Automated Aggregation of Geographic Objects

A New Approach to the Conceptual Generalisation of Geographic Databases

J.W.N. van Smaalen

NCG Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission

Delft, November 2003

Automated Aggregation of Geographic Objects
A New Approach to the Conceptual Generalisation of Geographic Databases
J.W.N. van Smaalen
Publications on Geodesy 55
ISBN 90 6132 282 0
ISSN 0165 1706

Published by: NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission, Delft, The Netherlands
Printed by: Optima Grafische Communicatie, Optima Graphic Communication, Rotterdam, The Netherlands
Cover: Multi-layer model for spatial abstraction

NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission
P.O. Box 5058, 2600 GB Delft, The Netherlands
Tel.: +31 (0)15 278 28 19
Fax: +31 (0)15 278 17 75
E-mail: ncg@citg.tudelft.nl
Website: www.ncg.knaw.nl

The NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission is an institute of the Royal Netherlands Academy of Arts and Sciences (KNAW)

Preface

Automating the process of map generalisation has been a scientific challenge for over 30 years and still there is no comprehensive practical method. In this study I deal with the database part of the problem which has received relatively little attention in comparison with the total effort put into map generalisation research. This way I hope to contribute to a generic map generalisation system that should be possible before long, by combining the research results of the past few decades.

This study was carried out during a few distinct periods. I first got involved in the subject of map generalisation in 1992 at the Canada Centre for Remote Sensing in Ottawa, where I assisted Dianne Richardson in implementing the method she had developed for her PhD research. A period that I look back upon with great pleasure. Subsequently, in 1994, I started a four-year PhD research project at Wageningen University, which was a follow-up of Dianne's work. It was during this time that I developed the concept of aggregation based on co-occurrence of classes. After this period the project came to a temporary standstill. Although I never lost the intention to finish it, it was only last year that I picked it up again and finished it, resulting in this dissertation.

During these periods a number of people have been involved. I would like to thank Martien Molenaar for his patience when it might have seemed that I would not finish my study with a dissertation, for providing me with the conceptual framework for my study and for leaving me the freedom to find my own solutions. Arnold Bregt, who got involved in the project at a later stage, but still provided invaluable input and very practical comments when most needed. As during one meeting at ITC, when we solved the remaining issues in a very constructive 15-minute discussion.

I would further like to thank all colleagues at the Surveying department (*Landmeetkunde*) for their collegiality and countless *gezellige* lunches at Unitas on Tuesdays when there were *pannekoeken* on the menu. Special thanks go to John Stuiver for introducing me to GIS in the first place and making me enthusiastic about its possibilities. René van der Schans for inspiring and animated discussions during the early stages of my study. Ron van Lammeren for all his help and enthusiasm. Elisabeth Addink, of course, for her wit and our collegial discussions, occasionally on the subject but most of the time diverting into the most ridiculous directions, which also made the time at *Landmeetkunde* memorable.

Further thanks go to Kees Bol for salvaging files that I had accidentally deleted from the server and Prof. Kruidhof, founder of the *Landmeetkunde* laboratory, for providing us with such an exceptional place to work, overlooking the ever-changing sight of the river Rhine flood plains.

Lots of thanks go to my parents, who have always encouraged me to study. Well mum, dad, this is about all I can do. And finally my girlfriend Ingrid for her love and support, as well as patiently answering people's questions whether I had finished my PhD yet.

Contents

Chapter 1: Introduction

1.1 What is generalisation?

Geographic information is gathered, structured and stored with a certain purpose in mind. Optimising data for a specific use often means that they become less suitable for other purposes. One of the issues relevant for the intended use is the determination of the appropriate level of spatial and thematic detail. Using a very detailed dataset is often not desirable as this may make it difficult to get an overview. Generalising the data is then an option. One possibility is to 'thin' the original data, but more often the original data will have to be transformed to create new, composite objects.

