International Journal of Applied Earth Observation
and Geoinformation 7 (2005) 97–106

INTERNATIONAL JOURNAL OF
APPLIED EARTH OBSERVATION
AND GEOINFORMATION

www.elsevier.com/locate/jag

Ontology-based integration of topographic data sets☆

Harry T. Uitermark [a,*], Peter J.M. van Oosterom [b], Nicolaas J.I. Mars [c], Martien Molenaar [d]

[a] *Dutch Cadastre, Hofstraat 110, 7311KZ Apeldoorn, The Netherlands*
[b] *OTB Research Institute, Delft University of Technology, Delft, The Netherlands*
[c] *Materials Science Centre, University of Groningen, Groningen, The Netherlands*
[d] *International Institute for Geo-Information Science and Earth Observation (ITC), Enschede, The Netherlands*

Received 15 August 2003; accepted 4 March 2005

Abstract

The integration of topographic data sets is defined as the process of establishing relationships between corresponding object instances in different, autonomously produced, topographic data sets of the same geographic space. The problem of integrating topographic data sets is in finding these relationships, considering the differences in content and abstraction. A conceptual framework is developed. Components of this framework are ontologies and sets of surveying rules. New in this approach is the introduction of a reference model. A reference model belongs uniquely to the combination of topographic data sets to be integrated. The framework is tested on two topographic data sets with area instances (polygons) which have crisp and complete boundaries and are not displaced for cartographic reasons. The overall conclusion is that the ontology-based framework is feasible, if (1) there is (at least partial) knowledge of the surveying rules, and (2) the data sets can be synchronized in time. The application of this framework is most suitable for object classes with instances that are easy to identify and have a limited spatial extent (e.g., buildings).
© 2005 Elsevier B.V. All rights reserved.

Keywords: Ontologies; Interoperability; Data integration; Topographic data sets; Update propagation

1. Introduction

This article explores the notion of semantic interoperability in the context of the integration of topographic data sets. Semantic interoperability is here informally understood as a merging operation of topographic data sets, whereby similarities are detected. Likewise, what is not similar is also distinguished as such. Our experience of detecting a similarity or dissimilarity comes from the integration of topographic data sets, also known as map integration. The integration of topographic data sets is defined as the process of establishing relationships between corresponding instances in different, autonomously produced, topographic data sets of the same geographic space.

☆ A Dutch version of this article appeared in Geodesia 44 (2002).

* Corresponding author. Tel.: +31 55 5285506;
fax: +31 55 5285036.
E-mail address: harry.uitermark@kadaster.nl (H.T. Uitermark).

0303-2434/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.jag.2005.03.002

Traditionally, in existing map series, corresponding instances are linked implicitly by a common spatial reference system, for example the national grid (Devogele et al., 1996; Sester et al., 1998; Kilpeläinen, 2000). The integration of topographic data sets aims at making links between corresponding instances explicit by investigating the way data sets are acquired. Motivation and background of this research is update propagation, which is the application of updates, from one data set to another. Update propagation is studied within the realm of traditional map series (van Wijngaarden et al., 1997; Uitermark et al., 1998). A necessary condition for update propagation is the integration of data sets. While the integration of data sets is intimately related to update propagation, the integration of data sets has also a purpose of its own. If one data set is more specific in certain attributes, and another data set is more precise in its geometry, then the combination of this information in a third data set means 'the best of both worlds'. With this third data set, queries can be answered that cannot be answered by the two data sets separately ('what roads of class X have a precision of class Y?').

2. A conceptual framework for integration

This section provides a foundation for a conceptual framework for the integration of topographic data sets. In GIS applications (as well in other applications) the crucial characteristic of a piece of information is about the entities that it refers to. It is this referential meaning that needs to be made explicit and organized (Guarino, 1997).

The key issue in the integration of data sets is finding corresponding instances. This process of semantic matching is only possible if the meaning of objects is clear. Central to a conceptual framework for integration is a mechanism that makes object definitions clear; that means, makes data sets semantically transparent to each other. In that respect the integration of data sets can be seen as a communication problem. Any successful communication requires a language that builds on a core of shared concepts (Kuhn, 2001).

