ONTOLOGY-BASED GEOGRAPHIC DATA SET INTEGRATION

Colophon

Manuscript prepared by the author using
Microsoft® Word text processor, with
MathType mathematical equation editor, and
EndNote®2 bibliography maker

Printed from *Adobe® Acrobat* pdf files by
PrintPartners Ipskamp B.V. Enschede

Cover design Frederik Helfrich BNO

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Uitermark, Harry

Ontology-based geographic data set integration
Harry Uitermark - [S.l. : s.n.]. - Ill.
Thesis Enschede. - With ref. - With summary.
ISBN 90-365-1617-X
Subject headings: ontologies / geographic information systems / data integration

ONTOLOGY-BASED GEOGRAPHIC DATA SET INTEGRATION

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 6 september 2001 te 13.15 uur

door

Henricus Theodorus Johannes Antonius Uitermark

geboren op 24 september 1946
te Haarlem

Dit proefschrift is goedgekeurd door de promotoren

Prof. Dr. Ir. N. J. I. Mars
Prof. Dr. Ir. M. Molenaar

To the Memory of My Father

Jaap Uitermark

19 March 1916
2 December 1983

Contents

Summary

Geographic data set integration is particularly important for *update propagation*, *i.e.* the reuse of updates from one data set in another data set. In this thesis geographic data set integration (also known as *map integration*) between two topographic data sets, GBKN and TOP10vector, is described. GBKN is a large-scale topographic data set and TOP10vector is a medium-scale topographic data set.

Geographic data set integration (or map integration) is defined as 'the process of establishing links between *corresponding object instances* in different, autonomously produced, geographic data sets of the same geographic space'. Corresponding object instances are *semantically similar*. Semantically similar means that corresponding object instances refer to the same terrain situation.

In the first part of this thesis a general introduction to geographic data set integration is given. Relevant literature is reviewed.

In the second part a *conceptual framework* for geographic data set integration is developed. Two important components of this framework are a *domain ontology* and a set of *surveying rules*. A domain ontology is important because it contains a set of *shared concepts*. It is this set of shared concepts of terrain situations that makes it possible to detect corresponding object instances.

The second important component of the framework is the set of surveying rules of a geographic data set. Surveying rules determine the *transformation* from a terrain situation into a geographic data set, as represented by object instances. Therefore, corresponding object instances from different geographic data sets must be *consistent* with different sets of surveying rules.

Surveying rules, by their very nature, determine the *level of abstraction* of a geographic data set. Different levels of abstraction between geographic data sets are associated with each other by two well-known abstraction mechanisms: a specialization-generalization hierarchy (a *taxonomy*), and a component-composite hierarchy (a *partonomy*). Using these abstraction mechanisms it is possible to model the semantic interconnectedness of object classes in a so-called *reference model*. Three types of semantic similarity between object classes of different data sets are distinguished in Chapter 2:

1. Equivalent object classes.
2. Object classes with a 'subclass-superclass' relationship.
3. Object classes with a 'composite class-component class' relationship.

In Chapter 3 the framework for geographic data set integration is mathematically translated into a system of set-theoretic formulae.

In the third part of this thesis the framework and its associated formulae system are tested on data sets from GBKN and TOP10vector. First of all a domain ontology with a basic set of six 'top-level' concepts is introduced. Candidates for this set of concepts are based on the Geo-Information Terrain Model (the Dutch *Terreinmodel*

Vastgoed). Domain ontology classes are further refined into subclasses, depending on the surveying rules of the two data sets. In this way a *common universe of discourse* for GBKN and TOP10vector is created.

Subsequently, in Chapter 4, the elements of this universe are structured in a reference model. The structuring is done using the abstraction mechanisms mentioned previously. As a consequence the reference model expresses every semantic similarity between both data sets involved. The concept of a *role* is introduced. A role reflects what object classes from different data sets are in *confrontation* with each other: these can be equivalent classes, subclasses, superclasses, component classes, or composite classes. Constructing a reference model is a highly cyclic and iterative activity, indicating that a geographic data set integration system should be a *learning* system.

The reference model is implemented and tested on actual GBKN and TOP10vector data sets in Chapter 5. Candidates for corresponding object instances are detected and subsequently checked for consistency with surveying rules. Many candidates are of a *complex* nature, *i.e.* groups or clusters of object instances correspond to each other. In order to be useful in, for example update propagation, complex correspondences should be broken down into simple ones (a subject for future research).

Object instances that do not participate in a correspondence are *singletons*. If all the roles between object classes of different data sets have been modeled completely and correctly then singletons indicate two types of errors:

1. Surveying rule errors, *i.e.* production omissions or maintenance errors.
2. Model errors, *i.e.* violations of underlying model assumptions.

In the fourth part of this thesis the framework is evaluated and conclusions are drawn. It is concluded that the problem of geographic data set integration 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*). Only a very small number of singletons are caused by model errors.

The overall conclusion of this research is that the ontology-based framework for geographic data set integration - with its formal mathematical foundation - and its subsequent implementation are feasible, subject to the conditions that geographic data sets are:

- two-dimensional vector data sets, where no object instances are displaced for cartographic reasons (traditionally, up to scale 1 : 12,500 - 15,000)
- with instances of area object classes
- with knowledge of surveying rules
- with thematic and geometric overlap, and with
- object instances, with crisp and complete boundaries.

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

Samenvatting

De *integratie* van geografische bestanden (= geo-bestanden) is met name van belang voor *mutatie-propagatie*, d.w.z het hergebruik van mutaties. In dit proefschrift is de integratie van geo-bestanden (ook wel bekend als *kaartintegratie*) beschreven tussen twee topografische bestanden, GBKN en TOP10vector. GBKN is een grootschalig en TOP10vector is een mid-schalig topografisch bestand.

De integratie van geo-bestanden is in dit onderzoek gedefinieerd als 'het proces van het tot stand brengen van koppelingen tussen *corresponderende instanties* uit verschillende, autonoom vervaardigde, geo-bestanden van hetzelfde gebied'. Corresponderende instanties zijn *semantisch verwant*. Semantisch verwant betekent dat zij verwijzen naar eenzelfde situatie in het terrein.

In het eerste deel van dit proefschrift wordt een inleiding over integratie van geo-bestanden gegeven. Van belang zijnde literatuur wordt nader beschouwd.

In het tweede deel wordt *een conceptueel raamwerk* voor het integreren van geo-bestanden ontwikkeld. Twee belangrijke componenten uit dit raamwerk zijn een *domein-ontologie* en een *verzameling van verkenningsregels*.

Een domein-ontologie is van belang omdat dit een verzameling is van *gedeelde begrippen*. Deze verzameling van gedeelde begrippen met betrekking tot het terrein, maakt het opsporen van corresponderende instanties (= correspondenties) mogelijk.

De tweede van belang zijnde component van het raamwerk is de verzameling van verkenningsregels van een geo-bestand. Verkenningsregels bepalen de *transformatie* van een terreinsituatie naar een geo-bestand, zoals dat bestaat uit instanties. Daarom dienen corresponderende instanties ook *consistent* te zijn met de verschillende verzamelingen van verkenningsregels van de betrokken bestanden.

Verkenningsregels bepalen door hun aard het *abstractieniveau* van een geo-bestand. Verschillende abstractieniveaus tussen geo-bestanden worden met elkaar in verband gebracht worden door middel van twee welbekende abstractie-mechanismen: een specialisatie-generalisatie hiërarchie (een taxonomie) en een component-composiet hiërarchie (een partonomie). Door gebruik te maken van deze abstractie-mechanismen is het mogelijk de samenhang van semantisch verwante objectklassen uit verschillende geo-bestanden te modelleren in een zogenoemd *referentiemodel*. Drie typen van semantische verwantschap tussen objectklassen uit verschillende bestanden worden in Hoofdstuk 2 onderscheiden:

1. Equivalente objectklassen.
2. Objectklassen, die een subklasse – superklasse relatie hebben.
3. Objectklassen, die een component – composiet relatie hebben.

Het raamwerk voor de integratie van geo-bestanden wordt in Hoofdstuk 3 vertaald in een stelsel van verzamelingtheoretische formules.

In het derde deel van dit proefschrift wordt het raamwerk en het bijbehorende formulestelsel getoetst op een GBKN- en een TOP10vector-bestand. Eerst wordt een domein-ontologie ontworpen van zes basisbegrippen. Kandidaten voor deze basis-

begrippen zijn gebaseerd op het *Terreinmodel Vastgoed*. Vervolgens worden de domein-ontologie-klassen verfijnd in subklassen, afhankelijk van respectievelijk de verkenningsregels van GBKN en TOP10vector. Aldus wordt een *gemeenschappelijk universum van begrippen* voor GBKN en TOP10vector gecreëerd.

Vervolgens worden in Hoofdstuk 4 de begrippen uit dit universum gestructureerd in een referentiemodel. Het structureren gebeurt op basis van de hiervoor genoemde abstractie-mechanismen, met als gevolg dat het referentiemodel iedere semantische verwantschap tussen de bestanden onderling uitdrukt. Het *rolbegrip* is hier van belang. Een rol is datgene, wat objectklassen uit verschillende geo-bestanden, in onderlinge confrontatie voor elkaar betekenen: equivalente klassen, subklassen, superklassen, componentklassen of composietklassen. Het construeren van een referentiemodel is cyclisch en iteratief, wat erop duidt dat een systeem voor de integratie van geo-bestanden een *lerend systeem* zou moeten zijn.

Het referentiemodel wordt geïmplementeerd en getoetst op actuele GBKN- en TOP10vector-gegevens in Hoofdstuk 5. Correspondentie-kandidaten worden opgespoord en vervolgens gecontroleerd op overeenstemming met de verkenningsregels. Vele correspondentie-kandidaten zijn *samengesteld* van aard, dat wil zeggen dat groepen of clusters van instanties met elkaar corresponderen. Teneinde toegepast te worden in mutatiepropagatie moeten samengestelde correspondenties afgebroken worden tot enkelvoudige (een onderwerp voor toekomstig onderzoek).

Instanties die niet voorkomen in correspondenties heten *singletons*. Indien alle rollen tussen objectklassen uit verschillende bestanden volledig en juist gemodelleerd zijn, dan duiden singletons op twee typen fouten:

1. Verkenningsregelfouten. Dit zijn fouten bij de inwinning of in het onderhoud.
2. Modelfouten. Dit zijn strijdigheden met onderliggende modelaannames.

In het vierde deel van dit proefschrift wordt het raamwerk geëvalueerd. De conclusie is dat het vraagstuk van de integratie van geo-bestanden oplosbaar is met behulp van een op ontologieën gebaseerde benadering. De combinatie van kandidaat-correspondenties en singletons, gevolgd door systematische controle garandeert dat alle correspondenties worden gevonden (*compleetheid*) en dat onderscheid te maken valt tussen consistente en niet-consistente correspondenties (*correctheid*). Slechts een gering aantal singletons worden veroorzaakt door modelfouten.

De algemene conclusie van dit onderzoek is dat het op ontologieën gebaseerde raamwerk voor de integratie van geo-bestanden, met zijn formele wiskundige grondslag, praktisch toepasbaar is, op voorwaarde dat de geo-bestanden:

- tweedimensionale vectorbestanden zijn, waar geen instanties verplaatst worden om cartografische redenen (traditioneel, tot schaal 1 : 12.500 - 15.000), die
- vlakobjecten bevatten, waarvan
- de verkenningsregels bekend zijn, met
- een thematische en geometrische overlap, en met
- instanties, die scherpe en volledige grenzen bezitten.

De toepassing van dit raamwerk is het meest geschikt voor objectklassen met instanties van beperkte ruimtelijke omvang (zoals gebouwen).

Preface

The work and research described in this thesis were carried out while I was working at the Netherlands Kadaster (Cadastre and Public Registers Agency). I am grateful to my employer and my colleagues for their support during that period.

My interest in Geographic Information Systems began more than twenty-five years ago, in 1975, when I visited AUTOCARTO II, an international symposium on so-called *computer assisted cartography* held in Reston, USA. On the basis of this experience I wrote an article about *topological data structures*, a new concept, for the Dutch geodetic press [1]. During the years of my professional career I have retained an interest in research matters, now and then culminating in an article about a new topic like *Prolog* [2].

For me to start a PhD research, however, was not an obvious activity. A combination of factors made it possible. First of all there was the suggestion that I undertake such research, made long ago by Theo Bogaerts, my supervisor in the Department of Geodesy at Delft University of Technology. Then, in 1995, there was an opportunity to do research on geographic data set integration, and I took a chance. This initiative was strongly encouraged by two Kadaster directors at that time, Victor van Dijk and Jan Sonnenberg. I was also lucky to find as a promotor Martien Molenaar, who recognized the importance of this research subject.

At the same time there was the happy circumstance that Peter van Oosterom joined the Kadaster, thus creating an ideal atmosphere for research and innovation. Certainly I thank Peter for being my daily supervisor and for his careful reading of my manuscripts.

Starting a PhD research is like entering a labyrinth. It was Frank van Wijngaarden, who was doing his master's thesis at the Kadaster, who showed me that there was a possible way out. And there were two other students who also amazed me with their achievements: Chang-Jin Kim and Anton Vogels.

The work reported in this thesis involves the fields of geodesy, computer science and mathematics. I have profited from the help of experts in these fields. In particular, I want to thank Rolf de By and Yashr Bishr, as well as R.M. Goldbach for his helpful remarks on my mathematical notation.

Further, I would like to thank John van Smaalen for providing me, at the right moment, with the TOP10vector data of this research. Of course, the copyright of this data belongs to the Topographical Survey (TDN), which I acknowledge. I would also like to thank the members of the TDN staff for their help in interpreting TOP10vector surveying rules.

---

[1] Uitermark, H.T. (1976). "Topologische gegevensstructuren voor kartografische gegevensbanken". *Nederlands Geodetisch Tijdschrift*, Vol. 6, No. 1, pp. 1-6.

[2] Uitermark, H.T. (1989). "Prolog en topologie. De programmeertaal Prolog toegepast bij relationele databases met topologisch gestructureerde ruimtelijke objecten". *Geodesia*, Vol. 31, No. 7/8, pp. 356-363.

Special thanks go to my long-time friend René van der Schans, whose creative ideas about the semantics and usability of geographic data sets are miles ahead of the mainstream of the geodesy profession.

Then there are people who are indispensable for the daily assistance they provide, such as Hans Vugts, Marleen Kleine and Anne Burghout from the Kadaster library, Robert Voss and Dirk van de Berg from the Kadaster printing-office, and Margreet Rombouts-Kroes from Kadaster Office Automation.

I am also much indebted to an anonymous person who invited me, in 1998, to participate in a Dagstuhl seminar on integrating spatial and temporal databases, organized by Oliver Günther, Timos Sellis, and Babis Theodoulidis. After all, my stay at Schloss Dagstuhl, situated in the Hunsrück in Germany, was a turning point in my research.

However, most of all I am indebted to Koos Mars. It is no exaggeration to say that nothing would have been achieved if Koos had not detected and characterized the *Terreinmodel Vastgoed* (Geo-Information Terrain Model) as an ontology. And the quality of this research is simply a result of his insistence on being clear in the formulation of my ideas. Once again, thank you Koos.

The completion of this research happens at a special moment.

On a personal level, I was very lucky to meet in the final stage of this research dear Tryntsje.

And, from a more business-like perspective, there is the future merging of the Topographical Survey (TDN) and Kadaster, making this research a timely event. There is also a growing interest in ontologies, especially with respect to the so-called Semantic Web [3]. And then there are the standardization activities of the Open GIS Consortium (OGC) with respect to GML, the geography 'dialect' of mark-up language XML. I sincerely hope that this research will find its place within all these developments.

Deventer, July 15, 2001 Harry Uitermark

[3] Berners-Lee, T., J. Hendler, and O. Lissila (2001). "The semantic Web". *Scientific American*, Vol. 282, No. 5, pp. 29-37.

Part 1: Introduction

1 Introduction

This research is a formal account of *geographic data set integration*, also known as *map integration*. Geographic data set integration (or map integration) is defined in this research as 'the process of establishing *relationships* between *corresponding object instances* in different, autonomously produced, geographic data sets of the same geographic space'.

Traditionally, in existing map series, corresponding object instances were linked *implicitly* by a common spatial reference system, for example the national grid (Devogele et al 1996; Sester et al 1998; Kilpeläinen 2000). Geographic data set integration aims at making links between corresponding object instances *explicitly* by investigating the way geographic data sets were acquired.

1.1 Motivation and Background

Motivation and *background* of this research is *update propagation*, which is the *reuse* of updates, from one geographic data set into another geographic data set. Update propagation is studied within the range of traditional topographic data sets, or map series (van Wijngaarden et al 1997; Uitermark et al 1998; Kim 1999; Vogels 1999).

A necessary condition for update propagation is geographic data set integration. Both issues, geographic data set integration and update propagation, are complicated enough to deserve a research project of their own. The first issue is chosen in this research, with an open eye towards its application in update propagation.

While geographic data set integration in this research is intimately related to update propagation, geographic data set integration has also a purpose, an aim of its own. Integrating two data sets may mean that the combination is more than the sum of its parts. 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 'best of both worlds'. With this third data set, queries can be answered that can not be answered by the two data sets separately.

1.2 An Informal Introduction to Geographic Data Integration

The collection of geographic data in order to produce a paper map, is an activity that has been going on for centuries. Only recently, since the last thirty years, geographic data is not stored on paper but in electronic, digital form. First in traditional plot files, and nowadays mostly in a dedicated information system with a special database,[1] called a Geographic Information System, abbreviated as GIS. This availability of geographic data in digital form makes it relatively easy to combine,

[1] Or a standard database, with extensions for geographic data.

or put together, geographic data sets of different origin, provided that these sets are of the same geographic space, and can be transformed to a common reference system. This transformation to a common reference system is sometimes trivial, or sometimes extremely complicated (see for example (Laurini 1998)). However, after this transformation another problem pops up, if one wants to compare and interpret the combined data sets on the basis of individual data elements, and draw conclusions from these comparisons. *This is the problem of geographic data set integration.*

Take for example maps of two simple geographic data sets (**Fig. 1**). Assume that both maps are from the same geographic space, and are the same 'snapshot' in time.

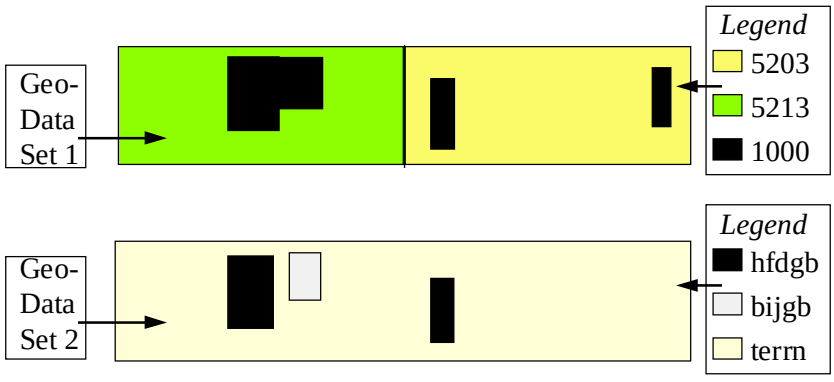


Fig. 1. Maps of two simple geographic data sets: same region, same moment in time. In what way are data sets similar, or different?

They look similar although there are differences. But how are we able to decide whether they resemble each other, or are different from each other? A simple *overlaying* of both maps might reveal coinciding areas. However, in order to interpret and draw valid conclusions from these coinciding areas, a necessary condition is the understanding of the *semantics*, the meaning of data sets.

Inspecting *legends* of both maps in **Fig. 1**, semantics of both data sets is far from clear. What are *class labels* as '5203' or 'hfdgb' supposed to mean? These class labels refer to object classes with definitions within different data models. To reconcile these different data models, it is useful, even mandatory, to investigate the way geographic data sets were acquired, which is to say how the transformation was from *real-world phenomena* to data sets.

But then there is still a problem. In order to express and compare *surveying rules*, used in the acquisition of data sets, a collection of *common ideas*, or *notions*, of terrain objects is needed. This collection of common definitions of terrain objects is in many cases not available, because geographic data sets are produced independently by different organizations, all with their own objectives and ideas about terrain objects. Therefore it is necessary to invent or construct a collection of

common definitions of terrain objects. Here is where a *domain ontology* is born, a collection of *shared concepts*, as a 'cover' for understanding object definitions in different geographic data sets.

To illustrate ideas as *surveying rules* and *domain ontology* take a simple terrain situation as in **Fig. 2**.

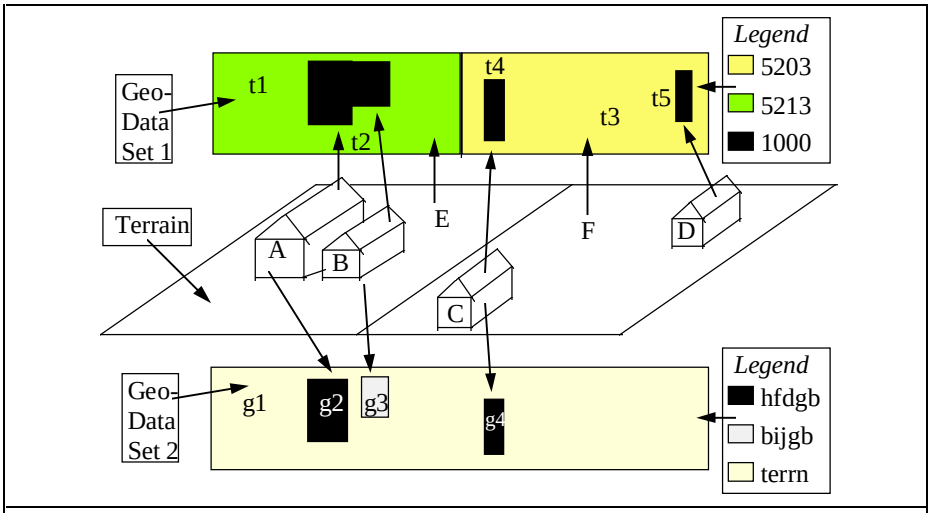


Fig. 2. Two transformations of a terrain situation. In comparing both geographic data sets it is mandatory to know surveying rules in order to conclude if both data sets are consistent with the same terrain situation.

In the terrain situation (**Fig. 2**, *middle*) there are four buildings, labeled A, B, C, and D, and two parcels E and F. In our domain ontology we have definitions for buildings and parcels, as well as their properties. This terrain situation is acquired with two different sets of surveying rules:

- According to surveying rules of Geographic Data Set 1:
  - buildings A, B, C, and D are acquired, and represented as t2, t4, and t5 with label '1000' in the map of Geographic Data Set 1 (**Fig. 2**, *above*). Observe that A and B are merged into t2, because A and B are sufficiently near to each other. 'Sufficiently' has a precise definition in surveying rules of Geographic Data Set 1
  - parcel E (grassland) and F (arable land) are acquired, and represented as t1 (label '5213') and t3 (label '5203') in the map of Geographic Data Set 1 (**Fig. 2**, *above*).
- According to surveying rules of Geographic Data Set 2:
  - buildings A, B, and C are acquired, recorded with different properties, and represented as g2 (label 'hfdgb'), g3 (label 'bijgb'), and g4 (label 'hfdgb') in

the map of Geographic Data Set 2 (**Fig. 2**, *below*). Building D is not represented because its area size is too small. Again, 'too small' is precisely defined in surveying rules of Geographic Data Set 2

– parcels E and F are acquired, and represented as g1 (label 'terrn') in the map of Geographic Data Set 2 (**Fig. 2**, *below*). E and F are merged into g1 because surveying rules state that recording different properties of E and F is not relevant for Geographic Data Set 2.

In order to understand semantic interconnectedness of both geographic data sets, domain ontology concepts as 'building' and 'parcel' are refined into concepts as 'mainbuilding', 'annex next to mainbuilding', 'free standing annex', 'arableland', and 'grassland'. By s*tructuring* these concepts in a *reference model*, where concept labels *refer* to class labels, meaning is given to the hidden semantics of geographic data sets (**Fig. 3**).

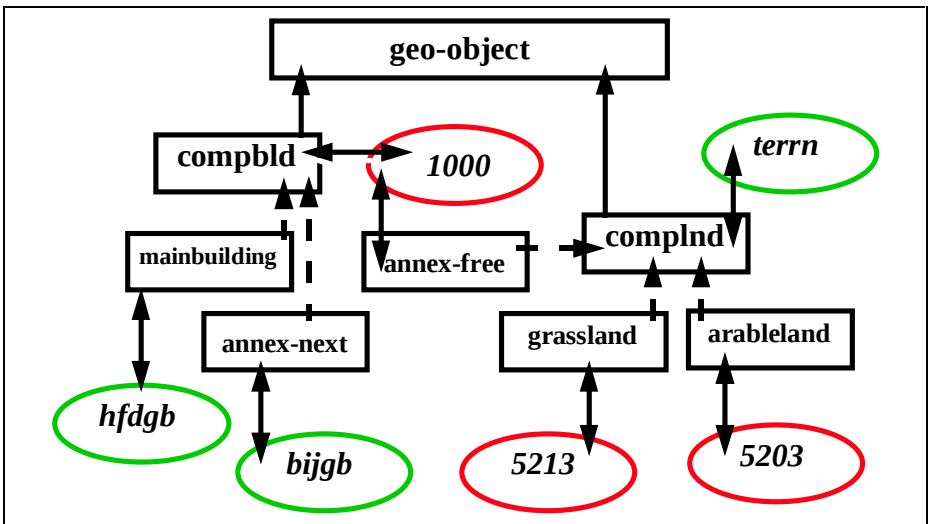


Fig. 3. Refined concepts from a domain ontology are structured in a reference model (concept labels in black rectangles). Concepts refer to object class labels (red and green ovals), revealing *semantic interconnectedness* of geographic data sets.

With a reference model it is possible to reason or form hypotheses about terrain situations that are consistently represented in both data sets. *This is what geographic data set integration is about*.

To do this reasoning, relationships between data elements from different sets, the *corresponding object instances*, must be known, and these relationships must also be *consistent with surveying rules* of data sets involved. Otherwise one can not determine whether data sets in **Fig. 2** are consistent with the same terrain situation.

A *first outcome* of integrating geographic data sets of the preceding example is a *list* of relationships between *candidates for corresponding object instances*:

{ {(t2, g2), (t2, g3)}, {(t1, g1), (t3, g1), (t5, g1)}, {(t4, g4)} }

In a subsequent action, candidates for corresponding object instances are checked *for consistency* with surveying rules.

From now on, if there is a modification in a terrain situation, which is succeedingly recorded for Geographic Data Set 1, it is clear from relationships between corresponding object instances, if and how Geographic Data Set 2 will be influenced. This will be a starting point for *update propagation*.

Here ends our informal introduction to ontology-based geographic data set integration. From now on the writing will become formal, eventually culminating in a set of concepts and formulae in which the problem of geographic data set will be stated, and successfully solved.

All concepts mentioned in this section will get their proper place and attention:

- Domain ontologies (Section 2.1)
- Surveying rules (Section 2.2)
- Reference models (Section 2.5),
- Corresponding object instances (Section 2.7), and
- Consistency checking (Section 2.9).

For a further understanding of the intricacies of geographic data set integration, the characteristics of a geographic data set are briefly mentioned.

1.3 Characteristics of a Geographic Data Set [2]

A geographic data set (or *geo-data set* for short) is an abstraction from a terrain situation, or real-world situation, with a collection of data elements (*object instances*), which represent real-world phenomena, with the central property that they are *fixed in relation to the earth surface*.

In this definition, two items are important:

1. It is an *abstraction* from a terrain situation: producing a geographic data set means defining a *classification* system (a system of classes) and rules for data capturing (surveying rules). Two a*bstraction mechanisms* are fundamental to classification processes:
   - there is a generalization/specialization classification, which means that classes are grouped in a *taxonomy*, with *superclasses* and *subclasses*, and
   - there is a composite/component classification, which means that classes are grouped in a *partonomy* of *composite classes* with *component classes* as constituents (a partonomy is also known as an aggregation hierarchy).

[2] Geographic data sets, in this research, are *object-structured* data sets; satellite imagery, and aerial photography are, by contrast, *pixel-structured* data sets.

2. Object instances represent real-world phenomena, which are *fixed in relation to the earth surface*: every object instance has a *geometric description* or *geometric attribute*, for example a list of coordinates in a spatial reference system [3].

The last item sets a geographic data set definitely apart from a non-geographic data set (**Fig. 4**).

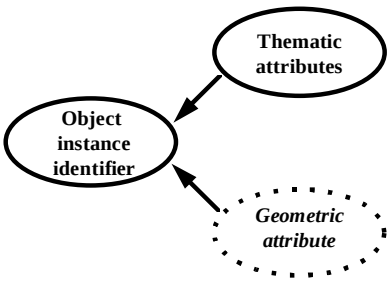


Fig. 4. The basic structure of an object instance from a geographic data set: an identifier, with a link to thematic attributes, and a link to a geometric attribute. After (Molenaar 1998, p.4).

Object instances are elementary building blocks of a data set. One of the thematic attributes is class membership. Information about the geometric attribute of an object instance has three aspects (Molenaar 1998, p.6):

1. Position and orientation: the situation of an object instance with respect to a coordinate system.
2. Size and shape: *metric* properties of an object instance, such as 'length' or 'width'.
3. Topology: *non-metric* properties *between* object instances, such as 'adjacency' or 'inside'.

Contrary to a map, a geographic data set does not have an explicit notion of *scale*, the ratio of the size of an object instance represented in a map, or on screen, and its size in reality. In a geographic data set it is according to (Devogele et al 1996) more sensible to replace the notion of scale with concepts as:

- precision: the degree in detail in abstractions
- accuracy: the relationship between an abstraction and the terrain that it claims to represent (in other words, the likelihood of errors), and
- resolution: the smallest object which can be represented.

[3] *Fixed in relation to the earth surface* is a relative concept. It depends on the time scale. In this respect geographic objects are considerably different from *moving objects*, like persons or vehicles.

1.4 Problem Definition of Geographic Data Set Integration

There are two broad categories of factors responsible for the differences between geographic data sets (Bishr 1997):

1. Differences in contents. Data sets are collected for specific purposes, sometimes totally different from one set to the next one (in other words, there are different *themes*). A geographic data set is a representation of a set of real-world phenomena. Different sets of real-world phenomena will imply different *contents* among geographic data sets.
2. Differences in abstraction and level of detail. In capturing real-world phenomena there is the process of *transforming* real-world phenomena into a data set representation. Different rules for surveying, for the same terrain situation, may lead to different object classes, with different attributes, and different geometric descriptions, by points, lines, or polygons.

Above all, these differences make it important to develop an understanding of the *semantics* of the data sets, that is to say what they *mean*. Semantics should be understood as the link between a terrain situation and a data set representation (Wintraecken 1987), or in other words between a reference model class and a data set object class.

The *problem* of geographic data set integration is defined as establishing relationships between corresponding object instances, *considering the differences* between geographic data sets to be integrated.

1.5 Research Objective

Given the problem definition of geographic data set integration, one might ask whether there are approaches and methodologies, which *reconcile* the apparent differences between geographic data sets, so that these sets can be integrated in a consistent manner. Reconciling differences means finding mechanisms that account for differences in contents, abstraction, and level of detail. The *objective* of this research is to invent, construct, and implement such a methodology, and test it on two existing geographic data sets.

More specifically, given two different geographic data sets, we try to answer the following research questions:

1. What kind of relationships exist between corresponding object instances?
2. How can we find corresponding object instances, and under what conditions can we find them?
3. How certain are we about completeness and correctness of these corresponding object instances, and how can we check their consistency?

1.6 A Review of Relevant Work on Geographic Data Set Integration

Relevant work on geographic data set integration can be found in three domains: artificial intelligence, computer science and geo-science. Each example is exceptional for the state-of-the art in that domain.

1.6.1 Examples from Three Domains

1. In the *Artificial Intelligence* domain, a *spatial* (= geographic) *ontology* is defined in (Benslimane et al 2000), with key features of urban planning applications, to provide a foundation for semantic reconciliation among themes that represent different urban infra-structures (for example land use, transportation, and power networks).

   – A top-level ontology represents concepts that are common to all themes. For every theme an ontology is defined. Every ontology of a theme has two levels: a functional level, and an application level. The latter represents the semantics of real-world objects, whereas the former consists of descriptions, which are used to define operations and constraints. Inter-ontology relationships are spatial relationships among object instances in one or more themes (Benslimane et al 2000, p.202). For example, a water-pipe object instance from a water supply theme is at the same location as a street object instance from a road network theme.

   ∗ Review: As far as the representation of the semantics of real-world objects is concerned, it is not mentioned how the relationships are between these *representations* and the real-world objects, nor is it mentioned how real-world objects are defined, and on what conditions they are acquired and transformed to data sets of the themes. Conclusion: in (Benslimane et al 2000) there is much emphasis on structure, contents, and behavior of object instances, but no attention is given to the acquisition phase. Which is not much of a surprise: the goal of (Benslimane et al 2000) is to provide support for queries over multiple themes, not update propagation.

2. In the *Computer Science* domain, *Schema Integration* (SI) has been the dominant methodology for data set integration. A schema refers to a data set specific description of object classes with their attributes. In SI the commonalities between the schemata of data sets are identified. Whether a commonality holds or not is based on the semantics of the data sets. From the commonalities a single unified description, the integrated schema, is derived (Larson et al 1989; Spaccapietra et al 1992; Dupont 1994; Castano et al 2001).

   – An example of SI for geographic data sets is described in (Devogele et al 1998). Here the process of unifying existing geographic data sets into a single framework is called *database integration*. Database integration implies that “whenever the existing databases contain duplicate, complementary or otherwise related descriptions of the same real-world phenomena, these

descriptions should be appropriately merged to provide a single picture of the overall data. This kind of merging is performed at the class level, resulting in the integrated schema, and *virtually at the instance level*, resulting in the integrated database" (Devogele et al 1998, p.341). Two road network databases are integrated. Commonalities between the road network schemata are defined as *Inter-schema Correspondence Assertions* (*ICA*). Assuming S_i and S_j denoting the schemata of two databases to be integrated, an ICA is defined as:

$$S_i\text{-item}_R \; \textit{correspondence-relationship} \; S_j\text{-item}_S$$

with *correspondence-relationship* one of the usual set relationships:

$$\{ \equiv, \subset, \subseteq, \cap, \supset, \supseteq, \neq \}$$

An ICA includes also a predicate for *instance matching*, based on location, that is to say their position in space. The instance matching predicate defines the correspondence at the instance level. The predicate may be simple (based on equality of coordinates) or complex (functions operating on thematic and geometric attributes). Special attention is given to so-called 'fragmentation/aggregation' conflicts that denote situations where there is either a 1 : n (n > 1) or a n : m (n > 1, m > 1) correspondence between instances from two databases (Devogele et al 1998, p.344). This is solved using the modeling concept of aggregation (composition link, part-of link). From the ICA's the integrated schema is built.

- Review: Schema integration has been applied to two existing geographic databases. Conflicts arising from differences in representations are solved ("Although some more complex conflicts are still pending ..." (Devogele et al 1998, p.348)). How these differences were detected is not mentioned. By visually inspecting data sets and discovering irregularities? There is only a general remark on surveying rules ("As no strong guidelines exist for data collection ..." (Devogele et al 1998, p.338)). 'Real-world phenomena' are mentioned but no reference is made to definitions nor to related abstraction processes in the acquisition of these phenomena. Relationships between instances are virtual, or implicit, not explicit. This is contrary with geographic data set integration, defined in this research. There is no reference to *consistency checking*, a fundamental notion in this research. Conclusion: This is a remarkable example from the computer science domain. Its strong point is the powerful modeling capacity, its weak point the relationship between database and reality.

3. In the *Geo-science* domain, methods from communication theory are adopted, such as *relational matching*. In (Sester et al 1998) geographic data set integration is defined as a matching problem, which means that geometrical elements of the data sets should be matched to each other. These elements should belong to data sets *of similar precision and resolution.*

- As an example, a road network data set is matched with a topographic data set. For the matching of both data sets *length*, *shape* and *position* of start and end points of road elements is considered. Then, the best matching of two data sets is a collection/combination of matching pairs (of elements) that maximizes a support function. On average 96% of the elements were matched correctly compared to manual matching (Sester et al 1998, p.344). The method is not usable for data sets of different resolution. To overcome this problem, both data sets have to be transformed to a similar precision and resolution. In (Sester et al 1998, p.345-) it is demonstrated how this can be done for built-up areas from a large-scale topographic map into built-up areas for a medium-scale topographic map. By comparing (1) acquisition rules of both data sets, and (2) visually inspecting both data sets, *aggregation rules*, which are rules that describe relationships between the data sets, are formulated. These rules are formalized in a data model that take topological relationships into account.

* Review: Relational matching is usable for integrating data sets that exhibit a similarity in precision and resolution. In addition, they must represent similar instances from a common object class, for example roads. To overcome this restriction of similarity in precision and resolution, the data sets have to be preprocessed to get similar representations. Again, they must belong to a common object class. This preprocessing is done with knowledge from 'acquisition rules' and by visually inspecting data sets and discovering regularities (Sester et al 1998, p.354). Conclusion: the integration seems to be done on a object class by object class basis, and there seems to be no provision for additional conditions ('exceptions') of the acquisition rules (= surveying rules). There is no reference made to *consistency checking.*

1.6.2 Relevant Work and the Approach of this Research

In contrast with the preceding examples, the approach of this research to geographic data set integration is to make semantics of data sets explicit by referring to terrain situations, define what terrain objects are of interest and how, and when they are acquired. Only then we can check if data sets do not contradict each other. This consistency checking is a necessary prerequisite for update propagation. In reviewing the preceding references, the overall conclusion is that these references do not give a proper answer for consistency checking.

Making semantics of data sets explicit brings an ontology-based approach to the foreground.

1.7 An Ontology-Based Approach to Geographic Data Set Integration

Making semantics explicit is a *communication* problem. Any successful communication requires a language that builds on a core of shared concepts (Kuhn 1996). An *ontology* is such a collection of shared concepts. Consequently, an *ontology-based approach* is chosen in this research.