In cartography, generalisation is generally carried out manually, focusing on the graphic representation at different scale levels. Most of the early research into the automation of procedures for map generalisation also focused mainly on cartographic, and therefore graphic aspects. Nowadays generalisation is part of the framework of geographic information processing. In this context we speak of conceptual or model generalisation, i.e. the process that creates a derived dataset with properties that are more desirable and usually less complex than the ones in the original dataset (João et al. 1993). Conceptual generalisation disregards the graphic limitations of the output medium and related operations like exaggeration, displacement and smoothing. Instead, it concentrates on operations such as classification, aggregation and elimination. Attempts have been made to create comprehensive systems for automated map generalisation. These attempts were often based on rules for manual generalisation, but these proved unsuitable for automated environments. Systems based on new rules specific to automated generalisation were more successful but their use was still limited to well-defined situations, regarding either the input data or the purpose of the generalised data, or both. Multi-purpose systems do not exist.

Generalisation holds an important position in the development of a theoretical framework for handling geographic information, as it deals with the structure and transformation of complex spatial notions at different levels of abstraction. A better understanding of these processes will also improve our understanding of data acquisition, analysis and visualisation.

Generalising a geographic dataset is like creating an abstract of a text: main and secondary issues are differentiated. A text is written for a certain purpose (to entertain, to inform, to influence etc.) and for a certain audience (children, students, employees etc.) An abstract is either meant for readers to determine if the text is worth reading, by giving an impression of the contents, or to be informed to a basic level. The target group an abstract is aimed at is often larger than the group that would read the text itself. The same applies to the generalisation of geo-data. A generalised presentation is a tool to aid in understanding and gaining access to the underlying, more detailed data. It all has to do with the context in which the information is used: *what* is communicated to *whom* and *why*?

1.2 Why generalise?

Maps of different scale levels have coexisted for centuries, so apparently there is a need for information at different levels of detail. There is a need to present data at a less detailed level in order to extract structural patterns from the dataset, patterns that are otherwise overlooked because of the sheer amount of data. Or simply because of a limited amount of space on the output medium.

Since the introduction of geographic information systems there has been another reason for generalising data, which is for data integration purposes. Generalisation plays an important role in the integration of spatial information from various sources. All of these reasons have one thing in common: they are context transformations (see section 3.3). The initial data are collected for a certain purpose and then adapted for another type of use by means of generalisation.

1.3 Problem definition

The current research into automated map generalisation appears to be in a cul-de-sac. It seems impossible to disconnect the issue of conceptual generalisation from the traditional environment of map generalisation. The requirement of an aesthetically pleasing map as the end result of the generalisation process often distracts from the actual issues involved. By dropping this constraint it should be possible to simplify matters significantly and get to the core of the problem more easily. The resulting database after generalisation is not necessarily suitable for visualisation in the form of a map. The database may contain objects that are relatively small or narrow so that they cannot be readily visualised. The visualisation of these objects becomes a secondary issue that is dealt with separately.

Thus far, most generalisation research worked towards a known end result, usually resembling a traditional paper map.

For real insight into the generalisation process we will concentrate on large generalisation steps. Most current research concentrates on small generalisation steps. The objects after generalisation are commonly the same objects as the ones in the initial dataset, except that some are omitted and the remaining simplified. With larger generalisation steps it is not possible to just select and eliminate. Instead, objects have to be combined to create new objects. Not different descriptions of the same object, but entirely new object classes.

In the past several generalisation operations have been developed for individual objects and dichotomous maps but the number of procedures for categorical maps is still limited (Peter and Weibel 1999, Galanda 2001) and the methods that do exist rely on similarity and importance factors that are hard to determine (van Putten and van Oosterom 1999, Bregt and Bulens 1996).

1.4 Objectives and limitations

The goal of this study is to develop a framework and a working prototype for the generalisation of object- and vector-based categorical maps - such as large-scale topographic data - based on inter-object relationships. We strive for a system that is to a large extent automated and can be operated by non-expert users.

Large-scale topographic maps are commonly object-based, categorical maps. The objects are classified: 'road', 'building' etc. The spatial inter-object relationships in large-scale datasets are often complicated. We will concentrate on the aggregation of objects belonging to distinct object classes, based on the spatial and thematic relationships between the objects in these classes. Not based on similarity but rather on relationships that indicate functional units at higher levels of abstraction.

This study is based on concepts from database systems research. These appear to be a sounder basis for a conceptual generalisation system as outlined than most current geo-generalisation research which is still strongly influenced by cartographic considerations. We will concentrate on conceptual generalisation, i.e. on operations such as aggregation and elimination rather than cartographic operations such as displacement, amalgamation and exaggeration. The method is therefore not based on manually generalised maps and existing rules for manual generalisation. We will concentrate on rules that are not bound to specific classes since their use is limited. The implementation uses regular database software, to prevent the method from being too dependent on any of the current GIS software packages. There is still much development going on in this area and the trend is towards a more open environment, in which the integration of spatial and non-spatial data will become easier and more and more common.