It is here that an ontology plays a fundamental role (Fonseca et al., 2002; Pundt and Bishr, 2002). The concept and definition of an ontology will be explained in Section 2.1. A domain ontology is an ontology with concepts from a certain discipline. A domain ontology is supplemented with application ontologies, for each and every topographic data set to be integrated.

Surveying rules are rules that govern the transformation process from the actual observed real-world objects into instances of the data set (Section 2.2).

Concepts from a domain ontology and information from surveying rules are used for the construction of a reference model. A reference model is a subset of concepts from a domain ontology, with additional structure, uniquely belonging to the combination of the data sets to be integrated (Section 2.3). Relationships between reference model concepts and application ontology concepts define the semantics of a data set (Section 2.4).

2.1. Definition of an ontology

The notion and use of an ontology is relatively young, although the term 'ontology' has a long history in philosophical tradition in conceiving ontology as the science that deals with the nature and organization of reality. However, in computer science, an ontology has to do with the explication of knowledge to overcome the problem of semantic diversity of different information sources (Visser et al., 2002). Ontology is defined here as 'a structured, limitative collection of unambiguously defined concepts' (Mars, 1995).

In this research an ontology for a certain discipline is called a domain ontology. The data sets studied here are from the discipline of topographic mapping. In a domain ontology for topographic mapping, definitions for topographic concepts are supplied, such as 'road', 'railway', or 'building'.

An ontology for a specific data set is called an application ontology. In data sets, labels for surveyed concepts are used, such as 'road', or 'building', but their precise meaning is not necessarily the same as similar names for concepts in the domain ontology. That is why we must make a strict distinction between concepts in the domain ontology, and concepts in the application ontologies. This distinction addresses naming diversity, like homonyms (same name used for different concepts), or synonyms (different names used for the same concept).

2.2. Surveying rules

People employ the concept of classes when they think of and communicate about real-world phenomena. Out of the many classes of real-world phenomena, only classes, relevant to a certain discipline, which can be identified and labeled, are included as concepts in a domain ontology. With this collection of classes we look at the real-world: it is as if we wear a pair of glasses, where only instances of classes of the domain ontology are passed through (Frank, 2001). From this filtered collection of real-world objects, only those relevant for a certain application, are captured into a data set. Surveying rules are rules that govern the transformation process from the actual observed real-world objects into instances of the data set.

2.3. Reference model and semantic relationships

In order to integrate different data sets a reference model is constructed. A reference model is a subset of refined domain ontology concepts, with additional structure, uniquely belonging to the combination of data sets to be integrated. Two well-known abstraction mechanisms are fundamental for this additional structure in the reference model:

1. There is a generalization/specialization classification, which means that classes are grouped into a taxonomy with superclasses and subclasses.
2. There is a composite/component classification, which means that classes are grouped into a partonomy, with composite classes and component classes.

Therefore, two basic relationships are defined within the reference model. Assume a reference model A with its finite set of class labels: $a1$, $a2$, $a3$, etc.

Definition 1a. The basic taxonomy relationship within a reference model, abbreviated as *taxon*, is between a subclass $a1$ and its superclass $a2$:

taxon($a1$, $a2$)

Definition 1b. The basic partonomy relationship within a reference model, abbreviated as *parton*, is between a component class $a3$ and its composite class $a4$:

parton($a3$, $a4$)

Relationships between reference model classes and application ontology classes, define the *semantics* of a data set. Assume a data set B with b as one of its classes.

Definition 2. The basic semantic relationship, abbreviated as *refers_to*, is between a reference model class a, and an application ontology class b:

refers_to(a, b)

Within a reference model, we assume a partonomy:

1. With a one level deep composite/component structure.
2. At least one component class is a constituent to a composite class. Multiplicity of instances – 0, 1, or more – of component and composite classes depends on contents and abstraction.[1]
3. Component classes are *non-exclusive*, therefore, they can be shared by other composite classes.

Furthermore, we impose a *many-to-one* integrity constraint on the *refers_to* relationship: for a given reference model class a, there is at most one application class that satisfies the relationship, but for a given application class b, there may be more reference model classes $a1$, $a2$, etc., satisfying the relationship. The motivation for this constraint is that a reference model should be *finely grained* enough to express every semantic similarity between reference model concepts and application ontology concepts.