Informally, an ontology is an inventory of things that are *usefully distinguished* in a given domain together with a definition of the properties of those things and the relations that hold among them. An ontology thus provides a vocabulary for a given domain, together with a set of definitions, which constrain the meaning of vocabulary terms to enable consistent interpretation of data framed in that vocabulary (Papaioannou 1998).

In our approach we distinguish two domains: the domain of an *application*, and the domain of a *discipline*. From now on we identify two types of ontologies:

1. An *application ontology*, with object classes from a geographic data set. In data set integration there are at least two application ontologies.
2. A *domain ontology*, with object classes from the discipline of topographic mapping (geographic data sets in this research are from the discipline of topographic mapping).

In Section 2.1 the topic of ontology will be further explained.

1.8 Research Design

This research is divided into *two* parts:

1. *Methodology development*. In Part 2 of this research, a conceptual and formal framework for geographic data set integration is developed. Elements of this conceptual and formal framework are ontologies and surveying rules. Concepts from a domain ontology are used to construct a reference model that accounts for the similarities between the geographic data sets. Between domain ontology concepts in the reference model, and the application ontology concepts, *semantic relationships* are defined.
2. *Implementing and testing the methodology on two existing geographic data sets*. In Part 3 of this research the methodology of Part 2 is tested on two existing geographic data sets:
   - surveying rules of both data sets are made explicit
   - both geographic data sets are compared in detail in order to find resemblances and differences that are not explained by surveying rules
   - from this information a reference model is constructed that accounts for these resemblances and differences
   - with the help of the reference model candidates for corresponding object instances are discovered, and checked for consistency with the surveying rules.

1.9 Scope and Limits of this Research

The motivation of this research does not come from theoretical interest but from an exceptionally practical problem: how to reuse updates from one geographic data set to update another geographic data set.

In order to propagate updates from one data set to another data set, it is necessary to find corresponding object instances, which means geographic data set integration.

This research concentrates on this problem, and its solution:

1. by developing a methodology. Mathematical methods are used to formalize this methodology.
2. the methodology is illustrated and tested in a case study with real geographic data sets.

This formal method makes the methodology *transferable* to other geographic data sets, with the following characteristics:

- object instances, with crisp and complete boundaries
- with a finite set of labels from a classification system
- with a thematic overlap (= object classes from semantically similar themes; will be explained more fully later on), and
- with a geometric overlap (an overlap in geographic space).

While the applied geographic data sets are real, neither *efficiency* of the methodology nor its *scalability* — the scaling up for practical application — are addressed in this research.

1.9.1 Geographic Data Sets of this Research

The methodology of geographic data set integration is tested on instances of *area object classes* of two *existing* geographic data sets, GBKN and TOP10vector:

- GBKN data set is a Dutch large-scale topographic data set (presentation scale 1 : 1,000). It is usually produced by photogrammetric *stereo plotting* with field completion. It is a nationwide mapping of buildings, roads, waterways, and railways.
- TOP10vector data set is a Dutch mid-scale topographic data set (presentation scale 1 : 10,000). It is usually produced by photogrammetric *mono plotting* with field completion. It is a nationwide mapping of buildings, roads, waterways, railways, and land use.

Both data sets are studied in the context of update propagation research (TDN and Kadaster 1995; TDN et al 1997; TDN and Kadaster 1999). They are good examples of a whole range of traditional topographic data sets, with crisp object boundaries and sharp object class classifications.

1.9.2 Imprecision of Data Sets

Geographic data sets in this research are from the practice of land surveying, or topographic mapping. Data sets have their ‘natural’ imprecision caused by production processes. However, a typical aspect of these large-scale and mid-scale

topographical data sets is that no object instance is *displaced* for cartographic, or representational reasons.

1.9.3 Structure of Data Sets

Data sets in this research are two-dimensional (R2) *vector data* sets. The structure of a vector data set is a combination of a *thematic* and a *geometric partition*. A partition is a subdivision of a data set S into a collection $\{A_i\}$ of non-overlapping, non-empty sub-sets:

- each element in S belongs to one of the A_i.
- the sets A_i are mutually disjoint: $A_i \cap A_j = \varnothing$, for $i \neq j$.

A *thematic* partition means that every terrain object belongs to exactly one object class. A *geometric* partition means that the combined geometric attributes of all terrain objects will result in a continuum with neither *gaps* nor *overlap*. The concept of this combination of thematic and geometric partition is known as *Single Valued Vector Map* (*SVVM*) (Molenaar 1989).

1.9.4 Synchronization of Data Sets

While this research originated in update propagation this subject as such will not be covered in this research, except for a few examples to illustrate its peculiarities, especially in relation to surveying activities.

In order to integrate geographic data sets it is necessary to synchronize data sets. That means, all geographic data sets involved should refer to a terrain situation of the same moment in time. Synchronizing geographic data sets is of paramount importance, and even more important for update propagation. However, in this research we concentrate on the linking aspect between different geographic data sets, and it is assumed that there is a mechanism for geographic data set synchronization (van Oosterom 1997).

This research started before such a mechanism was available for the data sets used in Chapter 4 and 5. Therefore they do not represent the same snapshot in time (GBKN: 1996, TOP10vector: 1995). Moreover, GBKN was (and still is) a line-structured geographic data set. For this research a GBKN is used, which was produced as a prototype for an object-structured GBKN, during experiments in 1996 (van der Veen and Uitermark 1995; Kadaster 1996).

1.9.5 Prototype Map Integrator

It is envisioned that in the future there will be a class of software modules, called *mediators*, which mediate between several different (geographic) databases (Wiederhold 1992). A mediator contains an expert's knowledge and makes that expertise available to an user application. Such a user application could be geographic data set integration. It is in this context that the term *Map Integrator* was coined (Uitermark 1996). This research touches at the issue of expert knowledge for

geographic data set integration, where this expert knowledge is formalized and ready to be integrated in an expert system module.

1.9.6 Geographic Data Set Integration and Interoperability

Geographic data set integration is getting more attention, not only for update propagation, but also for the more general goal of *sharing information* between different geographic information sources (Laurini 1993). Sharing and reusing data from various heterogeneous information systems is nowadays a remarkably important issue, brought up under the heading of *interoperability*. In the world of Geographic Information Systems (GIS) there is a worldwide organization, the Open GIS Consortium (OGC), which considers this interoperability as its mission (Open GIS Consortium Inc. 1996).

There are, roughly speaking, two different levels of interoperability. There is a technical level — or, the systems perspective — with an understanding of information processing issues, like network *protocols* and *standards* for data set files. And there is a semantics level — or, the data modeling perspective — with an understanding of the *semantics* of information processing. Geographic data set integration is at the core of that level, focusing on the resolution of differences in the underlying data set models (Bishr 1998; Hadzilakos et al 2000).

1.9.7 Research Tools

No applied scientific work can be done without a set of suitable tools. In this respect three software tools are mentioned:

- the Mapover/Topol-package for overlaying geographic data sets, finding intersecting points, and reconstructing topology (van Oosterom 1994; van Putten 1997)
- *Mathematica*©, a system for doing mathematics by computer (Wolfram 1996). This package is remarkably suitable for rapid program development, not for efficient computing. The Prolog interpreter used in this research was developed within *Mathematica*© (Maeder 1994)
- ArcView©, a geographic information system (ESRI 1994). This package is suitable for visualizations.

1.10 Thesis Overview

The organization of this thesis is in *four* parts:

- Part 1 (Introduction). This beginning chapter has been the first part.
- Part 2 (Methodology development). The second part explains the development of a methodology for geographic data set integration:

  - Chapter 2 gives a conceptual framework for ontology-based geographic data set integration, and gives formal definitions of relationships between reference model and application ontology, and
  - in Chapter 3 these relationships are translated in concepts from set theory, in order to determine corresponding object classes and instances.
- Part 3 (Practice). The third part explains the implementation and testing of geographic data set integration:
  - in Chapter 4 a reference model is constructed, and
  - implemented and tested on GBKN and TOP10vector data sets in Chapter 5.
- Part 4 (Evaluation and Conclusions). This thesis ends with:
  - an evaluation of experimental results in Chapter 6, with finally
  - the conclusions of this research in Chapter 7.

Part 2: Development of a Methodology for Geographic Data Set Integration

2 A Conceptual Framework for Integration

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

The key issue in geographic data set integration is finding corresponding object instances. This process of *semantic matching* is only possible if the meaning of objects is clear. Central in a conceptual framework for integration is a mechanism that makes object definitions clear; that means, make data sets *semantically transparent* to each other. In that respect geographic data set integration can be seen as a *communication* problem. Any successful communication requires a language that builds on a core of shared concepts (Kuhn 1996).

It is here that an ontology plays a fundamental role. 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 geographic data set to be integrated (Section 2.1.3). *Abstraction rules* define relationships between terrain and domain ontology; *surveying rules* define relationships between concepts of the domain ontology and concepts of application ontologies (Section 2.2). Surveying rules are context-dependent, that is to say they depend on the local situation in the terrain (Section 2.3).

In this research, geographic data sets are used from the discipline of *topographic mapping*. There is an official Dutch standard for topographic data set transfer, called *Geo-Information Terrain Model* (GTM) (Ravi 1995). Elements of the GTM Standard are used in this research for the construction of a domain ontology for topographic mapping (Section 2.4).

Concepts from domain ontology, and information from surveying rules are used in constructing a *reference model*. A reference model is a subset of concepts from a domain ontology, with additional structure, belonging to the combination of geo-data sets to be integrated. Relationships between reference model concepts and application ontology concepts define the semantics of a data set (Section 2.5).

Domain ontology, application ontologies, surveying rules, a reference model, and semantic relationships are the fundamental building blocks for a conceptual framework for ontology-based geographic data set integration (Section 2.6). With these building blocks *corresponding object classes* and *corresponding object instances* are defined — the latter being the ultimate goal of geographic data set

integration (Section 2.7). Important parts in the definition of corresponding object instances are *location* (Section 2.8) and *consistency* (Section 2.9).

2.1 Concept and 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, which deals with the nature and organization of reality (Smith 1996).[4] However, in Artificial Intelligence (AI), a subfield of computer science, an ontology has to do with the explication of knowledge to overcome the problem of semantic diversity of different information sources (Wache et al 2001).

2.1.1 Ontologies in Artificial Intelligence Literature

In Artificial Intelligence literature there is still a debate on the definition of an ontology. Most definitions converge to an ontology as a *conceptual reference system*, with a collection of concepts, classification hierarchies, and thesauruses (reference books in which natural language terms, referring to similar concepts are grouped together). However, definitions diverge on the issue of structure — the way an ontology is organized:

- Braspenning and Lemmens (Braspenning and Lemmens 1997) refer to concepts in an ontology as 'semantic primitives', which determine what we are able to express about our field of interest. Thus, what things exist, *not* what their properties are, nor their relationships.
- Huhns and Singh (Huhns and Singh 1997) indicate an ontology as a semantic network, a graph with concepts as nodes, and relationships as edges. This network is supplemented with additional properties, constraints, procedures, and rules, which determine the behavior of the concepts. Their ontology editor represents an ontology as an Entity-Relationship diagram.
- For Bishr (Bishr 1997) an ontology is a hierarchy of interconnected hyperonyms and hyponyms from a vocabulary that defines a shared domain. A hyperonym is a concept that embodies the meaning of other concepts, like 'piece of furniture' embodies 'table' and 'chair'. A hyponym refers to the inverse relationship: 'table' and 'chair' are hyponyms of 'piece of furniture'. Thus, the organizing principle is a class hierarchy, with generalization and specialization as abstraction mechanisms.
- Mizoguchi et al (Mizoguchi et al 1995) postulate an ontology as a system of concepts, a vocabulary, used as primitives in building an expert system.

---

[4] Ontology is a Greek word. The founding father of the doctrine of existence was the Greek philosopher Parmenides. The term *ontology* was coined by Clauberg in 1646 to indicate the traditional philosophy of Aristotle in *Metaphysics*, one of Aristotle's major works (Algemene Winkler Prins 1958, p.707).

In conclusion, previous ontology definitions emphasize use of concepts in a reference system, allowing for some structure between concepts. This links up with our definition of an ontology.

2.1.2 Definition of an Ontology in this Research

The definition of an ontology in this research is made operational as 'a structured, limitative collection of unambiguously defined concepts' (Mars 1995; van der Vet and Mars 1998).

This definition contains *four* items:

1. An ontology is a collection of *concepts*, rather than terms.
2. Concepts are to be *unambiguously* defined.
3. The collection is *limitative*.
4. The collection has *structure*. Structure means that the ontology contains relationships between concepts.

Many scientific and engineering disciplines have developed a subset of language, a *vocabulary*, and we find terminology committees charged with defining meaning and usage of specific *terms*. In an ontology concepts are used, not terms, preferably presented in a language-independent way (which is hard to realize most of the time). In addition, formal rules must limit possible interpretations of a concept, to be supplemented by an informal natural-language definition.

Concepts not in the ontology cannot be used. This item is closely related to the notion of *ontological commitment*, which is an agreement what collection of shared concepts to use. [5]

There is a similarity between our definition of an ontology on the one hand, and the thematic partition of a Single Valued Vector Map (SVVM) on the other hand (see Section 1.9.3). Both require an unambiguous description of the universe of discourse (the relevant terrain situations), and both require the exhaustiveness and limitativeness of concepts used in that description.

2.1.3 Domain Ontologies and Application Ontologies

In this research an ontology for a certain *discipline* is called a *domain* ontology. Geographic 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 certain geographic data set is called in this research an *application* ontology. In geographic data sets, names or labels for mapped or

---

[5] There is also an analogy with the notion of a *closed world assumption* in logic programming languages where the asserted clauses in the database are the *only* source of information (Malpas 1987, p.60).

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’s why we must make a distinction between concepts in the domain ontology, and concepts in application ontologies of data sets involved in the integration process.

This distinction also resolves *naming* diversity, like homonyms (same name used for different concepts), or synonyms (different names used for same concept).

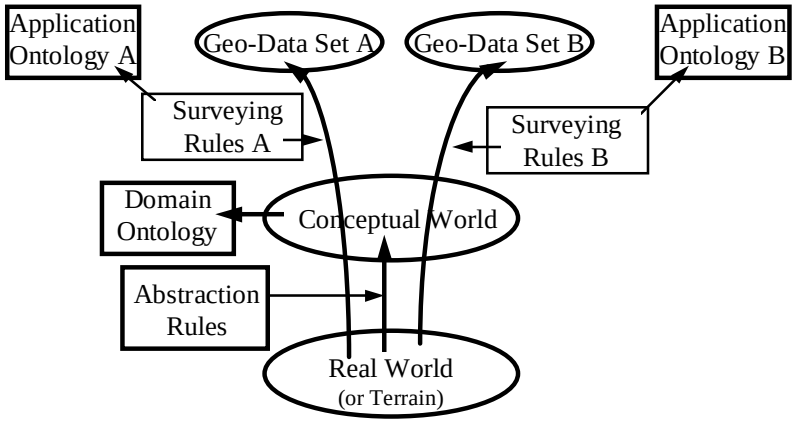


Fig. 5. The Real World (or Terrain) is abstracted according to *abstraction rules* into a Conceptual World, and described in a Domain Ontology. What is known in the Conceptual World is acquired according to surveying rules, and depending on the application, captured in a Geo-Data Set (adapted from (van der Schans 1994) and (Winter 2000, p.420)).

2.2 Abstraction Rules and Surveying Rules

Abstracting the Real World is a two-step process (**Fig. 5**):

1. There exist classes of real-world phenomena. There may be many classes of real-world phenomena, or terrain objects, but only terrain objects from classes, relevant for a certain discipline, which can be identified and labeled, are included as concepts, or object classes, in a domain ontology.[6] Rules which govern this selection — from classes of terrain objects into classes of the domain ontology — are defined as *abstraction rules*.[7]

2. With this collection of object classes we look at the terrain: it is as if we wear a pair of glasses, where only instances of object classes of the domain ontology are passed through. From this filtered collection of terrain objects — only those

[6] To be as general as possible we use the term *object class* as synonymous of *concept*.

[7] A fundamental problem is excluded from the discussion here: how to talk about the Real World without a real-world ontology? This is a *meta-meta activity*: how to formulate rules for the formation of abstraction rules (van der Schans 1997).

relevant for a certain application and included in an application ontology — are acquired or 'captured' into a geographic data set. *Surveying rules* (or, alternatively *acquisition rules*) are defined as rules, which *govern the transformation process* from the actual observed terrain objects, defined as instances from object classes in the domain ontology, into instances of geographic data set object classes, as defined in an application ontology.

Surveying rules define *what* object classes and *how* object classes are represented. Consequently, surveying rules include:

- inclusion rules: which instances of object classes are selected ('capture criteria' in Open GIS Consortium vocabulary (Open GIS Consortium Inc. 1998))
- simplification rules: how instances of object classes are simplified
- aggregation rules: how instances of object classes are merged, and
- representation rules: how instances of object classes are represented.

2.3 Surveying Rules and Context

The production of a geographic data set is done within a *context*, depending on the discipline of the user. Each discipline has its own definitions of object classes, and its attributes. Definitions depend on the aggregation level used: local, regional, national, etc. Each level has different terrain objects, which may be composites at another level, depending on the type of use: analysis, planning, or design (Molenaar 1998, p.157). [8]

This notion of context is broad. However, in this research the concept of context has a specific meaning. Surveying rules contain *additional conditions*, which are *not necessarily dependent* on properties of terrain objects *per se*, but also *on the situation* in the terrain, that is to say, relationships between terrain objects; for example, how far are terrain objects apart, or what kind of terrain objects are adjacent to each other? Consequently, context is determined by thematic, geometric, and topologic properties of possibly multiple terrain objects, and surveying rules are *context dependent*.

For example, two buildings in the terrain, less than two meters apart, may be acquired and represented as *one* single building instance in a data set; or a terrain situation, with sidewalks between flowerbeds, may be aggregated into *one* single composite flowerbed instance.

2.4 The Construction of a Domain Ontology for Topographic Mapping

In Section 2.1.3 and Section 2.2 it was argued that we need a domain ontology for geographic data set integration. This domain ontology should be 'rich' enough; that is to say should contain enough concepts for interconnecting different application ontologies.

---

[8] A formal context is defined as a triple (O, A, I) where O and A are sets and I is a binary relation between O and A: $I \subseteq O \times A$. Elements of O and A are respectively object classes and attributes (Wille 1992).

There is an official Dutch standard for topographic data set transfer, called *Geo-Information Terrain Model* (GTM) that pretends to be such a 'vehicle' (Ravi 1995). Let's see if elements of the GTM Standard are suitable for the construction of a domain ontology for topographic mapping.

2.4.1 GTM Standard

The subtitle of the GTM Standard puts the GTM in the position of a *classification*: "Terms, definitions and general rules for the classification and coding for earth related spatial objects". According to the Foreword the ultimate goal of the GTM is a *general* classification for the *transfer* of geo-information between organizations, such as municipalities, water boards, and electricity companies. It looks for a balance between a general, global approach versus a more specific approach in the description of geo-data set objects, with a tendency to a more global approach. Furthermore, the GTM is *terrain* related, not *map* related (van der Schans 1994). A terrain related description concentrates on the terrain and its geometric and non-geometric characteristics, independent of its future map representation. In addition, the focus of the GTM is *object-structured*, which means that recognizable objects in the terrain serve for the demarcation of listed elements in the classification. GTM defines an object as a 'phenomenon in the terrain that exists *independently* of other phenomena that can be recognized separately'. The level of detail of objects is in particular determined by the physical discernibility in the terrain (for example, *building* instead of *dwelling*).

2.4.2 GTM Standard as a Domain Ontology

Section 2.1.2 offered an operational definition of an ontology. This definition contained four items. These four items are summed up for the GTM Standard:

1. The GTM Standard is a collection of concepts.
2. GTM Standard concepts are defined in natural-language terms (for example, a 'road' is 'a leveled part for traffic on land').
3. The collection of concepts in the GTM Standard is limitative.
4. The GTM Standard has structure (concepts are classified into object classes; object classes belong to groups; every object class has a fixed set of attributes, with every attribute having a domain with values).

Based on the previous criteria we conclude that *the GTM Standard is an ontology*. In addition, the GTM Standard is related to the traditional discipline of topographic mapping and land surveying, therefore it is a *domain* ontology.

A critical issue is that definitions of GTM concepts are given in natural-language terms. Such definitions might lead to ambiguity. For example, the previous definition of 'road' does not give a clue for the lateral extension of a road: is a verge part of the road?

2.4.3 GTM Standard and its Usefulness for Data Set Integration

In Section 2.4.2 it was demonstrated that the GTM Standard is a domain ontology. How useful is this domain ontology for the integration of topographic data sets?

The GTM Standard originated within the professional circle of land surveyors. Therefore, the GTM Standard has a sufficient number of concepts for topographic data sets. These concepts are divided in object classes, with a sufficient number of attributes.

When two or more topographic data sets are integrated, most of the time not all possible topographic object classes are represented. Therefore one does not need all object classes from the GTM Standard. The same reasoning applies for the collection of attributes involved: while the GTM Standard has many attributes for a single object class, the number of attributes of a single object class in a data set is usually much less.

Furthermore, the GTM Standard has a global overall structure. The structure reflects the dominating view point of the Real World as a surface *divided* by road networks, railway networks, and water networks, with 'otherland' (= the rest of the Real World) in-between these networks. Road networks, railway networks, water networks, and 'otherland' can further be described in greater detail.

Differences in data sets are caused by differences in abstraction. Therefore, our conclusion is that the GTM Standard is useful for integration, provided we are able to:

1. define *subclasses*, possibly to the level of *data classes*, to express differences in abstractions between data sets (Section 2.5.1), and
2. add structure that reflects *compositions* in the data sets involved (Section 2.5.2).

Keeping these issues in mind brings us to the *construction* of reference models, where data sets get their semantic transparency.

2.5 The Construction of a Reference Model

In order to integrate different geographic data sets a *reference model* is constructed:

1. Object classes in a reference model are a *subset* of object classes from a domain ontology. This subset is determined by the geographic data sets to be integrated. Object classes from this subset are *refined* into *subclasses*. This refinement is also determined by the geographic data sets to be integrated (Section 2.5.1).
2. Object classes from this subset are *refined* into *subclasses* in a *taxonomy classification*. More structure is added to the reference model if object classes from different application ontologies are *composed* of each other. Then this composition is expressed as a *partonomy classification* in the reference model (Section 2.5.2).
3. Relationships between reference model object classes, and application ontologies object classes, define the semantics of geographic data sets. The basic

relationship, between a reference model class and a geographic data set class, is introduced in Section 2.5.3.

4. Relationships between object classes from different application ontologies are defined in Section 2.5.4. Three types of semantic relationships are defined: semantic equivalent, semantic related, and semantic relevant.
5. Finally, attention is given to special situations in the construction of a reference model: missing object classes (Section 2.5.5 and Section 2.5.7), and object class instances, acquired in parts (Section 2.5.6).

2.5.1 Object Classes for a Reference Model

Selection of object classes for a reference model depends on object classes in application ontologies. Surveying rules determine relationships between object classes from domain ontology (and, therefore reference model), and object classes from application ontologies. As was mentioned in Section 2.3, surveying rules are context dependent, that is to say dependent upon thematic, geometric and topologic properties of multiple terrain object instances. To avoid an explosion in the number of object classes in the reference model, context information is as much as possible excluded from the definition of these classes.

Therefore, the approach in this research is to include in the reference model information from surveying rules to the level of *data classes* (Molenaar 1998). Data classes are created by making *discrete* the value of an attribute by choosing useful limits. For example, domain object class ‘road’ is refined into three data classes: roads with (a) tracks ≤ 2 meters wide, (b) tracks 2 to 4 meters wide, and (c) tracks > 4 meters wide. Or, a characteristic attribute is chosen, like ‘free standing annex’ versus ‘adjacent annex’.

Excluding context from the reference model has the advantage, that it is easier to adapt a reference model, if we want to integrate another data set, with different context dependent surveying rules. Another advantage of controlling the number of classes is surveyability, to take in at a glance relationships between reference model and application ontologies (Artale et al 1996).

However, excluding context requires consistency checking of corresponding object instances (see Section 2.9).

2.5.2 Basic Structures in a Reference Model

As was mentioned before, two a*bstraction mechanisms* are fundamental in the production of geographic data sets:

- there is a generalization/specialization classification, which means that classes are grouped into a *taxonomy* with *superclasses* and *subclasses* (**Fig. 6**).
- there is a composite/component classification, which means that classes are grouped into a *partonomy*, with *composite* and *component classes* (**Fig. 7**).

In this research it is assumed that:

1. A partonomy has a *two-level* composite/component structure.
2. Component classes are *optional*: 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 context.
3. Component classes are *non-exclusive*, therefore can be shared by other composite classes.

Both classifications — taxonomy and partonomy — are basic structures for reference models, and combined into a tree-like structure. In Chapter 3 this tree-like structure will be defined as a finite directed graph.

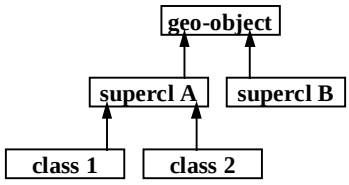


Fig. 6. A *taxonomy* as a basic structure for a reference model. Supercl A has classes 1 and 2 as subclasses. Supercl B has no subclasses. Both supercl A and B are subclasses from class geo-object.

Note. A *single headed solid arrow* denotes the subclass/superclass relationship.

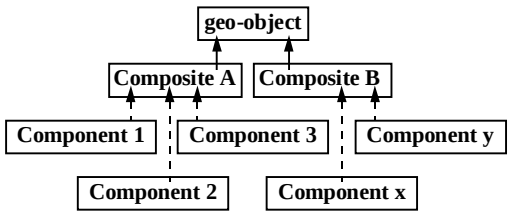


Fig. 7. A *partonomy* as a basic structure for a reference model. Composite class A has component classes 1, 2, and 3 as its constituents; composite class B has component classes x and y as its constituents (composite A and composite B are *subclasses* from class geo-object).

Note. A *single headed dashed arrow* denotes the component class/composite class relationship.

Two basic relationships are defined within the reference model. Assume a reference model A with its finite set of class labels: *a*1, *a*2, *a*3, etc.

Definition 1a. The basic taxonomy relationship, abbreviated as *taxon*, is between a subclass *a*1 and its superclass *a*2, within a reference model:

taxon(SubClass *a*1, SuperClass *a*2)	**(1)**

The *taxon* relationship is represented in **Fig. 6** with a single headed solid arrow.♦

Definition 1b. The basic partonomy relationship, abbreviated as *parton*, is between a component class *a*3 and its composite class *a*4, within a reference model:

parton(Component Class *a*3, Composite Class *a*4)	**(2)**

The *parton* relationship is represented in **Fig. 7** with a single headed dashed arrow.♦

2.5.3 Reference Models and Semantic Relationships

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

Definition 1c. The basic semantic relationship, abbreviated as *refers_to*, is between a reference model object class *a* (RefClass *a*), and an application ontology object class *b* (AppClass *b*):

refers_to(RefClass *a*, AppClass *b*)	**(3)**

The *refers_to* relationship is represented in **Fig. 8** with a double headed solid arrow. ♦

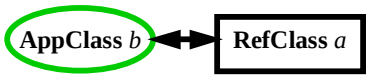


Fig. 8. The single basic semantic relationship *refers_to*(*a, b*). Observe that a *rectangle* denotes a reference model object class, an *oval* an application ontology object class, and a *double headed solid arrow* the *refers_to* relationship.

Within a data set, we impose a *many-to-one* integrity constraint on the *refers_to* relationship: for a given reference model object class *a*, there is at most one application object class that satisfies the relationship (**Fig. 9**, *left*), but for a given application object class *b*, there may be more reference model object classes *a1*, *a*2, etc, satisfying the relationship (**Fig. 9**, *right*).

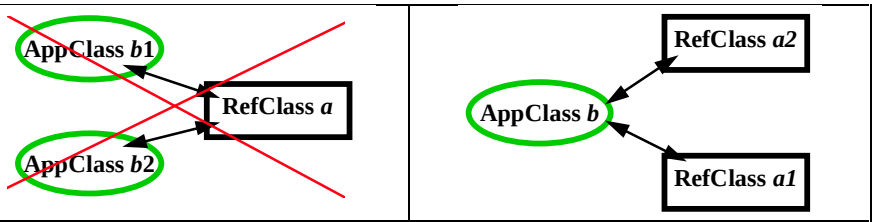


Fig. 9. A many-to-one constraint is imposed in referring from a reference model object class to an application ontology object class.

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.

2.5.4 Relationships between Application Ontologies

With the *refers_to* relationship, we define *relationships between object classes from different application ontologies.* These relationships determine the *semantics* of our universe of discourse in geo-data set integration.

Assume two geo-data sets B and C, with class label sets *B* and *C* respectively, and their reference model A, with class label set *A*. Let *b* a class label from *B*, *c* a class label from *C*, and *a*1, *a*2, *a*3, etc class labels from *A*.

Then three relationships are defined: [9]

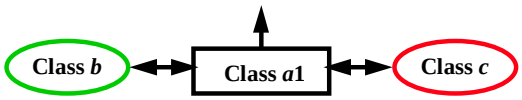


Fig. 10. The semantic equivalent relationship.

Definition 2 (see **Fig. 10**). There is a relation *semantic equivalent* (*Sequi*) between class *b*, and class *c*, if there exists a class *a*1, such that class *a*1 refers to *both* classes *b* and *c*:

$$Sequi = \{ (b,c) \in B \times C \mid \exists a1 \in A \ni refers\_to(a1,b) \wedge refers\_to(a1,c) \} \blacklozenge \quad \textbf{(4)}$$ [10]

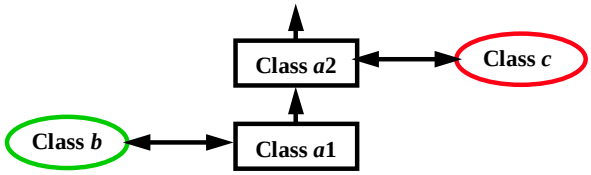


Fig. 11. The semantic related relationship.

Definition 3 (see **Fig. 11**). There is a relation *semantic related* (*Srla*) between class *b*, and class *c*, if there exist classes *a*1 and *a*2, such that class *a*1 refers to class *b*, and class *a*2 refers to *c*, with *a*1 a *subclass* of *a*2:

$$Srla = \{ (b,c) \in B \times C \mid \exists\, a1, a2 \in A \ni refers\_to(a1,b) \wedge refers\_to(a2,c) \wedge taxon(a1,a2) \} \blacklozenge \quad \textbf{(5)}$$

Note that in **(5)** the semantic related relationship is defined with the *taxon* subclass/superclass relationship. Semantic related relationships also hold between *other* levels of a taxonomy.

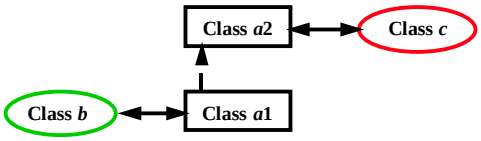


Fig. 12. The semantic relevant relationship (1).

[9] The terminology is from (Sheth and Kashyap 1993).
[10] ∋ denotes 'such that'.

Definition 4. There is a relation *semantic relevant* (*Srle*1 respectively *Srle*2) between class *b*, and class *c*, if:

(1) there exist classes *a*1 and *a*2, such that class *a*1 refers to class *b*, and class *a*2 refers to class *c*, with class *a*1 a *component class* of class *a*2 (see **Fig. 12**):

$$Srle1 = \{(b,c) \in B \times C \mid \exists\, a1, a2 \in A \ni refers\_to(a1,b) \wedge refers\_to(a2,c) \wedge parton(a1,a2)\} \quad \textbf{(6)}$$

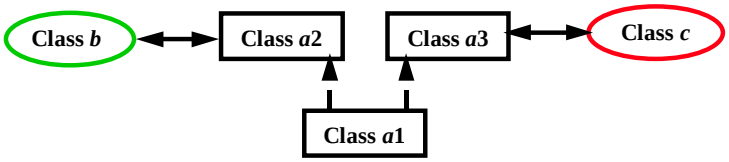


Fig. 13. The semantic relevant relationship (2).

(2) there exist classes *a*1, *a*2, and *a*3, such that class *a*2 refers to class *b*, and class *a*3 refers to class *c*, with class *a*1 a component class of both classes *a*2 and *a*3 (see **Fig. 13**):

$$Srle2 = \{(b,c) \in B \times C \mid \exists\, a1, a2, a3 \in A \ni refers\_to(a2,b) \wedge refers\_to(a3,c) \wedge parton(a1,a2) \wedge parton(a1,a3)\} \quad \textbf{(7)}$$ ♦

From now on we speak of *semantically similar classes*, or *compatible classes* whenever classes are *semantic equivalent, semantic related, or semantic relevant*, as defined in this section. Classes that are *not* semantic equivalent, semantic related, or semantic relevant are defined as *semantically non-similar* classes, or *incompatible* classes.

The three relations — 'equivalent', 'related', and 'relevant', with in addition 'incompatible' — are not always *disjoint* sets, that is to say do not always form a partition. That means, an element $(b, c) \in B \times C$ might sometimes belong to different relationships. This depends on the *role* an object class may have *in confrontation with* a different data set (more on roles in Section 4.5.1). However, between object classes from different data sets there are not more than three relationships. For a proof see Appendix B.

2.5.5 Object Classes Acquired for a Single Data Set

Object classes, exclusively acquired for a single data set, require special attention in the construction of a reference model. Apparently this object class does not exist for the other data set: during its acquisition it was 'filled in', or substituted by surrounding, adjacent object classes.

Here is an analogy with component classes. Component classes 'disappear' in composite classes. Therefore, if an object class in one data set is not acquired for the

other data set, then this object class is modeled as a component class of 'surrounding' object classes. See **Fig. 14**.

Here, as an example class *b* of data set B might be a building subclass class *a*1, which is not acquired for data set C, because its area size is less than a certain limit. To model this situation, class *a*1 refers to class *b*, with class *a*1 as component class of a composite class *a*2 that refers to land use class *c* of data set C, its 'surroundings'. *b* and *c* are therefore semantic relevant classes, according to **Definition 4 (1)**.

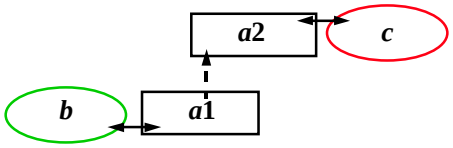


Fig. 14. Class *a*1, not acquired for data set C, becomes a component class of class *a*2.

2.5.6 Instances of Domain Classes Acquired in Parts

Another special case in the construction of reference models is the modeling of terrain situations, where instances of domain object classes are acquired in *parts*. To illustrate this point see **Fig. 15**.

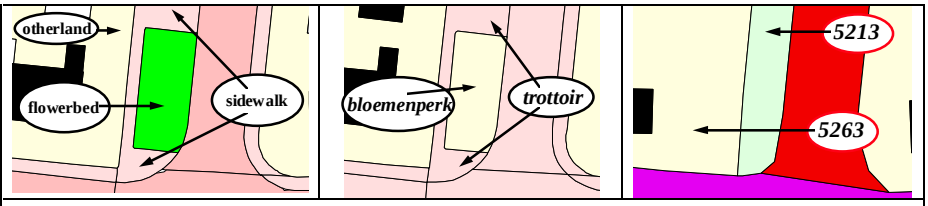


Fig. 15. A real-world situation where instances are acquired in parts.

Here an instance of 'sidewalk' (**Fig. 15**, *left*) is acquired as *one* instance *trottoir* in a data set (**Fig. 15**, *middle*). However, in acquiring this instance of 'sidewalk' for another data set, it is first divided into *parts*, *depending on context*. Parts are then combined with different domain classes. For example, a part of instance 'sidewalk' is combined with an instance of domain class 'flowerbed' (**Fig. 15**, *left*), and acquired as an instance *5213* (**Fig. 15**, *right*); another part of instance 'sidewalk' is combined with an instance of domain class 'otherland' (**Fig. 15**, *left*), and acquired as an instance *5263* (**Fig. 15**, *right*).

The solution for this situation is to model domain class 'sidewalk' in the reference model as a component class of both object classes involved. See **Fig. 16**. Here 'sidewalk' is a component class of both 'T_grass', which refers to class *5213*, and component class of 'T_other', which refers to class *5263*. *Trottoir* and *5213*, respectively *trottoir* and *5263* are therefore semantic relevant object classes, according to **Definition 4 (1)**.

The reference model construct in **Fig. 16**, where a component class is a constituent to more than one composite class, indicates a type of domain object class, where a *part* is of the *same kind of thing as its whole* (Artale et al 1996). For example, a part of 'sidewalk' is also a 'sidewalk'. We shall denote this type of class as *homogeneous decomposable* (more will be said about this in Section 2.9). This also demarcates the transition of the structure of the reference model from a *tree* into a *directed graph* (see Section 3.2).

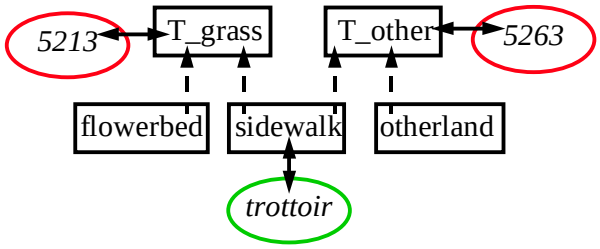


Fig. 16. Domain ontology class 'sidewalk' as a component class of two composite classes, 'T_grass', and 'T_other'.

2.5.7 Object Classes Not Represented in Both Data Sets

Usually there are many domain ontology object classes, which are not represented in both data sets. However, sometimes this situation needs attention in the construction of reference models.

