Paper presented at the STDBM'99 Workshop and published
in Proceedings International Workshop on Spatio-Temporal Database Management STDBM'99 (M. H. Böhlen, C. S. Jensen, and M. O. Scholl, eds.), pp. 60-78. Edinburgh, Scotland, UK, September, 10-11, 1999. Lecture Notes in Computer Science, Vol. 1678. Springer, Berlin.

Ontology-Based Geographic Data Set Integration

Harry T. Uitermark[1] , Peter J. M. van Oosterom[1] , Nicolaas J. I. Mars[2] , and Martien Molenaar[3]

[1] Kadaster (*Cadastre and Public Registry Agency*), Postbus 9046,
7300 GH Apeldoorn, The Netherlands.
{uitermark, oosterom}@kadaster.nl
[2] University of Twente, Enschede, The Netherlands.
mars@cs.utwente.nl
[3] International Institute for Aerospace Survey and Earth Sciences (ITC),
Enschede, The Netherlands.
molenaar@itc.nl

Abstract. In order to develop a system to propagate updates we investigate the semantic and spatial relationships between independently produced geographic data sets of the same region (data set integration). The goal of this system is to reduce operator intervention in update operations between corresponding (semantically similar) geographic object instances. Crucial for this reduction is certainty about the semantic similarity of different object representations. In this paper we explore a framework for ontology-based geographic data set integration, an ontology being a collection of shared concepts. Components of this formal approach are an ontology for topographic mapping (a domain ontology), an ontology for every geographic data set involved (the application ontologies), and abstraction rules (or capture criteria). Abstraction rules define at the class level the relationships between domain ontology and application ontology. Using these relationships, it is possible to locate semantic similarity at the object instance level with methods from computational geometry (like overlay operations). The components of the framework are formalized in the Prolog language, illustrated with a fictitious example, and tested on a practical example.

1 Introduction: Context, Related Work and Overview

Geographic Data Set Integration (or *Map Integration)* is the process of establishing *relationships* between *corresponding object instances* in different, autonomously produced, geographic data sets of a certain region [15]. The purpose of geographic data set integration is to share information between different geographic information sources. We are especially studying geographic data set integration in the context of *update propagation*, that is the *reuse* of updates from one geographic data set to another geographic data set ([21], [16], [22], [3]).

Geographic data set integration gets more and more attention nowadays since the digitizing of traditional map series has ended. In these map series, corresponding object instances were only linked implicitly by a common spatial reference system, e.g. the national grid. In order to make these relationships explicit geo-science researchers and computer scientists have developed various strategies. In the computer science domain, *schema integration* has been the dominant methodology for database integration [14]. That approach has been extended for geographic data sets [1]. Geo-scientists on the other hand have adopted methods from communication theory like *relational matching* [12]. In our case we adopted *ontologies* from the field of Artificial Intelligence [19]. The construction and use of ontologies for geographic data sets makes it possible to check the result of the geographic data set integration process for *inconsistencies*.

The organization of the paper is as follows. A framework for ontology-based geographic data set integration is presented in Section 2. The framework of Section 2 is represented in a formal manner with Prolog-statements in Section 3 (Prolog is a logic programming language; for references see [6]). The framework is the most important part of the paper, and to our best kwoledge, has not been presented before in literature. With a simple example the construction of domain and application ontology is illustrated in Section 4. There is a test on a practical example with real data in Section 5. Section 6 finishes with a discussion of the results and our conclusions.

We want to emphasize that this paper reports the exploration of ideas. While the applied geographic data sets are real we are not addressing the efficiency of the method nor its scalability. First we want to understand the principles of ontology-based integration.

Update propagation has many *temporal aspects*. However in this paper we concentrate on the linking aspect between different data sets. The notion of *synchronizing data sets* by using their temporal attributes is crucial for geographic data set integration. That issue together with update propagation is covered in earlier work ([16], [20]).

2 A Conceptual Framework for Ontology-Based Geographic Data Set Integration

Sharing and reusing data is a *communication* problem. Any successful communication requires a language which builds on a core of shared concepts [4]. An *ontology* is such a collection of shared concepts. Ontologies can be constructed for the conceptual dimensions of geographic objects, e.g. for geometry, topology, symbology of representations, and thematic contents [2]. In our research we emphasize the thematic contents, in particular in the field of *topographic mapping*. A *domain ontology* for topographic mapping will be introduced. A domain ontology must be supplemented with an *application ontology* for every geographic data set at hand. *Abstraction rules* define the relationships between the concepts of the domain ontology and the concepts of the application ontologies.

2.1 Ontologies

An ontology is a collection of shared concepts. More formally, the definition of an ontology in this research is “a structured, limitative collection of unambiguously defined concepts” [7]. This definition contains four elements:

1. An ontology is a collection of *concepts.*
2. The concepts are to be *unambiguously* defined.
3. The collection is *limitative*. Concepts not in the ontology cannot be used.
4. The collection has *structure*. Structure means that the ontology contains relationships between the concepts.

2.2 Domain Ontologies

An ontology for a certain discipline is called a *domain ontology*. This research uses data sets from the discipline of topographic mapping. In a domain ontology for such a discipline definitions of topographic objects, like *roads*, *railways*, and *buildings*, are given. As an example, the concept “road” is defined as “a leveled part for traffic on land”. In The Netherlands, a domain ontology for the discipline of topographic mapping is under construction (the **G**eo-Information **T**errain **M**odel: GTM; for details see [10]).