2.4. Relationships between different application ontologies

Based on the *refers_to* relationship, we define relationships between classes from different application ontologies. These relationships determine the *semantics* of our universe of discourse in the integration of data sets. Assume two data sets B

[1] At the class level are defined the object classes involved. The actual part-of relation happens at the instance level.

and C, with class label sets B and C, respectively, and their reference model A, with class label set A. Let b be a class label from B, c a class label from C, and $a1$, $a2$, $a3$, etc. class labels from A. Then the following *four* types of relationships are defined.

Definition 3a. There is a *equivalent class relationship* (*Sequi*) between classes b, and c, if there exists a class $a1$, such that class $a1$ refers to *both* classes b and c:

$$Sequi = \{(b,c) \in \mathrm{B} \times \mathrm{C} | \exists a1 \in \mathrm{A} \ni refers\_to\ (a1,b) \wedge refers\_to\ (a1,c)\}$$

where $\ni$ denotes 'such that'.

Definition 3b. There is a *subclass–superclass relationship* (*Srla*) between class b, and class c, if there exist classes $a1$ and $a2$, such that class $a1$ refers to class b, and class $a2$ refers to c, with $a1$ a *subclass* of $a2$:

$$Srla = \{(b,c) \in \mathrm{B} \times \mathrm{C} | \exists a1, a2 \in \mathrm{A} \ni refers\_to\ (a1,b) \wedge refers\_to\ (a2,c) \wedge taxon\ (a1,a2)\}$$

Definition 3c. There is an *explicit composite class–component class relationship* (*Srle*1) between class b, and class c, if there exist classes $a1$ and $a2$, such that class $a1$ refers to class b, and class $a2$ refers to class c, with class $a1$ a *component class* of class $a2$:

$$Srle1 = \{(b,c) \in \mathrm{B} \times \mathrm{C} | \exists a1, a2 \in \mathrm{A} \ni refers\_to\ (a1,b) \wedge refers\_to\ (a2,c) \wedge\ parton\ (a1,a2)\}$$

Definition 3d. There is a *implicit composite class–component class relationship* (*Srle*2) between class b, and class c, if there exist classes $a1$, $a2$, and $a3$, such that class $a2$ refers to class b, and class $a3$ refers to class c, with class $a1$ a component class of both classes $a2$ and $a3$:

$$Srle2 = \{(b,c) \in \mathrm{B} \times \mathrm{C} | \exists a1, a2, a3 \in \mathrm{A} \ni refers\_to\ (a2,b) \wedge refers\_to\ (a3,c) \wedge\ parton\ (a1,a2) \wedge\ parton\ (a1,a3)\}$$

If an element $(b, c) \in \mathrm{B} \times \mathrm{C}$ belongs to the types of relationships in Definitions 3a–3d than we speak of semantically similar classes. Otherwise classes are incompatible (for more details, see Uitermark, 2001).

3. Defining data set integration

Domain ontology, application ontologies, surveying rules, a reference model, semantic and geometric relationships are the fundamental building blocks for a conceptual framework for ontology-based integration of topographic data sets (Fig. 1). With these building blocks data set integration is defined, including corresponding classes and corresponding instances—the latter being the ultimate goal of the integration of data sets (Section 3.1). Important parts in the definition of corresponding instances are location (Section 3.2), consistency (Section 3.3) and synchronization of data sets (Section 3.4).

3.1. A definition of topographic data set integration

Definition 4a. *The integration of topographic data sets* is the process of establishing explicit relationships between *corresponding instances* in different, autonomously produced, topographic data sets of the same geographic space.

To define 'corresponding instances' we need to define 'corresponding classes':

Definition 4b. *Corresponding classes* are classes from different application ontologies, which have an 'equivalent', 'subclass–superclass', or an implicit/explicit 'composite class–component class' relationship.

Definition 4c. 'Corresponding instances' are instances:

1. from corresponding classes,
2. sharing a similar *location*,
3. *consistent* with the surveying rules, and
4. synchronized in *time*.

