are based on formal definitions of system data and behavior modeling [Embley et al., 1992]. The numbers shown near the connection of the objects are the cardinality constraints. These are non-negative integer numbers in the form *min:max*. The star designates an arbitrary non-negative number. The dotted line, shown in Figure 5, presents a special type of association. It represents a circumstantial association, as opposed to an essential association, which is represented as a continuous-solid line. A circumstantial association describes a relationship which depends on other conditions in the reference model. These conditions are not always satisfied. For example, *Pclass* is a member-of *Pcontext,* if it does not occur in any *Phierarchy.*

Let S be the federated schema which is described in the reference model.

The domain with respect to *S* is *Bool, string, mv, URL*
mv is a set of *strings,* and
URL= <prot>://<server>/<pathname>
Where *prot* is an Internet protocol : FTP, HTTP, SMTP, etc.
server is the IP address of the Internet server.
pathname is the location of the file name on the server

$f: D \rightarrow R$ is a bijective function with respect to *S*, where
$D = S \cup M \cup U \cup B$
S = {name, context, abstraction, hierarchy, superclass, subclass, class}
M = (attrib_list, class_list, hierarchy_list)
U = {address}
B= {type, key}

$R = R_1 \cup R_2 \cup R_3 \cup R_4$
$R = string \cup mv \cup URL \cup Bool$
$\forall s \in S : f(s) \in R_1$
$\forall m \in M : f(m) \in R_2$
$\forall u \in U : f(u) \in R_3$
$\forall b \in B : f(b) \in R_4$

The next constraints pertain to the relationship between the schema elements of the proxy context. We can establish the constraints that will be applied on hierarchies, classes and attributes of the federated schema.

The notation X.Y indicates a schema element X which has the attribute Y.

- ∀Pcontext.name (hierarchy-list | hierarchy-list = {Phierarchy.name})
- ∀Phierarchy.name (class-list | class-list ⊆ {class.name})
- ∀Pcontext.name (class-list | class-list = {class.name})

The above states that the set of Pcontext.hierarchy-list must contain all the elements of the set of *Phierarchy.name* at any state of the system. The list of classes in a hierarchy must be a proper set of the class names in the underlying proxy context. The class-list must contain all the class names in the underlying Pcontext.

Similarly we can state the constraints that apply to the attributes and their relationship with classes.

- ∀Pclass.name (attrib_list | attrib_list ⊆ {Pattribute.name})
- ∀Pattribute.name (Pclass | class ∈ {Pclass.name} ∧ name ∈ Pclass.attrib_list)
- ∀Pclass.name (hierarchy | hierarchy ∈ Pcontext.hierarchy_list).
- ∀Pclass.name (subclass | subclass ∈ Pcontext.class_list).
- ∀Pclass.name (superclass | superclass ∈ Pcontext.class_list).

4 EXAMPLE

Figure 6 shows the federated schema designed for sharing road information. Although the federated schema is not complete. Figure 7 shows the federated schema after it has been described in the reference model. The class *pavement* has two subclasses, *street* and *motorway.* The class *pavement* has a list of attributes (only the attribute *asphalt* and *speedlimit* are shown). The three classes, *pavements*, *street*, and *motorway* form the hierarchy *road*. The *road* hierarchy belongs to the proxy context *transportation*. The federated schema embedded in the reference model, forms the thematic description of the proxy context.

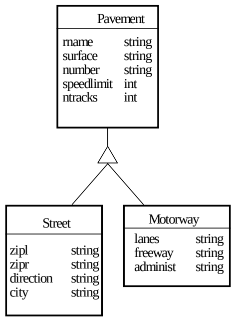

Figure 6: The federated schema for sharing road network

5 CONCLUSIONS

The paper introduced the semantic formal data structure, SFDS. It is believed that current technological and research development provide good foundation to build a rigorous model to achieve semantic interoperabilitiy. Although the prototype proved the applicability of the model, we still have a long way before we achieve semantic interoperability. A complete understanding of semantics and types of semantic differences is still lacking. Hence, it is required to analyze further case studies to investigate the types of semantic differences within and across different application domains.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