Two large-scale topographic datasets, the TOP10vector dataset of the Topographic Survey of the Netherlands and the Grootschalige Basiskaart Nederland (Large-scale base map of the Netherlands) are used as case study material. The cases are used to illustrate generalisation using the developed method.

1.5 Research questions

1. What are the consequences if we concentrate completely on conceptual, that is non-cartographical aspects of the generalisation process? What are the relevant operations in that case and how do we assess the result?
2. How are the objects in a categorical map interrelated thematically and spatially and how can we use these relationships for the definition of generalisation rules?
3. What parameters can be defined for the user to control the outcome of the generalisation process?
4. How can we minimise errors, such as shifts in thematic values and topological inconsistencies?

1.6 Relevance

The relevance of this research is motivated by the fact that generalised datasets are currently produced manually from detailed ones by skilled personnel at great expense, or new data are collected in the field. Mapping agencies produce datasets at different levels of detail. The production of large-scale datasets, in particular, is labour-intensive. At present, they are created relatively independently from one another, re-interpreting the field situation from aerial photographs, for instance. In other words: by means of an at least partly redundant data acquisition process. Or datasets are generalised manually from detailed ones. This requires skilled personnel and is therefore also an expensive method. Automated procedures are used in this manual process, but only to perform specific, detailed tasks like simplifying buildings and smoothing linear objects.

Some agencies are starting up combined efforts to share information. The Dutch Cadastre and the Topographic Survey of the Netherlands, for example, are looking into the possibilities of using the Grootschalige Basiskaart Nederland (GBKN) as a basis for the

TOP10vector produced and marketed by the Topographic Survey. This, too, calls for generalisation procedures.

The use of these expensive, large-scale datasets is currently quite limited, and sophisticated generalisation methods could expand this use significantly by enabling conversion of the information to suit specific applications. This study hopes to establish the requirements with regard to the structure and content of the input data in order to facilitate generalisation.

Moreover, generalisation performed by an average user is currently not possible, as there are no systems for real-time conceptual generalisation that require no expert knowledge. Generalisation for exploratory purposes is therefore not possible.

Chapter 2: Related literature

2.1 The early years: individual objects

Research into the possibilities of computer-assisted map generalisation started in the late 1960's with algorithms for line generalisation. This early work focused on the generalisation of individual objects (Robinson 1995). The goal was to simplify the spatial delineation of particular objects in order to represent them on a map at a smaller scale. Research into line generalisation issues continues up until this day, using increasingly advanced techniques (e.g. Herbert et al. 1992, Müller and Wang 1993, Werschlein and Weibel 1994, Plazanet et al. 1998). The line-generalisation algorithms were later joined by more specialised operations like the simplification of buildings (Lee 1999). For the most part, these algorithmic generalisation operations act locally.

2.2 Dichotomous or presence/absence maps

Work in the late 1970's and early 1980's concentrated on issues like object selection. It incorporated inter-object relationships, but only within single classes (Brassel and Weibel 1988). Different classes were generalised independently. This could explain, or be explained from, the focus on small-scale maps. Unlike large-scale maps, which contain mainly areal objects, small-scale maps contain mainly linear and point objects. The spatial interrelationships between these objects tend to be simpler, which allows them to be treated independently during generalisation (Robinson 1995).

The spatial data models used were still intended for drawing maps, not for modelling geographic information. These models failed to support more complex processes (Grelot 1986). It was acknowledged that a more comprehensive approach was needed to handle several objects simultaneously. This required data models that support spatial proximity, structure recognition and composite objects (Grelot 1986). Various data models emerged, primarily to enable operations such as the displacement and amalgamation of unconnected objects. These models were predominantly based on Delauney triangulation (Peng 1995, Bundy et al. 1995, Jones et al. 1995, Jones and Ware 1998, Liu 2002). More recently, methods for object displacement using the snakes concept (Burghardt and Meier 1997, Barrault et al. 2000) and finite elements analysis (Højholt 1998, Bader and Barrault 2001) were introduced. Besides these purely cartographic operations, the data models based on Delauney triangulation also enable operations such as object collapse or skeletonisation (e.g. Chithambaram et al. 1991). These are used on network structures such as road networks and hydrographical features to create topological linear graphs, either for presentation or analysis purposes. The creation of topological graphs by skeletonisation is often a pre-stage of the structural generalisation (Molenaar 1998) of network data. Skeletonising operations have since become part of regular GIS software (Lee 1999, Lee 2001).