3.2. Location and the integration of topograhic data sets

'Location' refers to the geometry of an instance in a topographic data set. In this research, 'sharing a similar location' in Definition 4c, is made operational by the overlap between different polygons. The

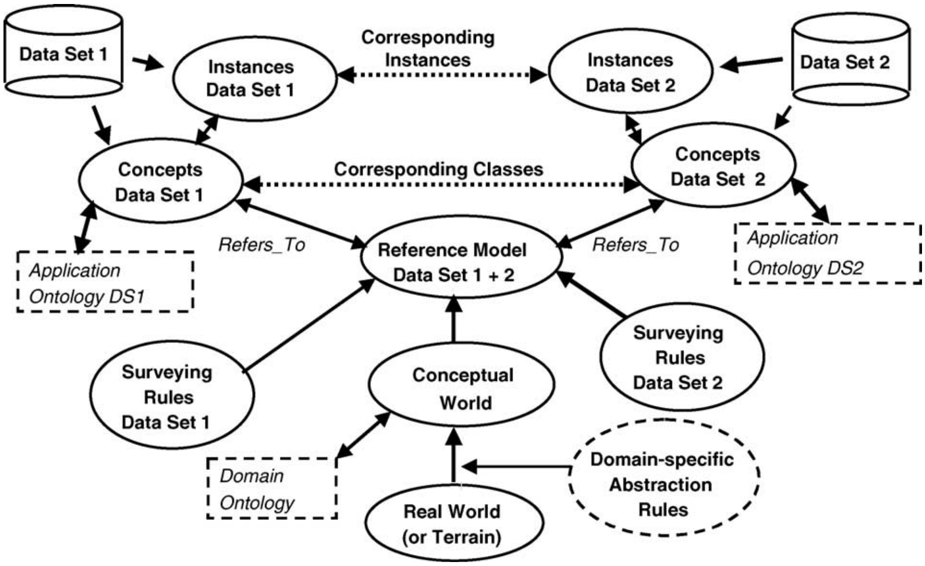

Fig. 1. An ontology-based framework for the integration of topographic data sets.

justification for choosing overlap as 'similar location' lies in the precision and accuracy of topographic data sets. Detecting overlap between instances of different data sets is done by an overlay operation (van Oosterom, 1994). Note that choosing overlap for 'similar location' removes in a certain sense stochasticity from data sets. Any amount of overlap is now sufficient to declare semantically similar instances as candidates for corresponding instances. This removal of stochasticity is only temporarily. Stochasticity is introduced again in consistency checking.

3.3. Consistency checking

The notion '*consistent with the surveying rules*' in Definition 4c is fundamental in this research, because inconsistencies should be resolved before update propagation can happen. 'Consistency' implies possible real-world situations that are correctly represented by corresponding instances. Or more formally, if data sets are consistent, we cannot refute possible real-world situations, represented by the data sets.

The reference model, and the overlay operation, detect candidates for corresponding instances. Then, in order to decide if candidates are consistent, we have to take additional conditions from surveying rules into account. Therefore, 'consistency checking' is defined in terms of reference model classes, overlap, and additional conditions from surveying rules.

Additional conditions are expressions with thematic, geometric, or topological attributes. For example, 'situated in urban region' is a thematic attribute, 'area size ≥ 9 m^2' is a geometric attribute, and 'adjacent to a road' is a topological attribute. Additional conditions imply additional criteria, whether a real-world object should be considered as member for a certain class, its class *intension*. After this decision – the actual application of surveying rules – the *extension* of a class is the set of all its members (Molenaar, 1998). Consistency checking can be made operational by testing whether different extensions of candidates satisfy *both* intensions.

However, candidates come in two types, simple or complex:

1. A *simple* candidate correspondence consists of a *pair* of corresponding instances, a *1 to 1* correspondence.
2. A *complex* candidate correspondence is an *n to m* correspondence.

Usually, simple correspondences are from 'equivalent' relationships, or 'subclass–superclass' relationships, and complex correspondences are from 'composite class–component class' relationships. Simple candidates need a minimum effort in consistency checking, but complex candidates need some mechanism to break down *n to m* correspondences into several *1 to 1* correspondences, a subject for future research.