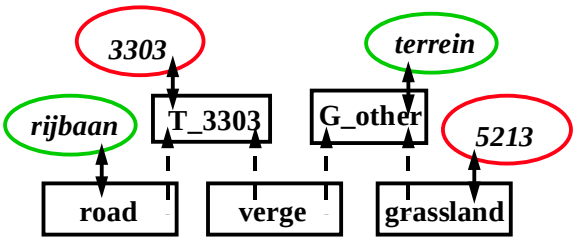


Fig. 17. Domain ontology class 'verge', represented in none of the data sets.

For example, a domain class 'verge' is in one data set combined with an (adjacent) domain class 'road' into a composite road class *3303*, and in another data set combined with an (adjacent) domain class 'grassland' into a land use class *terrein*. Hence, 'verge' will not be represented as an independent object class in one of the data sets. See **Fig. 17**. Here, the solution for this situation is to model 'verge' as a component class of both composite road class 'T_3303' (which refers to *3303*) and composite land use class 'G_other' (which refers to *terrein*). According to **Definition 4 (2)**, application classes *3303* and *terrein* in **Fig. 17** are therefore relevant to each other. Indeed, if instances of *3303* and *terrein* overlap geometrically

(where 'verge' is located), part of an instance of *3303* is part of an instance of *terrein*, and vice versa.

2.6 An Ontology-Based Conceptual Framework for Integration

Concepts introduced so far — domain ontology, application ontology, abstraction rules, surveying rules, reference model, and semantic relationships — are now configured into a *conceptual framework for ontology-based geographic data set integration*.

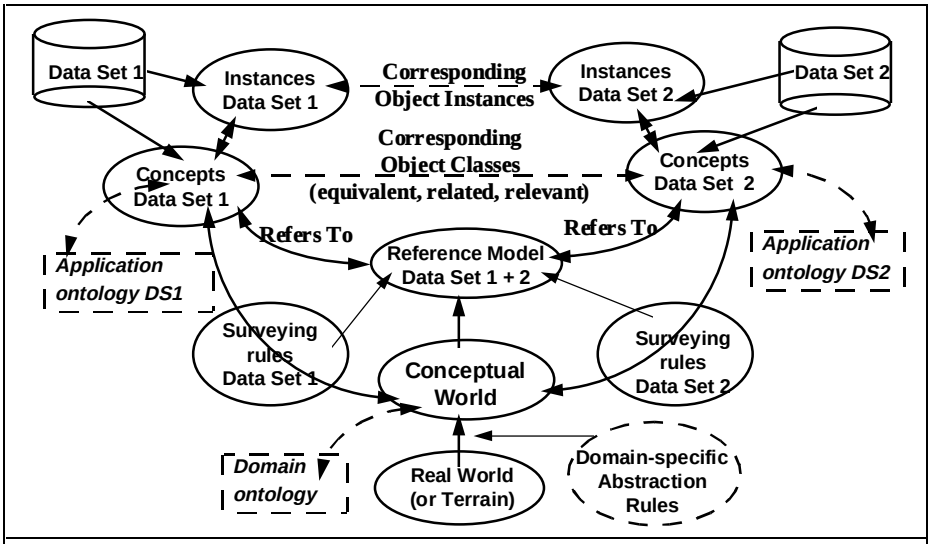


Fig. 18. An ontology-based framework for geographic data set integration.

- Upper-left and upper-right in **Fig. 18** are geographic data sets to be integrated ('Data Set 1' and 'Data Set 2'). Both data sets have their populations ('Instances Data Set 1' and 'Instances Data Set 2') and their concepts ('Concepts Data Set 1' and 'Concepts Data Set 2'), which are defined, and documented in application ontologies ('Application ontology DS1' and 'Application ontology DS2').
- Surveying rules capture relevant object classes for an application ('Surveying rules Data Set 1' and 'Surveying rules Data Set 2' in **Fig. 18**). Surveying rules are expressed between domain ontology object classes, and application ontology object classes.
- A reference model is constructed based on domain ontology object classes, information from surveying rules, and application ontologies object classes ('Reference Model Data Set 1 + 2' in **Fig. 18**).
- The semantics of data sets is defined by relationships between reference model object classes, and data set object classes ('Refers To' in **Fig. 18**).

- At the bottom of **Fig. 18** is the Real World (or Terrain). From this terrain, real-world phenomena, of interest, with certain properties, are grouped by abstraction rules in a class (the *principle of abstraction* (Lipschutz 1976)), defined as object classes in a conceptual world, and documented in a 'domain ontology'.

2.7 A Definition of Geographic Data Set Integration

In the preamble of Chapter 1 a definition of geographic data set integration was given:

Definition 5. '*Geographic data set integration*' is the process of establishing explicit relationships between corresponding object instances in different, autonomously produced, geographic data sets of the same geographic space. ♦

Definition 6. '*Corresponding object classes*' are object classes from *different* application ontologies, which are *semantic equivalent*, *semantic related*, or *semantic relevant*. ♦

A consequence of **Definition 6** is that corresponding object classes are identical with *semantically similar classes*, or *compatible classes*. Non-corresponding classes, or semantically non-similar classes are *incompatible classes*.

Definition 7. '*Corresponding object instances'* are object instances:

1. from corresponding object classes,
2. sharing same location, and
3. consistent with surveying rules. ♦

Note that 'corresponding object classes' and 'corresponding object instances' are schematically indicated in **Fig. 18**.

In the next sections, 'location' (item 2 in **Definition 7**), and 'consistent with surveying rules' (item 3 in **Definition 7**) of 'corresponding object instances' (or *correspondences* for short) are investigated.

2.8 Location in Geographic Data Set Integration

'Location' refers to the *geometric attribute* of an object instance in a geographic data set. In two-dimensional geo-data sets there are three types of geometric attributes: (a) a point, (b) a line, or (c) an area attribute. In this research, objects with area attributes are studied. An area attribute is a *polygon* with at least three *vertices*, which describe (a) the outer boundary, and if present (b) one or more inner boundaries (a polygon with one or more holes).

In this research, 'sharing same location' in **Definition 7**, is made operational by the *geometric overlap* between different polygons (see for example **Fig. 19**). The justification for choosing 'geometric overlap' as 'same location' lies in the precision

and accuracy of topographic data sets, together with the non-displacement property of object instances (Section 1.9.2).

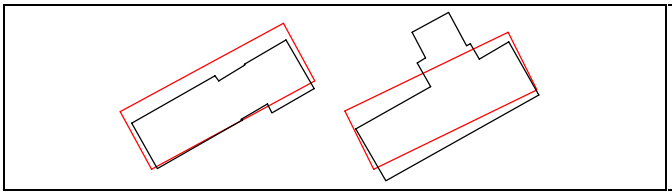


Fig. 19. Object instances from different topographic data sets (buildings, in *black* and *red*) sharing same location.

Note that choosing 'geometric overlap' for 'same location' removes in a certain sense *stochasticity* from data sets. Any amount of overlap is now sufficient to declare semantically similar object instances as *candidates* for corresponding instances. This removal of stochasticity is only temporarily. Stochasticity is introduced again in consistency checking.

Detecting overlap between object instances of different data sets is done by an *geometric overlay* operation (van Oosterom 1994). In Section 3.6 it is demonstrated how information from an overlay operation is used in finding candidates for correspondences.

2.9 Consistency Checking

The notion '*consistent with surveying rules*' in **Definition 7** is fundamental in this research, because inconsistencies should be solved before update propagation can happen.

The assessment of consistency is feasible, because it is assumed that in geo-data set integration, geographic data sets have well-defined sets of surveying rules. C*onsistency* means that corresponding object instances do not *contradict* any of the surveying rules.

2.9.1 The Motivation for Consistency Checking

Object classes from different data sets get their semantic similarity through the reference model. The construction of the reference model is based on surveying rules. If this construction is done correctly, then consistency is guaranteed at the class level (item 1 in **Definition 7**). However, consistency checking is also done at the instance level, for the following reasons:

1. Surveying rules contain additional conditions, which are context dependent. As was indicated in Section 2.5.1, context information is as much as possible excluded from object classes in the reference model. Therefore, this context information should be introduced again, if candidates for correspondences, based on reference model information, have been determined.
2. As was indicated in Section 2.8, choosing 'geometric overlap' for 'same location' removed stochasticity from data sets. This removal of stochasticity is only temporarily. Stochasticity should be introduced again, if candidates for correspondences, based on overlap information, have been determined.

2.9.2 Simple and Complex Correspondences

Instances in correspondences come in groups, or clusters, for the following reasons:

1. A group of instances may be components of a composite instance (according to **Definition 4**), or
2. Homogeneous decomposable object classes (see Section 2.5.6) may have instances, which are demarcated in an arbitrarily fashion, therefore creating groups of instances in correspondences.

Therefore, we define correspondences as *simple*, or *complex*:

- a *simple* set of corresponding object instances, or a *simple correspondence*, consists of a *pair* of corresponding object instances, that is to say a *1-to-1* correspondence, and
- a *complex* set of corresponding object instances, or a *complex correspondence*, is a *n*-to-*m* correspondence, with $(n \geq 1 \wedge m > 1) \vee (n > 1 \wedge m \geq 1)$.

Usually, simple correspondences are from semantic equivalent, or semantic related object classes, and complex correspondences are from semantic relevant object classes. A minimum effort in consistency checking is needed for simple correspondences. More effort is needed for complex correspondences, as is demonstrated in the next section.

2.9.3 Consistency Checking of Complex Correspondences

Globally, there are two ways to be more specific in statements about consistency of complex correspondences:

1. in a pre-processing step, break down object instances into *uniform elements* (Section 2.9.3.1), or
2. in a post-processing step, break down complex correspondences into *least common elements* (Section 2.9.3.2).

2.9.3.1 Demarcation of Uniform Elements

The idea here is, before overlaying data sets, to pre-process object instances in order to demarcate them into uniform elements, in such a way that, after overlaying data sets, simple (= 1-to-1) correspondences can be established. Before pre-processing, uniform elements should be defined in an unambiguous manner. For road networks there are definitions for ‘road segments’ and ‘road junctions’ in (Heres et al 1997). A method used in finding these ‘road segments’ and ‘road junctions’ automatically is based on *triangulating* the road network. In (Uitermark et al 1999b) *a constrained Delauney triangulation* is used to compute a skeleton of the road network. The nodes in the skeleton define the location of junctions. Edges of surrounding triangles are used to separate the road network into ‘road segments’ and ‘road junctions’. The

method is a useful basis for the demarcation of uniform elements, however, improvements are needed to make the method more robust.

2.9.3.2 Demarcation of Least Common Elements

Here the idea is, after overlaying data sets, and after finding candidates for correspondences, to post-process complex correspondences in order to demarcate them into *least common elements*. Least common elements are intuitively understood as the overlapping intersection between object instances from different data sets. For example, if an object instance in one data set represents streets A + B, and another object instance, in another data set, represents streets B + C, then their overlapping intersection — street B — is a least common element of both object instances.

A starting point for demarcating least common elements are *faces*, generated by overlaying data sets. However, due to *imprecision* faces might be *approximations* for these elements, and something has to be done to construct the desired least common elements (**Fig. 20**).

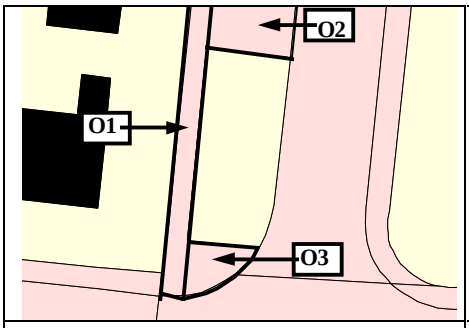
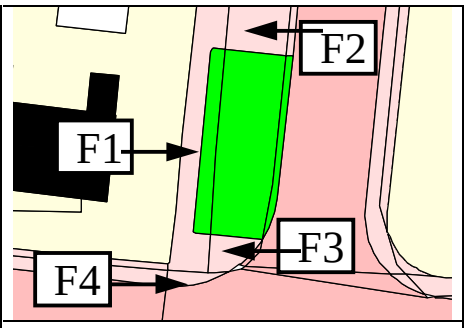


Fig. 20. Real-world ‘sidewalk’ in **Fig. 15** has for example {O1, O2, O3} as least common elements ...



... while faces {F1, F2, F3, F4}, generated by an overlay operation, are *approximations* for these elements.

2.9.4 A Definition of Consistency

Corresponding object instances are defined as ‘consistent with surveying rules’. Consistency implies possible real-world situations that are correctly represented by corresponding object instances. Or more formally, if data sets are consistent, we cannot refute possible real-world situations, represented by data sets.

Candidates for correspondences are detected by reference model, and overlay operation. Then, in order to decide if candidates are consistent, we have to take additional conditions from surveying rules into account. Therefore, a definition of ‘consistency’ is given in terms of reference model classes, overlap, and additional conditions from surveying rules.

Additional conditions are expressions with thematic, geometric, or topologic attributes. For example, ‘situated in urban region’ is a thematic attribute, ‘area size

≥ 9 square meters' is a geometric attribute, and 'adjacent to road' is a topologic attribute. Additional conditions imply additional criteria, whether a terrain object should be considered as member for a certain object class, its class *intension*. After this decision — the actual application of surveying rules — the *extension* of an object class is the set of all its members (Molenaar 1998). Consistency checking can be made operational by a test, whether different extensions of candidates satisfy *both* intensions.

If we break down complex candidates into simple candidates, that is to say 1-to-1 candidates, whether its components are object instances, uniform elements, or least common elements, then our definition for 'consistency' is as follows:

Definition 8. Let ($b1$, $c1$) be a simple candidate, with class labels b and c, respectively, *i.e.* b and c are corresponding classes, and $b1$ and $c1$ overlap each other. Then ($b1$, $c1$) is consistent, if both $b1$ and $c1$ satisfy intensions of class b and class c. ♦

In Chapter 5 this definition will be used in consistency checking of candidates for correspondences.

2.10 Discussion

This chapter presented a conceptual framework for geographic data set integration. Starting point in this framework is a mechanism to express meaning of geographic data sets in a language of shared concepts, a domain ontology. With references from concepts in data sets to concepts in a domain ontology, semantic matching is accomplished.

Concepts of a domain ontology are structured in a reference model to express levels of abstraction between data sets. The approach in this research is to construct a reference model in such a way that it gives specific information about semantic relationships between classes of different data sets. A consequence of this approach is that most object instances will be involved in some correspondence relationship, even with domain object classes that are acquired for a single data set (Section 2.5.5). Therefore, if certain object instances — so-called *singletons* — do not take part in a correspondence with other object instances then these singletons indicate most probably *surveying rule errors*. More will be said about singletons in Part 3.

However, while the reference model has specific information about semantic relationships between object *classes* of different data sets, this does not guarantee that corresponding object *instances* are consistent. As was mentioned previously, additional conditions in surveying rules — conditions often depending on context — are as much as possible excluded from the definition of object classes for the reference model.

In theory, it is possible to adjust the reference model for every additional condition by adding new object classes to the reference model. But in practice, apart from the fact, that this obscures the overview of the reference model (its 'surveyability'), it implies (a lot of) preprocessing of data sets. All attributes,

relevant in additional conditions — also attributes depending on context properties — have to be computed in advance.

This choice of computing implicit and context-sensitive attributes in advance, or not, is comparable to the choice between demarcating uniform elements, or least common elements (Section 2.9.3). Here, differences are:

- uniform elements can be computed independent from other data sets. Contrarily,
- least common elements mean computing elements and attributes 'on the fly', that is to say when they are needed.

The determination of advantages and disadvantages of both methods needs further research.

A final remark on *acquisition* versus *integration*. In acquiring a geo-data set, relevant terrain objects are defined, not necessarily based on an explicit domain ontology. Then, surveying rules are defined in terms of these relevant terrain objects, and terms of geo-data set classes. After that, the acquisition of the data set starts. This process looks like a *top-down* approach.

In geo-data set integration, starting point for integration are geo-data sets, and sets of surveying rules, not necessarily complete, nor necessarily defined in terms of shared concepts. That means that a domain ontology has to be defined, and based on its definitions, a reference model is constructed. Therefore, geo-data set integration is more like a *bottom-up* approach. This explains 'Terrain' situated at the *bottom*, and 'data sets' situated at the *top* of **Fig. 18**, the overall diagram of a conceptual framework for ontology-based geographic data set integration.

3 Finding Semantically Similar Classes and Instances

In Chapter 2 a conceptual framework for geographic data set integration was presented. Part of this framework is the definition of *semantic similarity* between object classes from different data sets. This semantic similarity is based on references from domain ontology object classes — structured in a reference model — to object classes in different geo-data sets.

In this chapter a *set-theoretic approach* for expressing semantically similar object classes is presented. The motivation for a set-theoretic approach is its simplicity over, for example, predicate calculus. We use *sets of concept labels* of reference model and application ontologies in elementary *set expressions*, with *relations* defined between these sets. Then, it is possible to express semantically similar classes as relations. There is an introduction on this subject in Section 3.1.

In Section 3.2 the taxonomy/partonomy structure of the reference model is also treated as a relation. In Section 3.3 the notion of an *ordered pair* is used to determine semantically similar classes. Furthermore, there is a section, where a model for computing ordered pairs of semantically similar classes is presented (Section 3.4), and a section with a model for computing the *type* an ordered pair — semantic equivalent, semantic related, or semantic relevant (Section 3.5). Both models are determined with relations, represented as *matrices*.

If semantically similar classes are known, then this information is combined with information from overlapping object instances, in order to find *candidates for corresponding object instances* (Section 3.6).

3.1 Introduction to Set-Theoretic Concepts

Let's consider three finite sets *B*, *C*, and *A* with concept labels: [11]

– *B* is the set of concept labels from data set B application ontology:

$B = \{b \mid b \text{ is a concept label from data set B application ontology}\}$	**(8)**

– *C* is the set of concept labels from data set C application ontology:

$C = \{c \mid c \text{ is a concept label from data set C application ontology}\}$	**(9)**

– *A* is the set of concept labels from the reference model of data set B and C:

$A = \{a \mid a \text{ is a concept label from the reference model of B and C}\}$	**(10)**

Binary relations are defined between sets *A* and *B*, and sets *A* and *C*:

[11] A concept label (or class label) is a distinct term that refers to a particular concept.

– relation *R*, defined as a subset of $A \times B$:

$$R \subset A \times B \quad \textbf{(11)}$$

with *elements* of *R* having properties, that will be explained in Section 3.3, and

– relation *S*, defined as a subset of $A \times C$:

$$S \subset A \times C \quad \textbf{(12)}$$

with elements of *S* having similar properties as elements of *R* in **(11)**, also further explained in Section 3.3.

There are *inverse* relations R^{-1} and S^{-1}, defined as:

$$R^{-1} = \{ \quad (b,a) \quad | \quad (a,b) \in R \quad \}$$
$$S^{-1} = \{ \quad (c,a) \quad | \quad (a,c) \in S \quad \} \quad \textbf{(13)}$$

With inverse relations R^{-1} and S^{-1} in **(13)**, relations are *composed* between concept labels — from now on: labels — of *B* and *C*, or *C* and *B* (with $\circ$ as the compose operator):

$$R^{-1} \circ S = \left\{ \quad (b,c) \in B \times C \quad | \quad \exists\, a \ni (b,a) \in R^{-1} \wedge (a,c) \in S \quad \right\}$$
$$S^{-1} \circ R = \left\{ \quad (c,b) \in C \times B \quad | \quad \exists\, a \ni (c,a) \in S^{-1} \wedge (a,b) \in R \quad \right\} \quad \textbf{(14)}$$

The interpretation of **(14)** is that if relations are defined between labels of the reference model, and labels of different application ontologies, then the relation between labels of different application ontologies is also known. That means, *semantically similar classes* can be expressed as relations.

3.2 Relations between Reference Model Labels

However, in order to proceed in this straightforward manner we have to take into account that in this research labels of *A* are ordered in a taxonomy/partonomy structure, that is to say labels of *A* form a relation *T* with elements from product set $A \times A$.

This relation brings us to the introduction of a concept from graph theory, the finite directed graph.

Labels of the reference model (set *A*) are structured as in a *finite directed graph H*, or *digraph H*, with labels as *nodes*, and *parton* and *taxon* relationships as *edges*.

Definition 9. Directed graph *H* consists of:

- Nodes. Each node represents a label of *A*
- Edges. Each edge represents a *taxon* or *parton* relationship between labels of *A*

- A distinguished node, labeled *geo-object*, called *root.* ♦

Directed graph *H* consists of two subgraphs:
– the first subgraph is the taxonomy, a *tree*, and
– the second subgraph represents the partonomy.

Definition 10. If a *path* is an alternating sequence of nodes and edges, in which all nodes are distinct, then *level L* of label *n* is defined as the length of the path — that is to say the number of edges — from label *n* to root *geo-object.* ♦

If the partonomy subgraph is structured according to the restrictions we imposed in Section 2.5.2 — a two-level component/composite structure, with shared component classes — then every path from label *n* to root *geo-object* is also the shortest path. Hence, a label of set *A* belongs to exactly one level, and levels of set *A* form a partition. Labels of set *A* are partitioned into subsets A_L for every level *L* of set *A*:

$$A_L \subset A, \quad \bigcup_{L=0}^{m} A_L = A, \quad A_L \cap A_K = \varnothing \text{ if } L \neq K \qquad \textbf{(15)}$$

with *m* + 1 the number of levels of *H*.

Let's consider the relation between labels of *A* at two consecutive levels of directed graph *H*.

Every path of length 1 between labels at different levels can be considered as a binary relation *T*[*L*] from a subset of $A_L \times A_{L-1}$:

$$T[L] = \{(a1, a2) \in A_L \times A_{L-1} \mid taxon(a1, a2) \vee parton(a1, a2)\} \qquad \textbf{(16)}$$

with
– A_L, A_{L-1} subsets of *A* defined in **(15)**
– $L \in \{1, \ldots, m\}$, *m* + 1 the number of levels of *H*
– *taxon* the subclass/superclass relationship between labels at level *L* and level *L* – 1 (Section 2.5.2), and
– *parton* the component/composite class relationship between labels at level *L* and level *L* – 1 (Section 2.5.2).

3.3 Semantically Similar Labels as Ordered Pairs

Consequently, relations *R* and *S* from Section 3.1 are redefined as follows.

R[*L*] is a relation defined as:

$$R[L] = \{(a,b) \in A_L \times B \mid \exists\, a1 \in A_{L-1} \ni (refers\_to(a,b) \wedge taxon(a,a1)) \vee (refers\_to(a,b) \wedge parton(a,a1)) \vee (refers\_to(a1,b) \wedge parton(a,a1))\} \qquad \textbf{(17)}$$

with the same symbols used in **(16)**, and *refers_to* the relationship between a reference model object class and an application ontology object class (Section 2.5.2). Every relation $R[L]$ divides labels from set B in *overlapping*, possibly *empty* subsets B_L defined as $B_L = \{b \in B \mid (a,b) \in R[L]\}$ for $L \in \{1,\ldots,m\}$ of H.

In a similar manner $S[L]$ is defined as:

$$S[L] = \{(a,c) \in A_L \times C \mid \exists\, a1 \in A_{L-1} \ni (refers\_to(a,c) \wedge taxon(a,a1)) \vee (refers\_to(a,c) \wedge parton(a,a1)) \vee (refers\_to(a1,c) \wedge parton(a,a1))\} \quad \textbf{(18)}$$

Likewise, every relation $S[L]$ divides labels from set C in *overlapping*, possibly *empty* subsets C_L defined as $C_L = \{c \in C \mid (a,c) \in S[L]\}$ for $L \in \{1,\ldots,m\}$ of H.

The motivation for this definition of $R[L]$, as in **(17)**, and $S[L]$, as in **(18)**, comes from **Definitions 2** up to **4** of semantic similarity in Chapter 2. In **Fig. 21** (*left*) the taxonomy subgraph part of the reference model is depicted, with subclasses, and superclasses, accountable for semantic equivalent and semantic related relationships. In **Fig. 21** (*middle*, and *right*) the partonomy subgraph is depicted, with its component classes, and composite classes, accountable for semantic relevant relationships.

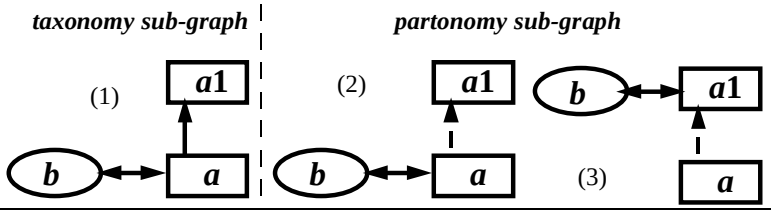


Fig. 21. (*a*, *b*) belongs to relation $R[L]$ iff (1) *a* refers to *b*, and *a* is a subclass of *a*1 (*left*), or (2) *a* refers to *b*, and *a* is a component class of *a*1 (*middle*), or (3) *a*1 refers to *b*, and *a*1 has *a* as component class (*right*).

Observe that in order to account for **Definition 4(2)** type situations, members of B (or C) become members of B_L (or C_L), if they have a reference to $a \in A_{L-1}$ (**Fig. 21**, *right*).

With the preceding relations $T[L]$, $R[L]$, and $S[L]$ we are able to formalize semantically similar labels between different application ontologies as *ordered pairs* (*b*, *c*). That means we can rewrite $R^{-1} \circ S$ in **(14)** in the following theorem:

Theorem 1. The set of ordered pairs of semantically similar labels (*b*, *c*) between data sets B and C, with label sets B and C is given by:

$$\bigcup_{L=1}^{m} \{ \bigcup_{K=1}^{m} (R[L]^{-1} \circ T \circ S[K]) \} \quad \textbf{(19)}$$

with:

– $\cup$ the union operator

– $L, K \in \{1, \ldots, m\}$, $m + 1$ the number of levels of directed graph H, with

– $$\begin{cases} T = \Delta_T \quad \text{(the identity relation in } T\text{)}, & (L = K) \\ T = T[L+1]^{-1} \circ T[L+2]^{-1} \circ \cdots \circ T[K]^{-1}, & (L < K) \text{, and} \\ T = T[L] \circ T[L-1] \circ \cdots \circ T[K+1], & (L > K) \end{cases}$$

– $T[L]$ the relation between level L and $L - 1$ in H as introduced in **(16)**.

For a proof of **Theorem 1** see Appendix B.

3.4 A Model for Computing Semantically Similar Classes

In order to make **Theorem 1** in **(19)** operational, a model for computing semantically similar class labels is developed. For that purpose we need a simple way of representing relations. *Matrices* are a good choice because then we can rely on matrix algebra.

First, matrix representations for relations are introduced (Section 3.4.1). Then it is shown how matrix multiplication is used in computing semantically similar classes (Section 3.4.2). Finally, all relations $R[L]$, $T[L]$, and $S[L]$, represented as matrices $\mathbf{R}[L]$, $\mathbf{T}[L]$, and $\mathbf{S}[L]$, with $L, K \in \{1, ..., m\}$, $m + 1$ the number of levels of the reference model, are regrouped into a single matrix expression (Section 3.4.3).

3.4.1 Relations Represented as Matrices

A binary relation can be represented as a matrix, for example relation $R[L]$ is represented as a rectangular array $\mathbf{R}[L]$, whose rows are labeled by the members of set A_L, and whose columns are labeled by the members of set B_L, where a *1* or *0* is put in each position of the array according to $a \in A_L$ is, or is not related to $b \in B_L$. Therefore, a 1 is put in $\mathbf{R}[L]$, whenever the relation $R[L]$ holds.

For example:

$$\mathbf{R}[L] = \left(\begin{array}{c|cccc} & b_1 & b_2 & \cdots & b_k \\ \hline a_1 & 1 & 0 & \cdots & 1 \\ a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_n & 1 & 0 & \cdots & 1 \end{array}\right) \qquad \textbf{(20)}$$

with

– $\mathbf{R}[L]$ the matrix of $R[L]$, a relation on subset $A_L \times B_L$

– n the number of labels of A_L, and

– k the number of labels of B_L

then, whenever there is a 1 in **(20)**, it relates to a member of subset $A_L \times B_L$.

Note that B_L is possibly empty, in that case $k = 0$, and $\mathbf{R}[L]$ does not exist.

The same reasoning applies to matrix $\mathbf{S}[L]$ of $S[L]$, a relation on subset $A_L \times C_L$.

3.4.2 Composition of Relations and Matrix Multiplication

Furthermore, we represent *composition of relations* as *matrix multiplication*. For example, $R^{-1} \circ T \circ S$ is represented as $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ (with $\mathbf{R}^T$ the transposed matrix of $\mathbf{R}$). Every non-zero entry of $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ tells us what labels are related by the composition of relations $R^{-1} \circ T \circ S$.

Therefore, ordered pairs of semantically similar classes (*b*, *c*) are found by matrix multiplication. For example, with

$\mathbf{R}[2]^T \cdot \mathbf{T}[3]^T \cdot \mathbf{S}[3]$	**(21)**

we find all ordered pairs (*b*, *c*) between level 2 labels of data set B, and level 3 labels of data set C.

In this way we are able to find pairs of labels that are semantically similar to each other at different levels of the reference model.

3.4.3 Computing Semantically Similar Classes

In order to compute *in one step* all semantically similar classes by matrix multiplication, all relations $R[L]$, $T[L]$, and $S[L]$, with $L \in \{1, ..., m\}$, $m + 1$ being the number of levels of the reference model, and represented as matrices $\mathbf{R}[L]$, $\mathbf{T}[L]$, and $\mathbf{S}[L]$, are regrouped into a single matrix expression:

Ordered pairs of semantically similar labels: $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$	**(22)**

with

- matrix **R,** representing all relations $R[L]$ between *A* and *B*,
- matrix **T**, representing all relations $T[L]$ between the levels of *A*, and
- matrix **S**, representing all relations $S[L]$ between *A* and *C*.

The regrouping is done as follows.

3.4.3.1 Regrouping Relations Between Reference Model Levels

In order to navigate through directed graph *H*, a matrix **T** will be constructed, with submatrices:

$\mathbf{T}[1], \mathbf{T}[2], \ldots, \mathbf{T}[L], \ldots \mathbf{T}[m-1], \mathbf{T}[m]$	**(23)**

as elements. These elements represent relations between consecutive levels in the reference model, with $L \in \{1, ..., m\}$, $m + 1$ the number of levels in directed graph H, as explained in Section 3.2.

Now we will describe the construction process of a square $n \times n$ matrix $\mathbf{T}$, with n the number of elements of set A.

First, there is *upward* propagation inside $\mathbf{T}$ from level L to level $L - 1$, from $L - 1$ to $L - 2$, and so on:

$$\mathbf{T} = \begin{pmatrix} . & . & . & . & . & . \\ \mathbf{T}[1] & . & . & . & . & . \\ . & \mathbf{T}[2] & . & . & . & . \\ . & . & \ddots & . & . & . \\ . & . & . & \mathbf{T}[m-1] & . & . \\ . & . & . & . & \mathbf{T}[m] & . \end{pmatrix} \quad \textbf{(24)}$$

Secondly, there is also propagation inside $\mathbf{T}$ in an *downward* direction from level L to level $L + 1$, $L + 1$ to level $L + 2$ and so on, with:

$$\mathbf{T}[1]^T, \mathbf{T}[2]^T, \ldots, \mathbf{T}[L]^T, \ldots \mathbf{T}[m-1]^T, \mathbf{T}[m]^T \quad \textbf{(25)}$$

as elements, the transposed matrix of every $\mathbf{T}[L]$. This is added to **(24)**:

$$\mathbf{T} = \begin{pmatrix} . & \mathbf{T}[1]^T & . & . & . & . \\ \mathbf{T}[1] & . & \mathbf{T}[2]^T & . & . & . \\ . & \mathbf{T}[2] & . & \ddots & . & . \\ . & . & \ddots & . & . & . \\ . & . & . & . & \mathbf{T}[m-1]^T & . \\ . & . & . & \mathbf{T}[m-1] & . & \mathbf{T}[m]^T \\ . & . & . & . & \mathbf{T}[m] & . \end{pmatrix} \quad \textbf{(26)}$$

Thirdly, whenever there are *equivalent* ordered pairs, then — upward or downward — propagation is not relevant. Therefore, we add a neutral element $\mathbf{I}$ to **(26)**:

$$\mathbf{T} = \begin{pmatrix} 1 & \mathbf{T}[1]^T & . & . & . & . \\ \mathbf{T}[1] & \mathbf{I} & \mathbf{T}[2]^T & . & . & . \\ . & \mathbf{T}[2] & \mathbf{I} & \ddots & . & . \\ . & . & \ddots & \ddots & \mathbf{T}[m-1]^T & . \\ . & . & . & \mathbf{T}[m-1] & \mathbf{I} & \mathbf{T}[m]^T \\ . & . & . & . & \mathbf{T}[m] & \mathbf{I} \end{pmatrix} \quad \textbf{(27)}$$

with $t_{11} = 1$ for the root *geo-object*.

And finally, there is propagation inside **T** between elements of set *B* at level *L*, and elements of set *C* at level *K*, with *L*, *K* ∈ {1, ..., *m*}:

$$\mathbf{T}[L,K] = \mathbf{T}[L] \cdot \mathbf{T}[L-1] \cdot \ldots \cdot \mathbf{T}[K+2] \cdot \mathbf{T}[K+1] \quad \textbf{(28)}$$

for *L* > *K*, and

$$\mathbf{T}[L,K]^T = \mathbf{T}[L+1]^T \cdot \mathbf{T}[L+2]^T \cdot \ldots \cdot \mathbf{T}[K-1]^T \cdot \mathbf{T}[K]^T \quad \textbf{(29)}$$

for *L* < *K*. This we put into **(27)**

$$\mathbf{T} = \begin{pmatrix} 1 & \mathbf{T}[1]^T & \mathbf{T}[1,2]^T & \cdots & \mathbf{T}[1,m-1]^T & \mathbf{T}[1,m]^T \\ \mathbf{T}[1] & \mathbf{I} & \mathbf{T}[2]^T & \cdots & \mathbf{T}[2,m-1]^T & \mathbf{T}[2,m]^T \\ \mathbf{T}[2,1] & \mathbf{T}[2] & \mathbf{I} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \ddots & \mathbf{T}[m-1]^T & \mathbf{T}[m-1,m]^T \\ \mathbf{T}[m-1,1] & \mathbf{T}[m-1,2] & \cdots & \mathbf{T}[m-1] & \mathbf{I} & \mathbf{T}[m]^T \\ \mathbf{T}[m,1] & \mathbf{T}[m,2] & \cdots & \mathbf{T}[m,m-1] & \mathbf{T}[m] & \mathbf{I} \end{pmatrix} \quad \textbf{(30)}$$

Matrix **T** in **(30)** is symmetric, where every non-zero entry t_{LK} $(= t_{KL})$, with $L \neq K$, is the number of paths from *L* to *K*, or *K* to *L* (see Appendix B for an explanation).

3.4.3.2 Regrouping Relations Between Reference Model and Set B

Matrix **R** is a matrix composed of every submatrix **R**[*L*] that represents the relation *R*[*L*], with *L* ∈ {1, ..., *m*}, *m* + 1 the number of levels in directed graph *H*. The dimensions of **R** are (*k* + 1) × *n*, with *k* the sum of elements of all subsets B_L , and *n* the number of elements of set *A*, the reference model labels. The first row of **R** contains zeros because by definition there is no *refers_to* relationship between root *geo-object* and set *B*:

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \mathbf{R}[1] & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{R}[2] & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \ddots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & \mathbf{R}[L] & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \mathbf{R}[m-1] & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \mathbf{R}[m] \end{pmatrix} \quad \textbf{(31)}$$

3.4.3.3 Regrouping Relations Between Reference Model and Set C

In a similar way as matrix **R** in **(31)**, matrix **S** is a matrix composed of every submatrix **S**[*L*] that represents relation *S*[*L*], with *L* ∈ {1, ..., *m*}, *m* + 1 the number

of levels in directed graph *H*. The dimensions of **S** are $(p + 1) \times n$, with p the sum of elements of all subsets C_L , and n the number of elements of set *A*, the reference model labels. The first row of **S** contains zeros because by definition there is no *refers_to* relationship between root *geo-object* and set *C*:

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \mathbf{S}[1] & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{S}[2] & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \ddots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}[L] & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \mathbf{S}[m-1] & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \mathbf{S}[m] \end{pmatrix} \quad \textbf{(32)}$$

With **(30)** up to **(32)** our expression in **(22)** is complete.

3.5 A Model for Computing Semantic Similarity Types

In this section it is shown how we can compute of an ordered pair of labels its *type* of semantic similarity, that is to say if corresponding object classes are semantic equivalent, semantic related, or semantic relevant.

Matrix **R** from Section 3.4.3.2 is divided into two matrices, under the law of matrix addition:

$$\mathbf{R} = \mathbf{R}rec + \mathbf{R}rcc \quad \textbf{(33)}$$

R*rec* in **(33)** is the matrix representation of relation *Rrec*, defined as:

$$Rrec = \{(a,b) \in A \times B \mid \exists\, a1 \in A \ni (refers\_to(a,b) \wedge taxon(a,a1)) \vee (refers\_to(a,b) \wedge parton(a,a1))\} \quad \textbf{(34)}$$

(*rec* is the abbreviation of *r*efers to *e*lementary *c*lass; elementary classes are subclasses, superclasses, or component classes).

R*rcc* in **(33)** is the matrix representation of relation *Rrcc*, defined as:

$$Rrcc = \{(a,b) \in A \times B \mid \exists\, a1 \in A \ni (refers\_to(a1,b) \wedge parton(a,a1))\} \quad \textbf{(35)}$$

(*rcc* is the abbreviation of *r*efers to *c*omposite *c*lass).

The splitting-up in **R***rec* and **R***rcc* is unambiguous, because there is at most one relationship from one label in the reference model (set *A*) to one label in the application ontology (set *B*), and this reference is either of type **(34)**, or of type **(35)** (see also equation **(17)** and **Fig. 21**).