Directed networks like hydrographical networks can be generalised based on various existing classification methods for stream elements like the Strahler and Horton classifications. Elements of the lowest stream order are eliminated first (Richardson 1993). Martinez Casasnovas generalised a dataset of a gully erosion pattern and matching catchment areas. This approach is scale-related; upstream leaves of the gully network that are too narrow to be represented at the target scale are eliminated and their catchment areas merged with their downstream neighbours (Martinez Casasnovas 1994). The catchment areas are assigned erosion severity measures, resulting in choropleth maps with different levels of spatial abstraction before and after generalisation. Both use a network of connected and directed arcs and nodes to describe the hydrological systems and use the topological structure to determine the significance values of the elements.

Undirected networks like roads are generally more difficult to classify automatically. They can be generalised using the shortest-path spanning trees method (Richardson and Thomson 1996). Within the road network, points of interest are identified. In small-scale data these could be towns, for example. These points are connected using a shortest-path algorithm. More frequently used segments remain existent at higher levels of generalisation than less used ones. More recent studies lean more on the directional continuity of the network's elements (Thomson and Richardson 1999). Peng described an elimination method for urban road networks (Peng 1997). In this approach, elements are eliminated taking into account the size of the enclosed areas as well as the topology of the network.

A common characteristic of the procedures for network generalisation as described above is that they are all essentially *database enrichment* (Mackaness et al. 1997) operations. Elements are classified according to their importance within the network. The classification is used to describe the same process or model at various levels of spatial abstraction, leaving out the least important elements with every step. Certain process variables remain invariable throughout the various levels of abstraction. In a road network, for example, the traffic intensity at intersections remains invariant, irrespective of the level of detail of the contributing network. In the hydrological network the discharge at the outlet of each segment stays invariant after generalisation of the upstream network. This happens as a consequence of merging the watersheds belonging to the eliminated network elements with their downstream neighbours (Martinez Casasnovas 1994). Generalisation based on the topological structure of the network is an example of structural generalisation (Molenaar 1998).

Selection and elimination are important operations in generalising dichotomous maps. They are generally based on quantitative attribute values like size - a spatial attribute -, but sometimes on ordinal attributes. Selection can also be based on a combination of attribute values. Populated places, for example, can be eliminated according to their status (city, town, village etc.) and population. These two attributes can be combined in several ways to create different selection methods (Richardson 1993).

2.3 Aggregation methods for categorical maps

Categorical maps are based on area partitions[1]. As a consequence, objects cannot simply be eliminated; the area of the eliminated object has to be reoccupied. There are several ways to achieve this.

Geometry-driven generalisation refers to a collection of methods where the local geometry is the driving factor for the generalisation process (Molenaar 1998). Merging small areas with one of their neighbours on the basis of the largest common boundary is a common technique in this category (ESRI 1995). Small areas can also be joined with the largest adjacent area (Bregt and Bulens 1996). Both methods result in contaminated

[1] In an area partition, each point in the 2D domain belongs to exactly one of the areas (polygons): that is, there are no overlaps or gaps.

objects. In an attempt to minimise these incorrect classifications, combinations with the smallest generalisation error can be joined first (Bregt and Bulens 1996). This approach utilises attributes and geometry. The user must supply similarity measures for all object classes.

Generalisation errors can be avoided if classification hierarchies are used. Adjacent areas are merged if they belong to the same superclass. This method was employed on land cover data by Richardson (Richardson 1993). By assigning an - application-specific - importance factor to the classes, some classes can be excluded from the aggregation process (Bregt and Bulens 1996). A disadvantage of this method is that these importance factors are hard to assess. Class-driven generalisation (Molenaar 1998) is based on attribute values, geometry playing a secondary role.

Class-driven generalisation is based on similarity, unlike functional generalisation (Molenaar 1998). Looking at a city, a number of different components can be distinguished: houses, roads, parks etc. These components are not related in the sense that they are similar, but when put together a new object - the city - is formed. They are related in a functional sense: the city cannot function properly without any one of these components. The components that make up the city are also connected in a spatial manner. Only components that are mutually connected can create a city. Map generalisation applications based on these functional relationships do not exist currently.