3.4. The synchronization of data sets

The data sets to be integrated are from the same geographic space. Furthermore, they should also represent the same 'snapshot' in time. Therefore, data sets must have 'history', that means a mechanism to roll back to a common moment in time, in order to get synchronized. Every instance should have a pair of attributes, *tmin* and *tmax*. An instance is valid starting from (and including) *tmin* and remains valid until (and excluding) *tmax*. Current instances get a special value *max_time* for *tmax*, where *max_time* is larger than any other time value. When an instance changes then *tmax* is set to current time, and a new version of this instance gets *tmin* as current time. In this way one is able to reconstruct a given situation at any point in history (van Oosterom, 1997).

4. An experiment

The conceptual framework for integration is tested on existing topographic data sets. The procedure for testing is as follows:

(a) Starting point are the data sets and their surveying rules. Two topographic data sets (GBKN and TOP10vector) are introduced in Section 4.1.
(b) A common subset of domain ontology concepts is chosen, and surveying rules of both data sets are made explicit. Furthermore, both data sets are compared in detail at the instance level in order to find resemblances and differences, which are not entirely explained by surveying rules (Section 4.2).
(c) From the previous information, a reference model is constructed. Special attention is given to certain real-world situations relevant in the construction of the reference model (Section 4.3).

A guideline for the construction of a reference model is presented in Section 4.4. The procedure is demonstrated for domain class 'Building'. For other domain classes, like 'Road' or 'Railway' (see Uitermark, 2001).

Table 1
GBKN class Building with labels and descriptions

Class label	Description
Hoofdgebouw	Mainbuilding ('Building' with one or more postal addresses)
Bijgebouw	Annex ('Building' without address)

4.1. The data sets

The integration process is investigated between *area classes* of two topographic data sets, GBKN and TOP10vector:

1. GBKN data set is a Dutch large-scale data set (presentation scale 1:1000). It is usually produced by photogrammetric *stereo plotting* with field completion. It is a nation-wide coverage of buildings, roads, railways and waterways (for 'Building' see Table 1).
2. TOP10vector data set is a Dutch medium-scale data set (presentation scale 1:10,000). It is usually produced by photogrammetric *mono plotting* with field completion. It is a nation-wide coverage of buildings, roads, railways, waterways and land use (for 'Building' see Table 2).

4.2. Refining domain class building with surveying rules

Domain ontology class 'Building' is defined in Table 3.

For GBKN, according to Table 1, we need reference model classes for:

- *hoofdgebouw*, defined as 'Building' with *one or more* addresses, and

Table 2
TOP10vector class Building with labels and descriptions

Class label	Description
1000	Mainbuilding or Annex
1050	Barn
1073	Greenhouse

Table 3
Domain ontology class 'Building' and its definition

Class label	Domain ontology concept definition
Building	Free standing covered area, partly or completely enclosed by walls, allowing access by people, and directly or indirectly connected to the terrain

- *bijgebouw*, defined as 'Building' *without* address.

Therefore, domain ontology class 'Building' is divided into two reference model subclasses:

1. 'mainbuilding', defined as 'Building' with one or more addresses and
2. 'annex', defined as 'Building' without address.

For TOP10vector, according to Table 2, we need reference model classes for:

- *1000*, defined as 'mainbuilding' or 'annex'
- *1050*, defined as 'annex' with a roof on poles, with not more than one wall, and
- *1073*, defined as 'annex' mainly made of glass.

TOP10vector class *1000* is a union of classes 'mainbuilding' and 'annex'. However, comparing surveying rules, according to Fig. 2, an 'annex' adjacent to 'mainbuilding' is not acquired in TOP10vector. Therefore, class 'annex' is divided into two reference model subclasses:

1. 'adjacent annex', defined as 'Building' without address, connected with 'mainbuilding'.
2. 'free standing annex', defined as 'Building' without address, not connected with 'mainbuilding'.

Note that we are obliged by the previous distinction – 'adjacent annex' and 'freestanding annex' – to introduce this distinction in GBKN class *bijgebouw*. This adding of semantics to class *bijgebouw* is done by a test of adjacency with class *hoofdgebouw*.

Both TOP10vector class *1050* ('barn') and TOP10-vector class *1073* ('greenhouse') are 'free standing annexes'. Consequently, 'free standing annex' is divided into three reference model subclasses:

1. 'barn', defined as 'free standing annex', with a roof on poles, with not more than one wall.
2. 'greenhouse', defined as 'free standing annex', mainly made of glass.
3. 'remaining free standing annex', defined as 'free standing annex', neither 'barn' nor 'greenhouse'.