The same splitting-up is applied to matrix **S** of Section 3.4.3.3:

$$\mathbf{S} = \mathbf{S}rec + \mathbf{S}rcc \quad \textbf{(36)}$$

S*rec* in **(36)** is the matrix representation of relation *Srec*, defined as:

$$Srec = \{(a,c) \in A \times C \mid \exists\, a1 \in A \ni (refers\_to(a,c) \wedge taxon(a,a1)) \vee (refers\_to(a,c) \wedge parton(a,a1))\} \quad \textbf{(37)}$$

S*rcc* in **(36)** is the matrix representation of relation *Srcc*, defined as:

$$Srcc = \{(a,c) \in A \times C \mid \exists\, a1 \in A \ni (refers\_to(a1,c) \wedge parton(a,a1))\} \quad \textbf{(38)}$$

Again, the splitting-up in **S***rec* and **S***rcc* is unambiguous, because there is at most one relationship from one label in the reference model (set *A*) to one label in the application ontology (set *C*), and this reference is either of type **(37)**, or of type **(38)** (see also equation **(18)** and **Fig. 21**).

Matrix **T** of Section 3.4.3.1 is divided into two matrices, under the law of matrix addition:

$$\mathbf{T} = \mathbf{I} + \mathbf{T}prop \quad \textbf{(39)}$$

with

- **I**, the diagonal of **T**, according to **(27)** the Identity Matrix, and
- **T***prop*, containing all off-diagonal elements of **T**. **T***prop* represents upward and downward *prop*agation along elements of the reference model.

We insert expressions **(33)**, **(36)**, and **(39)** into expression **(22)**:

$$\begin{aligned} &\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S} = (\mathbf{R}rec^T + \mathbf{R}rcc^T) \cdot (\mathbf{I} + \mathbf{T}prop).(\mathbf{S}rec + \mathbf{S}rcc) = \\ &\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec + \\ &\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec + \\ &\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc \end{aligned} \quad \textbf{(40)}$$

Then we assert the following theorems:

Theorem 2. *Semantic equivalent* ordered pairs of labels (*b*, *c*) are similar to:

$$\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec = \mathbf{R}rec^T \cdot \mathbf{S}rec \quad \textbf{(41)}$$

Theorem 3. *Semantic related* ordered pairs of labels (*b*, *c*) are similar to:

$\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec$	**(42)**

Theorem 4. *Semantic relevant* ordered pairs of labels (*b*, *c*) are similar to:

$\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$	**(43)**

Furthermore, semantic related ordered pairs of labels (*b*, *c*) in **(42)** can be broken down into two subsets, if we divide **T***prop* in:

$\mathbf{T}prop = \mathbf{T}propsper + \mathbf{T}propsub$	**(44)**

with in **(44)** **T***propsper* the upper-diagonal matrix of **T***prop*, and **T***propsub* the lower-diagonal of **T***prop*.

Theorem 5. Semantic related ordered pairs of labels (*b*, *c*) in **(42)**, where *b* is a *superclass* of *c*, are similar to:

$\mathbf{R}rec^T \cdot \mathbf{T}propsper \cdot \mathbf{S}rec$	**(45)**

Theorem 6. Semantic related ordered pairs of labels (*b*, *c*) in **(42)**, where *b* is a *subclass* of *c*, are similar to:

$\mathbf{R}rec^T \cdot \mathbf{T}propsub \cdot \mathbf{S}rec$	**(46)**

Finally, semantic relevant ordered pairs of labels (*b*, *c*) in **(43)** can be broken down into three subsets:

Theorem 7. Semantic relevant ordered pairs of labels (*b*, *c*) in **(43)**, where *b* is a *component class*, or a *constituent* of *c*, are similar to:

$\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$	**(47)**

Theorem 8. Semantic relevant ordered pairs of labels (*b*, *c*) in **(43)**, where *b* is a *composite class* of *c*, are similar to:

$\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec$	**(48)**

Theorem 9. Semantic relevant ordered pairs of labels (*b*, *c*) in **(43)**, where *b* is *composed* of *c*, and *c* is *composed* of *b*, are similar to:

$\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$	**(49)**

For a proof of **Theorem 2** up to **Theorem 9** see Appendix B.

The previous model for computing semantically similar classes is demonstrated in Chapter 5.

3.6 Finding Candidates for Corresponding Object Instances

Models for computing semantically similar classes were introduced in Section 3.4 and Section 3.5. In this section it is demonstrated how these models are used in finding candidates for corresponding object instances, the ultimate goal of geographic data set integration. First, overlapping object instances are identified (Section 3.6.1). Secondly, on the basis of this information, candidates are selected (Section 3.6.2). Thirdly, if all candidates are known, then object instances that have no possible correspondence with other object instances (*singletons*) are located (Section 3.6.3).

3.6.1 Finding Overlapping Object Instances

By *overlaying* the partition of data set B, with the partition of data set C, a new partition of *faces* is created. Every face *f* of this new partition has as elements:

$$\{fid, boid, coid\} \text{ with } \begin{cases} fid & \text{a } f\text{ace } id\text{entifier} \\ boid & \text{a data set B } o\text{bject } id\text{entifier} \\ coid & \text{a data set C } o\text{bject } id\text{entifier} \end{cases} \qquad \textbf{(50)}$$

From the faces of this new partition, a $nf \times nb$ matrix **FB** is set up, with *nf* the number of faces, and *nb* the number of data set B object instances.

Every element fb_{ij} of **FB** is 1 or 0, whether or not fid_i is part of $boid_j$:

$$\mathbf{FB} = \begin{array}{c|cccc} & boid_1 & boid_2 & \cdots & boid_{\text{nb}} \\ \hline fid_1 & fb_{11} & fb_{12} & \cdots & fb_{1\text{nb}} \\ fid_2 & fb_{21} & fb_{22} & \cdots & fb_{2\text{nb}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ fid_{\text{nf}} & fb_{\text{nf1}} & fb_{\text{nf2}} & \cdots & fb_{\text{nfnb}} \end{array} \qquad \textbf{(51)}$$

In a similar fashion a $nf \times nc$ matrix **FC** is set up, with *nc* the number of data set C object instances.

Every element fc_{ij} of **FC** is 1 or 0, whether or not fid_i is part of $coid_j$:

$$\mathbf{FC} = \begin{array}{c|cccc} & coid_1 & coid_2 & \cdots & coid_{\text{nc}} \\ \hline fid_1 & fc_{11} & fc_{12} & \cdots & fc_{1\text{nc}} \\ fid_2 & fc_{21} & fc_{22} & \cdots & fc_{2\text{nc}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ fid_{\text{nf}} & fc_{\text{nf1}} & fc_{\text{nf2}} & \cdots & fc_{\text{nfnc}} \end{array} \qquad \textbf{(52)}$$

Multiplying the transpose of **FB** with **FC** gives a $nb \times nc$ matrix **BC**:

$$\mathbf{BC} = (\mathbf{FB})^T \cdot \mathbf{FC} = \begin{array}{c|cccc} & coid_1 & coid_2 & \cdots & coid_{\mathrm{nc}} \\ \hline boid_1 & bc_{11} & bc_{12} & \cdots & bc_{1\mathrm{nc}} \\ boid_2 & bc_{21} & bc_{22} & \cdots & bc_{2\mathrm{nc}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ boid_{\mathrm{nb}} & bc_{\mathrm{nb}1} & bc_{\mathrm{nb}2} & \cdots & bc_{\mathrm{nbnc}} \end{array} \quad \textbf{(53)}$$

where every element bc_{ij} is the number of common faces between data set B object instance $boid_i$, and data set C object instance $coid_j$. Matrix **BC** tells us, what data set B object instances *overlap* data set C instances, or vice versa.

3.6.2 Finding Candidates for Corresponding Object Instances

Corresponding object instances share same location, which means that they overlap each other. But they also belong to semantically similar classes. Therefore, if we extract from matrix **BC** in **(53)** all overlapping ordered pairs of object instances with semantically similar labels — as in **(22)**, and refined in **(41)** up to **(49)** — we get a list of *ordered pairs of semantically similar overlapping object instances*.
For example, a list of ordered pairs of object instances like:

$$\{(boid_1, coid_5), (boid_2, coid_5), (boid_3, coid_7), (boid_3, coid_{12}), (boid_4, coid_{14}), \text{ etc.}\} \quad \textbf{(54)}$$

Furthermore, an object instance may overlap more than one other object instance. Therefore, list **(54)** is aggregated into *simple*, or *complex* correspondences:

$$\{ \{(boid_1, coid_5), (boid_2, coid_5)\}, \{(boid_3, coid_7), (boid_3, coid_{12})\}, \{(boid_4, coid_{14})\}, \text{ etc.}\} \quad \textbf{(55)}$$

This list contains *candidates for corresponding object instances*. Note that multiplicity of candidates in **(55)** is of type *n*-to-*m*, with $n, m \geq 1$, with 1-to-1 relationships as *simple* candidates, and all other relationships as *complex* candidates. Candidates means that we possibly deal with corresponding object instances, because consistency checking has not yet been done.

3.6.3 Finding Singletons

If the set of all candidates is known, then it is possible to determine *singletons*. Singletons of data set B are instances that do not have any correspondence with instances of data set C. Where it concerns data set B, it is simply the *set difference* (\ ) between:

- the set *boids* of data set B object identifiers, and

– the set *ccboids* of data set B object identifiers that participate in candidates for correspondences:

$$boids \setminus ccboids = \{ s \quad | \quad s \in boids, \quad s \notin ccboids \} \qquad \textbf{(56)}$$

with *s* an object identifier of a singleton of data set B.

For data set C a similar argument applies.

3.7 Discussion

In Section 3.2 it is assumed, that directed graph *H* is structured in such a way that every label of set *A* belongs to exactly one level. In order to realize this, it is necessary to prevent links between taxonomy subgraph and partonomy subgraph of the reference model. Links are possible, if an application object class has semantically different *roles*. In Part 3, an example is presented where an application object class has references to both a taxonomy subclass, and a partonomy component class. In order to prevent a mix up of references, within the reference model and between application ontology classes, distinct labels are used for the subclass role in the taxonomy, and the component class role in the partonomy. [12]

[12] This seems to touch the topic of *orthogonality* of taxonomy and partonomy, as mentioned in (Artale et al 1996) and (van der Vet and Mars 1998).

Part 3: Practice of Geographic Data Set Integration

Introduction to Part 3

In Part 2 a methodology for geographic data set integration has been introduced. Central in this methodology is a conceptual framework, the elements of which are ontologies, surveying rules, and reference models:

1. Ontologies are structured collections of unambiguously defined concepts.
2. Surveying rules govern the transformation process of actually observed terrain object instances (defined as object classes in the domain ontology) into instances of geographic data set object classes, as defined in application ontologies.
3. Reference models explain differences in abstraction and contents between geo-data sets to be integrated.

The framework has been developed with the objective to explain and reconcile differences between geographic data sets. Ultimately this methodology should solve the problem of geographic data set integration, which is defined in this research as the establishment of explicit relationships between corresponding object instances.

Part 3 shows a solution for this problem. It has two objectives:

1. It is a test and evaluation of the methodology developed in Part 2.
2. It is also an illustration of data set integration between topographic data sets, its 'practice'. It is a demonstration of all concepts introduced in Part 2.

With these two objectives in mind, Part 3 is divided into two chapters:

1. Chapter 4 deals with the construction of a reference model. As we have seen in Part 2, a reference model is the corner stone within the framework for data set integration. In a reference model, concepts from a domain ontology are refined and structured in such a way, that the reference model explains semantic inter-connectedness of geo-data sets. To realize this objective, surveying rules of data sets have to be studied. With this information at the class level, all concepts necessary to define a common universe for both data sets are identified. In comparing and inspecting both data sets at the instance level — visually, by overlaying both data sets — the previous information is cross-examined, and completed.
2. Chapter 5 deals with the implementation of the reference model of Chapter 4. Here candidates for correspondences are determined with the mathematical tools of Chapter 3. These candidates are checked for consistency with surveying rules, by inspecting data sets, or by visiting the test area. Special attention is dedicated to 'singletons', because this is a particular source of information about consistency.

   First ideas in Part 3 were earlier presented in (Uitermark et al 1999a).

4 Constructing a Reference Model

In Chapter 2, the concept of a reference model was introduced. The aim of a reference model is to express, or make clear semantic interconnectedness of data set classes. Basic mechanisms for expressing this semantic interconnectedness are the generalization/specialization classification (‘is-a’), and the composite/component classification (‘part-whole’).

In Chapter 3 these mechanisms were further formalized with the help of a directed graph structure. The objective of this formal approach was to develop a set-theoretic expression to compute every semantic similarity between object classes of different data sets.

In this chapter the *construction* of a reference model for two topographic data sets is demonstrated.

Essentially, a reference model is a subset of concepts from a domain ontology with additional structure. The structure is determined by concepts of two different application ontologies.

The procedure for constructing reference models is as follows:

- starting point are data sets and their surveying rules. Two topographic data sets — GBKN and TOP10vector — were earlier introduced in Part 1 and get more attention in Section 4.1
- a common subset of domain ontology concepts is chosen. Candidates for this subset come from the GTM Standard (Section 4.2)
- surveying rules of both data sets are made explicit. Their terminology is adapted to the subset of domain ontology concepts. This subset of concepts is refined into subclasses in Section 4.3
- both data sets are compared in great detail at the instance level in Section 4.4.2 in order to find resemblances and differences, which are not entirely explained by surveying rules
- from the previous information a reference model is constructed in Section 4.5, and, finally
- there is a discussion about the reference model in Section 4.6.

4.1 Geographic Data Sets

In this research, the geographic data set integration process is investigated between two topographic data sets, GBKN and TOP10vector.

4.1.1 GBKN Data Set

GBKN data set is a Dutch large-scale topographic data set (presentation scale 1 : 1,000). It is usually produced by photogrammetric *stereo plotting* with field completion. It is a nationwide mapping of buildings, roads, railways and waterways. The precision of GBKN is stated in terms of *relative precision*: in urban area the relative precision between two well defined points must be better than $20\sqrt{2}$ centimeters, and in rural area must be better than $40\sqrt{2}$ centimeters (Salzmann 1996). GBKN is updated continuously (van Oosterom 1997). **Table 1** gives an overview and description of GBKN object classes of test area Zevenaar (Kadaster 1996).

GBKN object class *inrichtingselement* in **Table 1** is in this research refined into the following subclasses:

- *berm* (‘verge’)
- *bloemenperk* (‘flowerbed’)
- *parkeerstrook* (‘parkingstrip’), and
- *trottoir* (‘sidewalk’).

The reason for this refinement is to study the effect of adding semantics to a GBKN data set.

4.1.2 TOP10vector Data Set

TOP10vector data set is a Dutch medium-scale topographic data set (presentation scale 1 : 10,000). It is usually produced by photogrammetric *mono plotting* with field completion. It is a nationwide mapping of buildings, roads, railways, waterways and land use. The precision of TOP10vector is stated in terms of *absolute precision* in relation to the *national reference system*: the location of points must be better than two meters. TOP10vector is updated every four years (van Asperen 1996). **Table 2** gives an overview and description of TOP10vector object classes (TDN 1995).

4.1.3 Test Area Zevenaar

Data for our test in geographic data set integration comes from test area Zevenaar. Its territory is mainly urban area. Its size is 30 hectares. Test area Zevenaar was chosen owing to the availability of an *object-structured* GBKN data set (Kadaster 1996). **Fig. 22** is a TOP10vector map, and **Fig. 42** is a GBKN map of test area Zevenaar.

GBKN class label	*Description*
hoofdgebouw	mainbuilding (building with one or more postal addresses)
bijgebouw	annex (building without address)
rijbaan	road
bermsloot	ditch, less than six meters wide
spoorbaan	railway
inrichtingselement	verge, flowerbed, parkingstrip, sidewalk
terrein	anything but *hoofdgebouw*, *bijgebouw*, *rijbaan*, *bermsloot*, *spoorbaan*, or *inrichtingselement*

Table 1. GBKN class labels and their description in test area Zevenaar.

TOP10 label	*Description*	*TOP10 label*	*Description*
1000	mainbuilding or annex	*3603*	cycletrack
1050	barn	*4000*	railway
1073	greenhouse	*5023*	woodland
3103	road, track ≥ 7m wide	*5203*	arableland
3203	road, track 4-7m wide	*5213*	grassland
3303	road, track 2-4m wide	*5263*	anything but *1000*, *1050*, *1073*, *3103*, *3203*, *3303*, *3533*, *3603*, *4000*, *5023*, *5203*, or *5213*
3533	street		

Table 2. TOP10vector class labels and their descriptions in test area Zevenaar.

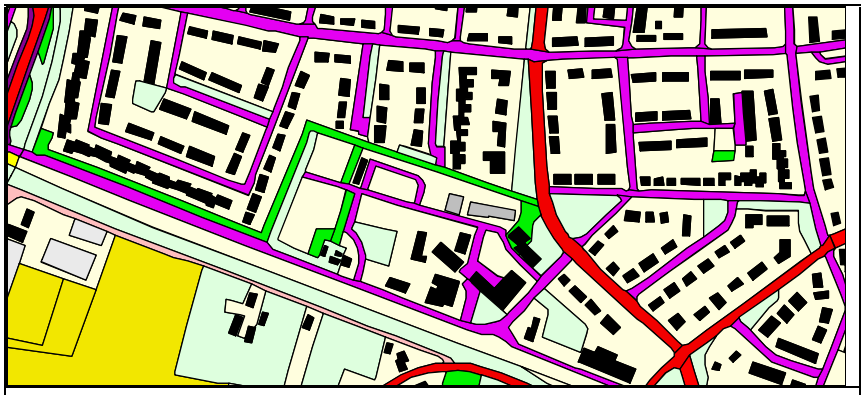


Fig. 22. TOP10vector map of test area Zevenaar.

4.2 Domain Ontology Concepts

Comparing GBKN data set in **Table 1** and TOP10vector data set in **Table 2** gives the impression that a small subset of concepts from the GTM Standard (Ravi 1995) will suffice for a domain ontology. See **Table 3** for an overview and definition.

According to **Table 3**, as far as it concerns GBKN and TOP10vector, the Real World (or terrain) is broken down into *six* object classes. *Four* of these classes (building, road, water, and land) are refined in the next section, depending on GBKN and TOP10vector data sets of test area Zevenaar.

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
road	leveled part of the terrain for traffic on land
railway	leveled part of the terrain for traffic on rails
water	part of the terrain covered by water
land	part of the terrain, having a distinct use or function, not being building, road, railway, or water
otherland	**land**, *not* having a distinct use or function

Table 3. Six domain ontology concepts and their definition.

Typographical convention

Class labels, shown in:
- **bold face**, are from *domain ontology* classes
- 'single quotes', are from *reference model* classes, and
- *italics*, are from GBKN and TOP10vector *application ontology* classes, with Dutch names for GBKN labels, and numbers for TOP10vector labels.

4.3 Refining Domain Ontology Concepts with Surveying Rules

Domain ontology concepts from **Table 3** are *refined* into subclasses for the reference model. This refinement is based on information from surveying rules. Surveying rules are necessary sources of information for geographic data set integration. Their terminology is adapted to the subset of domain ontology concepts in **Table 3**.

GBKN and TOP10vector are captured on the basis of different sets of surveying rules:

- for TOP10vector there is a set of surveying rules that is used nationwide and revised regularly (TDN 1999)

– for GBKN there are different sets of surveying rules, that vary regionally [13]. The rules for the GBKN data set in this research are based on the specification in (Kadaster 1992).

In the next three sections, domain ontology classes **building**, **road**, **water** and **land** will be refined. No refinement is needed for:

– **railway**, because for TOP10vector a class *4000* (**railway**), equivalent with GBKN class *spoorbaan* (**railway)** is created (Section 4.5.4), and

– **otherland**, because **otherland** is a left-over category that is not refined by definition. The explanation is that the real world is transformed into *partitions* in both data sets. Both GBKN and TOP10vector want to cover the terrain completely. Thus, what is not labeled explicitly becomes a left-over class. Later it will be shown that **otherland** is the *intersection* of GBKN and TOP10vector left-over classes (*terrein* and *5263*, respectively).

4.3.1 Refinements for Domain Ontology Class Building

Domain ontology class **building** is defined in **Table 3**. For GBKN, according to **Table 1**, we need refinements for:

– *hoofdgebouw*, defined as **building** with *one or more* addresses, and
– *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 refinements 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 (TDN 1999).

TOP10vector class *1000* seems a union of classes ‘mainbuilding’ and ‘annex’. However, anticipating Section 4.4.2 about comparing data sets visually, according to **Fig. 23**, ‘annex’ adjacent to ‘mainbuilding’ is not acquired as TOP10vector class *1000*.

Therefore, class ‘annex’ is divided into two reference model subclasses:

1. ‘free standing annex’, defined as **building** without address, not connected with ‘mainbuilding’, and

[13] During 1975-2000 GBKN was produced on a project-to-project base with different participants and different specifications.

2. ‘adjacent annex’, defined as **building** without address, connected with ‘mainbuilding’.

Both class *1050* (‘barn’) and 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’.

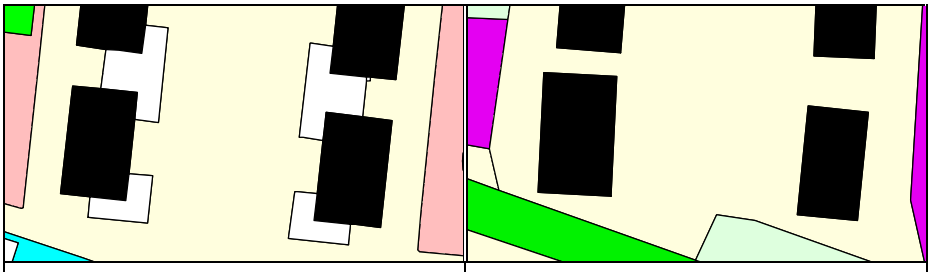


Fig. 23. ‘Annex’ (*white*), adjacent to ‘main-building’ (*black*) is not acquired for ... ... TOP10vector class *1000* (*black*), and ‘disappears’ in class *5263* (*yellow*).

‘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**.

With these **building** refinements, we can formulate surveying rules for GBKN and TOP10vector more precisely:

- GBKN surveying rules state in (Kadaster 1992) that:
  - ‘mainbuilding’ (*hoofdgebouw*) is acquired
  - ‘adjacent annex’ (*vastbijgebouw*) is acquired
  - ‘free standing annex’ (*losbijgebouw*) is acquired if situated:
    - in urban area, or
    - in rural area, with area ≥ 20m^2.
- TOP10vector surveying rules state in (TDN 1999) that:
  - ‘mainbuilding’ (*1000*), ‘remaining free standing annex’ (*1000*), ‘barn’ (*1050*), and ‘greenhouse’ (1053) are acquired if situated:
    - in urban area, accessible, with area ≥ 9m^2, or
    - in urban area, not accessible, with area ≥ 50m^2, or
    - in rural area, with area ≥ 9m^2.

Observe that these surveying rules contain *additional conditions* that constrain the acquisition of **building** (see Section 2.3). Some properties named in these conditions are *context dependent* properties: 'rural or urban area', and 'accessible'.[14]

Notice also that differences in GBKN and TOP10vector **building** surveying rules cause different *subsets* of **building** instances in both data sets.

4.3.2 Refinements for Domain Ontology Class Road

Domain ontology class **road** is defined in **Table 3**. Observe that in this definition no reference is made to 'road segments', or 'road junctions'. Therefore, **road** must be understood as an arbitrarily demarcated part of the real-world 'road network'. In fact, **road** is a homogeneous decomposable class (see Section 2.5.7). The same observation applies to GBKN road class *rijbaan* (**road**), and TOP10vector **road** classes *3103*, *3203*, *3303*, *3533*, or *3603*.

For TOP10vector, according to **Table 2**, we need refinements for:

– *3103*, described as road [15], track ≥ 7m wide, for local interconnecting traffic
– *3203*, described as road, track 4 - 7m wide, for local interconnecting traffic
– *3303*, described as road, track 2 - 4m wide, for local interconnecting traffic
– *3533*, described as road, in urban area, not for local interconnecting traffic, and
– *3603*, described as road for cyclists.

Therefore, five reference model subclasses for **road** are defined:

1. 'conngt7m', defined as road, track ≥ 7m wide, for local interconnecting traffic
2. 'conngt4m', defined as road, track 4 - 7m wide, for local interconnecting traffic
3. 'conngt2m', defined as road, track 2 - 4m wide, for local interconnecting traffic
4. 'street', defined as road, in urban area, not for local interconnecting traffic, and
5. 'cycletrack', defined as road for cyclists only.

For GBKN, according to **Table 1**, we need no refinements for GBKN class *rijbaan* because it is defined as **road**.

'Conngt7m', 'conngt4m', 'conngt2m', 'street', and 'cycletrack' become reference model classes. See **Table 4**.

Given these **road** refinements, then:

- GBKN surveying rules state that **road** (*rijbaan*) is acquired
- TOP10vector surveying rules state that **road** is acquired, if length ≥ 100 meters.

4.3.3 Refinements for Domain Ontology Class Water

Domain ontology class **water** is defined in **Table 3**.

---

[14] A **building** is 'accessible', if there is a **road** leading to **building**.

[15] In Section 4.4.2 the difference in semantics between a TOP10vector 'road' and domain ontology class **road** will be revealed.

For GBKN, according to **Table 1**, we need a refinement for:

– *bermsloot*, defined as ditch ≤ 6 meters wide.

Thus, domain ontology class **water** gets as subclass:

– 'ditch', defined as **water** ≤ 6 meters wide, interconnecting other **water**.

'Ditch' becomes a reference model class. See **Table 4**.

Given this **water** refinement, then:

- GBKN surveying rules state that 'ditch' with width ≥ 2 meters is acquired
- TOP10vector surveying rules state that 'ditch' is not acquired as area object, but as *line* object (and therefore not part of this research. See Section 1.9.1).

4.3.4 Refinements for Domain Ontology Class Land

Domain ontology class **land** is defined in **Table 3**. For GBKN, according to **Table 1**, we need refinements for *inrichtingselement*:

– *berm* ('verge')
– *parkeerstrook* ('parkingstrip')
– *trottoir* ('sidewalk'), and
– *bloemenperk* ('flowerbed').

Therefore, domain ontology class **land** gets four reference model subclasses:

1. 'verge', defined as strip of **land** ≤ 6 meters wide, one side adjacent to **road**.
2. 'parkingstrip', defined as paved strip of **land**, adjacent to **road**, as a provision for parking cars.
3. 'sidewalk', defined as paved strip of **land**, adjacent to **road**, as a provision for pedestrians.
4. 'flowerbed', defined as strip of **land**, adjacent, or inside 'sidewalk', planted with grass, flowers, or shrubs.

For TOP10vector, according to **Table 2**, we need refinements for:

– *5023* ('woodland')
– *5203* ('arableland'), and
– *5213* ('grassland').

Therefore, three more subclasses of **land** are added:

1. 'woodland', defined as **land** overgrown with such a number of leaf wood trees that their crowns form more or less a closed unity.
2. 'arableland', defined as **land** where agricultural products are cultivated.
3. 'grassland', defined as **land** mainly overgrown with a grass like vegetation.
4. 'sidewalk', 'flowerbed', 'parkingstrip', 'verge', 'arableland', 'woodland', and 'grassland' become reference model classes in **Table 4**.

Given these **land** refinements, then:

- GBKN surveying rules for **land** state that:
  - 'sidewalk', 'flowerbed', 'parkingstrip', and 'verge' are acquired as GBKN class *inrichtingselement*, and
  - 'woodland', 'arableland', 'grassland', and **otherland** are acquired as GBKN left-over class *terrein*.
- TOP10vector surveying rules for **land** state that:
  - 'woodland', 'arableland', and 'grassland' are acquired respectively as TOP10-vector classes *5023*, *5203*, and *5213* [16]
  - 'sidewalk', 'flowerbed', 'parkingstrip', 'verge' > 6 meters wide[17], and **otherland** are acquired as TOP10vector class *5263*.

Observe that GBKN class *terrein* and TOP10vector class *5263* have **otherland** in common. Therefore, the nature of **otherland** is revealed as *intersection* of two left-over classes: GBKN class *terrein* and TOP10vector class *5263*.

With this last refinement, we have created for GBKN and TOP10vector in test area Zevenaar *a common universe of discourse* (**Table 4**).

4.4 Comparing GBKN and TOP10vector Data Sets

In this section, GBKN and TOP10vector are compared visually, by overlaying maps of both data sets, in order to find resemblances and differences, which are not yet fully explained by surveying rules.

4.4.1 Comparing Buildings

Comparing GBKN and TOP10vector data sets with regard to **buildings** reveals that, if two or more **buildings** in the terrain are nearby each other, they are acquired in combination, and represented as a *single* TOP10vector object instance (**Fig. 24**). Indeed, according to TOP10vector surveying rules, **buildings** are represented in combination, if their distance is < 2 meters, except for a 'ditch' or 'footpath' between them (TDN 1999). Note that this situation causes complex n-to-1 correspondences.

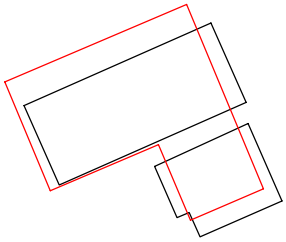


Fig. 24. Two **buildings**, < 2 meters apart ('mainbuilding', and 'free standing annex', in *black*), are combined into a single TOP10vector class *1000* instance (in *red*).

[16] 'Sidewalk', 'flowerbed', and 'parkingstrip' are sometimes added to *5213*, depending on context (see Section 4.4.4).

[17] 'verge' ≤ 6 meters wide is combined with **road** (see Section 4.4.2)

Domain ontology concept label	*Refined subclass in reference model*	*Definition 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'
road	cycletrack	**road** for cyclists	
	conngt7m	**road**, track ≥ 7 meters for local interconnecting traffic	
	conngt4m	**road**, track between 4 and 7 meters wide for local interconnecting traffic	
	conngt2m	**road**, track between 2 and 4 meters wide for local interconnecting traffic	
	street	**road** in urban area, not for local interconnecting traffic	
water	ditch	**water** ≤ 6 meters wide, and interconnecting other **water**	
railway		leveled part of the terrain for traffic on rails	
land	sidewalk	paved strip of **land** adjacent to **road** for pedestrians	
	flowerbed	strip of **land** adjacent or inside 'sidewalk', planted with grass, flowers, or shrubs	
	parkingstrip	paved strip of **land**, adjacent to **road** as a provision for parking cars	
	verge	strip of **land**, on one side adjacent to **road**	
	arableland	**land** where agricultural products are cultivated	
	grassland	**land** mainly overgrown with a grass like vegetation	
	woodland	**land** overgrown with such a number of leaf wood trees that their crowns form more or less a closed unity	
otherland		**land**, not 'sidewalk', 'flowerbed', 'parkingstrip', 'verge', 'arableland', 'grassland', or 'woodland'	

Table 4. Domain ontology concepts and their refinements into reference model subclasses for test area Zevenaar.

4.4.2 Comparing Roads

Comparing GBKN and TOP10vector data sets with regard to **roads**, reveals that 'verges' are sometimes included in TOP10vector **road** object instances (**Fig. 25**).

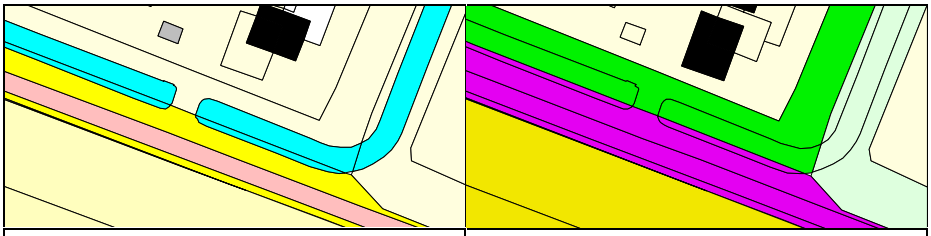


Fig. 25. GBKN *rijbaan* (**road**, in *pink*), and *berm* ('verge', in *dark yellow*) are ... ... combined into a single TOP10vector *3533* instance ('street', in *lilac*).

Indeed, there is an additional surveying rule for TOP10vector. Whenever there is a 'verge' adjacent to **road**, TOP10vector representation of **road** instances depend on the *width* of that 'verge'. If the width of 'verge' is:

- ≤ 6 meters wide: 'verge' and **road** are combined, and
- > 6 meters wide: 'verge' and **road** are represented separately.

4.4.3 Comparing Water

Comparing GBKN and TOP10vector with regard to **water**, reveals that 'ditches' in the terrain are not represented in TOP10vector. Every **water** instance in our test area is ≤ 6 meters wide, and therefore, according to TOP10vector surveying rules, represented as *line* object instance in TOP10vector (TDN 1999). Thus, area of 'ditches' is 'filled in' by surrounding terrain instances in TOP10vector. See **Fig. 26**.

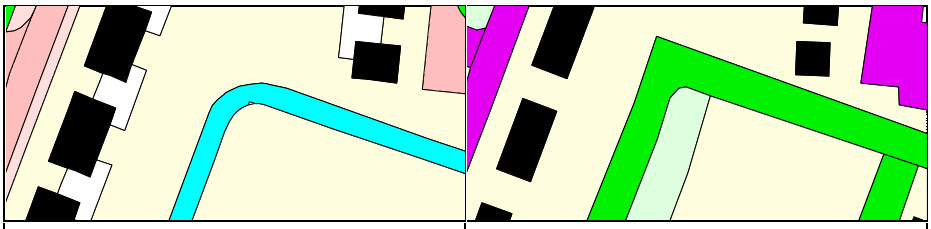


Fig. 26. 'Ditch' (in *blue*) is not represented as *area* instance in TOP10vector, and therefore 'swallowed' ... ... by surrounding terrain: 'woodland' (in *dark green*), 'grassland', (in *light green*), and **otherland** (in *beige*).

4.4.4 Comparing Land

Land classes from **Table 4** are 'sidewalk', 'flowerbed', 'parkingstrip', 'verge', 'arableland', 'grassland', and 'woodland' (from these, 'verge' has already been noticed with regard to **road** in Section 4.4.2).

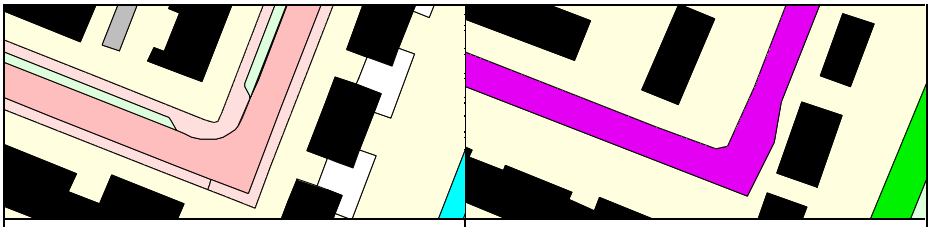


Fig. 27. 'Sidewalks' (*light pink*), 'parkingstrips' (*green*), and **otherland** (*yellow*) ...

... are combined into TOP10vector class *5263* ('TOP10vector otherland', *yellow*).

In **Fig. 27**, 'sidewalks' and 'parkingstrips' are on the one hand combined with **otherland** into TOP10vector class *5263* ('TOP10vector otherland'). On the other hand, in acquiring class *5213* ('TOP10vector grassland'), 'sidewalks' ≤ 2 meters wide, adjacent to 'flowerbeds' and 'parkingstrips' are combined with 'flowerbeds'. See **Fig. 28**.

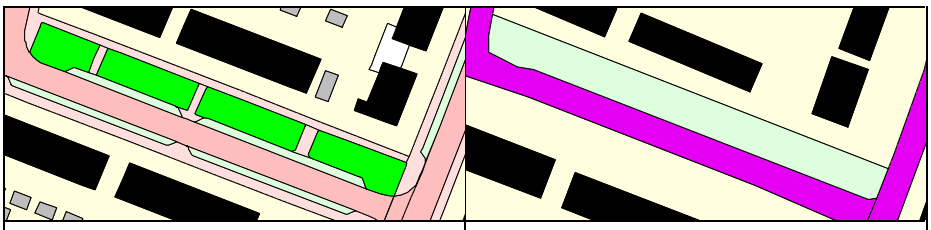


Fig. 28. 'Sidewalks' (*light pink*), 'flowerbeds' (*dark green*), and 'parkingstrips' (*light green*) ...

... are combined into a class *5213* instance ('TOP10vector grassland', *light green*).

In conclusion, TOP10vector object classes *5213* ('TOP10vector grassland'), and *5263* ('TOP10vector otherland') are composite classes. Apparently, TOP10vector class *5213* is used for two different real-world situations:

1. 'flowerbeds', ≤ 2 meters apart, including adjacent 'sidewalks' and 'parkingstrips', forming instances with area > 1000m^2, as in **Fig. 28**, or
2. 'grassland' (see **Fig. 29**).

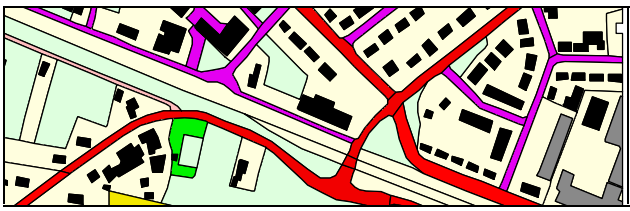


Fig. 29. The light green areas are 'grassland' instances, and represented as TOP10vector class *5213* instances ('TOP10vector grassland').

4.4.5 Comparing Left-over Classes (Otherland)

Comparing GBKN and TOP10vector data sets with regard to left-over classes *terrein* ('GBKN otherland') and *5263* ('TOP10vector otherland') reveals that:

– GBKN class *terrein* (‘GBKN otherland’) is a composition of (parts of) instances of *5023* (‘TOP10vector woodland’), *5203* (‘TOP10vector arableland’), *5213* (‘TOP10vector grassland’), and *5263* (‘TOP10vector otherland’), and

– TOP10vector class *5263* (‘TOP10vector otherland’) is a composition of (parts of) instances from GBKN classes *trottoir* (‘sidewalk’), *parkeerstrook* (‘parking-strip’), *bloemenperk* (‘flowerbed’), and *terrein* (‘GBKN otherland’).