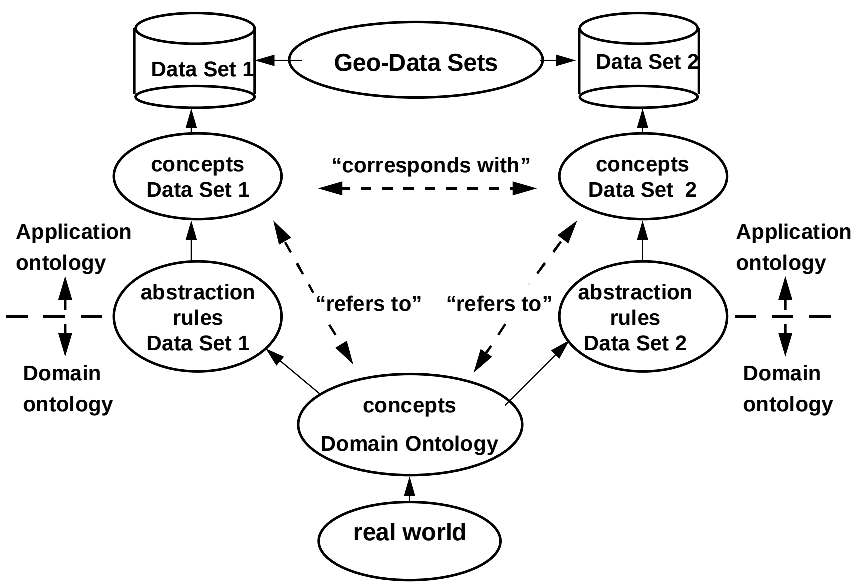

Fig. 1. An ontology-based framework for geographic data set integration.

2.3 Application Ontologies

A domain ontology (for topographic mapping, in our case) is the *first level* for the design of an ontology-based geographic data set integration framework. The *second level* concerns the actual geographic data sets. In these geographic data sets, names for mapped or surveyed concepts, such as “road” or “building” are used, but their precise meaning is not always the same as similar names for concepts in the domain ontology! That’s why we must make a distinction between concepts in the domain ontology and concepts used in the data sets, by constructing an *application ontology* for every data set involved in the integration process.

2.4 Abstraction Rules

Abstraction rules describe the transformation process from topographic objects (Real World objects) to geographic data set objects. So, abstraction rules define *what* topographic objects and *how* topographic objects are represented. Abstraction rules include:

- inclusion rules: what objects are selected (“capture criteria” in [9])
- representation rules: how objects are represented
- simplification rules: how objects are simplified
- aggregation rules: how objects are merged.

2.5 A Definition of Corresponding Object Instances

The abstraction rules define the relationships between the concepts in the application ontology and the concepts in the domain ontology. Concepts from *different* application ontologies are *semantically similar* if they refer to the *same* concepts (or *related*) concepts in the domain ontology (Fig. 1). C*orresponding object instances* can now defined as semantically similar and, in addition, *share the same location* (e.g. their geometry’s do overlap, or are near to each other). In the next section three types of semantic similarity will be introduced.

3 A Formal Expression of Ontologies in Prolog

3.1 Ontologies as Taxonomies

Ontologies in this research are structured like *taxonomies*. A taxonomy is like a tree with branches and leaves. It is a model for a hierarchy of classes, with concepts such as *sub*-classes and *super*-classes.

The basic taxonomy-structure is expressed and asserted as *Prolog facts* with the predicate name *taxon*:

```
taxon[SubClass, Class]
```

For example, *grassland* as a sub-class from terrain class TRN in the GTM domain ontology is expressed as:

```
taxon[grassland, trn]
```

A sub-class relationship *subClass* is recursively defined using the taxon-predicate in the following two *Prolog rules*:

```
subClass[X, X].
subClass[X, Z]:- (taxon[X, Y] && subClass[Y, Z]).
```
[1]

The first rule stops the recursion from the second rule that allows for sub-classes at any depth.

3.2 Semantic Relationships between Domain Ontology and Application Ontology

Relationships between the classes of the domain ontology and the classes of the application ontologies define the *semantics* of our universe of discourse. *Two relationships*, expressed as *Prolog clauses*, exist between concepts from domain ontology and application ontology:

- the *first* relationship concerns *equivalent* classes between domain ontology and application ontology:

```
referSToEquivalentClass[DomainClass, ApplClass].
```

- the *second* relationship relates (two or more) classes from the domain ontology that make up an aggregated (or composed) class in an application ontology:

```
refersToAggregateClass[DomainClass, ApplClass].
```

Note that these two relationships mean that the domain ontology should be rich enough to capture all the concepts of the application ontologies.

3.3 Semantic Relationships between Object Classes from Different Geographic Data Sets

With these two semantic relationships from the previous subsection *semantic relationships between object classes from different geographic data sets* are defined (after an idea in [13]):

1. Object classes from different application ontologies are *semantically equivalent* if they are equivalent with the *same class* in the domain ontology:

```
semanticEquivalentClass[ApplClass1, ApplClass2]:-
(refersToEquivalentClass[DomainClass, ApplClass1]
&&
 refersToEquivalentClass[DomainClass, ApplClass2]).
```

2. Object classes from different application ontologies are *semantically related* if they are equivalent with classes in the domain ontology, that are *sub-classes* or *super-classes* from each other:

```
semanticRelatedClass[ApplClass1, ApplClass2]:-
(refersToEquivalentClass[DomainClass1, ApplClass1]
```

[1] Prolog facts and rules are expressed in a syntax fashion compatible with a Prolog implementation in *Mathematica*® [23] made by R. E. Maeder [5].

```
&&
 refersToEquivalentClass[DomainClass2, ApplClass2])
&&
(subClass[DomainClass1, DomainClass2] ||
 subClass[DomainClass2, DomainClass1]).
```

which reads like: *semanticRelatedClass* ← *P1 & P2 & (P3 || P4)* where *Pi* stands for a precondition.

3. Object classes A and B from different application ontologies are *semantically relevant* if A is equivalent with a class in the domain ontology, that refers to an *aggregated class* in B, or vice versa:

```
semanticRelevantClass[ApplClass1, ApplClass2]:-
(refersToEquivalentClass[DomainClass, ApplClass1] ||
 refersToEquivalentClass[DomainClass, ApplClass2])
&&
(refersToAggregateClass[DomainClass, ApplClass2] ||
 refersToAggregateClass[DomainClass, ApplClass1]).
```

which reads like: *semanticRelevantClass* ← *(P1 || P2) & (P3 || P4)* where *Pi* stands for a precondition (this basic expression for *semanticRelevantClass* was extended to allow for cases where domain classes are sub-classes from each other).

