Geographical Information Management and Applications

UNIFORMITY OF TOPOGRAPHICAL KEY REGISTRATIONS

The derivation of data specifications in order to automatically generalize the BGT into a uniform BGT product and into a midscale data product

GIMA Master Thesis

Author
P.T.A. ten Rouwelaar

Universiteit Utrecht

TU Delft

University of Twente.

Wageningen UR

kadaster

Uniformity of topographical key registrations

The derivation of data specifications in order to automatically generalize the BGT into a uniform BGT product and into a midscale data product

<u>Master thesis</u>

Geographical Information Management and Applications (GIMA)

July 12th, 2015

Author: P.T.A. ten Rouwelaar BSc

Supervisor: Prof. dr. J. Stoter (TU Delft)

Co-supervisor: Prof. dr. ir. A.K. Bregt (WUR)

PREFACE

In this Master thesis, data specifications were derived from the BGT to create a more uniform BGT product and a midscale data product, which both can be integrated in the Dutch system of national key registrations. During an earlier phase of my Masters, I executed an assignment in the area of automatic generalization, which I found very interesting. Therefore, I decided to proceed with this topic for my Master thesis to learn more about the process of automatic generalization and the implementation of data specifications in the Dutch system of key registrations. During the development of this thesis, I got the opportunity to perform the methodology as an internship at the Dutch Cadaster. Looking back on this period, I can honestly say that it was a challenging period with the usual ups and downs when executing a Master thesis. In the end, this thesis created beautiful memories, connections, and experiences in the field of automatic generalization.

In this preface, I would like to take the opportunity to thank a few people for their support during my thesis. Firstly, I would like to thank Jantien Stoter. As my supervisor, she helped me during my thesis by reading and commenting all versions of my thesis, and she supported me when I needed it. She also gave me the opportunity to execute my study at the Dutch Cadaster in Zwolle. Secondly, I want to thank my co-supervisor Arnold Bregt for reading my thesis and giving me useful suggestions. And finally, I want to thank the employees of the department GMA at the Dutch Cadaster. In particular, I want to mention Vincent van Altena, Peter Lentjes, Marc Post, and Marcel Reuvers, who provided me with additional ideas and insights about the models, and the results I found. I also got the opportunity to cooperate with them in organizing the user consultations, which can be seen as continuation of this study. Finally, I want to thank my family and friends for listening to my stories and complaints during this period of my thesis and for supporting me where they could.

With this thesis, I intend to contribute to the uniformity of the system of key registrations in the Netherlands. I want to give more insight in the topic of automatic generalization and the derivation of data specifications in particular. In addition, I want to provide you with more ideas about the use of the BGT as starting point to automatically generalize the smaller scales in topographical key registrations.

Enjoy reading,

Patricia ten Rouwelaar
July 12th, 2015.

SUMMARY

Due to the implementation of a new topographical key registration, i.e. the ´Basisregistratie Grootschalige Topografie´ (BGT), a division of topographical key registrations in the Netherlands has been developed. At the one hand, the 'Basisregistratie Topografie' (BRT) is nowadays the most important topographical dataset containing all topographical objects in the Netherlands. At the other hand, the BGT, which will be available in 2016, is coming with similar topographical objects, but at larger scale. Following the 'collect once, use many times' principle, it is interesting to study how these two datasets can be integrated. The large scale BGT can then be used as input for the mid- and smaller scales with as main purpose to collect the topographical data at the largest scale and derive the mid- and smaller scales (automatically) from this dataset. In the end, it would be possible to replace the BRT by these derived datasets.

In this study, the data specifications are developed, which enables the step from the BGT towards a midscale BGT dataset. Therefore, four phases are passed: (1) identification of global data specifications with the help of 'reverse engineering' with TOP10NL, the 1:10,000 scale product of the BRT; (2) identification of detailed data specifications with the help of the implementation rules of TOP10NL; (3) testing the data specifications by implementing them into an automatic generalization system; and (4) analysis and evaluation of the developed (test) datasets. As preparation for this midscale dataset, first a uniform BGT has been developed, which aggregates the BGT on its optional data and virtual borders.

The development of the uniform BGT has resulted in a large scale dataset, which is uniform for the Netherlands. This dataset can, after a few adaptations in the model, possibly be added to the Dutch system of key registrations. However, when doing this, it might be interesting to study who is going to be the source holder of the uniform BGT, because the choice of this source holder not only affects how the dataset will be managed, but it can also affect how the dataset will look like.

The data specifications of the midscale BGT have been developed with the help of the implementation rules of TOP10NL in order to create a dataset comparable with TOP10NL. Therefore, 6 different cycles are being used: roads, buildings, water, nature, bridges & tunnels, and other. The implementation of the developed specifications resulted in the knowledge that the midscale BGT and TOP10NL were visually very comparable and that most data specifications were ready to become implemented. However, when looking at the semantics of objects within the midscale BGT and TOP10NL, there are many differences. Sometimes, the attributes or attribute values have the same name, but they have slightly different meanings. Therefore, users should identify which objects are relevant and which information should be maintained in the different topographical key registrations. And sometimes, it was not possible to derive similar objects, because they were not available in the uniform BGT. These objects can partly be derived with the help of additional sources (e.g. the optional part of IMGeo) or can be automatically generalized.

When considering the BGT as input dataset, instead of the commonly used TOP10NL, one topographical key registration can be created within the system of key registrations in the Netherlands. By deriving all mid- and smaller scales automatically from the BGT, the BRT does not have to be collected anymore. Although some enormous adjustments are needed to achieve this, this is also be seen as a chance to refresh the current TOP10NL with an entire new dataset, and to renew the topographical key registrations in the Netherlands. After the integration of the BGT and the BRT, even an extension towards the BAG, derived from the BGT might be considered.

SAMENVATTING

Door de ontwikkeling van een nieuwe topografische basisregistratie, de Basisregistratie Grootschalige Topografie (BGT) is er een tweedeling ontstaan binnen de topografische basisregistraties in Nederland. Enerzijds is er de Basisregistratie Topografie (BRT), welke inzicht geeft in de topografische objecten in heel Nederland op middelgrote en kleine schaal. Anderzijds is er de BGT, welke vanaf 2016 beschikbaar komt en als nieuwe dataset gebruikt kan worden voor ongeveer dezelfde topografische objecten als de BRT, maar dan op een groter schaalniveau. Volgens het principe ‘verzamel eens, gebruik vaker’ is het dan ook interessant om te onderzoeken hoe deze twee basisregistraties geïntegreerd kunnen worden tot één topografische basisregistratie op verschillende schaalniveaus. Alleen de BGT hoeft dan ingewonnen te worden en de middelgrote en kleinere schalen kunnen, al dan niet automatisch, worden afgeleid uit de BGT. Uiteindelijk zou de BRT op deze manier vervangen kunnen worden door deze afgeleide datasets.

Om dit te onderzoeken zijn er in deze studie data specificaties ontwikkeld die de stap van een grootschalige BGT naar een ‘midscale’ dataset mogelijk maken. Daarvoor zijn er vier fasen doorlopen: (1) het identificeren van globale specificaties met behulp van ‘reverse engineering’ met TOP10NL, het 1:10,000 product van de BRT; (2) het identificeren van gedetailleerde data specificaties met behulp van de implementatieregels van TOP10NL; (3) het testen van de data specificaties met behulp van automatische generalisatie; en (4) de analyse en evaluatie van de uiteindelijke (test) datasets. Als voorbereiding op deze midscale dataset is er eerst een uniforme BGT ontwikkeld die de optionele data en de virtuele grenzen aggregeert.

De ontwikkeling van een uniforme BGT resulteerde in een grootschalige dataset die uniform is voor heel Nederland. Deze dataset zou, na een paar aanpassingen in het model, eventueel toegevoegd kunnen worden aan het Nederlandse systeem van basisregistraties. Wanneer dit gebeurt, is het interessant om te kijken naar wie de bronhouder van deze uniforme BGT moet worden. De keuze van de bronhouder heeft namelijk niet alleen effect op hoe de dataset wordt beheerd, maar kan ook effect hebben op hoe de dataset eruit komt te zien.

De data specificaties voor de midscale BGT zijn ontwikkeld met behulp van de implementatieregels van TOP10NL om zo een dataset te creëren die vergelijkbaar is met TOP10NL. Dit is gebeurd in 6 verschillende cyclussen: wegen, gebouwen, water, natuur, bruggen & tunnels en overig. Na het implementeren van de ontwikkelde specificaties bleek dat visueel gezien de datasets erg op elkaar leken en dat de meeste data specificaties klaar waren voor gebruik. Maar wanneer de betekenissen van de objecten worden vergeleken, blijkt dat er wel degelijk nog veel verschillen in zitten. Soms hebben attributen en attribuut waarden dezelfde naam, maar hebben ze een net iets andere betekenis. Daarom zouden gebruikers moeten bekijken welke objecten relevant zijn en welke informatie behouden zou moeten worden in de verschillende topografische basisregistraties. Ook was het soms niet mogelijk om dezelfde objecten te verkrijgen, omdat ze niet in de uniforme BGT aanwezig waren. Deze objecten kunnen deels worden verkregen via andere bronnen (o.a. uit het optionele deel van IMGeo) of kunnen automatisch gegeneraliseerd worden.

Wanneer overwogen wordt om de BGT als input dataset te gebruiken, is het mogelijk om de BRT en de BGT samen te voegen tot één Nederlandse topografische basisregistratie. Door alle midden- en kleinschalige datasets automatisch af te leiden uit de BGT, wordt het onnodig om de BRT te blijven inwinnen. Hoewel er grote aanpassingen nodig zijn om dit voor elkaar te krijgen, kan het ook gezien worden als een kans om de huidige TOP10NL met een geheel nieuwe dataset op te frissen en om de hele BRT in Nederland te vernieuwen. Na de integratie van de BGT en de BRT zou zelfs overwogen kunnen worden om de generalisatie uit te breiden naar de BAG afgeleid uit de BGT.

TABLE OF CONTENTS

The uniform BGT

The midscale BGT

1. INTRODUCTION

Geo-information at different scales is becoming increasingly important to serve different types of information consumers. To collect and manage this geo-information nationwide, topographical key registrations are being used. These topographical key registrations encourage to collect geo-data once and then use them for many different applications (E-overheid, 2014; Kadaster, 2013). With the help of the automatic generalization, the production of topographic data at different scales has become more efficient and flexible at National Mapping Agencies (NMA's) (Stoter et al., 2009b). Automatic generalization can reduce data production costs and can improve data maintenance (Foerster et al., 2010).

The International Cartographic Association (ICA, 1973, p.137) has defined the term 'generalization' as the "selection and simplified representation of detail appropriate to the scale and/or purpose of the map". There are different types of generalization, i.e. 'interactive generalization', 'automatic generalization' and 'semi-automatic generalization'. During the interactive generalization, the cartographer adds his interpretation to the generalization process. The automatic generalization does not involve the cartographers' interpretation, but is based on well-formalized rules, which a generalization system is able to understand. During the semi-automatic generalization, the generalization process is executed to some extend with automatic generalization and in some cases, the experience of the cartographer will be involved (Stoter et al., 2009b). Following the 'collect once, use many times' principle, it is interesting to (automatically) derive small scale datasets out of large scale datasets (Stoter, 2009). Therefore, the generalization process should be understood as a succession of three generalization methods. Firstly, 'object generalization' transforms the real world into an abstract representation. Secondly, 'model generalization' generalizes this abstract representation by simplifying the data structure. And finally, 'cartographic generalization' generalizes the map visually (Cecconi, 2003).

In contrast to object generalization and cartographic generalization, model generalization can be executed using automatic generalization. Therefore, the first step is to develop data specifications to describe on which aspects are needed to execute the generalization. By using a mix of generalization operators in the automatic generalization system, it is possible to automate the generalization of these data specifications. Examples of these operators are: 'simplification', which simplifies objects; 'aggregation', which combines objects both geometrically and thematically; and '(class) selection', which deletes or eliminates objects (Kazemi et al, 2004; Oosterom, 1995; Smaalen & Stoter, 2008).

The development and the implementation of data specifications for automatic generalization is an ongoing process. In the Netherlands, due to traditional aspects, different topographical key registrations have been developed. Historically, the 'Basisregistratie Topografie' (BRT) is an important key registration, because it contains a diverse set of data and is nationwide available. The BRT contains maps at a 1:10,000 scale and smaller (Altena et al., 2013). TOP10NL is BRT's most detailed scale. This is an object-oriented 1:10,000 dataset, which is being used by many GIS and web applications and other flexible visualizations (Kadaster, 2013).

In 2012, a new topographical key registration has been established for large scale maps, called 'Basisregistratie Grootschalige Topografie' (BGT). This is an object-oriented topographical key registration especially meant for large scales (from 1:500 until 1:5,000). The purpose of BGT is to obtain one large scale topographical key registration for the Netherlands in 2016 (Altena et al., 2013; Brink et al., 2013). The main goal of this registration is to get uniformity within the Dutch government. It will improve the usage of key registrations for many different applications, it will ease the communication within the Dutch government and it reduces costs within governmental chains (Brink et al., 2013).

At this time, BGT source holders are preparing the data for implementation in 2016. The content of the BGT is defined in the information model IMGeo, which describes how to organize the geographical data of the BGT. IMGeo contains more information than BGT, because it describes both the key registration objects with minimal requirements and optional IMGeo objects. These objects will be both collected by BGT source holders and submitted to the national portal (Brink et al., 2013; Stoter. 2013). In several cases, the BGT source holders prefer to split objects into objects with so-called 'virtual borders' to be able to add their own attribute values. By the establishment of the BGT, the decision is made that BGT source holders do not need to aggregate these objects. Aggregation in this context means that neighboring objects with the same attribute values can be combined. Due to the complexity of aggregation, this aggregation will not be executed as part of the BGT (Stoter, 2013). However, since users may prefer an aggregated, uniform product, I will investigate how such a product can be established for the Netherlands using automatic generalization. In addition, I will study if some other details should be eliminated from this nationwide large scale product or if some other objects should be created. In the following sections of this study, the term 'uniform BGT' will be used to define this product.

But, the developments are going further. Following the 'collect once, use many times' principle, and to encourage the consistency and efficiency of topographical key registrations, it is interesting to study if the large scale BGT and the midscale BRT can be integrated. Then, only the most detailed BGT needs to be collected and the BRT, i.e. the midscale and small scale data can be automatically derived from this dataset. Currently, the Dutch Cadaster is investigating the possibility to find a solution on how to derive the midscale TOP10NL from the BGT due to differences in the data structure and in the semantics of those data. Although it is possible to derive a midscale dataset from the BGT, this dataset will contain less information than TOP10NL, due to less required objects in the BGT and also the data structure will differ from TOP10NL. Therefore, it is interesting to study if the resulting data can retain the requirements of a key registration and preserve the data which users need, possibly by adding extra information to the process (Altena et al., 2013). In this study, I will develop data specifications, which can be used to automatically derive this midscale product. In the following sections, the midscale product, that in the end can replace TOP10NL, will be defined with the term 'midscale BGT'.

The aim of this study is to derive data specifications ready for implementation in an automatic generalization system for both the uniform BGT and the midscale BGT taking into account the requirements of a key registration. Hereby, the aggregation of the uniform BGT can be seen as first step towards the midscale BGT (Stoter, 2013). In this study, the following objectives are addressed:

- To develop an aggregated, uniform BGT product as input for the midscale BGT;
- To develop data specifications based on the uniform BGT as input for the midscale BGT;
- To test the data specifications by implementing them into an automatic generalization system;
- To analyze and evaluate the resulting midscale BGT.

This study focuses on the derivation of data specifications rather than on the automatic generalization itself. However, the data specifications will be implemented in small BGT datasets to test the developed specifications and to create recommendations about missing specifications after analyzing the differences with the current available TOP10NL.

The data specifications will be formalized using the templates as developed in Stoter et al. (2009a). To ensure the developed products will fit the Dutch system of key registrations, the developed data specifications should follow the key registration requirements. The key registration requirements for the uniform BGT are already adjusted and specified in IMGeo (Brink et al., 2013). The key registration requirements for the midscale BGT are not yet identified. Therefore, the assumption is made that the implementation rules of TOP10NL (Kadaster, 2012) in combination with the key registration requirements of the BGT (Brink et al., 2013) will be sufficient to define the data specifications for the midscale BGT. In both the generalization products, existing guidelines may be used as global guidelines, but deviations from these guidelines are allowed if the experiments show reason for this.

In 2013, the Dutch Cadaster executed an initial investigation on how to generalize the BGT into TOP10NL (Altena et al., 2013) and recently they continued this investigation to derive TOP10NL out of the BGT with the help of users input to define the key registration requirements. To benefit optimally of the knowledge and experience of the Dutch Cadaster, I got the opportunity to execute this study at the topographical department of the Dutch Cadaster in Zwolle. In return, the Dutch Cadaster uses the results and recommendations of this study to execute their further investigations on this topic.

This study exists of two parts, i.e. the development of data specifications for the uniform BGT and the development of data specifications for the midscale BGT. As the preparation of these two generalization products, this study will be justified by giving more insight in what already has been studied in the area of automatic generalization (see Chapter 2) and in the area of topographical key registrations (see Chapter 3). Then, the methodology of this study will be explained (see Chapter 4), the test datasets will be outlined on their usage and their quality (see Chapter 5) and the generic requirements, which apply for both generalization products will be explained (see Chapter 6). The development of the data specifications for both generalization products starts in Chapter 7. Both generalization products follow four phases. In Chapter 7 all phases of the uniform BGT as first step towards the midscale BGT will be outlined. The four phases of the midscale BGT will be outlined in multiple chapters. Firstly, the global data specifications for the midscale BGT will be specified (see Chapter 8); secondly, the detailed data specifications will be derived for the midscale BGT (see Chapter 9); the third phase will show the implementation of those specifications into an automatic generalization system using ArcGIS ModelBuilder (see Chapter 10); and finally, the generalization product will be analyzed and recommendations will be given about the missing requirements in the midscale BGT (see Chapter 11). This study will end with some generic conclusions and recommendations (see Chapter 12).

To describe the different items within a dataset, the terms 'object type', 'object', 'attribute' and 'attribute value' will be used. An 'object' is defined as the digital representation of a spatial item on the map. Objects usually belong to a class of objects, i.e. 'object types', and contain common attributes and attribute values. Attributes are used to describe information about a geographic object in a GIS, usually stored in a table and linked to the object by a unique identifier (e.g. name, length, function, etc.). 'Attribute values' are the information within those attributes (ESRI, 2014b). In this study, these different items will be recognizable by the manner of writing: OBJECTTYPE, *attribute*, and 'attribute value'.

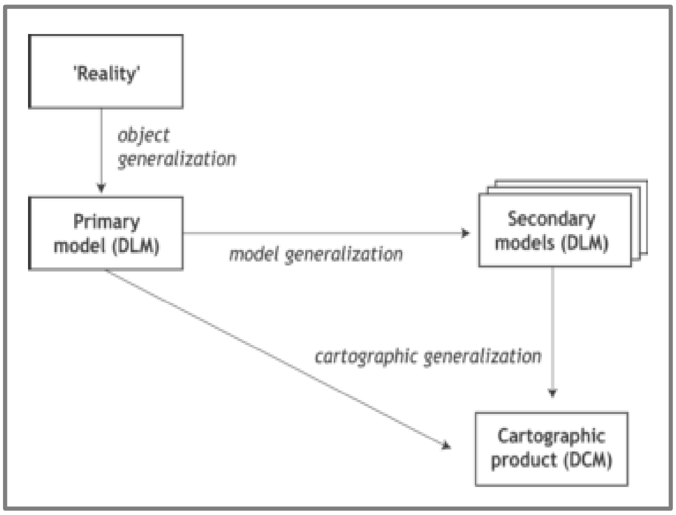
2. THEORETICAL BACKGROUND

Automatic generalization can be defined by "the generation of abstract features from a rich database through computer algorithms rather than a human's judgment" (Kazemi et al., 2004, p.1). In this chapter, more insight will be given in what have already been studied in the area of automatic generalization and how to deal with the different phases in this study.

2.1. THE GENERALIZATION PROCESS

The generalization process can be defined using three different generalization methods: 'object generalization', 'model generalization', and 'cartographic generalization' (see Figure 2.1). The first step of this generalization process models reality into a primary Digital Landscape Model (DLM). In this model, real objects are abstractly represented using object generalization. The second step generalizes the primary DLM into a secondary DLM using model generalization (Cecconi, 2003). Model generalization can be executed following the 'feature reactive approach', which deals with equaling the map structure and identifying the application needs of an object-oriented data model. This step can be automated. Therefore, clear specifications are important to prioritize objects. These specifications need to be translated in appropriate algorithms to execute automatic generalization (Kazemi et al., 2004). In contrast with model generalization, cartographic generalization is more focused on the visual representation of the model and will be used to build the final Digital Cartographic Model (DCM). Cartographic generalization can be executed by means of the 'cartographic driven approach'. This approach deals with the representation of data at a required scale and is primarily based on the skills and views of the cartographer. This is also the main shortcoming of this approach, because the skills and views of the cartographer cannot completely be automated (Cecconi, 2003; Kazemi et al., 2004). In this study, data specifications will be derived to automatically create a secondary DLM. Therefore, the focus will be on model generalization and both object generalization and cartographic generalization will be disregarded.

Figure 2.1: Model of the generalization process



Source: Grünreich, 1985 in Cecconi, 2003, p.11.

2.2. THE DERIVATION OF DATA SPECIFICATIONS

Because a map is never homogeneous in detail or in amount of objects, it is very difficult to create a uniform set of specifications applicable for every generalization type. One of the challenges is to develop these specifications in such a format and with such knowledge level that they can steer the process of automatic generalization (Buttenfield & McMaster, 1991; Stoter et al., 2009a).

The knowledge of specifications for (automatic) generalization can be acquired with the help of 'knowledge acquisition'. Knowledge acquisition can be defined as *"the formalization of generalization knowledge"* (Weibel, 1995, p.59). To access this generalization knowledge, different types of knowledge acquisition are developed. Firstly, 'text documents' can be analyzed to gain an initial idea of the generalization process. However, during the process it is often turned out that those documents are rather vague, incomplete, and short of explaining, or conflicts arise between rules due to the sequence of the documents. Secondly, 'cartographer expert interviews' can be conducted to acquire knowledge about the generalization at the source. The third type of knowledge acquisition is 'reverse engineering'. Reverse engineering starts the generalization by looking at the required end results and attempts to identify tasks that will lead to this result. The output of reverse engineering is often a semi-formal description instead of formal rules. However, it encourages communication between the knowledge engineer and the cartographer and in combination with other knowledge acquisition techniques it supports the development of the generalization process. 'Machine learning' is the fourth type of knowledge acquisition. Machine learning recognizes patterns in a database and based on these patterns, initial rules can be developed. With machine learning, unexpected relations and rules can be discovered. The fifth type of knowledge acquisition is 'neural networks'. This is a specific form of machine learning, which is capable of structuring, classification and template matching. In generalization systems, the most common structuring with neural networks is to classify objects. In addition, neural networks are relevant by the evaluation of parameters. The final type of knowledge acquisition is to gather generalization knowledge by investigating logs of interactive generalization systems, also called 'amplified intelligence' (Weibel, 1995). A combination of those knowledge acquisition techniques is often the most effective way to gather the knowledge for specifications to execute generalization (Muller et al., 1995; Stoter et al., 2009a; Weibel, 1995). In this study, the most common knowledge acquisition techniques used are the analyzing of text documents and reverse engineering with TOP10NL as reference.

The formalization of data specifications consists of two main steps: firstly, to describe both specifications encoded in the data and specifications added by the cartographer's knowledge in such a way that users will understand what the system should achieve. And secondly, to translate these specifications into mathematical rules, which the generalization system can read and automatically can execute (Stoter et al., 2009a; Stoter et al., 2009b). In many researches, constraints are used to define specifications for automatic generalization and to evaluate the automatic generalization process. Constraints describe the generalization output without addressing the method to achieve results to assure that the generalization can be executed with different programs and with different tools. Stoter et al. (2009a) developed a template, which can be used to describe the specifications using constraints. The focus of that research was on cartographic generalization, but also other parts of the generalization process can be defined with this template. In this study, the template will be used to describe the specifications encoded in the data as part of model generalization.

The methodology of Stoter et al. (2009a) followed three steps: firstly, the constraints were defined by adding specifications acquired by knowledge acquisition. Secondly, the constraints were harmonized to create some general concepts within the template. And finally, the constraints were evaluated with the help of three evaluation methods: 'expert evaluation', 'automated constraint-based evaluation' and visual comparison of outputs' (see Section 2.5). The template distinguishes between constraints on one object, on two objects, and on a group of objects. In Table 2.1, the items of the template are identified based on these three template types.

The 'constraint type' is one of the items, which were harmonized. After this harmonization, the constraint type consists of three different types: 'preservation constraints', containing aspects like topology, position, orientation, and shape; 'legibility constraints', which consists of minimal dimensions and granularity; and 'model generalization constraints', which contains the rare model generalization constraints executed in Stoter et al. (2009a). It consists of removing certain objects from the data (Stoter et al., 2009a). In this study, all constraint types are based on 'model generalization constraints'. Therefore, a new classification should be developed to define the constraint types specifically for model generalization.

The difference between 'condition for being concerned with this constraint' and the 'condition to be respected' is that the condition for being concerned with this constraint can be seen as a pre-condition, which should be executed in advance of this constraint, while the condition to be respected gives information on how to handle with this constraint. The condition to be respected is always filled in the specifications, while the condition for being concerned with this constraint not always occurs (Stoter et al., 2009a).

To create the best generalization results, the constraints should be described as formal as possible. In addition, when analyzing the generalization results, missing constraints should be added to the format and unclear constraints need to be refined (Stoter et al., 2009a).

Table 2.1: Items of the template, specified in items on one object, on two objects and on a group of objects

Items on one object	Items on two objects	Items on a group of objects
Generic Constraint ID	Generic Constraint ID	Generic Constraint ID
Constraint Type	Constraint Type	Constraint Type
Geometry Type	Geometry Type	Geometry Type
Feature class	Feature class 1	Feature class
Condition for being concerned with this constraint	Condition for object in class 1 for being concerned with this constraint	Kind of objects of the initial data composing the group
	Feature class 2	Condition (in the initial data) for group being concerned with this constraint
	Condition for object in class 2 for being concerned with this constraint	
	Condition (in the initial data) for them to be concerned with this constraint	
Constrained property	Constrained property	Constrained property
Condition depends on initial value?	Condition depends on initial value?	Condition depends on initial value?
Condition to be respected	Condition to be respected	Condition to be respected
Action	Action	Action
Importance of constraint	Importance of constraint	Importance of constraint
Exception	Exception	Exception
Schema to illustrate if needed	Schema to illustrate if needed	Schema to illustrate if needed

Source: Stoter et al., 2009a.

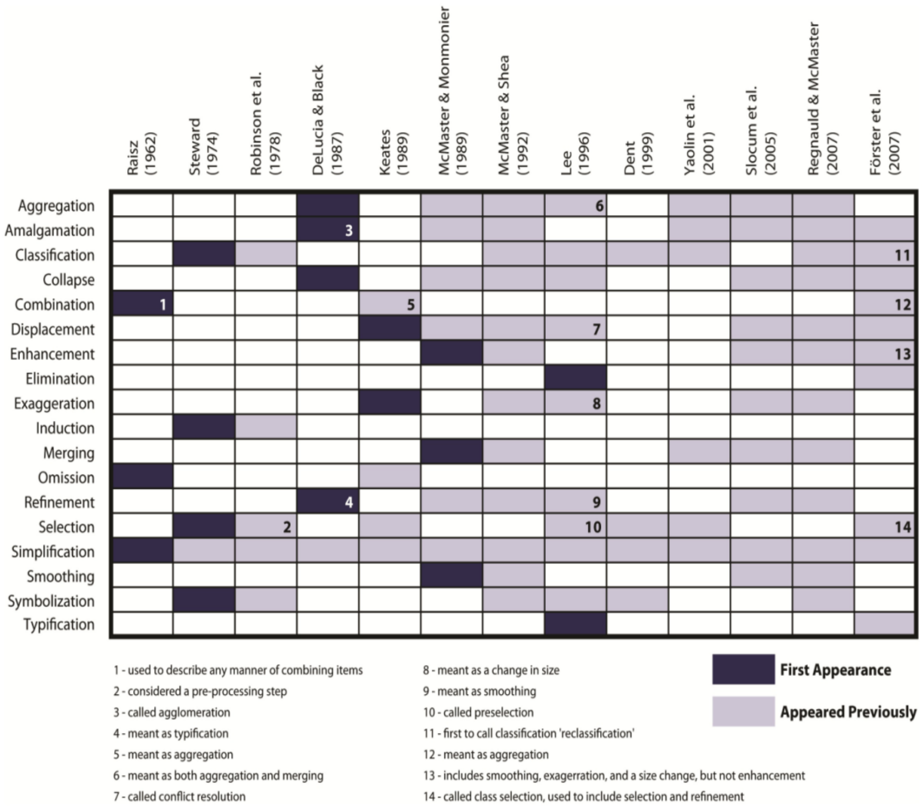
2.3. OPERATORS FOR AUTOMATIC GENERALIZATION

Many researches have tried to define the perfect classification of operators for automatic generalization. However, there are no standard definitions of these operators. Therefore, there are many different classifications available, which are mainly based on the perception of the researcher and their application area (Kazemi et al., 2004). Roth et al. (2012) have compared many of those classifications proposed in different scientific articles (see Figure 2.2). As it turned out, only the operator 'simplification' is acknowledged in all scientific articles. Other operators were also readopted, but less frequently.

Noteworthy is that the definitions of the operators also differ per researcher. For example in Choe & Kim (2007), the operator 'elimination' is seen as model generalization operator, which will be executed as starting point of the workflow. In this article, the operator is defined by *"operator that removes features within each feature class that are unsuitable given the purpose of the application"* (p.104). However, in Foerster et al., (2007), the operator 'elimination' is seen as cartographic operator, which should be executed after the model generalization operators. In this article, the definition of elimination is *"operator that removes the graphic object from the map display"* (p.12). As it turns out, the elimination operator in Choe & Kim is seen as the equivalent of the '(class) selection' operator. Also Roth et al. (2012) mentioned these differences. In Figure 2.2, some comments were made about which operator was actually meant when comparing the definitions.

Another drawback of those many different operators is that there are some operators only mentioned a long time ago. Therefore, in this section, only the most relevant operators mentioned in Roth et al. (2012) will be defined. The generalization operators 'induction' and 'omission' will not be defined due to the fact that they are not mentioned in years. The remaining operators will be defined with the help of the definitions in Foerster et al. (2007), who made a distinction between operators applicable for model generalization and operators applicable for cartographic generalization (see Table 2.2). Although the focus of this study is on model generalization, both model generalization and cartographic generalization operators will be discussed in this section to create a more general overview of operators within (automatic) generalization.

Figure 2.2: Generalization operators identified in several scientific articles



Source: Roth et al., 2012, p.35.

Table 2.2: Generalization operators specified in model generalization operators and in cartographic generalization operators

Model generalization operators	Cartographic generalization operators
Aggregation	Amalgamation
Amalgamation	Displacement
(re)classification	Elimination
Collapse	Enhancement
(class) Selection	Exaggeration
Simplification	Smoothing
	Typification

Source: Foerster et al., 2007; Roth et al., 2012.

2.3.1. MODEL GENERALIZATION OPERATORS

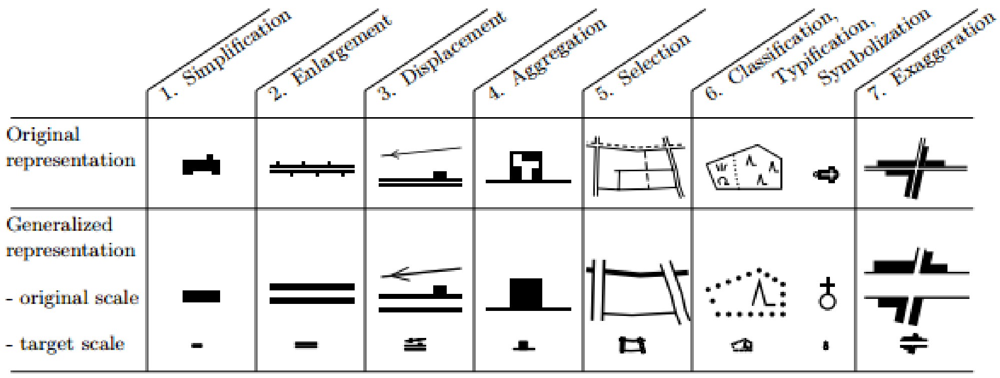
The operators defined in Foerster et al. (2007) as model generalization operators are 'amalgamation', '(re)classification', 'collapse', '(class) selection', and 'simplification'. As stated in Figure 2.2, also the operator 'combination' was defined, but Foerster et al. (2007) used the definition of the operator 'aggregation' to explain the operator 'combination'.

'Amalgamation' merges different graphic objects, which belong to the same object type to one object while protecting the original shape of the outer geometries. It conducts the new outline boundary for the new geometry. The operator amalgamation can be applied both as model generalization operator and as cartographic generalization operator. 'Classification' (or reclassification) enables searching out certain attribute values and gives them a new attribute value. '(Class) selection' also searches out certain attribute values. However it will keep the hierarchy of the objects intact, in contrast with the operator (re)classification. 'Collapse' reduces the complexity of objects and is often triggered by the operator (re)classification. An example is to reduce a polygon into a center point (Foerster et al., 2007). The operator ‘aggregation’ is the counter part of the collapse operator. It aggregates groups of features based on neighboring objects. An example is to combine different polygonal objects into one polygon, or to combine a set of points into one polygon. The operator aggregation has a great impact on the resulting map, because it can change both geometrically and thematically (Foerster et al., 2007; Kazemi et al., 2004; Stoter, 2013). Finally, ‘simplification’ as the word already implies simplifies features. It reduces the amount of data and it deletes aspects of a geometry based on certain criteria (Foerster et al., 2007).

2.3.2. CARTOGRAPHIC GENERALIZATION OPERATORS

The cartographic generalization operators defined in Foerster et al. (2007) are ‘amalgamation’, ‘displacement’, ‘enhancement’, ‘elimination’ and ‘typification’. ‘Amalgamation’ is already noticed in Section 3.2.1, however this operator can be used in both model and in cartographic generalization. In cartographic generalization, it will be used on object level. ‘Displacement’ ensures the moving of objects to prevent overlap. As noticed in Figure 2.2, the generalization operator ‘enhancement’ is in Foerster et al., 2007) defined as a combination of ‘smoothing’ and ‘exaggeration’. In this definition, enhancement emphasizes an object visually. An example of this operator is the smoothing of roads, which can also be seen as a separate operator. The same counts for the operator ‘exaggeration’, which enlarges objects to visualize it better. An example is the enlargement of specific buildings, which are identified based on the importance of those buildings. ‘Elimination’ removes a graphic object from the map display. ‘Symbolization’ symbolizes lines, polylines and points in a structured and clear way. ‘Typification’ reduces the density of spatial objects and its level of detail by replacing a set of graphic objects into a smaller set of graphic objects (Foerster et al., 2007). Figure 2.3 visualizes some of these cartographic operators. It will give more insight in the influence of these generalization operators on the final map (Haunert, 2008).

Figure 2.3: Cartographic generalization operators

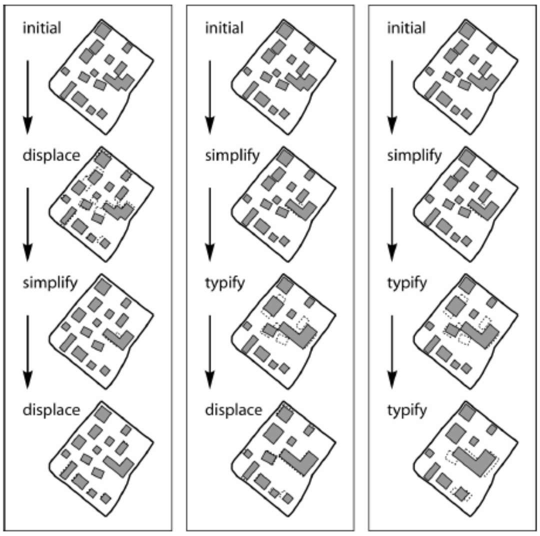


Source: Hake et al., 1994 in Haunert, 2008, p.16.

2.3.3. SEQUENCE OF GENERALIZATION OPERATORS

Operators are often limited within their application. Cartographic operators can be applied on all objects, but model operators are less widely applicable. They can often handle only a specific geometry type (point, line, or polygon). For example, the operator 'simplification' can only handle lines and polygons, and the operator 'collapse' can only be applied on polygons (Choe & Kim, 2007). It is important to notice that there are such limitations by applying the operators, because it can affect the sequence of generalization enormously. The sequence of generalization is important because it greatly influences the generalization process (see Figure 2.4) (Neun et al., 2009). When reducing the database complexity during the generalization and applying the operators in another sequence, the efficiency of the generalization will be maximized. Choe & Kim proposed a framework with the sequence of operators to handle with the different objects (see Table 2.3).

Figure 2.4: Influence of the sequence of applied operators in the generalization process



Source: Neun et al., 2009, p.435.

Table 2.3: Proposed workflows for several features

Features	Geometry	Characteristics	Workflow
Elevation point, bus stop	Point	Unclusterable	Elimination[1]
Tree	Point	Clusterable	Aggregation, elimination[1]
Road (centerline), river (centerline)	Line	Connectable	Elimination[1], classification, simplification
Contour	Line	Unconnectable	Elimination[1], simplification
Parcel	Area	Adjacent	Classification, elimination[1], simplification
River, road	Area	Connectable	Elimination[1], collapse simplification
Building	Area	Clusterable	Aggregation, elimination[1], collapse, simplification
Bridge, reservoir	Area	Unclassified	Elimination[1], collapse, simplification

[1]In the research of Choe & Kim (2007), elimination is meant as 'class selection' (see Section 2.3).

Source: Choe & Kim, 2007, p.107.

2.4. GENERALIZATION OPERATORS IN ARCGIS

In ArcGIS, different tools are available to execute the different generalization operators. In this section, the specific ArcGIS tools are linked to the generalization operators as defined in Section 2.3. Again, to maintain the general overview, both tools for model generalization (see Table 2.4) and tools for cartographic generalization will be outlined (see Table 2.5). However, the focus should be on model generalization.

2.4.1. MODEL GENERALIZATION OPERATORS IN ARCGIS

The difference between amalgamation and aggregation is that amalgamation combines every object and aggregation combines neighboring objects. For both operators, the Dissolve tool can be used in ArcGIS, provided that the 'create multipart features' should be selected when executing an amalgamation, and not selected when executing an aggregation. The tool Unsplit Line aggregates line objects (ESRI, 2014a).

There is no separate tool in ArcGIS, which reclassify vector data, in contrast with raster data, which can make use of the Reclassify tool. Therefore, a combination of tools need to be executed. The first tool is Add Field, which adds a new column to the data table. The second tool is Select Layer by Attribute, which selects all objects for which the reclassification is the same. Finally, with Calculate Field, the specific value will be added as attribute values for these objects (ESRI, 2014a).

For the operator collapse, some ArcGIS tools are available in the toolset. The first is Collapse Dual Lines to Centerline, which derives one line in the middle instead of two lines. The second tool is Collapse Road Detail, which removes road details, e.g. roundabouts. Both tools are supposed to work on road networks. Merge combines two or more datasets into a new dataset. Therefore, the input dataset must have the same type. For example, it is not possible to merge a polygonal dataset with a line dataset (ESRI, 2014a).

Selections can be made with the help of Select Layer by Attribute or Select Layer by Location. Select Layer by Attribute selects objects where specific defined attribute values are the same. The Select Layer by Location specifies all objects within the layer which for example intersect, contains, is within, or have their center in another layer. The location of these objects will define whether or not these objects will be selected (ESRI, 2014a).

The final operator, simplification, has three tools in the generalization toolset: Simplify building, Simplify line, and Simplify Polygon. Simplify Building simplifies the boundary of buildings while their initial shape and size is maintained. Simplify Polygon simplifies also extraneous bends while preserving the initial shape (ESRI, 2014a).

Table 2.4: Model generalization operators in ArcGIS

Model operator	ArcGIS tools	Location (toolbox)
Aggregation	Dissolve, Unsplit Line	Data management toolbox
Amalgamation	Dissolve	Data management toolbox
(Re)classification	Add Field, Select Layer By Attribute values, calculate attribute	Data management toolbox
Collapse	Collapse Dual Lines to Centerline, Collapse Road Detail	Cartography toolbox
Merge	Merge	Data management toolbox
(class) Selection	Select Layer by attribute values, Select Layer By Location	Analysis toolbox
Simplification	Simplify Building, Simplify Line, Simplify Polygon	Cartography toolbox

Source: ESRI, 2014a.

2.4.2. CARTOGRAPHIC GENERALIZATION OPERATORS IN ARCGIS

Also for cartographic operators, there are some tools available within ArcGIS (see Table 2.5). Amalgamation can be executed with the Merge Divided Roads tool, which merges roads containing the same attribute values and situated parallel to each other within a certain merge distance. Displacement can be executed with Resolve Building Conflicts and Dissolve Road Conflicts. Both tools look at graphic conflicts between symbolized features. These tools should be executed after finalizing all symbols. Eliminate, Thin Road Network, and Trim Line delete objects or part of objects which are too small to display in the map image. Eliminate merges selected polygons together to eliminate small polygons in between. Thin Road Network simplifies the road network. And Trim Line deletes part of objects that does not touch another line at the beginning or the end of a line. The operator typification does not have tools available in ArcGIS. However, this operator is also not that often mentioned in the different articles as specified in Figure 2.2 (ESRI, 2014a).

Table 2.5: Cartographic generalization operators in ArcGIS

Cartographic operator	ArcGIS tools	Location (toolbox)
Amalgamation	Merge divided roads	Cartography toolbox
Displacement	Resolve building conflicts, resolve road conflicts	Graphic conflicts toolset
Elimination	Eliminate, thin road network, trim line	Data management toolbox
Enhancement	Smooth line, smooth polygon,	Cartography toolbox
Typification	No tools available in ArcGIS	-

Source: ESRI, 2014a.

2.5. EVALUATION OF THE GENERALIZATION PROCESS

When implementing model generalization, some general requirements should be met: the results of the method should be predictable and repeatable; the amount of deviations between the initial model and the resulting model should be as few as possible; the amount of objects within the dataset should be reduced as much as possible, without violating the purpose of the map; the integrity of original objects should not be violated; the procedure should be controllable by users with minimized parameters and obvious results; and finally, the efficiency of the model should be correct, aiming at data reduction and speeding up computations (Weibel, 1995). The system itself should be able to analyze the map content and measure patterns between geographic objects. Then, the system needs to manipulate those objects in the map space to generate a good mixture of objects. Also, the automatic generalization system needs to evaluate the solutions to refine the system and measures the success of the system. In addition, the system needs a framework, which enables the design of the model at various levels of granularity (Lamy et al., 1999).

The evaluation of this automatic generalization process should balance between human and machine evaluation to create a good overview of the quality of data specifications, the automatic generalization system itself, and the results of automatic generalization (Stoter et al., 2009a). The generalization process as a whole can be evaluated on three different moments. 'A priori evaluation' is necessary to select suitable study areas and assess the existing algorithms and techniques. This evaluation will be executed before the actual knowledge acquisition starts. The second evaluation is 'a posteriori evaluation', which compares and ranks different generalization alternatives during the generalization process. This will be executed by comparing and ranking the different generalization techniques and algorithms. The final evaluation moment is 'ad hoc evaluation'. This evaluation will be performed as control analysis when finishing the generalization process. It will recognize conflicts and measure running times. The results improve knowledge about metadata. Also different parameters can be compared to find the best generalization solution (Weibel, 1995).

As already mentioned in Section 2.2, data specifications can be evaluated with three different methods: 'expert evaluation', 'automated constraint-based evaluation', and 'visual comparison of outputs'. 'Expert evaluation' means that data specifications were evaluated by cartographer experts. These experts complete a survey focusing on both global indicators and on individual constraints of the data specifications. The constraints can be summarized into different types (e.g. granularity, minimal dimensions, relative position etc.) and cartographers assess these types of constraints in terms of bad, badly, well, or very well. 'Automated constraint-based evaluation' compares the measured value with the ideal value. For example, the minimum area of buildings can be compared with the ideal value and therefore, too small buildings can be recognized in the system. However, it occurs that the ideal value is not easy to determine, because the ideal value might consist of rules containing aspects as 'keep most important buildings', which is rather vague. Therefore, the data specifications should be described more formal to keep these aspects out of the evaluation. Finally, the evaluation method 'visual comparison of outputs' will compare the implementation of the data specifications with the expected output. It results in more insight in the completeness, clarity and interdependences of constraints. In addition, it reveals the influence of the tester's experience with the generalization system and data on the generalized output (Stoter et al., 2009a). In this study, the evaluation method 'visual comparison of outputs' will be the most important evaluation method executed.

2.6. PREPARATION OF THIS STUDY

Focusing on automatic generalization, three concepts are of main importance: 'model generalization', 'knowledge acquisition', and the 'evaluation of generalization alternatives'. These three concepts are seen as the so-called building blocks of automatic generalization, i.e. the basics of automatic generalization (Weibel, 1995). In this study, these three concepts are integrated when formalizing the data specifications to execute automatic generalization.

Therefore, the knowledge acquisition techniques 'analyzing text documents' and 'reverse engineering' will be used to acquire the data specifications. These data specifications will be formalized using the template of Stoter et al. (2009a). To see if the data specifications are sufficient and to reveal the cartographers' experience, the data specifications will be implemented in an automatic generalization system using the operators and their associated ArcGIS tools and evaluated with the help of the evaluation method 'visual comparison of outputs'. The exact method will be discussed in Chapter 4. But first, in the next chapter, this study will be justified by explaining the case on which this method is performed.

3. DESCRIPTION OF DUTCH TOPOGRAPHICAL KEY REGISTRATIONS

In this study, data specifications will be developed to integrate the Dutch topographical key registrations. A key registration is an official and governmental registration containing data of high quality. In the Netherlands, different key registrations are available to centrally collect and maintain all governmental data. The data and quality of these key registrations are regulated by law and therefore entirely clear about the content, purpose, quality and accessibility of the dataset. Governmental organizations are obligated to use these datasets when executing their public tasks. Private companies are not obligated to use these key registrations but can if they want to. The different key registrations together form a system of key registrations which can be linked to each other. With this system, the data only need to be obtained for the most relevant key registrations and when other key registrations need the same information, this connection can be made and the data can be easily added (Brink et al., 2013; Kadaster, 2013).

Topographical key registrations contain spatial information and are therefore very useful for solving geo-related tasks. The main purpose of a topographical key registration is to reuse the dataset many times as base for many geo-related tasks. To encourage the use of topographical key registrations for both governmental as for private use, the requirements of the topographical key registrations are based on the national NEN 3610 standards and on the European directive INSPIRE. In addition, the topographical key registrations should be linked to other relevant key registrations and to relevant data models and the key registrations should be of high quality (Brink et al., 2013; Kadaster, 2013).