As was mentioned before, the intersection of GBKN class *terrein* (‘GBKN otherland’) and TOP10vector class *5263* (‘TOP10vector otherland’) reveals the nature of domain ontology class **otherland**, or more specifically, the interpretation of reference model class **otherland** when integrating GBKN and TOP10vector data sets. Or, in other words, **otherland** is that part of the terrain having no distinct use or function for both GBKN *and* TOP10vector.

4.4.6 Comparing the Overlay of Both Data Sets

We complete this section with an overall comparison of GBKN and TOP10vector by a *geometric overlay* of both data sets. The sample is from test area Zevenaar introduced in Section 4.1.3.

Table 5 gives an overlay across GBKN classes and TOP10vector classes. It contains information about the overlap between object classes, and therefore if object classes *might* correspond to each other:

– a dash entry (-) means ‘no overlapping faces’, an indication for incompatible classes, and

– a non-zero entry means ‘overlapping faces’, an indication for *potential* compatible classes, because overlap is also caused by *imprecision*, and *errors* in data sets.

However, all compatible classes were determined systematically, in this section and Section 4.3. Therefore, we are able to discriminate between on the one hand compatible classes, and on the other hand between imprecision and errors. Therefore, we transform **Table 5** into **Table 6**, where compatible classes ‘share’ a common ‘underlying’ reference model class. For example, ‘mainbuilding’ (abbreviated as ‘mb’) is tentatively the underlying reference model class for GBKN class *hoofdgebouw* and TOP10vector class *1000* [18].

Overlap caused by imprecision and errors is indicated with ‘s/e’ in **Table 6**. Note the large amount of faces, classified as ‘s/e’, between GBKN *terrein* (‘GBKN otherland’) and TOP10vector *1000* (‘mainbuilding or annex’), or GBKN *hoofdgebouw* (‘mainbuilding’) and TOP10vector *5263* (‘TOP10vector otherland’). The explanation is the difference in coordinates between GBKN *hoofdgebouw* (‘mainbuilding’) and TOP10vector *1000* (‘mainbuilding or annex’). See **Fig. 24**, for example.

---

[18] Later a reference model class ‘composite building’ will be introduced (Section 4.5.2).

TOP10vector→ / *GBKN ↓*	*1000*	*1050*	*1073*	*3103*	*3203*	*3303*	*3533*	*3603*	*4000*	*5023*	*5203*	*5213*	*5263*	*Total*
hoofdgebouw	220	-	-	-	-	-	3	-	-	1	-	3	249	476
losbijgebouw	17	2	2	-	-	1	4	-	-	2	2	12	129	171
vastbijgebouw	47	-	-	-	-	-	-	-	-	-	-	-	88	135
rijbaan	-	-	-	12	1	-	51	-	1	-	-	15	54	134
fietspad	-	-	-	-	2	-	-	4	-	2	-	3	-	11
berm	-	-	-	-	4	-	7	4	-	-	-	-	2	17
bermsloot	-	-	-	-	-	1	2	-	-	3	2	4	1	13
spoorbaan	-	-	-	1	-	-	2	-	5	-	-	-	-	8
bloemenperk	-	-	-	4	-	-	22	-	-	-	-	24	67	117
parkeerstrook	-	-	-	3	-	-	16	-	-	-	-	4	19	42
trottoir	-	-	-	10	-	-	75	-	-	-	-	22	68	175
terrein	241	2	2	3	-	3	19	1	-	10	3	25	39	348
Total (col)	525	4	4	33	7	5	201	9	6	18	7	112	716	1647

Table 5. Distribution of overlapping faces between GBKN classes (*rows*) and TOP10vector classes (*columns*) in test area Zevenaar.

TOP10vector→ / *GBKN ↓*	*1000*	*1050*	*1073*	*3103*	*3203*	*3303*	*3533*	*3603*	*4000*	*5023*	*5203*	*5213*	*5263*
hoofdgebouw	mb	-	-	-	-	-	s/e	-	-	s/e	-	s/e	s/e
losbijgebouw	rfa	brn	gh	-	-	s/e	s/e	-	-	rfa	rfa	rfa	rfa
vastbijgebouw	s/e	-	-	-	-	-	-	-	-	-	-	-	adj
rijbaan	-	-	-	ct7	ct4	ct2	strt	-	s/e	-	-	s/e	s/e
fietspad	-	-	-	-	s/e	-	-	ctk	-	s/e	-	s/e	-
berm	-	-	-	vrg	vrg	vrg	vrg	vrg	-	-	-	-	s/e
bermsloot	-	-	-	-	-	s/e	s/e	-	-	dth	dth	dth	dth
spoorbaan	-	-	-	s/e	-	-	s/e	-	rlw	-	-	-	-
bloemenperk	-	-	-	s/e	-	-	s/e	-	-	-	-	fbd	fbd
parkeerstrook	-	-	-	s/e	-	-	s/e	-	-	-	-	psp	psp
trottoir	-	-	-	s/e	-	-	s/e	-	-	-	-	sdk	sdk
terrein	s/e	s/e	s/e	s/e	-	s/e	s/e	s/e	-	wd	ald	gld	old

Table 6. Compatible object classes of **Table 5** (in *gray*), and their tentatively underlying reference model classes.[19] *s/e* indicate overlap caused by imprecision, and errors (s = slivers, very small area fragments caused by imprecision; e = error).

[19] Abbreviations: mb = mainbuilding; rfa = remaining free annex; brn = barn; gh = greenhouse; adj = adjacent annex; ct2, ct4, ct7 = roadtracks; strt = street; vrg = verge; dth = ditch; rlw = railway; fbd = flowerbed; psp = parkingstrip; sdk = sidewalk; wd = woodland; ald = arable land; gld = grassland; old = otherland.

4.5 Constructing a Reference Model

For the construction of a reference model we have made two steps in previous sections:

1. We refined domain ontology classes into subclasses, depending on surveying rules of GBKN and TOP10vector data sets (Section 4.3). These subclasses become reference model object classes (**Table 4**).
2. We discovered interconnectedness between GBKN and TOP10vector data sets by visually inspecting maps of their overlapping data sets (Section 4.4). This interconnectedness suggests 'structure' for the reference model.

Steps we are now taking are:

3. Define structure between reference model object classes, and
4. Determine relationships between reference model object classes and application ontologies object classes.

These steps are treated in Section 4.5.2 up to Section 4.5.7. The result of these steps is a reference model with object classes that is *semantically rich* and *finely grained* enough to express every *semantic similarity* between object classes from different application ontologies.

First of all we give a 'heuristic' for Step 3 and Step 4 regarding reference model construction.

4.5.1 A Guiding Principle for Reference Model Construction

After Step 1 (making explicit GBKN and TOP10vector surveying rules in Section 4.3), and Step 2 (comparing GBKN and TOP10vector data sets in Section 4.4), there will be 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 another application class. With the reference model classes in **Table 4** as 'building blocks' we express these roles between application classes. To facilitate the construction of the reference model (its taxonomy subgraph and 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 semantic equivalent relationship, then identify its reference model class, and put it in the taxonomy subgraph (*e.g.* **Fig. 32** and **Fig. 40**).
   - in a semantic related relationship, then identify its reference model classes, create a new reference model superclass, and put it in the taxonomy subgraph (*e.g.* **Fig. 30**-*right* and **Fig. 40**).

– in a semantic relevant relationship, then identify its reference model classes, create a new reference model composite class, and put it in the partonomy subgraph (*e.g.* **Fig. 30**, *left* and **Fig. 40**).

2. Determine for every reference model class its *relationship* with object classes in application ontologies.

This guiding principle is now applied to GBKN and TOP10vector data sets. We will look at domain classes **building** (Section 4.5.2), **road** (Section 4.5.3), **railway** (Section 4.5.4), **land** and **otherland** (Section 4.5.5), and **water** (Section 4.5.6).

4.5.2 A Reference Model for Buildings

GBKN **building** classes are *hoofdgebouw*, *losbijgebouw*, and *vastbijgebouw* (**Table 1** and Section 4.3.1):

– *hoofdgebouw* ('mainbuilding') has according to Section 4.4.1 a role as component class with respect to TOP10vector *1000* ('mainbuilding or annex')

– *losbijgebouw* ('free standing annex') has three roles:

1. According to Section 4.4.1 it has a role as component class of TOP10vector class *1000* ('mainbuilding or annex').

2. According to Section 4.3.1 it has a role as superclass of TOP10vector classes *1050* ('barn') and *1073* ('greenhouse').

3. According to **Table 6** it has a role as component class of TOP10vector **land** classes *5023*, *5203*, *5213*, and *5263* (this role of *losbijgebouw* is modeled in Section 4.5.5).

– *vastbijgebouw* ('adjacent annex') has according to Section 4.3.1 (**Fig. 23**) and **Table 6** a role as component class of TOP10vector *5263* (this role of *vastbijgebouw* is modeled in Section 4.5.5).

TOP10vector **building** classes are *1000*, *1050*, and *1073* (**Table 2**):

– *1000* ('mainbuilding or annex') has according to Section 4.4.1 a role as composite class with respect to GBKN classes *hoofdgebouw* ('mainbuilding'), and *losbijgebouw* ('free standing annex', except 'barn' or 'greenhouse'). Therefore, we define a composite reference model class 'composite building' (abbreviated as 'compbldg'), with 'mainbuilding', and 'remaining free standing annex' (abbreviated as 'remfreeannex') as components (**Fig. 30**, *left*)

– *1050* ('barn') and *1073* ('greenhouse') have according to Section 4.3.1 and **Table 6** roles as subclasses of GBKN *losbijgebouw* ('free standing annex'). Therefore, we define a reference model superclass 'T_freeannex', with 'barn', and 'greenhouse' as components (**Fig. 30**, *right*).

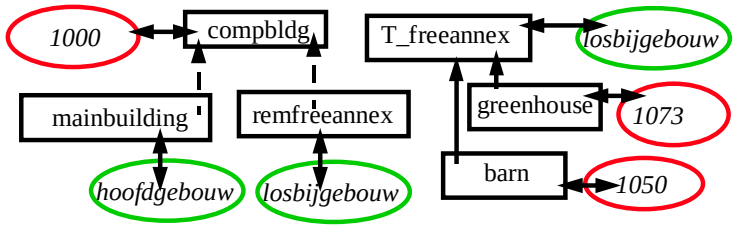


Fig. 30. Reference model construct for **buildings**.

4.5.3 A Reference Model for Roads

GBKN **road** classes are *rijbaan* (**Table 1**), and *fietspad* [20]:

– *rijbaan* (**road**) and *berm* ('verge') have according to **Fig. 25** in Section 4.4.2 roles as component classes with respect to TOP10vector classes *3103*, *3203*, *3303*, and *3533*

– *fietspad* ('cycletrack') and *berm* ('verge') have according to **Fig. 25** in Section 4.4.2 roles as component classes with respect to TOP10vector *3603*.

TOP10vector **road** classes are *3103*, *3203*, *3303*, *3533*, and *3603* (**Table 2**):

– *3103*, *3203*, *3303*, *3533*, and *3603* have according to **Fig. 25** in Section 4.4.2 roles as composite classes with respect to GBKN *rijbaan* (**road**) and GBKN *berm* ('verge'). Therefore, we define composite reference model classes 'T_3103', 'T_3203', 'T_3303', 'T_3533', and 'T_3603', respectively, with 'conngt2m', 'conngt4m', 'conngt7m', 'street', and 'cycletrack' as components, respectively. The second component class is 'verge' (**Fig. 31**).

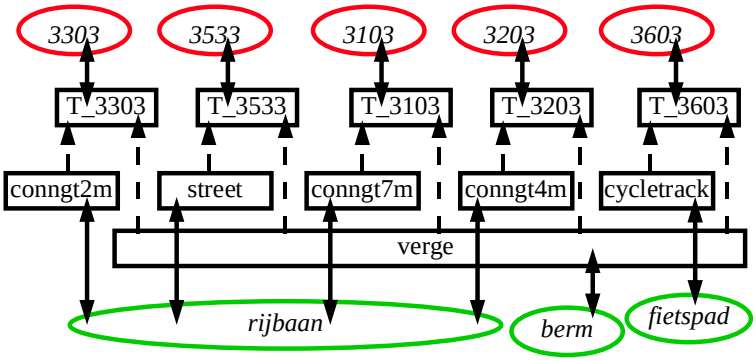


Fig. 31. Reference model construct for **roads**.

[20] Some instances of *rijbaan* (**road**) were reclassified into *fietspad* ('cycletrack'). For an explanation see Section 5.1.

4.5.4 A Reference Model for Railway

Instances of **railway** in the terrain are not acquired as area objects for TOP10vector. For test area Zevenaar, TOP10vector instances, which overlap with GBKN instances *spoorbaan* (**railway**) are recoded into a new TOP10vector object class *4000* (**railway**). Therefore, *spoorbaan* (**railway**) and *4000* (**railway**) have an equivalent relationship to each other. See the reference model construct in **Fig. 32**.

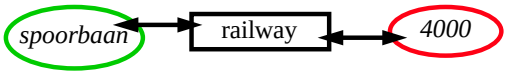


Fig. 32. Reference model construct for **railway**.

4.5.5 A Reference Model for Land

TOP10vector **land** classes are *5023*, *5203*, *5213*, and *5263* (**Table 2**):

- *5023* ('woodland') has according to **Table 6** a role as composite class with respect to GBKN *bermsloot* ('ditch'), and GBKN *losbijgebouw* ('remaining free annex). Therefore, we define a composite reference model class 'TOP10vector woodland' (abbreviated as 'T_wood'), with 'woodland', 'ditch', and 'remaining free standing annex' (abbreviated as 'remfreeannex') as components (**Fig. 33**, **Fig. 34** and **Fig. 35**)
- *5203* ('arableland') has according to **Table 6** a role as composite class with respect to GBKN *bermsloot* ('ditch'), and GBKN *losbijgebouw* ('remaining free annex). Therefore, we define a composite reference model class 'TOP10vector arableland' (abbreviated as 'T_arable'), with 'arableland', 'ditch', and 'remaining free standing annex' (abbreviated as 'remfreeannex') as components (**Fig. 33**, **Fig. 34** and **Fig. 35**)
- *5213* ('grassland') has according to Section 4.4.4 a role as composite class with respect to GBKN *bloemenperk* ('flowerbed'), GBKN *parkeerstrook* ('parkingstrip'), and GBKN *trottoir* ('sidewalk'); and according to **Table 6** with respect to GBKN *bermsloot* ('ditch'), and GBKN *losbijgebouw* ('remaining free annex). Therefore, we define a composite reference model class 'TOP10vector grassland' (abbreviated as 'T_grass') with 'grassland', 'flowerbed', 'parkingstrip', 'sidewalk', 'ditch', and 'remaining free standing annex' (abbreviated as 'remfreeannex') as components (**Fig. 33** up to **Fig. 38**)
- *5263* (TOP10vector left-over class) has according to Section 4.4.4 a role as composite class with respect to GBKN *bloemenperk* ('flowerbed'), GBKN *parkeerstrook* ('parkingstrip'), and GBKN *trottoir* ('sidewalk'); according to **Table 6** with respect to GBKN *bermsloot* ('ditch'), and GBKN *losbijgebouw* ('remaining free annex); and according to **Fig. 23** with respect to GBKN *vast-bijgebouw* ('adjacent annex'). Therefore, we define a composite reference model

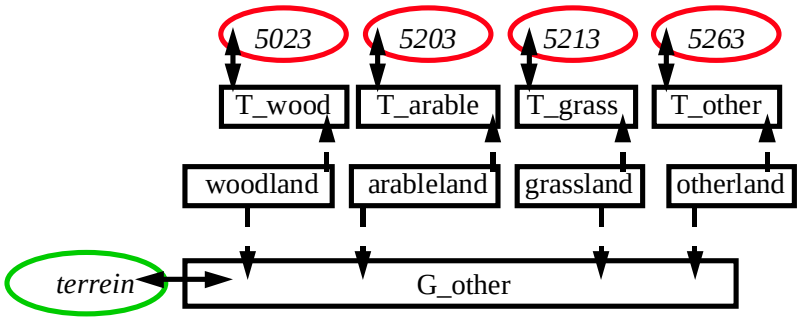


Fig. 33. Reference model construct for TOP10vector composite land classes *5023*, *5203*, *5213*, and *5263*, and composite GBKN class *terrein* ('GBKN otherland'). Red ovals = TOP10vector labels; green ovals = GBKN labels; black boxes = reference model labels; solid arrow = *taxon* relationship; dashed arrow = *parton* relationship; double arrow = *refers_to* relationship.

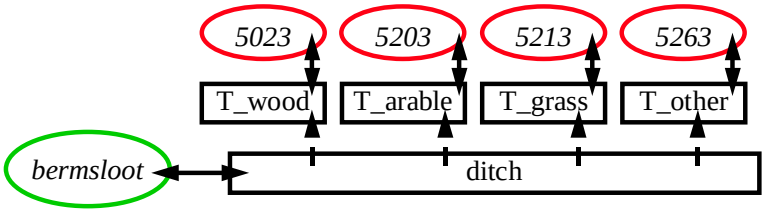


Fig. 34. Reference model construct for GBKN object class *bermsloot* ('ditch').

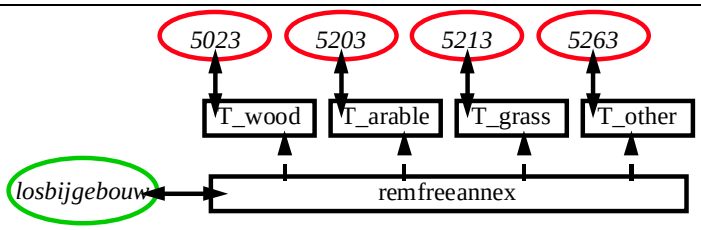


Fig. 35. Reference model construct for GBKN object class *losbijgebouw* ('remaining free standing annex') in its role as component class of TOP10vector composite **land** classes.

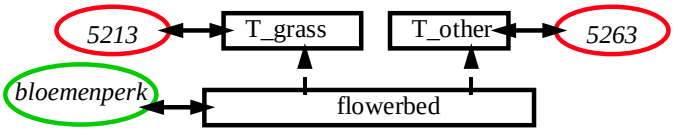


Fig. 36. Reference model construct for GBKN object class *bloemenperk* ('flowerbed').

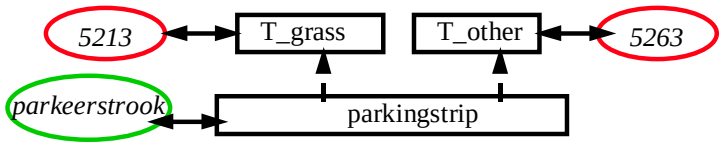


Fig. 37. Reference model construct for GBKN object class *parkeerstrook* (‘parkingstrip’).

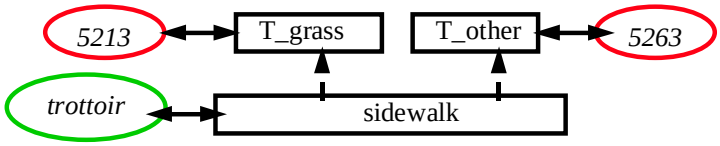


Fig. 38. Reference model construct for GBKN object class *trottoir* (‘sidewalk’).

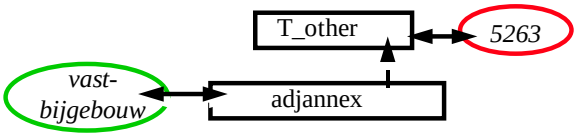


Fig. 39. Reference model construct for GBKN class *vastbijgebouw* (‘adjacent annex’).

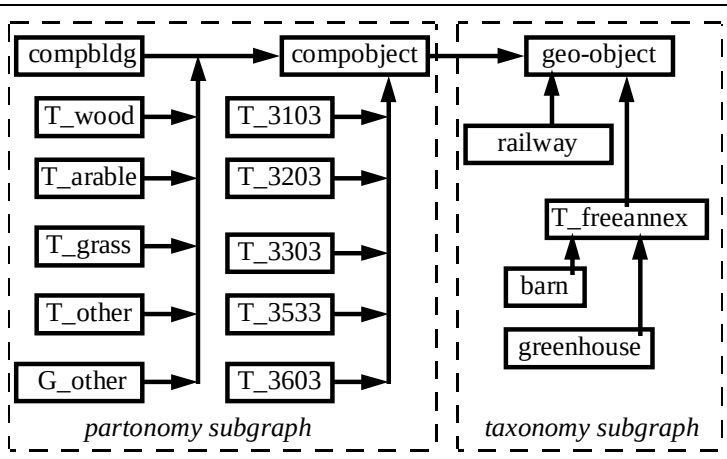


Fig. 40. The partonomy subgraph and taxonomy subgraph of the complete reference model. For component classes and semantic relationships with application classes see **Fig. 30** up to **Fig. 39**.

class ‘TOP10vector otherland’ (abbreviated as ‘T_other’) with **otherland**, ‘flowerbed’, ‘parkingstrip’, ‘sidewalk’, ‘ditch’, ‘remaining free standing annex’ (abbreviated as ‘remfreeannex’), and ‘adjacent annex’ (abbreviated as ‘adjannex’) as components (**Fig. 33** up to **Fig. 39**).

GBKN **land** classes are *berm*, *bloemenperk*, *parkeerstrook*, *trottoir*, and *terrein* (**Table 1**):

– *berm* (‘verge’) has according to Section 4.4.2 a role as component class with respect to TOP10vector *3103*, *3203*, *3303*, *3533*, and *3603* (**Fig. 31**)

– *bloemenperk* (‘flowerbed’) has according to Section 4.4.4 a role as component class with respect to TOP10vector *5213* and *5263* (**Fig. 36**)

– *parkeerstrook* (‘parkingstrip’) has according to Section 4.4.4 a role as component class with respect to TOP10vector *5213* and *5263* (**Fig. 37**)

– *trottoir* (‘sidewalk’) has according to Section 4.4.4 a role as component class with respect to TOP10vector *5213* and *5263* (**Fig. 38**)

– *terrein* (GBKN left-over class) has according to **Table 6** a role as composite class with respect to TOP10vector *5023* (‘TOP10vector woodland’), *5203* (‘TOP10-vector arableland’), *5213* (‘TOP10vector grassland’), and *5263* (‘TOP10vector otherland’). Therefore, we define a composite reference model class ‘GBKN otherland’ (abbreviated as ‘G_other’) with ‘woodland’, ‘arableland’, ‘grassland’, and **otherland** as component classes (**Fig. 33**).

4.5.6 A Reference Model for Water

GBKN **water** class is *bermsloot* (‘ditch’). According to TOP10vector surveying rules ‘ditch’ is not represented as area object class in TOP10vector. According to **Table 6**, class *bermsloot* (‘ditch’) overlaps four TOP10vector **land** classes (*5023*, *5203*, *5213*, and *5263*). Therefore, GBKN *bermsloot* (‘ditch’) has a role as component class with respect to TOP10vector classes *5023*, *5203*, *5213*, and *5263*. See **Fig. 34**.

4.5.7 Completing the Reference Model

Finally, the reference model is completed by ‘adding’ all parts in **Fig. 30** up to **Fig. 39** into one schema, where all composite classes are grouped in a partonomy subgraph, under abstract class *composite object* (abbreviated as ‘compobject’). See **Fig. 40**.

4.6 Summary and Discussion

This chapter presented the construction of a reference model. It started with data sets (**Table 1** and **Table 2**), and their surveying rules. Then, a domain ontology with a basic set of six ‘top-level’ concepts was introduced (**Table 3**). Candidates for this set of concepts were based on the GTM Standard. Concepts are refined into

subclasses, depending on data sets involved. This refinement into subclasses was based on surveying rules. Consequently, a number of subclasses for **buildings**, **roads**, **water**, and **land**, were added to the reference model, culminating into a common universe of discourse for GBKN and TOP10vector (**Table 4**). The refinement into subclasses allowed us to formalize surveying rules in terms of this common universe. After that, data sets were compared visually. Most of what was discovered visually was also confirmed by surveying rules. This resulted in a better understanding of the semantic interconnectedness of data sets, especially the interpretation of left-over classes *terrein* ('GBKN otherland') and *5263* ('TOP10vector otherland'), and their relationship with domain class **otherland** (Section 4.4.5). To make the story complete, tables were given of the geometric overlay of data sets (**Table 5** and **Table 6**). The information of these tables reveal interconnecting reference model classes. However, care must be taken because of errors and imprecision of data sets.

Now this approach in Section 4.3 and Section 4.4 seems linear, but it is not. It is cyclic, and iterative. Even more cyclic, and iterative is the construction of the reference model in Section 4.5. The idea is to design a structure that is semantically rich and finely grained enough, to express every semantic similarity between data sets. To facilitate the design a 'guiding principle', a heuristic was presented. (Section 4.5.1). Central in this 'guiding principle' is the concept of *role*. A role is what a data set class is in confrontation with another data set class: this can be equivalent class, subclass, superclass, component class, or composite class.

A data set class may have different roles. Let us explain this with respect to GBKN classes *hoofdgebouw* ('mainbuilding') and *losbijgebouw* ('free standing annex'), and TOP10vector class *1000* ('mainbuilding or annex'). Confronting these classes, with each other and with other classes, reveal their different roles:

- *hoofdgebouw* has a role as equivalent class with respect to *1000* (as in **Fig. 19**), or
- *hoofdgebouw* has a role as component class with respect to *1000* (as in **Fig. 24**).

It is this last role that is modeled in **Fig. 30**. The restrictions we imposed on the component/composite structure (in Section 2.5.2) allowed us to make this choice. However, both roles could have been modeled simultaneously and independently, creating a pair of object classes (*hoofdgebouw*, *1000*) that is both equivalent *and* relevant. The same situation applies to GBKN *losbijgebouw* ('free standing annex') and TOP10vector *1000* ('mainbuilding or annex').

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 5.6.1), because if modeled it would obscure the overview of the reference model (its surveyability).

Fig. 41 is an illustration of the previous discussion. Here we see the procedure for reference model construction in this chapter (the upper part of **Fig. 41**). In Chapter 5 the reference model is applied (the lower part of **Fig. 41**). This leads to a better

understanding of data sets involved (*e.g.* how often a role occurs). This *feedback* possibly adapts the reference model, the limit of which is a *learning* system.

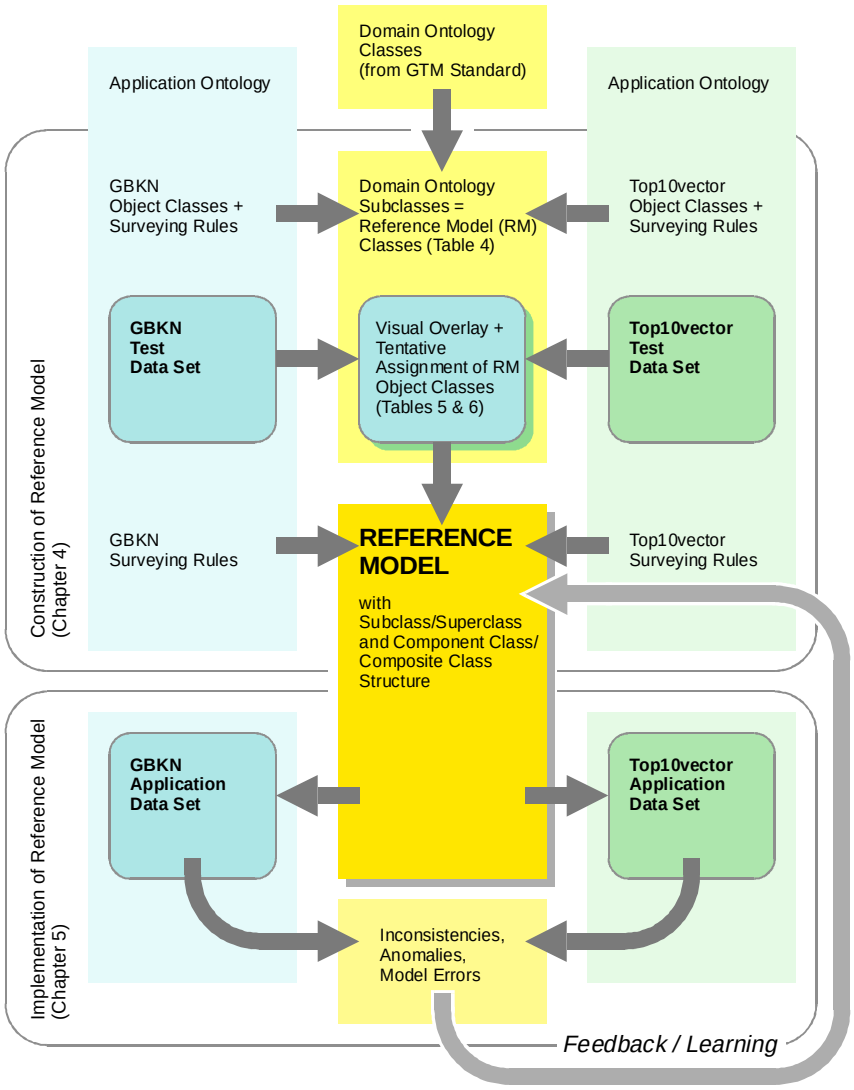


Fig. 41. A procedure for reference model construction and application.

Note that in this research, test data sets (**Fig. 41**, *upper part*) are not different from application data sets (**Fig. 41**, *lower part*). In this chapter insight was gained with some examples from test area Zevenaar. In Chapter 5 the whole test area population is investigated.

5 Implementing a Reference Model

A reference model is a tool for finding corresponding object *classes* in different geo-data sets. Geographic data set integration ultimately deals with *instances* of object classes. In this chapter it is shown how we get from classes to instances.

First of all, the reference model of Chapter 4 is implemented, and applied to GBKN and TOP10vector class label sets from test area Zevenaar. This results in ordered pairs of compatible classes (Section 5.1). Then, this result is applied to instances of GBKN and TOP10vector data sets, resulting in candidates for correspondences, presented in Section 5.2. Candidates are checked for consistency in Section 5.3 up to Section 5.7. This chapter is closed with a discussion in Section 5.8.

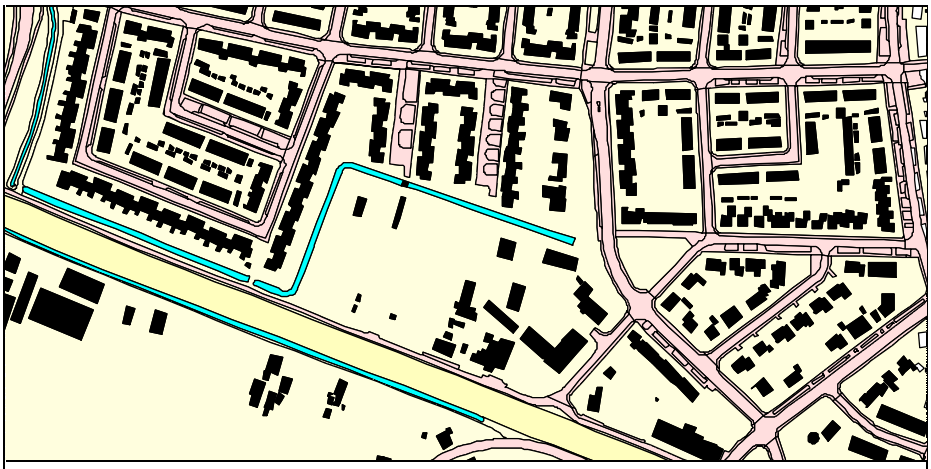


Fig. 42. GBKN map of test area Zevenaar.

5.1 Applying the Reference Model

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

- GBKN data set in **Fig. 42** has 694 object instances, from twelve classes, with a distribution according to **Table 7**, and
- TOP10vector data set in **Fig. 22** has 295 object instances, from thirteen classes, with a distribution according to **Table 8**.

GBKN data set in this research has been adapted by creating new object classes:

- object class *bijgebouw* (‘annex’) has been expanded with object classes for *vastbijgebouw* (‘adjacent annex’) and *losbijgebouw* (‘free standing annex’). Note that here a context property is used to create data classes for ‘annex’

Label	*Referring to RM class* [21]	*#*	*With RM components*
berm	'verge'	8	-
bermsloot	'ditch'	6	-
bloemenperk	'flowerbed'	71	-
fietspad	'cycletrack'	2	-
hoofdgebouw	'mainbuilding'	221	-
losbijgebouw	'T_freeannex'('barn'+'green-house') , 'remaining free standing annex'	140	-
parkeerstrook	'parkingstrip'	27	-
rijbaan	'conngt2m', 'conngt4m', 'conngt7m', 'street'	38	-
spoorbaan	**railway**	1	-
terrein	'GBKN otherland'	25	'woodland', 'arableland', 'grassland', **otherland**
trottoir	'sidewalk'	67	-
vastbijgebouw	'adjacent annex'	88	-

Table 7. Distribution of 694 GBKN object instances in test area Zevenaar.

Label	*Referring to RM class*	*#*	*With RM components*
1000	'composite building'	167	'mainbuilding', 'remfreeannex'
1050	'barn'	2	-
1073	'greenhouse'	2	-
3103	'T_3103'	6	'conngt7m', 'verge'
3203	'T_3203'	1	'conngt4m', 'verge'
3303	'T_3303'	1	'conngt2m', 'verge'
3533	'T_3533'	38	'street', 'verge'
3603	'T_3603'	2	'cycletrack', 'verge'
4000	**railway**	6	-
5023	'T_woodland'	8	'woodland', 'ditch', 'remfreeannex'
5203	'T_arableland'	3	'arableland', 'ditch', 'remfreeannex'
5213	'T_grassland'	22	'grassland', 'ditch', 'flowerbed', 'parking-strip', 'sidewalk', 'remfreeannex'
5263	'TOP10vector other-land'	37	**otherland**, 'ditch', 'adjannex', 'flowerbed', 'parkingstrip', 'sidewalk', 'remfreeannex'

Table 8. Distribution of 295 TOP10vector object instances in test area Zevenaar.

[21] RM = Reference model.

- instances of *terrein* ('GBKN otherland') ≤ 6 meters wide, adjacent to *rijbaan* (**road**), are identified and labeled as object class *berm* ('verge')
- some instances of *rijbaan* (**road**) are identified and labeled as object class *fietspad* ('cycletrack').

TOP10vector data set is also adapted for this research:

- in anticipation of a future redefinition of TOP10vector object class **railway**, instances of object classes *5213* ('TOP10vector grassland') and *5263* ('TOP10vector otherland'), coinciding with GBKN class *spoorbaan* (**railway**) are reclassified into class *4000* (**railway**).

The motivation for the previous adaptations is to 'enrich' data sets with more semantics in order to be more specific in correspondences.

To compute semantically similar classes by matrix multiplication, all relationships in the constructs of the reference model in Chapter 4 are translated into matrix **T**, mentioned in Chapter 3. Then, relationships between reference model classes, and application ontologies classes, as depicted in **Fig. 30** up to **Fig. 40**, are translated into matrices **R** and **S**, also mentioned in Chapter 3 (In Appendix A, complete documentation is given of the construction of **R**, **T**, and **S**).

All ordered pairs of semantically similar class labels, between GBKN and TOP10vector, are expressed by:

$\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S} \cong$ [22] { (*berm*, *3103*), (*berm*, *3203*), (*berm*, *3303*), (*berm*, *3533*), (*berm*, *3603*), (*bermsloot*, *5023*), (*bermsloot*, *5203*), (*bermsloot*, *5213*), (*bermsloot*, *5263*), (*bloemenperk*, *5213*), (*bloemenperk*, *5263*), (*fietspad*, *3603*), (*hoofdgebouw*, *1000*), (*losbijgebouw*, *1000*), (*losbijgebouw*, *1050*), (*losbijgebouw*, *1073*), (*losbijgebouw*, *5023*), (*losbijgebouw*, *5203*), (*losbijgebouw*, *5213*), (*losbijgebouw*, *5263*), (*parkeerstrook*, *5213*), (*parkeerstrook*, *5263*), (*rijbaan*, *3103*), (*rijbaan*, *3203*), (*rijbaan*, *3303*), (*rijbaan*, *3533*), (*spoorbaan*, *4000*), (*terrein*, *5023*), (*terrein*, *5203*), (*terrein*, *5213*), (*terrein*, *5263*), (*trottoir*, *5213*), (*trottoir*, *5263*), (*vastbijgebouw*, *5263*) }	**(57)**

The 34 ordered pairs of labels in **(57)**, out of a potential of 12 × 13 = 156 pairs of labels, are semantically similar.

Next, we compute the *type* of semantic similarity of ordered pairs of labels in **(57)**, which is to say if corresponding classes are equivalent, related, or relevant:

- Semantic *equivalent* ordered pairs of labels between GBKN and TOP10vector are expressed by:

[22] ≅ denotes 'is similar to'.

$\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec \quad \cong \quad \{ (spoorbaan, 4000) \}$	**(58)**

as can be verified in **Fig. 32**.

– Semantic *related* ordered pairs of labels between GBKN and TOP10vector are expressed by:

$\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec \quad \cong$ { (*losbijgebouw, 1050*), (*losbijgebouw, 1073*) }	**(59)**

as can be verified in **Fig. 30** (*right*), where *losbijgebouw* is a superclass of *1050* and *1073*:

$\mathbf{R}rec^T \cdot \mathbf{T}propsper \cdot \mathbf{S}rec \quad \cong$ { (*losbijgebouw, 1050*), (*losbijgebouw, 1073*) }	**(60)**

– Semantic *relevant* ordered pairs of labels from GBKN and TOP10vector are presented in **(61),** as can be verified in **Fig. 30** (*left*), **Fig. 31**, **Fig. 34** up to **Fig. 39**.