In terms of *cartographic generalization* there is an analogy between semantic relatedness and *class driven* object generalization, or, semantic relevantness and *geometry driven* object generalization [8].

4 Demonstrating and Illustrating the Concepts of Ontology-Based Geographic Data Set Integration

The geographic data sets involved in this research are introduced in this section. With a simple example, highly schematic, and error free ("perfect data", with respect to semantics and geometric accuracy), the construction of domain and application ontology is illustrated.

4.1 Two Topographic Data Sets of The Netherlands

In this research, the geographic data set integration process is investigated between two different topographic data sets:

1. The first data set is a large-scale topographic data set (presentation scale 1 : 1,000), known as *GBKN*. It is usually produced by photogrammetric *stereo plotting* with field completion. It has an nation wide coverage of buildings, roads and waterways. The accuracy of the GBKN is stated in terms of *relative precision*: in built-up areas the relative precision between two well defined points should be better than $20\sqrt{2}cm$, and in rural areas better than $40\sqrt{2}cm$ [11]. The GBKN is updated continuously [20].

2. The second data set is a mid-scale topographic data set (presentation scale 1 : 10,000), known as *TOP10vector*. It is usually produced by photogrammetric *mono plotting* with field completion. It has a nation wide coverage of buildings, roads, waterways and terrain objects. The accuracy of TOP10vector is stated in terms of *absolute precision* in relation to the *national reference system*: the location of points should be better then *two meter*. TOP10vector is updated every four years [18].

The two data sets are produced by different organizations and are an accurate representation of the terrain. There is no displacement of representations for cartographic reasons. However, in the mid-scale map, some object representations (like buildings) are simplified and aggregated.

4.2 A Simple, Highly Schematic, and Error Free Example

After this introduction of the two geographic data sets involved a simple example will be given. The simple example refers to a Real World situation explained in the following subsection.

4.2.1 The Real World

The Real World of our example, depicted in Fig. 2, consists of:

- several buildings, labeled with class label BLD. *Main building* and *annex* are sub-classes from BLD
- some land parcels with different land use, labeled with TRN. *Grass land* and *arable land* are sub-classes from TRN
- a riding track between 4 and 7 meters wide, labeled with *conngt4m*, which is a sub-class from *riding tracks*, with on each side
- a verge. One verge is less or equal *6m* wide (labeled with *vergele6m*); the other verge is more then *6m* wide (*vergegt6m*)
- there are also ditches, labeled with WTR. *Road ditch* is sub-class from WTR.

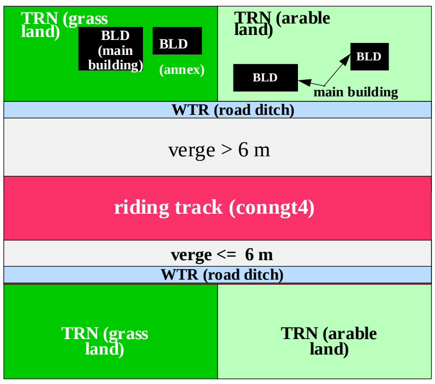

Fig. 2. The Real World of the example with labels from the GTM domain ontology.

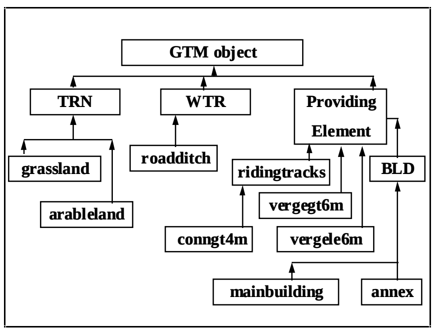

Fig. 3. The GTM domain ontology classification.

Of course there is more present in the Real World but all the objects that are not relevant for topographic mapping are filtered out. It is as if we wear a pair of spectacles with glasses where only objects of the domain ontology are passed through.

4.2.2 The Domain Ontology of the Real World

The domain ontology of the example is based on the GTM domain ontology in [10] and depicted in a class hierarchy in Fig. 3. Riding tracks, verges and buildings are all sub-classes from *Providing Element*, which is a super-class of geographic objects, describing the terrain in more detail.

Using the classification above a taxonomy is set up with facts like:

```
taxon[grassland, trn].
taxon[arableland, trn].
taxon[trn, gtmobject].
taxon[roadditch, wtr].
taxon[wtr, gtmobject].
taxon[conngt4m, ridingtracks].
taxon[ridingtracks, providingelement].
taxon[vergele6m, providingelement].
taxon[vergegt6m, providingelement].
taxon[mainbuilding, bld].
taxon[annex, bld].
taxon[bld, providingelement].
taxon[providingelement, gtmobject].
```

4.2.3 Topographic Data Set 1: an Application Ontology of the GBKN

For the GBKN application ontology of the example seven object classes are relevant:

- buildings with a (street) address (labeled *hoofdgebouw*)
- buildings without an address (*bijgebouw*)
- verges ≤ 6 meters wide (*bermsm6m*)
- verges > 6 meters wide (*bermbr6m*)
- road ditches (*bermsloot*)
- (paved or unpaved) road surfaces (*rijbaan*)
- terrain, or anything that can not be classified according the classes mentioned before (*terrein*).

We use Dutch labels for identification of the concepts. After all, the language of these labels is not essential for understanding the concepts. The concepts get their meaning in their actual relationship with the domain ontology! The GBKN abstraction rules state that six GBKN application ontology object classes are equivalent with domain ontology object classes:

```
refersToEquivalentClass[vergele6m, bermsm6m].
refersToEquivalentClass[vergegt6m, bermbr6m].
refersToEquivalentClass[ridingtracks, rijbaan].
refersToEquivalentClass[roadditch, bermsloot].
refersToEquivalentClass[mainbuilding, hoofdgebouw].
```

```
refersToEquivalentClass[annex, bijgebouw].
```

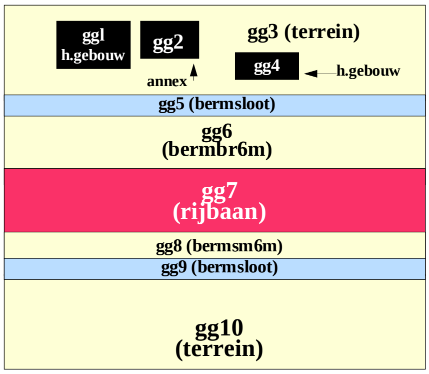

Fig. 4. The GBKN as an abstraction from the Real World (Fig. 2). gg1, gg2, ..., gg10 are object identifiers (oid's).