Furthermore, comparing GBKN and TOP10vector data sets reveals with regard to 'Building' that, if two or more individuals of 'Building' in the terrain are nearby each other, they are acquired in combination, and represented as a *single* TOP10vector object instance (Fig. 3). Indeed, according to TOP10vector surveying rules, individuals of 'Building' are acquired in combination, if their distance is <2 m, except for a 'ditch' or 'footpath' between them. Note that this situation causes a complex 2 to 1 correspondence.

'Mainbuilding', 'adjacent annex', and 'free standing annex' (with 'barn', 'greenhouse', and 'remaining free standing annex' as subclasses) become reference model classes. An overview of reference model classes is presented in Table 4, a common *universe of discourse* for GBKN and TOP10vector with respect to 'Building'.

4.3. Special situations in the construction of a reference model

After the refinement of a domain ontology class we have to construct a reference model with the

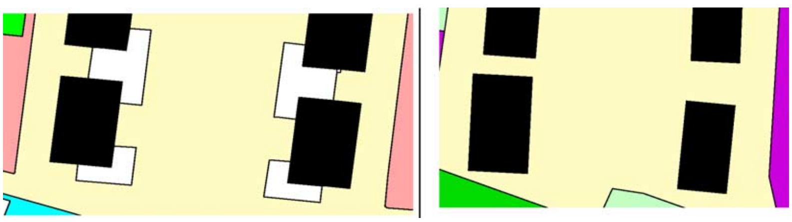

Fig. 2. 'Annex' (white), adjacent to 'mainbuilding' (black) is not acquired for TOP10vector class *1000* (black), and 'disappears' in class *5263* (pink-yellow).

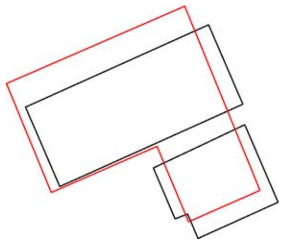

Fig. 3. Two 'Buildings', less than two meters apart ('mainbuilding', and 'free standing annex', in black), are combined into a single TOP10vector class *1000* instance (in red).

relationships defined in Section 2.4. There are three situations that require special attention:

1. Classes, exclusively acquired for only one of the data sets. For example, a building subclass is not acquired for one of the data sets because its area size is smaller than a certain limit. Apparently this class does not exist for the other data set: during its acquisition it was substituted by surrounding, adjacent classes. In reference model constructing, an analogy with component classes is used. Component classes 'disappear' in composite classes. Therefore, if a class in one data set is not acquired for the other data set, then this class is modeled as a component class of 'surrounding' classes.
2. Instances of domain classes that are acquired in *parts*. For example, a part of an instance of class 'sidewalk' is combined with an instance of class 'road', and another part of an instance of class 'sidewalk' is combined with a 'land use' class. Thus, parts of domain classes are transformed within one data set into different application classes. The solution for this situation is to model such a domain class as a *shared* component class of both application classes involved.
3. Domain classes, which are not represented in both data sets. For example, a class 'verge' is in one data set combined with an (adjacent) class 'road' into a composite road class, and in another data set combined with an (adjacent) class 'grass land' into a composite land use class. Hence, 'verge' will not be represented as an independent class in neither data set. Here, the solution for this situation is also to model 'verge' as a shared component class of both the composite road class and the composite land use class.

4.4. A guiding principle for constructing a reference model

In Table 4, we have the refined subclasses for domain class 'Building'. After the activities in the previous subsections there is an indication, which classes can be seen as 'subclass–superclass', or 'component class - composite class' to each other, i.e. what *role* an application class has with respect to an application class in another data set. For every role we need a *distinct* reference model class.