3.1. TOPOGRAPHICAL KEY REGISTRATIONS IN THE NETHERLANDS

In the Netherlands, there are many different topographical key registrations. In this section, only the 'Basisregistratie Topografie' (BRT), the 'Basisregistratie Grootschalige Topografie' (BGT) and the 'Basisregistratie Adressen & Gebouwen' (BAG) will be outlined to create an overview of the most relevant topographical key registrations used in this study.

3.1.1. BASISREGISTRATIE TOPOGRAFIE (BRT)

The 'Basisregistratie Topografie' (BRT) is currently the most important topographical key registration in the Netherlands. This key registration contains maps with scales at 1:10,000; 1:50,000; 1:100,000; 1:250,000; 1:500,000; and 1:1,000,000. The BRT is nationwide available and consists of objects with different kinds of topographical information e.g. roads, waterways, natural areas, and many other topographical objects. The data is used for many GIS- and web applications and other visualizations, e.g. in topics concerning defense, traffic and transport, and urban planning (Ministerie van Infrastructuur & Milieu, 2014; Kadaster, 2013).

TOP10NL is BRT's most detailed scale. Since 2007, TOP10NL has been produced by the Dutch Cadaster, and since 2013, this dataset has been used as source for automatic generalization systems to derive the smaller scales (Kadaster, 2013). TOP10NL is produced by cartographers, who make use of aerial photographs to recognize the different objects and digitalize these following the implementation rules of TOP10NL (Kadaster, 2012). This digitalization is executed with the software ArcGIS and ESPA. Because it is not possible to recognize every object based on aerial photographs, the data will be improved with field exploration. In this field trip, for example the function of buildings, geographical names, and the usage of terrain areas will be added. When all data is collected, checked and verified, TOP10NL will be saved in the central database (Kadaster, 2012; Kadaster, 2013). TOP10NL is updated five times a year, and also the automatic generalization towards smaller scales will be immediately reproduced after this update (Jonge, 2014).

3.1.2. BASISREGISTRATIE GROOTSCHALIGE TOPOGRAFIE (BGT)

In 2012, the 'Basisregistratie Grootschalige Topografie' (BGT) was established. This object-oriented topographical key registration is meant for the large scales from 1:500 until 1:5,000. In 2016, the BGT will be implemented in the entire Netherlands. The content of the BGT is defined in the information model IMGeo. This model describes which objects should be implemented, the semantics of these objects and how these objects should be displayed. IMGeo consists of two parts: firstly, the description of the different objects in the BGT, which are regulated by law and are required to implement in the BGT; and secondly, optional IMGeo information, which can be added to the dataset, but is not obligated (Brink et al., 2013).

The BGT is developed by source holders, who are collecting and implementing their data themselves for their specific area, following the rules of IMGeo. Examples of source holders are ministries, municipalities, water boards, 'ProRail' and 'Rijkswaterstaat'. When developing the BGT, source holders are enabled to split objects with the purpose to add their own attributes (i.e. 'virtual borders', see Section 3.2.1). After implementing the data, they upload it to the national portal, where objects will be checked and verified by an organization called the SVB-BGT (Brink et al., 2013).

3.1.3. BASISREGISTRATIE ADRESSEN & GEBOUWEN (BAG)

Another relevant dataset is the 'Basisregistratie Adressen & Gebouwen' (BAG). This key registration is formally described in one law, but technically, it consists of two key registrations. On the one hand it contains information about all Dutch addresses, e.g. street names, house numbers, residential areas, etc. On the other hand, it consists of all buildings in the Netherlands with information about the function of these buildings, accommodations etc. (BAG BAO, 2013). In this study, the BAG will be used for comparison purposes when generalizing the buildings of the BGT.

3.2. TOWARDS AN INTEGRATED TOPOGRAPHICAL KEY REGISTRATION

In 2016, two different key registrations will exist presenting topographical information in the entire Netherlands: one for the large scales (BGT) derived by source holders and one for the mid- and smaller scales (BRT) derived with the use of aerial photographs. A logical next step would be to research the possibility of the integration of those two maps, and eventually to create one key registration with all topographical information integrated. With this step, the 'collect once, use many times' principle will be encouraged and the consistency and efficiency of topographical key registrations will be maximized. Then, only the most detailed dataset needs to be collected by source holders and all other datasets, i.e. midscale and small scale datasets can be automatically derived from this dataset (Altena et al., 2013; Stoter, 2009; Stoter, 2013).

3.2.1. THE UNIFORM BGT

To prepare the BGT dataset for this step, the unnecessary and target specific information should be eliminated from the BGT. As mentioned in Section 3.1.2, BGT source holders are allowed to split their objects with the main purpose to add their own information to the data. In IMGeo, the rule is adopted that these 'virtual borders' do not need to be aggregated by source holders, but can be implemented in the BGT as separate objects. The main reason is that these objects do not need to be aggregated because of the complexity of this aggregation. Besides, when looking at the costs of creating this product, it seems more productive to create this product at one location instead of many different source holders trying to aggregate their objects themselves (Stoter, 2013).

Stoter (2013) researched the most extreme scenarios of aggregating objects. This aggregation resulted in the description of four scenarios which could be adopted. Firstly, the aggregation could be obtained as separate product, which is periodically developed and is uniform for the entire Netherlands. This product can be used for data analyses. However, this product will not have the same actuality as the initial data product. The second scenario is that the data will not be aggregated, but this 'aggregation' will be executed only visually. The third scenario is that nothing will happen and that the virtual borders will be kept in the dataset. Perhaps, private companies will need such a uniform product and will be triggered to create one. The fourth and most recommended alternative is to aggregate the product as first step towards the automatic generalization of smaller scales. In this alternative, the unnecessary and target specific data will be eliminated before the real data generalization starts (Stoter, 2013).

The company Webmapper already succeeded to execute the second scenario of Stoter (2013) and aggregated the BGT visually. By adding different layers of visualization on top of each other, the company was able to visually remove the virtual borders. The main shortcoming of this approach is that the aggregation only contains a visual representation. It does not have effect on the underlying datasets of the BGT. For example, the objects with virtual borders are in the underlying data still divided as two (or more) objects. The amount of objects within the BGT will remain the same (Interview with Edward Mac Gillavry, 2013, see Appendix A).

In this study, the fourth scenario of Stoter (2013) will be followed. Therefore, a uniform BGT product will be developed with as main focus to use it as input for the generalization of a midscale BGT data product. In addition, this 'uniform BGT' will be judged if it can be useful as separate product.

3.2.2. THE MIDSCALE BGT

In July 2013, the Dutch Cadaster executed a preliminary research to derive BRT's TOP10NL out of the BGT. In a five day challenge, they have tried to generalize the BGT. Although five days were too short to create a good working automatic generalization system, the results were promising. The main conclusion was that a topographical product at a midscale can be derived from the BGT. However, the midscale product will differ from the current TOP10NL and will contain less information than TOP10NL due to missing information in the BGT and differences in semantics in the data structure. Which information will be missing, and how TOP10NL exactly will change is not researched in this five day challenge (Altena et al., 2013).

In addition to this five day challenge, Nagel (2014) has researched the semantic differences between TOP10NL and the BGT. This comparison reveals that the semantics of both datasets differ tremendously and cannot be adopted easily when generalizing the BGT datasets into TOP10NL. For example, the same names were used for attribute values, but the semantics were different. Or different names were used for attribute values, which eventually have the same meaning. When automatic generalizing a TOP10NL out of the BGT, it is of major importance to define the semantics properly, because it can affect users (Nagel, 2014).

The next step in the generalization process is to develop data specifications, which create this midscale data product, derived from the BGT. The development and the implementation of these data specifications reveal the missing data specifications after analyzing the differences between the current available TOP10NL and the newly developed 'midscale BGT' product. In this study, these data specifications are developed and recommendations are made about how to handle with these missing specifications.

4. METHODOLOGY

The methodology of this study is comparable to the methodology used in Stoter et al. (2009b), who researched the possibility to derive BRT's TOP50NL of BRT's TOP10NL. The research of Stoter et al. (2009b) consists of two steps: firstly, specifications were identified, which cartographers use to implement when interactively generalizing the map; and secondly, those specifications were implemented in an automatic generalization system to compare the different maps. If required, the specifications were enriched or re-implemented with additional information from other sources.

This study follows the same methodology structure. But this study differs fundamentally due to the fact that this study will be executed with a very different, relatively new, and large scale dataset (i.e. the BGT). Therefore, for each generalization product (both the uniform BGT and the midscale BGT), four phases will be run trough. Firstly, the product will be specified by identifying the global specifications (phase 1), followed by the developing of the detailed data specifications (phase 2). Then, those specifications will be implemented into an automatic generalization system (phase 3). And finally, the results will be analyzed with the main purpose to give recommendations about missing or unclear data specifications and the resulting generalization products as national key registration (phase 4).

In both generalization products, every phase will be executed in six cycles. In every cycle, other object types of the BGT will be integrated. This method ensures that at the end of this study, the required data structure of the BGT is covered. These six cycles are following a specific order: 'roads', 'buildings', 'water', 'nature', 'bridges & tunnels', and 'other'. The reason behind this order is that roads and buildings are the most prominent object types in a midscale data product. Therefore, it is important to start with those object types to integrate the cycles more easily in a later stadium. Bridges and tunnels are treated as separate cycle, because they can refer both on roads as on water and on nature. In Figure 4.1, the parts, phases and cycles are visualized. In Table 4.1, the distribution of the required object types of the BGT over the different cycles is revealed.

Figure 4.1: An overview of the methodology used in this study

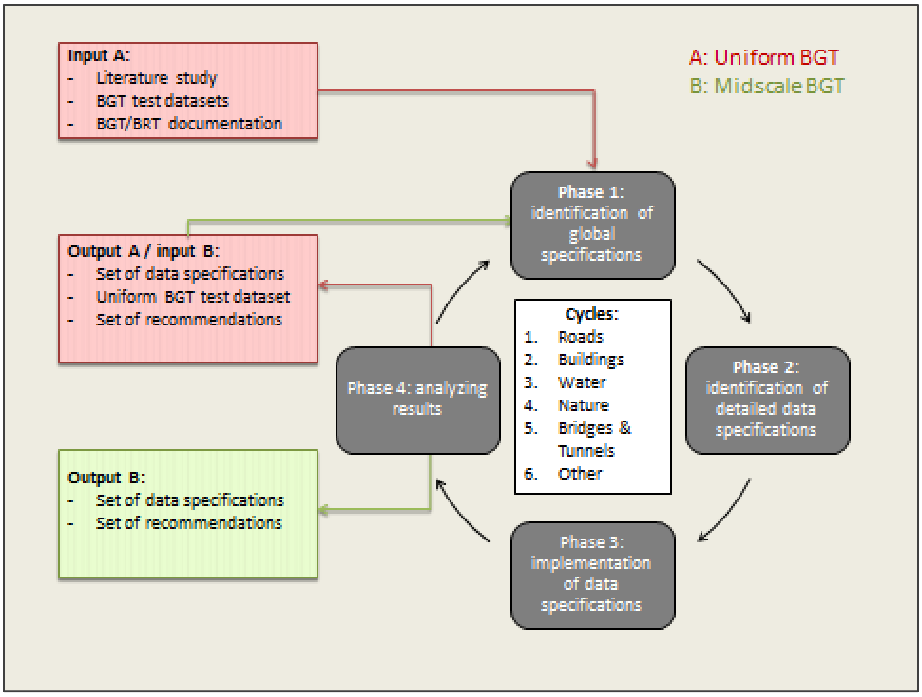


Table 4.1: The distribution of the object types of the BGT over the different cycles

Cycle	BGT object types
Roads	PARTOFROAD, SUPPORTIVEPARTOFROAD, RAILWAY
Buildings	BUILDING
Water	PARTOFWATER, SUPPORTIVEPARTOFWATER
Nature	COVEREDPARTOFTERRAIN, UNCOVEREDPARTOFTERRAIN
Bridges & Tunnels	PARTOFBRIDGE, PARTOFTUNNEL
Other	FUNCTIONALAREA, ENGINEERINGSTRUCTURE, REMAININGSTRUCTURE, SEPARATIONS

Source: Brink et al., 2013.

In this chapter, the methodology will be further specified. Firstly, the different inputs will be outlined (see Section 4.1). Then, the four phases will be explained for the uniform BGT (see Section 4.2) and for the midscale BGT (see Section 4.3). After that, the expected outputs will be explained (see Section 4.4). This chapter ends with an overview of the main principles which are maintained in this study to keep focused on the goals of this study (see Section 4.5).

4.1. INPUT

In the theoretical background of this study (see Chapter 2), more insight is given on what is already researched in the area of automatic generalization and the development of data specifications. The gathering of these concepts into a literature study have led to an extensive knowledge on knowledge acquisition, data specifications, tools for automatic generalization and on the evaluation of (automatic) generalization and data specifications in particular. Throughout the phases of this study, these concepts will be used and implemented.

To develop the different data specifications, the knowledge acquisition technique 'analyzing text documents' will be used. The generic requirements for a key registration will be gathered from the information models, which describe both the BGT and the BRT (Brink et al., 2013, Kadaster, 2013). The requirements for the uniform BGT are already defined in IMGeo (Brink et al., 2013). The requirements to describe the midscale BGT are not yet determined. Therefore, the assumption is made that the implementation rules of TOP10NL, which describe how to create TOP10NL, as specified in the 'Verkenningsvoorschriften BRT' (version 2012.2) (Kadaster, 2012) in combination with the standard BGT requirements of IMGeo (Brink et al., 2013) are sufficient to gather the data specifications for a midscale BGT.

Additional information that is used by the generalization of the midscale BGT are the 'Objectenhandboek BGT' (Brink et al., 2012), which specifies each attribute value in detailed pictures to give more information on how the BGT is developed; the semantic differences of the BRT in comparison with the BGT (Nagel, 2014); and the documentation of the five day challenge, which the Dutch Cadaster has executed to make a first step in the generalization of the BGT towards a BRT (Altena et al., 2013).

In addition, BGT test datasets will be used to test the developed specifications in an automatic generalization system. In Chapter 5, the choice for the BGT test datasets used in this study will be outlined and the quality of those test datasets will be discussed. Also some other datasets will be used for comparison, i.e. BRT's TOP10NL and the BAG.

4.2. THE PHASES OF DEVELOPING SPECIFICATIONS FOR THE UNIFORM BGT

The aim of the uniform BGT is to create test datasets without all unnecessary data as preparation for the midscale BGT. In addition, some recommendations will be made about the uniform BGT as a nationwide key registration. To obtain this goal, data specifications will be developed and implemented in the BGT test datasets to obtain an aggregated, uniform BGT product. Also some other basic generalization operators will be applied. The uniform BGT will be developed following the four phases:

Phase 1: Identification of global specifications
In this phase, the uniform BGT will be identified with the help of the knowledge acquisition techniques as outlined in Section 2.2. The main goal of this identification is to get information on how the BGT should be aggregated into a uniform BGT product, which can be used both as input for the midscale BGT, and as national key registration. In the uniform BGT, this phase consists of the identification of the uniform BGT by recognizing the virtual borders and other unnecessary data in the test datasets. .

Phase 2: Identification of detailed data specifications
After knowing how the uniform BGT should be identified, the detailed data specifications, which aggregate these unnecessary data, can be defined. This set of data specifications will ensure uniformity of the BGT or at least uniformity of the test datasets of the BGT. The data specifications will be developed following the templates of Stoter et al. (2009a), which were identified in Section 2.2. As already outlined in this section, the data specifications should be formulated as formal as possible.

Phase 3: Implementation of data specifications
The main purpose of the third phase is to create a uniform BGT test dataset as independent dataset and use this test dataset as input for the midscale BGT. The automatic generalization will be executed in ArcGIS with ModelBuilder. ArcGIS has different toolboxes, which contain an efficient and universal system of tools usable for automatic generalization (see Section 2.4). This is also the testing phase of the data specifications, which are developed in phase 2.

Phase 4: Analysis and evaluation
In the final phase, the results of the previous three phases will be analyzed. The main goal of this phase is to create recommendations about the uniform BGT as national key registration. In addition, recommendations will be developed about the derived data specifications and about the processes of automatic generalization. When analyzing the data specifications, the formality of description, missing data specifications, and unclear data specifications should be refined (Stoter et al., 2009a).

4.3. THE PHASES OF DEVELOPING SPECIFICATIONS FOR THE MIDSCALE BGT

The aim of the midscale BGT is to develop data specifications derived from the BGT, which can be used by the integration of the key registrations. In this part, data specifications are developed with the help of the following four phases:

Phase 1: Identification of global specifications
In this phase, the knowledge acquisition techniques ‘reverse engineering’ will be used to make a comparison between the required object types, attributes and attribute values of the BGT and TOP10NL to identify how the midscale BGT will look like and what the expected effects and differences are when replacing TOP10NL. These comparisons are made manually and with a special attention to the generalization that needs to be applied.

Phase 2: Identification of detailed data specifications
After the identification of the expected differences between the midscale BGT and TOP10NL, the detailed data specifications will be developed with the help of the template of Stoter et al. (2009a). This will be executed following the six cycles and will result in a set of data specifications, which are ready to implement into the automatic generalization system. These data specifications will be developed with the help of the implementation rules of TOP10NL in combination with IMGeo, as already mentioned in Section 4.1.

Phase 3: Implementation of data specifications

The main purpose of the third phase is to test whether or not the data specifications that are developed really work in practice. In contrast with the uniform BGT, the focus of the midscale BGT will be on the development of data specifications rather than on the development of a working generalization system. Still, this phase will be executed to gain the quality of the developed data specifications as comprehensive as possible. The automatic generalization will be executed in ArcGIS with ModelBuilder and with the help of the generalization operators as outlined in Section 2.3.

Phase 4: Analysis and evaluation

The final phase is to analyze the results and to create recommendations on whether or not the data specifications of the midscale BGT product serve the needs of a key registration as replacement of TOP10NL. In this phase, recommendations will be developed with the help of the evaluation method 'visual comparison of outputs', whereby the developed midscale BGT will be compared with TOP10NL. With the help of the resulting recommendations, the data specifications will be analyzed and the formality of the description, missing data specifications, and unclear data specifications will be discussed. When dealing with missing data specifications, the following questions should be answered: is it rational that missing information will be added to the source data? Is it possible to add missing information from supplementary data? Or is the generalization system able to generate the missing information automatically (Altena et al., 2013)?

4.4. OUTPUT

The output of this study will be a set of data specifications for the uniform BGT and the midscale BGT written down following the template as specified in Stoter et al. (2009a) (see Section 2.2). These specifications can be used in the near future for the creation of a nationwide topographical key registration. In addition, a test dataset of the uniform BGT will be created with as main purpose to test the data specifications of the midscale BGT. Furthermore, a set of recommendations will be part of the output, as well for the uniform BGT as for the midscale BGT.

4.5. MAIN PRINCIPLES IN THIS STUDY

In this study, four main principles are maintained to keep focused on the main goal. Firstly, the midscale BGT will be developed seen from a 'BGT-perspective'. This means that the structure and semantics of the BGT object types, attributes and attribute values will be remained as much as possible. However, the main differences with BRT's TOP10NL will be emphasized and where there is enough reason to adopt the structure of TOP10NL, this structure will be applied. I have chosen for this perspective, because this requires the least amount of adjustments, e.g. in the semantics of the attribute values or in the structure of the dataset, and at the same time reveals the most important differences.

The second principle in this study is that only the required object types, attributes and attribute values of the BGT and the BRT will be used in this study. Therefore, in the uniform BGT, the optional object types and attributes as defined in IMGeo will be excluded. These optional object types and attributes are not uniform obtained and therefore not uniform for the entire Netherlands.

The third principle is that the midscale BGT will be compared based on BRT's TOP10NL, which is currently the only midscale product that is nationwide available (Altena et al., 2013). The BGT and the BRT will, eventually, be united into one key registration, and therefore, it is interesting to investigate the differences between those products. The differences between the datasets reveal the data specifications, which cannot be solved from a BGT-perspective. The missing data specifications, which are probably of main importance for the midscale BGT will be converted into requirements.

The fourth principle is that both generalization products as developed in this study should meet the quality of the topographical key registrations so that the resulting product can be implemented in the system of topographical key registrations (see Chapter 6). The only exception is that the history of objects will not be created, because the datasets will become enormous during the testing phase of this study (Altena et al., 2013).

5. SELECTION AND QUALITY OF THE BGT (TEST) DATASETS

BGT test datasets will be used to develop the different data specifications and to test those specifications in an automatic generalization system. Due to the fact that source holders are still trying to get the data ready for implementation in 2016, at the start of this study, only some example datasets were made available on the website of Geonovum. These example datasets were built as example for source holders and are not officially checked and verified by the SVB-BGT. However, these example datasets are modified to create a 100% technical validation of the dataset. This means that the content of the dataset can deviate from reality (Geonovum, 2013). The example datasets 'Amersfoort' and 'Maastricht' are used to develop and implement the specifications in this study. During the execution of this study a few source holders managed to submit data to the national portal, which resulted in two completed datasets, i.e. within the municipality of Dronten, and within the municipality of Valkenswaard. These datasets were verified by the SVB-BGT and ready for implementation in the national database. I have chosen to use one of these datasets, i.e. the dataset of Dronten, as verification of the generalization to ensure the model is correct and works on 'real' BGT data.

5.1. SELECTION OF THE BGT TEST DATASETS

The three BGT datasets used in this study, i.e. Amersfoort, Maastricht and Dronten, are selected based on the amount of object types and the availability of many different attribute values in the dataset. Another selection is based on the total amount of objects. With a compacter dataset, the running time of the model will be less, which is more efficient during the testing phase. In Table 5.1, the amount of objects is shown per available BGT dataset. In Appendix B, the differences in attribute values between the different datasets are made available.

Table 5.1: The amount of objects within each object type per available (test) dataset

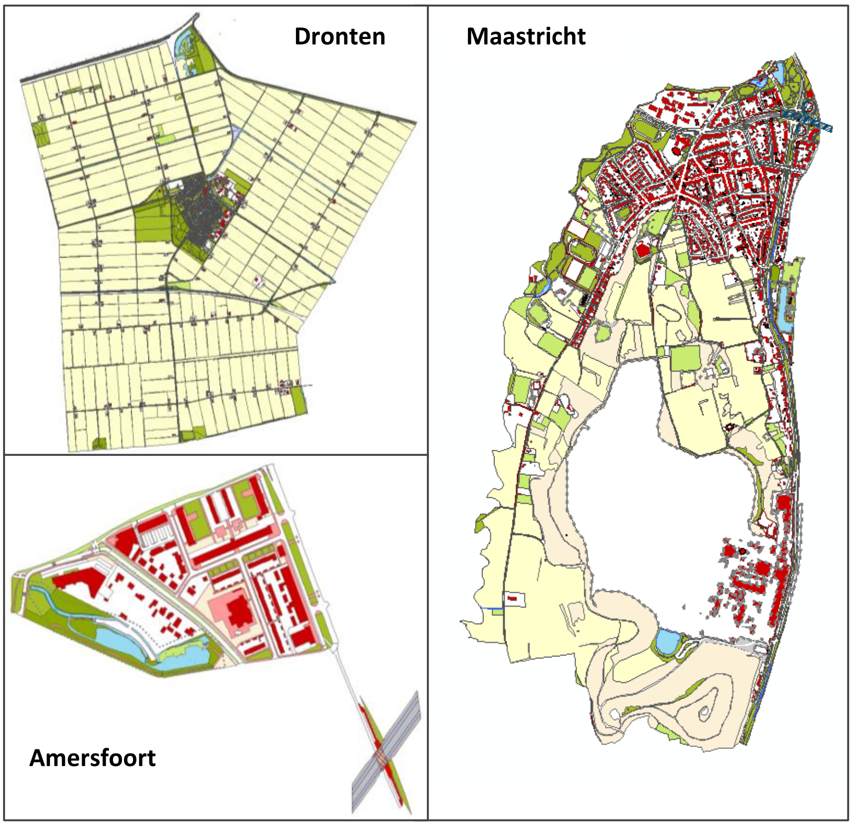
Cycle	BGT Object type	Amersfoort dataset	Dronten dataset	EZ_Limburg dataset	Maastricht dataset	Valkenswaard dataset	Venray dataset
Roads	PARTOFROAD	94	2187	-	1043	12531	19685
	SUPPORTIVEPARTOFROAD	46	565	-	151	1770	6118
	RAILWAY	4	-	-		-	-
Buildings	BUILDING	164	5023	-	2592	21259	29525
Water	PARTOFWATER	1	798	-	13	2545	9635
	SUPPORTIVEPARTOFWATER	1	93	-	1	451	7272
Nature	COVEREDPARTOFTERRAIN	51	2147	3105	1053	9039	13947
	UNCOVEREDPARTOFTERRAIN	24	655	-	329	2680	23294
B&T	PARTOFBRIDGE	5	78	-	37	68	528
	PARTOFTUNNEL	1	-	-		-	3
Other	FUNCTIONALAREA	1	-	-	42	-	-
	ENGINEERINGSTRUCTURE (polygon)	1	50	-	5	11	77
	ENGINEERINGSTRUCTURE (line)	4	118	-		99	-
	REMAININGSTRUCTURE	7	2	-	271	66	15366
	SEPARATIONS (polygon)	7	-	-	39	745	305
	SEPARATIONS (line)	23	3	-	493	8620	-
	UNCLASSIFIEDOBJECT	3	-	-		-	-
Total		437	11719	3105	6069	59884	125755

The main advantage of the Amersfoort dataset (see Figure 5.1, lower left) is that it contains all object types of the BGT. In addition, it is a very small dataset, which ensures that the model will run very fast on this dataset. As outlined in Table 5.1, this dataset contains 437 objects. Therefore, the Amersfoort dataset is valuable to use in the early stages of developing the model. However, other datasets should be used to detect the main exceptions in the dataset, for which the model should be modified. Another disadvantage of this dataset is that it only contains urban area, which means that the model cannot be tested for natural areas.

The Maastricht dataset (see Figure 5.1, right) is a bigger dataset than Amersfoort. This dataset contains both urban areas as well as outlying natural areas. In addition, this dataset is chosen mainly because it contains a variety of different attribute values in comparison with the Amersfoort dataset (see Appendix B) and is therefore complementary to the Amersfoort dataset.

The Dronten dataset (see Figure 5.1, upper left) is the first official BGT dataset. It is checked and verified by the SVB-BGT and ready for implementation in the national database. The dataset contains a small part of the municipality of Dronten, with the village of Swifterbant in the center of the dataset. The dataset is quite big comparing to the other datasets, but not as big as the Valkenswaard dataset, which is the second dataset that was officially launched. Therefore, the Dronten dataset is selected as verification dataset in this study. In addition, this dataset has relatively more natural areas than the other test datasets, and therefore the addition of this dataset will result in an overall good mixture of urban and natural areas.

Figure 5.1: Screenshot of the test datasets: Dronten (upper left), Amersfoort (lower left) and Maastricht (right).



The dataset of EZ_Limburg contained only one object type, i.e. COVEREDPARTOFTERRAIN, and the dataset of Valkenswaard and Venray were too big for the purposes of this study, i.e. testing the data specifications. In addition, these three datasets did not contain an additional amount of different attribute values. And therefore, these three datasets were not selected as test datasets in this study.

5.2. QUALITY OF THE BGT TEST DATASETS

As already mentioned, the BGT test datasets were checked on its amount of required attribute values. However, after the selection of those three datasets, still not all attribute values can be tested. In Table 5.2, a list of those missing attribute values per required object type is outlined. One of the reasons to have those missing information is because attribute values are not psychically present in Amersfoort, Maastricht or Dronten. For example, the attribute 'ocean' is obviously missing in those three datasets, and there is no 'tram' available. These attribute values cannot be implemented in the BGT for that area, so these attribute values will not be added to the test datasets. Because most missing attribute values are also not available in the datasets which are not selected in this study (with the exception of 'level crossing' and 'coniferous forest', both available in the Venray dataset), the decision is made that none of the missing attribute values will be discussed in this study.

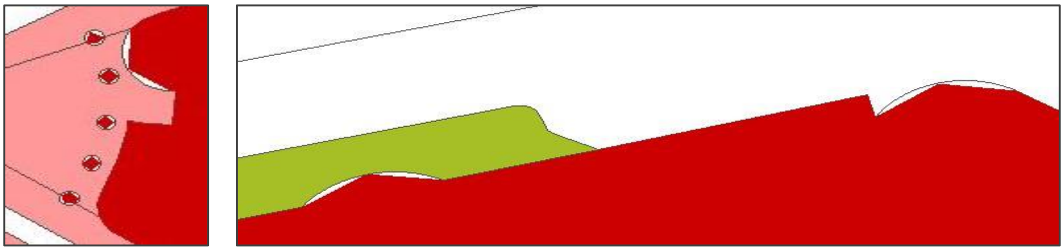
Table 5.2: Missing attribute values in the test datasets

Cycle	Object type	Attribute	Attribute values
Roads	PARTOFROAD	*Function*	*'level crossing'*, 'path for air traffic', 'bridle path'
	RAILWAY	*Function*	'metro', 'tram'
Water	PARTOFWATER	*Type*	'ocean'
	SUPPORTIVEPARTOFROAD	*Type*	'silt'
Nature	COVEREDPARTOFTERRAIN	*physicalOccurrence*	*'coniferous forest'*, 'heath', 'dune', 'reed land', 'saltmarsh', 'tree cultivation'
Other	FUNCTIONALAREA	-	'barrier'
	ENGINEERINGSTRUCTURE	*Type*	'bucking', 'platform', 'sluice', 'breakwater'
	REMAININGSTRUCTURE	*Type*	'open shed', 'storage tank', 'settling tank'
	SEPARATIONS	*Type*	'quay wall'

Attribute values in italics: the attribute values which are available in other BGT test datasets.

The test datasets were made available in GML, which the Cadaster converted into file geodatabases to be applicable in ArcGIS. However, ArcGIS does not recognize the concepts of arcs (in Dutch: 'gestrookte bogen'). This resulted in holes as visualized in Figure 5.2. The problem is notified by ESRI, and is not solvable at this moment. If problems occur to these arcs, this will be explained when implementing the data specifications (phase 3) of both generalization products.

Figure 5.2: Examples of holes due to 'arcs' in ArcGIS



5.3. OTHER DATASETS USED FOR COMPARISON

In addition to the BGT test datasets, two other datasets are used for comparison. As already outlined in Section 4.5, TOP10NL will be used to compare the developed BGT products. Therefore, TOP10NL 1.1 will be used as reference. The second dataset, which will be used for comparison is the BAG. This dataset will be used to compare the buildings in the BAG with the buildings in the uniform and the midscale BGT.

6. GENERIC REQUIREMENTS OF A KEY REGISTRATION

In this chapter, the generic requirements of a key registration, which should be applied on both the uniform BGT and the midscale BGT, will be discussed. Firstly, the quality requirements of topographical key registrations will be outlined. Secondly, some of those quality aspects will be discussed to determine how the generic requirements should look like when applying those to the generalization products.

6.1. QUALITY REQUIREMENTS OF TOPOGRAPHICAL KEY REGISTRATIONS

To determine the quality of topographical key registrations, some generic concepts should be met. These concepts are: 'actuality', 'positional accuracy', 'completeness', 'logical consistency', 'thematic accuracy', and 'time accuracy'. 'Actuality' measures the degree in which data harmonizes the reality within a specific time interval. The obtaining and the processing of key registrations need to take place as frequent as possible. 'Positional accuracy' measures the degree in which the coordinates of an object harmonizes reality. Every object within an object type contains a minimum accuracy level in which the coordinate should match the physical environment. Concepts of positional accuracy are 'precision', which measures the degree in which the obtaining and processing of an object contains the same results; 'reliability', which checks the location of a point for the second time to make sure the location is correct; and 'idealization', which is the precision of a measured point. For example, corner points of buildings can be measured more precisely than the borders of a river (Brink et al., 2013).

'Logical consistency' determines that all objects together should cover ground level completely. This means that there should be no gaps or overlaps in the dataset on a specific designated 'ground level'. 'Completeness' is the degree in which objects match the physical environment. For every polygonal object on ground level, the completeness should be 100%. For the remaining objects, the completeness should be 98%. During transition, all objects need to be filled, including their geometry, attributes, and attribute values. However, some attribute values cannot be filled yet. 'Thematic accuracy' is the correctness of the dataset. This contains the degree in which the data and the physical situation are the same. For example, names and house numbers should have a thematic accuracy of minimal 98% (Brink et al., 2013).

And finally, 'time accuracy' registers the following times in topographical key registrations: *objectBeginTime, objectEndTime, timeOfRegistration, endRegistration, LV-publicationdate, gainingDate*. These dates and times will be registered following the notation from ISO (yyyy-mm-ddThh:mm:ss), which means that the accuracy of time registrations is in years, months, days, hours, minutes and seconds. However, sometimes only dates are sufficient (yyyy-mm-dd). Obviously, the times will be registered following the Dutch time zone (Brink et al., 2013).

6.2. GENERIC REQUIREMENTS FOR THE GENERALIZATION PRODUCTS

In this section, the concepts 'logical consistency', 'identification of objects and time accuracy' and the 'completeness of objects' will be discussed. How to deal with those concepts after generalizing the BGT into the uniform BGT and the midscale BGT?

(a) Logical consistency

Between the BGT and TOP10NL there is a major difference in its definition of ground level. In the BGT all polygonal objects should cover ground level completely. To ensure there are no gaps or overlap between the objects in the dataset, the attribute *relativeHeight* is used to determine ground level (level 0) per object. All objects together with this ground level cover the area completely. In reality, not all objects are on the same level. For example, bridges cross roads or waterways, and tunnels go underneath roads or waterways. These objects contain a different *relativeHeight*. Bridges can be recognized with a positive level (level 1, 2, 3, etc.) and tunnels receive a negative level (level -1, -2, -3, etc.). This number is relative to its neighboring objects (Brink et al., 2013). In contrast to the BGT, TOP10NL has a different way of determining ground level. In TOP10NL, only the polygonal object types PARTOFROAD, PARTOFWATER, and TERRAIN are part of ground level. In addition, TOP10NL sees ground level as what is visible from the aerial photographs of which TOP10NL is developed. This means that every object on top is seen as ground level and every object below these objects will receive a negative attribute value within the attribute *heightLevel*. For example, a bridge will receive *heightLevel* 0 and is therefore part of ground level. All objects underneath this bridge (e.g. objects of the object type PARTOFWATER) will receive a negative value (-1, -2, -3, etc.) relative to its neighboring objects (Kadaster, 2013). Because this study is seen from a BGT-perspective, the choice is made to allow all available polygonal objects within the midscale BGT to determine ground level and the *relativeHeight* is preserved as attribute instead of *heightLevel*.

(b) Identification and time accuracy.

Another requirement of a (topographical) key registration concerns the identification of objects. Every object within a key registration should contain a unique identification. This identification of objects in a topographical key registration is of main importance for the linkages between the key registrations. This identification is selected conform NEN 3610 standards and consists of two parts: a name (*namespace*), which identifies the dataset, and an identification code (*identificationcode* or *identificationLocalID*). The identification will be identified at the developing of an object, together with the *objectBeginTime*, which determines the creation of an object, and the *timeOfRegistration*, which determines the time of change of an object. Both attribute values will be set on the same date when an object is created. As soon as the object is registered in the national facility, it will get a *LV-publicationdate*, which is the date of publication. As long as the object exists, the identification of an object cannot change. When an object changes, the object receives an *endRegistration* and the source holder will create a new version of the object with the same *namespace, identificationcode, objectBeginTime,* and a newly developed *timeOfRegistration*. When the object is reregistered in the national facility, it will receive a new *LV-publicationdate*. When the situation completely changes, because an object splits or combines with other objects, new objects will be created and old objects will be expired. In case of expiring, the *namespace* and the *identificationcode* are still registered, but the *objectEndTime* and the *endRegistration* will be added (Brink et al., 2013; Kadaster, 2013). In this study, when an object changes in the uniform BGT, this object will be treated as new object. In the midscale BGT, due to scale changes, all objects will be treated as new objects, the history of objects will not be determined (see Section 4.5).

(c) Completeness of objects

In the BGT, an object can be unclassified in two different ways: firstly, via the object type UNCLASSIFIEDOBJECT, containing all unknown objects, which are even not specified per object type. And secondly, objects within the generic object types, which are not specified yet. These objects often have the attribute value ‘in transition’ to specify that the attribute values are still to come. Both objects in UNCLASSIFIEDOBJECT and objects containing the attribute value ‘in transition’ can be part of ground level, and therefore, these objects should be maintained in both generalization products.

THE UNIFORM BGT

7. THE UNIFORM BGT AS INPUT FOR THE MIDSCALE BGT

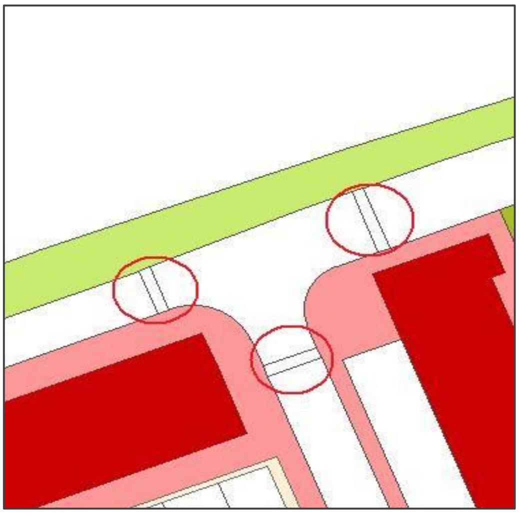
To prepare the generalization of the midscale BGT, the uniform BGT is developed, which aggregates the BGT test datasets on all its unnecessary and target specific data. The definition of aggregation in this context is that neighboring objects with the same attribute values should be combined (Stoter, 2013). The aggregation will result in a compacter dataset, which eases the use of the BGT in the generalization into a midscale BGT. In addition, the uniform BGT as an entirely new product in line with the key registration requirements will be discussed.

7.1. PHASE 1: IDENTIFICATION OF GLOBAL SPECIFICATIONS

Figure 7.1: Example of virtual borders (red circles)

As already outlined in Section 3.2.1, the 'virtual borders' are causing inconsistency in the BGT dataset. The objects with virtual borders are collected and maintained by source holders to be able to add their organization specific attribute values to the dataset and these should be aggregated with neighboring objects. By law, it is not obligated to aggregate these virtual borders, because law only regulates source data. Therefore, in IMGeo, the rule is adopted that objects with virtual borders may be registered in the BGT as two different objects. However, a derived product like the BGT can be developed, provided that the data cannot have another quality or other user requirements than the source data (Brink et al., 2013; Stoter, 2013). An example of virtual borders as existing in the test datasets is outlined in Figure 7.1. Here, three objects contain the same required attribute values, i.e. 'verge' (*function*), 'greening' (*physicalOccurrence*), and the optional IMGeo attribute value 'grass-herbaceous plants' (*plusPhysicalOccurrence*). The source holder thought it necessary to split those objects instead of combining them. In the uniform BGT, these objects will be aggregated into one object.

Figure 7.2: Example of optional objects (red circles)



However, not only virtual borders are causing inconsistency in the dataset. IMGeo also consists of optional data. This optional data will not always be gained by source holders, and therefore, this optional data is not uniform for the entire Netherlands (Altena et al., 2013; Brink et al., 2013). Therefore, in this study, the assumption is made that the uniform BGT and the midscale BGT should exist of the required BGT as specified in IMGeo (see Section 4.5). The optional data can be recognized within the BGT both as object types and as optional attributes containing optional attribute values within required object types. Due to the diversity of optional data within the objects, this can result in many different objects existing side by side containing the same required data, but different optional data. To make the BGT more uniform, both in the visualization of the product and in the reduction of data, these objects should be aggregated. For example, in Figure 7.2, the optional attribute value 'speed bump' is added to the object type PARTOFROAD. The required attribute values are the same for every surrounding object in PARTOFROAD. Therefore, these objects should be aggregated.

As mentioned in Section 3.2.1, an important aspect of the uniform BGT is that the aggregation should be executed with the attention on the attributes and attribute values of the BGT. In contrast with the work of Webmapper, who achieved the aggregation based on the visual representation of objects, this uniform BGT should accomplish that the data behind the BGT is also aggregated. In the next sections, the data specifications, the model and the results will be discussed.

7.2. PHASE 2: IDENTIFICATION OF DETAILED DATA SPECIFICATIONS

In IMGeo is exactly determined which object types, attributes and attribute values are required in the BGT and which object types, attributes and attribute values are optional (see Appendix C for a list with required and optional object types and attributes). In this phase, the optional object types will be excluded. The optional attributes and virtual borders will be aggregated following the data specifications as developed in this phase. Therefore, the template 'items on one object' will be used (Stoter et al., 2009a). In Appendix D, the data specifications for the uniform BGT are defined.

It is important to have knowledge which attributes should be remained and which attributes should be aggregated. Therefore, the required attributes of every object type will be selected (see Appendix C, Table C.1). For example, the required attributes of the object types PARTOFROAD are *function, physicalOccurrence,* and *onSlope*. These attributes should remain in the uniform BGT. In addition, the attribute *relativeHeight* will be added to these attributes. Because then, objects will not be aggregated when they cross each other on a different height. For example, junctions will be considered as one road type instead of multiple roads on top of each other when not adding *relativeHeight* to the attributes on which the aggregation is executed.

The data specifications will be formalized for every object type. Therefore the **condition to be respected** is set on 'spatially neighboring objects with the same attribute values on [required attributes] and *relativeHeight*'. The **action** will be 'aggregate'.

Some exceptions can be recognized when developing the data specifications for every object type of the BGT. Firstly, the object type BUILDING only contains the required attribute *identificationOfBAGBuilding*, which ensures the connection of the BGT with the BAG. When executing the aggregation on this attribute, the prospect is that none of the objects will aggregate due to the fact that every building should have its unique BAG identification number. Therefore, it is not necessary to aggregate the object type BUILDING. However, in this study, this will be tested to see if it really results in the same amount of objects.

Secondly, the BGT object type UNCLASSIFIEDOBJECT is exceptional because it identifies objects, which are not identified yet to which object type it belongs. The objects within UNCLASSIFIEDOBJECT do not contain required attributes, which can be used to aggregate the objects. It is possible that neighboring objects of this object type are very different from each other. Therefore, in the uniform BGT, UNCLASSIFIEDOBJECT should not be aggregated.

In Appendix C, the attribute values of the required attributes per object type are mentioned (see Table C.1). One of these attribute values is 'not-bgt', which appears in the object types FUNCTIONALAREA, ENGINEERINGSTRUCTURE, REMAININGSTRUCTURE, and SEPARATIONS. As the name already implies, the 'not-bgt' attribute value is officially not part of the required BGT. However, the source holder found it too important to add to the required attributes, often along with an optional attribute value. Due to the fact that it was initially not required information, the assumption is made that it would not be important and therefore should not be adopted in the uniform BGT. However, the possibility exist that these objects exist on ground level, which means that it would create holes after eliminating the objects. To analyze how many objects this concern and what impact it has on the uniform BGT, the objects with 'not-bgt' attribute values will be eliminated from the dataset.

The data specification which encourages this elimination is formalized with the **condition to be respected** set on 'objects with attribute values 'not-bgt' within the attribute *type*'. In addition, the data specifications, which aggregates the objects of the object types FUNCTIONALAREA, ENGINEERINGSTRUCTURE, REMAININGSTRUCTURE, and SEPARATIONS, are adapted. Therefore, the **condition to be concerned with this constraint** is set on 'objects with attribute values 'not-bgt' should be eliminated'. The **condition to be respected** in these data specifications is set on 'spatially neighboring objects with the same values on *type* and *relativeHeight*' to aggregate the objects in these object types. When this results in the elimination of many 'not-bgt' objects on ground level, an appropriate solution should be found to preserve the logical consistency.

After aggregating the different objects, the formal key registration requirements should be added to the data again and changed where necessary. The attributes that need to be added are: *namespace, identificationLocalID, objectBeginTime, timeOfRegistration, sourceholder, inResearch,* and *bgt-status.* As stated in Section 6.2, when an object splits or combines, new objects should be created and old objects should be expired. The expiring of objects is outside the scope of this study (see Section 4.5). The *identificationLocalID, objectBeginTime,* and *timeOfRegistration* of the objects which did aggregate, need to be changed to specify the new object. Therefore the data specification is developed with the **condition for being concerned with this constraint** is set on 'aggregation needs to be executed and aggregated objects >1'. The *identificationLocalID* will be identified with a random generated code containing the source holder and a random code (*sourceholder*.RandomString). The random code should have 32 characters containing random letters (varying from a-f) and numbers (varying from 0-9). *ObjectBeginTime* and *timeOfRegistration* should be defined by the date (and time, in case of *timeOfRegistration*) of the specific moment that the model runs.

7.3. PHASE 3: IMPLEMENTATION OF DATA SPECIFICATIONS

To create the uniform BGT, the data specifications as identified in Section 7.2 are outlined in models developed in ArcGIS ModelBuilder (see Appendix E). These models work on all test datasets as introduced in Chapter 5. In this chapter, the method of model building and the main results of the uniform BGT are explained.

7.3.1. MODEL

To execute the aggregation, the Dissolve tool is used to aggregate polygons, and the Unsplit Line tool is used to aggregate lines. These two tools aggregate objects based on selected attributes. Only these attributes will remain in the resulting dataset. The data specifications justify the attributes on which the aggregation is based. These required attributes are selected as Dissolve Fields. However, when executing the model, it turned out that not all Dissolve Fields are defined in the data specifications, because they are debatable. These attributes are *bgt-status* and *sourceholder.* The attribute *bgt-status* clarifies if the object exist in the physical environment or if the object is still under construction. Therefore, the choice should be made if this is important information to keep separate in the uniform BGT. Another choice is if the attribute *sourceholder* should be added as Dissolve Field. When adding *sourceholder* as Dissolve Field, the borders between the areas of source holders will fade and more objects will aggregate. Following IMGeo, both attributes should be added as Dissolve Field, and therefore, the decision is made to do so. However, the discussion, which could arise, will be further specified in Section 7.4.

After dissolving the objects, the data only consist of fields on which the aggregation is executed. To add the relevant IMGeo-Object attributes again, an overlay with the old data is implemented. Therefore, the Union tool is executed for polygonal objects, and the Intersect tool is used for line objects. These tools define all objects based on the map and combine those with the data of the original dataset. As specified in the data specifications, the aggregated objects should acquire a new *identificationLocalID, objectBeginTime,* and *timeOfRegistration.* The *identificationLocalID* is developed by adding the source holder, followed by a dot and then a new random ID of 32 characters. The random ID is created with a python script. The attributes *objectBeginTime* and *timeOfRegistration* are developed by adding the date and time of running the model as a VB-expression (see Table 7.1).