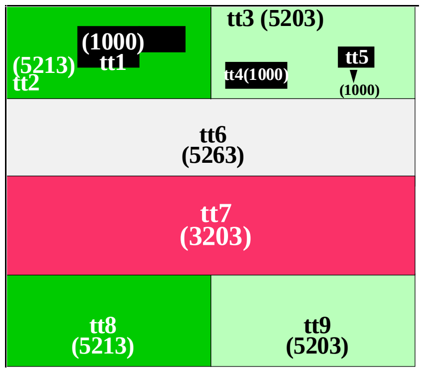

Fig. 5. The TOP10vector as an abstraction from the Real World (Fig. 2). tt1, tt2, ..., tt10 are object identifiers (oid's).

GBKN application ontology object class *terrein* is aggregated from two domain ontology classes *grassland* and *arableland*:

```
refersToAggregateClass[grassland, terrein].
refersToAggregateClass[arableland, terrein].
```

If the GBKN abstraction rules are applied to the Real World situation, then a map like Fig. 4 is produced. In Fig. 4 not all objects of the Real World in Fig. 2 are represented. The GBKN abstraction rules state that a *building object* will be represented in the GBKN if the building in the Real World:

- has an address, or
- is situated in urban area, and is accessible, or
- is situated in rural area, with an area > $20m^2$.

If the rightmost building in Fig. 2 is situated in rural area, with an area < $20m^2$, then it is not represented in Fig. 4.

Also, there is no distinction in different land use in the GBKN.

4.2.4 Topographic Data Set 2: an Application Ontology for TOP10vector

For the TOP10vector application ontology of the example five object classes are relevant:

- buildings (labeled *1000*)
- roads with a paved surface 4 to 7 meters wide (*3203*)
- arable land (*5203*)
- grass land (*5213*)
- land not classified in any other way (*5263*).

Here the labels are numbers (codes) that represent TOP10vector application ontology concepts. Also these concepts get their meaning in their actual relationship with the domain ontology concepts! The TOP10vector abstraction rules state that three TOP10vector application ontology object classes are equivalent with domain ontology object classes:

```
refersToEquivalentClass[grassland, 5213].
refersToEquivalentClass[arableland, 5203].
refersToEquivalentClass[bld, 1000].
```

TOP10vector application ontology object class *3203* is aggregated from domain ontology classes *vergele6m* and *conngt4m*:

```
refersToAggregateClass[vergele6m, 3203].
refersToAggregateClass[conngt4m, 3203].
```

Also, TOP10vector application ontology object class *5263* is aggregated from domain ontology object class *vergegt6m* (and other classes, not present in this example):

```
refersToAggregateClass[vergegt6m, 5263].
```

If the TOP10vector abstraction rules are applied to the Real World situation in Fig. 2 then a map like Fig. 5 is produced.

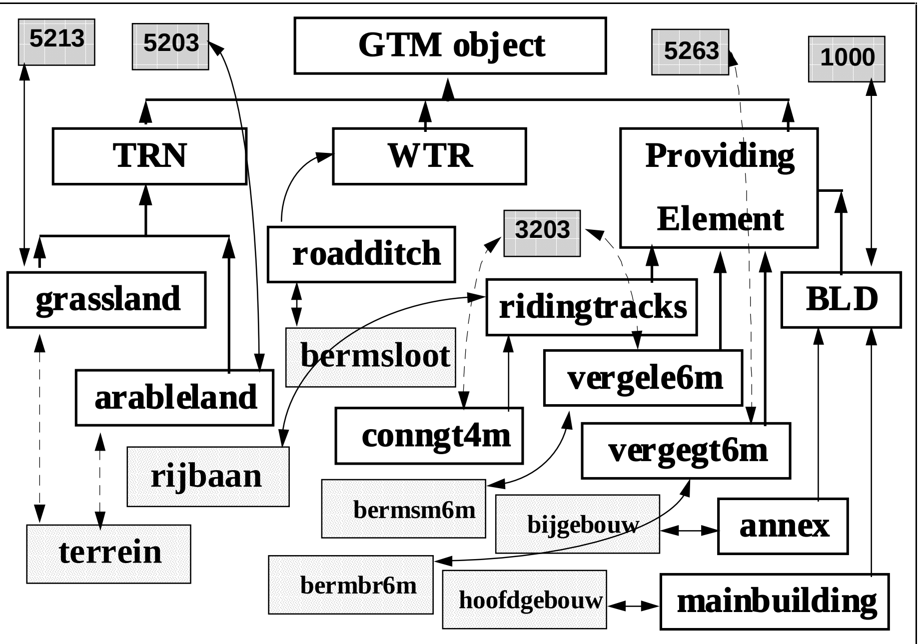

Fig. 6. The semantic relationships between domain ontology classes and application ontology classes.

Here also there is no 1-to-1 correspondence between building objects of the Real World and building objects represented in TOP10vector. The TOP10vector abstraction rules state that a building object is represented in TOP10vector if the building in the Real World:

- is situated in urban area, accessible, with area > 9m^2, or
- is situated in urban area, not accessible, with area > 50m^2, or
- is situated in rural area, with area > 9m^2.

If the last condition applies to the rightmost building in the Real World in Fig. 2 then it is represented in TOP10vector in Fig 5. Also, if *two (or more) buildings* in the terrain are *less than 2 meters* apart they are aggregated, as can be seen in Fig. 5. Furthermore, *ditches less than 6 meters wide* are represented as line objects in TOP10vector. So they do not appear in the partition of the TOP10vector map in Fig. 5. Note that *riding track between 4 and 7 meters wide* and *verge less than 6 meters wide* are aggregated to one road object (class *3203*).