To facilitate the construction of the reference model, having a taxonomy subgraph and a partonomy subgraph, a *guiding principle* is presented:

1. Determine for every application class its role in a semantic similarity. If its role is in:
   a. an equivalent relationship, then identify its reference model class, and put it in the taxonomy subgraph;
   b. a 'subclass–superclass' relationship, then identify its reference model classes, create if

Table 4
Domain ontology class 'Building' and its refinements into reference model subclasses for GBKN and TOP10vector

Domain ontology class	Refined subclass in reference model	Definition of refined subclass in reference model
Building	Mainbuilding	'Building' with one or more addresses
	Adjacent annex	'Building' without address connected with 'Mainbuilding'
	Free standing annex	'Building' without address not connected with 'Mainbuilding'
	Barn	'Free standing annex' with a roof on poles with not more than one wall
	Greenhouse	'Free standing annex' mainly made of glass
	Remaining free standing annex	'Free standing annex' neither 'Barn' nor 'Greenhouse'

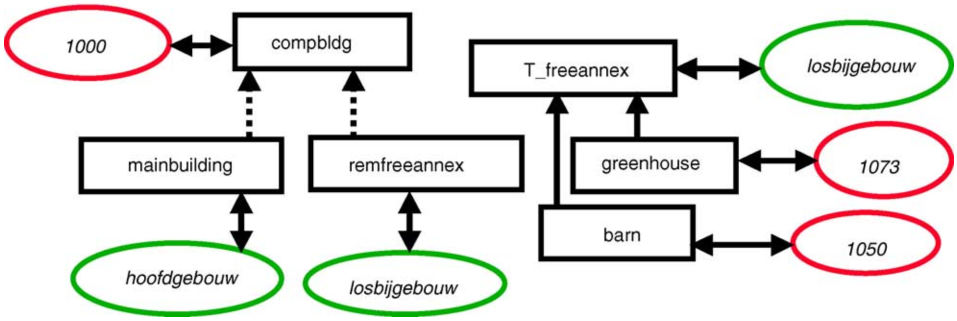

Fig. 4. Reference model constructs for 'Building': (red ovals) TOP10vector labels; (green ovals) GBKN labels; (black boxes) reference model labels; (solid arrow) *taxon* relationship; (dashed arrow) *parton* relationship; (double arrow) *refers_to* relationship.

necessary a *distinct* reference model *superclass*, and put it in the taxonomy subgraph;

c. a 'composite class–component class' relationship, then identify its reference model class or classes, create if necessary a *distinct* reference model *composite class*, and put it in the partonomy subgraph.

2. Determine for every application class its *refers_to* relationship with classes from the reference model.

For an example, see Fig. 4. On the right-hand side is GBKN class *losbijgebouw* in its role as a distinct 'free standing annex' superclass, labeled 'T_freeannex', for TOP10vector classes *1073* and *1050* (rule 1b of the guideline). On the left-hand side is GBKN class *losbijgebouw* in its role as component class of a distinct 'mainbuilding or annex' composite class 'compbldg' (=composite building) for TOP10vector class *1000* (rule 1c of the guideline; Fig. 3).

Now this 'guideline' seems linear, but it is not. It is cyclic, and iterative. The idea is to design a structure that is semantically rich and finely grained enough, to express every semantic similarity between data sets. Central in this 'guideline' is the concept of *role*. A role is what a data set class is in confrontation with another data set class: this can be an equivalent class, a subclass, a superclass, a component class, or a composite class.

A data set class may have different roles. The question if all roles should be modeled depends on their occurrence. If a role is very rare (e.g. a very small 'mainbuilding', i.e. a transformer station, that will be acquired for GBKN, not for TOP10vector), then it can be treated as an exception (a singleton, see Section 4.5), because if modeled it would obscure the clarity of the reference model.

4.5. Evaluation experimental results

The reference model is applied to sample data sets from GBKN and TOP10vector:

1. The GBKN data set has 694 instances, from 12 classes.
2. The TOP10vector data set has 295 instances, from 13 classes.

Candidates for corresponding instances are detected, and subsequently checked for consistency with surveying rules. Many candidates are of a complex type, i.e. groups, or clusters, of instances corresponding to each other (Section 3.3).

When candidates are inspected, it is concluded that:

- 198 candidates are consistent and
- seven candidates are inconsistent.

Inconsistency of candidates is caused by surveying rule errors, which are detected accordingly.

Twenty-eight instances do not participate in a correspondence. These are called *singletons*. If all the roles between classes of different data sets have been modeled completely and correctly then a singleton indicate possibly two types of errors:

- Surveying rule errors, i.e. production omissions and maintenance errors, or

- Model errors, i.e. violations of underlying model assumptions.