Table 7.1: The development of the attributes: *identificationLocalID, objectBeginTime* and *timeOfRegistration* with Python and VB expressions

Attribute	Python-Expression	VB-Expression
Identification LocalID	Def CalcGUID() import uuid return str(uuid.uuid4()).lower()	[*sourceholder*] & "." & [*identificationcode*]
objectBeginTime		Year(Date) & "-" & Month(Date) & "-" & Day(Date)
timeOfRegistration		Year(Date) & "-" & Month(Date) & "-" & Day(Date) & "T" & Time()

Source: Thoreleifson, 2010.

The method of this model is used to model every required object type and with the Dissolve Field as specified in the data specifications (with the addition of *bgt-status* and *sourceholder*). In some object types, the Dissolve tool does not only aggregate objects, but also split multipart objects. These objects are not neighboring objects, but have the same *identificationLocalID* in the BGT, which means that it can be considered as one object. In these object types, the Dissolve tool is executed twice. Firstly, with the required attributes as Dissolve Field, and secondly with the *identificationLocalID* as Dissolve Field and the Create Multipart Features selected. This ensures that the multipart polygons were combined again. In the test datasets, these multipart objects occurred only in the object types BUILDING, ENGINEERINGSTRUCTURE, and REMAININGSTRUCTURE. However, in other datasets, it might also occur in other object types. For now, this solution has been tested on the object types on which it occurs.

In addition, the model of the object types containing the 'not-bgt' attribute values differ slightly. Here a reverse selection of the 'not-bgt' attribute values is executed prior to the aggregation of the objects to exclude the 'not-bgt' attribute values in the uniform BGT.

7.3.2. RESULTS

The different models as developed in the previous section resulted in the aggregation of many different objects. All 'virtual borders' and optional IMGeo attributes were aggregated with their neighboring objects. The object types, which did not aggregate objects, did not contain neighboring objects with the same attribute values, or only contain one object in the dataset.

In some object types, many different objects aggregated into one object. However, this aggregation does also have a downside. When the dataset gets bigger, also the objects that were created are longer and bigger. For example, in PARTOFROAD, the roads were aggregated mostly on crossings and when that happens more often, the danger is that objects with very complex shapes will occur (see Figure 7.3). This would be a challenge for the visualization of those roads. One of the solutions could be to add a road network, in which the aggregation will be executed. But also in other object types, this trend of large and complex objects occurs (see Figure 7.4).

Figure 7.3: PARTOFROAD aggregated at the corners resulting in one large and complex feature - Dronten

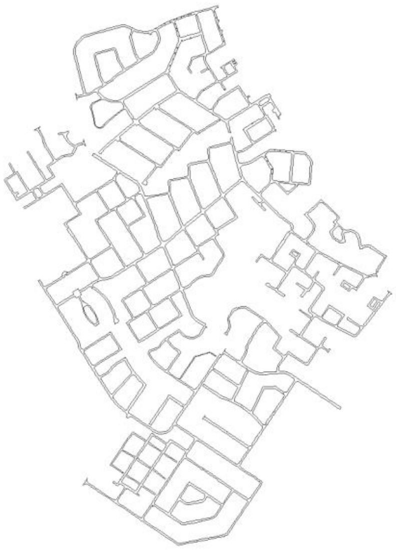
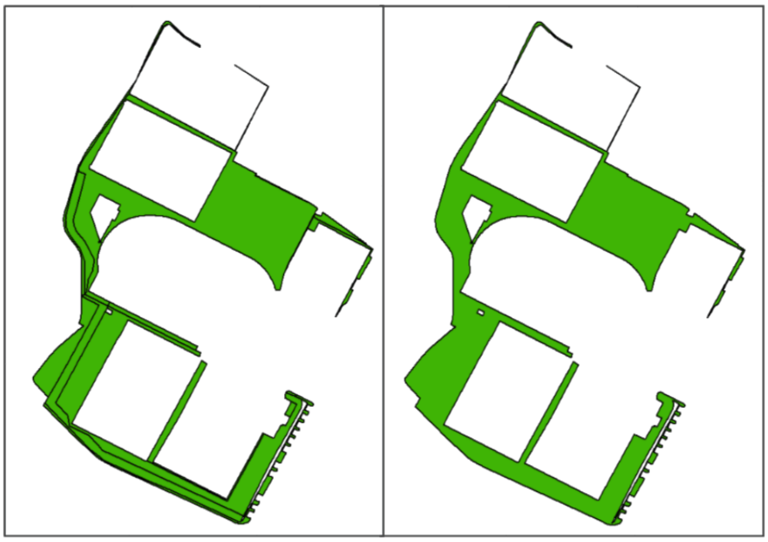


Figure 7.4: The aggregation of 17 features of COVEREDPARTOFTERRAIN into one object. The original state (left) in comparison with the aggregated state (right) - Maastricht



When looking at more detail to the results, it is remarkable that in the object type PARTOFROAD, still some 'speed bumps' can be recognized, although the optional attribute *plus-function* is eliminated. In Figure 7.5 and in Figure 7.6, these results are visualized. Figure 7.5 displays the 'speed bumps' as discussed in Section 7.1. These objects have the same required attribute values as their neighboring objects, i.e. 'roadway local road' (*function*) and 'open pavement' (*physicalOccurrence*) and is therefore aggregated as expected.

Figure 7.5: The 'speed bumps' with 'open pavement' in the original state (left), aggregated state (middle) and physical state as in Google Streetview (right)

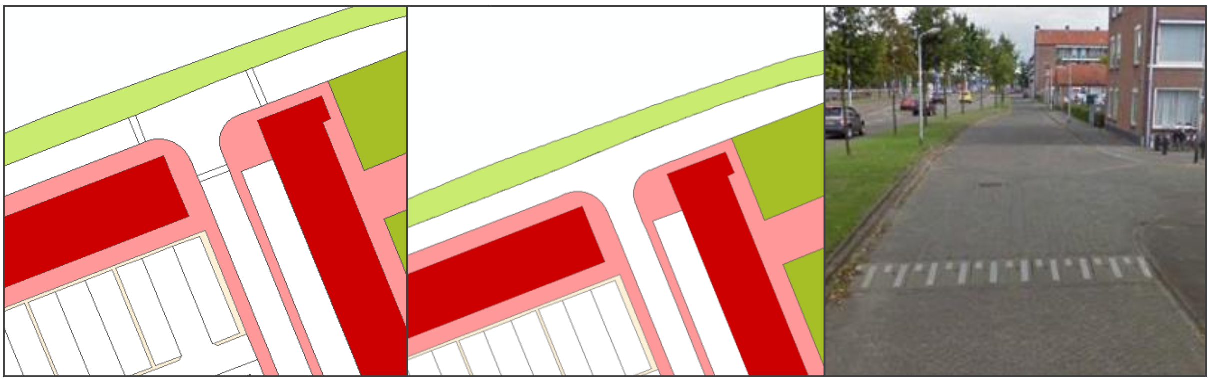
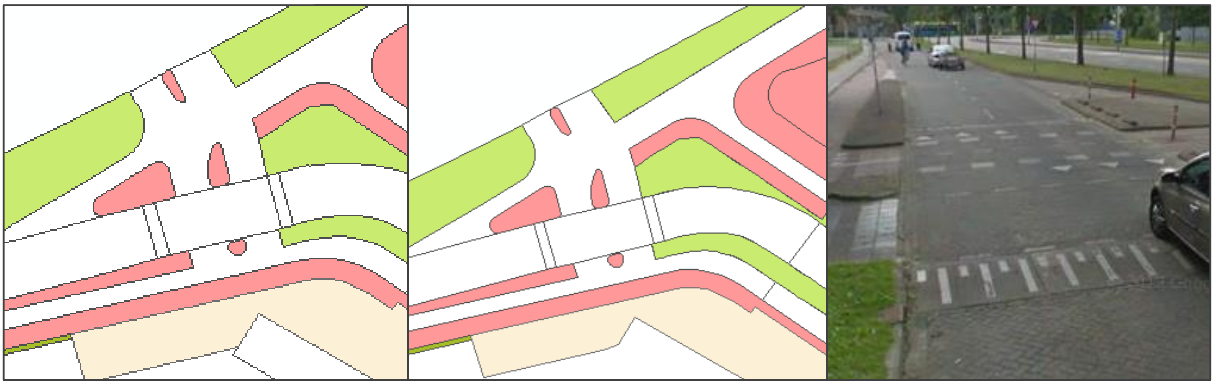


Figure 7.6 shows also the attribute values 'roadway local road' (*function*) in the neighboring objects, but the difference lay within the material of the roads. In this example, the 'speed bumps' are not aggregated, because the attribute *physicalOccurrence* of the 'speed bumps' consists of material which is difficult to remove ('closed pavement'), while their neighboring roads consist of easily removable material ('open pavement'). The data behind this object is still changed, because the attribute *plus-function*, which contains the optional IMGeo information that it was a 'speed bump' is not visible anymore.

Figure 7.6: The ‘speed bumps’ with ‘closed pavement’ in the original state (left), aggregated state (middle) and physical state as in Google Streetview (right)



Another remark is that, when running the model on the object type BUILDING, some objects in the Dronten dataset did aggregate, while it was initially not necessary to aggregate BUILDING (see Section 7.2). This mostly concerns new buildings, which are not yet available in the BAG, and therefore containing the same attribute value within the attribute *identificationOfBAGBuildings*. This is not a problem when using the BGT as preparation for the midscale BGT due to the fact that in the midscale BGT all neighboring buildings should aggregate, regardless their attribute value in *identificationOfBAGBuildings*. However, it is an important notice when using the uniform BGT as separate product.

Figure 7.7: Eliminated ‘niet-bgt’ objects in REMAININGSTRUCTURE (in yellow)

In Section 7.2, the concern was raised that ‘not-bgt’ objects create holes after eliminating them on ground level. After testing this issue, in two of the three datasets no problems occur. However, in the dataset of Maastricht, indeed many holes arise in the object type REMAININGSTRUCTURE (in yellow in Figure 7.7). In addition, in the input data of Maastricht, no extra information is available in the optional data, which makes it more complex to get knowledge what type of REMAININGSTRUCTURE it concerns and to find the appropriate solution for this kind of problems. Following Appendix C, these objects are probably of the optional type ‘bunker’, ‘feed silo’ or ‘shed’. Visually, it seems logical to add these objects to the object type BUILDING. However, they are not part of the BAG buildings and therefore cannot receive an attribute value in the required attribute *identificationOfBAGBuildings*. In TOP10NL, the solution for these buildings, which do not contain BAG identification, is to add it to the TOP10NL object type LAYOUTELEMENT (Kadaster, 2012). Therefore, the most conventional solution is to keep these objects as REMAININGSTRUCTURE in the uniform BGT and to remove these objects in the midscale BGT.

7.4. PHASE 4: ANALYSIS AND EVALUATION

After implementing the data specifications, the aggregation of the BGT resulted in three uniform test datasets, which contain the attributes and attribute values as required in the IMGeo documents. Only the BUILDINGS are not necessary to aggregate, because then the connection with the BAG can be retained. The aggregation of the Amersfoort dataset resulted in a total reduction of 47 objects which is a percentage of 10.8% of all objects in that dataset. In comparison, the Maastricht dataset reduces 18.6% and the Dronten dataset reduces 16.3% (see Table 7.2). However, if this is a good result is debatable. As outlined in Section 7.3.1, the objects resulted in very large objects with interesting shapes. Therefore, the choice should be made if there should be limitations on this aggregation, for example with a restriction in size.

Table 7.2: The amount of objects of the three datasets, before and after the aggregation into the uniform BGT

		Amersfoort dataset				Maastricht dataset				Dronten dataset			
Cycle	BGT Object type	Before	After	Difference	%	Before_	After_	Difference_	%_	Before.	After.	Difference.	%.
Roads	PARTOFROAD	94	75	-19	-20,2%	1043	740	-303	-29,1%	2187	1412	-775	-35,4%
	SUPPORTIVEPARTOFROAD	46	44	-2	-4,3%	151	147	-4	-2,6%	565	462	-103	-18,2%
	RAILWAY	4	4	0	0,0%								
Buildings	BUILDING	164	164	0	0,0%	2592	2592	0	0,0%	5023	4989	-34	-0,7%
Water	PARTOFWATER	1	1	0	0,0%	13	13	0	0,0%	798	760	-38	-4,8%
	SUPPORTIVEPARTOFWATER	1	1	0	0,0%	1	1	0	0,0%	93	61	-32	-34,4%
Nature	COVEREDPARTOFTERRAIN	51	51	0	0,0%	1053	839	-214	-20,3%	2147	1427	-720	-33,5%
	UNCOVEREDPARTOFTERRAIN	24	20	-4	-16,7%	329	308	-21	-6,4%	655	598	-57	-8,7%
B&T	PARTOFBRIDGE	5	5	0	0,0%	37	37	0	0,0%	78	51	-27	-34,6%
	PARTOFTUNNEL	1	1	0	0,0%								
Other	FUNCTIONALAREA	1	0	-1	-100,0%	42	0	-42	-100,0%				
	ENGINEERINGSTRUCTURE (polygon)	1	1	0	0,0%	5	5	0	0,0%	50	49	-1	-2,0%
	ENGINEERINGSTRUCTURE (line)	4	0	-4	-100,0%					118	0	-118	-100,0%
	REMAININGSTRUCTURE	7	1	-6	-85,7%	271	5	-266	-98,2%	2	2	0	0,0%
	SEPARATIONS (polygon)	7	7	0	0,0%	39	37	-2	-5,1%				
	SEPARATIONS (line)	23	12	-11	-47,8%	493	215	-278	-56,4%	3	1	-2	-66,7%
	UNCLASSIFIEDOBJECT	3	3	0	0,0%								
Total		437	390	-47	-10,8%	6069	4939	-1130	-18,6%	11719	9812	-1907	-16,3%

In general, the uniform BGT can be used as input for the midscale BGT, but the interesting part is also to consider this uniform BGT as entirely new product, which can be used as key registration on the 1:500 until a 1:5,000 scale. To specify this uniform BGT as national product, one of the key elements is who is going to be the source holder of the uniform BGT? At the one hand, it seems logical to change the source holder after aggregating the BGT, because other companies or people than the source holder have modified the source data and are therefore responsible for these new data objects. An effect of this new source holder is that the aggregation model will aggregate more objects, because it also aggregates objects, which contain the same attribute values, but are crossing borders of areas with different source holders. On the other hand, it seems more logical to remain the original source holder, because the source holder possesses the data and it should be logical that it remains there. Besides, not all objects are aggregated and these objects should stay the responsibility of the source holder, who developed the objects. In addition, it could be interesting for users of the uniform BGT to seen which source holder is responsible for which source data. The decision should be made in accordance with users, source holders and other stakeholders. When the original source holder is not designated to be source holder of the uniform BGT, the attribute *sourceholder* should not be part of the Dissolve Field. Also the *identificationLocalID* should be changed for every object, so that it contains the new source holder. And when the *identificationLocalID* has changed for every object, it is logical to see every object as entirely new object, and therefore, also the *objectBeginTime* and the *timeOfRegistration* have to be changed.

Another recommendation is to reconsider the content of the uniform BGT. The purpose of users to use the map is therefore of main importance. Based on logical thinking, the testing phase of the uniform BGT resulted in some strange results, which should be reconsidered by users if they really need these results modelled by following the requirements as developed in IMGeo. An example of these results is the elimination of the optional attribute value 'speed bump'. Now some 'speed bumps' are aggregated, because they had a similar physical appearance (*physicalOccurrence*) with neighboring roads. Other 'speed bumps' are still visible, because they had a different physical appearance. Users should decide whether or not the physical appearance of the roads is important and if they want to see the reason why these objects exist. Perhaps, the optional attribute value 'speed bump' should be maintained only when it did not aggregate, or maybe it is not important to see if it is a 'speed bump' as long as the physical appearance is noticed. Users should decide on this.

After these alterations, the uniform BGT can be used as separate key registration with as main purpose to have a simple, large scale key registration within the system of key registrations. The main usage of this uniform BGT will be for different geo-related tasks, which do not need extra optional information. However, it should be considered what effects the implementation of such a product has on the usage of other products in the system of key registrations. Likely, the usage of other key registrations, i.e. the BGT with optional objects, and the midscale data product will become less when adding a uniform BGT to the key registrations. In addition, this usage should be considered in cooperation with the upload times of every available product. Obviously, the product that will be updated more often will be acquired more.

THE MIDSCALE BGT

8. PHASE 1: IDENTIFICATION OF GLOBAL SPECIFICATIONS

In the first phase of the midscale BGT, the knowledge acquisition tool 'reverse engineering' will be used to get an impression of how the midscale BGT should look like and what the expected effects and differences are when replacing TOP10NL by a dataset derived from the uniform BGT. Therefore, TOP10NL will be analyzed and compared to the current available objects in the uniform BGT. This comparison will be made both visually and with further insight in the data of objects. In this phase, only the required object types, attributes and attribute values of both the BGT and TOP10NL will be included (see Section 4.5, and Appendix F). Therefore, the three developed test datasets of the uniform BGT will be used as input datasets (see Chapter 7). This phase will result in global specifications identifying the main expected consequences of creating a midscale data product from the uniform BGT. The generic rules as outlined in Chapter 6 still apply.

8.1. GENERIC DIFFERENCES

When comparing the required object types of the BGT with the required object types of TOP10NL, the BGT contains more required object types than TOP10NL. Obviously, the large scale of the BGT allows more object types to be visible, because more detail can be added. For example, the object types SUPPORTIVEPARTOFROAD and SUPPORTIVEPARTOFWATER contain both relatively small objects, which are not relevant in TOP10NL due to its scale. Therefore, these objects will be eliminated in the midscale BGT. In addition, the object types PARTOFBRIDGE and PARTOFTUNNEL are added to the BGT as separate object types, while in TOP10NL the bridges and tunnel are only recognized as objects with optional, supplementary attribute values in the object types PARTOFROAD, PARTOFRAILWAY, PARTOFWATER, and TERRAIN. The choice should be made how these objects should be modelled in the midscale BGT (see Section 8.6) (Brink et al., 2013; Kadaster, 2013).

When looking at the geometry of the object types, also some differences are recognizable, because of scale differences. The BGT object types mostly consist of polygonal geometry and TOP10NL mostly consist of multiple geometries, based on the size of objects (see Table 8.1). Due to these scale differences, many of the geometries need to be collapsed (Brink et al., 2013; Kadaster, 2013).

Table 8.1: The geometry of the required object types of the BGT and BRT's TOP10NL

BGT object types	BGT geometry	BRT object types	BRT geometry
PARTOFROAD	Polygon	PARTOFROAD	Point, line, polygon
SUPPORTIVEPARTOFROAD	Polygon	PARTOFRAILWAY	Point, line
RAILWAY	Line	BUILDINGCOMPLEX	Polygon
BUILDING	Polygon	PARTOFWATER	Point, line, polygon
PARTOFWATER	Polygon	TERRAIN	Polygon
SUPPORTIVEPARTOFWATER	Polygon	LAYOUTELEMENT	Point, line
COVEREDPARTOFTERRAIN	Polygon	RELIEF	Point, line
UNCOVEREDPARTOFTERRAIN	Polygon	REGISTRATIONAREA	Point, polygon
PARTOFBRIDGE	Polygon	GEOGRAPHICALAREA	Point, polygon
PARTOFTUNNEL	Polygon	FUNCTIONALAREA	Point, polygon
FUNCTIONALAREA	Polygon		
REMAININGSTRUCTURE	Polygon		
ENGINEERINGSTRUCTURE	Line, polygon		
SEPARATIONS	Line, polygon		
UNCLASSIFIEDOBJECT	Point, line, polygon		

Source: Brink et al., 2013; Kadaster, 2013.

8.2. ROADS

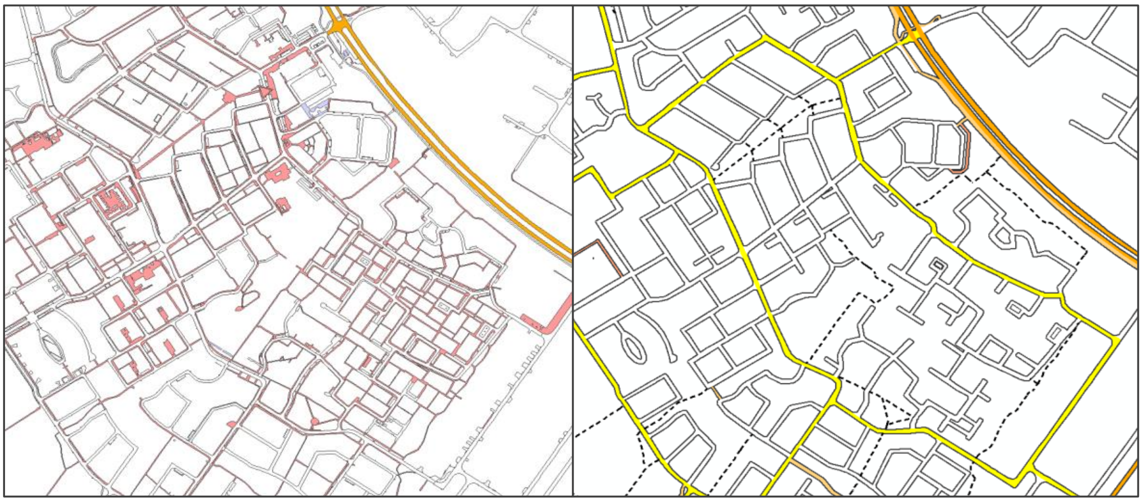
Visually, both the BGT and TOP10NL are very detailed with displaying its roads, meaning that almost every road in the physical area is visible. However, also some differences can be recognized. Firstly, the BGT displays roads only as polygonal objects, while TOP10NL uses both polygonal and line objects to display the roads. Secondly, next to these roads, in TOP10NL also centerlines and connection points, which indicate the corners of the roads, are available. And in the BGT, the additional object type SUPPORTIVEPARTOFROAD is defined with the attribute values 'verge' and 'traffic isle' (see Figure 8.1).

Figure 8.1: Visual comparison of the roads of the uniform BGT (left) with the roads of TOP10NL (right) in Maastricht

The main difference between the BGT and TOP10NL lies within the semantics of the roads. In TOP10NL, a differentiation is made between the different kind of roads (*typeOfRoad*) and its main function (*mainRoadUse*). In the BGT, this distinction is not made, but only the main function of the roads is outlined (*function*). While some attribute values within these attributes seem similar, these might not be similar at all, because the hierarchy of the roads is not similar defined. For example, 'roadway freeway' (BGT) is defined as roads, which contain the blue ANWB-signs, which specify that it is a freeway, and 'roadway regional road' (BGT) is defined as the roads without these blue ANWB-signs, but are connecting urban areas. In TOP10NL, the attribute values 'main road' and 'regional road' are specified. Both attribute values can contain the blue ANWB-signs, but the main difference lies within the connection between bigger urban places (i.e. 'main road') and smaller urban places (i.e. 'regional road'). Another example is that in the BGT 'roadway local road', 'cycle path', and 'footpath' are specified, while in TOP10NL the attribute values 'local road' and 'street' are used with additional attribute values within the attribute *mainRoadUse* to define the usage of this particular road (e.g. 'cyclists/mopeds', 'pedestrians', 'mixed traffic'). In the BGT it results in more objects with the attribute values 'roadway local road' in urban areas than in TOP10NL, because most local roads in urban areas are also accessible for cars and therefore in the BGT seen as local road and in TOP10NL seen as street. These objects in the BGT are often divided in 'roadway local road' and the parallel available 'cyclepath' or 'footpath' separately displayed, while in TOP10NL only one object is specified with *mainRoadUse* 'mixed traffic' (see Figure 8.2) (Brink et al., 2013; Kadaster, 2013).

When comparing the attribute *physicalOccurrence* in the BGT with *physicalOccurrence* in TOP10NL, it appears that the difference between 'closed pavement' and 'open pavement' has disappeared. In TOP10NL, the attribute value 'paved' outlines both attribute values. Furthermore, in TOP10NL some required attributes are specified, which are not specified in the BGT. These attributes are *yes/noSeparationOfLanes* and *typeOfInfrastructure*. These attributes cannot be specified from a BGT-perspective and a solution should be found when executing the fourth phase.

Figure 8.2: Objects with attribute value 'roadway local road' in the BGT (white roads, right), in comparison with the 'street' objects (white roads, left) and 'local road' objects (yellow roads, left) in TOP10NL

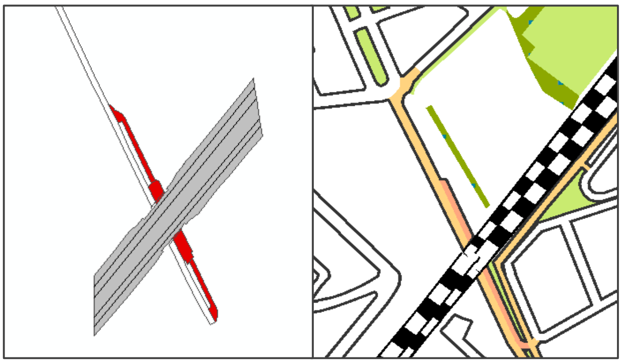


In the midscale BGT, the division should be made between polygonal objects and line objects, the objects of SUPPORTIVEPARTOFROAD should be eliminated, and a method should be found to create centerlines and connection points. In addition, the decision should be made about how to deal with the semantic differences within the attribute values. Because it is not easy to transform the BGT roads in such a way that it can equalize the exact semantics of the BRT, I made the decision to follow the BGT perspective and use the semantics of the BGT to generalize the roads into a midscale BGT. However, these semantic differences should be treated carefully when analyzing them in the current implementation rules of TOP10NL, because those are based on the semantics used in TOP10NL.

Railways

When comparing the object types RAILWAY (BGT) and PARTOFRAILWAY (TOP10NL), only a few differences within the data can be outlined. RAILWAY does only have the required attribute *function* with the attribute values 'train', 'metro', and 'tram', while PARTOFRAILWAY also distinguish 'mixed'. In addition, TOP10NL consists of the additional attributes *typeOfInfrastructure, widthOfRailway* and *numberOfTracks*, which cannot be derived from the BGT (see Figure 8.3).

Figure 8.3: Visual comparison of RAILWAY within the uniform BGT (left) with PARTOFRAILWAY of TOP10NL (right)



In addition, the polygonal objects of railways are differently located. In the BGT, the objects with attribute value 'railroad' are displayed as objects within the object type PARTOFROAD, while the related objects in TOP10NL with the attribute value 'railroad body' are outlined in the object type TERRAIN. Following the BGT perspective, I have chosen to keep the attribute value 'railroad' as part of the object type PARTOFROAD.

8.3. BUILDINGS

When comparing the buildings of the BGT with the buildings of TOP10NL, two major differences are noticeable. Firstly, the BGT geometries represent buildings in where they intersect the terrain, while TOP10NL draws buildings looking at the aerial view of the buildings; and secondly, the buildings in TOP10NL are not part of ground level, while the buildings of the BGT are (see Chapter 6). In TOP10NL, the object type BUILDINGCOMPLEX is located on top of the object type TERRAIN with the attribute value 'other'.

When comparing the differences of the BGT and TOP10NL with the BAG, it reveals that the BAG also consists of buildings seen from an aerial perspective. In Figure 8.4, these differences are made clear by a building with pillars. In the BGT, the pillars itself are specified, because they intersect the terrain, while in TOP10NL and the BAG, these pillars are not visible from an aerial perspective and therefore not separately outlined. In addition, the error of arcs (see Section 5.3) reveals that the buildings of the BGT are part of ground level, because holes occur due to these arcs.

Figure 8.4: Building with pillars reveals the geometry at surface level in the uniform BGT (first picture, left) and at aerial perspective in TOP10NL (second picture) and the BAG (third picture). The fourth picture (right) shows the physical appearance of the building as photographed in Google Maps

The main choice in this cycle is whether the buildings of the BGT should be adopted to generalize the midscale (and eventually extend it for the BAG) or to use the buildings of the BAG to generalize the buildings of the midscale BGT. In addition, the choice should be made whether or not the buildings should be part of ground level. For now, the decision is made that buildings will be created following the BGT-perspective, which means that the buildings of the BGT are used and also that the ground level for buildings will be maintained. The main reason for this decision is that many differences are expected when comparing the results of the buildings in the midscale BGT created from the BGT buildings with the current TOP10NL buildings. Therefore, the impact of the different choices can be revealed most. In the fourth phase, the impact of this choice will be analyzed.

When focusing on the generalization aspects, the difference between the BGT and TOP10NL is that in the BGT every single house is outlined with the attribute *identificationOfBAGBuildings*, which ensures the connection with the BAG. Due to scale changes in TOP10NL this attribute does not exist, and all connection houses and buildings closer than 2 meters are combined into building blocks. In addition, small houses are eliminated and the shape of buildings is tremendously simplified (see Figure 8.5).

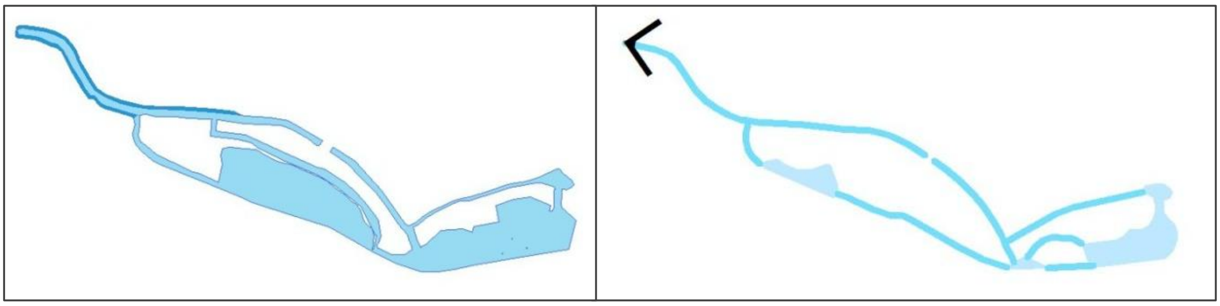
Figure 8.5: Differences between buildings in the uniform BGT (left), in TOP10NL (middle) and in the BAG (right) in Dronten

8.4. WATER

In the cycle water, similar aspects as in the cycle roads are noticeable. For example, in the BGT only polygonal geometry is outlined, while in TOP10NL a differentiation is made between polygonal geometry and line geometry. Noticeable is that a single polygonal object in the BGT can exist of more polygons and lines in TOP10NL, because sometimes the object is smaller than the required 6 meters and sometimes, the same object is wider than 6 meters (see Figure 8.6). Another similarity with the cycle roads is the object type SUPPORTIVEPARTOFWATER, which objects will be too small for the midscale dataset and therefore not visible in TOP10NL. In TOP10NL these objects are merged with the object types PARTOFWATER or TERRAIN (Stoter, 2009).

In the midscale BGT, a differentiation should be made between polygonal objects and line objects, and the objects of SUPPORTIVEPARTOFWATER should be merged with PARTOFWATER or TERRAIN.

Figure 8.6: Polygonal geometry of the cycle water in the uniform BGT (left) comparing with the multiple geometry objects in TOP10NL (right) in Amersfoort

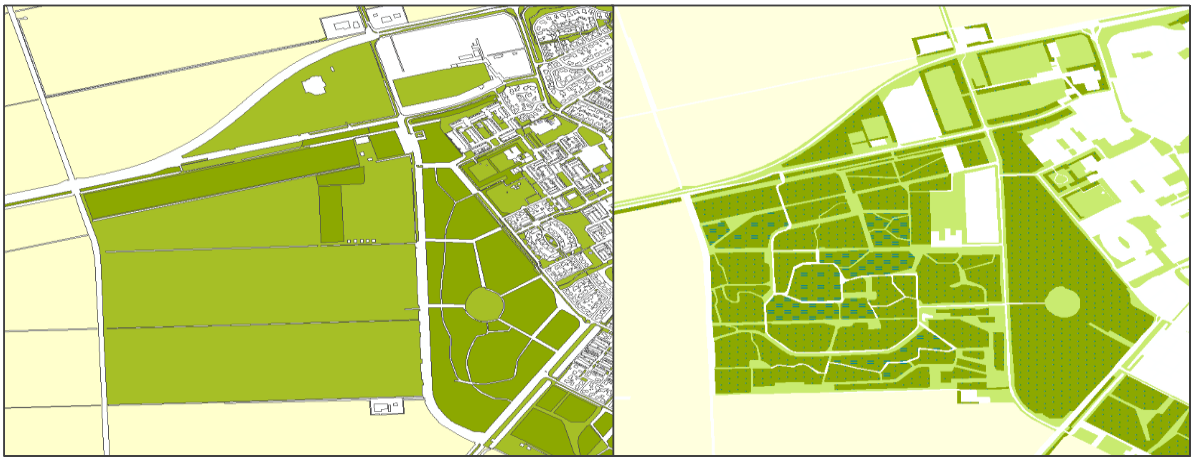


When comparing the required attributes of both datasets, it reveals that the attributes *mainDrain* and *occurrence* are not available in the required attributes of the BGT. Also, the division between the attributes *typeOfWater* and *function* is not specified in the BGT, which results in many attribute values, which are not (exactly) similar defined in both datasets. Comparable to the cycle roads, I made the decision to try to generalize the cycle water following the BGT-perspective as much as possible to reveal the differences and challenges in this cycle.

8.5. NATURE

In the BGT, the object types COVEREDPARTOFTERRAIN and UNCOVEREDPARTOFTERRAIN are used to define natural areas. In TOP10NL only the object type TERRAIN exists. At first sight, most attributes and attribute values seem different to each other, because most attribute values are named differently. Although most semantics indeed slightly differ from each other, there are also some similarities within their semantics. For example, ‘coniferous forest’ (BGT) is similar to ‘forest: coniferous forest’ (TOP10NL), and ‘grassland’ (TOP10NL) is a combination of ‘agrarian grassland’ (BGT) and ‘remaining grassland’ (BGT) (Stoter, 2009). In Figure 8.7, the differences are visually represented in the dataset of Dronten. Where the BGT only consist of the attribute value ‘greening’ within the object type COVEREDPARTOFTERRAIN does TOP10NL specify a lot more objects within the different attribute values. Another difference between the BGT and TOP10NL is that the attribute value ‘yard’ (BGT) in the object type UNCOVEREDPARTOFTERRAIN cannot specifically be found in TOP10NL’s TERRAIN. These objects are mostly shown in TERRAIN as objects with the attribute value ‘other’ (as which also the buildings are displayed, see Section 8.3).

Figure 8.7: Visual comparison of the terrain areas of the uniform BGT (left) with the terrain areas of TOP10NL (right) in Dronten



8.6. BRIDGES & TUNNELS

As already mentioned in Section 8.1, bridges and tunnels in the BGT are outlined with separate object types PARTOFBRIDGE and PARTOFTUNNEL, while bridges and tunnels in TOP10NL are not separately specified (see Figure 8.8 and Figure 8.9). In TOP10NL, the bridges and tunnels can be recognized in the object types PARTOFROAD, PARTOFRAILWAY, PARTOFWATER, and TERRAIN, containing a different *heightLevel.* In addition, the optional attribute *physicalOccurence* (TOP10NL) contains the attribute values to specify the bridges (with attribute values 'on the movable part of the bridge', 'on the fixed part of the bridge', or 'in tunnel'). In the BGT it is not recognizable whether or not the bridges are movable or fixed (Brink et al., 2013; Kadaster, 2013). In the midscale BGT, I made the decision to follow again the BGT-perspective and to keep the object types PARTOFBRIDGE and PARTOFTUNNEL. However, small bridges and tunnels should be eliminated.

Figure 8.8: Visual comparison of the bridges of the uniform BGT (left) with the bridges of TOP10NL (right) in Maastricht

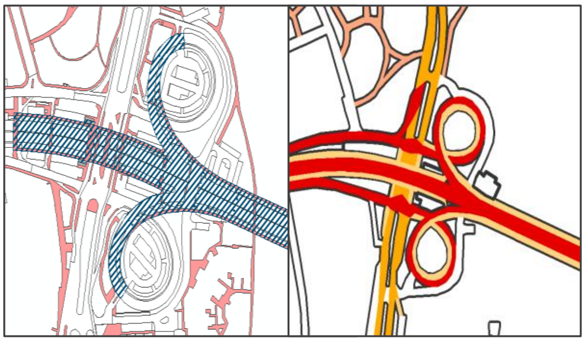
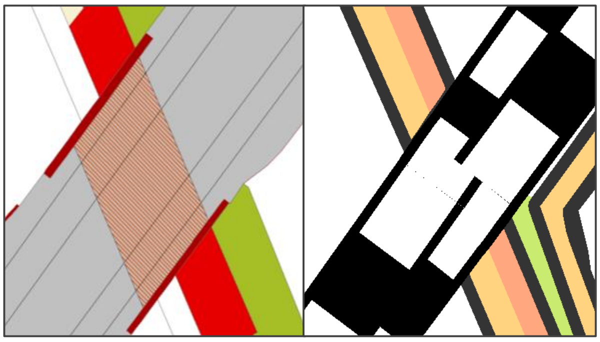


Figure 8.9: Visual comparison of the tunnels of the uniform BGT (left) with the tunnels of TOP10NL (right) in Amersfoort



8.7. OTHER

The attribute values within the object types ENGINEERINGSTRUCTURE, REMAININGSTRUCTURE and SEPARATIONS can be recognized within the object type LAYOUTELEMENT of TOP10NL. However, TOP10NL consists of 80 different required types of attribute values within the object type LAYOUTELEMENT, but only 11 attribute values can be potentially derived from the BGT (see Table 8.2) (Nagel, 2014; Stoter, 2009). The main decision in this cycle is which attribute values will be necessary to maintain in the midscale BGT and which attribute values will never be used. Following the research of Nagel (2014), the attribute values 'tree', 'tree line', 'hedge' and 'fence' are often named by users as most important. For now, the decision is made not to select which attribute values are probably interesting to maintain, but to generalize the information of the BGT into the midscale BGT. Visually, these BGT objects can be recognized as polygonal data or as line data, while LAYOUTELEMENT in TOP10NL often is recognized as line or point data. Therefore, this generalization should be made.

Table 8.2: TOP10NL attribute values, which can be potentially derived by the following BGT attribute values

TOP10NL attribute value (LAYOUTELEMENT)	Can be derived from BGT attribute value	Within BGT object type
Jetty	Jetty	ENGINEERINGSTRUCTURE
Tree line	Wooded bank	COVEREDPARTOFTERRAIN
Noise barrier	Noise barrier	SEPARATIONS
Bucking	Bucking	ENGINEERINGSTRUCTURE
Hedge	Wooded bank	COVEREDPARTOFTERRAIN
Fencing	Fence	SEPARATIONS
Power pole	Power pole	ENGINEERINGSTRUCTURE
Loading platform	Platform	ENGINEERINGSTRUCTURE
Wall	Wall	SEPARATIONS
Breakwater/crib	Breakwater	ENGINEERINGSTRUCTURE
Wind turbine	Wind turbine	REMAININGSTRUCTURE

Source: Nagel, 2014.

In addition, there are also required object types of TOP10NL, which are not mentioned, i.e. FUNCTIONALAREA, RELIEF, and GEOGRAPHICALAREA. Noticeable is that the objects of FUNCTIONALAREA (BGT) and FUNCTIONALAREA (TOP10NL) cannot be compared, because these object types contain very different attribute values. In the object type RELIEF, only the attribute value 'slope/height difference' can be derived from the BGT with the help of the slopes, which are defined as required attribute values in PARTOFROAD, COVEREDPARTOFTERRAIN, UNCOVEREDPARTOFTERRAIN and PARTOFWATER (Nagel, 2014). GEOGRAPHICALAREA is not available as required object type in the BGT. However, it might be possible to add this information from the optional BGT object type GEOGRAPHICALAREA.

9. PHASE 2: IDENTIFICATION OF DETAILED DATA SPECIFICATIONS

In the second phase, the data specifications will be determined in more detail based on the analysis of the implementation rules of TOP10NL. In the implementation rules of TOP10NL, many rules are outlined concerning the different cycles (Kadaster, 2012). It concerns rules, which are already mentioned in the first phase (see Chapter 8) and rules, which are concerning more specific information on how to create TOP10NL. In this chapter, the rules relevant for automatic generalization of the (uniform) BGT into a midscale BGT are gathered and converted into data specifications.

The template of Stoter et al. (2009a) is used to describe the data specifications. Therefore, the template 'constraints on one object' is used. In the sporadic cases that the data specifications consist of two or more kinds of objects, this is identified within the item **class**. In this template, the items **generic constraint ID, constraint type, geometry type, class, condition for object being concerned with this constraint, condition depends on initial value, condition to be respected, action,** and **exception** are identified as formal as possible. The **constraint type** will be identified when harmonizing the data specifications in Chapter 11. In the following sections, the generic data specifications usable for the entire midscale BGT product, and the more specific data specifications per cycle will be developed (see Appendix G).

9.1. GENERIC DATA SPECIFICATIONS

To create a new topographical key registration, the objects, which are considered as new objects, should receive a new *namespace, identificationcode, objectBeginTime,* and *timeOfRegistration* (see Section 6.2b). Due to scale changes and the sensitivity of the generalization, the decision is made to consider every object in the midscale BGT as new object. Therefore, after the generalization to the new scale, all objects receive this *namespace, identificationcode, objectBeginTime,* and *timeOfRegistration* (Altena et al., 2013; Brink et al., 2013). Because all objects in the midscale BGT are considered as entirely new objects, the source holder should also be changed in the midscale BGT. Perhaps, it might be interesting to know who the initial source holder of the BGT was. However, most often, it is not possible to recognize this source holder, because objects are combined or relocated. Therefore, the decision is made to change the source holder in the midscale BGT.

In the template, the same data specifications as in the uniform BGT are outlined (see Section 7.2), with the main exception that now all objects were changed, instead of only the aggregated objects (see Appendix G, Table G.1, generic constraint IDs Gen1 – Gen 1e).

Some implementation rules concern rules generic for different cycles. For example, a few rules are specified, which distinguish between built-up area and natural area. In the implementation rules, no conditions are outlined to specify when the area can be considered as built-up area (Kadaster, 2012). Therefore, trial and error should be used in the next phase, when implementing the data specifications to create this data specification.

Another generic concept is a 'classified road'. A classified road is defined in the implementation rules with the attribute values 'motorway', 'main road', and 'regional road' (Kadaster, 2012). Because the hierarchy of the BGT differs from the hierarchy of TOP10NL, the choice is made that the classified road of the BGT will be defined by the attribute values 'roadway motorway', 'roadway freeway', and 'roadway regional road' (see Appendix G, Table G.1, generic constraint ID Gen2).

9.2. ROADS

Many rules as outlined in the implementation rules of TOP10NL are concerned with the cycle roads. To create a more structured overview within these rules, I made a division of 10 arbitrary aspects to outline the main subjects of these rules. These 10 aspects are: (a) geometry; (b) SUPPORTIVEPARTOFROAD; (c) cycle paths; (d) footpaths;

(e) parking areas; (f) driveways; (g) crossings & centerlines; (h) dead-ends; (i) physical occurrence; and (j) additional rules. In appendix G, Table G.2, the developed data specifications can be found.

(a) Geometry

As already mentioned in Section 8.2, the scale of the product requires roads with polygonal geometry and roads with line geometry. Therefore, roads wider than 2 meters should be displayed as polygonal objects, and roads smaller than 2 meters should be outlined as line objects. The width of these roads should be measured without its verges (Kadaster, 2012). Therefore, the data specification is developed, which says that all objects in PARTOFROAD (**class**) measured without its verges (**condition for being concerned with this constraint**) with a width smaller than 2 meters (**condition to be respected**) should be changed into line objects (**action**) (see Appendix G, Table G.2, generic constraint ID RdsA1).

Because most of these polygonal objects are part of ground level, the conversion of these polygonal objects into line objects will result in holes. Therefore, these holes should be filled with neighboring objects.

(b) SUPPORTIVEPARTOFROAD

In the implementation rules of TOP10NL, the rules outline to which objects verges belong. A verge smaller than 6 meters and without a slope move to its adjacent object in PARTOFROAD when it consist of neighboring objects within the object types PARTOFROAD, BUILDING, or PARTOFWATER. When it does not consist of neighboring objects within those object types, the verge moves to the object type TERRAIN. Verges wider than 6 meters or smaller than 6 meters containing a slope move to its adjacent object in TERRAIN. Verges in between roads (i.e. 'traffic isles') move to TERRAIN when the verge is wider than 6 meters and the length is bigger than 50 meters, and to LAYOUTELEMENT when the verge is smaller than 6 meters and the length is bigger than 50 meters (Kadaster, 2012).

These rules are divided into five data specifications. The data specifications are based on the model generalization operator 'aggregation', because verges should be moved and aggregated with other objects than initially. The objects which have a width smaller than 6 meters and do not contain a slope (as defined in the **condition to be respected**) should be specified if they are neighboring to objects in PARTOFROAD, BUILDING, or PARTOFWATER. Therefore, this part is specified in the **condition for being concerned with this constraint**. The **action** specifies to which object types the verges should be moved (see Appendix G, Table G.2, generic constraint IDs RdsB1 – RdsB5).

(c) Cycle path

In the rules concerning cycle paths, three main divisions are made. Firstly, the division between cycle paths, which are situated parallel to another road, or cycle paths, which are situated freely (in Dutch: 'vrijliggend'). Secondly, the division is made between cycle paths within built-up areas and outside built-up areas. Thirdly, cycle parts behave differently when they are wider than 2 meters comparing to roads smaller than 2 meters. In Table 9.1, the different behaviors are explained.

Table 9.1: Rules about the bicycle paths within the cycle roads

RdsC1	'cycle path'	Parallel	>2m	Outside built-up area	Display as polygonal object
RdsC2	'cycle path'	Parallel	>2m	Inside built-up area	Display if parallel to a classified road
RdsC3	'cycle path'	Parallel	<2m	Outside built-up area	Display as line object
RdsC4	'cycle path'	Parallel	<2m	Inside built-up area	Do not display
RdsC5	'cycle path'	Freely	>2m	-	Display as polygonal object
RdsC5	'cycle path'	Freely	<2m	-	Display as line object

Source: Kadaster, 2012.

In the template, these rules are outlined as data specifications. Therefore, the **condition for object being concerned with this constraint** is set on if the data specifications should be situated parallel or freely and the **condition to be respected** concern whether the width is bigger or smaller than 2 meters and whether the object is inside or outside built-up area. The **actions** show how the objects should be displayed (See Appendix G, Table G.2, generic constraint IDs RdsC1 – RdsC5).

(d) Footpath

Most footpaths should be eliminated in the midscale BGT. Only the freely situated footpaths with a length bigger than 100 meters should be displayed. Footpaths, which are situated parallel towards another road, should not be displayed, unless they are for a relatively short distance neighboring a non-classified continuous road or when they are important for the neighborhood in which the road is situated. These footpaths cannot be smaller than 250 meters. However, it is difficult to determine whether or not a footpath is important for the neighborhood. In addition, footpaths around buildings or building blocks should not be displayed (Kadaster, 2012).