4.2.5 Querying the Semantic Relationships

With the relationships defined and expressed between the classes of domain ontology and application ontologies we can ask questions about the semantic relationships between the object classes of the two application ontologies. See Fig. 6.

In Fig. 6 concepts are symbolized by:
- white rectangles: these are domain ontology concepts
- light shaded rectangles: these are GBKN application ontology concepts, and
- dark shaded rectangles: these are TOP10vector application ontology concepts.

The arrows in Fig. 6 represent the relationships between the concepts:
- a single headed arrow refers to a *domain sub-class* relationship
- a double headed solid arrow refers to a *equivalence* relationship
- a double headed dashed arrow refers to an *aggregate* relationship.

For example, (building) class *1000* in TOP10vector application ontology refers to (building) class *BLD* in domain ontology, while (building) class *hoofdgebouw* in GBKN application ontology refers to (building) class *mainbuilding* in domain ontology, which is a sub-class from the more general domain ontology class *BLD*. So both classes are semantically related:

In[ ]:=[2] **`Query[ semanticRelatedClass[hoofdgebouw, 1000]]`**

Out[ ]= `Yes`

Another example: (road) class *rijbaan* in GBKN refers to equivalent road class *ridingtracks* in domain ontology, while domain ontology (road) class *conngt4m* is aggregated in TOP10vector (road) class *3203*. So both classes are semantically relevant:

In[ ]:= **`Query[ semanticRelevantClass[rijbaan, 3203]]`**

Out[ ]:= `Yes`

[2] *In[ ]:=* and *Out[ ]=* are traditional *Mathematica* prompts.

4.2.6 Finding Corresponding Object Instances

The next task is how to find *corresponding object instances*, that means objects from semantically similar object classes from the different data sets involved, which in addition, share the same location (according to the definition in Section 2).

In order to determine if object instances share the same location we *overlay* both data sets (GBKN and TOP10vector) from our fictitious example, using their common spatial reference system, creating a new partition of faces. See Fig. 7.

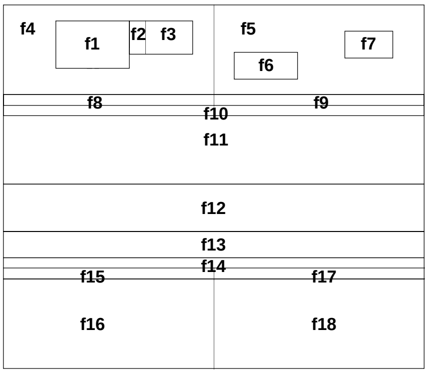

Fig. 7. The combination of GBKN (Fig. 4) and TOP10vector (Fig. 5).

Every face refers to exactly one GBKN object instance and exactly one TOP10-vector object instance (and every object instance refers to exactly one object class).

The procedure for finding corresponding object instances takes as input the list of faces of this combination of GBKN data set and TOP10vector data set (as in Fig. 7) and proceeds with querying the semantic similarity of the object classes from every face, according to the previous subsection, which results in Table 1.

Face-id	*GBKN-oid*	*GBKN-class*	*TOP-oid*	*TOP-class*	*Semantic similarity*
f1	gg1	hoofdgebouw	tt1	1000	related
f2	gg3	terrein	tt1	1000	incompatible
f3	gg2	bijgebouw	tt1	1000	related
f4	gg3	terrein	tt2	5213	relevant
f5	gg3	terrein	tt3	5203	relevant
f6	gg4	hoofdgebouw	tt4	1000	related
f7	gg3	terrein	tt5	1000	incompatible
f8	gg5	bermsloot	tt2	5213	incompatible
f9	gg5	bermsloot	tt3	5203	incompatible
f10	gg5	bermsloot	tt6	5263	incompatible
f11	gg6	bermbr6m	tt6	5263	relevant
f12	gg7	rijbaan	tt7	3203	relevant
f13	gg8	bermsm6m	tt7	3203	relevant
f14	gg9	bermsloot	tt7	3203	incompatible
f15	gg9	bermsloot	tt8	5213	incompatible
f16	gg10	terrein	tt8	5213	relevant
f17	gg9	bermsloot	tt9	5203	incompatible
f18	gg10	terrein	tt9	5203	relevant

Table 1. The faces from the example with the semantic similarity of their classes.

Semantically similar faces (equivalent, related, or relevant) are stored in a list; semantically incompatible faces in another. Faces in the first list are re-grouped using the object-id's from both GBKN object instances and TOP10vector object instances. The same happens to the second list after discarding object-id's that appear in the first list. What follows are two lists:

1. a list of corresponding object instances:
```
{
{{gg1, gg2}, {tt1}}, {{gg10}, {tt8, tt9}},
{{gg3}, {tt2, tt3}}, {{gg4}, {tt4}},
{{gg6}, {tt6}}, {{gg7, gg8}, {tt7}}
}
```
   In this list:
   - `gg1` and `gg2` are the buildings that is also represented by building `tt1`
   - terrain object `gg10` is similar to the aggregated terrain objects `tt8` and `tt9`
   - terrain object `gg3` is similar to the aggregated terrain objects `tt2` and `tt3`
   - `gg4` and `tt4` represent the same building
   - `gg6` and `tt6` represent the same terrain object
   - riding track `gg7` and verge `gg8` are similar to road object `tt7`
2. a list of object instances in both maps that have no correspondences at all:
```
{
{{gg5}, {}}, {{gg9}, {}}, {{}, {tt5}}
}
```
   In this list:
   - road ditches `gg5` and `gg9` are not represented in TOP10vector
   - building `tt5` has no counter object in GBKN.