When singletons are inspected it is concluded that:

- 26 singletons are surveying rule errors, ('production omissions'), and
- two singletons are model errors, that is to say violations of underlying model assumptions.

These last two singletons are caused by instances that are small with respect to the imprecision of the data sets. Because a very small number of singletons is caused by model errors it is concluded that the problem of the integration of topographic data sets can be solved with an ontology-based approach. The combination of candidates for correspondences and singletons, followed by systematic inspection, ensures that we can find all correspondences (completeness), and discriminate between consistent and inconsistent correspondences (correctness).

The overall conclusion is that the ontology-based framework for integration and its subsequent implementation is feasible, subject to the conditions that there is:

1. At least partial knowledge of the surveying rules. If knowledge of surveying rules is incomplete, then a solution for incomplete surveying rules is comparing and inspecting data sets at the instance level (visually, by overlaying both data sets).
2. The data sets can be synchronized in time.

The application of this framework is most suitable for object classes with instances that are easy to identify and have a limited spatial extent (e.g. buildings).

References

Devogele, T., Trevisan, J., Raynal, L., 1996. Building a multi-scale database with scale-transition relationships. In: Proceedings of the Seventh International Symposium on Spatial Data Handling SDH-96, International Geographical Union, pp. 6.19–6.33.

Fonseca, F.T., Egenhofer, M.J., Agouris, P., Câmara, G., 2002. Using ontologies for integrated geographic information systems. Trans. GIS 6 (3), 231–257.

Frank, A.U., 2001. Tiers of ontology and consistency constraints in geographical information systems. Int. J. Geogr. Inform. Sci. 15 (7), 667–668.

Guarino, N., 1997. Semantic matching: formal ontological distinctions for information organization, extraction, and integration. In: Proceedings of the International Summer School, SCIE-97, Springer, Berlin, pp. 139–170.

Kilpeläinen, T., 2000. Maintenance of multiple representation databases for topographic data. Cartogr. J. 37 (2), 101–107.

Kuhn, W., 2001. Ontologies in support of activities in geographical space. Int. J. Geogr. Inform. Sci. 15 (7), 613–631.

Mars, N.J.I., 1995. What is an ontology? Seminar on the impact of ontologies on reuse, interoperability and distributed processing. Unicom, Uxbridge, Middlesex, UK, pp. 9–19.

Molenaar, M., 1998. An introduction to the Theory of Spatial Object Modelling for GIS. Taylor and Francis, London.

Pundt, H., Bishr, Y., 2002. Domain ontologies for data sharing-an example from environmental monitoring using field GIS. Comput. Geosci. 28, 95–102.

Sester, M., Anders, K.-H., Walter, V., 1998. Linking objects of different spatial data sets by integration and aggregation. GeoInformatica 2 (4), 335–357.

Uitermark, H.T., 2001. Ontology-based geographic data set integration. PhD Thesis. Computer Science Department, University of Twente, Enschede, The Netherlands (http://purl.org/utwente/36476).

Uitermark, H.T., van Oosterom, P.J.M., Mars, N.J.I., Molenaar, M., 1998. Propagating updates: finding corresponding objects in a multi-source environment. In: Proceedings of the Eighth International Symposium on Spatial Data Handling SDH'98, International Geographical Union, pp. 580–591.

van Oosterom, P.J.M., 1994. An R-tree based map-overlay algorithm. In: Proceedings of the Fifth European Conference on Geographical Information Systems (EGIS/MARI'94), EGIS Foundation, pp. 318–327.

van Oosterom, P.J.M., 1997. Maintaining consistent topology including historical data in a large spatial database. Proc. Auto-Carto 13, 327–336.

van Wijngaarden, F.A., van Putten, J.D., van Oosterom, P.J.M., Uitermark, H.T., 1997. Map Integration. Update propagation in a multi-source environment. In: Proceedings of the Fifth ACM Workshop on Advances in Geographic Information Systems ACM-GIS'97, ACM, New York, pp. 71–76.

Visser, U., Stuckenschmidt, H., Schuster, G., Vögele, T., 2002. Ontologies for geographic information processing. Comput. Geosci. 28, 103–117.