In the template, the rule that 'important' footpaths should stay when they are smaller than 250 meters will not be specified, because it is very difficult to automatically decide which footpaths are important and which are not. The other rules are specified in the template. The data specification which says that footpaths should not be displayed unless they are for a relatively short distance neighboring a non-classified road is specified with the **condition for being concerned with this constraint** set on 'parallel' and the **condition to be respected** set on 'part of continuous roads AND neighboring a non-classified road for a relatively short distance'. However, the exact parameter should be developed in the third phase. The **exception** is when these roads are smaller than 250 meters. These should not be displayed.

'Footpath on stairs' is only displayed when the length is bigger than 100 meters. When 'footpath on stairs' smaller than 100 meters consists of a neighboring footpath, the combination of this length will be used and the footpath rules will be applied. Otherwise, the 'footpath on stairs' will be eliminated. In addition, only extended 'pedestrian areas' will be displayed. An example of these extended 'pedestrian area' is the walking area within a city center. However, the decision about what is exactly the definition of 'extended' can be part of discussion (Kadaster, 2012).

(e) Parking areas

Parking areas bigger than 1000m² will be displayed with the exception of parking areas surrounded by forested areas and of orientation value (as specified in **exception**). Parking areas smaller than 1000m² will be added to adjacent objects within PARTOFROAD or TERRAIN. Roads on parking areas are only displayed when they are continuing at the other side of the parking area (Kadaster, 2012).

(f) Driveways

Driveways should not be identified as separate road type. Therefore, the driveways should be added to its adjacent PARTOFROAD with the attribute values of the neighboring road, which is highest in hierarchy. Therefore, the **action** within the data specification is set on 'move to adjacent road type highest in hierarchy'.

Driveways located on a dike should be longer than 100 meters to display, also when both in-driveways and exit-driveways are connected (Kadaster, 2012). Therefore, the **condition to be respected** is specified as 'located on a dike AND length > 100m'.

(g) Crossings & centerlines

Crossings and roundabouts should be mentioned as separate objects within the object types PARTOFROAD with attribute values 'crossing' and 'connection' in the attribute *typeOfInfrastructure*. These crossings contain multiple geometries with point data of the crossings, as already mentioned in Section 8.2.The middle of roundabouts should always be displayed no matter what size it is (Kadaster, 2012).

In addition, every road object should get its own centerline, with the same attributes and attribute values as its underlying polygonal object type (Kadaster, 2012). The data specification developed for this rule is based on the model generalization operator collapse, because centerlines should be created out of polygons. The **condition to be respected** is set on 'all roads with the same attributes/attribute values as its polygonal object'. And the **action** will be 'create centerlines.

(h) **Dead-ends**

Most dead-ended roads should be longer than 100 meters to be displayed in TOP10NL. However, there are a few exceptions: dead-ended roads with attribute value 'half-paved' and dead-ended roads behind buildings should not be displayed when smaller than 250 meters; and roads smaller than 100 meters ending on a parking area should be displayed. These rules are specified as **exception** of the data specification. Another exception is that dead-ended roads with the attribute value 'local road' should not be bigger than 250 meters. However, because of the differences in semantics, this will be a problem when generalizing the BGT following the BGT-perspective (see Section 8.2). Therefore, the standard value of 100 meters will be used to eliminate the roads with attribute value 'roadway local road' in the BGT (see Chapter 10) (Kadaster, 2012).

(i) **Physical occurrence**

As specified in Section 8.2, but not specified in the implementation rules of TOP10NL, because the differentiation is not made in TOP10NL, the attribute values 'closed pavement' and 'open pavement' should be combined into the object type 'paved'. As already seen while developing the uniform BGT (see Chapter 7), the division of those two attributes results in very small objects, which are not necessary to mention in a midscale BGT.

(j) **Additional rules**

In addition, some other rules are outlined in the implementation rules of TOP10NL. These rules concern specific attribute values like 'bridle path' or 'bus lane'. Also rules about the TOP10NL object type PARTOFRAILWAY are outlined. These rules are specified in Table 9.2 (Kadaster, 2012) (see Appendix G, Table G.2, generic constraint IDs RdsJ1 – RdsJ10).

Table 9.2: Additional rules within the cycle roads

RdsJ1	'bridle path', smaller than 2 meters should not be displayed
RdsJ2	'bus lane' integrated within a road should not be displayed separately.
RdsJ3	'bus lane' with its own road, should be displayed separately.
RdsJ4	'bus lane', which is closed with a specific barrier, should not be displayed.
RdsJ5	PARTOFRAILWAY should consist of centerlines with the same attributes/attribute values as its polygonal object in PARTOFRAILWAY
RdSJ6	PARTOFRAILWAY changing tracks over smaller than 5km constant will not be displayed
RdsJ7	PARTOFRAILWAY temporary tracks with a separate trace parallel to other traces should be displayed
RdsJ8	PARTOFRAILWAY both 'single trace' and 'double trace' with a length smaller than 500 meters should not be displayed
RdsJ9	PARTOFRAILWAY 'switch' within tracks should not be displayed
RdsJ10	PARTOFRAILWAY tracks situated on a dike should be mentioned by adding relief lines

Source: Kadaster, 2012.

9.3. BUILDINGS

As already specified in Section 8.3, buildings should be combined when neighboring to other buildings and not directly neighboring buildings within a distance of 2 meters should be combined. Exceptions are when buildings are bounded by objects of PARTOFROAD or PARTOFWATER (Kadaster, 2012). This rule is specified in the data specifications as both an aggregation constraint, which specifies the object type BUILDING (**class**) neighboring to other buildings (**condition to be respected**) should be combined (**action**); and as amalgamation constraint, which specifies that objects of the object type BUILDING (**class**) not directly neighboring to other objects, but within a distance of 2 meters (**condition to be respected**) also should be combined (**action**) with the exception of objects bounded by object types PARTOFROAD or PARTOFWATER (**exception**) (see Appendix G, Table G.3).

Another rule is that small buildings should be eliminated. Following the implementation rules of TOP10NL, the buildings should not be bigger than 3x3 meters or have a diameter smaller than 4 meters. Also, buildings in built-up areas, which cannot be recognized from access roads, will be removed when smaller than 50m². Patios and courtyards should be eliminated when smaller than 1000m² (Kadaster, 2012).

Finally, buildings should be refined. Sheds do not need to be part of the buildings and corridors, sky bridges, expansions etc. will only be outlined when they are bigger than 3 meters or with an area bigger than 3x3 meters. Also openings within buildings smaller than 3 meters or which are not public available will be added to the buildings (Kadaster, 2012).

The remaining rules in the implementation rules of TOP10NL consider the *overpass* of buildings or the different functions of buildings. The attribute *overpass* does not count for objects in the BGT, because the subsurface of the BGT objects should not overlap with other objects. The different functions of buildings are not obligated in TOP10NL and are also not obligated in the BGT and therefore not included in these data specifications (Brink et al., 2012; Kadaster, 2013).

9.4. WATER

As already outlined in Section 8.4, SUPPORTIVEPARTOFWATER should be combined with the object types PARTOFWATER or TERRAIN. Therefore, the decision is made to add 'shore/ditch' to the objects of PARTOFWATER and to add 'silt' to the objects of TERRAIN. In addition, the water objects should be divided in polygonal objects (wider than 6 meters) and line objects (smaller than 6 meters) (Kadaster, 2012). Because objects can be split into multiple objects with multiple geometry types, the **condition for being concerned with this constraint** is set on 'objects can be split into multiple polygon and line objects' (see Appendix G, Table G.4).

With regards to the size of water objects, the following rules apply: 'trenches' and 'ditches' are only displayed when they are bigger than 50 centimeters. 'lakes', 'tanks' and 'swim basins' are only displayed when they are bigger than 50m². Due to the fact that most attribute values do not exist in the BGT, the attribute values 'water plain' (BGT) will be used to specify 'lakes' (TOP10NL), and 'trench' (BGT) will specify the 'trenches' (TOP10NL). However, another rule in the implementation rules apply that all polygonal water objects should be bigger than 50m². Therefore, this rule should be treated carefully when implementing it in an automatic generalization system.

9.5. NATURE

In this cycle, the object types COVEREDPARTOFTERRAIN and UNCOVEREDPARTOFTERRAIN will be merged. Following the implementation rules of TOP10NL not many rules can be applied to generalize the BGT into a midscale BGT based on the cycle nature, but some rules are relevant. These rules concern the attribute value 'yard'. For example, a forest within a yard, or within built-up areas should be bigger than 1000m². In the template, this specification is specified with **class** set on 'forests within COVEREDPARTOFTERRAIN' and the **condition for being concerned with this constraint** set on within 'yard' or within built-up areas'. Small roads towards yards should be combined with 'yards' (Kadaster, 2012).

In addition, the minimum size of terrains is discussed. Terrain bordered by 'hard topography' (i.e. terrain objects bordered by objects within the object types PARTOFROAD or PARTOFWATER) cannot have a minimum size, because holes cannot be filled when eliminating these objects. Terrain bordered by 'soft topography' (i.e. terrain objects bordered by neighboring terrain objects) should only be mentioned when bigger than 1000m² (Kadaster, 2012). Therefore, the data specification is developed which specifies that the terrain objects, bordered by soft topography smaller than 1000m² should be combined with neighboring terrain objects. The terrain objects bordering hard topography is seen as an **exception** of this rule (see Appendix G, Table G.5).

Because of the division in hard topography and soft topography, this cycle can be easily used to repair the ground level of the midscale BGT. When not otherwise indicated, the holes will be filled with the neighboring terrain objects. However, this part should be executed after the cycle 'other', because first some polygons of the object types ENGINEERINGSTRUCTURE, REMAININGSTRUCTURE, and SEPARATIONS at ground level should be removed.

9.6. BRIDGES & TUNNELS

In Section 8.6, the decision is made to keep the bridges and tunnels following the BGT-perspective. This means that the object types PARTOFBRIDGE and PARTOFTUNNEL will be maintained, instead of adding the bridges and tunnels as attribute values to the midscale BGT. Therefore, the bridges and tunnels, which are too small, should be eliminated. This means that bridges or tunnels crossing line objects should not be shown and that bridges and tunnels, which are not connected anymore with road objects or water objects (in case of aqueducts), because they are eliminated enduring the generalization, will be eliminated.

The data specifications, which are developed, mention both object types separately. Therefore, both object types contain the data specification, which has the **condition to be respected** 'crossing line objects of PARTOFROAD or PARTOFWATER' with the **action** 'do not display'. And both object types contain the data specification which has the **condition to be respected** 'bridges/tunnels not connected to PARTOFROAD or PARTOFWATER' with the **action** 'do not display' (see Appendix G, Table G.6).

9.7. OTHER

As outlined in Section 8.7, the attribute values in the cycle other will not be changed into TOP10NL attribute values in the midscale BGT. The only aspect that needs to be executed is that polygonal objects in this cycle will be changed into point or line objects due to the scale aspects. Therefore, polygonal objects within the object types ENGINEERINGSTRUCTURE and REMAININGSTRUCTURE will be changed into point objects. And polygonal objects within the object type SEPARATIONS will be changed into line objects and merged with the already existing line objects in the BGT object type SEPARATIONS. Therefore, the data specifications are developed, based on the generalization operator collapse, because the geometry will be changed from polygon into another type (see Appendix G, Table G.7, generic constraint IDs Oth1 – Oth4).

10. PHASE 3: IMPLEMENTATION OF DATA SPECIFICATIONS

In this phase, the data specifications as developed in Chapter 9 will be tested by implementing the specifications in ArcGIS models. These models are displayed in Appendix H. The data specifications will be implemented per cycle and tested on unclear specifications. These unclear specifications will be improved where necessary. Noticeable is that this phase is a repetitive testing phase of the data specifications and that in every cycle improvements can be made to optimize the results, data specifications, process, and performance.

10.1. GENERIC MODELS

In the data specifications in Chapter 9, sometimes rules are specified, which concern objects in multiple cycles. Therefore, some general models are developed, which can be applied every time a cycle needs it. This section is organized as follows. Firstly, (a) the distinguishing between built-up area and natural area is discussed. Secondly, (b) the model is outlined which translates objects with polygonal geometry into objects with line geometry. And finally, (c) the model is discussed, which changes every object into a new midscale BGT object with the new formal key registration requirements.

(a) Built-Up Area

In the previous chapter, the importance of a specification distinguishing between built-up area and natural area is outlined. To specify which area of the map is built-up area, the Delineate Built-Up tool is used. This tool creates polygons based on the clustering of input buildings (BUILDING_uniform). Because the parameters of when it can be considered as built-up area is not specifically outlined in the implementation rules of TOP10NL (see Section 9.1), I have chosen for a Grouping Distance of 130 meters and a minimum area of 800,000m² specified in the Make Feature Layer tool (see Appendix H, Figure H.1). After some trial and error, these parameters turn out to give the desired visual results in the dataset of Maastricht and Dronten (see Figure 10.1). The dataset of Amersfoort seems too small to specify built-up area. However, the physical environment of the Amersfoort dataset is completely inside built-up area. Therefore, the choice is made to see the entire dataset as built-up area and to verify the models when applying them on the Amersfoort dataset to create the best results.

Figure 10.1: Results of the model to specify Built-up area in Dronten

These parameters are of main importance to add to the developed data specifications, because otherwise, it is not clear how the built-up area is specified. Therefore, these parameters are used to describe the new data specification (see Table 10.1). However, these parameters are based on visual rationalities, based on only two datasets, and can be altered when necessary.

Table 10.1: Additional rule concerning built-up areas

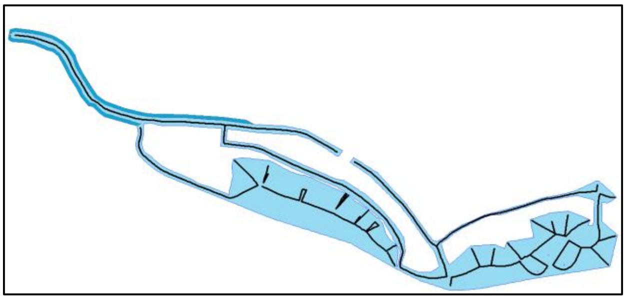
Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
Gen4		Polygon	PAND (Built-up area)	-	Yes	Grouping Distance = 130m AND Minimum Area = 800,000m²	Specify built-up area	-	Specified based on the datasets of Dronten and Maastricht.

(b) Polygon To Line

Another challenge, which is specified in the previous chapter, is that polygonal objects should be transformed into line objects. Therefore, a model, available at the ArcGIS Resource Center, was used to create centerlines (ESRI, 2011) (see Appendix H, Figure H.2). Then, this model is used as input for another model, which changes the centerlines to create a more smoothened and logical 'network' containing the data of the input. Therefore, the created centerlines were smoothened and smaller end lines were eliminated. Finally, the lines were extended to be linked with other lines when they are within a distance of 3 meters from each other (see Appendix H, Figure H.3).

It resulted in smoothened lines, which were not always linked to each other (see Figure 10.2). These lines need a lot more adaptations before they connect into a line network. However, the lines are of such quality that it can show how the midscale BGT will look like in comparison with TOP10NL, and therefore, these extra adaptations became out of the scope of this study.

Figure 10.2: Results of smoothened lines in Amersfoort



(c) Formal Key Registration Requirements

Finally, the identification of objects should be changed in all object types. Therefore, the general model is developed, which changes the *sourceholder, namespace, identificationLocalID, objectBeginTime,* and *timeOfRegistration* (see Appendix H, Figure H.4). This model is added after the generalization of every object into the new scale. In Table 10.2, the calculations of these attributes are outlined.

Table 10.2: The development of the attributes: *sourceholder, namespace, identificationLocalID, objectBeginTime,* and *timeOfRegistration* with Python and VB-expressions

Attribute	Python-Expression	VB-Expression
Sourceholder		"BGT10"
Namespace	-	"NL.BGT10"
identificationLocalID	Def CalcGUID() import uuid return str(uuid.uuid4()).lower()	[*sourceholder*] & "." & [*identificationcode*]
objectBeginTime	-	Year(Date) & "-" & Month(Date) & "-" & Day(Date)
timeOfRegistration	-	Now()

Source: Thoreleifson, 2010.

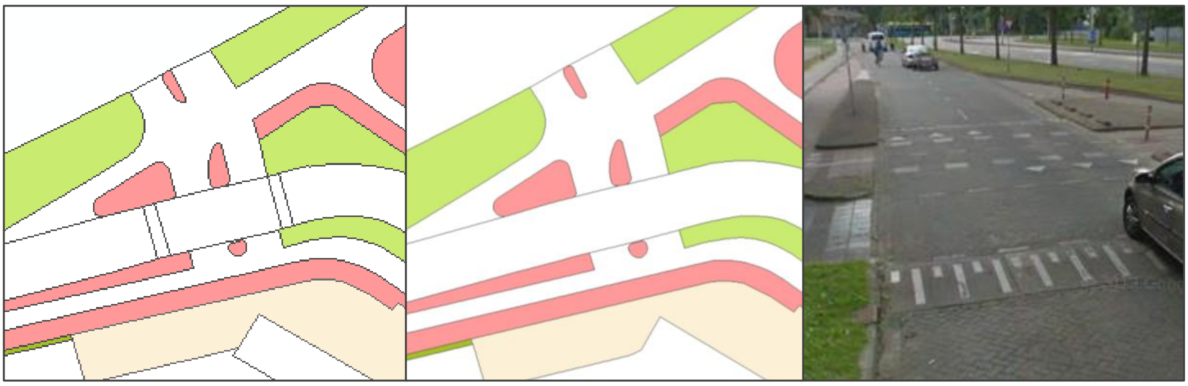
10.2. ROADS

To test the data specifications for the cycle roads as developed in Section 9.2, the same arbitrary aspects are used. However, when applying the model, it occurred to be better to implement those rules in another sequence to get a more optimal behavior of the model. Therefore, the aspects will be outlined in the following sequence: (a) physical occurrence; (b) parking areas; (c) driveways; (d) footpaths; (e) geometry; (f) SUPPORTIVEPARTOFROAD; (g) cycle paths; (h) dead-ends; and (i) crossings & centerlines. The additional rules have not been tested.

(a) Physical occurrence

To combine the attributes 'closed pavement' and 'open pavement', the objects with both attribute values are selected and recalculated into 'paved'. Then, a Dissolve is executed to combine the roads with these attribute values. It resulted in the aggregation of all remaining 'speed bumps' as discussed in Chapter 7 (see Figure 10.3) (see Appendix H, Figure H.5).

Figure 10.3: Dataset without 'speed bumps' (middle) due to the combination of 'closed pavement' and 'open pavement' into 'paved' in Amersfoort



(b) Parking areas

To move the parking areas smaller than 1000m^2 to a neighboring object in PARTOFROAD or TERRAIN, the object types COVEREDPARTOFTERRAIN, UNCOVEREDPARTOFTERRAIN, and PARTOFROAD are merged and the tool Polygon Neighbors is used to determine all neighboring objects within these object types. Then, all objects smaller than 1000m^2 are selected and added to the neighboring object type based on the longest adjacent connection. These objects receive the required attribute values of this neighboring object, and then a Dissolve is executed to aggregate the neighboring objects with the same attribute values (see Appendix H, Figure H.6).

(c) Driveways

To add the driveways to the highest classified road, the hierarchy of the roads needs to be specified. This is achieved with the help of numbers (the smaller the number, the higher in hierarchy). This is easy to automate, but it requires many different actions in ArcGIS (Add Field – Select Layer by Attribute – Calculate Field repetitive for every specific number in the hierarchy). Therefore, to fasten the tests for automatic generalization, this is executed with an additional table in Excel. In this Excel, the roads were numbered based on '[*function*].[*physicalOccurrence*]'. Then, the neighboring roads of the driveways were outlined with the Polygon Neighbors tool and the resulting neighboring roads were joined with the hierarchy of the roads to specify which neighboring road has the highest classification (see Appendix H, Figure H.7).

(d) Footpaths

Firstly, footpaths smaller than 100 meters were eliminated. Officially, the data specifications mention that only freely situated footpaths smaller than 100 meters should be eliminated. However, due to the fact that most parallel footpaths are also eliminated and the remaining footpaths should be longer than 250 meters, it is easier to start with all footpaths.

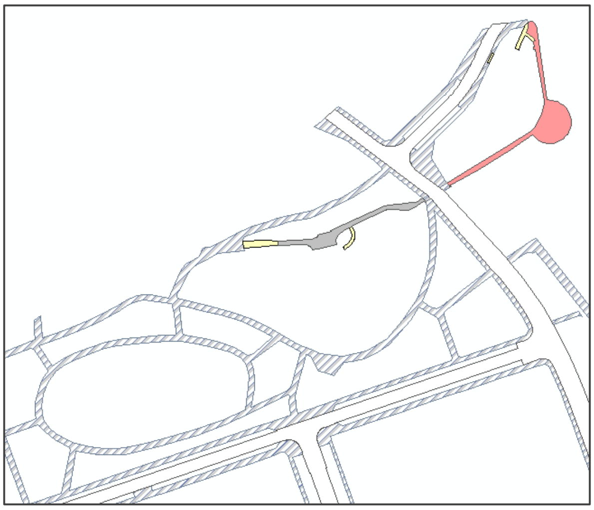
To specify whether the footpaths are situated parallel to a non-classified road, the hierarchy of the roads as outlined in driveways (see c) is reused. The footpaths, which are neighboring a non-classified road, are remained in the dataset, provided that they are longer than 250 meters. The footpaths neighboring a classified road are only remained in the dataset when they are not more than 20 meters parallel to the road. In addition, 'footpath on stairs' smaller than 100 meters and pedestrian areas smaller than 1000m² were eliminated (see Appendix H, Figure H.8).

When comparing the model with the data specifications, it occurs that most data specifications were very unclear due to lines in the implementation rules specifying *important* footpaths, *over a relatively short distance*, and *extended* pedestrian areas. The parameters, which are identified during the implementation will be added to the data specifications to create stronger data specifications.

The model resulted in the elimination of many different objects (see Figure 10.4). However, sometimes too many objects were eliminated, because in this model, the 'footpaths' and 'footpaths on stairs' were not combined as outlined in the data specifications. Therefore, also the 'footpaths on stairs' smaller than 100 meters were eliminated when they were neighboring 'footpaths' longer than 100 meters. Therefore, this model should be extended by combining these attribute values 'footpaths' and 'footpaths on stairs' before the elimination of the footpaths starts.

The elimination of the footpaths and pedestrian areas results in holes in the dataset. Because it is not yet clear where these footpaths should go to, these holes will be fixed at the end of the generalization (see Section 10.8).

Figure 10.4: Results after eliminating the footpaths, the eliminated paths are visualized in stripes, in yellow, the eliminated 'footpath on stairs'



(e) Geometry

To specify the width of the roads, a buffer of -1 is created to specify which roads are wider than 2 meters and which roads are smaller than 2 meters. The roads, which intersect with the buffer are wider than 2 meters and should remain as polygonal objects. The remaining roads should become line objects. To change the polygonal objects smaller than 2 meters, the model 'Polygon to Line' is used, as developed in Section 10.1b. How to deal with the resulting holes is further outlined in Section 10.8 (see Appendix H, Figure H.9).

The approach of using buffers to determine the width of an object differs from the approach described in TOP10NL, because it determines the maximum width instead of the mean width of the roads. Therefore, some differences occurred between the midscale BGT and TOP10NL (see Section 11.1.2)

(f) SUPPORTIVEPARTOFROAD

Objects in SUPPORTIVEPARTOFROAD wider than 6 meters or on a slope should be added to TERRAIN and objects in SUPPORTIVEPARTOFROAD smaller than 6 meters and without a slope should be added to its neighboring object in PARTOFROAD. To specify the width of SUPPORTIVEPARTOFROAD, again a buffer is executed with a parameter of -3. Then, the objects with an attribute value 'true' in the attribute *onSlope* are selected. These two types of objects are merged and moved to the object type TERRAIN. The remaining objects of SUPPORTIVEPARTOFROAD (smaller than 6 meters and without a slope) are specified. These objects are merged with the object type PARTOFROAD and moved to its neighboring road based on its maximum length. However, it also occurred that an object in SUPPORTIVEPARTOFROAD did not consist of a neighboring road, because these roads are already eliminated in previous models. These objects are merged with the object types, which will be moved to the object type TERRAIN.

In this model, no exclusion has been made between 'verges' and 'traffic isles', which means that all objects smaller than 6 meters and with a length bigger than 50 meters are selected and moved to the object type

LAYOUTELEMENT (see Section 10.7). Finally, a Dissolve is executed to combine the objects of PARTOFROAD with the added objects of SUPPORTIVEPARTOFROAD (see Appendix H, Figure H.10).

(g) Cycle paths

Cycle paths wider than 2 meters need two actions. Firstly, cycle paths within built-up areas should be remained when they are neighboring classified roads. Therefore, the neighboring roads are specified with the tool Polygon Neighbors, and the model Built-Up Area (as specified in Section 10.1a) is used to determine the cycle paths within built-up areas. Secondly, the remaining cycle paths outside built-up areas should be maintained. With the help of the first part of the model, these objects were already added to the dataset again. However, for testing purposes, the cycle roads outside built-up areas are selected and checked manually if they remained in the dataset, which they did (see Appendix H, Figure H.11).

Cycle paths smaller than 2 meters, should only be displayed when situated outside built-up area. Therefore, again the Built-Up Area tool is used to determine which line objects with attribute value 'cycle path' are inside built-up areas and which are situated outside built-up areas. The cycle paths outside built-up areas are remained in the dataset (see Appendix H, Figure H.12).

In this model, I did not make the distinction between parallel cycle paths and freely situated cycle paths. Therefore, not all freely situated cycle paths are remained in the dataset. However, the results should display if this is really necessary to add.

(h) Dead-ends

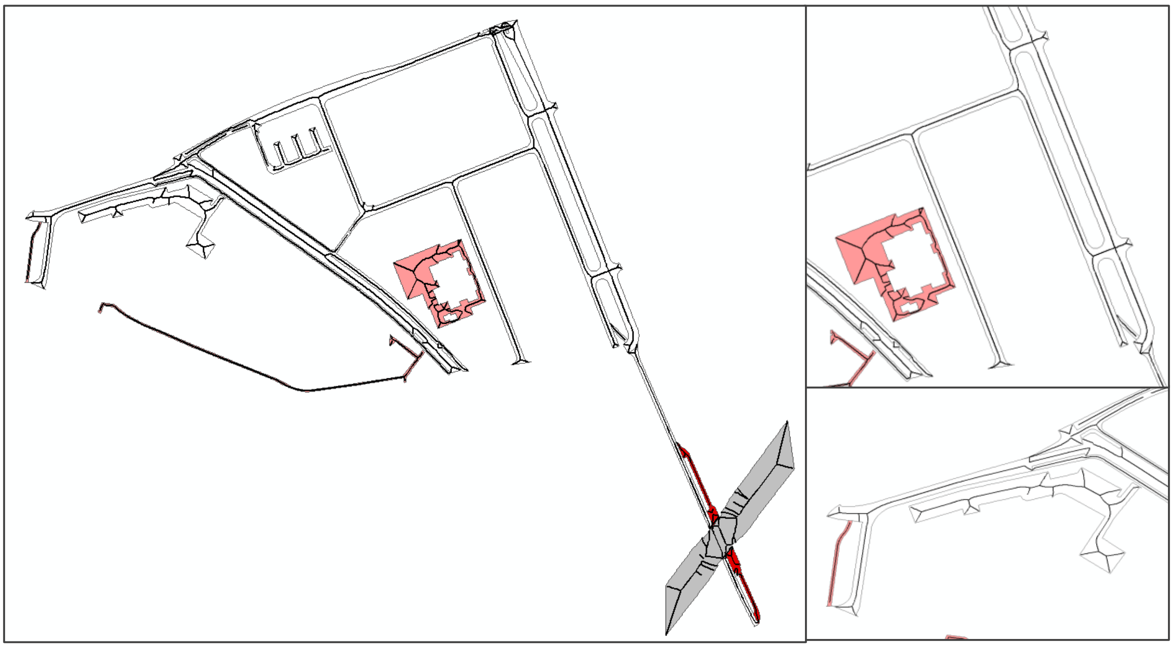
Dead-ends have not been tested in this study. However, a quick check has been executed manually after running the previous models. It resulted in the knowledge that more dead-ends should be eliminated during the generalization. However, to ease this elimination, Firstly, the crossings should be determined, because then, parts of roads smaller than 100 meters can be eliminated more easily.

(i) Crossings & centerlines

Crossings and centerlines have not extensively been tested in this study. The focus has been on the object type PARTOFROAD. However, the crossings seem very important to create, because these crossings can be used to define the attribute values within the required TOP10NL attribute *typeOfInfrastructure* (i.e. 'crossing' and 'connection'). In addition, these crossings can be used to limit the enormous objects as created in the uniform BGT, and expanded in the midscale BGT (see Section 7.4). In Altena et al. (2013) already some major improvements were made concerning this subject. Therefore, I decided to focus on the other parts of this study instead of creating crossings again.

The centerlines can be added with the help of the Polygon to Line tool as developed in Section 10.1b. I have tested this model for the resulting midscale BGT polygonal roads within the dataset of Amersfoort (see Appendix H, Figure H.13). In Figure 10.5 (next page), the results are visualized. It seems possible, but the connection between those lines should definitely more optimized.

Figure 10.5: Results of creating centerlines in Amersfoort



10.3. BUILDINGS

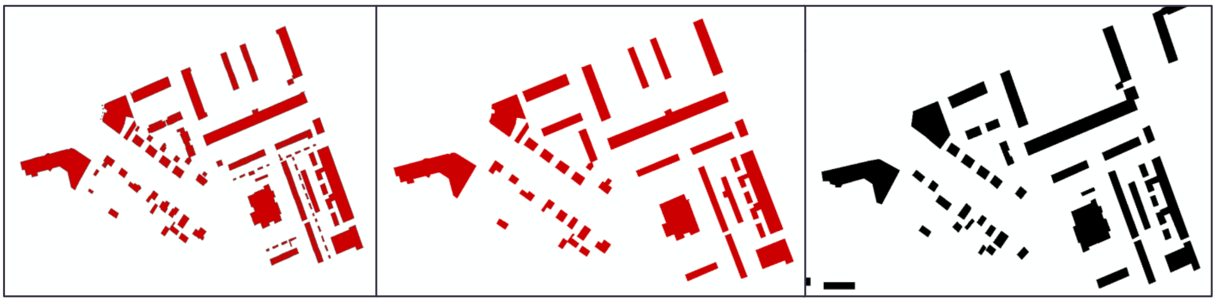
To aggregate the neighboring buildings within a distance of 2 meters, the tool Aggregate Polygons is used. This tool aggregates the objects based on a specific Aggregation Distance (2 meters) and with the possibility to assign Barrier Features (PARTOFWATER and PARTOFROAD). A drawback of this tool is that it loses all attributes and their attribute values after the aggregation. Therefore, the input data is transformed into point data and added to the aggregated data based on an Intersection.

Then, small buildings should be eliminated. In the data specifications, the distinction was made between all buildings smaller than 3x3 meters or with a diameter smaller than 4 meters. In addition, buildings within built-up area, which cannot be recognized from access roads smaller than $50m^2$, should be eliminated. To ease the automatic generalization, I decided to delete all buildings outside built-up areas with an area of $9m^2$ instead of measuring both sides and the diameter of the building. Inside built-up area, I decided to delete all buildings smaller than $50m^2$, because it is not that easy to specify the access roads or to automatically recognize which building can be seen from this access road.

After that, the buildings were simplified with a Simplification Tolerance of 3 meters and the Minimum Area again set on $9m^2$. This encourages the data specifications of eliminating the sheds, corridors, sky bridges, expansions etc. smaller than 3 meters or within an area smaller than 3x3 meters. Finally, the patios or courtyards were specified and eliminated when smaller than $1000m^2$. Therefore, the tool Eliminate Polygon Part is used (see Appendix H, Figure H.14).

The model resulted in the aggregation of many buildings. When comparing these buildings based on the subsurface (midscale BGT) with the buildings based on aerial view (TOP10NL), it is surprising that the buildings are quite similar to each other, despite the different input datasets used (see Section 8.3) and the parameter changes. The amount of buildings is similar, which is surprising due to the fact that more buildings inside built-up areas should have been eliminated than the data specifications indicate. Visually, the buildings seem less simplified than TOP10NL. However, this is easy to adjust by using some stricter simplification parameters (see Figure 10.6).

Figure 10.6: Results of the automatic generalization of the cycle buildings (middle) in comparison with the buildings of TOP10NL (right) and the uniform BGT (left) in Amersfoort



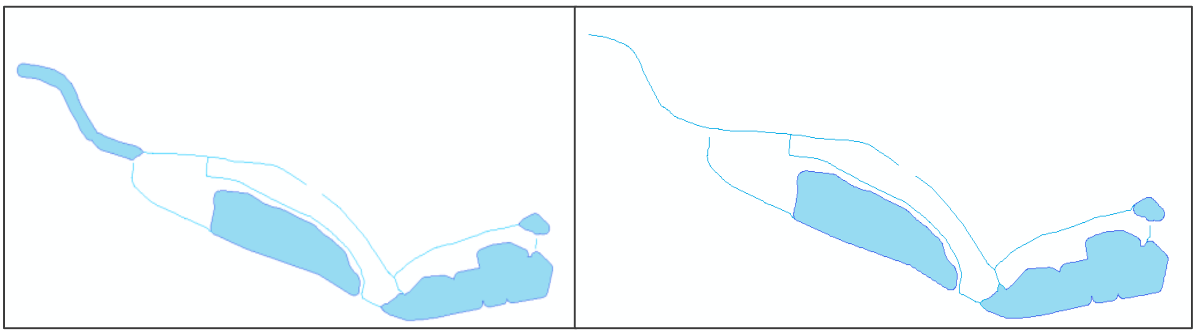
To ease the generalization process, other parameters are used (e.g. 9m² instead of 3x3 or 4 diameters). Because these parameters resulted in a dataset comparable to TOP10NL, these parameters will be approved and changed within the data specifications.

10.4. WATER

As mentioned in Section 9.4, the attribute value 'shore/ditch' of SUPPORTIVEPARTOFWATER should be added to the object type PARTOFWATER. When doing so, the water will become wider, which affects the rule of the objects wider and smaller than 6 meters. Therefore, I have run the model twice: with and without the combination of water with the objects of SUPPORTIVEPARTOFWATER to see how this affects the midscale BGT. In Figure 10.7 is shown that this can have a major impact on the visual appearance of the dataset. However, because of the rule that 'shore/ditch' should be added to PARTOFWATER instead of TERRAIN, the choice is made to add this attribute value before changing the geometry.

To change geometry, a buffer with a distance of -3 meters is used to determine the width of the water objects. The objects which are entirely inside this buffer should become line objects, and the objects intersecting the buffer, should remain polygonal objects. Because it is also possible to split objects into polygon and line objects, the parts of objects, which should remain polygonal are erased from the dataset. The remaining polygons are changed into line objects. The translation into line objects has been executed with the model Polygon to Line (as specified in Section 10.1b). In addition, the optional attribute *widthClass* is added, which specifies the line objects bigger than 3 meters and smaller than 3 meters, and the polygonal object bigger than 6 meters. This attribute is optional in TOP10NL. However, it is very easy to add to the attributes and gives additional information about the water objects (see Appendix H, Figure H.15).

Figure 10.7: Changed geometry with 'shore/ditch' (left) versus changed geometry without 'shore/ditch' (right) in the Amersfoort dataset



10.5. NATURE

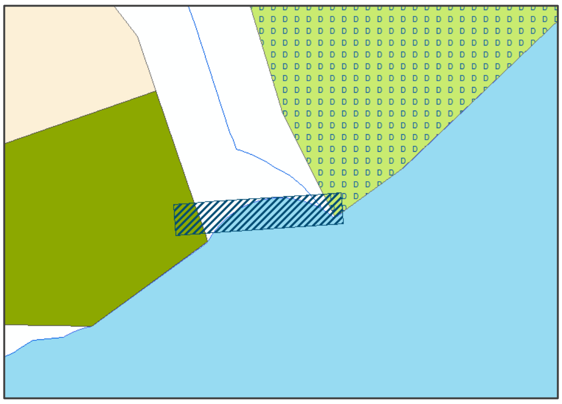
Firstly, the objects with attribute values 'agrarian grassland' (BGT) and 'remaining grassland' are combined. Secondly, the objects which should be moved to TERRAIN as specified in the previous cycles are gathered (i.e. objects within the object types COVEREDPARTOFTERRAIN and UNCOVEREDPARTOFTERRAIN, specific parking areas, specific verges, and objects with the attribute value 'silt' from SUPPORTIVEPARTOFWATER). These objects are merged with the output of the first step to collect all terrain objects. Then, the objects were outlined whether or not they were bordered by the objects of 'hard topography' (i.e. PARTOFROAD and PARTOFWATER) or 'soft topography' (i.e. TERRAIN). Therefore, the object types PARTOFROAD and PARTOFWATER are added to the object type TERRAIN and with the help of the Polygon Neighbors tool, the neighboring objects were specified. When an object was bordering terrain objects, the objects should be bigger than 1000m² to exist as separate object. If not, the objects should be merged with the neighboring terrain object. When an object was bordering only 'hard topography', the objects are stayed the same, no matter its size. An exception are the forests outside built-up area, which has a minimum size of 50m². These are selected and added to the remaining terrain objects again (see Appendix H, Figure H.16 and H.17).

Due to earlier elimination of objects, the dataset still exists of a lot of holes. It turned out that it might be better to execute this model after the solvation of these holes. Therefore, the second part of the model is executed as part of Section 10.8 where the holes will be solved.

10.6. BRIDGES & TUNNELS

Bridges & tunnels should not cross line objects and should be connected to objects in the object type PARTOFWATER or PARTOFROAD. Therefore, the just created polygonal object types of PARTOFWATER and PARTOFROAD are merged with the object type PARTOFBRIDGE AND PARTOFTUNNEL to see which polygonal objects in PARTOFWATER or PARTOFROAD are neighboring these objects. When a bridge or tunnel does not have neighboring objects in PARTOFWATER or PARTOFROAD, these bridges or tunnels will be eliminated from the dataset (see Appendix H, Figure H.18 and H.19).

Figure 10.8: Objects of PARTOFBRIDGE which remain in the midscale BGT, but should be removed in Amersfoort



At first sight, this model results in the elimination of the right objects. However, it also occurs that objects of PARTOFBRIDGE or PARTOFTUNNEL are not crossing objects of PARTOFWATER or PARTOFROAD, but only lie parallel to these objects (see Figure 10.8). Although it is an exception, therefore should be thought of a solution.

10.7. OTHER

The polygonal objects of the uniform BGT are converted into point objects or line objects. Therefore, the objects of ENGINEERINGSTRUCTURE and REMAININGSTRUCTURE were converted to points and merged with each other into one new object type, called LAYOUTELEMENT_point (see Appendix H, Figure H.20). The objects of SEPARATIONS were converted into line objects a merged with the already existing line objects into the new object type LAYOUTELEMENT_line (see Appendix H, Figure H.21).

The conversion of polygonal objects into points or line objects results automatically into holes in the dataset, because the polygonal objects were part of ground level. Therefore, in the next section, these holes will be filled with neighboring objects, mostly of the object type TERRAIN.

10.8. INTEGRATING CYCLES

To perceive the quality of a key registration again, the logical consistency should be protected. Therefore, during the generalization of the midscale BGT, already all eliminated polygonal objects, and polygonal objects which changed geometry were collected. Now, all resulting objects on ground level are selected (*relativeHeight* = 0) and merged with these eliminated objects on ground level. Then, the tool Polygon Neighbors is used to define where the eliminated objects should go to (see Appendix H, Figure H.22). Therefore, a few rules are developed. These rules should be added to the data specifications of the midscale BGT (see Table 10.3).

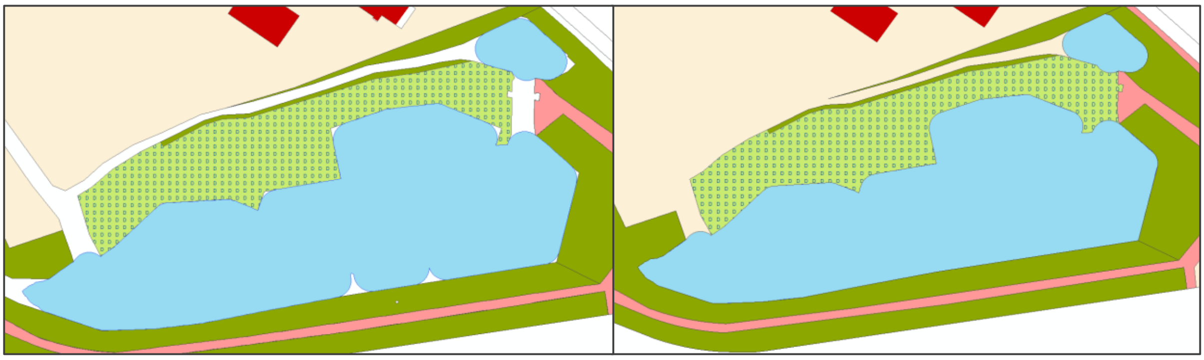
Table 10.3: New developed data specifications to perceive the logical consistency of the dataset

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
IntA1		Polygon	Resulting holes	*relativeHeight*= 0	No	Neighboring terrain objects	Change into neighboring terrain object	-	-
IntA2		Polygon	Resulting holes	*relativeHeight* = 0	No	Not neighboring terrain objects	Change into neighboring object based on maximum length	-	-

Finally, this model is extended with the rules of the cycle nature, because the data specification of 'hard topography' and 'soft topography could not be completed until the main holes were solved (see Section 10.5 and Appendix H, Figure H.17).

However, this will not solve all holes created in this generalization, because some objects were simplified and therefore, the objects changed. To solve these holes and overlaps within the dataset, the model of Altena et al. (2013) is used and modified (see Appendix H, Figure H.23, H.24, H.25, and H.26). In these models, the topology issues, due to the executed generalization are specified. The overlapping parts are recognized and deleted, and the holes were filled with neighboring objects (see Figure 10.9). Finally, the identification of objects is changed in all object types following the developed tool Formal Key Registration Requirements as developed in Section 10.1c. Unfortunately not all parts of ground level could be fixed with these models. However, the extra adaptations are out of the scope of this study and therefore left as recommendation for further research in this study.

Figure 10.9: Before (left) and after (right) filling holes to acquire logical consistency in the dataset



11. PHASE 4: ANALYSIS AND EVALUATION

In the fourth phase of the midscale BGT, the developed data specifications will be analyzed and the midscale BGT will be evaluated as replacement of TOP10NL in the system of key registrations in the Netherlands. Therefore, the evaluation method 'visual comparison of outputs' will be used to evaluate the data specifications of the midscale BGT on its missing information due to missing data or due to processes that cannot be automated (see Section 11.1). Then, the developed data specifications will be analyzed and modified when necessary (see Section 11.2). This phase will result in recommendations, which can be used for further development of a midscale data product in line with the topographical key registrations in the Netherlands (see Section 11.3).

11.1. VISUAL COMPARISON OF OUTPUTS

Visually, the generalization of the uniform BGT into a midscale BGT resulted in a dataset with a lot of inconsistencies and differences comparing to TOP10NL (see Figure 11.1, 11.2, and 11.3). Due to the focus on model generalization and because the model is not yet perfect, many objects are not simplified and there are still holes in the resulting maps. However, when looking more into the data per cycle, also many similarities with TOP10NL can be noticed. In this section, the similarities and differences of the developed midscale BGT with TOP10NL will be analyzed per cycle to reveal the missing information of the midscale BGT. The analysis will be executed both visually and with further insight in the data of the midscale BGT.

Figure 11.1: Resulting midscale BGT Amersfoort dataset (left) in comparison with TOP10NL (right)

Figure 11.2: Resulting part of the midscale BGT Maastricht dataset (left) in comparison with TOP10NL (right)

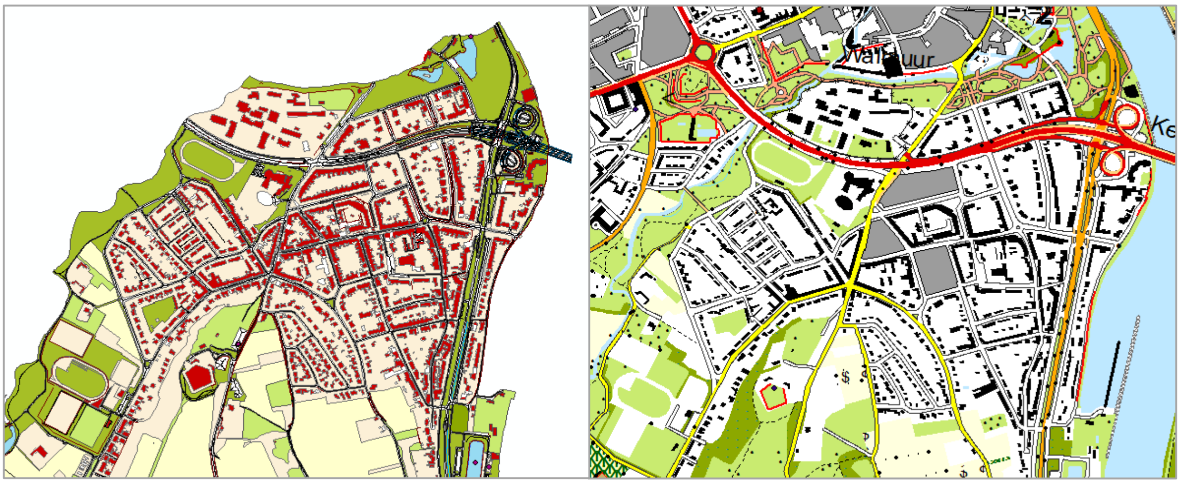
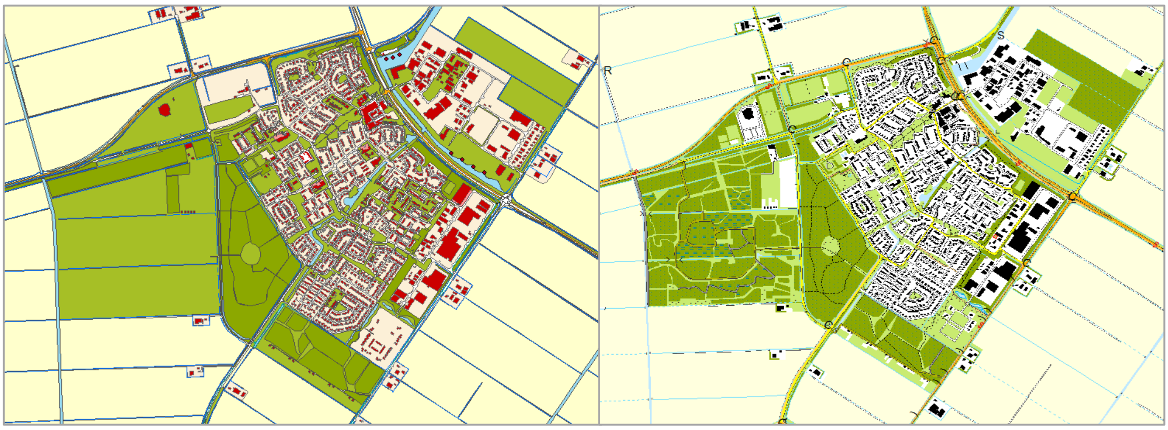


Figure 11.3: Resulting part of the midscale BGT Dronten dataset (left) in comparison with TOP10NL (right)



11.1.1. GENERIC RESULTS

In general, the implementation of the data specifications into a working generalization system occurred as expected: the indicated object types were eliminated, the designated objects were collapsed into objects with different geometries, and most object types have decreased in amount of objects. In Table 11.1, the number of objects of the three datasets are outlined and compared to the number of objects within the uniform BGT. To better compare the amount of objects before and after the generalization of the midscale BGT, some object types of the uniform BGT are summed. For example, the object types PARTOFROAD and SUPPORTIVEPARTOFROAD of the uniform BGT are summed to compare to the object type PARTOFROAD of the midscale BGT.