4.2.7 Consistency Checking

Now that the relationships between semantically similar object instances are established, it is possible (and necessary) to check correspondences for *consistency*. Consistency, in this context, means: *in accordance with the abstraction rules of both data sets*. For example, capture criteria for buildings (as part of the abstraction rules) were formulated in subsections 4.2.3 and 4.2.4, but these capture criteria do not appear as *restrictions* (or *constraints*) in the *`refersToEquivalentClass`* or *`refersToAggregateClass`* relationships. Take, for example, the corresponding building object instances:

```
{{gg4}, {tt4}}
```

If GBKN object instance `gg4` has as attributes:

- address: True
- situated in: rural area
- accessible: True
- area category: $20m^2$ - $50m^2$

and TOP10vector object instance `tt4` has as attributes:

- situated in: rural area
- accessible: True
- area category: $20m^2$ - $50m^2$

then according to the abstraction rules for buildings there must be a representation in *both* GBKN and TOP10vector. So we may conclude that

```
{{gg4}, {tt4}}
```

is a 1-to-1 correspondence that is semantically related *and* consistent with the abstraction rules. In a similar way other correspondences can be checked (using procedures for checking areas or distances).

5 A Practical Test

In this section we test our geographic data set integration framework in the test area Zevenaar.

5.1 The Test Area Zevenaar

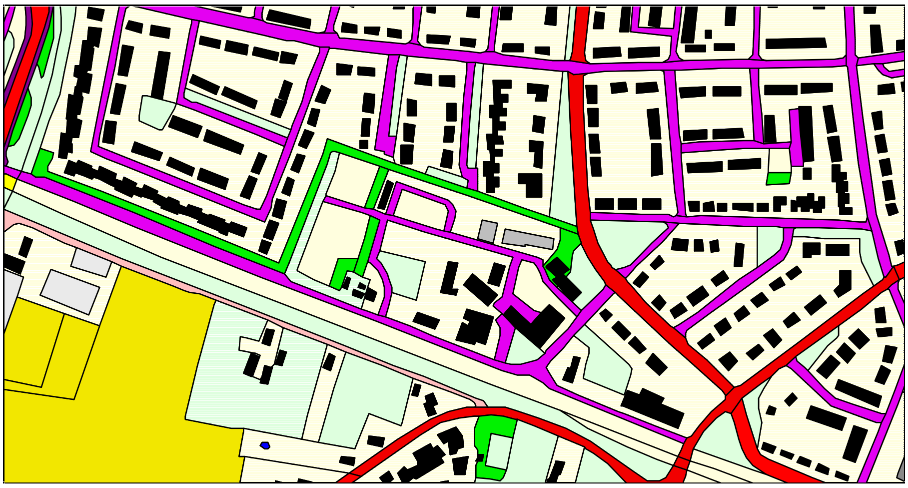

Fig. 8. The TOP10vector map of the test area Zevenaar.

The test area Zevenaar is a (mainly) built-up area (See Fig. 8). Its size is 0.3 km^2. The combination of GBKN and TOP10vector data sets is a partition of 1661 faces, with:

- 690 GBKN object instances, from nine object classes:
  - road ditches (labeled *bermsloot*: 6 instances)
  - large flowerbeds (*bloemenperk1*: 46 instances)
  - small flowerbeds (*bloemenperk2*: 24 instances)
  - buildings (*gebouw*: 450 instances)
  - parking strips (*parkeerstrook*: 27 instances)
  - road surfaces (*rijbaan*: 34 instances)
  - railways (*spoorbaan*: 1 instance)
  - side walks (*trottoir*: 71 instances)

  - terrain's, or anything that can not be classified according the classes mentioned before (*terrein*: 31 instances).
- 281 TOP10vector object instances, from twelve object classes:
  - buildings (labeled *1000*: 167 instances)
  - barns (*1050*: 2 instances)
  - green houses (*1073*: 3 instances)
  - roads with a paved surface less than 4 meters wide (*3103*: 4 instances)
  - roads with a paved surface 4 to 7 meters wide (*3203*: 1 instance)
  - roads with a paved surface more than 7 meters wide (*3303*: 1 instance)
  - streets (*3533*: 27 instances)
  - cycle tracks (*3603*: 3 instances)
  - leaf wood land (*5023*: 8 instances)
  - arable land (*5203*: 3 instances)
  - grass land (*5213*: 22 instances)
  - land not classified in any other way (*5263*: 40 instances).

The relationships between the object classes from both data sets, and the domain

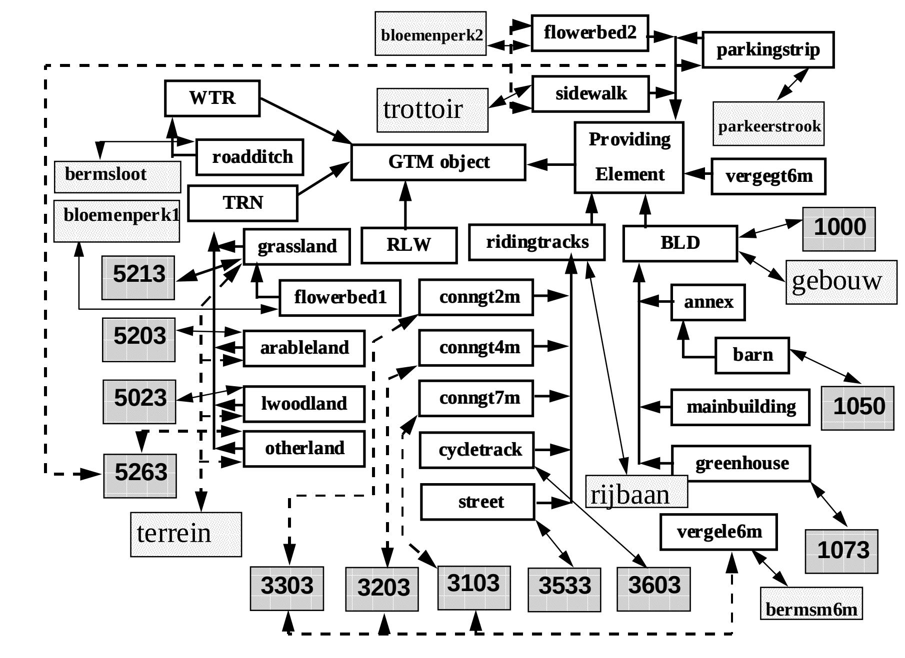

Fig. 9A. The relationships between the object classes from GBKN and TOP10vector data sets, and the domain ontology. See text for explanation.

ontology are represented in Fig. 9A.