$\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc \quad \cong$ { (*berm, 3103*), (*berm, 3203*), (*berm, 3303*), (*berm, 3533*), (*berm, 3603*), (*bermsloot, 5023*), (*bermsloot, 5203*), (*bermsloot, 5213*), (*bermsloot, 5263*), (*bloemenperk, 5213*), (*bloemenperk, 5263*), (*fietspad, 3603*), (*hoofdgebouw, 1000*), (*losbijgebouw, 1000*), (*losbijgebouw, 5023*), (*losbijgebouw, 5203*), (*losbijgebouw, 5213*), (*losbijgebouw, 5263*), (*parkeerstrook, 5213*), (*parkeerstrook, 5263*), (*rijbaan, 3103*), (*rijbaan, 3203*), (*rijbaan, 3303*), (*rijbaan, 3533*), (*terrein, 5023*), (*terrein, 5203*), (*terrein, 5213*), (*terrein, 5263*), (*trottoir, 5213*), (*trottoir, 5263*), (*vastbijgebouw, 5263*) }	**(61)**

In **(61)**, most GBKN classes are constituents of TOP10vector classes ($\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$). Since $\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec = \mathbf{0}$ there are no TOP10vector classes that are constituents of GBKN classes. The only GBKN composite class *terrein* ('GBKN otherland') has four TOP10vector composite classes as its constituents (or, vice versa):

$\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc \quad \cong$ { (*terrein, 5023*), (*terrein, 5203*), (*terrein, 5213*), (*terrein, 5263*) }	**(62)**

as can be verified in **Fig. 33**.

5.2 Results of the Reference Model

Overlaying GBKN and TOP10vector data sets generated a (new) partition of 1647 faces (see **Table 5** and **Table 6**). From this partition, a 694 × 295 **GT** matrix of overlapping GBKN and TOP10vector object instances is set up. Matrix **GT** tells us, which GBKN object instances overlap TOP10vector object instances, or vice versa. Matrix **GT** has 1475 non-zero entries, which means 1475 overlapping pairs of object instances.

Corresponding object instances belong to (a) semantically similar classes, and (b) share same location, which means overlap each other. If we extract from 1475 overlapping pairs of object instances of matrix **GT**, all overlapping pairs of object instances with semantically similar labels as in **(57),** we get a list of 824 pairs of semantically similar overlapping object instances.

Since for example a GBKN instance may be a component instance of a TOP10vector composite instance, the list of 824 ordered pairs of semantically similar overlapping object instances is aggregated into 205 (simple and complex) correspondences, involving 681 GBKN instances, and 280 TOP10vector instances (see for an explanation Section 3.6.2). These 205 correspondences are in fact *candidates*: consistency checking has yet to been done.

If the set of all candidates is known, then it is possible to determine *singletons*, which mean object instances not having any correspondence with other object instances:

- where it concerns GBKN, it is the difference between a total of 694 GBKN object instances, and the set of 681 GBKN object instances participating in candidates, being 694 – 681 = 13 GBKN singletons, and
- where it concerns TOP10vector, it is the difference between a total of 295 TOP10vector object instances, and the set of 280 TOP10vector object instances participating in candidates, being 295 – 280 = 15 TOP10vector singletons.

In the subsequent sections we will look at candidates for three classes: **buildings** (Section 5.3), **roads** (Section 5.4), and **land** (Section 5.5). 'Singletons' will be discussed in Section 5.6. Class **water** will be considered with class **land**, because subclass 'ditch' is not acquired for TOP10vector. Class **railway** is not considered because for this class consistency is guaranteed by the recoding operation, mentioned previously.

5.3 Consistency of Building Candidates

With reference model construct for **buildings** (**Fig. 30**), translated into equation $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ **(57)**, we get:

{ (*hoofdgebouw*, *1000*), (*losbijgebouw*, *1000*), (*losbijgebouw*, *1050*), (*losbijgebouw*, *1073*) }

as ordered pairs of compatible GBKN and TOP10vector **building** labels. With GBKN and TOP10vector **building** instances as input (summarized in **Table 7** and **Table 8**) and applied to matrix **GT** in Section 5.2, we get 163 candidates (**Table 9**).

To check consistency, we use surveying rules for **buildings**, formulated in Section 4.3.1:

1. Simple correspondences of type (*hoofdgebouw*, *1000*) imply possible real-world situations, where 'mainbuilding' is situated in:

   – urban area, accessible, with area ≥ 9m^2, or
   – urban area, not accessible, with area ≥ 50m^2, or
   – rural area, with area ≥ 9m^2

   which can be determined by applying, for example, propositional calculus to **building** surveying rules (see Appendix C). In order to decide if candidates of type (*hoofdgebouw*, *1000*) are consistent, we have to test whether *extensions* of candidates satisfy foregoing *intensions*, *i.e.* the combination of GBKN and TOP10vector surveying rules (see Section 2.9.4). Indeed, instances of simple candidates (*hoofdgebouw*, *1000*) in **Table 9** are inspected, and satisfy their intensions. Therefore, we conclude that they are consistent with surveying rules.

Candidates			*# GBKN*		# TOP10vector		
Type	*#*	*Class labels*	*hoofd*	*losbij*	*1000*	*1050*	*1073*
simple	137	(*hoofdgebouw*, *1000*)	137	-	137	-	-
	6	(*losbijgebouw*, *1000*)	-	6	6	-	-
	2	(*losbijgebouw*, *1050*)	-	2	-	2	-
	2	(*losbijgebouw*, *1073*)	-	2	-	-	2
complex	14	(*hoofdgebouw* + *losbijgebouw*, *1000*)	78	11	14	-	-
	1	(1-*hoofdgebouw*, 2-*1000*)[23]	1	-	2	-	-
	1	(2-*hoofdgebouw*, 2-*1000*)[23]	2	-	2	-	-
Total	163		218	21	161	2	2

Table 9. Distribution of simple and complex **building** candidates.

2. Simple correspondences of types { (*losbijgebouw*, *1000*), (*losbijgebouw*, *1053*), (*losbijgebouw*, *1073*) } imply possible real-world situations, where 'free standing annex', is situated in:

   – urban area, accessible, with area ≥ 9m^2, or
   – urban area, not accessible, with area ≥ 50m^2, or
   – rural area, with area ≥ 20m^2

[23] **Fig. 44** and **Fig. 45**.

which can be determined by applying, for example, propositional calculus to **building** surveying rules (see Appendix C). In order to decide if candidates of types { (*losbijgebouw*, *1000*), (*losbijgebouw*, *1053*), (*losbijgebouw*, *1073*) } are consistent, again we have to test whether extensions of candidates satisfy foregoing intensions. All ten candidates in **Table 9** have instances situated in rural area, with area ≥ 20m^2, and are therefore consistent with surveying rules.

3. Complex correspondences of type (*hoofdgebouw* + *losbijgebouw*, *1000*) in **Table 9** imply possible real-world situations mentioned previously in item 1 and item 2 above. However, according to Section 4.4.1, we need an additional check for the distance between two, or more GBKN instances *hoofdgebouw* ('mainbuilding'). In **Fig. 43** there is an example where this distance is < 2 meters, therefore the check succeeds.

4. According to surveying rules **building** correspondences should be simple, or, if complex, of (*n*-GBKN-to-*1*-TOP10vector) multiplicity. However, two candidates in **Table 9** have a different multiplicity, explained as follows:

   – in **Fig. 44** there is one instance of GBKN *hoofdgebouw* ('mainbuilding') corresponding with two instances of TOP10vector *1000* ('mainbuilding or annex'). It is caused by the absence of a proper demarcation between GBKN *vastbijgebouw* ('adjacent annex') and GBKN *hoofdgebouw* ('mainbuilding'), as was detected after field inspection. Therefore, it is classified as surveying rule error.

   – in **Fig. 45** there are two instances of GBKN *hoofdgebouw* ('mainbuilding') corresponding with two instances of TOP10vector *1000* ('mainbuilding or annex'). It is caused by an unexplained geometrical difference — a translation — of six meters between data sets, therefore one *1000* instance overlaps two *hoofdgebouw* instances, resulting in this complex candidate. Large errors in location — blunders, not imprecision — are classified as surveying rule errors, hence this situation is also a surveying rule error.

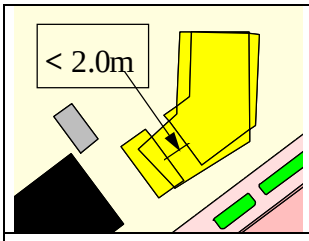


Fig. 43. An additional check for nearby buildings.[24]

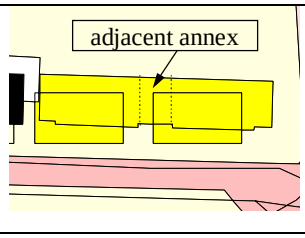


Fig. 44. An 'adjacent annex' is not represented.[24]

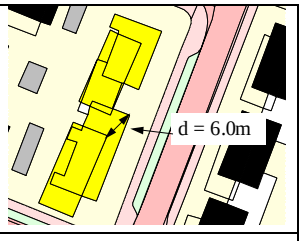


Fig. 45. An unexplained difference of six meters.[24]

[24] Relevant objects are high lighted in *yellow* in an overlay of GBKN and TOP10vector.

5.4 Consistency of Road Candidates

With reference model construct for **roads** (**Fig. 31**), translated into equation $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ **(57)**, we get:

{ (*berm*, *3103*), (*berm*, *3203*), (*berm*, *3303*), (*berm*, *3533*), (*berm*, *3603*), (*fietspad*, *3603*), (*rijbaan*, *3103*), (*rijbaan*, *3203*), (*rijbaan*, *3303*), (*rijbaan*, *3533*) }

as ordered pairs of compatible GBKN and TOP10vector **road** labels. With GBKN and TOP10vector **road** instances as input (summarized in **Table 7** and **Table 8**) and applied to matrix **GT** in Section 5.2, we get 20 candidates (**Table 10**).

There is a single additional condition for TOP10vector, namely that **roads** should be more than 100 meters long. All 20 candidates in **Table 10** agree on this point. Therefore, they are consistent with surveying rules.

Candidates		*# GBKN instances*			*# TOP10vector instances*			
Type	*#*	*rijbaan*	*berm*	*fietspad*	*3103*	*3203*	*3533*	*3603*
simple	11	11	-	-	-	-	11	-
complex	9	27	8	2	6	1	21	2
Total	20	38	8	2	6	1	32	2

Table 10. Distribution of simple and complex **road** candidates.

Almost half of candidates in **Table 10** are of a complex nature, for example see **Fig. 46** and **Fig. 47**.

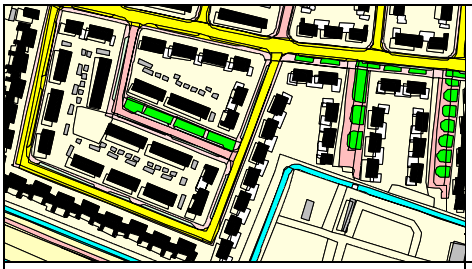


Fig. 46. A 5-to-6 **road** candidate.

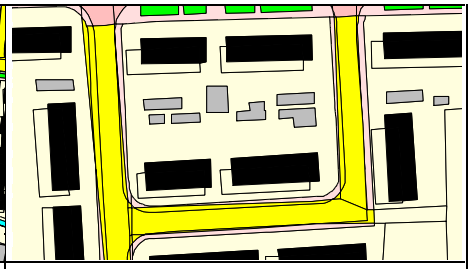


Fig. 47. A 2-to-3 **road** candidate.

To check these candidates in a automatic fashion requires a mechanism to break complex candidates down into simple candidates, as was mentioned in Section 2.9.3.

5.5 Consistency of Land Candidates

With reference model constructs for **land** (**Fig. 33** up to **Fig. 39**), translated into equation $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ **(57)**, we get:

{ (*bermsloot*, *5023*), (*bermsloot*, *5203*), (*bermsloot*, *5213*), (*bermsloot*, *5263*), (*bloemenperk*, *5213*), (*bloemenperk*, *5263*), (*losbijgebouw*, *5023*), (*losbijgebouw*, *5203*), (*losbijgebouw*, *5213*), (*losbijgebouw*, *5263*), (*parkeerstrook*, *5213*), (*parkeerstrook*, *5263*), (*terrein*, *5023*), (*terrein*, *5203*), (*terrein*, *5213*), (*terrein*, *5263*), (*trottoir*, *5213*), (*trottoir*, *5263*), (*vastbijgebouw*, *5263*) }

as ordered pairs of compatible GBKN and TOP10vector **land** labels. With GBKN and TOP10vector **land** instances as input (summarized in **Table 7** and **Table 8**) applied to matrix **GT** in Section 5.2, we get 21 candidates (**Table 11**).

In **Table 11**, all candidates except one are of a complex nature. This is partly caused by GBKN left-over class *terrein* ('GBKN otherland'), where instances can be as big as 50,000 m^2 (**Fig. 48**). To check these candidates in a automatic fashion requires also a mechanism to break them down into simple candidates, as mentioned in Section 2.9.3.

Candidates		*# GBKN instances*							*#TOP10 instances*			
Type	*#*	*berm sloot*	*ter- rein*	*bloe mpk*	*los- bijg.*	*par- kstk*	*trot- toir*	*vast bijg.*	*50 23*	*52 03*	*52 13*	*52 63*
simple	1	-	1	-	-	-	-	-	-	-	-	1
complex	20	6	24	70	118	22	65	87	8	3	22	35
Total	21	6	25	70	118	22	65	87	8	3	22	36

Table 11. Distribution of simple and complex **land** candidates.

There is a single additional condition for GBKN class *losbijgebouw* ('free standing annex') in **land** candidates. A GBKN *losbijgebouw* ('free standing annex') in a **land** candidate implies possible real-world situations, where a 'free standing annex' is situated in:

- urban area, not accessible, with area < 50m^2, or
- urban area, with area < 9m^2.

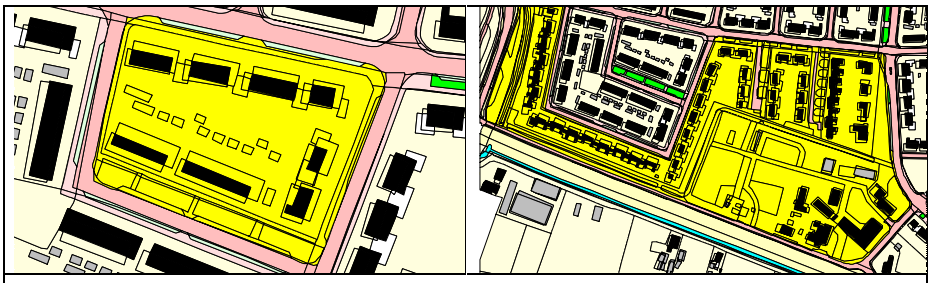


Fig. 48. Complex **land** candidates (in *yellow*). Right, a very large *terrein* instance.

It happens that most ‘free standing annexes’ satisfy this rule (**Fig. 49**). However, some ‘free standing annexes’, situated in rural area, should have TOP10vector counterparts, and are therefore classified as surveying rule errors (**Fig. 50**).

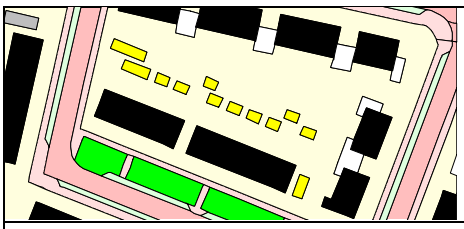


Fig. 49. GBKN *losbijgebouw* (‘free standing annex’, *yellow*) situated in inaccessible urban area, correctly without TOP10vector counterpart.

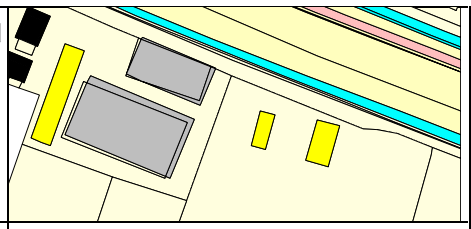


Fig. 50. GBKN *losbijgebouw* (‘free standing annex’, in *yellow*) situated in rural area, incorrectly without TOP10vector counterpart.

5.6 Singletons

A reference model is designed in such a way that specific information is established about semantic interconnectedness of data sets, even for classes that are not acquired for a one of the data sets. Therefore, instances of data sets, not present in the set of correspondences — singletons — indicate *anomalies*. If all roles of object classes of both data sets are modeled in the reference model, then singletons are caused by surveying rule errors, including data set errors (like coding errors), and differences in actuality (synchronization errors). However, singletons, as we will see in this section, reveal also violations of underlying assumptions of the methodology developed in this research — *model errors*.

5.6.1 Singletons of GBKN Buildings

A GBKN **building**, having no TOP10vector **building** counterpart, is implied by *five* possible real-world situations:

1. ‘adjacent annex’, or
2. ‘mainbuilding’, with area < 9m^2, or
3. ‘mainbuilding’, in urban area, not accessible, with area < 50m^2, or
4. ‘free standing annex’, in urban area, not accessible, with area < 50m^2, or
5. ‘free standing annex’, in urban area, with area < 9m^2.

This can be determined by applying for example propositional calculus to **building** surveying rules in Section 4.3.1 (See Appendix C).

Now, from these five situations, ‘adjacent annex’ (item 1) is modeled as component class of TOP10vector *5263* (‘TOP10vector otherland’, Section 4.5.5), and ‘free standing annex’ (item 4 and item 5) is modeled in a role of component class of TOP10vector **land** classes (Section 4.5.5).

'Mainbuildings' (item 2 and item 3) represent rare situations (*e.g.* a 'mainbuilding' with area < $9m^2$ might be a transformer station). This role of 'mainbuilding' is not modeled in Chapter 4 (but it could be done by making class 'mainbuilding' also a **land** component class, like 'remfreeannex' in **Fig. 35**). Hence, in order to discriminate between these exceptions and GBKN singletons caused by surveying rule errors, we have to check if GBKN singletons possibly satisfy additional conditions in item 2 and item 3 above.

In the test set there are four GBKN **building** singletons. These singletons are inspected, manually, and visually by field inspection:

– three singletons are 'mainbuildings', situated in urban area, accessible with area ≥ $9m^2$, therefore indicating TOP10vector surveying rule errors (**Fig. 51**)

– one singleton is 'free standing annex', situated in urban area, accessible, but with area ≥ $9m^2$. Therefore, it should have a TOP10vector counterpart.

The last situation of 'free standing annex' could have been part of a **land** correspondence, as a component class of *5263* ('TOP10vector otherland', Section 4.5.5). However, it is not 'detected' by *5263*, because in **Fig. 52** 'free standing annex' is completely overlapped and inside an (incompatible) TOP10vector *3533* ('street'). This is caused by imprecision. A surveying rule error is detected by an unexpected and undesirable situation. Therefore, this singleton is classified as *model error*, in the sense of a *violation* of the underlying assumption that precision of topographic data sets is sufficient enough to use overlap for 'same location' for candidate correspondences (or *no* overlap for 'different location'). This situation touches the issue of *resolution* of data sets, *i.e.* the representation of small objects.

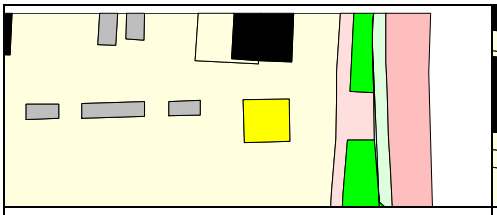


Fig. 51. A GBKN singleton *hoofdgebouw* ('mainbuilding', in *yellow*), incorrectly without TOP10vector counterpart.

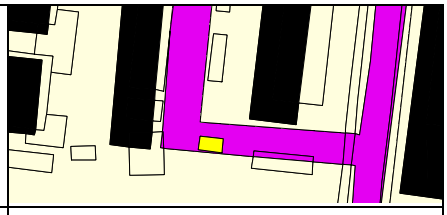


Fig. 52. A GBKN *losbijgebouw* ('free standing annex', in *yellow*), completely inside a TOP10vector *3533* ('street').

5.6.2 Singletons of TOP10vector Buildings

A TOP10vector **building**, having no GBKN **building** counterpart, is implied by *one* possible real-world situation:

– 'free standing annex', in rural area, with area between $9m^2$ and $20m^2$.

This can be determined by applying for example propositional calculus to **building** surveying rules in Section 4.3.1 (See Appendix C).

Note that this role is not modeled in Chapter 4. Modeling this role requires a component class for ‘GBKN otherland’, where this component class refers to *1000* (‘mainbuilding or annex’). However, it happens that this situation does not occur in test area Zevenaar.

There are six TOP10vector **building** singletons. These singletons are inspected, manually, and visually by field inspection:

– five singletons are ‘free standing annexes’, situated in urban area, indicating therefore GBKN surveying rule errors, and

– one singleton is ‘mainbuilding’. Therefore, it is also a GBKN surveying rule error.

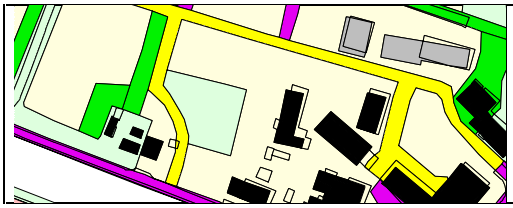


Fig. 53. A TOP10-vector **road** instance (in *yellow*) with no GBKN **road** counterpart.

5.6.3 Singletons of Roads

There are seven TOP10vector **road** singletons. The reason that their GBKN **road** counterparts are missing is that line-structured GBKN **road** elements, located in a municipality yard, were not object-structured in the 1996 experiment (Kadaster 1996), and consequently not included as object-structured GBKN **road** instances (**Fig. 53**).

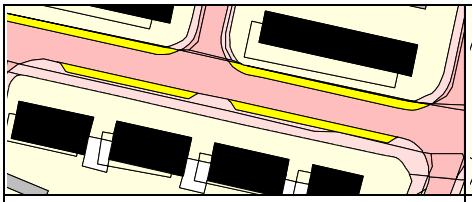


Fig. 54. Two ‘parkingstrips’ (south side, in *yellow*), and two ‘sidewalks’(north side, in *yellow*) inside a *3533* (‘street’).

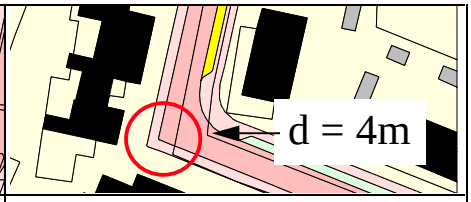


Fig. 55. A ‘parkingstrip’ (in *yellow*) lies inside a ‘street’, caused by an unexplained displacement (4.00m) of TOP10vector with regard to GBKN.

5.6.4 Singletons of Land

There are nine GBKN **land** singletons:

– two ‘sidewalks’, four ‘parkingstrips’, and one ‘flowerbed’, are all inside instances of TOP10vector class *3533* (‘street’, **Fig. 54**). These are all TOP10vector surveying rule errors.

– one ‘parkingstrip’ also inside a *3533* (‘street’), caused by an unexplained displacement — a gross error — of TOP10vector, with regard to GBKN (**Fig. 55**), and

– a very small ‘adjacent annex’ of 2.6 m^2, lying inside a TOP10vector *1000* (‘mainbuilding’). It is not compatible with this ‘environment’. Therefore it is singled out as singleton (**Fig. 56**).

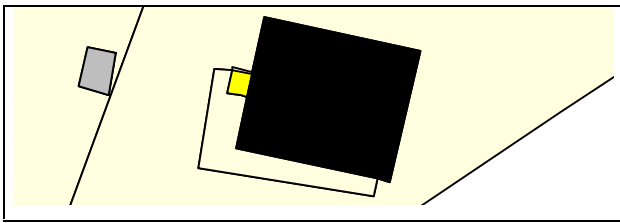


Fig. 56. A small ‘adjacent annex’ (2.6m^2, in *yellow*) lies inside a TOP10vector *1000* (*black outline*). It is not compatible with this ‘environment’.

If we consider gross errors as surveying rule errors, then eight GBKN singletons are TOP10vector surveying rule errors [25]; the last GBKN singleton is a model error.

There is also one TOP10vector **land** singleton. It is caused by a coding error.

5.7 Geometric Overlap and Stochasticity

In Section 2.8 it is stated that choosing ‘geometric overlap’ for ‘same location’ removes in a sense *stochasticity* from geo-data sets. Any amount of overlap is sufficient to declare instances of semantically similar classes as candidates for correspondences (**Fig. 57**).

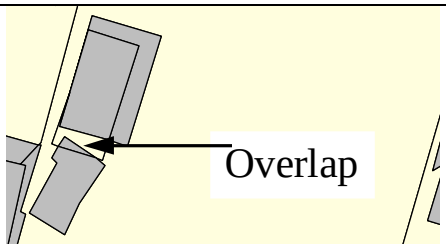


Fig. 57. Any amount of overlap creates candidates: two instances of GBKN *losbijgebouw* (‘free standing annex’, in *gray*) and one TOP10-vector *1000* (‘mainbuilding or annex’, with *black outline*), a 2-to-1 candidate.

However, stochasticity is introduced again in consistency checking. To filter out non-significant overlap a ‘heuristic’, a simple rule is used. For example, overlap is non-significant, if the fraction of ‘overlap area’ and ‘instance area’ is less than a certain threshold, *e.g.* 0.05 (Uitermark et al 1998).

[25] Gross, and systematic errors are surveying errors. Imprecision — also called *random* errors — is not an ‘error’, because imprecision is an inherent property of the surveying process, a stochastic process.

5.8 Summary and Discussion

This chapter presented the implementation of the reference model, designed in Chapter 4. While Chapter 4 was about semantic similarity between classes, this chapter was about semantic similarity between instances of these classes.

It started in Section 5.1 with the presentation of GBKN and TOP10vector test data sets, in **Table 7** and **Table 8** respectively. Based on reference model constructs, designed in Chapter 4 (and expressed as matrix **T**), and *refers_to* relationships between reference model classes and GBKN and TOP10vector classes (expressed as matrices **R** and **S** respectively), compatible object classes were determined with equation $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ **(57)**. The many relevant class labels in equation **(61)** reflect the *difference in abstraction* between GBKN and TOP10vector, because almost every GBKN class is a component class of a TOP10vector class.

	Candidates		*Instances*	
RM class ↓	*Single*	*Complex*	*# GBKN*	*# TOP10vec*
building	147	16	239	165
road	11	9	48	41
land + water	1	20	393	69
railway	-	1	1	5
Total	159	46	681	280

Table 12. Distribution of candidates for correspondences and their instances.

In Section 5.2, matrix **GT** of overlapping instances is used as a 'sieve' for semantically similar labels, resulting in 824 pairs of object identifiers of overlapping, and semantically similar instances. After aggregating pairs with identical identifiers, it ends up with 205 simple and complex candidates. This reduction from 824 pairs to 205 n-to-m candidates ($n \geq 1$ or $m \geq 1$) is once more related to the difference in abstraction between GBKN and TOP10vector (although GBKN has very large *terrein* — 'GBKN otherland' — instances).

In Section 5.3 to Section 5.5 every candidate was inspected by looking at data set attributes, their representations in maps, but also by visiting the test area. Half of **road** correspondences in Section 5.4, and almost all **land** correspondences in Section 5.5 are of a complex nature (**Table 12**). To be useful in update propagation, it is necessary to be more specific in statements about correspondences. In order to be more specific, complex correspondences should be broken down into simple correspondences, in a way suggested in Section 2.9.4.

Singletons — instances of data sets, not present in any correspondence — were presented in Section 5.6. Singletons reveal possible surveying rule errors, but also violations of underlying model assumptions, model errors.

However, almost every singleton is a surveying rule error. Two singletons are model errors (**Table 13**). Both are connected to the 'overlap issue'. The underlying assumption in this research that precision of topographic data sets is sufficient enough to use overlap for 'same location' for corresponding object instances, does

not always hold for objects, which are small with respect to the imprecision of data sets.

RM class ↓	*# GBKN singletns*	*# TOP10 singletns*	*# Surveying Rule Errors*	*# Model Errors*
building	4	6	9	1
road	-	7	7	-
land + water	9	1	9	1
railway	-	1	1	-
Total	13	15	26	2

Table 13. Distribution of singletons for GBKN and TOP10vector.

A final word on consistency checking, that is to say if candidates are consistent, or are influenced by surveying rule errors. In this research a definition of consistency was:

> Let (*b*1, *c*1) be a simple candidate, with class labels *b* and *c*, respectively, *i.e. b* and *c* are corresponding classes, and *b*1 and *c*1 overlap each other. Then (*b*1, *c*1) is consistent, if both *b*1 and *c*1 satisfy intensions of class *b* and class *c*.

GBKN		TOP10vector	
Class Label	Class Intension	Class Label	Class Intension
hoofd-gebouw	'mainbuilding'	*1000*	'mainbuilding' or 'remaining free standing annex': – in urban area, accessible, with area ≥ 9m^2, or – in urban area, not accessible, with area ≥ 50m^2, or – in rural area, with area ≥ 9m^2.
vast-bijgebouw	'adjacent annex'	*1050*	'barn': – in urban area, accessible, with area ≥ 9m^2, or – in urban area, not accessible, with area ≥ 50m^2, or – in rural area, with area ≥ 9m^2.
los-bijgebouw	'free standing annex': – in urban area – in rural area, with area ≥ 20m^2	*1073*	'greenhouse': – in urban area, accessible, with area ≥ 9m^2, or – in urban area, not accessible, with area ≥ 50m^2, or – in rural area, with area ≥ 9m^2.

Table 14. Summary of **building** surveying rules.

This definition was demonstrated for **building** surveying rules in Section 5.3, Section 5.5, and Section 5.6. To illustrate once again its importance for geographic data set integration, **building** surveying rules for GBKN and TOP10vector are summarized in **Table 14**. 'Satisfying both intensions' means that instances, for example in a candidate (*losbijgebouw*, *1000*), should not contradict class intensions of both *losbijgebouw* and *1000*, that is to say should imply a 'free standing annex', situated in:

- urban area, accessible, with area $\geq 9m^2$, or
- urban area, not accessible, with area $\geq 50m^2$, or
- rural area, with area $\geq 20m^2$

as can be seen in **Table 14**.

In conclusion, we can check consistency on the condition that we know class intensions, which are based on surveying rules. However, some conditions in intensions are context-dependent, like 'urban or rural area', and 'accessibility'. These attributes are not easy determined automatically.

Part 4: Evaluation and Conclusions

6 Evaluation Experimental Results

In Section 6.1 the experimental results of Chapter 5 are summarized. To evaluate these experimental results, it is necessary to investigate the representativeness of the test data (Section 6.2) and to establish a standard for completeness and correctness (Section 6.3).

6.1 Experimental Results

The experiment in geographic data set integration offers the following results.

Starting with 694 GBKN instances + 295 TOP10vector instances = 989 instances, we get 205 (simple and complex) candidates for correspondences, and 28 singletons.

When candidates are inspected, it is concluded that:

- 198 candidates are consistent, and
- 7 candidates are inconsistent. Inconsistency of candidates is caused by surveying rule errors, which are detected accordingly.

When singletons are inspected it is concluded that:

- 26 singletons are surveying rule errors, that is to say omissions caused in the production and maintenance of both data sets (‘production omissions’), and
- 2 singletons are model errors, that is to say violations of underlying model assumptions. These singletons are very small instances, therefore sensitive to the imprecision of the surveying process.

6.2 Sample Size of Test Data

Test area Zevenaar was chosen owing to the availability of an *object-structured* GBKN data set. GBKN is a nationwide mapping of buildings, roads, waterways, and railways, but is available in *line-structured* format *only*.

In 1996 experiments were done to restructure a line-structured GBKN data set into an object-structured GBKN data set (Landelijk Samenwerkingsverband GBKN 1997). The objective of this restructuring was to get insight and experience in possibilities and consequences of an object-structured GBKN.

An area of size 380 hectares was chosen for no particular reason in the municipality of Zevenaar, and subsequently restructured. Principles of restructuring were based on the GTM Standard (Ravi 1995), and in no respect related to TOP10vector. Thus, it is plausible that the chosen area is unbiased, and representative, due to its size of 380 hectares.

From this restructured GBKN data set of 380 hectares, a sample of 30 hectares was chosen as test area for this research. The location choice of the sample was based on a balanced distribution of object classes in urban area as well as in rural area. The sample size choice of 30 hectares (about one thousand instances), was based on manageability, because results had to be checked manually. Furthermore, the sample contained:

– all GBKN area object classes, and

– all TOP10vector area object classes, except for some land use types (like sand, or heather), water types (like ponds, or lakes) or road types (like roads for regional connecting traffic). However, missing TOP10vector area object classes are in no respect different from area object classes in the sample.

In addition, the sample has area classes from three groups:

1. Classes with a limited spatial extent (*e.g.* buildings)

2. Network-like classes (*e.g.* roads), and

3. Classes with holes (*e.g.* land use categories).

It is difficult to imagine that there is another group of area classes.

Given these preliminaries we conclude that the sample is representative for the purpose of testing the geographic data set integration framework.

6.3 A Standard for Completeness and Correctness

The objective of geographic data set integration is to establish explicit links between similar terrain descriptions — creating correspondences. Terrain descriptions are represented by object instances in geo-data sets. To establish links between object instances, a method was developed, and implemented as an automatic process with candidates for correspondences as output. Against which standard do we compare this output?

Alternatively, the output of the data set integration process could also have been produced by a manual procedure. A user, trained in the interpretation of maps, and instructed in the semantic similarity between object classes, is able enough to detect and determine similar terrain descriptions, correspondences. In addition, consistency checking can be done manually by inspecting attributes of instances. In fact, in this research all candidates were checked manually.

Ideally, for independent comparison of results, different users should be involved in the production of manual output. However, this is not done because the whole procedure — albeit tedious — is simple and straightforward enough to be done objectively.

Therefore, a ‘standard’ for completeness and correctness is the comparison of the output of the automatic process against the manual output of a trained and instructed user, in this case the author of this research.

Then, completeness and correctness of correspondences mean two things:

1. Did we find *all* correspondences, and
2. Are all correspondences found *really* correspondences, that is to say without errors?

As we have seen in Chapter 5 we found candidates for correspondences. Candidates were inspected and declared consistent, or declared inconsistent, that is to say influenced by surveying rule errors. We also found singletons. After inspection, most singletons were declared surveying rule errors. However, two singletons were not surveying rule errors, but model errors. Therefore, the combination of candidates *and* singletons, followed by systematic inspection, ensures us to find *all* correspondences ('completeness'), and to discriminate between consistent and inconsistent correspondences ('correctness'). Finally, we also know the cause of model errors: objects that are small with respect to the imprecision of data sets.

7 Conclusions

In this chapter, theory of Part 2 and practice of Part 3 converge into our final conclusions. First of all, there is a conclusion with respect to the research objective of this study in geographic data set integration (Section 7.1). Secondly, research questions stated earlier in Section 1.5 will now be answered (Section 7.2). Finally, there is an overall conclusion in Section 7.3, and recommendations for future research in Section 7.4.

7.1 Research Objective

The research objective of this study was to solve the problem of geographic data set integration, considering the differences between geo-data sets. More specifically the objective was to develop and implement a methodology that could reconcile the apparent differences between geo-data sets.

The conclusion with respect to the research objective is that the problem of geo-data set integration can be solved with an ontology-based approach. An ontology-based approach presupposes a domain ontology. Concepts from a domain ontology have been refined with respect to geo-data sets involved. These refined concepts have been structured in a reference model. The structure of the reference model, as well as the refined concepts, were based on surveying rules. In this way semantic interconnectedness was explained between object classes of different data sets. This semantic interconnectedness has been transferred to data set object instances, using location information from a geometric overlay of data sets, creating candidates for corresponding object instances. Then, the last step in solving the geo-data set integration problem was checking consistency of candidates with surveying rules, resulting in corresponding object instances.

7.2 Research Questions

Geographic data set integration is defined in this research as ‘the process of establishing explicit relationships between corresponding object instances in different, autonomously produced, geographic data sets of the same geographic space’. Given two different geographic data sets, we answer three research questions with respect to corresponding object instances:

1. What kind of relationships exist between corresponding object instances?
2. How can we find corresponding object instances, and under what conditions can we find them?
3. How certain are we about completeness and correctness of these corresponding object instances, and how can we check their consistency?

The answers to these questions will now be given.

7.2.1 Relationships Between Corresponding Object Instances (Question 1)

Corresponding object instances are instances from corresponding object classes. In this research, corresponding object classes are defined as classes referring to similar classes in a reference model. The structure of a reference model is based on two abstraction mechanisms, a generalization/specialization classification (a taxonomy), and a composite/component classification (a partonomy). Given this structure, three types of relationships between corresponding object classes were defined (Section 2.5.4):

1. Classes with a 'semantic equivalent relationship' refer to the same class in the taxonomy subgraph of the reference model, and are therefore 'equivalent' to each other.
2. Classes with a 'semantic related relationship' refer to classes at different levels in the taxonomy subgraph of the reference model, and have therefore a 'subclass/superclass' relationship to each other.
3. Classes with a 'semantic relevant relationship' refer to different levels in the partonomy subgraph of the reference model, and have therefore a 'composite class/component class' relationship to each other.

Classes with an 'equivalent', 'related', or 'relevant' relationship are defined as semantically similar classes, or compatible classes. Classes that are not semantically similar to each other are defined as incompatible classes.

The answer to this research question is that each and every pair of classes of different data sets belongs to one of these relationships (for a proof see Appendix B).

7.2.2 How to Find Corresponding Object Instances (Question 2)

In this research corresponding object instances are defined as instances:

1. From semantically similar classes,
2. Sharing same location, and
3. Consistent with surveying rules.

To find corresponding object instances, the following three steps have to be taken:

– step 1 is the construction of a reference model. A reference model is based on knowledge of surveying rules of geo-data sets to be integrated. Surveying rules state, which terrain situations to acquire for a geo-data set from a set of terrain situations, defined as concepts in a domain ontology (*e.g.* **Table 3**). Refining domain ontology classes into reference model classes creates a common universe of discourse (*e.g.* **Table 4**). With reference model classes as 'building blocks', structure is added to the reference model, in such a way that it reflects the type of semantic similarity between classes, that is to say, which classes are semantically similar classes

– step 2 is the execution of a geometric overlay operation, to determine ‘sharing same location’, and

– step 3 is consistency checking of candidates for corresponding object instances.