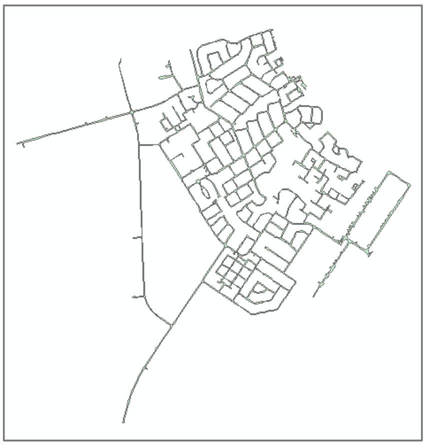
Table 11.1: The amount of objects of three datasets, before and after the generalization into the midscale BGT

		Amersfoort dataset				Maastricht dataset				Dronten dataset			
Cycle	Midscale BGT Object type	Before	After	Difference	%	Before_	After_	Difference_	%_	Before.	After.	Difference.	%.
Roads	PARTOFROAD (polygon)	119[a]	19	-100	-84,0%	887[a]	97	-790	-89,1%	1874[a]	277	-1597	-85,2%
	PARTOFROAD (line)		0	0			1	1			4	4	
	ROADS_centerlines		196	196			1564	1564					
	RAILWAY	4	4	0	0,0%	-	-	-	-	-	-	-	-
Buildings	BUILDING	164	46	-118	-72,0%	2592	696	-1896	-73,1%	4989	2168	-2821	-56,5%
Water	PARTOFWATER (polygon)	2[b]	4	2	100,0%	14[b]	5	-9	-64,3%	821[b]	86	-735	-89,5%
	PARTOFWATER (line)		8	8			5	5			1280	1280	
Nature	TERRAIN	71[c]	56	-15	-21,1%	1147[c]	421	-715	-62,3%	2025[c]	1272	-753	-37,2%
B&T	PARTOFBRIDGE	5	4	-1	-20,0%	37	24	-13	-35,1%	51	48	-3	-5,9%
	PARTOFTUNNEL	1	1	0	0,0%	-	-	-	-		-	-	
Other	LAYOUTELEMENT (line)	19[d]	21	2	10,5%	252[d]	270	18	7,1%	1[d]	121	120	12000,0%
	LAYOUTELEMENT (point)	2[e]	2	0	0,0%	10[e]	10	0	0,0%	51[e]	51	0	0,0%
Totaal		387	361	-26	-6,7%	4939	3093	-1835	-37,4%	9812	5307	-4505	-45,9%

[a] = PARTOFROAD_UNIFORM + SUPPORTIVEPARTOFROAD_UNIFORM
[b] = PARTOFWATER_UNIFORM + SUPPORTIVEPARTOFWATER_UNIFORM
[c] = COVEREDPARTOFTERRAIN_UNIFORM + UNCOVEREDPARTOFTERRAIN_UNIFORM
[d] = SEPARATIONS_UNIFORM (line) + SEPARATIONS_UNIFORM (polygon) + ENGINEERINSTRUCTURE_UNIFORM (line)
[e] = ENGINEERINGSTRUCTURE_UNIFORM (polygon) + REMAININGSTRUCTURE_UNIFORM (polygon)

Noticeable is that some object types increased in amount of objects after the generalization. This can be explained by two reasons: Firstly, the rule that water objects can be split into both polygon and line objects, which automatically results in multiple objects. And secondly, the tool Polygon to Line, as developed in Section 10.1b is not working optimally yet, and therefore, more lines than existing polygons are identified. Another notice is the enormous elimination of polygonal objects within the cycle roads. It turns out that in each BGT test dataset about 85% of the polygonal objects were eliminated or collapsed into line objects. However, a similar problem as in the uniform BGT occurs (see Section 7.4). Due to the model allowing objects to aggregate based on their required attribute values, enormous objects were created (see Figure 11.4). Within the cycle roads, this could

Figure 11.4: One polygonal object within the object type PARTOFROAD in the Dronten dataset



be solved by acquiring the crossings of the roads. With the help of this information, the roads can be split up into multiple road objects divided by their crossings.

11.1.2. ROADS

Although the number of roads is not correct for a midscale dataset, it seems that the resulting roads are visually quite comparable to TOP10NL (see Figure 11.5). This is surprising because two sources are used to acquire these roads of the midscale BGT, i.e. the BGT attribute values and the TOP10NL implementation rules. However, also some differences can be recognized. For example, it turned out that many roads displayed as line objects in TOP10NL are still displayed as polygonal objects in the midscale BGT. The reason for this is that in the midscale BGT the parameter of 2 meters is implemented as maximum, while in TOP10NL the mean width of 2 meters is used, which results in different geometry (see Figure 11.6). Therefore, the data specifications should be adapted with a parameter based on its mean width or the data specifications should be changed. Because the concerned objects will be too small for a 1:10,000 scale, I decided to change the data specifications by adding its mean width.

Figure 11.5: Visual comparison of the roads in the midscale BGT (right) with the roads in TOP10NL (left) in the Dronten dataset

Figure 11.6: Geometry differences due to different parameters in the Dronten dataset (left) compared to TOP10NL (right)

Another difference is that I did not succeed to identify which roads are parallel and which roads are freely situated, which resulted in differences between the midscale BGT and TOP10NL. This information about parallel roads can be helpful, for example when determining the attribute value ‘mixed traffic’. In reality and in the uniform BGT, often parallel roads are determined as separate objects, while in the midscale BGT, these roads should be combined into one road. When knowing which roads lie parallel to each other, it is easier to determine which roads are combined and should contain the attribute value ‘mixed traffic’. In addition, some data specifications were not tested because of this drawback in the model. Therefore, there should be further investigations executed on how to deal with those parallel roads.

As already expected, some of the required attributes in TOP10NL are missing in the data of the midscale BGT. These attributes are *typeOfInfrastructure* and *yes/noSeparationOfLanes*. It might be possible to acquire the attribute values of *typeOfInfrastructure* with the help of automatic generalization when corner points of the roads can be determined. Therefore, the object type PARTOFROAD should be divided into multiple road segments based on crossings and connections. This will result in an increased amount of objects, although less complex (see Section 11.1.1). Then, the attribute *typeOfInfrastructure* can be calculated and the attribute values, which determine the 'crossings' and 'connections' can be identified. The attribute *yes/noSeparationOfLanes* should be added from additional sources, because it seems not possible to acquire this from the BGT-perspective. However, here should be questions whether or not this information is interesting to keep in the midscale BGT.

In addition, some changes should be added to acquire a cleaner dataset, e.g. dead-ends should be deleted, the additional rules should be added, and simplifications might be necessary.

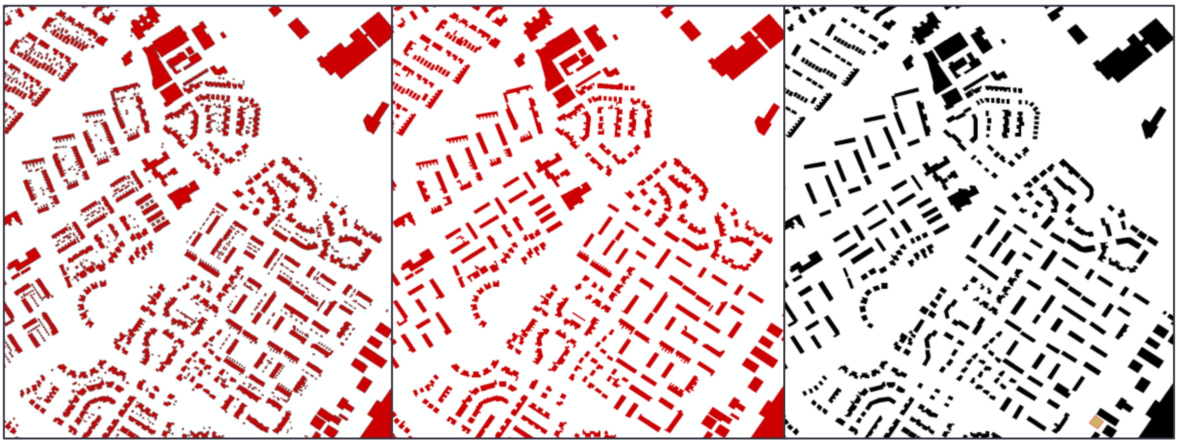
Railways

The object type RAILWAY has not been tested due to the lack of enough objects in the test datasets. However, a few attributes can be expected as not possible to derive from the BGT. These attributes are *typeOfInfrastructure, widthOfRailway,* and *numberOfTracks*. TOP10NL already derives the attribute *numberOfTracks* from external data of ProRail (Kadaster, 2012), which means that this information can be easily added to the midscale BGT. However, due to the fact that ProRail is source holder of the BGT, perhaps it would be better to add this information to the initial BGT to maintain the uniformity within the topographical key registrations. The attribute *typeOfInfrastructure* can be added with the same automatic generalization method as outlined previously in this section. The attribute *widthOfRailway* can be calculated by measuring the width of the objects in the BGT. However, it might generate better results when acquiring the exact measurements of these railways in the open field with data from ProRail.

11.1.3. BUILDINGS

As already mentioned in Section 10.3, the buildings of the midscale BGT seem comparable to the buildings of TOP10NL. Although buildings as subsurface are used instead of buildings at an aerial perspective, and different parameters are used, this still resulted in a similar amount of buildings as in TOP10NL (see Figure 11.7).

Figure 11.7: Results of the automatic generalization of the cycle buildings (middle) in comparison with the buildings of TOP10NL (right) and the uniform BGT (left) in Dronten



However, due to the use of buildings as subsurface and the implementation of these objects within ground level, the attribute *overpass*, which determines whether or not a building overlaps with another object type (e.g. a building with the first floor leaning over a road or water object), is not visible anymore. Although this is an optional attribute in TOP10NL, it might be very interesting information to keep in the midscale BGT. However, this should be discussed with users of the BRT. When it is important to keep the information about overpasses in the midscale BGT, the optional attribute value 'overpass' as specified in IMGeo within the object type ENGINEERINGSTRUCTURE can be used (see Appendix C, Table C.3). Another possible solution is that the generalization can be executed with the help of buildings revealed from the BAG, which consists of buildings at an aerial perspective. Therefore, the same data specifications as developed in Section 9.3 can be applied, completed with the rules about overpasses (as specified in the implementation rules of TOP10NL (Kadaster, 2012)). In addition, the buildings should not be part of ground level anymore, because the objects visualizing overpasses should overlap with objects of other object types.

When using the BAG as input dataset, it will be superfluous to keep collecting the BGT buildings. But, due to little changes, it might also be interesting to consider the buildings of the BGT as input dataset. In terms of the 'collect once, use many times' principle, it might be possible to expand the buildings of the BGT with the information of the BAG in order to create one topographical key registration derived from the BGT to outline all addresses and buildings in the Netherlands (see Section 11.3).

11.1.4. WATER

Visually, the water objects of the midscale BGT are behaving as expected. The water objects changed into line objects and objects split into multiple polygon and line objects where necessary. However, this also occurs on very small parts in both geometries (see red circle, Figure 11.8). Therefore, an exception should be added to the rule which prevent those small objects. Therefore, the rule, which specifies that the polygonal objects cannot be smaller than 50m² should be added later in the model. This ensures that these small polygons will be eliminated. However, then another solution should be found to extend the lines where these small polygons occur.

Figure 11.8: The results of water objects split into multiple polygon and line objects in the cycle water (middle) in comparison with the water objects of TOP10NL (right) and the uniform BGT (left) in the Dronten dataset. The red circle shows an example of the small parts occurring in the midscale BGT.

The generalization of the cycle water can be executed when following the BGT-perspective. Although this resulted in the missing attributes *mainDrain, occurrence,* and a missing division between the attributes *typeOfWater* and *function*. Whether or not these are important attributes to keep within a midscale BGT should be analyzed by users of the BRT. However, visually similar results were created without these attributes. When it is important information, perhaps the water boards have this extra information available and are willing to add this to the BGT. In addition, the optional attribute *widthClass* was very easy to add and this can also be easily added within the cycle roads. However, whether or not this is important information should also be outlined, because when nobody uses this type of information, it is insignificant to add it to the midscale BGT.

11.1.5. NATURE

The most important data specification developed within the cycle nature was the combination of terrain objects smaller than 1000m², with the exception of objects bordering objects defined by 'hard topography'. When analyzing the cycle nature, it occurs that this rule can be used to also decrease the amount of holes as developed within the previous cycles. Therefore, this cycle has been executed after the cycle other.

However, the model executing this data specification is not optimally developed yet. Sometimes, this model works perfectly, closing holes with neighboring objects and combining small terrain objects. For example, in Figure 11.9, a successful combination of objects is visible showing that the footpaths are eliminated and the resulting holes are filled with the neighboring objects with the attribute value 'greening' within the object type TERRAIN. However, these results do not always occur. In Figure 11.10, it seems that not all holes were filled, or that some holes were filled, but not with the right attribute values. Therefore, these errors should be further analyzed in order to create a more consistent dataset.

Another aspect within this cycle, that should be specified, is which attribute values are relevant to show and which attribute values are too specific for this type of scale. For example, the attribute value 'shrubs' is not defined in TOP10NL, and therefore, the assumption should be made that this is too much information for this type of scale. Another example is that at this type of scale the different types of forests seem not relevant anymore, and that these types of forested areas can be aggregated. However, this should be decided by the current users of the BRT, because they can suggest best what they think is useful.

Figure 11.9: Successful combination of objects within the cycle nature (middle), compared to the uniform BGT (left) and TOP10NL (right)

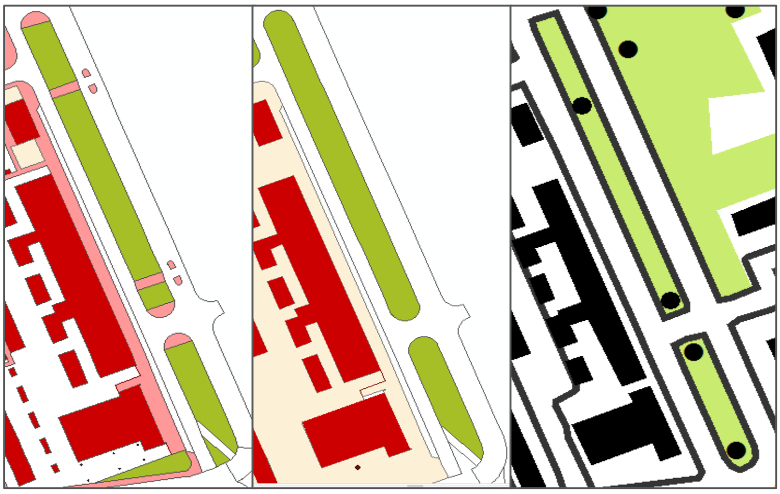
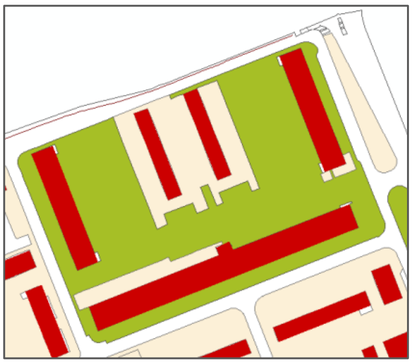


Figure 11.10: Unsuccessful combination within the cycle nature

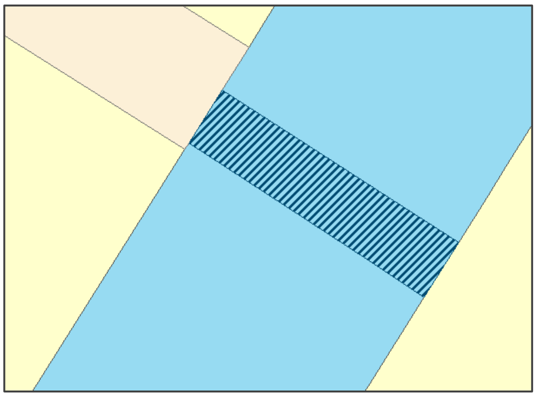


11.1.6. BRIDGES & TUNNELS

Also the choice to keep the object types PARTOFBRIDGE and PARTOFTUNNEL should be presented to users of the BRT to outline whether it is necessary to specify bridges and tunnels separately in the midscale BGT dataset.

Because the choice is made to keep the attribute *relativeHeight,* it is easier to make selections based on the positive attribute values to select bridges and on the negative attribute values to select tunnels. Therefore, these object types, which specify bridges and tunnels might seem overflowing. However, when the attribute *heightLevel* will be remained, it might be interesting because then it is not possible anymore to select bridges and tunnels separately. This decision should be considered with users of the BRT. When the objects are interesting to add in separate object types, a solution should be found to specify which bridges or tunnels are lying parallel to the roads or waterways and which roads or waterways are used to access these bridges and tunnels (see Figure 11.11).

Figure 11.11: An example of a bridge which is not accessed by road or waterways



11.1.7. OTHER

In the final cycle, the main choice is made that the attribute values and the semantics of the BGT are used to create LAYOUTELEMENT in the midscale BGT. The result is that many differences exist between TOP10NL and the midscale BGT, because only 11 out of the 80 attribute values are similar. However, before gathering the attribute values of TOP10NL, users should decide which attribute values are relevant to add to the midscale dataset. Because of the enormous variety in the attribute values, it might be that some values will not be used anymore and can be kept out of the midscale BGT. Then, the decision should be made how to create the relevant attribute values. Some attribute values can be derived from the optional attribute values from the BGT. Other attribute values might be derivable from additional datasets, or the attribute values should be added to the source data (see Table 8.2, p.50).

In addition, it might be confusing to use the same semantics for object types, attributes, or attribute values, which slightly differ from another dataset. For example, I named the object type within this cycle LAYOUTELEMENT, while it differs enormously with the object type LAYOUTELEMENT of TOP10NL. Therefore, I recommend changing the name when object types, attributes, or attribute values changes semantics.

11.2. ANALYZING DATA SPECIFICATIONS

A total amount of 71 data specifications are specified to derive the midscale BGT out of the uniform BGT. However, the data specifications are not tested by cartographer' experts and therefore, the quality of these data specifications are not what it should be. As already outlined, more attention should be paid to the detailed formulation of the data specifications. At this moment, the translation of the implementation rules sometimes resulted in vague descriptions, such as 'most important buildings' etc., which are very difficult to automatize. When implementing these data specifications in the model, most of these vague descriptions are already recognized and modified by the exact parameters used. However, when testing these data specifications by cartographer' experts, it might be possible to recognize these data specifications in an earlier stage.

Table 11.2: The occurrence of data specifications per cycle

Cycle	Occurrence
Generic data specifications	9
Roads	36
Buildings	7
Water	4
Nature	7
Bridges & Tunnels	4
Other	4
Total	**71**

When comparing the amount of data specifications per cycle, it occurs that the cycle roads contain the most amount of data specifications. Therefore, as expected, this cycle can be seen as the toughest cycle in the generalization towards a midscale BGT (see Table 11.2).

As promised in Section 2.2, the data specifications need to be harmonized to create more overview on what kind of data specifications are developed. Therefore, the generalization operators are used to define the **constraint type**. These generalization operators are used, because the actions seem the most important aspects of the model generalization. However, the action itself in the data specifications are often more specified than only these generalization operators. Therefore, I decided to specify the **constraint type** with these operators. In addition, the data specifications, which change the data of all objects (e.g. identification of objects), are specified with the **constraint type** 'data management'. In Table 11.3, the amount of data specifications per model generalization operator is described.

Table 11.3: The amount of data specifications per constraint type

Constrain type (model generalization operators)	**Occurrence**
Aggregation	10
Amalgamation	2
(Re)classification	6
Collapse	7
(Class) selection	35
Simplification	2
Merge	3
Data management	6
Total	**71**

In general, most developed data specifications can be implemented in an automatic generalization system using ArcGIS. As already indicated in Section 11.1, some required attributes of TOP10NL cannot be specified from a BGT-perspective. However, these attributes might be possible to automatically generate, or can be acquired from additional sources (see Table 11.4). It might be interesting to acquire these attributes at the source data (the initial BGT) in order to keep the datasets uniform within the content of the data.

Table 11.4: Missing attributes of TOP10NL in the midscale BGT

Midscale BGT object type	Missing attribute	Derived by...?
PARTOFROAD	*Yes/NoSeparationOfLanes*	Additional source (RWS)
	typeOfInfrastructure	Automatic generalization
RAILWAY	*typeOfInfrastructure*	Automatic generalization
	widthOfRailway	Additional source (ProRail) or automatic generalization
	numberOfTracks	Additional source (ProRail)
BUILDINGS	*Overpass*	IMGeo attribute values or the BAG
PARTOFWATER	*mainDrain*	Additional source (water boards)
	Occurrence	Additional source (water boards)

The analysis of the data specifications resulted in the knowledge that the implementation rules of TOP10NL are made unnecessary difficult over the years. For example in the cycle buildings, where different parameters and exceptions are used, while a simplification of these parameters gave a similar result. Therefore, it might be interesting to be very critical when implementing a new key registration, such as the BGT, to ensure that this key registration can be kept simple and organized, containing only relevant and necessary data.

11.3. THE MIDSCALE BGT AS REPLACEMENT OF TOP10NL

In general, most data specifications specifying the midscale BGT from a BGT-perspective can be implemented in an automatic generalization system. However, whether this resulting midscale dataset is usable as implementation for the system of key registrations in the Netherlands should be judged by users of TOP10NL. Therefore, users should identify what objects are relevant and which information should be maintained in the different topographical key registrations. With more information about the use and implementation of the topographical key registrations, decisions can be made about what objects are relevant and which information should be maintained in the different topographical key registrations. In addition, a decision can be made about what is the best manner to derive those data and in what type of dataset is the best manner to implement this in the system of key registrations. Therefore, it is interesting to acquire all attribute values at the source data, instead of adding different types of data to different types of scales. Then, these attribute values should be similar at every scale and there should be no doubts about the meaning of every attribute value.

In addition, a closer look should be taken to the definitions of object types, attributes and attribute values. Now, both in the BGT and in the BRT, different attributes and attribute values are available, which contain the same name, but mean different things, or different attributes and attribute values have the same meanings, but have different names. Therefore, I will recommend to use the BGT- attribute values, because these attribute values are the ones that available at the largest scale.

The smaller scales of the BRT derived from a midscale BGT is very good possible. Although here the same differences in semantics occur. Of course, when trying to maintain most objects exactly the same as TOP10NL, the entire models of the BRT generalization can be reused. However, it might be a good idea to specify again the relevant and necessary data. This will cost initially more time, but in the end, it will ensure the organized dataset entirely derived from the BGT.

When combining the BGT and the BRT into one topographical key registration, it might be interesting to derive also the other topographical key registrations based on its relevant information. For example, as already outlined in Section 11.1.3, the data of the BAG can be added to the initial BGT dataset in order to create one topographical key registration, which contains all relevant information. The separation of data will be solved with this system.

Another possibility is to change the entire system of topographical key registrations, for example by using themes instead of object oriented datasets. These themes can then be obtained individually, which might be interesting when a company only needs the information about roads or water. Therefore, the different cycles as used in this study (with the bridges and tunnels implemented in roads and water). This increases the use of network-based analyses.

12. CONCLUSIONS AND RECOMMENDATIONS

In the Netherlands, the many different topographical key registrations have resulted in a separation of geographic data within the system of key registrations. With the development of the BGT, the system can benefit by integrating the topographical key registrations again based on the 'collect once, use many times' principle. With this principle, the topographic data can be acquired at the largest scale of the BGT and the midscale and small scale topographical key registrations can be derived from this dataset. In this study, I have developed data specifications in order to generalize the BGT into two different generalization products in line with the system of Dutch key registrations, i.e. an aggregated, uniform data product, and a midscale data product. In order to structure this study, I have addressed the following objectives:

- To develop an aggregated, uniform BGT product as input for the midscale BGT;
- To develop data specifications based on the uniform BGT as input for the midscale BGT;
- To test the data specifications by implementing them into an automatic generalization system;
- To analyze and evaluate the resulting midscale BGT.

In the subsequent sections of this chapter, the main findings of those objectives will be outlined, followed by some recommendations. This chapter will end with a reflection of my study and some thoughts about how this study can be expanded by executing future research.

12.1. MAIN FINDINGS & RECOMMENDATIONS

12.1.1. THE DEVELOPMENT OF A UNIFORM BGT PRODUCT

As first step towards a midscale BGT derived from the BGT, I have developed an aggregated, uniform BGT. Therefore, I have proposed data specifications, which aggregates both virtual borders (which source holders add when they are implementing the BGT) and optional objects (which are not available nationwide). The required attributes, as defined in IMGeo, were used to determine which objects should be aggregated and which objects could stay in the uniform BGT. The implementation of the data specifications have resulted in the aggregation of many virtual borders and optional objects. The main advantage of the used generalization method is that these objects were not only aggregated visually, but the data behind the objects were also aggregated.

However, when generalizing the BGT into a uniform BGT, some complications came forward. Firstly, the aggregation of these objects resulted in very large and complex objects, due to the aggregation of too many objects with the same required attribute values. Therefore, a solution should be found on how to deal with those enormous amounts of objects. A possibility could be to expand the data specifications with constraints on the area or length of objects, or to split objects at crossings. Another possibility is to add water or road networks to create more connection between the road or water objects. Secondly, it has occurred that the optional attribute values disappeared, while some of these attribute values explained the entire object (e.g. 'speed bumps'). Therefore, one of the main recommendations for the uniform BGT is to identify which required data of the BGT is relevant and which data is not.

In general, the uniform BGT has resulted in a compacter dataset containing all required object types, attributes and attribute values, which can easily be used as the input for the midscale BGT. In addition, it is interesting to consider if the uniform BGT can also be used as separate key registrations with as main purpose to have a simple large scale key registration within the system of key registrations in the Netherlands. Therefore, firstly, the adaptations as described above should be executed (when using the uniform BGT only as input for the midscale BGT, those adaptations are also possible in a later stadium of the automatic generalization). And then, the uniform BGT can be used as separate key registration with a 1:500 until a 1:5,000 scale. The main usage of this uniform

BGT will be for different geo-related tasks, which do not need extra optional information. However, it should be considered what the effects of the availability of such a product are on the usage of other products in the system of key registrations. Probably, the usage of other key registrations, i.e. the BGT with optional objects, and the midscale data product, will become less important when adding a uniform BGT to the key registrations. Another decision that should be made when implementing the uniform BGT as key registration, is who is going to be the source holder of the uniform BGT? In general, there are two options: firstly, the source holders of the BGT can also be the source holder of the uniform BGT; and secondly, the uniform BGT can be seen as the responsibility of the organization which has executed the aggregation. The decision will have an effect on the results of the uniform BGT, because more objects will aggregate when the source holder is the same nationwide.

12.1.2. THE DEVELOPMENT OF DATA SPECIFICATIONS FOR THE MIDSCALE BGT

In order to develop a midscale BGT product, the first step is the development of data specifications following the template of Stoter et al. (2009a). The data specifications are developed based on the implementation rules of TOP10NL and a reverse engineering of the BGT with TOP10NL. However, the data specifications are being developed following the BGT-perspective in order to reveal the most differences between the BGT and TOP10NL. This way means that all semantics, which are defined in the uniform BGT, are being used and being combined with the implementation rules of TOP10NL.

The development of data specifications has resulted in 71 different data specifications divided over 6 different cycles: roads, buildings, water, nature, bridges & tunnels, and other. The cycle roads contained most data specifications, which is a cycle that expects to be the toughest cycle when implementing the data specifications.

Sometimes, the translation of the implementation rules will result in some vague descriptions (e.g. 'most important buildings'). Other data specifications were based on attributes or attribute values, which were not available in the BGT. In order to formulate the data specifications as formal as possible, it is necessary to implement these data specifications in a model. Only then, those vague descriptions can be recognized and parameters can be added.

12.1.3. THE IMPLEMENTATION OF THE DATA SPECIFICATIONS IN AN AUTOMATIC GENERALIZATION SYSTEM

With the help of ModelBuilder in ArcGIS, I have created a model including most data specifications. The main purpose of the model is to test those data specifications and to investigate if a midscale BGT can be derived from the (uniform) BGT dataset. It was found that most data specifications are able to be implemented in an automatic generalization model based on the required data of the BGT. Also, it was possible to complete most of the data specifications with 'vague descriptions' with the parameters resulting from the model. Sometimes, it even occurred that the other parameters specified in the implementation rules were formulated too difficult, because easier parameters were resulting in similar results than those indicated in the implementation rules of TOP10NL. For example, the elimination of buildings with an area larger than $9m^2$ was easier than the elimination of buildings with areas of 3x3m, but these eliminations resulted in similar maps.

This study focused on the data specifications, which can be used for model generalization. This means that the data specifications for cartographic generalization are not yet identified. In addition, the model developed in ArcGIS focused on the purpose to test the data specifications, which means that this model is not yet showing the final results of a midscale dataset. The main struggles within the model were to create a Polygon to Line model, and to identify which objects were parallel and which objects were freely situated. In addition, a better solution should be found to create a perfectly logical consistent dataset.

12.1.4. ANALYZING THE MIDSCALE BGT

Visually, the generalization of the uniform BGT into a midscale BGT has resulted in a dataset with a lot of inconsistencies and differences comparing to TOP10NL. Due to the focus on model generalization and because the model is not yet perfect, many objects are not simplified and there are still holes in the resulting maps. However, when we are looking more into the details of the data per cycle, also many similarities with TOP10NL can be noticed. Especially, the cycle buildings is interesting to outline. Here, many similarities can be recognized within the buildings of the midscale BGT comparing to those of the TOP10NL. Although buildings as subsurface were used instead of buildings at an aerial perspective and different parameters were used, it still resulted in a similar amount of buildings as in TOP10NL.

However, not only visually, but also in the data behind the objects also enormous differences have been noticed, due to differences in semantics. It seems that most attributes and attribute values are returning in the midscale BGT, but sometimes with a very slightly different meaning. In addition, the data, which is not available in the midscale BGT compared to TOP10NL, can be acquired from additional sources or can be automatically generated. And therefore, it might be interesting to consider this dataset as replacement of TOP10NL. Especially within the cycle 'other', the attribute values were so differently specified that I decided to generalize only those objects, which were available in the required BGT.

In general, it seems that the midscale BGT can eventually be used as input within the Dutch key registrations by using data specifications from a BGT-perspective. However, users should identify which objects are relevant and which information should be maintained in the different topographical key registrations. With more information about the use and implementation of the topographical key registrations, considered decisions can be made about what objects are relevant and which information should stay in the different topographical key registrations. In addition, decisions can be made about answering questions like: what is the best way to derive those data? And what is the best way to implement those data in the system of key registrations? Therefore, it is interesting to put all attribute values in one source data, instead of adding different types of data to different types of scales. Then, these attribute values should be more similar at every scale and there should be no doubts about the meaning of every attribute value.

When considering the BGT as input dataset, instead of the commonly used TOP10NL, one topographical key registration can be created within the system of key registrations in the Netherlands. By deriving all mid- and smaller scales automatically from the BGT, the BRT does not have to be collected anymore. Despite the enormous consequences, this is also the chance to refresh the current TOP10NL with an entire new dataset, and to renew the topographical key registrations in the Netherlands. And after the integration of the BGT and the BRT, even an extension towards the BAG, derived from the BGT can be considered.

12.2. REFLECTION

The data specifications were developed following the template of Stoter et al. (2009a). In this template, different items were identified, such as geometry type, class, condition for object being concerned with this constraint and condition to be respected. This template was initially meant for cartographic generalization, although it can also be used as input for model generalization, on which this study has focused upon. The data specifications were formulated as formal as possible. However, the data specifications are not checked with the help of other evaluation methods than 'visual comparison of outputs' and therefore, this is a major limitation of this study.

Another limitation of this study is that my limited experience within the field of generalization can easily have an effect on the model, which I have developed. Therefore, this study can be continued by editing, refining and structuring the model in which the data specifications were tested. More experienced cartographers are able to add the missing data specifications, unclear sections, such as polygon to line, parallel and freely situated objects to the model. The model might also be rearranged in a good working sequence and the model might even be refined if possible.

12.3. FUTURE RESEARCH

In order to derive other datasets out of the BGT, some more studies are required. The first and main important step in this field is to contact BRT users and let them identify what kinds of datasets they really desire. Within these consultations, it is important to focus on the differences within TOP10NL and trying to identify which objects are important for the users to keep and which objects can be indicated as unnecessary within both the uniform and the midscale BGT. Actually, this important next step has already been set right after I have executed the methodology of my study, which resulted in some interesting thoughts about the usage of the BRT. However, it might also be interesting to find out if it is really necessary to require a logical consistent dataset or if it is more desired to change the entire system of topographical key registrations, for examples by using themes and network-based datasets. This feedback can be used to refine the data specifications.

In addition, this study can be continued by editing, refining and structuring the model in which the data specifications have been tested. In this model, my little experience within the field of generalization can easily be recognized. Therefore, more experienced cartographers could add the missing data specifications, unclear sections, such as polygon to line, parallel and freely situated objects into the model, the model could be rearranged in a good working sequence and the model could be refined if possible. Also, the cartographic elements could be added as much as possible.

When the BGT can be successfully generalized in a dataset comparable to TOP10NL, it might be interesting to look further than this dataset and to outline how to derive the smaller scales from this dataset. This again could be outlined in consideration with the users of these BRT datasets. Eventually, this could result in a successful integration of the large scale BGT and the mid- and small scale BRT. And it might even be interesting to complete this Dutch topographical key registration by researching how to derive the BAG from the BGT.

REFERENCES

Altena, V. van, J. Bakermans, B. Bruns, P. Lentjes, R. Nijhuis, M. Post, M. Reuvers, J. Stoter. (2013). High 5: Van BGT naar kleinschalige topografie *Kadaster* (pp. 1-58): Kadaster.

BAG BAO (2013). Processenhandboek BAG: basisregistraties adressen en gebouwen, versie 2013. Kadaster. Apeldoorn, november 2013.

Brink, L. v. d., Krijtenburg, D., Eekelen, H. v., Maessen, B., & Reuvers, M. (2012). Basisregistratie Grootschalige Topografie: Objectenhandboek 1.1 (pp. 1-346): Ministerie van Infrastructuur en Milieu, Geonovum.

Brink, L. v. d., Krijtenburg, D., Eekelen, H. v., & Maessen, B. (2013). Basisregistratie grootschalige topografie Gegevenscatalogus BGT (Vol. 1.1.1, pp. 1-91): Ministerie van Infrastructuur en Milieu, Geonovum.

Buttenfield, B. P., & McMaster, R. B. (1991). *Map Generalization: Making rules for knowledge representation*: Longman Scientific & Technical New York.

Cecconi, A. (2003). *Integration of cartographic generalization and multi-scale databases for enhanced web mapping* (Doctoral dissertation, Universität Zürich).

Choe, B. N., & Kim, Y. G. (2007). Framework and workflows for spatial database generalization. *Transactions in GIS, 11*(1), 101-114.

E-overheid (2014). Basisregistraties [Online]. http://www.e-overheid.nl/onderwerpen/stelselinformatiepunt/stelsel-van-basisregistraties/basisregistraties [Cited on March 7th, 2014].

ESRI (2011). ArcGIS Resource Center - Polygon to Centerline. [Online]. http://resources.arcgis.com/gallery/file/geoprocessing/details?entryID=EF0C96FF-1422-2418-7F9F-B0A8839FC796 [Cited on May 22th 2014].

ESRI (2014). ArcGIS Help 10.1 [Online]. http://resources.arcgis.com/en/help/main/10.1/ [Cited on February 13th, 2014].

ESRI (2014b). GIS Dictionary. [Online]. http://support.esri.com/en/knowledgebase/GISDictionary/search [Cited on March 11th, 2014].

Foerster, T., Stoter, J., & Kobben, B. (2007). *Towards a formal classification of generalization operators.* Paper presented at the Proceedings of the 23rd International Cartographic Conference.

Foerster, T., Stoter, J., & Kraak, M. J. (2010). Challenges for automated generalisation at European mapping agencies: A Qualitative and quantitative analysis. *Cartographic Journal, 47*(1), 41-54. doi: 10.1179/000870409X12525737905123.

Geonovum (2013). Toolkit BGT|IMGeo [Online]. http://www.geonovum.nl/onderwerpen/bgt-imgeo-standaarden/toolkit-bgtimgeo#voorbeeldbestand [Cited on: November 18th, 2013; July 7th, 2014].

Grunreich, D. (1985). Computer-Assisted Generalisation. In: Papers CERCO Cartography Course. Institut für Angewandte Geodäsie, Frankfurt a.M.

Hake, G., Grünreich, D., & Meng, L. (1994). *Kartographie. Visualisierung raum-zeitlicher Informationen.* Walter de Gruyter & co, Berlin, Germany.

Haunert, J.H. (2008). Aggregation in map generalization by combinatorial optimization. Ph.D. Thesis, Leibniz Universität Hannover.

ICA. (1973). *Multilingual dictionary of technical terms in cartography*: Franz Steiner Verlag GMBH - Wiesbaden / Germany.

Jonge, A. de (2014). Functionele specificaties TOP10Extra. Van TOP10NL naar TOP10Extra. Kadaster. Versie 1.2. 23 Januari 2014.

Kadaster. (2012). Verkenningsvoorschriften. Handleiding voor: Kadaster, Geo-informatie Proces Inwinning TOP10NL (2012.1 ed.).

Kadaster. (2013). Basisregistratie Topografie: Catalogus en Productspecificaties (pp. 1-233): Kadaster. (Reprinted from: Version 2.1).

Kazemi, S., S. Lim, C. Rizos (2004), A review of map and spatial database generalization for developing a generalization framework, School of Surveying & Spatial Information Systems, the University of New South Wales, Sydney, NSW 2052, Australia, PS WG IV/3 Generalization and Data Mining.

Lamy, S., A. Ruas, Y. Demezeau, M. Jackson. (1999). The Application of Agents in Automated Map Generalisation. . *Presented at the 19th ICA Meeting Ottawa Aug 14-21 1999*.

Ministerie van Infrastructuur en Milieu (2014). [Online]. http://www.bag.vrom.nl/basisregistraties-van-het-ministerie-van-infrastructuur-en-milieu/basisregistratie-topografie [Cited on July 24th 2014].

Müller, J. C., R. Weibel, J. P. Lagrange, F. Salgé. (1995). Generalization: state of the art and issues. In *GIS and generalization: methodology and practice* (pp. 3-17).

Nagel, R. (2014). Towards a TOP10NL derived from the BGT: An investigation of the semantic differences and user needs, MSc Thesis GIMA, 22-05-2014

Neun, M., Burghardt, D., & Weibel, R. (2009). Automated processing for map generalization using web services. *GeoInformatica, 13*(4), 425-452.

Oosterom, P. van (1995). The GAP-tree, an approach to "on-the-fly' map generalization of an area partitioning. *GIS and generalization: methodology and practice*, 120-132.

Roth, R. E., Brewer, C. A., & Stryker, M. S. (2012). A typology of operators for maintaining legible map designs at multiple scales. *Cartographic Perspectives*(68), 29-64.

Smaalen, J.W.N van, J. Stoter (2008). Automatische generalisatie van TOP10NL naar TOP50NL haalbaar? Geo-Informatie Nederland, Geo-INFO 2008-2.

Stoter, J. (2009). Towards one domain model and one key register topography.

Stoter, J., Burghardt, D., Duchêne, C., Baella, B., Bakker, N., Blok, C., . . . Schmid, S. (2009a). Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems, 33*(5), 311-324. doi: 10.1016/j.compenvurbsys.2009.06.002

Stoter, J., J. van Smaalen, N. Bakker, P. Hardy. (2009b). Specifying Map Requirements for Automated Generalization of Topographic Data. *The Cartographic Journal, 46*(3), 214-228. doi: 10.1179/174327709X446637.

Stoter, J. (2013). Rapport Aggregatie en de BGT (Definitief 1.0 ed.). Geonovum. Augustus 2013.

Thoreleifson, T. (2010). How to Generate GUIDs in ArcMap. Eagle Pipeline GIS Blog. *Eagle Information Mapping, Inc.* [Online]. http://eaglemap.com/blog/bid/45555/How-to-Generate-GUIDs-in-ArcMap [Cited on March 28th, 2014; May 2nd 2014].

Weibel, R. (1995). Three essential building blocks for automated generalization. In *GIS and generalization: methodology and practice* (pp. 56-69).

APPENDICES

A. INTERVIEW WITH EDWARD MAC GILLAVRY

Edward Mac Gillavry, Webmapper 23 December 2013, 14:00

Subject: how to visually aggregate the BGT

In this interview, Edward Mac Gillavry has been questioned about the visual aggregation of the BGT. Edward Mac Gillavry works at Webmapper, a cartographic developing and consulting company with its specialty in web cartography located at Amsterdam. Earlier, Edward Mac Gillavry was involved in the development of the BRT background map and now, the BGT needed to be visually aggregated. The development of the visual aggregation of the BGT needed to go from scratch and satisfy the National and European standards. Geonovum developed a manual for the web cartography and made a template in Excel to describe the filters, the attribute values and their visualizations (e.g. line sizes, colors, and levels of scale) for every object. These documents were used to create Standard Layer Descriptors in XML, which are used by the visualization of the BGT. Now, the required objects of the BGT are already visualized. The optional IMGeo objects are not yet visualized.

The standard visualizations are resembled to the visualizations of the BRT. In terms of web cartography, the background map of the BGT should look like the background map of the BRT and the BGT visualization should look like the visualization of the TOP10NL. By adding different layers of visualization on top of each other, the company was able to visually remove the virtual borders. This knowledge of visual aggregation is mainly based on Geoserver and other open source. This process has started in 2010. In 2015 the visualization will be finished. It should be nice when the generalization of the BGT into a midscale product will succeed. However it is a pity that this will be executed with ArcGIS' ModelBuilder. It would be really nice to have this as an open source, so everyone could benefit from the results of this study.

Everybody thought that the generalization of TOP10NL to TOP50 was not possible, but when you look what users really want, it did work out at the end. It is better to have up to date data which users are interested in, than to have data what is nice to look at. Therefore, TOP10NL may be cleaned a bit. For example, there are 800 municipality houses in TOP10NL, while there are fewer municipalities. And in the province of Zeeland, they added the names of farms, while they do not do that in another province. Another example is that TOP10NL was primarily based on defense, which means that hotels were only available when they had a specific amount of rooms that a battalion of soldiers could sleep there and gas stations were only displayed when you can access those with tanks. The EU encourages now the uniformness of the datasets with the help of INSPIRE.

I want to thank Edward Mac Gillavry for his time and contribution to this thesis.