Again, the rectangles represent the concepts:

- White rectangles: these are domain ontology concepts

- Light shaded rectangles: these are GBKN application ontology concepts, and
- Dark shaded rectangles: these are TOP10vector application ontology concepts.

The "tangle of lines" are the relationships between the concepts:

- a single headed arrow refers to a *domain sub-class* relationship, e.g. the Prolog fact
  ```
  taxon[parkingstrip, providingelement].
  ```
  expresses that domain ontology object class *parkingstrip* is a sub-class from domain ontology object class *providing element*.
- a double headed solid arrow refers to a *equivalence* relationship, e.g. the Prolog fact
  ```
  refersToEquivalentClass[parkingstrip, parkeerstrook].
  ```
  expresses that GBKN application ontology object class *parkeerstrook* is equivalent with the domain ontology object class *parkingstrip*.
- a double headed dashed arrow refers to an *aggregate* relationship, e.g. the four Prolog facts
  ```
  refersToAggregateClass[parkingstrip, 5263].
  refersToAggregateClass[sidewalk, 5263].
  refersToAggregateClass[flowerbed2, 5263].
  refersToAggregateClass[otherland, 5263].
  ```
  expresses that TOP10vector application ontology object class *5263* is aggregated from (= composed of) domain ontology object classes *parkingstrip*, *sidewalk*, *flowerbed2* and *otherland*.

If we query the classes of every face of the GBKN/TOP10vector partition for its semantic similarity, and apply the procedure for finding corresponding object instances then we get the following result:

- 204 correspondences, involving 483 GBKN instances, and 268 TOP10vector instances
- 207 GBKN instances and 13 TOP0vector instances that have no corresponding instance at all.

6 Discussion and Conclusion

In the previous section we did an experiment on ontology-based geographic data set integration with practical data from the test area Zevenaar (Fig. 8). In the next Section the result of this experiment will be evaluated.

6.1 Types of Errors

There are two main sources for errors:

1. The abstraction rules are not applied correctly, including classification errors, and up-to-date-ness
2. The different accuracy's of the geographic data sets involved.

The last error will influence the result, because the mechanism for relating corresponding object instances, relies on the assumption that semantically similar object instances do, at least partly, overlap.

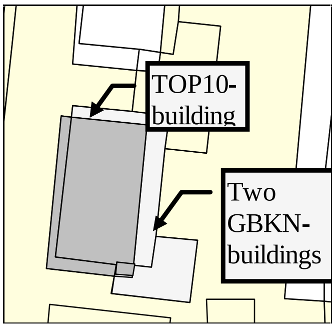

Fig. 9B. A (wrong) correspondence between one TOP10vector building (gray) and two GBKN buildings (light gray).

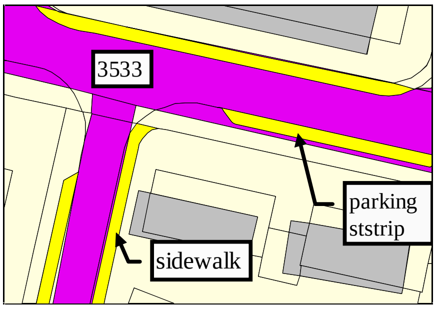

Fig. 10. A *sidewalk* outside (street) *3533* (= correct) and a *parkingstrip* inside *3533* (= wrong).

In Fig. 9B is a (wrong) correspondence between one TOP10vector building instance and two GBKN building instances because there is also an overlap between the TOP10vector building and the small GBKN building. The overlapping face is not filtered out because it is just above some threshold. This example emphasizes the importance of checking all found correspondences for this kind of error before starting the consistency checking, as mentioned is Section 4.

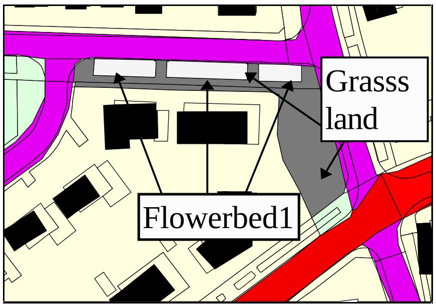

Fig. 11. Three GBKN flowerbed1 *(bloemenperk1)* objects (gray) correspond correctly with a TOP10vector grass land *(5213)* object (darker gray), but differ greatly in extension.

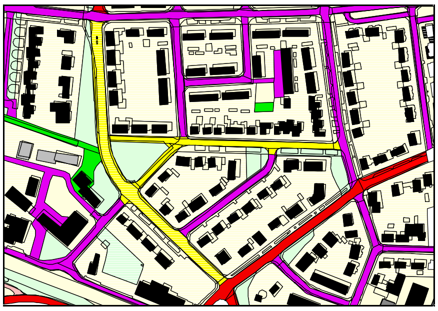

Fig. 12. A *5-to-3* type correspondence of road objects (light color): five GBKN riding tracks (*rijbaan*) objects, with three TOP10vector road objects (one *3103*, two *3533* objects).

If we look at the 220 instances that have no corresponding object, then 12 (or 5.5%) of them should have a corresponding object. See Fig. 10, where GBKN class parking strip (*parkeerstrook*) is overlapped by TOP10vector road class *3533*, which is not in accordance with the TOP10vector abstraction rules (Fig. 9A). The reason for this error maybe two-fold: the abstraction rules are not applied well for

TOP10vector object class *3533* or it is due to a road reconstruction, that means up-to-date-ness. We repeat here that both data sets should have the same time stamp, that means should be synchronized.

Another source for errors is when after synchronization both maps represent scenes that match partly or not at all. See Fig. 11.

Here also some kind of error checking should signal these mismatches.