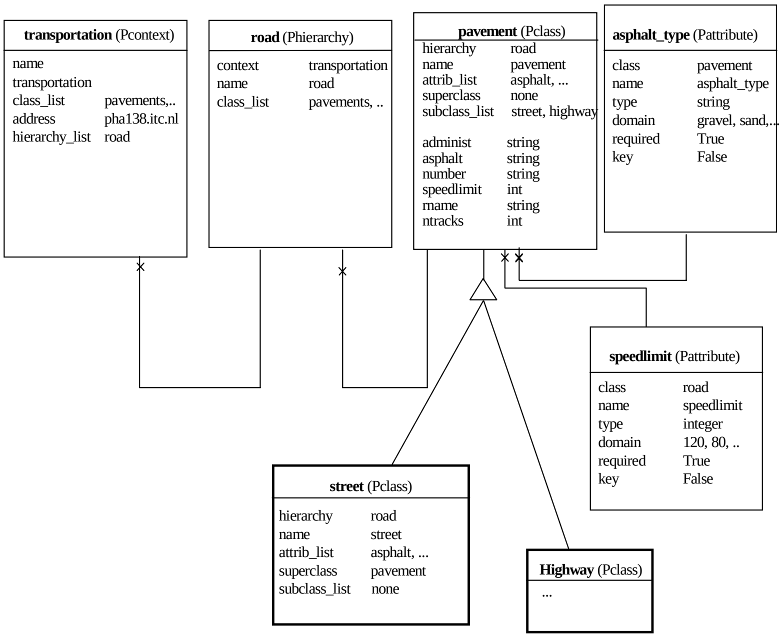

Figure 7: The federated schema described in the reference model.

6 REFERENCES

Guarino, N., 1997, "*Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration"*. Summer School on Information Extraction, Frascati, 14-19/97.

Bishr Y. 1998: " *Overcoming the semantic and other barriers to GIS interoperability*". The International Journal of GIS, No 3 Vol. 12, 1998

Bishr Y., and Radwan M., 1996: "*Design Of Federated Databases For Multi-Level Decision Making in watershed Management*". Workshop on the integration of GIS and remote sensing, ISPRS, WG II/2 1996, Milan, Italy.

Bishr Y., 1996: "*A Hierarchical Spatial Canonical Data Model - Towards Federating Heterogeneous GISs*". ISPRS 96 Vienna, Austria.

Bishr, Y, 1997: *"Semantic Aspects of Interoperable GIS"*. Ph.D. Thesis, 160 PP. ISBN 90 6164 141 1.

Molenaar, M. , 1996 (a). *"A Syntactic Approach for Handling the Semantics of Fuzzy Spatial Objects"*. In: Geographic Objects with Indeterminate Boundaries, P.A. Burroug and A.U. Frank (eds.), Taylor and Francis, London pp. 207-224.

Molenaar, M., 1996 (b) **."***The Role of Topologic and Hierarchical Spatial Object Models in Database Generalization".* In: Methods for the generalization of geo-database, M. Molenaar (ed.), Netherlands Geodetic Commission, New Series, Nr. 43, Delft, pp. 13-36.

Molenaar, M., 1989. *Single Valued Vector Maps – a Concept in GIS".* Geo-Informationssysteme, vol. 2, no. 1, 1989, pp. 18-26.

Molenaar, M., 1991. *"Status and Problems of Geographical Information Systems. The Necessity of a Geoinformation Theory"*. ISPRS Journal of Photogrammetry and Remote Sensing, 46 (1991). Elsevier Science Publishers B.V., Amsterdam, pp. 85-103.

Molenaar, M., 1994. *"A Syntax for the Representation of Fuzzy Spatial Objects"* In Molenaar, M. and S. de Hoop (eds.), *AGDM'94 Spatial Data Modelling and Query Languages for 2D and 3D Applications*, Publications on Geodesy - New Series, No. 40, pp. 155-169, Netherlands Geodetic Commission, Delft.

Molenaar, M., 1996 (c). "*An Introduction into the Theory of Topologic and Hierarchical Object Modeling in Geo-Information Systems*". Lecture Notes, Department of Land-Surveying & Remote Sensing, Wageningen Agricultural University, The Netherlands, 186 pages.

Molenaar, M., 1996 (d)*."Discrete Spatial Models for Fuzzy Geographical Objects"*. In: Progress in Industrial Mathematics at ECMI 94, H. Neunzert (ed.), Wiley and Teubner, New York, pp. 454-483.

Smith, J.M., and Smith, D.C.P., 1977. *"Database Abstractions: Aggregation and Generalization"*. ACM Transactions on Database Systems 2, June 1977, pp. 105-133.

Embley, D.W., Kurtz, B.D., and Woodfield, S.N., 1992. *"Object Oriented Systems Analysis - A Model Driven Approach"*. Yourdon Press Computing Series. Prentice-Hall, Inc., A Simon & Schuster Company, Englewood Cliffs, New Jersey 07632. 302 pages. ISBN 0 13 629973 3.