B. BGT DATASETS COMPARED ON THEIR AVAILABLE REQUIRED ATTRIBUTE VALUES

Table B.1: The test datasets compared on their available required attribute values

Cycle	Object types	Attributes	Attribute values	Geometry	Amersfoort	Dronten	EZ_ Limburg	Maastricht	Valkenswaard	Venray	Combination
1.Roads	WEGDEEL	*functie*	ov-baan	polygon				●	●		●
			overweg	polygon						●	●
			spoorbaan	polygon	●					●	●
			baan voor vliegverkeer	polygon							● (red)
			rijbaan autosnelweg	polygon		●					●
			rijbaan autoweg	polygon	●	●					●
			rijbaan regionale weg	polygon		●				●	●
			rijbaan lokale weg	polygon	●	●		●	●	●	●
			fietspad	polygon	●	●		●	●	●	●
			voetpad	polygon	●	●		●	●	●	●
			voetpad op trap	polygon				●		●	●
			ruiterpad	polygon							● (red)
			parkeervlak	polygon	●	●		●	●	●	●
			voetgangersgebied	polygon	●	●				●	●
			inrit	polygon	●	●		●	●	●	●
			woonerf	polygon		●			●		●
			transitie			●				●	●
		fysiekVoorkomen	gesloten verharding	polygon	●	●		●	●	●	●
			open verharding	polygon	●	●		●	●	●	●
			half verhard	polygon	●	●		●	●	●	●
			onverhard	polygon		●		●	●	●	●
			transitie			●			●	●	●
		kruinlijn		line							● (red)
		opTalud	ja								● (red)
			nee		●	●		●	●	●	●
	ONDERSTEUNEND WEGDEEL	*functie*	verkeerseiland		●	●		●	●		●
			berm		●	●		●		●	●
		fysiekVoorkomen	gesloten verharding			●		●		●	●
			open verharding	polygon	●	●		●	●	●	●
			half verhard	polygon				●	●	●	●
			onverhard	polygon				●		●	●
			groenvoorziening	polygon	●	●		●	●	●	●
			transitie			●			●	●	●
		kruinlijn		line							● (red)
		opTalud	Ja					●			●
			Nee		●	●		●	●	●	●
	SPOOR	*functie*	trein	line	●						●
			sneltram	line							● (red)
			tram	line							● (red)
2.Buildings	PAND		Grondvlaksituatie BAGPND	multipolygon	●	●		●		●	●
3.Water	WATERDEEL	*type*	zee	polygon							● (red)
			waterloop	polygon	●	●		●	●	●	●
			watervlakte	polygon		●		●	●	●	●
			greppel, droge sloot	polygon		●		●	●	●	●
			transitie			●					●
	ONDERSTEUNEND WATERDEEL	*type*	oever, slootkant	polygon	●	●		●	●	●	●
			slik	polygon							● (red)
			transitie			●					●
4.Nature	BEGROEIDTERREINDEEL	*fysiekVoorkomen*	loofbos	polygon				●		●	●
			gemengd bos	polygon	●	●	●	●	●	●	●

			naaldbos	polygon						●	●
			heide	polygon			●		●	●	●
			struiken	polygon	●						●
			houtwal	polygon				●		●	●
			duin	polygon							● (red)
			moeras	polygon	●	●		●			●
			rietland	polygon							● (red)
			kwelder	polygon							● (red)
			fruitteelt	polygon			●	●		●	●
			boomteelt	polygon			●		●	●	●
			bouwland	polygon		●	●	●	●	●	●
			grasland agrarisch	polygon			●	●	●	●	●
			grasland overig	polygon		●	●	●		●	●
			groenvoorziening	polygon	●	●	●	●	●	●	●
			transitie			●			●	●	●
		kruinlijn		line							● (red)
		opTalud	ja		●			●			●
			nee		●	●	●	●	●	●	●
	ONBEGROEID TERREINDEEL	*fysiekVoorkomen*	erf	polygon	●	●		●	●	●	●
			gesloten verharding	polygon		●		●	●	●	●
			open verharding	polygon	●	●		●	●	●	●
			half verhard	polygon		●		●	●	●	●
			onverhard	polygon	●	●		●	●	●	●
			zand	polygon		●					●
			transitie						●	●	●
		kruinlijn		line							● (red)
		opTalud	ja								● (red)
			nee		●	●		●	●	●	●
5.Bridges & Tunnels	OVERBRUGGINGSDEEL	-	Overbruggingsdeel		●	●		●	●	●	●
	TUNNELDEEL	-	Tunneldeel		●					●	●
6.Other	OVERIGBOUWWERK	*Type*	overkapping	Multipolygon	●					●	●
			open loods	polygon							● (red)
			opslagtank	polygon							● (red)
			bezinkbak	polygon					●		●
			windturbine	polygon		●					●
			lage trafo	polygon				●			●
			bassin	polygon				●	●		●
	KUNSTWERKDEEL	*Type*	hoogspanningsmast	Multipoint/ Multipolygon	●						●
			gemaal	polygon							● (red)
			perron	polygon							● (red)
			sluis	polygon					●		●
			strekdam	polygon							● (red)
			steiger	polygon				●			●
			stuw	line/ polygon		●		●			●
			transitie			●			●		●
	SCHEIDING	*Type*	muur	line/ polygon	●			●	●	●	●
			kademuur	line/ polygon							● (red)
			damwand	line		●				●	●
			geluidsscherm	line	●						●
			walbescherming	line	●			●			●
			hek	line	●			●			●
	ONGECLASSIFICEERDOBJECT	-	OngeclassificeerdObject	Polygon	●						●
	FUNCTIONEELGEBIED		kering	polygon							● (red)
TOTAAL					40	50	9	51	44	55	81

C. REQUIRED AND OPTIONAL DATA OF THE BGT

Table C.1: Required data of the BGT

BGT object types	BGT attributes	BGT attribute values
WEGDEEL	*functie*	ov-baan, overweg, spoorbaan, baan voor vliegverkeer, rijbaan autosnelweg, rijbaan autoweg, rijbaan regionale weg, rijbaan lokale weg, fietspad, voetpad, voetpad op trap, ruiterpad, parkeervlak, voetgangersgebied, inrit, woonerf, transitie
	fysiekVoorkomen	gesloten verharding, open verharding, half verhard, onverhard, transitie
	kruinlijn	
	wegdeelOpTalud	Ja, nee
ONDERSTEUNEND WEGDEEL	*functie*	Verkeerseiland, berm
	fysiekVoorkomen	gesloten verharding, open verharding, half verhard, onverhard, groenvoorziening, transitie
	kruinlijn	
	ondersteunendWegdeelOpTalud	Ja, nee
SPOOR	*functie*	Trein, sneltram, tram
PAND		Grondvlaksituatie BAGPND
WATERDEEL	*type*	Zee, waterloop, watervlakte, greppel/droge sloot, transitie
ONDERSTEUNEND WATERDEEL	*type*	Oever/slootkant, slik, transitie
BEGROEIDTERREINDEEL	*fysiekVoorkomen*	Loofbos, gemengd bos, naaldbos, heide, struiken, houtwal, duin, moeras, rietland, kwelder, fruitteelt, boomteelt, bouwland, grasland agrarisch, grasland overig, groenvoorziening, transitie
	kruinlijn	
	BegroeidTerreindeelOpTalud	Ja, nee
ONBEGROEID TERREINDEEL	*fysiekVoorkomen*	Erf, gesloten verharding, open verharding, half verhard, onverhard, zand, transitie
	kruinlijn	
	OnbegroeidTerreideelOpTalud	Ja, nee
OVERBRUGGINGSDEEL	-	Overbruggingsdeel
TUNNELDEEL	-	Tunneldeel
OVERIGBOUWWERK	*Type*	Overkapping, open loods, opslagtank, bezinkbak, windturbine, lage trafo, bassin, niet-bgt
KUNSTWERKDEEL	*Type*	Hoogspanningsmast, gemaal, perron, sluis, strekdam, steiger, stuw, niet-bgt, transitie
SCHEIDING	*Type*	Muur, kademuur, damwand, geluidswand, walbescherming, hek, niet-bgt
FUNCTIONEELGEBIED		Kering, niet-bgt
ONGECLASSIFICEERDOBJECT	-	OngeclassificeerdObject

Table C.2: Optional object types and their attributes and attribute values

Object type	Attribute	Attribute values	Geometry
BAK	*Plus-type*	afval apart plaats, afvalbak, drinkbak, bloembak, zand-/zoutbak, container	Point
BORD	*Plus-type*	Informatiebord, plaatsnaambord, straatnaambord, verkeersbord, scheepvaartbord, verklikker, transportleiding, reclamebord, wegwijzerwaarschuwingshek, dynamische snelheidsindicator	Point
GEBOUWINSTALLATIE	*Plus-type*	Bordes, luifel, toegangstrap	Polygon
INSTALLATIE	*Plus-type*	Pomp, zonnepaneel	Point
KAST	*Plus-type*	CAI-kast, elektrakast, gaskast, telecomkast, rioolkast, openbare verlichtingkast, verkeersregelinstallatiekast, telkast, GMS kast	Point
MAST	*Plus-type*	bovenleidingmast, laagspanningsmast, straalzender, zendmast, radarmast	Point
OVERIGESCHEIDING	*Plus-type*	- Typen zoals scheiding -	Point / Line
PAAL	*Plus-type*	lichtmast, telpaal, portaal, verkeersregelinstallatiepaal, verkeersbordpaal, polder, haltepaal, vlaggenmast, afsluitpaal, praatpaal, hectometerpaal, dijkpaal, drukknoppaal, grensmarkering, sirene	Point
PUT	*Plus-type*	Benzine-/olieput, brandkraan/-put, drainageput, gasput, inspectie-/rioolput, kolk, waterleidingput	Point
SENSOR	*Plus-type*	Camera, debietmeter, hoogtedetectieapparaat, detectielus, weerstation, flitser, waterstandmeter, windmeter, lichtcel, GMS sensor, radar detector	Point
STRAATMEUBILAIR	*Plus-type*	Abri, bolder, brievenbus, fietsenrek, kunstobject, openbaar toilet, slagboom, speelvoorziening, telefooncel, bank, picknicktafel, fontein, lichtpunt, parkeerbeugel, betaalautomaat, reclamezuil, fietsenkluis, herdenkingsmonument	Point
WATERINRICHTINGSELEMENT	*Plus-type*	Remmingswerk, betonning, geleidewerk, vuilvang, meerpaal, hoogtemerk	Line / Point
WEGINRICHTINGSELEMENT	*Plus-type*	Molgoot, lijnafwatering, wegmarkering, wildrooster, rooster, geleideconstructie, balustrade, boomspiegel, verblindingswering	Point / line /polygon
VEGETATIEOBJECT	*Plus-type*	Boom, haag	Point / line /polygon
REGISTRATIEFGEBIED - BUURT / OPENBARERUIMTE/ STADSDEEL / WATERSCHAP/ WIJK		-	

Table C.3: Optional attributes and attribute values within required object types

Object type	Optional attributes	Optional attribute values	In Amersfoort	In Maastricht	In Dronten
WEGDEEL	*Plus-functie*	Verbindingsweg, calamiteitendoorsteek, verkeersdrempel	Calamiteitendoorsteek, verkeersdrempel	-	Calamiteitendoorsteek, verkeersdrempel
	Plus-fysiekVoorkomen	asfalt, cementbeton, betonstraatstenen, gebakken klinkers, tegels, sierbestrating, beton element, grasklinkers, schelpen, puin, grind, gravel, boomschors, zand,	Asfalt, gebakken klinkers, tegels, beton element, grasklinkers, schelpen, grind	Asfalt, cementbeton, betonstraatstenen, tegels, sierbestrating, beton element, grasklinkers, grind, zand, bitumen, asfalt rood	Asfalt, beton element, betonstraatstenen, cementbeton, gebakken klinkers, grasklinkers, gravel, tegels,
ONDERSTEUNEND WEGDEEL	*Plus-functie*	-	-		-
	Plus-fysiekVoorkomen	Asfalt, cementbeton, betonstraatstenen, gebakken klinkers, tegels, sierbestrating, beton element, grasklinkers, schelpen, puin, grind, gravel, boomschors, zand, bosplantsoen, gras- en kruidachtigen, planten, struikrozen, heesters, bodembedekkers	Gras- en kruidachtigen	Cementbeton, betonstraatstenen, tegels, sierbestrating, grind, zand, bosplantsoen, gras, heesters, bodembedekkers, blokhaag	Gras- en kruidachtigen
SPOOR	*Plus-functie*	(haven)kraan	-	-	-
PAND	-	-	-	-	-
WATERDEEL	*Plus-type*	Rivier, sloot, kanaal, beek, gracht, bron, haven, meer/plas/ven/vijver	-	-	kanaal
ONDERSTEUNEND WATERDEEL	*Plus-type*	-	-	-	-
BEGROEID TERREINDEEL	*Plus-fysiekVoorkomen*	Griend en hakhout, open duinvegetatie, gesloten duinvegetatie, laagstam boomgaarden, hoogstam boomgaarden, wijngaarden, klein fruit, akkerbouw, braakliggend, vollegrondsteelt, bollenteelt, bosplantsoen, gras- en kruidachtigen, planten, struikrozen, heesters, bodembedekkers		Akkerbouw, blokhaag, bodembedekkers, boomplantvak, bosplantsoen, bronhouder: eli, gazon, gras- en kruidachtigen,haag, planten, recreatie, sierheesters, speeltuin, sportterrein: voetbalveld, struikrozen, vollegrondsteelt: volkstuin, wisselplantsoen	Bodembedekkers, bosplantsoen, gras- en kruidachtigen, heesters, planten
ONBEGROEID TERREINDEEL	*Plus-fysiekVoorkomen*	Asfalt, cementbeton, kunststof, betonstraatstenen, gebakken klinkers, tegels, sierbestrating, beton element, grasklinkers, schelpen, puin, grind, gravel, boomschors, zand, strand en strandwal, zandverstuiving	Zand	Asfalt, bedrijvigheid, begraafplaats, betonstraatstenen, cementbeton, gravel, grind, kunststof, school, sierbestrating, tegels, zand	Kunststof, zand
OVERBRUGGINGS DEEL	*hoortByType*	Brug, aquaduct, viaduct, ecoduct, fly-over	Brug	-	
	typeOverbruggingsdeel	Dek, landhoofd, pijler, sloof, pylon		-	Dek, landhoofd, pijler, sloof
TUNNELDEEL		-		-	
OVERIGBOUWWERK		Bunker, voedersilo, schuur	Schuur	-	-
KUNSTWERKDEEL		Keermuur, overkluizing, duiker, faunavoorziening, vispassage, bodemval, coupure, ponton, voorde	Duiker	-	Duiker
SCHEIDING		Draadraster, faunaraster		-	-
FUNCTIONEEL GEBIED		Bedrijvigheid, natuur en landschap, landbouw, bewoning, infrastructuur verkeer en vervoer, infrastructuur waterstaatwerken, waterbergingsgebied, maatschappelijke en/of publieksvoorziening, recreatie, begraafplaats, functioneel beheer, functioneel beheer: hondenuitlaatplaats, recreatie: speeltuin, recreatie: park, recreatie: sportterrein, recreatie: camping, recreatie: bungalowpark, recreatie: volkstuin, bushalte, carpoolplaats, benzinestation, verzorgingsplaats	Natuur en landschap	Bedrijvigheid, begraafplaats, maatschappelijke en/of publieksfunctie, recreatie, recreatie: park, recreatie: speeltuin, recreatie: sportterrein, recreatie: sportterrein voetbalveld, recreatie: volkstuin	-

D. DATA SPECIFICATIONS OF THE UNIFORM BGT

Table D.1: Data specifications of the uniform BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Constrained property	Condition depends on initial value?	Condition to be respected	Action	Importance of constraint	Exception	Remarks
RdsA1	Aggregation	Polygon	WEGDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *functie* - *fysiekVoorkomen*; - *wegdeelOpTalud*; and - *relatieveHoogteligging*	Aggregate	5		Add *bronhouder* and *bgt-status* to condition to be respected
RdsA2	Aggregation	Polygon	ONDERSTEUNENDWEGDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *functie*; - *fysiekVoorkomen*; - *ondersteunendWegdeelOpTalud*; and - *relatieveHoogteligging*	Aggregate	5		Add *bronhouder* and *bgt-status* to condition to be respected
RdsA3	Aggregation	Line	SPOOR	-	Spatially neighbouring objects	yes	Objects with the same values on: - *functie*; and - *relatieveHoogteligging*	Aggregate	4		Add *bronhouder* and *bgt-status* to condition to be respected
BldsA1	Aggregation	Polygon	PAND	-	Spatially neighbouring objects	yes	Objects with the same values on: - *identificatieBAGPND*; and - *relatieveHoogteligging*	Aggregate	1		Should not be aggregated
NtrA1	Aggregation	Polygon	WATERDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *type*; and - *relatieveHoogteligging*	Aggregate	5		Add *bronhouder* and *bgt-status* to condition to be respected
NtrA2	Aggregation	Polygon	ONDERSTEUNENDWATERDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *type*; and - *relatieveHoogteligging*	Aggregate	5		Add *bronhouder* and *bgt-status* to condition to be respected
NtrA3	Aggregation	Polygon	BEGROEIDTERREINDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *fysiekVoorkomen*; - *begroeidTerreindeelOpTalud*; and	Aggregate	5		Add *bronhouder* and *bgt-status* to

							- *relatieveHoogteligging*			condition to be respected
NtrA4	Aggregation	Polygon	ONBEGROEID TERREINDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *fysiekVoorkomen*; - *onbegroeidTerreindeelOpTalud*; and - *relatieveHoogteligging*	Aggregate	5	Add *bronhouder* and *bgt-status* to condition to be respected
B&TA1	Aggregation	Polygon	TUNNELDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *relatieveHoogteligging*	Aggregate	5	Add *bronhouder* and *bgt-status* to condition to be respected
B&TA2	Aggregation	Polygon	OVERBRUGGINGSDEEL	-	Spatially neighbouring objects	yes	Objects with the same values on: - *relatieveHoogteligging*	Aggregate	5	Add *bronhouder* and *bgt-status* to condition to be respected
OthA1a	(class) selection	Polygon	FUNCTIONEEL GEBIED	-	-	Yes	Objects with attribute values 'niet-bgt' within *type*	Eliminate	4	
OthA1b	Aggregation	Polygon	FUNCTIONEEL GEBIED	objects with attribute value 'niet-bgt' should be eliminated	Spatially neighbouring objects	No	Objects with the same values on: - *type*; and - *relatieveHoogteligging*	Aggregate	3	Add *bronhouder* and *bgt-status* to condition to be respected
OthA2a	(class) selection	Line	KUNSTWERKDEEL	-	-	Yes	Objects with attribute values 'niet-bgt' within *type*	Do not display	3	
OthA2b	Aggregation	Line	KUNSTWERKDEEL	objects with attribute value 'niet-bgt' should be eliminated	Spatially neighbouring objects	No	Objects with the same values on: - *type*; and - *relatieveHoogteligging*	Aggregate	3	Add *bronhouder* and *bgt-status* to condition to be respected
OthA3a	(class) selection	Polygon	OVERIGBOUWWERK	-	-	Yes	Objects with attribute values 'niet-bgt' within *type*	Do not display	3	
OthA3b	Aggregation	Polygon	OVERIGBOUWWERK	objects with attribute value 'niet-bgt' should be eliminated	Spatially neighbouring objects	No	Objects with the same values on: - *type*; and - *relatieveHoogteligging*,	Aggregate	3	Add *bronhouder* and *bgt-status* to condition to be respected
OthA4a	(class) selection	Line	SCHEIDING	-	-	Yes	Objects with attribute values 'niet-bgt' within *type*	Do not display	3	

OthA4b	Aggregation	line	SCHEIDING	objects with attribute value 'niet-bgt' should be eliminated	Spatially neighbouring objects	No	Objects with the same values on: - *type*; and - *relatieveHoogteligging*,	Aggregate	3		Add bronhouder and bgt-status to condition to be respected
OthA5	Aggregation	Polygon	ONGECLASSIFICEERDOBJECT		Spatially neighbouring objects	Yes	Objects with the same values on: - *relatieveHoogteligging*	Aggregate	1		Should not be aggregated
GenB1	Data management	All	All aggregated objects	Aggregation (Generic ID A) needs to be executed.	-	No	IMGeo attributes: *objectBeginTijd*, *objectEindTijd*, *identificatie, tijdstipRegistratie, eindRegistratie, LV-publicatiedatum, bronhouder, inOnderzoek, relatieveHoogteligging, bgt-status*	Add IMGeo features to aggregated object types	4		
GenB2	Data management	All	All aggregated objects	Aggregation (Generic ID A) need to be executed and aggregated objects > 1	-	No	Random generate *identificatiecode* (sourceholder.randomString). Randomstring should have 32 characters, containing random letters (varying from a-f) and numbers (varying from 0-9)	Generate *identificatiecode*	4	Unchanged objects	
GenB3	Data management	All	All aggregated objects	Aggregation (Generic ID A) need to be executed and aggregated objects > 1	-	No	Generate *tijdstipRegistratie* with the date that the object changed (jjjj-mm-ddThh:min:sec)	Generate *tijdstipRegistratie*	4	Unchanged objects	
GenB4	Data management	All	All aggregated objects	Aggregation (Generic ID A) need to be executed and aggregated objects > 1	-	No	Generate *objectBeginTijd* with the date that the object is created and should look like jjjj-mm-dd	Generate *objectBeginTijd*	4	Unchanged objects	

Source: Stoter et al., 2009a.

Figure E.1: Formal key registration requirements

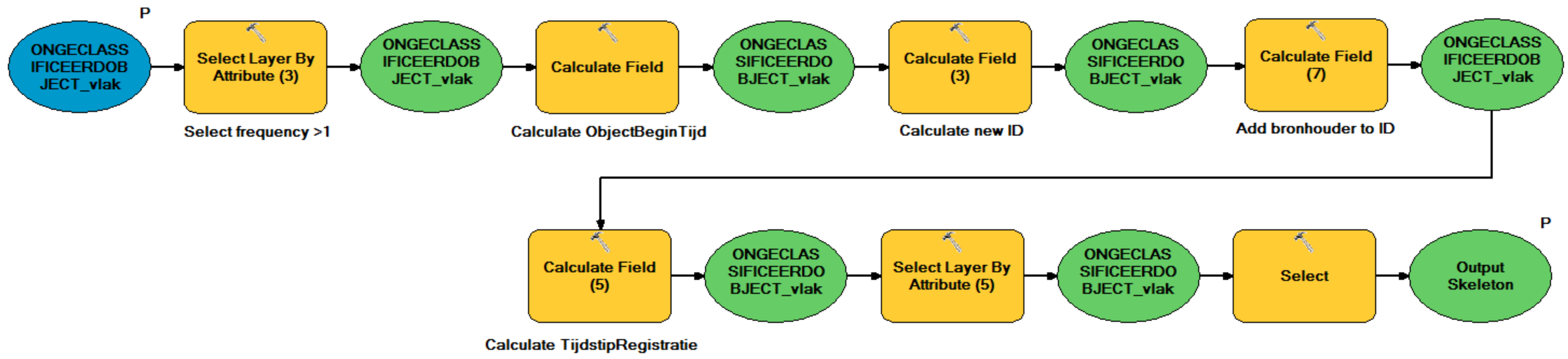


Figure E.2: PARTOFROAD

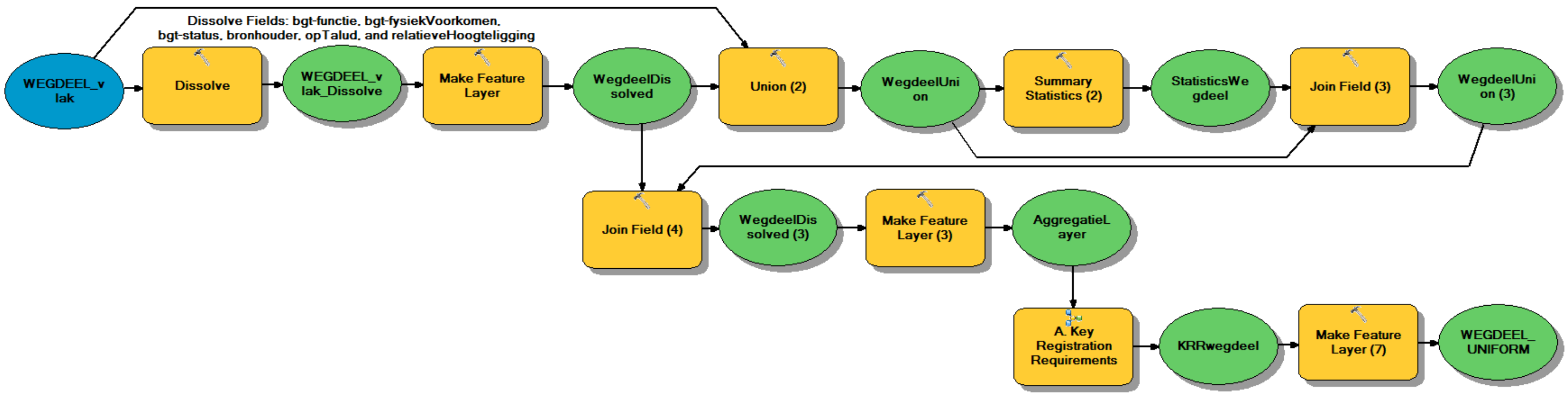


Figure E.3: SUPPORTIVEPARTOFROAD

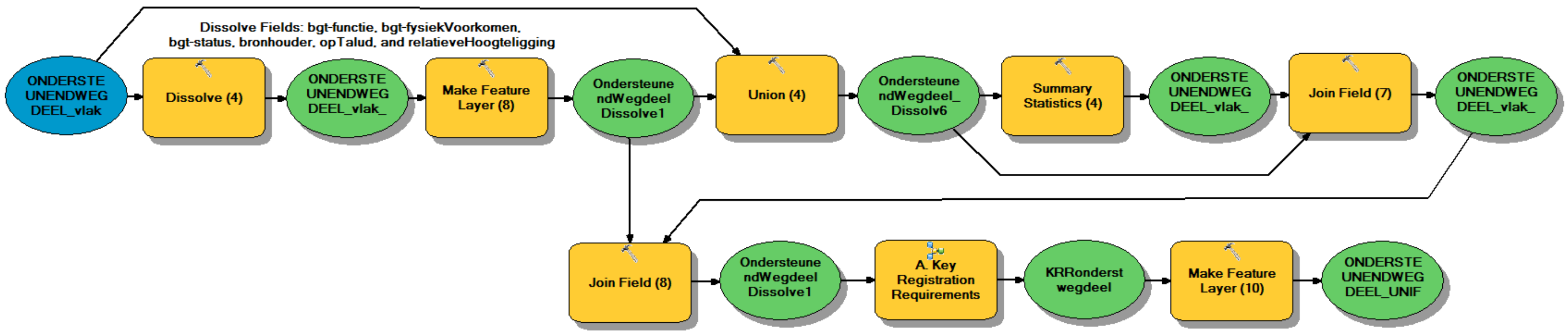


Figure E.4: RAILWAY

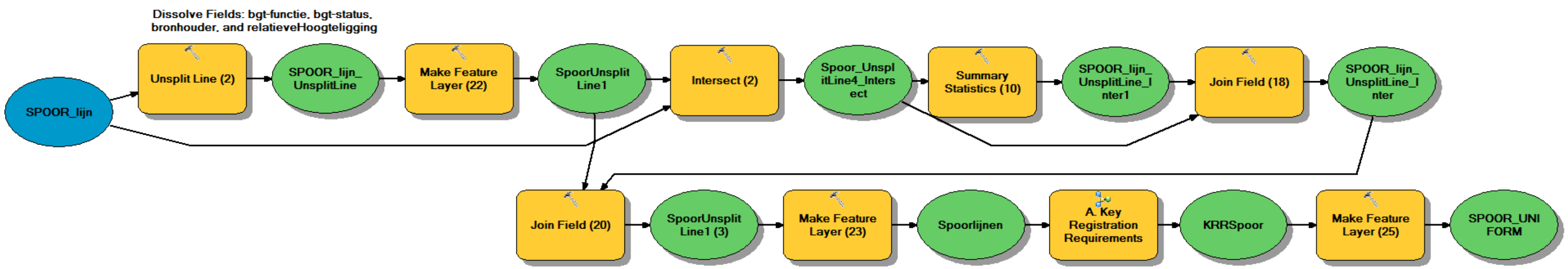


Figure E.5: BUILDING

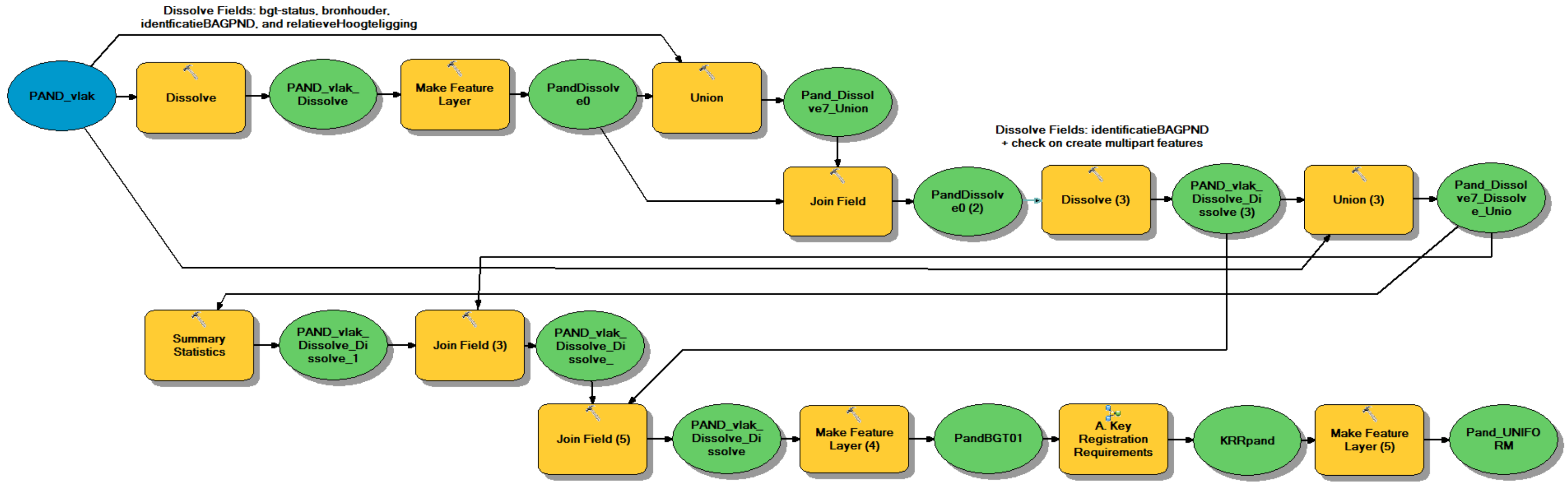


Figure E.6: PARTOFWATER

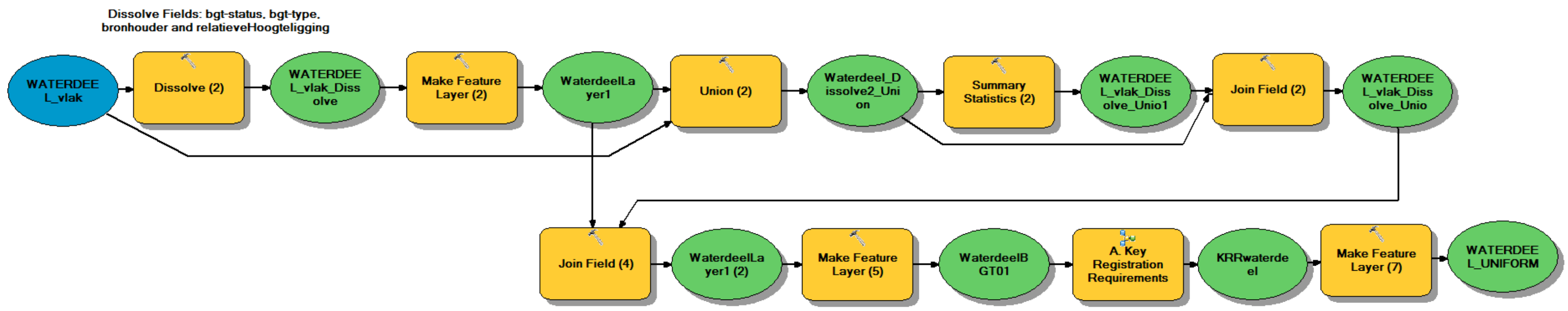


Figure E.7: SUPPORTIVEPARTOFWATER

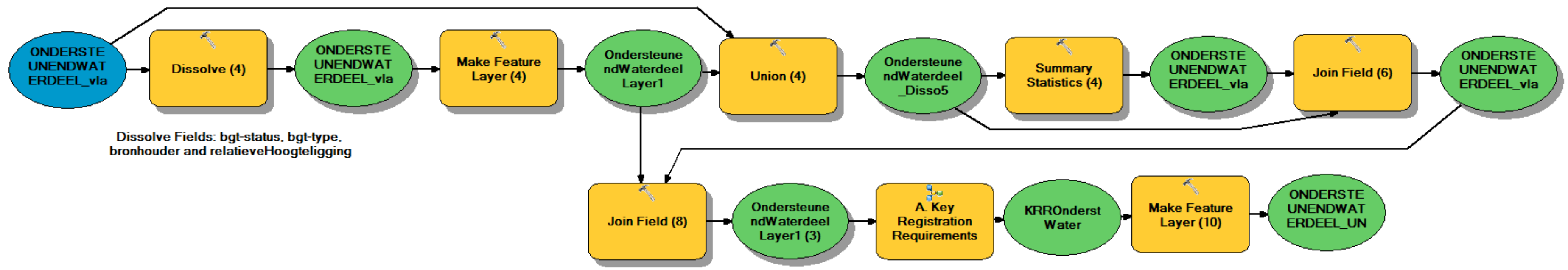


Figure E.8: COVEREDPARTOFTERRAIN

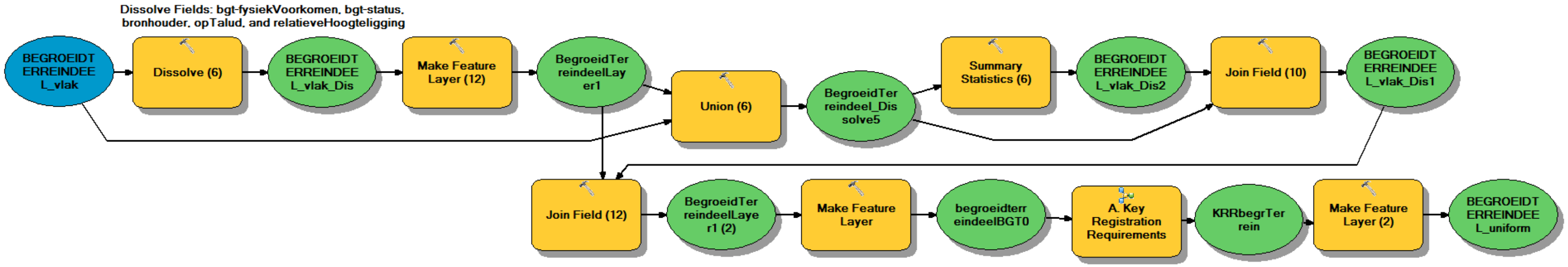


Figure E.9: UNCOVEREDPARTOFTERRAIN

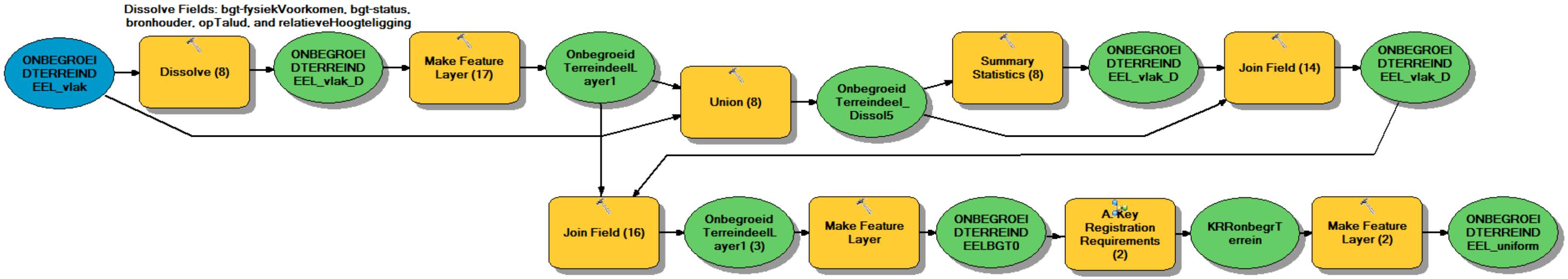


Figure E.10: PARTOFBRIDGE

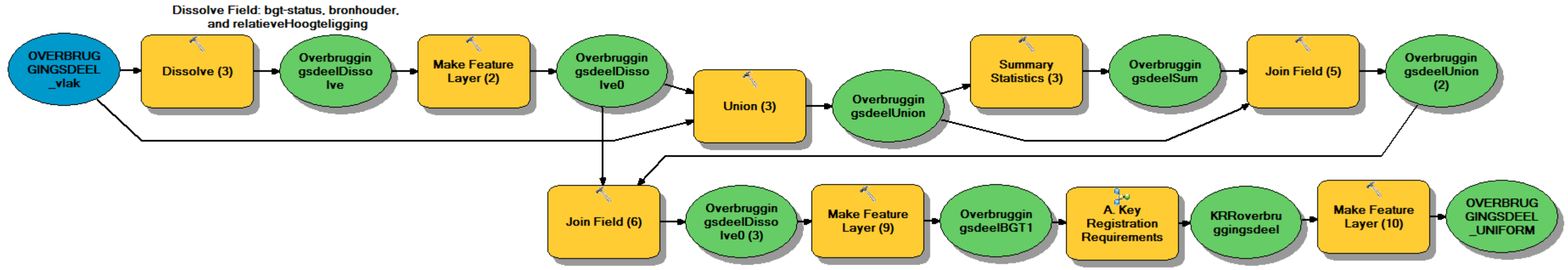


Figure E.11: PARTOFTUNNEL

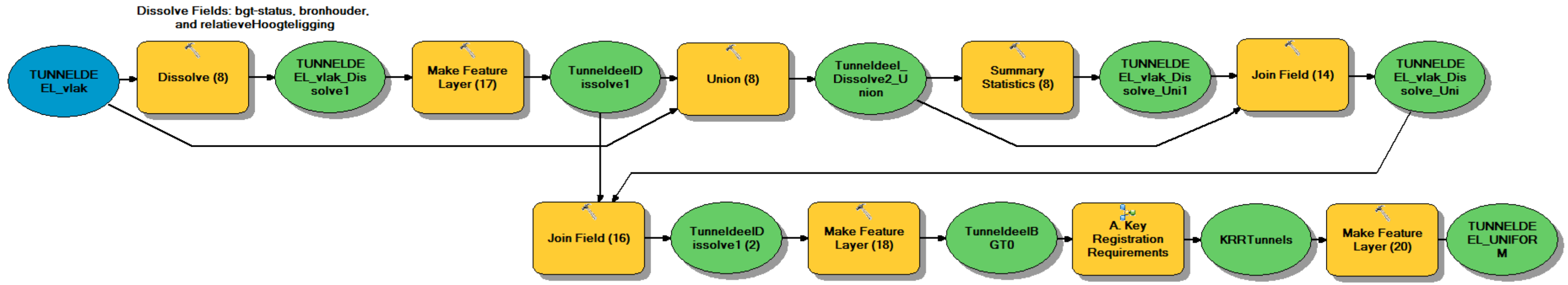


Figure E.12: FUNCTIONALAREA

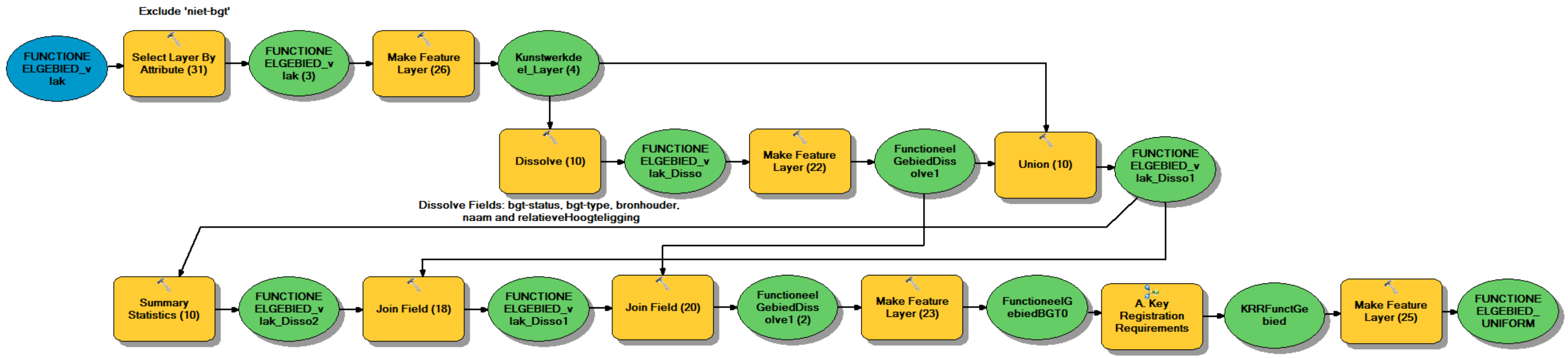


Figure E.13: ENGINEERINGSTRUCTURE_polygon

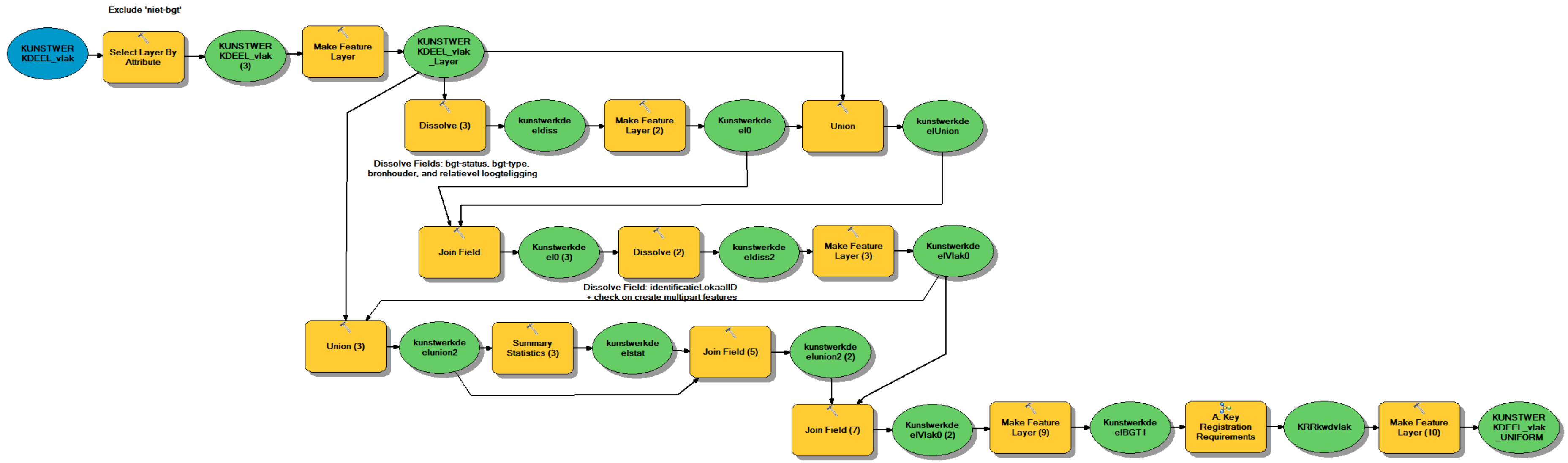


Figure E.14: ENGINEERINGSTRUCTURE_line

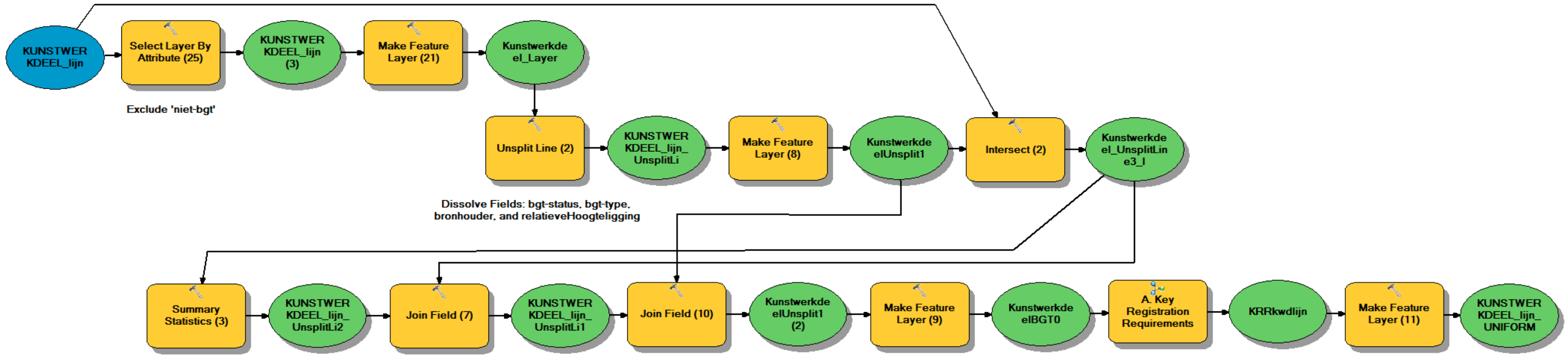


Figure E.15: REMAININGSTRUCTURE

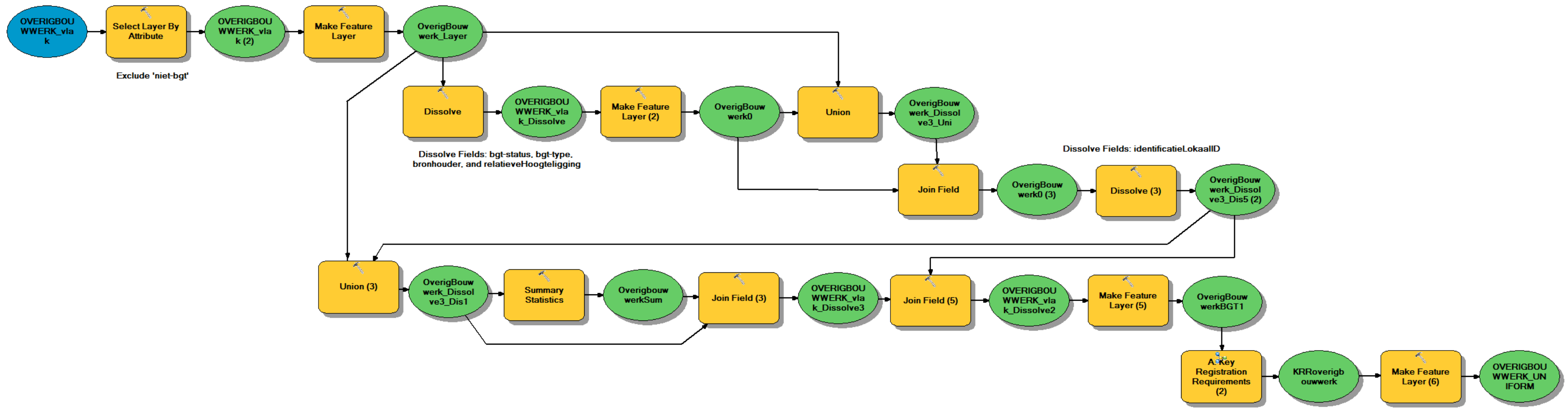


Figure E.16: SEPARATIONS_polygon

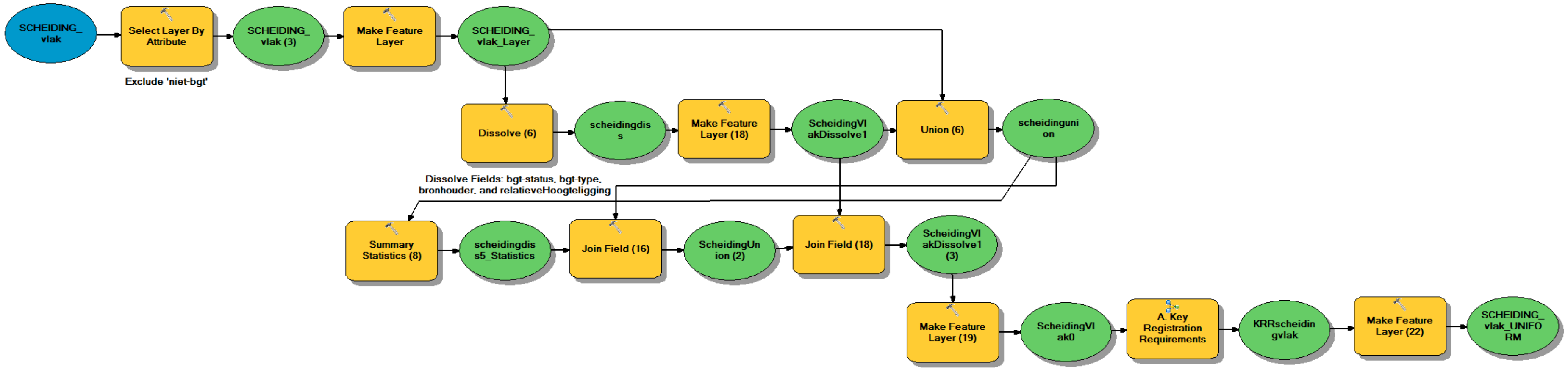
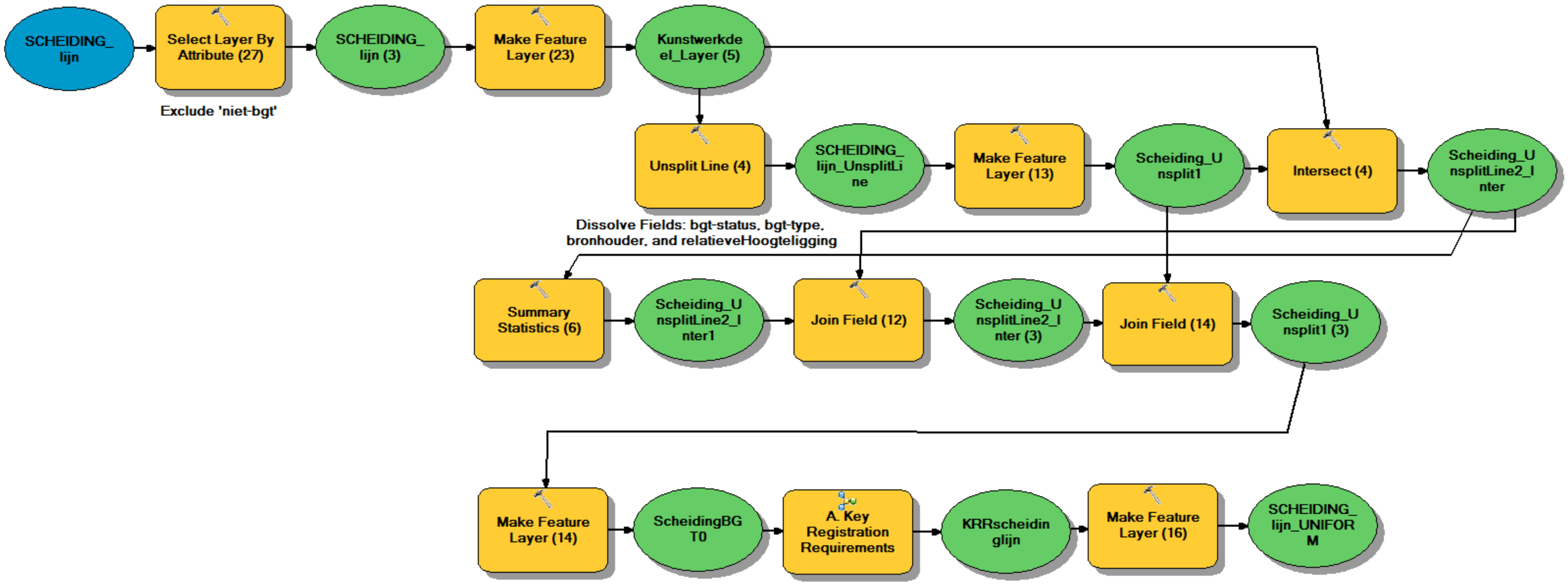