Therefore, conditions under which we are able to find corresponding object instances are:

1. Knowledge of surveying rules of geo-data sets (see Section 7.2.2.2).

2. Thematic and geometric overlap between geo-data sets (see Section 7.2.2.3), and

3. Object instances, with crisp, and complete boundaries (see Section 7.2.2.4).

However, there is also a zero condition with respect to applicable geo-data sets (Section 7.2.2.1).

7.2.2.1 Condition 0: Applicable Geo-Data Sets

The methodology of geographic data set integration was developed for topographic data sets with instances of area object classes:

– topographic data sets are two-dimensional (R2) vector data sets. A vector data set is a combination of a thematic and a geometric partition, also known as Single Valued Vector Map (SVVM) (Molenaar 1989). A thematic partition means that every terrain object belongs to exactly one object class. A geometric partition means that the combined geometric attributes of all terrain objects will result in a continuum with neither gaps nor overlap

– topographic data sets have their ‘natural’ imprecision caused by production processes. However, a typical aspect of topographic data sets is that no object instances are displaced for cartographic or representational reasons (traditionally, up to scale 1 : 12,500 - 15,000)

– in this research topographic data sets with instances of area object classes were studied, that is to say instances with a polygon as geometric attribute.

Therefore, the methodology in this research is applicable for topographic data sets with instances of area object classes.

7.2.2.2 Condition 1: Knowledge of Surveying Rules of Data Sets

In a reference model, concepts from a domain ontology are refined and structured in such a way, that the reference model explains semantic interconnectedness of geo-data sets. To realize this objective, surveying rules of geo-data sets have to be known. With this class level information, all concepts necessary to define a common universe for both data sets have to be identified (Section 4.2, Section 4.3, and **Table 4**). If knowledge of surveying rules is incomplete, then a solution for incomplete surveying rules is comparing and inspecting both geo-data sets at the instance level — visually, by overlaying both data sets (Section 4.4).

7.2.2.3 Condition 2: Thematic and Geometric Overlap

The key issue in geographic data set integration is finding corresponding object instances. This is a matching process. Matching is only possible if geo-data sets are:

- from the same geographic space, so there is geometric overlap, a trivial condition, and
- if their semantics, or themes — at least partial — can be expressed in a common language with a core of shared concepts. Possibly this core of shared concepts needs translation to a domain ontology, or is based on a domain ontology like the GTM Standard (Ravi 1995). The former situation needs more effort than the latter, but in both cases thematic overlap is a condition for geo-data set integration.

7.2.2.4 Condition 3: Crisp Object Instances

Crisp object instances are from crisp geo-data sets. A crisp geo-data set is defined as a set with instances, representing discontinuous real-world phenomena. For example, the transition between 'building' and 'surrounding terrain' is discontinuous. Topographic data sets represent discontinuous real-world phenomena. There is no problem with object class definition or object instance boundary definition. However, *locating* a boundary is an uncertainty problem, that depends on imprecision (stochasticity), and idealization. Imprecision depends on surveying instruments. Idealization depends on the precision to what extent a boundary can be defined as a line. For example, the boundary between 'grassland' and 'ditch' must be idealized as line, in order to be acquired efficiently and economically (Salzmann and Kenselaar 1998). Crisp geo-data sets have fuzzy geo-data sets as their opposite. In a fuzzy data set real-world phenomena are distributed gradually and continuously over space. For example, the boundary between beach and foreshore may be gradual, as through a transition zone rather than a discontinuous boundary (Cheng et al 2001).

7.2.3 Consistency, Completeness, and Correctness of Correspondences (Question 3)

Corresponding object instances should be consistent with surveying rules. In this research consistency is defined as satisfying class intensions of corresponding object classes. Thus, we can check consistency on the condition that we know class intensions, which are based on surveying rules. Therefore, this condition is similar to Condition 1 in Section 7.2.2.2.

The combination of candidates *and* singletons, followed by systematic inspection, ensures us that *all* correspondences ('completeness') are found, and to discriminate between consistent and inconsistent correspondences ('correctness'). However, in this respect two singletons were no surveying rule errors, but model errors. They concerned singletons with instances that are small with respect to the imprecision of data sets.

Therefore, the outcome of this statistical experiment is that out of a total of 205 candidates for correspondences, together with 28 singletons, there are two situations that can not be handled correctly by the methodology of this research.

7.3 Overall Conclusion

The overall conclusion of this research is that the framework for geographic data set integration (Chapter 2), with its formal mathematical foundation (Chapter 3), and its subsequent implementation (Chapter 4 and Chapter 5), is feasible if conditions mentioned in Section 7.2.2 are applicable. The application of this framework is most suitable for object classes with instances that are easy to identify and with a limited spatial extent (*e.g.* buildings).

7.4 Future Research

In this research several issues regarding geographic data set integration were encountered. Therefore, for the following issues future research is recommended:

1. Geographic data set integration presupposes object-structured data sets. GBKN is mostly a line-structured data set, and therefore not useful in data set integration. Research in restructuring GBKN into an object-structured format is therefore recommended.
2. Correspondences in geo-data set integration are more specific if object classes are more specific. For example, the GBKN data set in this research was expanded with additional subclasses. GBKN *inrichtingselement* was made more specific: ‘verge’, ‘cycletrack’, ‘sidewalk’, ‘parkingstrip’, etc (Section 4.1.1). Therefore, research in adding semantics to a geo-data set is recommended.
3. Complex correspondences are caused by (a) classes having a component class/composite class relationship with each other, or by (b) homogeneous decomposable object classes with instances, demarcated in an arbitrarily fashion (see Section 2.5.6). To be useful, for example in update propagation, it is desirable to be more specific in a statement about a correspondence of object instances. In order to be more specific, it is necessary to break down complex candidates into simple ‘candidates’. ‘Uniform elements’ and ‘Least common elements’ are suggested in Section 2.9.3. Therefore, research to break down complex correspondences into comparable elements is recommended (Uitermark et al 1999b).
4. Geometric overlap is used for ‘same location’ in correspondences. Any amount of geometric overlap is sufficient to declare semantically similar object instances as candidates for correspondences. To filter out non-significant geometric overlap a ‘heuristic’, a simple rule was used in this research (Section 5.7). The filtering out of non-significant overlap needs more sophistication. Research in mathematical models that deal with imprecision, expressed in terms of variances and probabilities, is therefore recommended (Winter 2000).

5. The restriction in this research to area objects is not a limitation, because point objects and line objects can be transformed temporarily into area objects by creating buffers and zones around their point-like and line-like locations (Harvey et al 1998). However, creating buffers and zones may temporarily violate the assumption of data sets with a thematic and geometric partition (see Section 1.9.2). Research is recommended to bring point objects and line objects within the framework of this research.
6. Other issues important for future research are:
   - temporal aspects, or history of data sets
   - how to create a 'best set' from two data sets
   - the role of fuzzy data sets (*e.g.* soil maps), and
   - the relationship between geo-data set integration and cartographic generalization.
7. It is envisioned that in the future there will be a class of software modules, called mediators, which mediate between several different geographic databases (Section 1.9.3). Research on how the framework of geo-data set integration fits in this mediator paradigm is recommended, with special attention to automatic update propagation.

8 Author and Subject Index

K

L

M

O

P

R

S

T

U

V

W

9 References

Algemene Winkler Prins (1958). Encyclopedie (editor H.R. Hoetink), Vol. 7. Amsterdam: Elsevier, 798 pages.

Artale, A., E. Franconi, N. Guarino, and L. Pazzi (1996). "Part-whole relations in object-centered systems: an overview". *Data & Knowledge Engineering*, Vol. 20, No. 3, pp. 347-383.

Benslimane, D., E. Leclercq, M. Savonnet, M.-N. Terrasse, and K. Yétongnon (2000). "On the definition of generic multi-layered ontologies for urban applications". *Computers, Environment and Urban Systems*, Vol. 24, No. 3, pp. 191-214.

Bishr, Y.A. (1997). "Semantic aspects of interoperable GIS". PhD Thesis, Wageningen Agricultural University, Wageningen, The Netherlands (ITC Publication Series No. 56).

Bishr, Y.A. (1998). "Overcoming the semantic and other barriers to GIS interoperability". *Int. J. Geographical Information Science*, Vol. 12, No. 4, pp. 299-314.

Braspenning, P. and I. Lemmens (1997). "Kennisrepresentatietalen in de negentiger jaren (*in Dutch*)". *Informatie*, Vol. 39, No. 6, pp. 40-48.

Castano, S., V. De Antonellis, and S. De Capitani di Vimercati (2001). "Global viewing of heterogeneous data sources". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No. 2, pp. 277-297.

Cheng, T., M. Molenaar, and H. Lin (2001). "Formalizing fuzzy objects from uncertain classification results". *Int. J. Geographical Information Science*, Vol. 15, No. 1, pp. 27-42.

Devogele, T., C. Parent, and S. Spaccapietra (1998). "On spatial database integration". *Int. J. Geographical Information Science*, Vol. 12, No. 4, pp. 335-352.

Devogele, T., J. Trevisan, and L. Raynal (1996). *Building a multi-scale database with scale-transition relationships.* Proceedings 7th International symposium on Spatial Data Handling SDH'96 (M.J. Kraak and M. Molenaar, eds.). Delft, The Netherlands, August, 12-16. International Geographical Union. Vol. I, pp. 6.19-6.33.

Dupont, Y. (1994). *Resolving fragmentation conflicts in schema integration.* Proceedings Entity-Relationship Approach--ER'94 (P. Loucopoulos, ed.).

Berlin. Lecture Notes in Computer Science, Vol. 881. Springer, Berlin, pp. 513-532.

ESRI (1994). Introducing ArcView. Redlands, CA, USA: Environmental Systems Research Institute, Inc, 98 pages.

Guarino, N. (1997). *Semantic matching: formal ontological distinctions for information organization, extraction, and integration.* Proceedings International Summer School, SCIE-97 (M.T. Pazienza, ed.). Frascati, Italy. Lecture Notes in Computer Science, Vol.1299. Springer, Berlin, pp. 139-170.

Hadzilakos, T., G. Halaris, M. Kavouras, M. Kokla, G. Panopoulos, I. Paraschakis, T. Sellis, L. Tsoulos, and M. Zervakis (2000). "Interoperability and definition of a national standard for geospatial data: the case of the Hellenic Cadastre". *JAG*, Vol. 2, No. 2, pp. 120-128.

Harvey, F., F. Vauglin, and A. Atef Bel Hadj (1998). *Geometric matching of areas. Comparison measures and association links.* Proceedings 8th International Symposium on Spatial Data Handling SDH'98 (T.K. Poiker and N. Chrisman, eds.). Vancouver, Canada, July 11-15. International Geographical Union, pp. 557-568.

Heres, L., J. den Hartog-Sprockel, and P. Plomp (1997). "NWB-Kernmodel en extensies (*in Dutch*)". Report LAV2_007. Adviesdienst Verkeer en Vervoer, Heerlen.

Huhns, M.N. and M.P. Singh (1997). "Ontologies for agents". *IEEE Internet Computing*, Vol. 1, No. 6, pp. 81-83.

Kadaster (1992). "Bestek inzake de fotogrammetrische kartering in het kader van het project Duiven-Zevenaar (*in Dutch*)". Report. Dienst van het Kadaster en de Openbare Registers, Apeldoorn.

Kadaster (1996). "Onderzoek objectgerichte GBKN (OOG). De objectgerichte GBKN van Zevenaar (*in Dutch*)". Report 6166. Dienst van het Kadaster en de Openbare Registers, Apeldoorn.

Kilpeläinen, T. (2000). "Maintenance of multiple representation databases for topographic data". *The Cartographic Journal*, Vol. 37, No. 2, pp. 101-107.

Kim, C.-J. (1999). "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.

Kuhn, W. (1996). Semantics of geographic information. Geoinfo-series, Vol. 7. Vienna: Department of Geoinformation, Technical University, 108 pages.

Landelijk Samenwerkingsverband GBKN (1997). “Onderzoek objectgerichte GBKN (OOG). De vervaardiging van de objectgerichte GBKN van Zevenaar. Nota ten behoeve van de Definitiefase”. Report. Landelijk Samenwerkingsverband GBKN, Amersfoort.

Larson, J.A., S.B. Navathe, and R. Elmasri (1989). “A theory of attribute equivalence in databases with application to schema integration”. *IEEE Transactions on Software Engineering*, Vol. 15, No. 4, pp. 449-463.

Laurini, R. (1993). *Updating and sharing geographic information: GIS challenges for the year 2000.* Proceedings Fourth European Conference & Exhibition on Geographical Information EGIS'93. Genoa, Italy, March 29 - April 1. EGIS Foundation, Utrecht/Amsterdam, pp. 1656-1667.

Laurini, R. (1998). “Spatial multi-database topological continuity and indexing: a step towards seamless GIS data interoperability”. *Int. J. Geographical Information Science*, Vol. 12, No. 4, pp. 373-402.

Lipschutz, S. (1976). Discrete mathematics. Schaum Outline Series. New York: McGraw-Hill, 249 pages.

Maeder, R.E. (1994). “Logic Programming I: The interpreter”. *The Mathematica Journal*, Vol. 4, No. 1, pp. 53-63.

Malpas, J. (1987). PROLOG: a relational language and its applications. London: Prentice-Hall International, 465 pages.

Mars, N.J.I. (1995). *What is an ontology?* Proceedings Seminar on the impact of ontologies on reuse, interoperability and distributed processing (A. Goodall, ed.). London, November 7. Unicom, Uxbridge, Middlesex, UK, pp. 9-19.

Mizoguchi, R., J. Vanwelkenhuysen, and M. Ikeda (1995). *Task ontology for reuse of problem solving knowledge.* Proceedings Second International Conference on Building and Sharing Very Large-Scale Knowledge Bases (N.J.I. Mars, ed.). Enschede, The Netherlands. IOS Press, Amsterdam, pp. 46-57.

Molenaar, M. (1989). “Single valued vector maps: a concept in Geographic Information Systems”. *Geo-Informationssysteme*, Vol. 2, No. 1, pp. 18-26.

Molenaar, M. (1998). An introduction to the theory of spatial object modelling for GIS. London: Taylor and Francis, 246 pages.

Open GIS Consortium Inc. (1996). “Open GIS Consortium. Promoting distributed geoprocessing through cooperative technology development, partnerships, and industry consensus”. Report. Open GIS Consortium, Wayland, Massachusetts, USA.

Open GIS Consortium Inc. (1998). “The OpenGIS Abstract Specification Model”. Report 98-100. Open GIS Consortium, Wayland, Massachusetts, USA.

Papaioannou, V. (1998). "HERE: A hypermedia environment for requirements engineering". PhD Thesis. Department of Computation, University of Manchester Institute of Science and Technology (UMIST), Manchester, UK.

Ravi (1995). "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)". Report. Ravi Netherlands Council for Geographic Information, Amersfoort, The Netherlands.

Salzmann, M.A. (1996). "On the modeling of geometric quality for large-scale mapping products". *Surveying and Land Information Systems*, Vol. 56, No. 3, pp. 149-155.

Salzmann, M.A. and F. Kenselaar (1998). *A unified approach to quality control for cadastral surveying: concepts and implementation of the HTW-manual.* Proceedings FIG XXI international congress. Brighton, UK, July 19-25.

Sester, M., K.-H. Anders, and V. Walter (1998). "Linking objects of different spatial data sets by integration and aggregation". *GeoInformatica*, Vol. 2, No. 4, pp. 335-357.

Sheth, A. and V. Kashyap (1993). *So far (schematically) yet so near (semantically).* Proceedings IFIP (D.K. Hsiao, E.J. Neuhold, and R. Sacks-Davis, eds.). Elsevier Science Publishers B.V. (North-Holland), pp. 283-312.

Smith, B. (1996). "Mereotopology: a theory of parts and boundaries". *Data & Knowledge Engineering*, Vol. 20, No. 3, pp. 287-303.

Spaccapietra, S., C. Parent, and Y. Dupont (1992). "Model independent assertions for integration of heterogeneous schemas". *VLDB Journal*, Vol. 1, pp. 81-126.

TDN (1995). "Nederland Digitaal. Produktbeschrijving TOP10vector (*in Dutch*)". Report. Topografische Dienst, Emmen.

TDN (1999). "Regels voor de verkenning van topografische objecten (11e revisie) (*in Dutch*)". Report. Topografische Dienst, Emmen.

TDN and Kadaster (1995). "GBKN-mutaties en TOP10vector. Onderzoek naar de uitwisseling van GBKN-mutaties met TOP10vector. Deelonderzoek: GBKN-inhoud versus TOP10vector-inhoud (*in Dutch*)". Report. Topografische Dienst/Dienst van het Kadaster en de Openbare Registers, Emmen/Apeldoorn.

TDN and Kadaster (1999). "GBKN-mutaties en TOP10vector. Onderzoek naar de uitwisseling van GBKN-mutaties met TOP10vector. Deelonderzoek: Tussentijdse herziening van TOP10vector met GBKN-mutaties (*in Dutch*)".

Report. Topografische Dienst/Dienst van het Kadaster en de Openbare Registers, Emmen/Apeldoorn.

TDN, Provinciaal Vastgoed Overleg Zeeland, and Kadaster (1997). "GBKN-mutaties en TOP10vector. Onderzoek naar de uitwisseling van GBKN-mutaties met TOP10vector. Deelonderzoek: Bijhouding van TOP10vector met de GBKN (*in Dutch*)". Report. Topografische Dienst/Provinciaal Vastgoed Overleg Zeeland/Dienst van het Kadaster en de Openbare Registers, Emmen/Middelburg/Apeldoorn.

Uitermark, H.T. (1996). *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.). Barcelona, Spain, March 27-29. IOS Press, Amsterdam. Vol. I, pp. 92-95.

Uitermark, H.T., P.J.M. van Oosterom, N.J.I. Mars, and M. Molenaar (1998). *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.). Vancouver, Canada, July 11-15. International Geographical Union, pp. 580-591.

Uitermark, H.T., P.J.M. van Oosterom, N.J.I. Mars, and M. Molenaar (1999a). *Ontology-based geographic data set integration.* Proceedings International Workshop on Spatio-Temporal Database Management STDBM'99 (M.H. Böhlen, C.S. Jensen, and M.O. Scholl, eds.). Edinburgh, Scotland, UK, September, 10-11. Lecture Notes in Computer Science, Vol. 1678. Springer, Berlin, pp. 60-78.

Uitermark, H.T., A.B.M. Vogels, and P.J.M. van Oosterom (1999b). *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.). Zürich, Switzerland, March 10-12. Lecture Notes in Computer Science, Vol. 1580. Springer, Berlin, pp. 177-188.

van Asperen, P.C.M. (1996). *Digital updates at the Dutch Topographic Service.* Proceedings XVIII ISPRS-Congress. Vienna, Austria, July 9-19.

van der Schans, R. (1994). *Theoretische aspecten van DLM/DKM: symmetrische structuren (in Dutch).* Proceedings Studiedag DLM/DKM (R. van der Schans, ed.). Emmen, The Netherlands. Nederlandse Commissie voor Geodesie. Vol. 33, pp. 3-37.

van der Schans, R. (1997). *A quest for optimal expression of objects and actions in GIS.* Proceedings GIS-Interfaces for Environmental Control (J. van Arragon, ed.). Utrecht, February, 19-20. NexpRI, pp. 11-20.

van der Veen, J.B. and H.T. Uitermark (1995). “Naar een objectgerichte GBKN?! Van basiskaart tot basisbestand via het objectbegrip (*in Dutch*)”. *Geodesia*, No. 2, pp. 83-88.

van der Vet, P.E. and N.J.I. Mars (1998). “Bottom-up construction of ontologies”. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, No. 4, pp. 513-526.

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

van Oosterom, P.J.M. (1997). *Maintaining consistent topology including historical data in a large spatial database.* Proceedings Auto-Carto 13. Seattle, Washington, USA, pp. 327-336.

van Putten, J.D. (1997). “Experiences with the GAP-tree”. Master Thesis. Computer Science, University of Utrecht, Utrecht (INF-SCR-97-30).

van Wijngaarden, F.A., J.D. van Putten, P.J.M. van Oosterom, and H.T. Uitermark (1997). *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.). Las Vegas, Nevada, USA, November 13-14. ACM, New York, pp. 71-76.

Vogels, A.B.M. (1999). “Propagatie van GBKN-wegenmutaties naar de TOP10vector (*in Dutch*)”. Master Thesis. Geodesy Department, Technical University Delft, Delft, The Netherlands.

Wache, H., T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner (2001). “Ontology-based integration of information. A survey of existing approaches”. http://www.tzi.de/buster/papers/survey.pdf

Wiederhold, G. (1992). “Mediators in the architecture of future information systems”. *IEEE Computer*, Vol. 25, No. 3, pp. 38-49.

Wille, R. (1992). “Concept lattices and conceptual knowledge systems”. *Computers Math. Applic.*, Vol. 23, No. 6-9, pp. 493-515.

Winter, S. (2000). “Uncertain topological relations between imprecise regions”. *Int. J. Geographical Information Science*, Vol. 14, No. 5, pp. 411-430.

Wintraecken, J.J.V.R. (1987). Informatie-analyse volgens NIAM in theorie en praktijk. The Hague: Academic Press, 400 pages.

Wolfram, S. (1996). The Mathematica Book, 3rd ed. Champaign, IL, USA: Wolfram Media, 1403 pages.

Appendix A

In this appendix the computing model $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ — applied in Section 5.1 — is illustrated.

A1 Computing Semantically Similar Labels with $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$

The reference model (RM) — developed in Chapter 4 — is schematically depicted in **Fig. 30** up to **Fig. 40**.

In **Fig. 40** we see at level [0]: 'geo-object', as root; at level [1]: 'compobject', 'railway', and 'T_freeannex'; at level [2]: 'barn', 'compbldg', etc; and at level [3]: all component classes — the lowest level of the RM (and not shown in **Fig. 40**). This partitioning of RM labels into subsets at different levels are the A_L subsets mentioned in **(15)** in Section 3.2.

Now we set up relation matrix **T**[*L*] as in **(16)**. Each time there is a *taxon* or *parton* predicate between RM labels at level L and $L - 1$, a '1' is put in matrix **T**[*L*], otherwise a '0'. See **T**[1] and **T**[2] in **Table A-1** of this Appendix.

Next, with subsets A_L of RM labels — in the second column of **Table A-1**— we determine subsets B_L of GBKN labels.

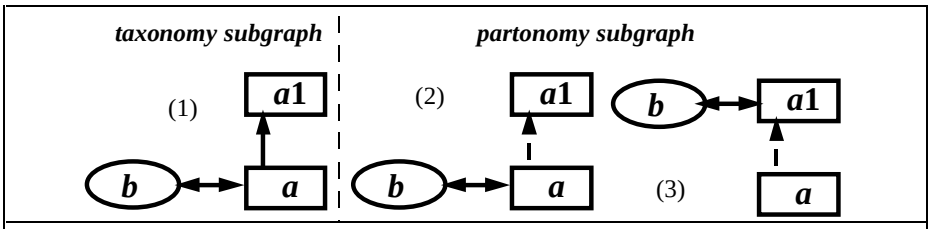


Fig. A-1 (*a*, *b*) belongs to relation *R*[*L*] iff (1) *a* refers to *b*, and *a* is a subclass of *a*1 (*left*), or (2) *a* refers to *b*, and *a* is a component class of *a*1 (*middle*), or (3) *a*1 refers to *b*, and *a*1 has *a* as component class (*right*).

For example, if we enumerate RM situations in **Fig. A-1** then:

1. in **Fig. 30** (*right*) 'T_freeannex' refers to *losbijgebouw*, and 'T_freeannex' is a subclass from 'geo-object' (**Fig. 40**), therefore *losbijgebouw* belongs to the same level as 'T_freeannex'. See subset B_1 in column two in **Table A-2**.
2. in **Fig. 39** 'adjannex' refers to *vastbijgebouw*, and 'adjannex' is a component class of 'T_other', therefore *vastbijgebouw* belongs to the same level as 'adjannex'. See subset B_3 in column two in **Table A-2**.
3. in **Fig. 33** 'G_other' refers to *terrein*, and 'otherland' is a component class of 'G_other', therefore *terrein* belongs to the same level as 'otherland'. See subset B_3 in column two in **Table A-2**.

L	A_L	*Relation matrix* **T**[L] *between level* L and $L-1$
[0]	geo-object	-
[1]	compobject railway T_freeannex	$\mathbf{T}[1] = \begin{pmatrix} & \text{geo-object} \\ \text{compobject} & 1 \\ \text{railway} & 1 \\ \text{T\_freeannex} & 1 \end{pmatrix}$
[2]	barn compbldg G_other greenhouse T_arable T_grass T_other T_wood T_3103 T_3203 T_3303 T_3533 T_3603	$\mathbf{T}[2] = \begin{pmatrix} & \text{compobject} & \text{railway} & \text{T\_freeannex} \\ \text{barn} & 0 & 0 & 1 \\ \text{compbldg} & 1 & 0 & 0 \\ \text{G\_other} & 1 & 0 & 0 \\ \text{greenhouse} & 0 & 0 & 1 \\ \text{T\_arable} & 1 & 0 & 0 \\ \text{T\_grass} & 1 & 0 & 0 \\ \text{T\_other} & 1 & 0 & 0 \\ \text{T\_wood} & 1 & 0 & 0 \\ \text{T\_3103} & 1 & 0 & 0 \\ \text{T\_3203} & 1 & 0 & 0 \\ \text{T\_3303} & 1 & 0 & 0 \\ \text{T\_3533} & 1 & 0 & 0 \\ \text{T\_3603} & 1 & 0 & 0 \end{pmatrix}$
[3]	adjannex arableland conngt2m conngt4m conngt7m cycletrack ditch flowerbed grassland woodland mainbuilding otherland parkingstrip remfreeannex sidewalk street verge	$\mathbf{T}[3] = \begin{pmatrix} 0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&1&0&0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&1&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&1&0&0&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1 \end{pmatrix}$

Table A-1. Relation matrices **T**[L] between levels $L = \{0, 1, 2, 3\}$ of the RM in **Fig. 40**. Note that for space reasons, labels are omitted in **T**[3].

Next, we set up relation matrices **R**[L] between RM labels and GBKN data set labels. See column three of **Table A-2**. Note that there are no GBKN object classes at level [2], therefore **R**[2] does not exist.

L	B_L	*Inverse relation* $\mathbf{R}[L]^{-1}$ *between RM labels and GBKN labels*
[1]	*losbijgebouw* *spoorbaan*	$\mathbf{R}[1]^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
[2]	∅	-
[3]	*berm* *bermsloot* *bloemenperk* *fietspad* *hoofdgebouw* *losbijgebouw* *parkeerstrook* *rijbaan* *terrein* *trottoir* *vastbijgebouw*	$\mathbf{R}[3]^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Table A-2. Inverse relation matrix $\mathbf{R}[L]^{-1}$ for each level between RM labels and GBKN labels. Note that for space reasons $\mathbf{R}[L]^{-1}$ is shown instead of $\mathbf{R}[L]$.

In a similar way subsets C_L of TOP10vector labels and relation matrices $\mathbf{S}[L]$ between RM labels and TOP10vector labels are set up. See columns two and three in **Table A-3**.

L	C_L	*Inverse relation* $\mathbf{S}[L]^{-1}$ *between RM labels and TOP10vector labels*
[1]	*4000*	$\mathbf{S}[1]^{-1} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$
[2]	*1050* *1073*	$\mathbf{S}[2]^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
[3]	*1000* *3103* *3203* *3303* *3533* *3603* *5023* *5203* *5213* *5263*	$\mathbf{S}[3]^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$

Table A-3. Inverse relation matrix $\mathbf{S}[L]^{-1}$ for each level between RM label and TOP10vector. Note that for space reasons $\mathbf{S}[L]^{-1}$ is shown instead of $\mathbf{S}[L]$.

Next, submatrices **R**[1], **R**[2], and **R**[3] are regrouped into matrix **R**, as indicated in Section 3.4.3.2:

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ \mathbf{R}[1] & 0 & 0 \\ 0 & \mathbf{R}[2] & 0 \\ 0 & 0 & \mathbf{R}[3] \end{pmatrix} \text{ and, with } \mathbf{R}[2] \text{ not existing: } \mathbf{R} = \begin{pmatrix} 0 & 0 \\ \mathbf{R}[1] & 0 \\ 0 & 0 \\ 0 & \mathbf{R}[3] \end{pmatrix} \textbf{(Table A-5)}$$

Also, submatrices **S**[1], **S**[2], and **S**[3] are regrouped into matrix **S**, as indicated in Section 3.4.3.3:

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 \\ \mathbf{S}[1] & 0 & 0 \\ 0 & \mathbf{S}[2] & 0 \\ 0 & 0 & \mathbf{S}[3] \end{pmatrix} \textbf{(Table A-6)}$$

Next, submatrices **T**[0], **T**[1], **T**[2], and **T**[3] are regrouped into matrix **T** as indicated in Section 3.4.3.1:

$$\begin{pmatrix} 1 & \mathbf{T}[1]^{-1} & \mathbf{T}[1]^{-1}.\mathbf{T}[2]^{-1} & \mathbf{T}[1]^{-1}.\mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1} \\ \mathbf{T}[1] & \mathbf{I} & \mathbf{T}[2]^{-1} & \mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1} \\ \mathbf{T}[2].\mathbf{T}[1] & \mathbf{T}[2] & \mathbf{I} & \mathbf{T}[3]^{-1} \\ \mathbf{T}[3].\mathbf{T}[2].\mathbf{T}[1] & \mathbf{T}[3].\mathbf{T}[2] & \mathbf{T}[3] & \mathbf{I} \end{pmatrix}$$

With **R**, **T**, and **S** in this format we can compute $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ symbolically:

$$\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S} = \begin{pmatrix} 0 & \mathbf{R}[1]^{-1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{R}[3]^{-1} \end{pmatrix} \cdot$$

$$\begin{pmatrix} 1 & \mathbf{T}[1]^{-1} & \mathbf{T}[1]^{-1}.\mathbf{T}[2]^{-1} & \mathbf{T}[1]^{-1}.\mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1} \\ \mathbf{T}[1] & \mathbf{I} & \mathbf{T}[2]^{-1} & \mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1} \\ \mathbf{T}[2].\mathbf{T}[1] & \mathbf{T}[2] & \mathbf{I} & \mathbf{T}[3]^{-1} \\ \mathbf{T}[3].\mathbf{T}[2].\mathbf{T}[1] & \mathbf{T}[3].\mathbf{T}[2] & \mathbf{T}[3] & \mathbf{I} \end{pmatrix} \cdot$$

$$\begin{pmatrix} 0 & 0 & 0 \\ \mathbf{S}[1] & 0 & 0 \\ 0 & \mathbf{S}[2] & 0 \\ 0 & 0 & \mathbf{S}[3] \end{pmatrix} =$$

$$\begin{pmatrix} \mathbf{R}[1]^{-1} \cdot \mathbf{T}[1] & \mathbf{R}[1]^{-1} & \mathbf{R}[1]^{-1} \cdot \mathbf{T}[2]^{-1} & \mathbf{R}[1]^{-1} \cdot \mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1} \\ \mathbf{R}[3]^{-1} \cdot \mathbf{T}[3].\mathbf{T}[2].\mathbf{T}[1] & \mathbf{R}[3]^{-1} \cdot \mathbf{T}[3].\mathbf{T}[2] & \mathbf{R}[3]^{-1} \cdot \mathbf{T}[3] & \mathbf{R}[3]^{-1} \end{pmatrix} \cdot$$

$$\begin{pmatrix} 0 & 0 & 0 \\ \mathbf{S}[1] & 0 & 0 \\ 0 & \mathbf{S}[2] & 0 \\ 0 & 0 & \mathbf{S}[3] \end{pmatrix} =$$

$$\begin{pmatrix} \mathbf{R}[1]^{-1} \cdot \mathbf{S}[1] & \mathbf{R}[1]^{-1} \cdot \mathbf{T}[2]^{-1} \cdot \mathbf{S}[2] & \mathbf{R}[1]^{-1} \cdot \mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1} \cdot \mathbf{S}[3] \\ \mathbf{R}[3]^{-1} \cdot \mathbf{T}[3].\mathbf{T}[2] \cdot \mathbf{S}[1] & \mathbf{R}[3]^{-1} \cdot \mathbf{T}[3] \cdot \mathbf{S}[2] & \mathbf{R}[3]^{-1} \cdot \mathbf{S}[3] \end{pmatrix}$$

Table A-4. $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ in symbolical notation.

And, if we substitute the values of **Table A-2** into matrix **R** we get:

GBKN label→ / *RM label ↓*	*losbijgebouw*	*spoorbaan*	*berm*	*bermslot*	*bloemperk*	*fietspad*	*hofdgebouw*	*losbijgebouw*	*parkeerstrook*	*rijbaan*	*terrein*	*trottoir*	*vastbijgebw*
geo-object	**0**							**0**					
compobject	0	0											
railway	0	1						**0**					
T_freeannex	1	0											
barn													
compbldg													
G_other													
greenhouse													
T_arable													
T_grass													
T_other	**0**							**0**					
T_wood													
T_3103													
T_3203													
T_3303													
T_3533													
T_3603													
adjannex			0	0	0	0	0	0	0	0	0	0	1
arableland			0	0	0	0	0	0	0	0	1	0	0
conngt2m			0	0	0	0	0	0	0	1	0	0	0
conngt4m			0	0	0	0	0	0	0	1	0	0	0
conngt7m			0	0	0	0	0	0	0	1	0	0	0
cycletrack			0	0	0	1	0	0	0	0	0	0	0
ditch			0	1	0	0	0	0	0	0	0	0	0
flowerbed			0	0	1	0	0	0	0	0	0	0	0
grassland	**0**		0	0	0	0	0	0	0	0	1	0	0
woodland			0	0	0	0	0	0	0	0	1	0	0
mainbuilding			0	0	0	0	1	0	0	0	0	0	0
otherland			0	0	0	0	0	0	0	0	1	0	0
parkingstrip			0	0	0	0	0	0	1	0	0	0	0
remfreeannex			0	0	0	0	0	1	0	0	0	0	0
sidewalk			0	0	0	0	0	0	0	0	0	1	0
street			0	0	0	0	0	0	0	1	0	0	0
verge			1	0	0	0	0	0	0	0	0	0	0

Table A-5. Matrix **R** between RM labels and GBKN labels. **R***rec* is shown in *light gray*, **R***rcc* in *dark gray* (**0** is a null matrix).

Similarly, we substitute the values of **Table A-3** into matrix **S**:

TOP10 label→ / *RM label ↓*	*4000*	*10503*	*10733*	*10003*	*31033*	*32033*	*33033*	*35333*	*36033*	*50233*	*52033*	*52133*	*52633*
geo-object	**0**	**0**		**0**									
compobject	0												
railway	1	**0**											
T_freeannex	0												
barn		1	0										
compbldg		0	0										
G_other		0	0										
greenhouse		0	1										
T_arable		0	0										
T_grass		0	0										
T_other	**0**	0	0					**0**					
T_wood		0	0										
T_3103		0	0										
T_3203		0	0										
T_3303		0	0										
T_3533		0	0										
T_3603		0	0										
adjannex				0	0	0	0	0	0	0	0	0	1
arableland				0	0	0	0	0	0	0	1	0	0
conngt2m				0	0	0	1	0	0	0	0	0	0
conngt4m				0	0	1	0	0	0	0	0	0	0
conngt7m				0	1	0	0	0	0	0	0	0	0
cycletrack				0	0	0	0	0	1	0	0	0	0
ditch				0	0	0	0	0	0	1	1	1	1
flowerbed				0	0	0	0	0	0	0	0	1	1
grassland	**0**	**0**		0	0	0	0	0	0	0	0	1	0
woodland				0	0	0	0	0	0	1	0	0	0
mainbuilding				1	0	0	0	0	0	0	0	0	0
otherland				0	0	0	0	0	0	0	0	0	1
parkingstrip				0	0	0	0	0	0	0	0	1	1
remfreeannex				1	0	0	0	0	0	1	1	1	1
sidewalk				0	0	0	0	0	0	0	0	1	1
street				0	0	0	0	1	0	0	0	0	0
verge				0	1	1	1	1	1	0	0	0	0

Table A-6 Matrix **S** between RM labels and TOP10vector labels. **S***rec* is shown in *light gray*, **S***rcc* in *dark gray* (**0** is a null matrix).

Now we display **T** numerically. It is split — due to space reasons — into two tables:

1	$\mathbf{T}[1]^{-1}$	$\mathbf{T}[1]^{-1}.\mathbf{T}[2]^{-1}$
$\mathbf{T}[1]$	**I**	$\mathbf{T}[2]^{-1}$
$\mathbf{T}[2].\mathbf{T}[1]$	$\mathbf{T}[2]$	**I**
$\mathbf{T}[3].\mathbf{T}[2].\mathbf{T}[1]$	$\mathbf{T}[3].\mathbf{T}[2]$	$\mathbf{T}[3]$

Table A - **7**

$\mathbf{T}[1]^{-1}.\mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1}$	
$\mathbf{T}[2]^{-1}.\mathbf{T}[3]^{-1}$	
$\mathbf{T}[3]^{-1}$	
I	

Table A - **8**

RM label→ / *RM label* ↓	geo	com	rlw	Tfa	brn	cbl	Gol	gnh	Tald	Tgl	Tol	Twl	T31	T32	T33	T35	T36
geo-object	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
compobject	1				0	1	1	0	1	1	1	1	1	1	1	1	1
railway	1		**I**		0	0	0	0	0	0	0	0	0	0	0	0	0
T_freeannex	1				1	0	0	1	0	0	0	0	0	0	0	0	0
barn	1	0	0	1													
compbldg	1	1	0	0													
G_other	1	1	0	0													
greenhouse	1	0	0	1													
T_arable	1	1	0	0													
T_grass	1	1	0	0													
T_other	1	1	0	0					**I**								
T_wood	1	1	0	0													
T_3103	1	1	0	0													
T_3203	1	1	0	0													
T_3303	1	1	0	0													
T_3533	1	1	0	0													
T_3603	1	1	0	0													
adjannex	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
arableland	2	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
conngt2m	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
conngt4m	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
conngt7m	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
cycletrack	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ditch	4	4	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
flowerbed	2	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
grassland	2	2	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
woodland	2	2	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
mainbuilding	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
otherland	2	2	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
parkingstrip	2	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
remfreeannex	5	5	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0
sidewalk	2	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
street	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
verge	5	5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table A-7. Matrix **T** (row 1-34, column 1-17). **I** is the identity matrix.