Finally, some of the correct correspondences are from a complex *n-to-m* type, that is not very useful in pinpointing updates. See Fig. 12. Here five GBKN road objects correspond to three TOP10vector road objects. A solution for, at least, road networks is to fragment them first to homologous parts, like road segments and road junctions, as explained in [17].

6.2 Conclusion

Ontology-based geographic data set integration is a formal, yet simple and efficient, approach. With only a small set of rules, relationships are defined between concepts from the domain ontology and concepts from the application ontologies. Introduction of the abstraction rules gives, in principle, the possibility of consistency checking of the corresponding object instances. Future research will concentrate on this issue.

References

1. T. Devogele, C. Parent, and S. Spaccapietra. "On spatial database integration", *Int. J. Geographical Information Science*, Vol. 12, No. 4, pp. 335-352, 1998.
2. F. Hernández Rodríguez, G. Bravo Aranda, and A. Martín Navarro. *An ontology-based approach to spatial information modelling.* Proceedings 7th International symposium on Spatial Data Handling (SDH'96) (M. J. Kraak and M. Molenaar, eds.). Vol. II, pp. 12B.17-12B.29. Delft, The Netherlands, August 12-16, 1996. International Geographical Union.
3. C.-J. Kim. *Implementation of semantic translation for finding the corresponding geometric objects between topographic databases.* Master Thesis, International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede, The Netherlands, 1999.
4. W. Kuhn. Semantics of geographic information, Geoinfo-series, Vol.7. Vienna: Department of Geoinformation, Technical University pages, 1996.
5. R. E. Maeder. "Logic Programming I: The interpreter", *The Mathematica Journal*, Vol. 4, No. 1, pp. 53-63, 1994.
6. J. Malpas. PROLOG: a relational language and its applications. London: Prentice-Hall International, 465 pages, 1987.
7. N. J. I. Mars. *What is an ontology?* Proceedings Seminar on the impact of ontologies on reuse, interoperability and distributed processing (A. Goodall, ed.), pp. 9-19. London, November 7, 1995. Unicom, Uxbridge, Middlesex, UK.
8. M. Molenaar. An introduction to the theory of spatial object modelling for GIS. London: Taylor and Francis, 246 pages, 1998.
9. Open GIS Consortium Inc. "The OpenGIS Abstract Specification Model", Open GIS Consortium. Report No. 98-100. Wayland, Massachusetts, USA, 1998.

10. Ravi. "Geo-information terrain model. A Dutch standard for: Terms, definitions and general rules for the classification and coding of objects related to the earth's surface (NEN3610)", Ravi Netherlands Council for Geographic Information. Amersfoort, The Netherlands, 1995.
11. M. A. Salzmann. "On the modeling of geometric quality for large-scale mapping products", *Surveying and Land Information Systems*, Vol. 56, No. 3, pp. 149-155, 1996.
12. M. Sester, K.-H. Anders, and V. Walter. "Linking objects of different spatial data sets by integration and aggregation", *GeoInformatica*, Vol. 2, No. 4, pp. 335-357, 1998.
13. A. Sheth, and V. Kashyap. *So far (schematically) yet so near (semantically).* Proceedings IFIP (D. K. Hsiao, E. J. Neuhold, and R. Sacks-Davis, eds.), pp. 283-312, 1993. Elsevier Science Publishers B.V. (North-Holland).
14. S. Spaccapietra, C. Parent, and Y. Dupont. "Model independent assertions for integration of heterogeneous schemas", *VLDB Journal*, Vol. 1, pp. 81-126, 1992.
15. H. T. Uitermark. *The integration of geographic databases. Realising geodata interoperability through the hypermap metaphor and a mediator architecture.* Proceedings Second Joint European Conference & Exhibition on Geographical Information (JEC-GI'96) (M. Rumor, R. McMillan, and H. F. L. Ottens, eds.). Vol. I, pp. 92-95. Barcelona, Spain, March 27-29, 1996. IOS Press, Amsterdam.
16. H. T. Uitermark, P. J. M. van Oosterom, N. J. I. Mars, and M. Molenaar. *Propagating updates: finding corresponding objects in a multi-source environment.* Proceedings 8th International Symposium on Spatial Data Handling SDH'98 (T. K. Poiker and N. Chrisman, eds.), pp. 580-591. Vancouver, Canada, July 11-15, 1998. International Geographical Union.
17. H. T. Uitermark, A. B. M. Vogels, and P. J. M. van Oosterom. *Semantic and geometric aspects of integrating road networks.* Proceedings Second International Conference on Interoperating Geographic Information Systems (INTEROP'99) (A. Vckovski, K. E. Brassel, and H.-J. Schek, eds.), pp. 177-188. Zürich, Switzerland, March 10-12, 1999. Lecture Notes in Computer Science, Vol.1580. Springer, Berlin.
18. P. C. M. van Asperen. *Digital updates at the Dutch Topographic Service.* Proceedings XVIII ISPRS-Congress. Vienna, Austria, July 9-19, 1996.
19. P. E. van der Vet, and N. J. I. Mars. "Bottom-up construction of ontologies", *IEEE Transactions on knowledge and data engineering*, Vol. 10, No. 4, pp. 513-526, 1998.
20. P. J. M. van Oosterom. *Maintaining consistent topology including historical data in a large spatial database.* Proceedings Auto-Carto 13, pp. 327-336. Seattle, Washington, USA, 1997.
21. F. A. van Wijngaarden, J. D. van Putten, P. J. M. van Oosterom, and H. T. Uitermark. *Map Integration. Update propagation in a multi-source environment.* Proceedings 5th ACM Workshop on Advances in Geographic Information Systems ACM-GIS'97 (R. Laurini, ed.), pp. 71-76. Las Vegas, Nevada, USA, November 13-14, 1997. ACM, New York.
22. A. B. M. Vogels. *Propagatie van GBKN-wegenmutaties naar de TOP10vector (in Dutch).* Master Thesis, Geodesy Department, Technical University Delft, Delft, The Netherlands, 1999.
23. S. Wolfram. *Mathematica.* A system for doing mathematics by computer, 2nd ed. Reading, Massachusetts, USA: Addison-Wesley, 961 pages, 1991.