Figure E.17: SEPARATIONS_line



F. REQUIRED DATA OF THE BGT IN COMPARISON WITH THE REQUIRED DATA OF TOP10NL

Table F.1: BGT's required data and TOP10NL's required data per cycle

BGT object types	BGT attributes	BGT attribute values	BRT object types	TOP10 attributes	TOP10 attribute values
WEGDEEL	*Functie*	ov-baan, overweg, spoorbaan, baan voor vliegverkeer, rijbaan autosnelweg, rijbaan autoweg, rijbaan regionale weg, rijbaan lokale weg, fietspad, voetpad, voetpad op trap, ruiterpad, parkeervlak, voetgangersgebied, inrit, woonerf, transitie	WEGDEEL	*typeInfrastructuur*	Verbinding, kruising, overig verkeersgebied
	fysiekVoorkomen	gesloten verharding, open verharding, half verhard, onverhard, transitie		*typeWeg*	Autosnelweg, hoofdweg, regionale weg, lokale weg, straat, startbaan/landingsbaan, rolbaan/platform, overig, onbekend
	Kruinlijn			*hoofdverkeers-gebruik*	Snelverkeer, gemengd verkeer, busverkeer, fietsers/bromfietsers, voetgangers, ruiters, vliegverkeer, parkeren, parkeren: carpoolplaats, parkeren: P+R parkeerplaats, overig, onbekend
	OpTalud	Ja, nee		*gescheidenRijbaan*	Ja, nee
ONDERSTEUNEND WEGDEEL	*Functie*	Verkeerseiland, berm		*verhardingstype*	Verhard, half verhard, onverhard, onbekend
	fysiekVoorkomen	gesloten verharding, open verharding, half verhard, onverhard, groenvoorziening, transitie	SPOORBAANDEEL	*typeInfrastructuur*	Verbinding, kruising
	Kruinlijn			*typeSpoorbaan*	Trein, tram, metro, gemengd
	OpTalud	Ja, nee		*spoorbreedte*	Normaalspoor, smalspoor, gemengd
SPOOR	*Functie*	Trein, sneltram, tram		*aantalSporen*	<nummer>
PAND		Grondvlaksituatie BAGPND	GEBOUW	-	-
WATERDEEL	*type*	Zee, waterloop, watervlakte, greppel/droge sloot, transitie	WATERDEEL	*typeWater*	Waterloop, meer/plas/ven/vijver, greppel/droge sloot, zee, droogvallend, bron/wel, onbekend
ONDERSTEUNEND WATERDEEL	*type*	Oever/slootkant, slik, transitie		*hoofdafwatering*	Ja, nee
				functie	Drinkwaterbekken, haven, natuurbad, viskwekerij, vistrap, vloeiveld, waterval, waterzuivering, zwembad, overig, onbekend
				voorkomen	Met riet, overig
BEGROEIDTERREINDEEL	*fysiekVoorkomen*	Loofbos, gemengd bos, naaldbos, heide, struiken, houtwal, duin, moeras, rietland, kwelder, fruitteelt, boomteelt, bouwland, grasland agrarisch, grasland overig, groenvoorziening, transitie	TERREIN	*typeLandgebruik*	Aanlegsteiger, akkerland, basaltblokken/steenglooiing, bebouwd gebied, boomgaard, boomkwekerij, bos: gemengd bos, bos: griend, bos: loofbos, bos: naaldbos, dodenakker, dodenakker met bos, fruitkwekerij, grasland, heide, laadperron, populieren, spoorbaanlichaam, zand, overig, onbekend
	kruinlijn				
	opTalud	Ja, nee			
ONBEGROEID TERREINDEEL	*fysiekVoorkomen*	Erf, gesloten verharding, open verharding, half verhard, onverhard, zand, transitie			
	kruinlijn				
	opTalud	Ja, nee			

OVERBRUGGINGSDEEL	-	Overbruggingsdeel	*In WEGDEEL, WATERDEEL en TERREIN*	-	
TUNNELDEEL	-	Tunneldeel	*In WEGDEEL, WATERDEEL en TERREIN*	-	
OVERIGBOUWWERK	*Type*	Overkapping, open loods, opslagtank, bezinkbak, windturbine, lage trafo, bassin	INRICHTINGSELEMENT	*Type*	Aanlegsteiger, baak, bomenrij, boom, boorput, boortoren, BOS-pomp, brandtoren, dam/koedam, dukdalf, gaswinning, gedenkteken/monument, geluidswering, gemaal, golfmeetpaal, GPS kernnetpunt, grenspunt, heg/haag, hekwerk, helikopterlandingsplatform, hoogspanningsleiding, hunebed, kaap, kabelbaan, kabelbaanmast, kapel, kilometerpaal, kilometerpaal spoorweg, kilometerpaal water, kilometerraaibord, kilometerraaipaal, koepel, koeltoren, kogelvanger schietbaan, kraan, kruis, laadperron, leiding, licht/lichtopstand, lichttoren, luchtvaartlicht, markant object, muur, oliepompinstallatie, paal, paalwerk, peilmeetstation, peilschaal, pijler, radarpost, radiobaken, radiotelescoop, RD punt, schietbaan, schoorsteen, seinmast, sluisdeur, stormvloedkering, station, strandpaal, strekdam/krib/golfbreker, stuw, tol, toren, uitzichttoren, verkeersgeleider, visplaats, vlampijp, wegafsluiting, wegwijzer, windmolen, windmolen: watermolen, windmolen: korenmolen, windmolentje, windturbine, zeevaartlicht, zendmast, zichtbaar wrak, overig, onbekend
KUNSTWERKDEEL	*Type*	Hoogspanningsmast, gemaal, perron, sluis, strekdam, steiger, stuw, transitie			
SCHEIDING	*Type*	Muur, kademuur, damwand, geluidsscherm, walbescherming, hek			
FUNCTIONEELGEBIED		Kering	FUNCTIONEEL GEBIED	*Type*	Arboretum, bedrijventerrein, begraafplaats, boswachterij, bungalowpark, camping/parkeerterrein, caravanpark, circuit, crossbaan, dierentuin/safaripark, eendenkooi, emplacement, erebegraafplaats, gaswinning, gebied met hoge objecten, gebouwencomplex, golfterrein, grafheuvel, grindwinning, groeve, haven, heemtuin, helikopterlandingsterrein, infiltratiegebied, jachthaven, kartingbaan, kazerne/legerplaats, landgoed, mijn, mijnsteenberg, militair oefengebied/schietterrein, mosselbank, natuurgebied/natuurreservaat, oliewinning, openluchtmuseum, openluchttheater, park, pinetum, plantsoen, productie-installatie, recreatiegebied, renbaan, skibaan, slipschool, sluizencomplex, sportterrein/sportcomplex, stortplaats, tankbaan, tennispark, transformatorstation, tuincentrum, verzorgingsplaats, viskwekerij, vliegveld/luchthaven, volkstuinen, werf, wildwissel, windturbinepark, woonwagencentrum, ijsbaan, zandwinning, zenderpark, zoutwinning, zuiveringsinstallatie, zweefvliegveldterrein, zwembad complex, onbekend
ONGECLASSIFICEERDOBJECT	-	OngeclassificeerdObject	GEOGRAFISCH GEBIED	*Type*	Bank/ondiepte/plaat, bosgebied, buurtschap, duingebied, eiland, geul/vaargeul, heidegebied, heuvel/berg, huizengroep, kaap/hoek, meer/plas/ven/vijver, plaats/bewoond oord, polder, streek/veld, terp, vliedberg, wad, woonwijk, zee, zeegat/zeearm, overig, onbekend
			RELIEF	*Type*	Dieptelijn, dieptepunt, hoogtelijn, hoogtepunt, kade/wal, laagwaterlijn, peil, peil: winterpeil, peil: zomerpeil, talud/hoogteverschil, steile rand/aardrand, onbekend
			REGISTRATIEF GEBIED	*Type*	Land, provincie, gemeente, stadsdeel, wijk, buurt, waterschap, nationaal park, Bundesland, Regierungsbezirk, Kreis

G. DATA SPECIFICATIONS OF THE MIDSCALE BGT

Table G.1: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning generic constraints

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
Gen1	Data management	All	All (generic attributes)	Generalization to the new scale should be finalized	No	New developed objects in the new scale should contain new attribute values within the attributes *bronhouder, namespace, identificatiecode, objectBeginTijd,and tijdstipRegistratie.*	Generate new *bronhouder, namespace, identificatiecode, objectBeginTijd,and tijdstipRegistratie*.	-	-
Gen1a	Data management	All	All (*namespace)*	Generalization to the new scale should be finalized	No	Generate a new *namespace* for the midscale BGT	Generate *namespace*	-	-
Gen1b	Data management	All	All (*bronhouder)*	Generalization to the new scale should be finalized	No	Generate the source holder of the midscale BGT	Generate *bronhouder*	-	*Who is going to be the source holder of the midscale BGT?*
Gen1c	Data management		All *(identificatiecode)*	Generalization to the new scale should be finalized	No	Random generate *identificatiecode* following the format 'source holder.randomString'. RandomString should have 32 characters, containing random letters (varying from a-f) and numbers (varying from 0-9).	Generate *identificatiecode*	-	-
Gen1d	Data management		All *(objectBeginTijd)*	Generalization to the new scale should be finalized	No	Generate *objectBeginTijd* with the data that the object is created following the format 'jjjj-mm-dd'	Generate *objectBeginTijd*	-	-
Gen1e	Data management		All *(tijdstipRegistratie)*	Generalization to the new scale	No	Generate *tijdstipRegistratie* with the data and time that the object	Generate *tijdstipRegistratie*	-	-

				should be finalized		is changed following the format 'jjjj-mm-ddThh:min:sec'			
Gen2	(Re)classificatio n	All	WEGDEEL (classified road)	-	Yes	The attribute values 'rijbaan autosnelweg', 'rijbaan autoweg', and 'rijbaan regionale weg'	Together specified as 'classified road'	-	-
Gen3	(Re)classificatio n	All	- (harde topografie)	-	Yes	The object types WEGDEEL, WATERDEEL	Together specified as 'harde topografie'	Object types TERREIN (zachte topografie)	
Gen4	Amalgamation	Polygon	PAND (Built-up area)	-	Yes	Grouping Distance = 130m AND Minimum Area = 800,000m²	Specify built-up area	-	Specified based on the datasets of Dronten and Maastricht.

Table G.2: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning road constraints

Generic Constrai nt ID	Constraint Type	Geomet ry Type	Class	Condition for object being concerned with this constraint	Conditio n depends on initial value?	Condition to be respected	Action	Exception	Remarks
RdsA1	Collapse	Polygon	WEGDEEL	Measured without verges	Yes	Width <2m	Collapse into line objects	-	Width = MEAN width
RdsB1	Aggregation	Polygon	ONDERSTEUNEND-WEGDEEL ('berm')	Neighboring to objects in WEGDEEL, PAND, or WATERDEEL	Yes	Width <6m AND without a slope	Move to adjacent object in WEGDEEL	-	Width = MEAN width
RdsB2	Aggregation	Polygon	ONDERSTEUNEND-WEGDEEL ('berm')	NOT neighboring to objects in WEGDEEL, PAND, or WATERDEEL	Yes	Width < 6m AND without a slope	Move to TERREIN	-	Width = MEAN width
RdsB3	Aggregation	Polygon	ONDERSTEUNEND-WEGDEEL ('berm')	-	Yes	Width >6m OR with a slope	Move to TERREIN	-	Width = MEAN width
RdsB4	Aggregation	Polygon	ONDERSTEUNEND-WEGDEEL ('verkeerseiland')	-	Yes	Width <6m AND length >50m	Move to INRICHTINGSELE MENT	-	Width = MEAN width

RdsB5	Aggregation	Polygon	ONDERSTEUNEND-WEGDEEL ('verkeerseiland')	-	Yes	Width >6m AND length >50m	Move to TERREIN	-	Width = MEAN width
RdsC1	(Class) selection	Polygon	WEGDEEL ('fietspad')	Parallel	No	Width >2m AND outside built-up area (Gen4)	Display as polygonal object	-	Width = MEAN width
RdsC2	(Class) selection	Polygon	WEGDEEL ('fietspad')	Parallel	No	Width >2m AND inside built-up area (Gen4)	Do not display	Parallel to a classified road (Gen2)	Width = MEAN width
RdsC3	(Class) selection	Line	WEGDEEL ('fietspad')	Parallel	No	Width <2m AND outside built-up area (Gen4)	Display as line object	-	Width = MEAN width
RdsC4	(Class) selection	Line	WEGDEEL ('fietspad')	Parallel	No	Width <2m AND inside built-up area (Gen4)	Do not display	-	Width = MEAN width
RdsC5	(Class) selection	Polygon / line	WEGDEEL ('fietspad')	Vrijliggend	Yes	All	Display	-	-
RdsD1	(Class) selection	Polygon / line	WEGDEEL ('voetpad')	~~Vrijliggend~~	Yes	Length >100m	Display	-	-
RdsD2	(Class) selection	Polygon / line	WEGDEEL ('voetpad')		Yes	Around building blocks	Do not display	-	-
RdsD3	(Class) selection	Polygon / line	WEGDEEL ('voetpad')	Parallel	Yes	Part of a continuous road AND neighboring a non-classified road for a relatively short distance	Display	<250m	*relatively short distance* is set on >20m
RdsD4	(Class) selection	Polygon / line	WEGDEEL ('voetpad op trap')	-	Yes	Length <100m and not neighboring a 'voetpad'	Do not display	-	-
RdsD5	(class) selection	Polygon	WEGDEEL ('voetgangersgebied')	-	Yes	Extended 'voetgangersgebied'	Display as polygonal object	-	*Extended* is set on >1000m²
RdsD6	Aggregation	Polygon	TERREIN ('trottoir')	-	Yes	Width >6m AND in between two polygons with another attribute value	Display as TERREIN	-	Width = MEAN width
RdsE1	(Class) selection	Polygon	WEGDEEL ('parkeervlak')	-	Yes	<1000m	Do not display	Parking lots surrounded by forested areas AND of orientating value	-
RdsF1	(Re)classificati on	Polygon / line	WEGDEEL ('inrit')	-	Yes	Do not display as separate road type	Move to the adjacent road type highest in hierarchy	-	-

RdsF2	(Class) selection	Polygon / line	WEGDEEL ('inrit')	-	Yes	Located on a 'dijk' AND length >100m (also when in and exit driveways are connected)	Display	-	-
RdsG1	(Re)classification	Polygon / line	WEGDEEL ('rotonde'/'kruising')	-	No	'rotonde' or 'kruising' should get the highest classification of the neighboring road	Reclassify with same attribute values as adjacent object highest in hierarchy	Driveways ending in roundabout receive the attribute values of the continuing road	-
RdsG2	Collapse	Polygon	WEGDEEL ('kruising')	-	No	All crossings should be defined as point data with the same attributes/attribute values as its polygonal object in WEGDEEL	Create point data	-	-
RdsG3	(Class) selection	Polygon	WEGDEEL ('rotonde')	-	No	Middle of roundabouts of any size	Display as TERREIN	-	-
RdsG4	Collapse	Polygon	WEGDEEL	-	No	All roads with the same attributes/attribute values as its polygonal object in WEGDEEL	Create centerlines	-	-
RdsH1	(Class) selection	Polygon / line	WEGDEEL	-	No	Length <100m	Do not display	Roads with attribute value 'half verhard' and dead ended roads behind buildings should not be displayed when <250m. Roads <100 meters on parking lots should be displayed	
RdsI1	(Re)classification	Polygon / line	WEGDEEL ('open verharding' AND 'gesloten verharding')	-	Yes	Combine 'open verharding' AND 'gesloten verharding'	Combine into 'verhard'	-	-
RdsJ1	(Class) Selection	Polygon / line	WEGDEEL ('ruiterpad')	-	Yes	Width <2m	Do not display	-	Width = MEAN width
RdsJ2	(Class) selection	Polygon / line	WEGDEEL ('busbaan')	-	Yes	'busbaan' integrated within a road	Do not display separately	-	-

RdsJ3	(Class) selection	Polygon / line	WEGDEEL ('busbaan')	-	Yes	'busbaan' with its own road	Display separately	-	-
RdsJ4	(Class) selection	Polygon / line	WEGDEEL ('busbaan')	-	Yes	'busbaan' which is closed with a specific barrier	Do not display	-	-
RdsJ5	Collapse	Line	SPOOR	-	No	All roads with the same attributes/attribute values as its polygonal object in SPOORBAANDEEL	Create centerlines	-	-
RdsJ6	(Class) selection	Line	SPOOR (changing tracks)	-	Yes	Changing tracks <5km constant	Do not display	-	-
RdsJ7	(Class) selection	Line	SPOOR (temporary tracks)	Parallel	Yes	With separate trace	Display	-	*How to define temporary?*
RdsJ8	(Class) selection	Line	SPOOR (both 'enkelspoor' and 'dubbelspoor')	-	Yes	Length <500m	Do not display	-	-
RdsJ9	(Class) selection	Line	SPOOR ('wissels')	-	Yes	'wissels' within tracks	Do not display	-	-
RdsJ10	(Class) selection	Line	SPOOR ('dijk') and RELIEF	-	Yes	Track situated on 'dijk'	Mention relieflines	-	-

Table G.3: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning building constraints

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
Blds1	Aggregation	Polygon	PAND	-	Yes	Neighboring other buildings	Combine	-	-
Blds2	Amalgamation	Polygon	PAND	-	Yes	Not direct neighboring objects, but within a distance of 2 meters	Combine	Objects bounded by WEGDEEL or WATERDEEL	-
Blds3	(Class) selection	Polygon	PAND	-	No	Small buildings <3x3m or with a diameter <4m	Do not display	-	Changed into all buildings <9m²
Blds4	(Class) selection	Polygon	PAND	-	No	Buildings not visible from continuous road within built-up area AND area <50m²	Do not display	-	Changed into all buildings within built-up area <50m²
Blds5	(Class) selection	Polygon	PAND (patios or courtyards)	-	No	Area <1000m²	Do not display	-	-

Blds6	Simplification	Polygon	PAND (‘luifels’, ‘loopgangen’, ‘luchtbruggen’ ‘uitbouwsels’ etc.)	-	No	Width <3m OR area <3x3m	Do not display		Changed into <3m or <9m^2
Blds7	Simplification	Polygon	PAND (openings within buildings)	-	No	Length <3m or not public available	Do not display	-	Difficult to determine which openings are public available

Table G.4: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning water constraints

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
Wtr1	Aggregation	Polygon	ONDERSTEUNEND-WATERDEEL (‘oever/slootkant’)			The attribute value ‘oever/slootkant’ should be combined with WATERDEEL	Move to adjacent object in WATERDEEL	-	-
Wtr2	Collapse	Polygon	WATERDEEL	Objects can be split in multiple polygon and line objects	Yes	Width <6m	Change into line objects	-	-
Wtr3	(Class) selection	Polygon	WATERDEEL ‘greppel/droge sloot’	-	Yes	Width <50cm	Do not display	-	Changed into 50m^2
Wtr4	(Class) selection	Polygon	WATERDEEL ‘watervlakte’	-	Yes	Area <50m^2	Do not display	-	Changed in all WATERDEEL objects

Table G.5: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning nature constraints

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
Ntr1	(Re)classification	Polygon	BEGROEID-TERREINDEEL ('grasland agrarisch' and 'grasland overig')	-	Yes	Combine 'grasland agrarisch' AND 'grasland overig'	Combine into 'grasland'	-	-
Ntr2	Merge	Polygon	BEGROEID-TERREINDEEL and ONBEGROEID-TERREINDEEL	-	Yes	Combine two object types into TERREIN	Combine into TERREIN	-	-
Ntr3	Aggregation	Polygon	ONDERSTEUNEND-WATERDEEL ('slik')			The attribute value 'slik' should be combined with TERREIN	Move to adjacent object in TERREIN	-	-
Ntr4	(Class) selection	Polygon	TERREIN ('bos')	Within 'erf' or within built-up area (Gen4)	No	Area <1000m²	Do not display	-	-
Ntr5	Aggregation	Polygon / line	WEGDEEL & TERREIN (small roads towards 'erf')			Small roads towards 'erf' should be combined with 'erf'	Combine with 'erf'	-	-
Ntr6	(Class) selection	Polygon	TERREIN	Terrain objects bordered by 'zachte topografie' (Gen4)	No	Area <1000m²	Move to neighboring terrain objects	'bos' outside built-up area <50m²	-
Ntr7	(Class) selection	Polygon	TERREIN	Terrain objects bordered by 'harde topografie' (Gen4)	No	Does not have a minimum size to display	Display as separate objects	-	-

Table G.6: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning bridges & tunnels constraints

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
BT1	(Class) selection	Polygon	OVERBRUGGINGS-DEEL	-	No	Crossing line objects of WEGDEEL or WATERDEEL	Do not display	-	-
BT2	(Class) selection	Polygon	TUNNELDEEL	-	No	Crossing line objects of WEGDEEL or WATERDEEL	Do not display	-	-
BT3	(Class) selection	Polygon	OVERBRUGGINGS-DEEL	-	No	Bridges not connected to WEGDEEL or WATERDEEL	Do not display	-	-
BT4	(Class) selection	Polygon	TUNNELDEEL	-	No	Tunnels not connected to WEGDEEL or WATERDEEL	Do not display	-	-

Table G.7: Data specifications of the midscale BGT using the format of Stoter et al. (2009a) distinguishing between constraints on one object concerning other constraints

Generic Constraint ID	Constraint Type	Geometry Type	Class	Condition for object being concerned with this constraint	Condition depends on initial value?	Condition to be respected	Action	Exception	Remarks
Oth1	Collapse	Polygon	KUNSTWERKDEEL & OVERIGBOUWWERK	-	Yes	All polygonal objects should be changed into point objects	Change into point objects	-	-
Oth2	Merge	Point	KUNSTWERKDEEL & OVERIGBOUWWERK	Point objects should be created (OthA1)	No	Combine two datasets into INRICHTINGSELEMENT_punt	Merge into INRICHTINGSELEMENT_punt	-	Rename INRICHTINGS ELEMENT
Oth3	Collapse	Polygon	SCHEIDING_polygon	-	Yes	All polygonal objects should be changed into line objects	Change into line objects	-	-
Oth4	Merge	Line / Line	SCHEIDING_polygon and SCHEIDING_lijn	Line objects should be created (OthA3)	No	Combine SCHEIDING with existing SCHEIDING_lijn into INRICHTINGSELEMENT_lijn	Merge into INRICHTINGSELEMENT	-	Rename INRICHTINGS ELEMENT
Int1	Aggregate	Polygon	Resulting holes	*relatieveHoogteligging* = 0	No	Neighboring terrain objects	Change into neighboring terrain object	-	-
Int2	Aggregate	Polygon	Resulting holes	*relatieveHoogteligging* = 0	No	Not neighboring terrain objects	Change into neighboring object based on MaxLength	-	-

H. MODELS OF THE MIDSCALE BGT

GENERAL MODELS

Figure H.1: Built-up area

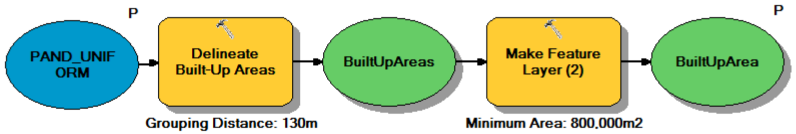
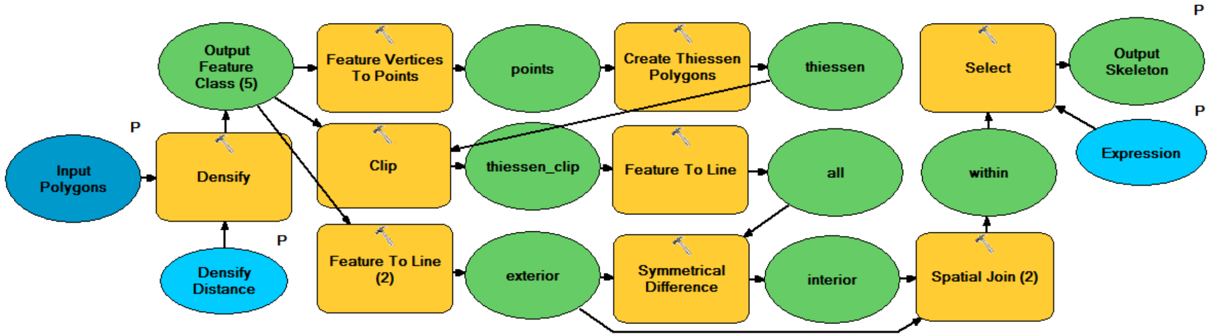


Figure H.2: Polygons to Lines (1) – create skeleton



Source: ESRI, 2011.

Figure H.3: Polygon to Lines (2)