RM label→ / *RM label* ↓	adj	ald	ct2	ct4	ct7	ctk	dth	fbd	gld	wld	mb	old	psp	rfa	sdk	strt	vrg
geo-object	1	2	1	1	1	1	4	2	2	2	1	2	2	5	2	1	5
compobject	1	2	1	1	1	1	4	2	2	2	1	2	2	5	2	1	5
railway	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T_freeannex	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
compbldg	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
G_other	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
greenhouse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T_arable	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
T_grass	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0
T_other	1	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0
T_wood	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0
T_3103	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
T_3203	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
T_3303	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T_3533	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
T_3603	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
adjannex	**I**																
arableland																	
conngt2m																	
conngt4m																	
conngt7m																	
cycletrack																	
ditch																	
flowerbed																	
grassland																	
woodland																	
mainbuilding																	
otherland																	
parkingstrip																	
remfreeannex																	
sidewalk																	
street																	
verge																	

Table A-8. Matrix T (row 1-34, column 18-34). **I** is the identity matrix.

With this preliminary work we now compute corresponding object classes with $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$ using **Table A-5** up to **Table A-8**.

TOP10 label→ / *GBKN label* ↓	*4000*	*10503*	*10703*	*10003*	*31033*	*32033*	*33033*	*35333*	*36033*	*50223*	*52033*	*52133*	*52633*
losbijgebouw	0	1	1	0	0	0	0	0	0	0	0	0	0
spoorbaan	1	0	0	0	0	0	0	0	0	0	0	0	0
berm	0	0	0	0	1	1	1	1	1	0	0	0	0
bermsloot	0	0	0	0	0	0	0	0	0	1	1	1	1
bloemenperk	0	0	0	0	0	0	0	0	0	0	0	1	1
fietspad	0	0	0	0	0	0	0	0	1	0	0	0	0
hoofdgebouw	0	0	0	1	0	0	0	0	0	0	0	0	0
losbijgebouw	0	0	0	1	0	0	0	0	0	1	1	1	1
parkeerstrook	0	0	0	0	0	0	0	0	0	0	0	1	1
rijbaan	0	0	0	0	1	1	1	1	0	0	0	0	0
terrein	0	0	0	0	0	0	0	0	0	1	1	1	1
trottoir	0	0	0	0	0	0	0	0	0	0	0	1	1
vastbijgebouw	0	0	0	0	0	0	0	0	0	0	0	0	1

Table A-9. The multiplication of matrices $\mathbf{R}^T$, **T**, and **S**.

If we translate **Table A-9** into pairs of labels we get:

{ (*berm*, *3103*), (*berm*, *3203*), (*berm*, *3303*), (*berm*, *3533*), (*berm*, *3603*), (*bermsloot*, *5023*), (*bermsloot*, *5203*), (*bermsloot*, *5213*), (*bermsloot*, *5263*), (*bloemenperk*, *5213*), (*bloemenperk*, *5263*), (*fietspad*, *3603*), (*hoofdgebouw*, *1000*), (*losbijgebouw*, *1000*), (*losbijgebouw*, *1050*), (*losbijgebouw*, *1073*), (*losbijgebouw*, *5023*), (*losbijgebouw*, *5203*), (*losbijgebouw*, *5213*), (*losbijgebouw*, *5263*), (*parkeerstrook*, *5213*), (*parkeerstrook*, *5263*), (*rijbaan*, *3103*), (*rijbaan*, *3203*), (*rijbaan*, *3303*), (*rijbaan*, *3533*), (*spoorbaan*, *4000*), (*terrein*, *5023*), (*terrein*, *5203*), (*terrein*, *5213*), (*terrein*, *5263*), (*trottoir*, *5213*), (*trottoir*, *5263*), (*vastbijgebouw*, *5263*) }

as was mentioned in Section 5.1.

Note that the partition of **Table A-9** is similar to the partition of the end result in **Table A-4**, as can be verified by evaluating every single part.

A2 Types of Semantic Similarity

In order to compute the type of semantic similarity of an ordered pair of $\mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{S}$, we divide **R** into **R***rec* + **R***rcc*.

R*rec* contains, according to its definition in **(34)** in Section 3.5, all GBKN labels with references to

- the taxonomy subgraph of the RM, and
- the partonomy subgraph of the RM, where it concerns *component* classes.

R*rec* is indicated with a light gray shade in **Table A-5**.

R*rcc* contains, according to its definition in **(35)** in Section 3.5, all GBKN labels with references to
- the partonomy subgraph of the RM, where it concerns *composite* classes.

In a similar way **S** is split into **S***rec* + **S***rcc*. See **(37)** and **(38)** in Section 3.5.

If we compute all *equivalent* ordered pairs with $\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec$ we get:

TOP10 label→ / *GBKN label* ↓	*4000*	*10050*	*10073*	*10000*	*31003*	*32003*	*33003*	*35333*	*36033*	*50233*	*52003*	*52133*	*52633*
losbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-
spoorbaan	1	-	-	-	-	-	-	-	-	-	-	-	-
berm	-	-	-	-	-	-	-	-	-	-	-	-	-
bermsloot	-	-	-	-	-	-	-	-	-	-	-	-	-
bloemenperk	-	-	-	-	-	-	-	-	-	-	-	-	-
fietspad	-	-	-	-	-	-	-	-	-	-	-	-	-
hoofdgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-
losbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-
parkeerstrook	-	-	-	-	-	-	-	-	-	-	-	-	-
rijbaan	-	-	-	-	-	-	-	-	-	-	-	-	-
terrein	-	-	-	-	-	-	-	-	-	-	-	-	-
trottoir	-	-	-	-	-	-	-	-	-	-	-	-	-
vastbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-

Table A-10. The multiplication of $\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec$ (- denotes 0).

or, translated into a pair of labels:

{ (*spoorbaan*, *4000*) }.

Next, all *related* ordered pairs are computed with $\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec$:

TOP10 label → / *GBKN label ↓*	*4000*	*1050*	*1073*	*1000*	*3103*	*3203*	*3303*	*3533*	*3603*	*5023*	*5203*	*5213*	*5263*
losbijgebouw	-	1	1	-	-	-	-	-	-	-	-	-	-
spoorbaan	-	-	-	-	-	-	-	-	-	-	-	-	-
berm	-	-	-	-	-	-	-	-	-	-	-	-	-
bermsloot	-	-	-	-	-	-	-	-	-	-	-	-	-
bloemenperk	-	-	-	-	-	-	-	-	-	-	-	-	-
fietspad	-	-	-	-	-	-	-	-	-	-	-	-	-
hoofdgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-
losbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-
parkeerstrook	-	-	-	-	-	-	-	-	-	-	-	-	-
rijbaan	-	-	-	-	-	-	-	-	-	-	-	-	-
terrein	-	-	-	-	-	-	-	-	-	-	-	-	-
trottoir	-	-	-	-	-	-	-	-	-	-	-	-	-
vastbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-

Table A-11. The multiplication of $\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec$ (- denotes 0).

or, translated into pairs of labels: { (*losbijgebouw*, *1050*), (*losbijgebouw*, *1073*) }.

Finally, all *relevant* ordered pairs are computed with
$\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$:

TOP10 label→ / *GBKN label ↓*	*4000*	*1050*	*1073*	*1000*	*3103*	*3203*	*3303*	*3533*	*3603*	*5023*	*5203*	*5213*	*5263*
losbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	-
spoorbaan	-	-	-	-	-	-	-	-	-	-	-	-	-
berm	-	-	-	-	1	1	1	1	1	-	-	-	-
bermsloot	-	-	-	-	-	-	-	-	-	1	1	1	1
bloemenperk	-	-	-	-	-	-	-	-	-	-	-	1	1
fietspad	-	-	-	-	-	-	-	-	1	-	-	-	-
hoofdgebouw	-	-	-	1	-	-	-	-	-	-	-	-	-
losbijgebouw	-	-	-	1	-	-	-	-	-	1	1	1	1
parkeerstrook	-	-	-	-	-	-	-	-	-	-	-	1	1
rijbaan	-	-	-	-	1	1	1	1	-	-	-	-	-
terrein	-	-	-	-	-	-	-	-	-	1	1	1	1
trottoir	-	-	-	-	-	-	-	-	-	-	-	1	1
vastbijgebouw	-	-	-	-	-	-	-	-	-	-	-	-	1

Table A-12. The multiplication of
$\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$ (- denotes 0).

or, translated into pairs of labels:

{ (*berm*, *3103*), (*berm*, *3203*), (*berm*, *3303*), (*berm*, *3533*), (*berm*, *3603*), (*bermsloot*, *5023*), (*bermsloot*, *5203*), (*bermsloot*, *5213*), (*bermsloot*, *5263*), (*bloemenperk*, *5213*), (*bloemenperk*, *5263*), (*fietspad*, *3603*), (*hoofdgebouw*, *1000*), (*losbijgebouw*, *1000*), (*losbijgebouw*, *5023*), (*losbijgebouw*, *5203*), (*losbijgebouw*, *5213*), (*losbijgebouw*, *5263*), (*parkeerstrook*, *5213*), (*parkeerstrook*, *5263*), (*rijbaan*, *3103*), (*rijbaan*, *3203*), (*rijbaan*, *3303*), (*rijbaan*, *3533*), (*terrein*, *5023*), (*terrein*, *5203*), (*terrein*, *5213*), (*terrein*, *5263*), (*trottoir*, *5213*), (*trottoir*, *5263*), (*vastbijgebouw*, *5263*) }.

This result can be split into three partitions:

1. the first partition concerns GBKN component classes as constituents of TOP10vector composite classes: $\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$. See the light gray shading in **Table A-12**. Or, translated into pairs of labels:

   { (*berm*, *3103*), (*berm*, *3203*), (*berm*, *3303*), (*berm*, *3533*), (*berm*, *3603*), (*bermsloot*, *5023*), (*bermsloot*, *5203*), (*bermsloot*, *5213*), (*bermsloot*, *5263*), (*bloemenperk*, *5213*), (*bloemenperk*, *5263*), (*fietspad*, *3603*), (*hoofdgebouw*, *1000*), (*losbijgebouw*, *1000*), (*losbijgebouw*, *5023*), (*losbijgebouw*, *5203*), (*losbijgebouw*, *5213*), (*losbijgebouw*, *5263*), (*parkeerstrook*, *5213*), (*parkeerstrook*, *5263*), (*rijbaan*, *3103*), (*rijbaan*, *3203*), (*rijbaan*, *3303*), (*rijbaan*, *3533*), (*trottoir*, *5213*), (*trottoir*, *5263*), (*vastbijgebouw*, *5263*) }

2. the second partition concerns GBKN composite classes composed of TOP10-vector component classes: $\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec$. There are no such classes.

3. the third partition concerns GBKN composite classes composed of TOP10vector composite classes (or, vice versa). See the dark gray shading in **Table A-12**. Or, translated into labels:

   { (*terrein*, *5023*), (*terrein*, *5203*), (*terrein*, *5213*), (*terrein*, *5263*) }.

Appendix B

Basic structures for the reference model (taxonomy and partonomy) were introduced in Chapter 2, together with a basic semantic relationship between a reference model class and an application class (‘*refers_to*’). Then, based on this *refers_to* relationship, three semantic relationships between classes of different data sets were defined: classes with a semantic equivalent, related, or relevant relationship. It is proved in Section B1 that each and every pair of classes (1) belongs to one of these relationships, or (2) are incompatible classes.

Reference model matrix **T** was constructed in four steps in Section 3.4.3.1. It was stated that every non-diagonal entry of **T** is the number of paths, from one reference model class to another reference model class. This claim is explained in Section B2.

The set of semantically similar ordered pairs of labels was expressed in **Theorem 1**. This set can be broken down into subsets, depending on type of semantic similarity (**Theorem 2** up to **Theorem 9**). All theorems are proved in Section B3.

B1 The Complete Set of Relationships between Object Classes

Three semantic relationships — ‘equivalent’, ‘related’, and ‘relevant’ — were defined in Section 2.5.4. These relationships represent the semantics between classes of different data sets. Each and every pair of classes belongs to one of these relationships, given the definitions and constructs for structuring a reference model (RM).

Essentially, basic RM constructs can be summarized in one single diagram. In **Fig. B-1** we see:

- the partonomy RM subgraph: composite classes *A* and *B*, and component classes *E*, *F*, and *G*, where class *F* is shared by both classes *A* and *B*, and
- the taxonomy RM subgraph: superclasses *C* and *D*, with *C* having no subclasses, and *D* having classes *H* and *I* as subclasses.

It is supposed that integrating *two* data sets requires a RM with *two* abstraction levels (this restriction was made for the partonomy part of the RM in Section 2.5.2). Integrating a *third* data set might require a *third* level of abstraction, but in that case it is supposed that object classes from a certain data set have references to (and from) not more than two distinct RM levels. Given the RM diagram in **Fig. B-1**, and object classes from two data sets B and C, then the *proof* that three relationships will comprise the complete set, is given by enumerating each and every possibility. Suppose we have an object class *b*, and an object class *c*, from data set B and C, respectively. If RM class *A* refers to class *b*, then theoretically class *c* has a reference to one out of nine RM classes {*A*, *B*, *C*, *D*, *E*, *F*, G, *H*, *I*} in **Fig. B-1**.

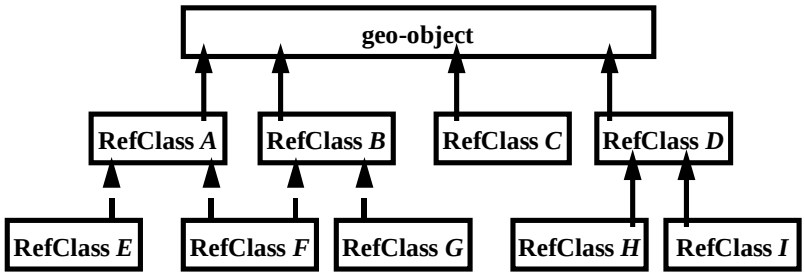


Fig. B-1. The basic RM constructs summarized in one single diagram.

		Class *c* has a reference from RM class:								
		A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
Class *b* from data set B has a reference from RM class:	*A*	*np1*	*rele* (**Def. 4-2**) or *np2**	*inc*	*inc*	*rele* (**Def. 4-1**)	*rele* (**Def. 4-1**)	*inc** or *np3*	*inc*	*inc*
	B	*rele* or *np2**	*np1*	*inc*	*inc*	*inc** or *np3*	*rele*	*rele*	*inc*	*inc*
	C	*inc*	*inc*	*equi* (**Def. 2**)	*inc*	*inc*	*inc*	*inc*	*inc*	*inc*
	D	*inc*	*inc*	*inc*	*np1*	*inc*	*inc*	*inc*	*rela* (**Def. 3**)	*rela* (**Def. 3**)
	E	*rele*	*inc** or *np3*	*inc*	*inc*	*np1*	*np4*	*inc*	*inc*	*inc*
	F	*rele*	*rele*	*inc*	*inc*	*np4*	*np1*	*np5*	*inc*	*inc*
	G	*inc** or *np3*	*rele*	*inc*	*inc*	*inc*	*np5*	*np1*	*inc*	*inc*
	H	*inc*	*inc*	*inc*	*rela*	*inc*	*inc*	*inc*	*np1*	*np6*
	I	*inc*	*inc*	*inc*	*rela*	*inc*	*inc*	*inc*	*np6*	*np1*

Table B-2. The (symmetric) outcome of every combination between class *b* and class *c* (* denotes ...*if there is reference from F*).

The same holds, if RM class *B* refers to class *b*, then theoretically class *c* has again a reference to one out of nine RM classes {*A*, *B*, *C*, *D*, *E*, *F*, G, *H*, *I*}, and so on.

See **Table B-2**, where all 9 × 9 possibilities are summarized. Each cell contains the outcome of its combination. This outcome is either equivalent (*equi*), related (*rela*), relevant (*rele*), or incompatible (*inc*).

However, a combination might be 'not possible' (*np*), if it's remembered that the RM models interconnectedness, or abstractions *between* data sets, not *within* data sets. A data set to be integrated might have some hierarchy, but this hierarchy does not concern the structure of the RM. Therefore, where it concerns the RM, it is supposed that object classes from the same data set are not modeled as RM subclasses, or superclasses of each other, or as RM component classes, or composite classes of each other.

With this basic modeling assumption, the explanation for 'not possible' (*np #*) in **Table B-2** is as follows:

- *np1*. If both *b* and *c* have references to the same RM class, then any subclass, or component class for that RM class does not exist (the basic modeling assumption). Therefore, combinations (*A*, *A*) to (*I*, *I*) are not possible, except (*C*, *C*), the equivalent relationship (**Def. 2**).
- *np2*. If *refers_to*(*A*, *b*), and *refers_to*(*B*, *c*), with no *refers_to*(*F*, _), then *b* and *c* are relevant to each other (**Def. 4-2**). With a *refers_to*(*F*, _), then *b* and *c* are supposed to be from the same data set, therefore this combination is not possible.
- *np3*. If *refers_to*(*A*, *b*), and *refers_to*(*G*, *c*) with a *refers_to*(*F*, _), then *b* and *c* are incompatible. With no *refers_to*(*F*, _), then if *A* and *B* refer to different data sets, then *b* and *c* belong to the same data set, therefore this combination is not possible.
- *np4*. If *refers_to*(*E*, *b*) and *refers_to*(*F*, *c*), then *b* is supposed to belong to the same data set as *c*, therefore this combination is not possible.
- *np5*. If *refers_to*(*F*, *b*) and *refers_to*(*G*, *c*), then *b* is supposed to belong to the same data set as *c*, therefore this combination is not possible.
- *np6*. If *refers_to*(*H*, *b*), and *refers_to*(*I*, *c*), then *b* is supposed to belong to the same data set as *c*, therefore this combination is not possible.

The outcome of this enumeration of combinations, together with the previous assumptions and considerations, makes us conclude that there are only four types of semantic relationships: equivalent, related, relevant, or incompatible.

B2 Reference Model Matrix T and Connectivity

Matrix **T**, representing all relations *T*[*L*] between reference model levels, is stepwise constructed in Section 3.4.3.1. It was stated that every non-diagonal entry of **T**

is the number of paths, from one reference model class to another reference model class. To explain this claim, the concept of a *connectivity* matrix *C* is introduced.
Let *H* be a directed graph with nodes $a_1, a_2, \ldots, a_m$. The connectivity matrix of *H* is the $n \times n$ matrix $\mathbf{C}_H = (c_{ij})$ where c_{ij} = the number of edges beginning at a_i and ending at a_j. Entries of $\mathbf{C}_H$ will be zeros and ones. Now, a theorem from graph theory [1] is that the *ij* entry of matrix $\mathbf{C}_H^m$ (= $\mathbf{C}_H$ to the power *m*) gives the number of paths of length *m* from node a_i to a_j.
Experimentally, the following expression holds for **T**:

$$\mathbf{T} = \{\mathbf{C}_H + \mathbf{C}_H^2 + \cdots + \mathbf{C}_H^m\} + \{(\mathbf{C}_H)^T + (\mathbf{C}_H^2)^T + \cdots + (\mathbf{C}_H^m)^T\} + \mathbf{I}^n$$

with *m* + 1 the number of levels of *H*. Any path in *H* must have length *m* or less. Thus, the first right part of this expression can be understood as the summation of paths of length 1, 2, ..., *m*. This is a lower diagonal matrix. To this lower diagonal matrix, its transpose is added (an upper diagonal matrix, the second right part of the expression), together with a $n \times n$ identity matrix (the third right part of the expression). Experimentally this is equal to **T**. Therefore, matrix **T** is symmetric, where every entry t_{ij} $(= t_{ji})$, with $i \neq j$, is the number of paths from *i* to *j*, or *j* to *i*.

B3 Semantically Similar Labels as Ordered Pairs

Theorem 1. The set of ordered pairs of semantically similar labels (*b*, *c*) between data sets B and C, with label sets *B* and *C* is given by:

$$\bigcup_{L=1}^{m} \{ \bigcup_{K=1}^{m} (R[L]^{-1} \circ T \circ S[K]) \}$$

with:

- $\bigcup$ the union operator
- *R*[*L*] and *S*[*K*] as defined in **(17)** and **(18)**
- *L*, *K* ∈ {1, ... , *m*}, with *m* + 1 the number of levels of directed graph *H*
- $\begin{cases} T = \Delta_T \quad \text{(the identity relation in } T\text{)}, & (L = K) \\ T = T[L+1]^{-1} \circ T[L+2]^{-1} \circ \cdots \circ T[K]^{-1}, & (L < K) \text{, and} \\ T = T[L] \circ T[L-1] \circ \cdots \circ T[K+1], & (L > K) \end{cases}$
- *T*[*L*] the relation between different levels in set *A* as defined in **(16)**.

[1] S. Lipschutz. Discrete mathematics, Schaum Outline Series. New York: McGraw-Hill, 249 pages, 1976.

Proof of **Theorem 1**.

1. If $L = K$ then $T = \Delta_T$, and $R[L]^{-1} \circ T \circ S[K] = R[L]^{-1} \circ S[K] = R[L]^{-1} \circ S[L]$. Thus, with $R[L]$ the relation between reference model labels of A_L and class labels of B_L, and with $S[L]$ the relation between reference model labels of A_L and class labels of C_L, the union $\bigcup R[L]^{-1} \circ S[L]$ encompasses all semantically similar ordered pairs (b, c) between B_L and C_K for each and every level, where $L = K$, with $L, K \in \{1, \ldots, m\}$.

2. If $L < K$ then $T = T[L+1]^{-1} \circ T[L+2]^{-1} \circ \cdots \circ T[K]^{-1}$. This series of propagations is illustrated in **Fig. B-2**.

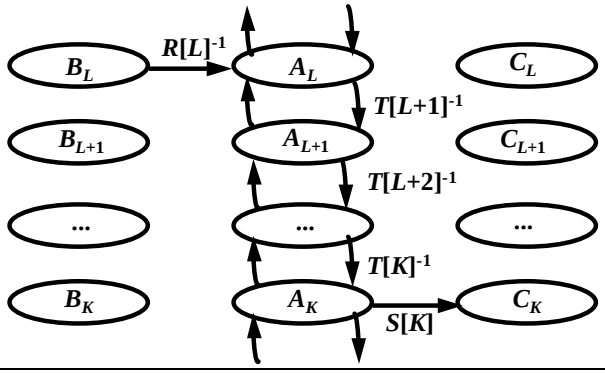


Fig. B-2. The downward propagation between level L and level K in the reference model.

Thus, with $R[L]$ the relation between reference model labels of A_L and class labels of B_L, and with $S[K]$ the relation between reference model labels of A_K and class labels of C_K, union $\bigcup R[L]^{-1} \circ T[L+1]^{-1} \circ T[L+2]^{-1} \circ \cdots \circ T[K]^{-1} \circ S[K]$ encompasses all semantically similar ordered pairs (b, c) between B_L and C_K for each and every combination of levels where $L < K$, with $L, K \in \{1, ..., m\}$.

3. If $L > K$ then $T = T[L] \circ T[L-1] \circ \cdots \circ T[K+1]$. This series of propagations is illustrated in **Fig. B-3**.

Thus, with $R[L]$ the relation between reference model labels of A_L and class labels of B_L, and with $S[K]$ the relation between reference model labels of A_K and class labels of C_K, union $\bigcup R[L]^{-1} \circ T[L] \circ T[L-1] \circ \cdots \circ T[K+1] \circ S[K]$ encompasses all semantically similar ordered pairs (b, c) between B_L and C_K for each and every combination of levels where $L > K$, with $L, K \in \{1, ..., m\}$.

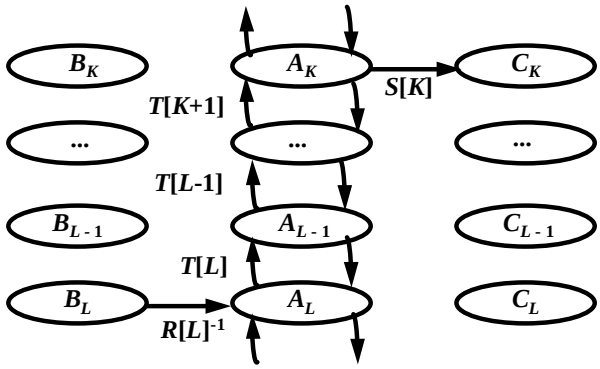


Fig. B-3. The upward propagation between level *L* and level *K* in the reference model

Theorem 2. *Semantic equivalent* ordered pairs of labels (*b*, *c*) are similar to:

$$\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec = \mathbf{R}rec^T \cdot \mathbf{S}rec$$

Proof of **Theorem 2**.

R*rec* and **S***rec* are matrix representations of relations *Rrec* and *Srec*, respectively:

– *Rrec* is by definition the set of ordered pairs (*a*, *b*) between RM subclasses, or RM component classes of label set *A* and label set *B*, and

– *Srec* is by definition the set of ordered pairs (*a*, *c*) between RM subclasses, or RM component classes of label set *A* and label set *C*.

References from labels *b* and *c* to the *same RM component* class are not possible, the basic modeling assumption in Section B1 (*np*1 in **Table B-2**). Therefore, where it concerns **R***rec* and **S***rec*, only references apply to RM *subclasses*, and because **T** = **I** there is no upward, or downward propagation, and *b* and *c* must have a reference from the same subclass. Thus, $\mathbf{R}rec^T \cdot \mathbf{I} \cdot \mathbf{S}rec$ is similar to all equivalent ordered pairs (*b*, *c*).

Theorem 3. *Semantic related* ordered pairs of labels (*b*, *c*) are similar to:

$$\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec$$

Proof of **Theorem 3**, *including* **Theorem 5** *and* **Theorem 6**.

R*rec* and **S***rec* are matrix representations of relations *Rrec* and *Srec*, respectively:

– *Rrec* is by definition the set of ordered pairs (*a*, *b*) between RM subclasses, or RM component classes of label set *A* and label set *B*, and

– *Srec* is by definition the set of ordered pairs (*a*, *c*) between RM subclasses, or RM component classes of label set *A* and label set *C*.

References from labels *b* and *c* to the *same* RM *component* class are not possible, the basic modeling assumption in Section B1 (*np*1 in **Table B-2**). Therefore, where it concerns both **R***rec* and **S***rec*, only references apply to RM *subclasses*. Thus, with **T***prop* representing upward and downward propagation between RM subclasses, and excluding equivalent classes (**T***prop* = **T** − **I**), $\mathbf{R}rec^T \cdot \mathbf{T}prop \cdot \mathbf{S}rec$ is similar to all related ordered pairs (*b*, *c*).

In addition, with $\mathbf{T}prop = \mathbf{T}propsper + \mathbf{T}propsub$, respectively its upper-diagonal matrix and lower-diagonal matrix, then

– $\mathbf{R}rec^T \cdot \mathbf{T}propsper \cdot \mathbf{S}rec$ is similar to all related ordered pairs, where *b* is a super-class of *c*, because **T***propsper* links higher level taxonomy classes with lower level taxonomy classes (**Theorem 5**). Conversely,

– $\mathbf{R}rec^T \cdot \mathbf{T}propsub \cdot \mathbf{S}rec$ is similar to all related ordered pairs where *b* is a subclass of *c*, because **T***propsub* links lower level taxonomy classes with higher level taxonomy classes (**Theorem 6**).

Theorem 4. *Semantic relevant* ordered pairs of labels (*b*, *c*) are similar to:

$$\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec + \mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$$

Proof of **Theorem 4**, *including* **Theorem 7**, **Theorem 8**, *and* **Theorem 9**

R*rec* and **S***rec* are matrix representations of relations *Rrec* and *Srec*, respectively:

– *Rrec* is by definition the set of ordered pairs (*a*, *b*) between RM subclasses, or RM component classes of label set *A* and label set *B*, and

– *Srec* is by definition the set of ordered pairs (*a*, *c*) between RM subclasses, or RM component classes of label set *A* and label set *C*.

R*rcc* and **S***rcc* are matrix representations of relations *Rrcc* and *Srcc*, respectively:

– *Rrcc* is by definition the set of ordered pairs (*a*, *b*) between RM composite classes of label set *A*, and label set *B*, and

– *Srec* is by definition the set of ordered pairs (*a*, *c*) between RM composite classes of label set *A*, and label set *C*.

Therefore, where it concerns combinations of **R***rec*, **S***rec*, **R***rcc*, and **S***rcc*, only references apply to RM *component classes* and *composite* classes. Then:

– $\mathbf{R}rec^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$ represents ordered pairs (*b*, *c*) between component classes and composite classes (**Theorem 7**);

– $\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rec$ represents ordered pairs (b, c) between composite classes, and component classes (**Theorem 8**);

– $\mathbf{R}rcc^T \cdot \mathbf{T} \cdot \mathbf{S}rcc$ represents ordered pairs (b, c) between composite classes, and composite classes (**Theorem 9**).

Appendix C

Propositional calculus is applied to **building** surveying rules in Section 4.3.1. *Statements* will be denoted by *words* (a sequence of letters), like *address, urban, area9*, etc, meaning: 'it is true that this class has address', 'is situated in urban area', 'with area ≥ 9m^2', etc. Statements can be composed into compound statements with connectives: & (logical *and*), or | (logical *or*). Any statement can be negated, symbolically: ~urban, ~area9, etc, meaning: 'it is not true that this class is situated in urban area', 'has area size ≥ 9m^2', etc (where the negation of 'urban' is 'rural', of 'area size ≥ 9m^2' is 'area size < 9m^2', etc).

C1 Building Surveying Rules

Given GBKN surveying rules for **buildings** in Section 4.3.1, then expression

```
hoofdgebouw = address
```

states that **building** with address ('mainbuilding'), is acquired as (GBKN class) *hoofdgebouw*.

Similarly, expression

```
vastbijgebouw = ~address & ~free
```

states that an adjacent **building**, without address ('adjacent annex'), is acquired as (GBKN class) *vastbijgebouw*.

Furthermore, expression

```
losbijgebouw =
(~address & free & urban) |
(~address & free & ~urban & area20)
```

states that a free standing **building** without address ('free standing annex'), situated in urban area, or situated in rural area, with area ≥ 20m^2, is acquired as (GBKN class) *losbijgebouw*.

Given TOP10vector surveying rules for **buildings** in Section 4.3.1, then expression

```
t10xx =
((address | (~address & free)) & urban & access &
area9) |
((address | (~address & free)) & urban & ~access &
area50) |
((address | (~address & free)) & ~urban & area9)
```

states that 'mainbuilding', or 'free standing annex', is acquired as (TOP10vector class) *10xx*, if it is situated in urban area, accessible, with area ≥ 9m^2, or if it is

situated in urban area, not accessible, with area ≥ 50m^2, or if it is situated in rural area, with area ≥ 9m^2.

C2 Consistency of Building Candidates

To know real-world situations, implied by a simple **building** candidate of type {*hoofdgebouw*, *1000*}, we apply the algebra of propositions to conjunction:

hoofdgebouw ∧ *t10xx*

i.e. to both *intensions* of *hoofdgebouw* and *10xx*, with the help of *Mathematica* function LogicalExpand: [2]

```
LogicalExpand[hoofdgebouw & t10xx]
```

resulting in three compound statements:

```
access & address & area9 & urban |
address & area50 & urban & ~access |
address & area9 & ~urban
```

which is equivalent to 'mainbuilding', situated in:

- urban area, accessible, with area ≥ 9m^2, or
- urban area, not accessible, with area ≥ 50m^2, or
- rural area, with area ≥ 9m^2

as mentioned in Section 5.3 (first item).

Similarly, real-world situations implied by simple candidates of type {*losbijgebouw*, *1000*}, {*losbijgebouw*, *1050*}, and {*losbijgebouw*, *1073*} is the expansion of conjunction *losbijgebouw* ∧ *t10xx*, again *intensions* of *losbijgebouw*, and *1000*, *1050*, or *1073*, respectively:

```
LogicalExpand[losbijgebouw & t10xx]
```

resulting in three compound statements:

```
access & area9 & free & urban & ~address |
area50 & free & urban & ~access & ~address |
area20 & area9 & free & ~address & ~urban
```

which is — observing that the last compound statement can only be true if area ≥ 20m^2 — equivalent to a 'free standing annex', situated in:

- urban area, accessible, with area ≥ 9m^2, or
- urban area, not accessible, with area ≥ 50m^2, or
- rural area, with area ≥ 20m^2

as mentioned in Section 5.3 (second item).

---

[2] S. Wolfram. The Mathematica Book, 3rd ed. Champaign, IL, USA: Wolfram Media, 1403 pages, 1996.

C3 Buildings in Land Candidates

To know real-world situations, implied by a GBKN *losbijgebouw* in a **land** candidate, conjunction *losbijgebouw* ∧ *~t10xx* is expanded:

```
LogicalExpand[losbijgebouw & ~t10xx]
```

resulting in a series of compound statements:

```
access & free & urban & ~address & ~area9 |
area20 & free & ~address & ~area9 & ~urban |
free & urban & ~access & ~address & ~area50 |
free & urban & ~address & ~area50 & ~area9 |
access & area20 & free & ~address & ~area9 & ~urban |
area20 & free & ~access & ~address & ~area9 & ~urban |
area20 & free & ~address & ~area50 & ~area9 & ~urban |
free & urban & ~access & ~address & ~area50 & ~area9 |
area20 & free & ~access & ~address & ~area50 & ~area9 &
~urban
```

Removing contradictions (*e.g.* area ≥ 20 m^2 & area < $9m^2$) and stricter statements implied by more general statements, we get

```
free & urban & ~access & ~address & ~area50 |
free & urban & ~address & ~area9
```

which is equivalent to a ‘free standing annex’ situated in:

- urban area, not accessible, with area < $50m^2$, or
- urban area, with area < $9m^2$

as mentioned in Section 5.5.

C4 Building Singletons

To know real-world situations implied by a GBKN **building** singleton, we expand compound statement (hoofdgebouw ∨ vastbijgebouw ∨ losbijgebouw) ∧ ~t10xx:

```
LogicalExpand[(hoofdgebouw | vastbijgebouw |
losbijgebouw) & ~t10xx]
```

After removing contradictions, this results in five compound statements,:

```
address & ~area9 |
~address & ~free |
address & urban & ~access & ~area50 |
free & urban & ~address & ~area9 |
free & urban & ~access & ~address & ~area50
```

which is equivalent to:

- ‘mainbuilding’, with area < $9m^2$, or
- ‘adjacent annex’, or
- ‘mainbuilding’, situated in urban area, not accessible, with area < $50m^2$, or

- a ‘free standing annex’, in urban area, not accessible, with area < 50m^2, or
- a ‘free standing annex’, situated in urban area, with area < 9m^2

as mentioned in Section 5.6.1.

Finally, to know real-world situations implied by a TOP10vector **building** singleton, we expand compound statement ~(*hoofdgebouw* ∨ *vastbijgebouw* ∨ *losbijgebouw*) ∧ *t10xx*:

```
LogicalExpand[~(hoofdgebouw | vastbijgebouw |
losbijgebouw) & t10xx]
```

resulting in:

```
area9 & free & ~address & ~area20 & ~urban
```

which is equivalent to:

- ‘free standing annex’, in rural area, with area between 9m^2 and 20m^2

as mentioned in Section 5.6.2.

Curriculum Vitae

Harry Uitermark was born on September 24, 1946, in Haarlem, The Netherlands, where he grew up and attended primary school. Later he moved to Venlo where he completed secondary school in 1963 (HBS-B, St. Thomascollege) [3].

Initially he pursued a military career at the Royal Netherlands Naval College in Den Helder, from 1963 up to 1968, but changed to a civil career by studying geodesy at Delft University of Technology, obtaining an M.Sc. in 1973. It was in Delft where he met his second promotor, Martien Molenaar.

From 1973 up to 1975 he was part-time lecturer at the Department of Geodesy (with Theo Bogaerts as supervisor), studied psychology at Leiden University, and was part-time mathematics teacher (Ichthus college, Rotterdam).

From 1976 up to 1979 he worked at BGC, Consultants in Urban Planning, in Deventer, as a researcher in traffic studies. It was here where he met his first promotor, Koos Mars.

From 1979 up to 1984 he was an independent consultant in geodesy and urban planning, a member of ONRI, the Dutch Association of Consulting Engineers.

In 1984 he joined the Netherlands Kadaster (Cadastre and Public Registers Agency), first in Rotterdam as a project manager (with Joost Marissen as supervisor). Here he got the opportunity to study Information Management at Tias Business School (KUB, Tilburg), in co-operation with Washington University (St. Louis, USA), obtaining an M.Sc. in 1990.

Later he went to the Kadaster corporate staff in Apeldoorn, where he became a GIS-consultant, working with Hiddo Velsink, Jan Denekamp, Jaap van der Veen, and Martin Salzmann, successively.

It was in Apeldoorn where he started in 1995 his PhD research, stimulated by his involvement in research projects as update propagation between GBKN and TOP10vector [4], and the object-structured GBKN [5].

---

[3] For a reflection on his first experiences with maps, see H. T. Uitermark. “A la recherche de la pierre perdue”, *Geodesia*, Vol. 41, No. 5, pp. 235-236, 1999.

[4] TDN and Kadaster (1995). “GBKN-mutaties en TOP10vector. Onderzoek naar de uitwisseling van GBKN-mutaties met TOP10vector. Deelonderzoek: GBKN-inhoud versus TOP10vector-inhoud”. Topografische Dienst/Dienst van het Kadaster en de Openbare Registers, Emmen/Apeldoorn.

[5] Van der Veen, J.B. and H.T. Uitermark (1995). “Naar een objectgerichte GBKN?! Van basiskaart tot basisbestand via het objectbegrip”. *Geodesia*, Vol. 37, No. 2, pp. 83-88.