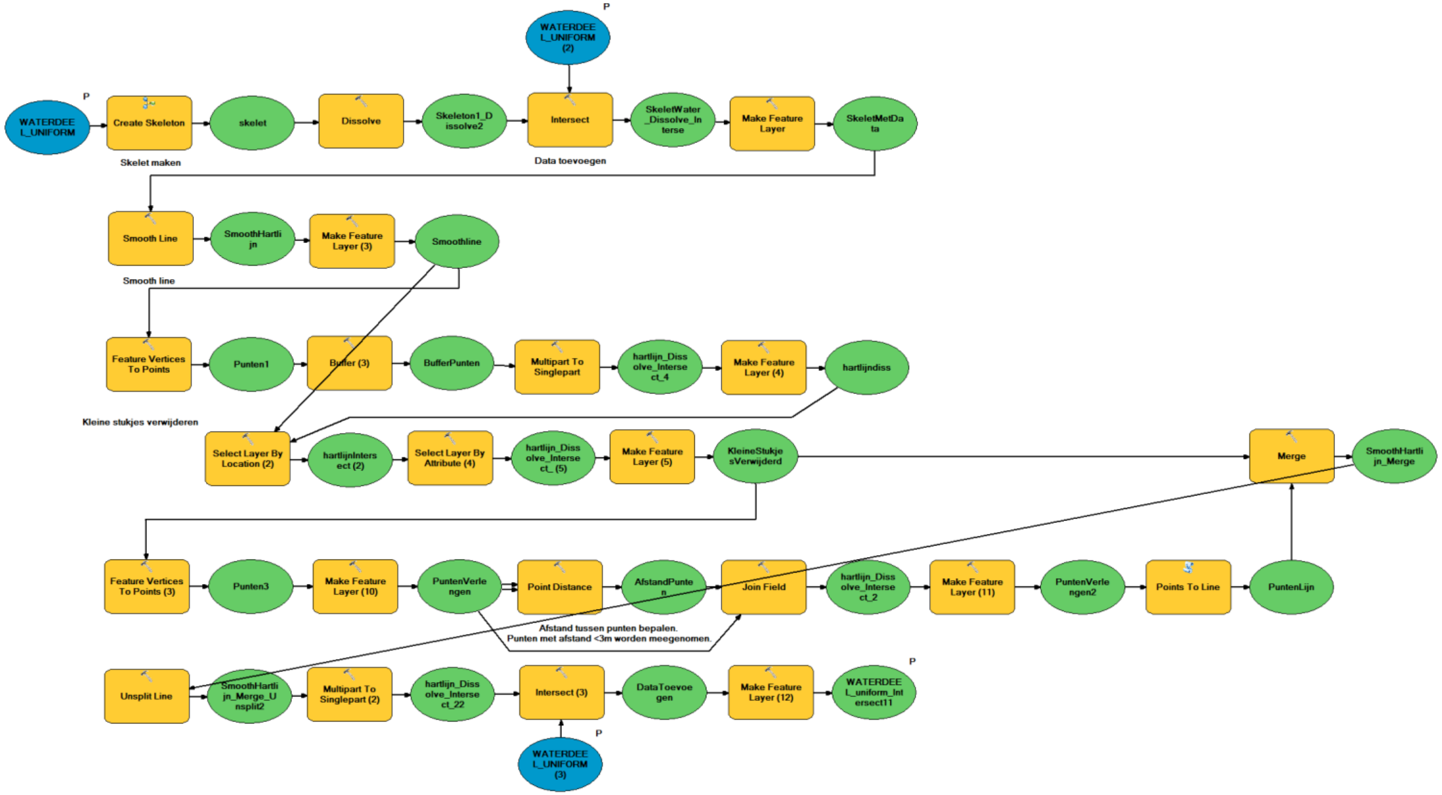


Figure H.4: Formal key registration requirements

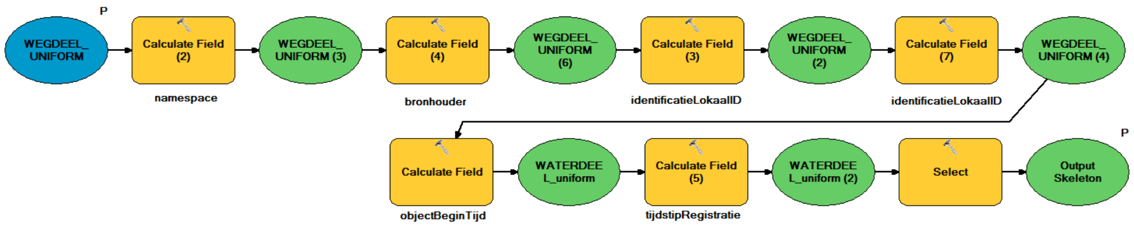


Figure H.5: Physical appearance

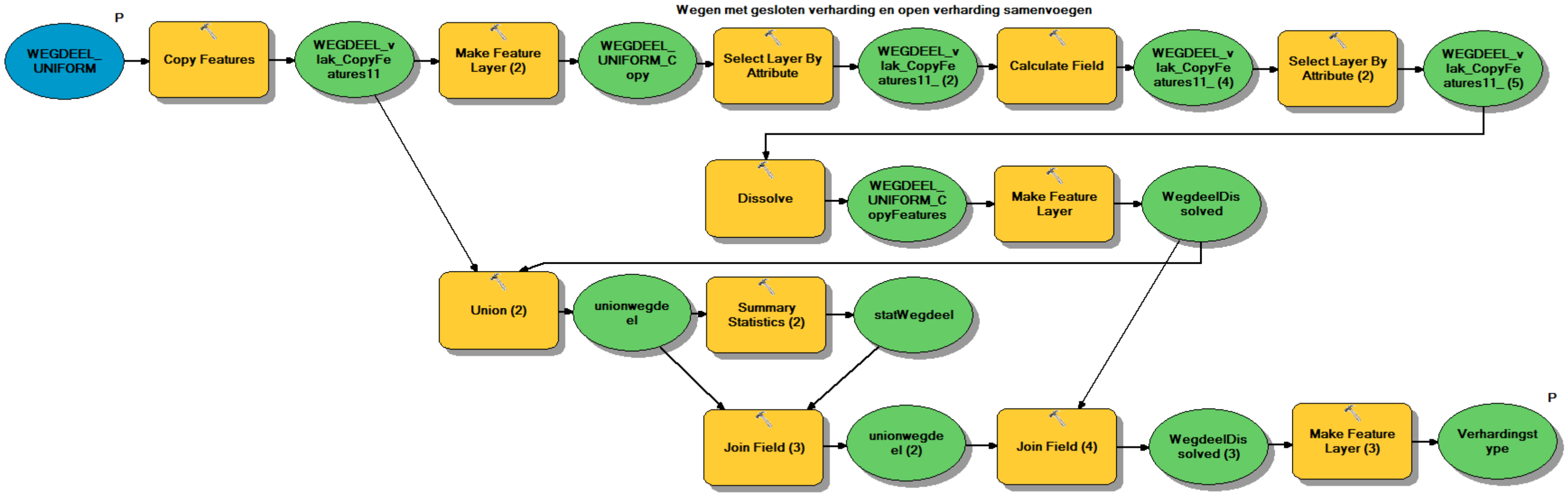


Figure H.6: Parking areas

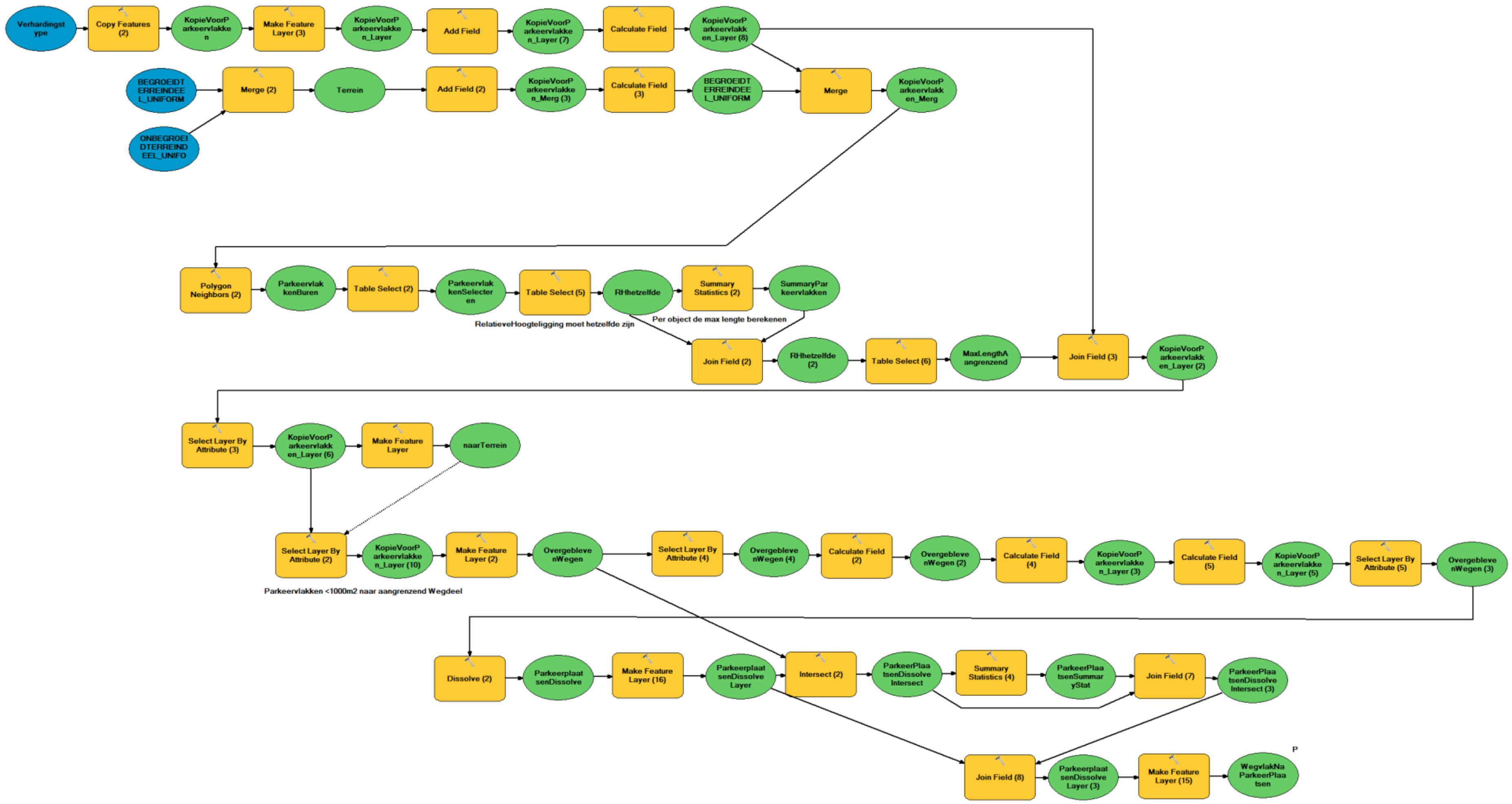


Figure H.7: Driveways

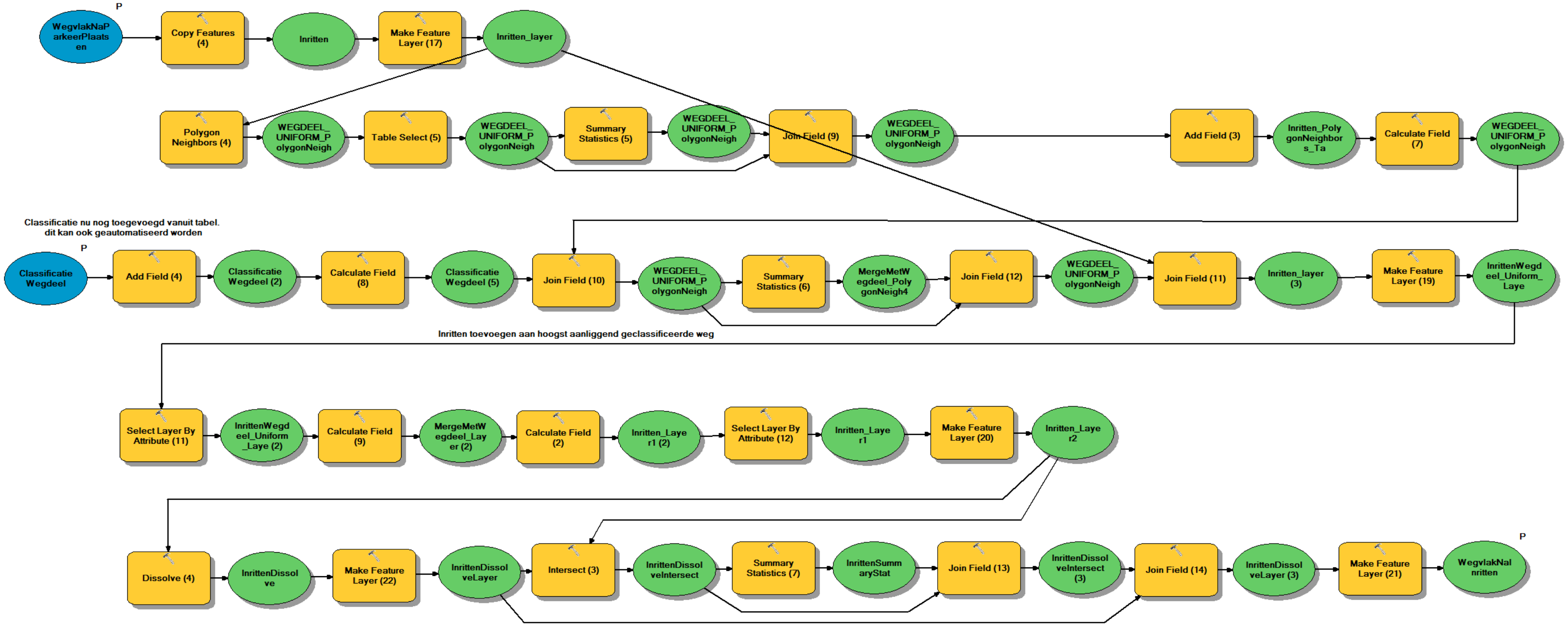


Figure H.8: Footpaths

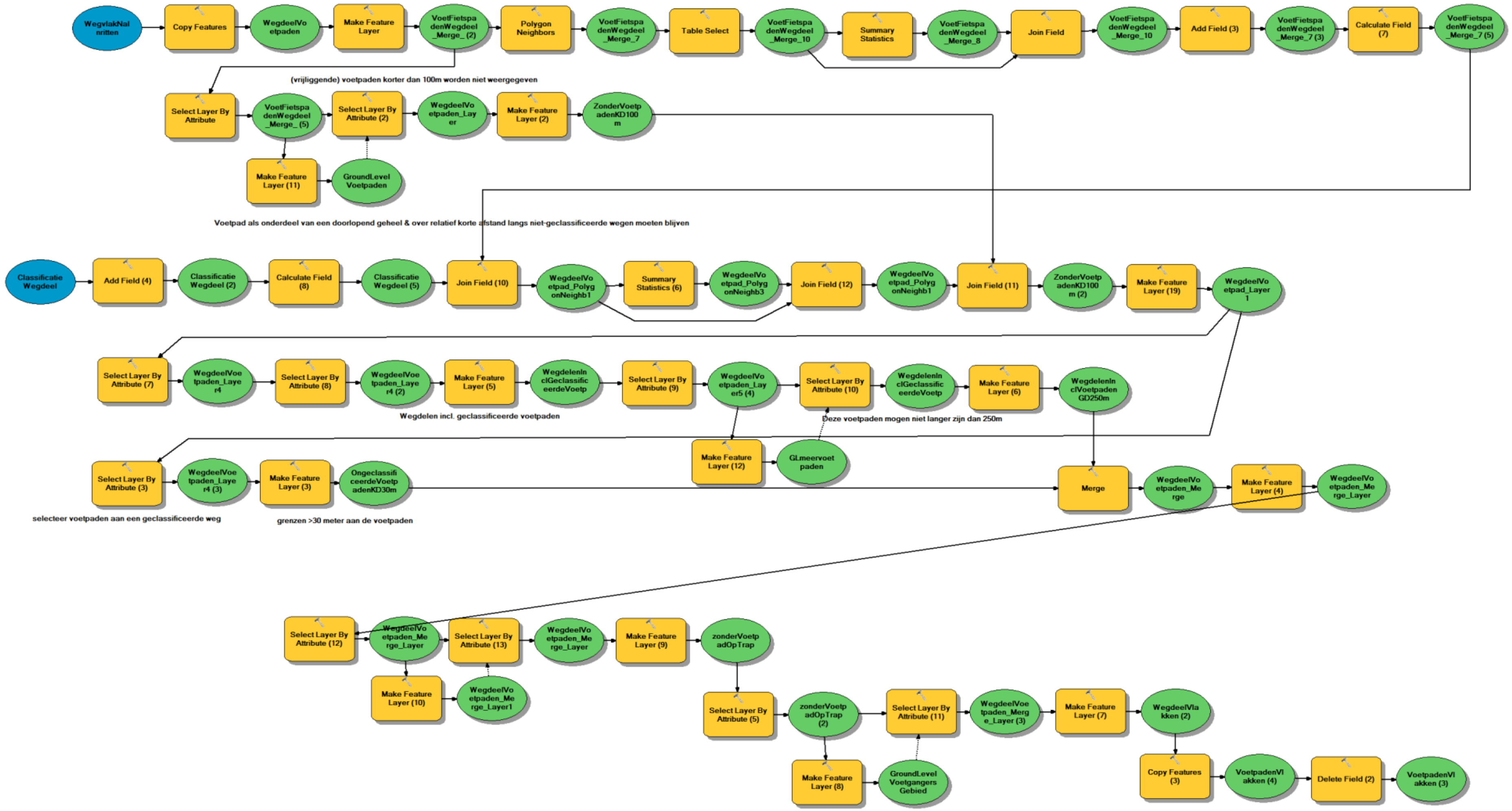


Figure H.9: Geometry

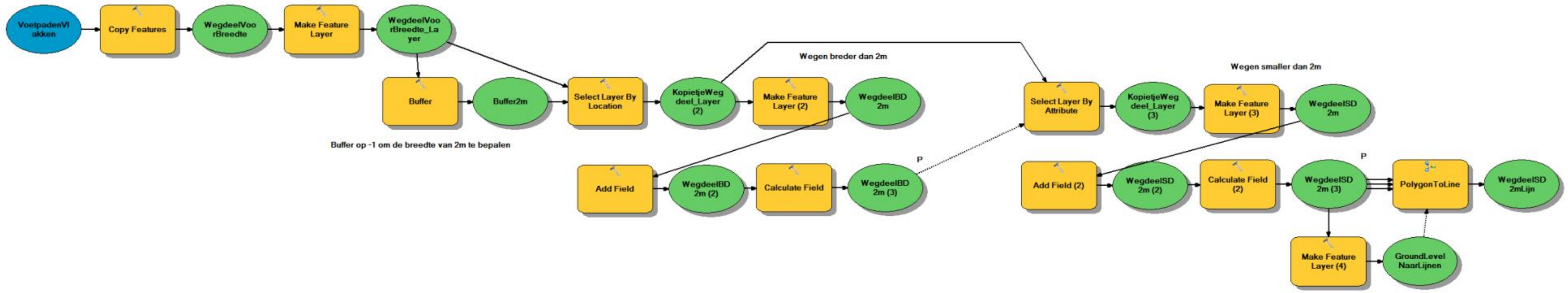


Figure H.10: SUPPORTIVEPARTOFROAD

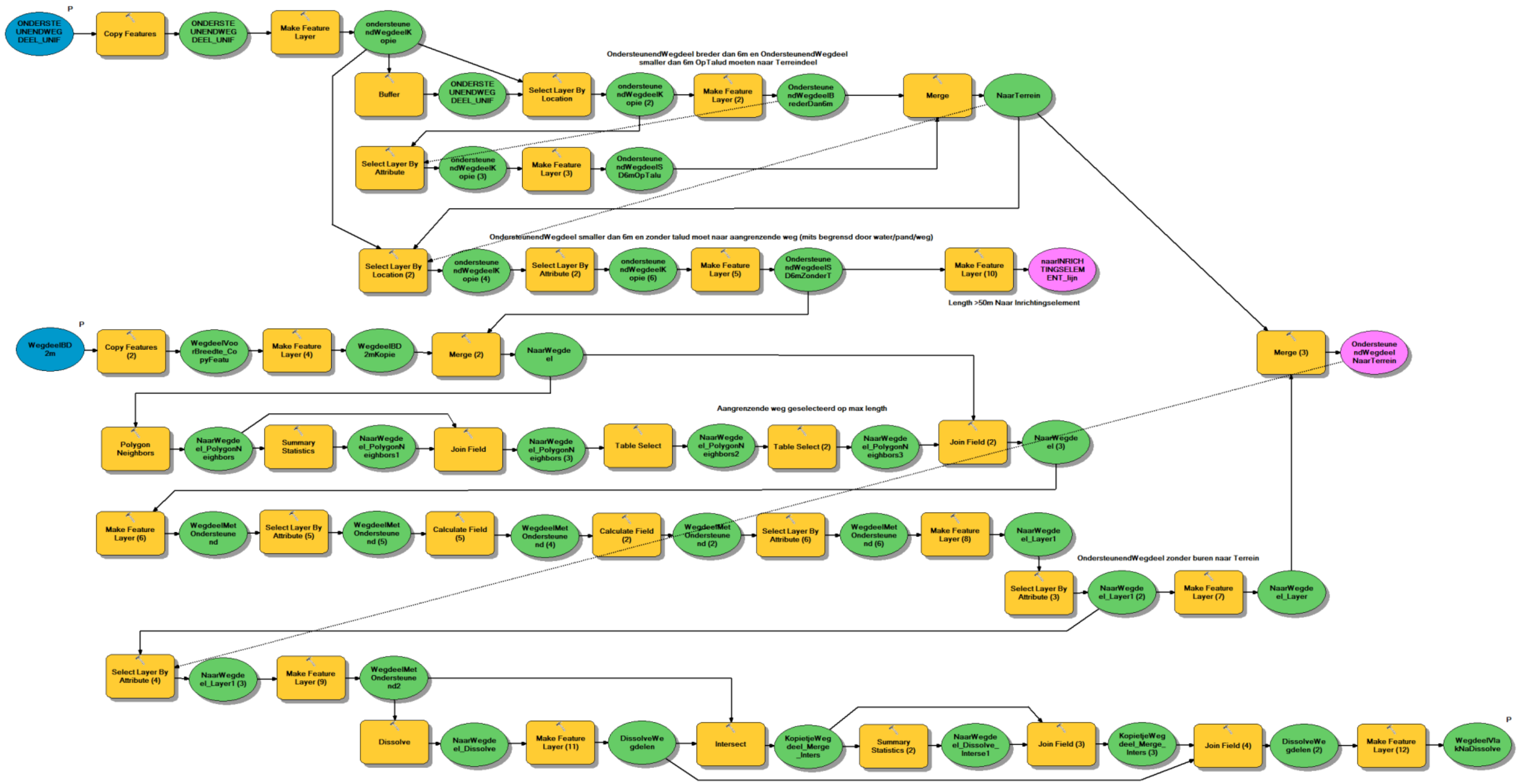


Figure H.11: Bicycle paths (1) – polygons

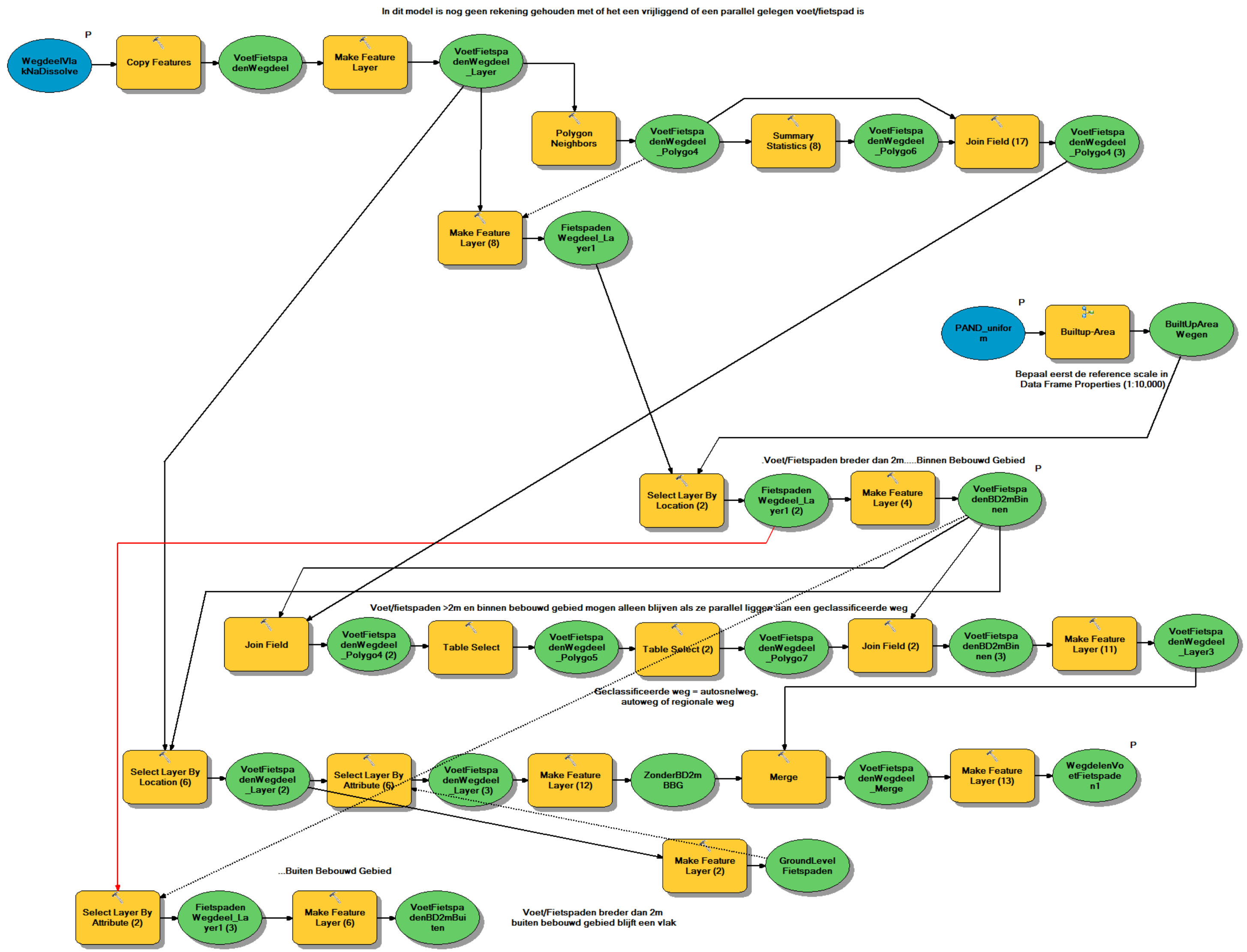


Figure H.12: Bicycle paths (2) - lines

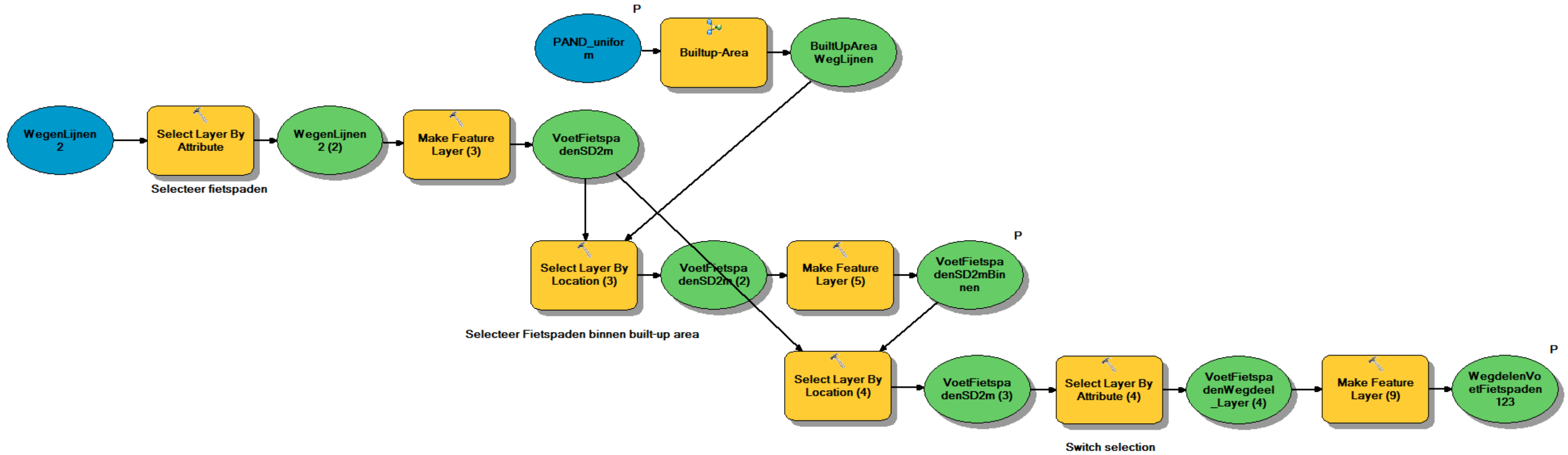


Figure H.13: Centerlines

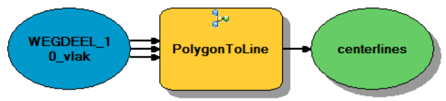


Figure H.14: Buildings

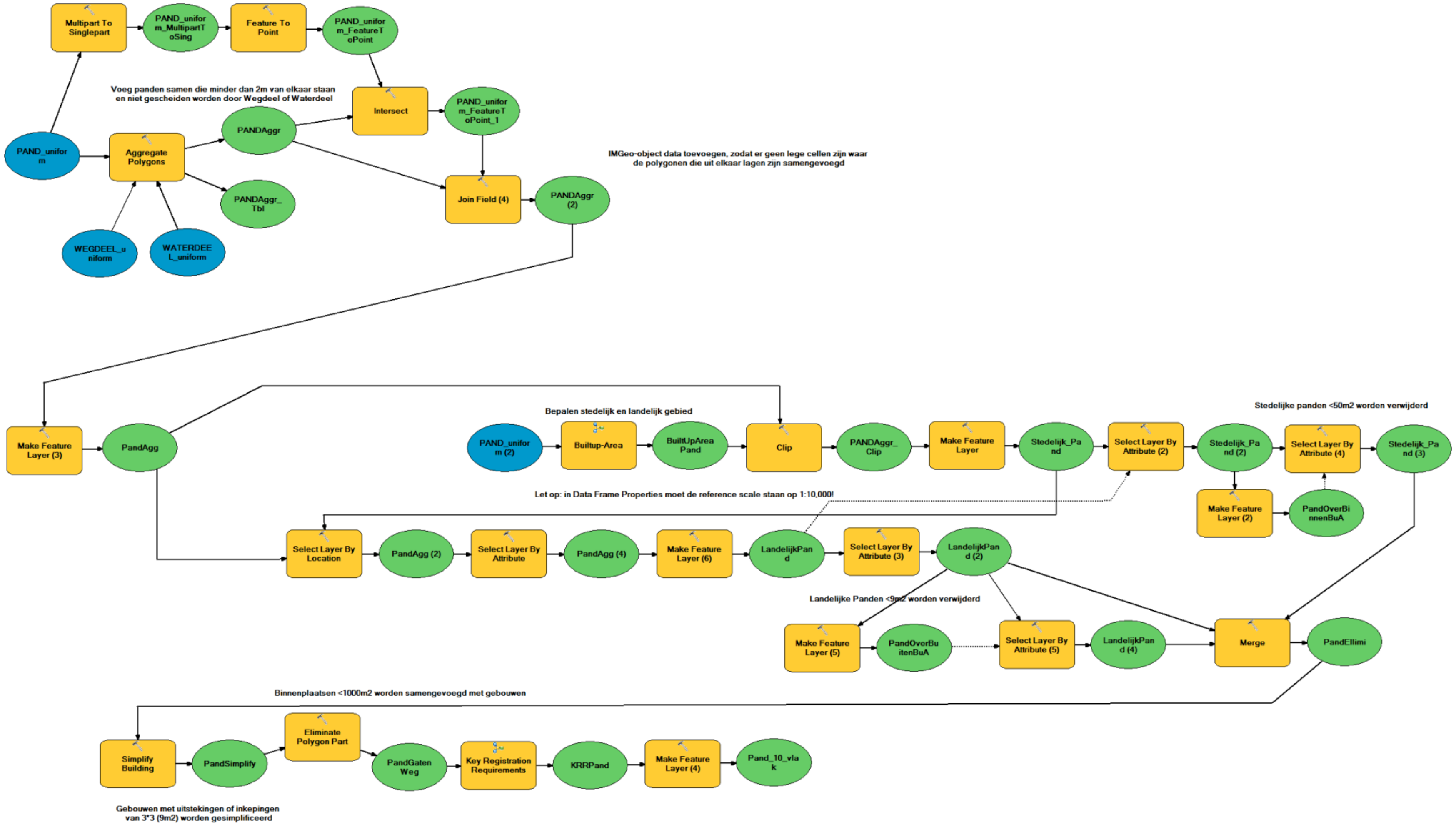


Figure H.15: Water

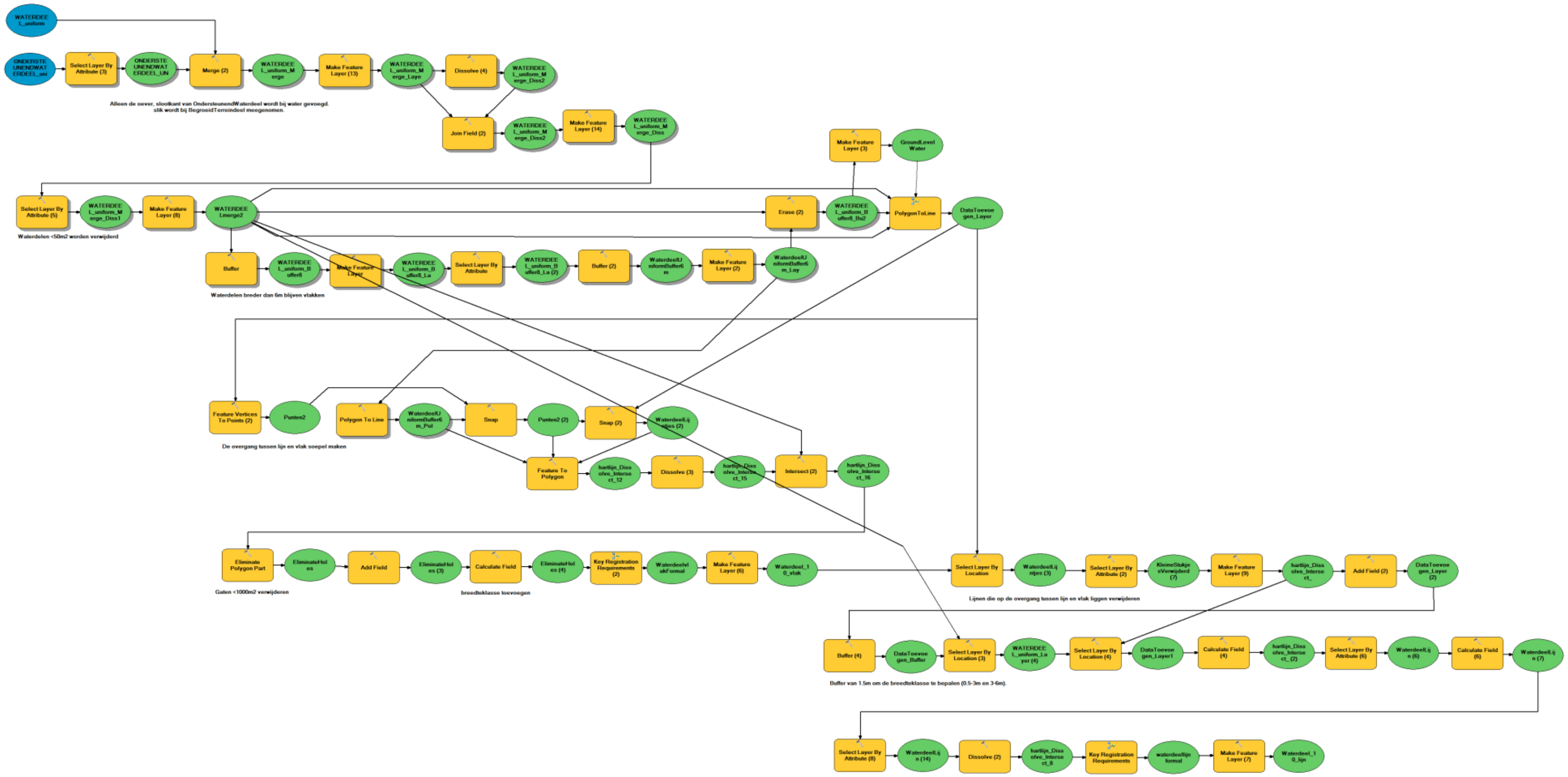


Figure H.16: Nature part 1

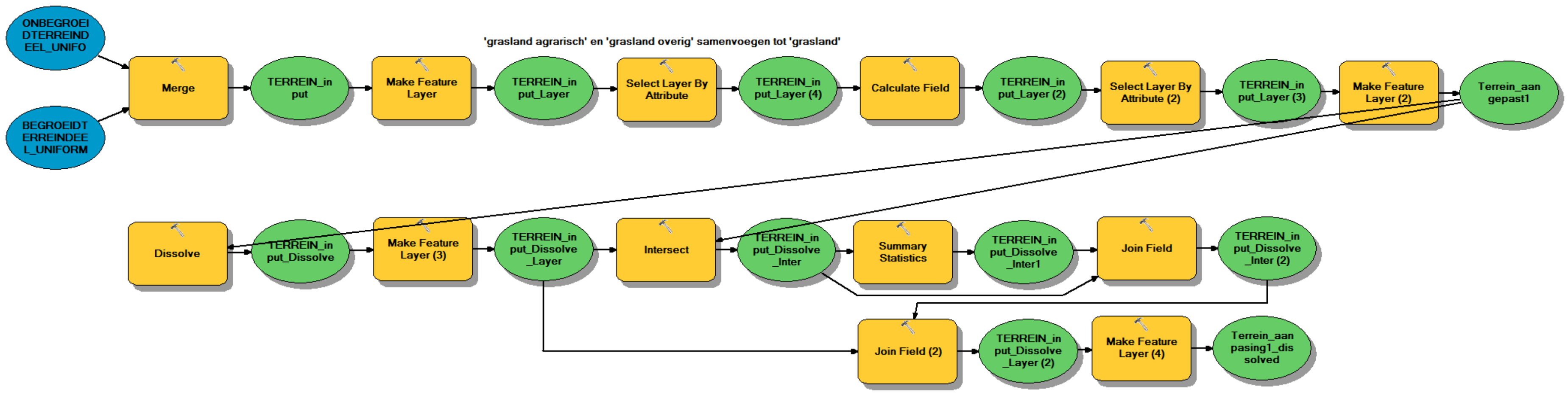


Figure H.17: Nature part 2 (executed after the determination of holes, see Figure H.22)

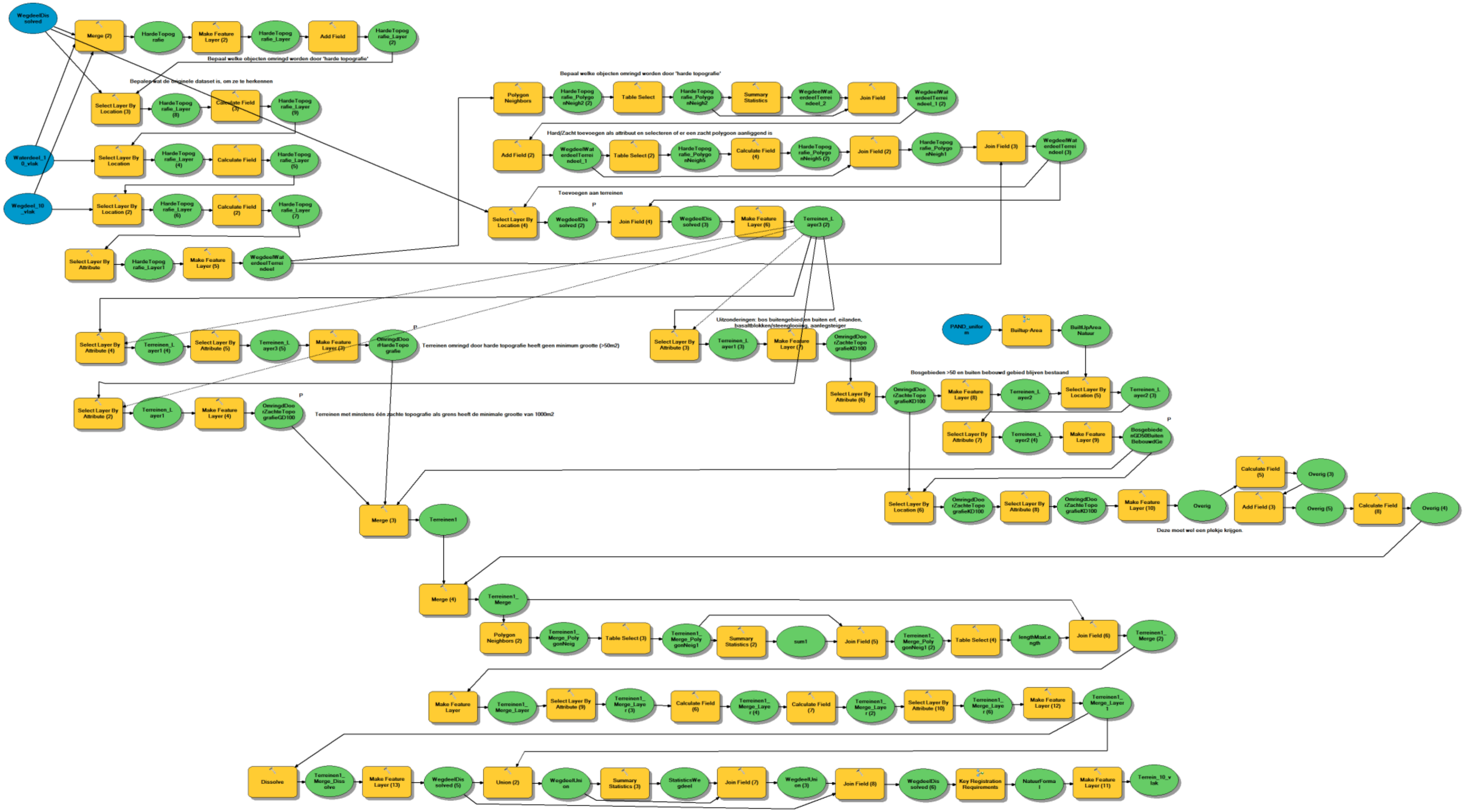


Figure H.18: Bridges

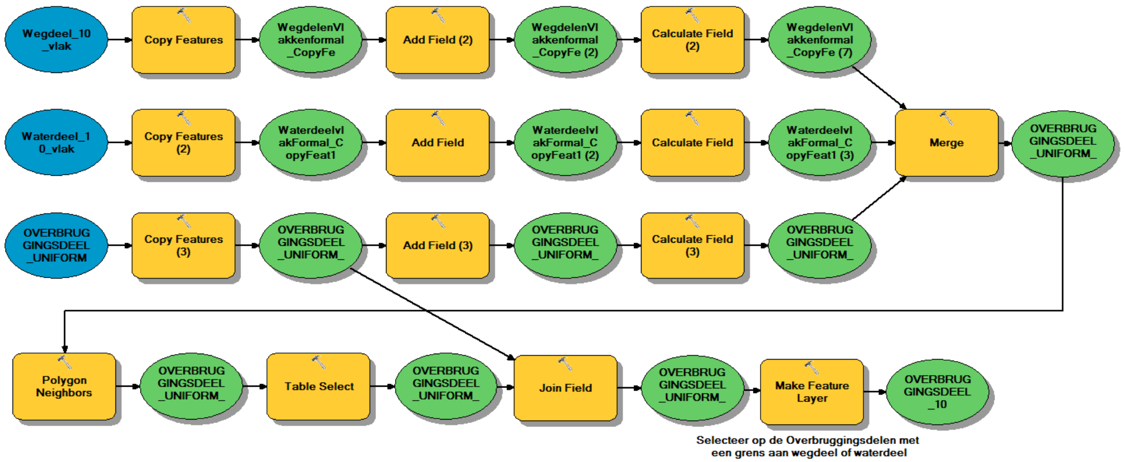


Figure H.19: Tunnels

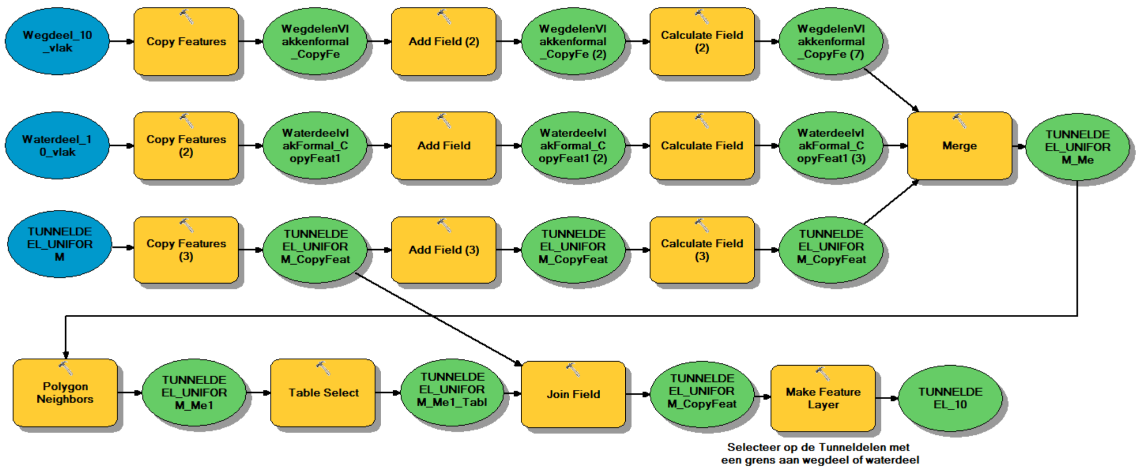


Figure H.20: LAYOUTELEMENT_point (ENGINEERINGSTRUCTURE & REMAININGSTRUCTURE)

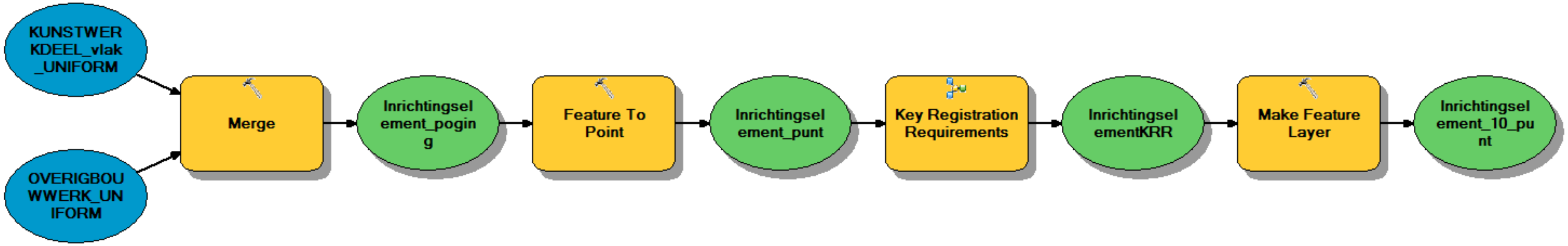


Figure H.21: LAYOUTELEMENT_line (SEPARATIONS)

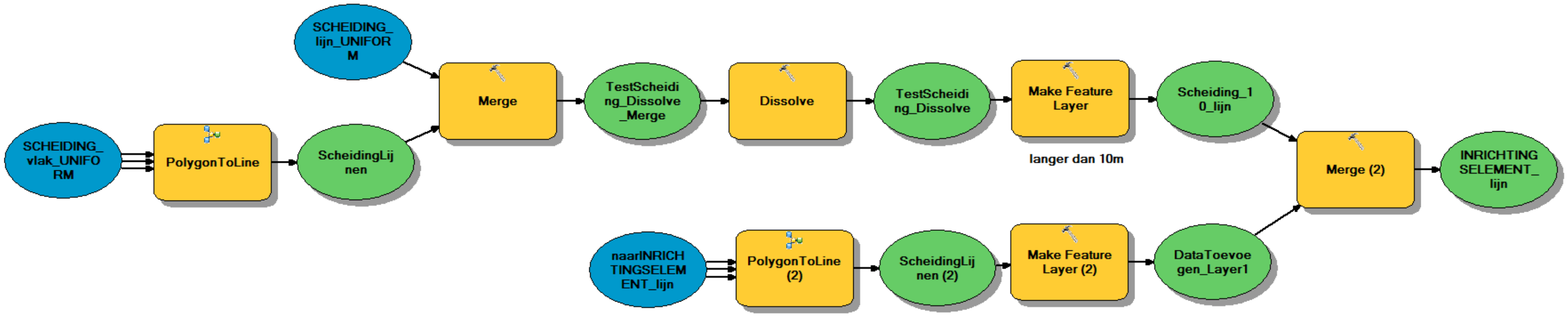


Figure H.22: Integrating eliminated objects

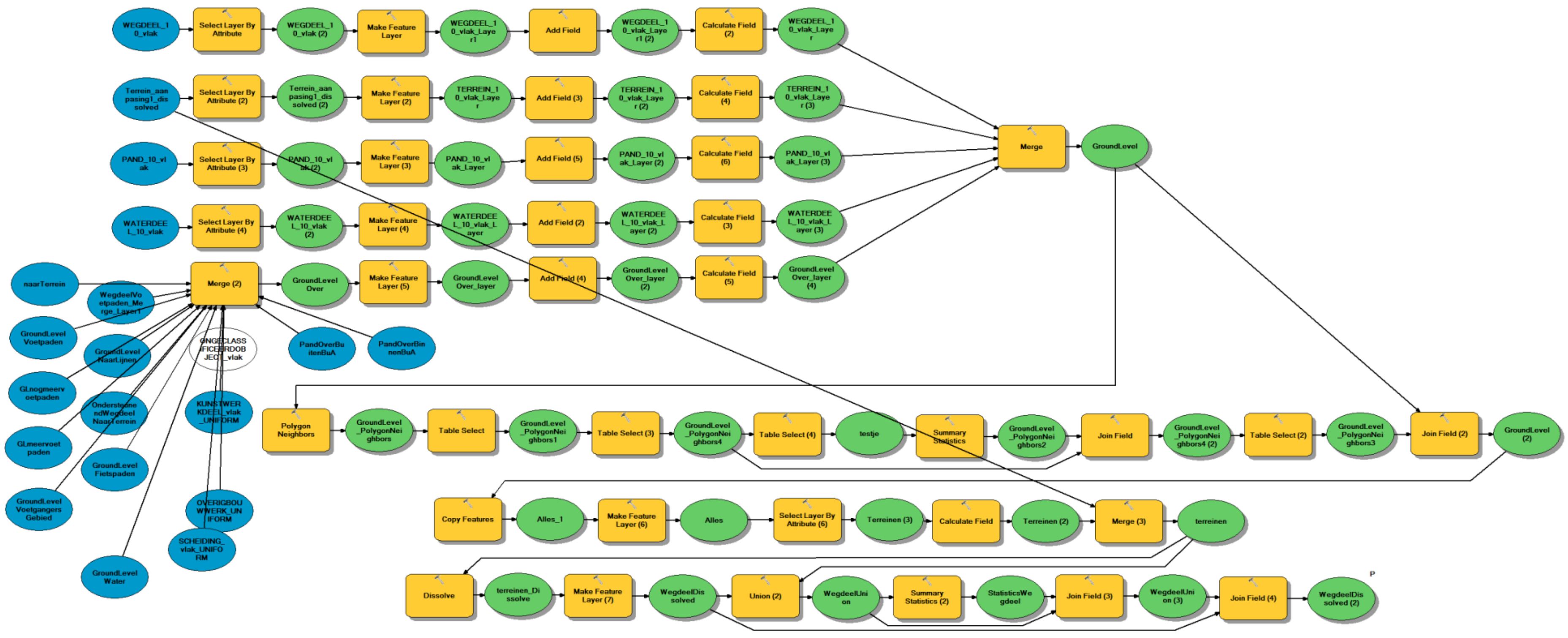
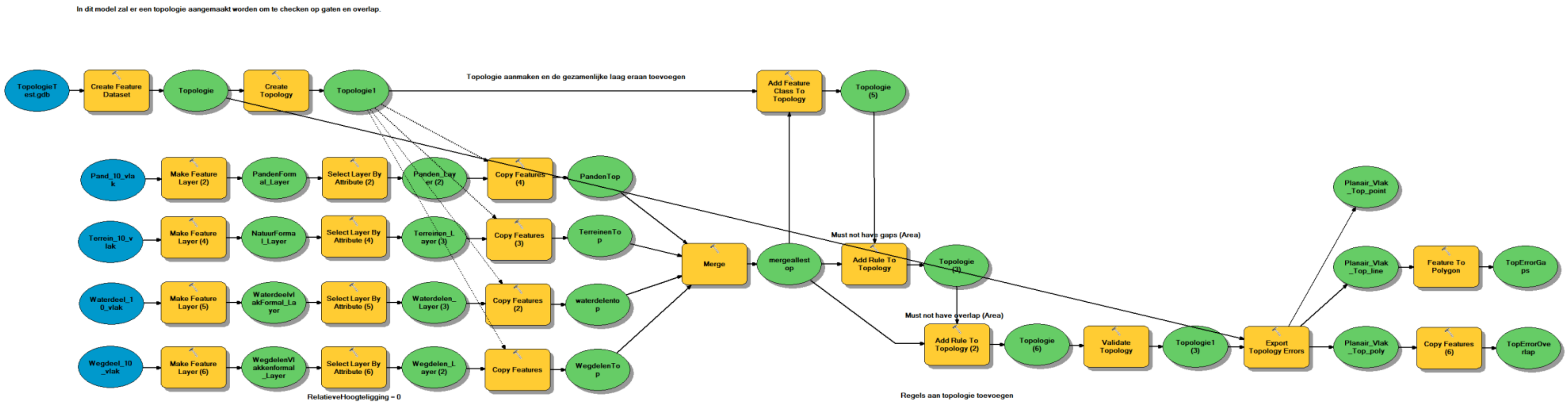
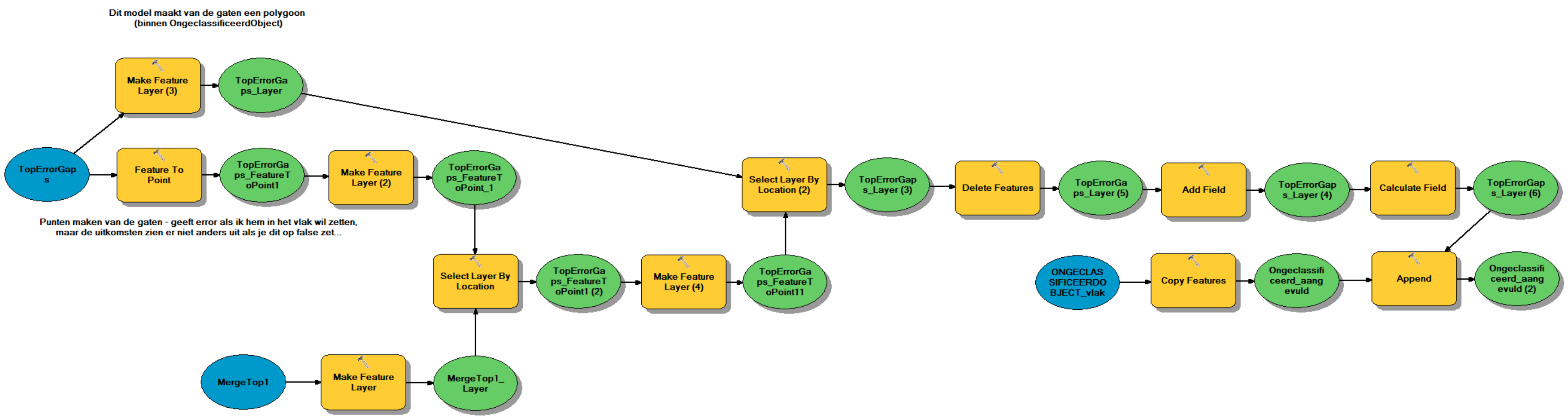


Figure H.23: Step 1



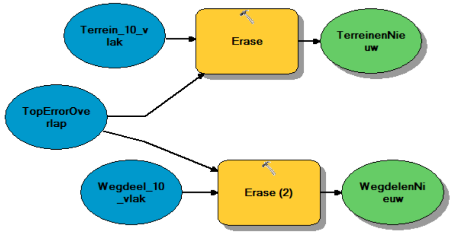
Source: based on Altena et al., 2013

Figure H.24: Step 2



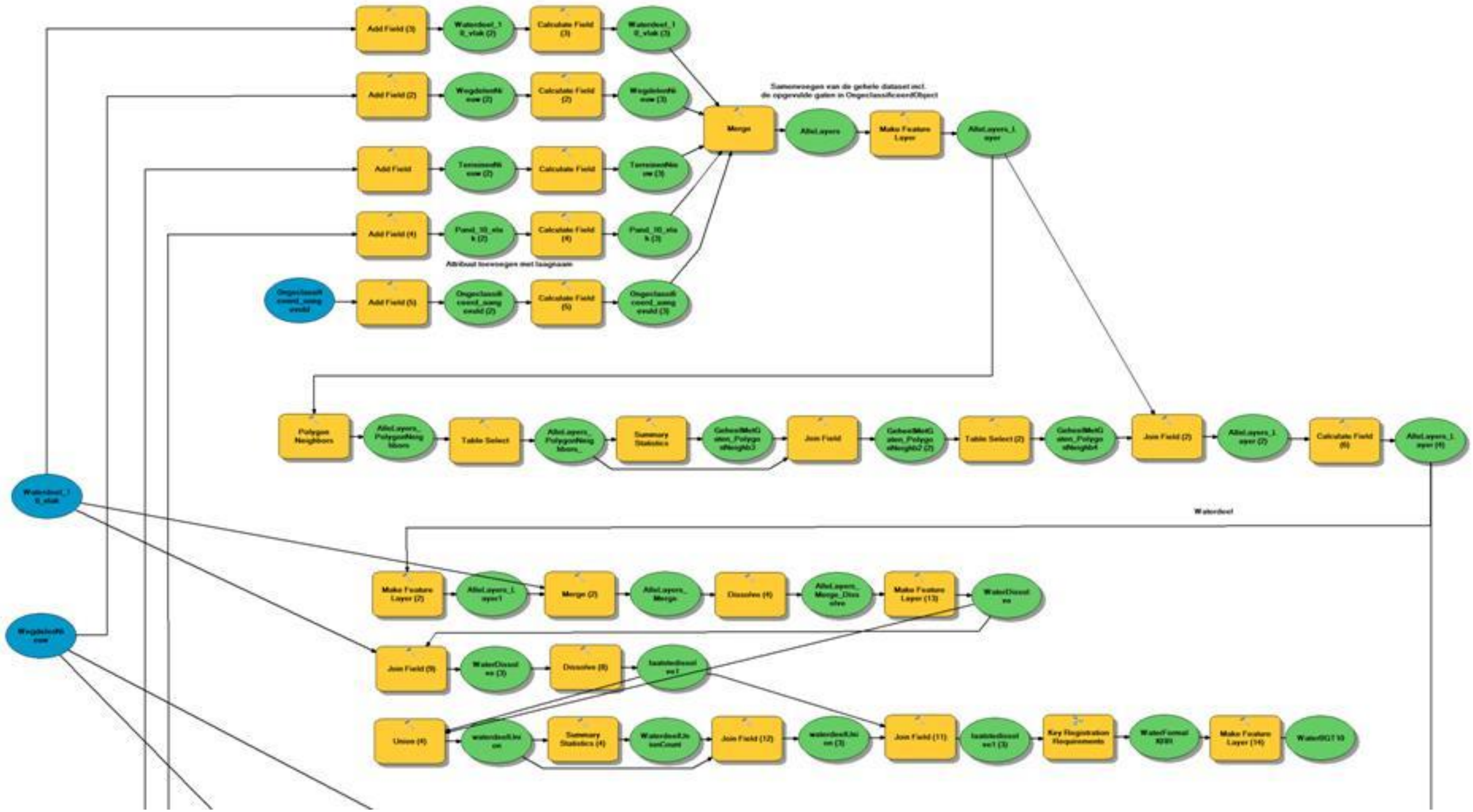
Source: based on Altena et al., 2013

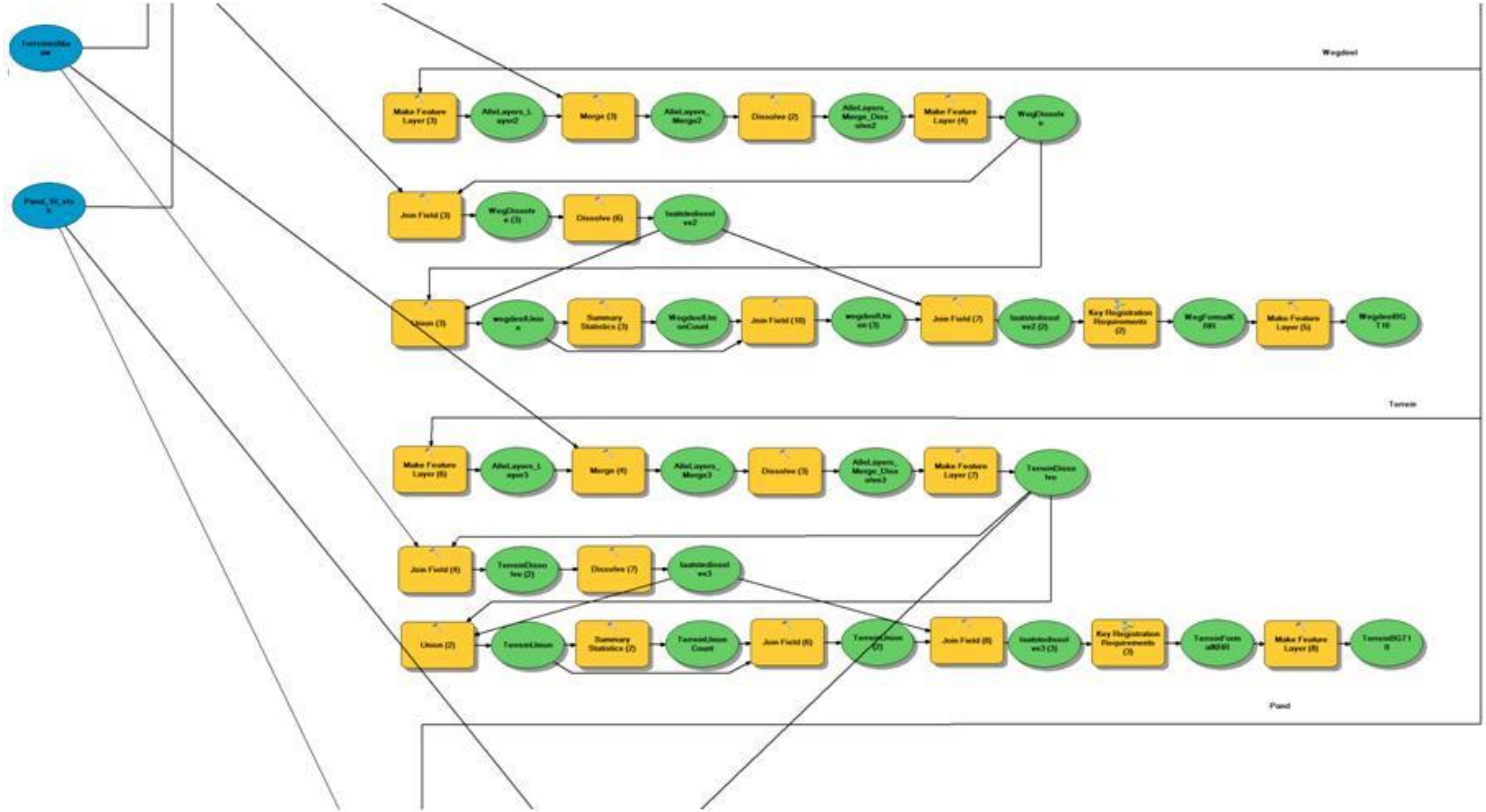
Figure H.25: Step 3



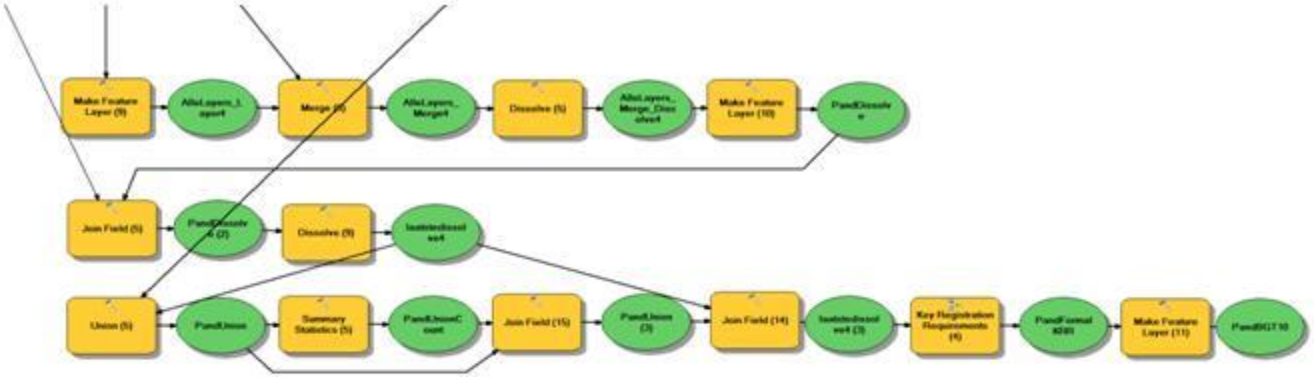
Source: based on Altena et al., 2013

Figure H.26: Step 4





Wegdeel
Make Feature Layer (3)
Merge (3)
Dissolve (2)
Make Feature Layer (4)
Join Field (3)
Dissolve (6)
Union (3)
Summary Statistics (3)
Join Field (10)
Join Field (7)
Key Registration Requirements (2)
Make Feature Layer (5)
Terrein
Make Feature Layer (6)
Merge (4)
Dissolve (3)
Make Feature Layer (7)
Join Field (4)
Dissolve (7)
Union (2)
Summary Statistics (2)
Join Field (6)
Join Field (8)
Key Registration Requirements (3)
Make Feature Layer (8)
Pand



Source: based on Altena et al., 2013