A common feature of the more sophisticated aggregation approaches is that they are all based on either an existing classification hierarchy or a similarity matrix, containing similarity measures for all possible combinations of object classes in the dataset. Classification hierarchies are seldom readily available and similarity matrices are very difficult to establish. The aggregation methods described employ a topological data model, based on a topological graph (Langran 1991), to enable the determination of neighbours.

2.4 Cartographic vs model generalisation

Speaking about the generalisation of geographic information we have to realise that its origins lay in map generalisation. With the advent of geographic information systems an important change occurred. The notion of ‘scale’, which in a map implicitly acts as a measure for the level of generalisation, is no longer a property of the data. The term scale should therefore be abandoned and replaced by precision, accuracy and resolution (Müller et al. 1995). Scale only enters the picture when the data are visualised in the form of a map, using a restricted display area.

The distinction between cartographic and model generalisation (Figure 1) began to appear in literature in the mid-1980’s (Grünreich 1985, Brassel and Weibel 1988). It was emphasised that the conceptual generalisation and the problems related to the graphical limitations of the output medium should be handled separately (Kilpelaïnen 1992). Real-world features should be modelled instead of their cartographic representations (Mark 1991). Kilpelaïnen even made a distinction between the *actual* generalisation problems and the problems related to the graphic display. Nyerges identifies only four conceptual operations: classification, class generalisation, association and aggregation (Nyerges 1991). Displacement (Müller et al. 1995), exaggeration, simplification and smoothing, on the other hand, are operations that are typically restricted to the cartographic domain.

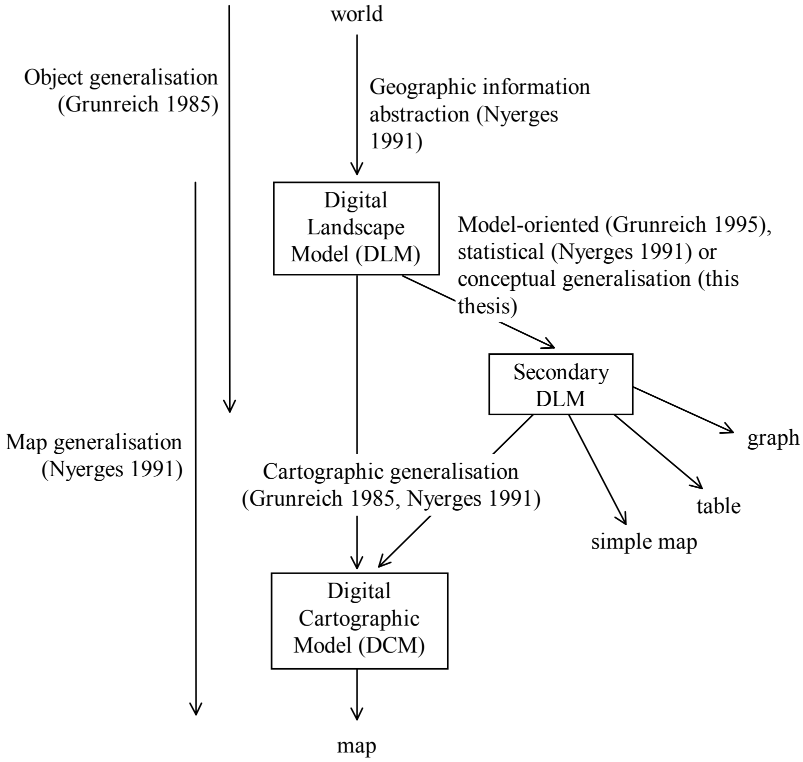


FIGURE 1. DISTINCTION BETWEEN CONCEPTUAL AND CARTOGRAPHIC GENERALISATION

In the early 1990's generalisation research evolved from seeking specific solutions to more comprehensive approaches and the definition of strategic models. The approach was top-down; instead of just working towards a desired and known result (how), the approaches encompassed the determination of why and when to generalise (Brassel and Weibel 1988, McMaster 1991). Although the approaches were intended to be general, the resulting sets of rules are usually very specific and scale-related (Mark 1991, McMaster 1991, Shea 1991). Moreover, the top-down approaches have thus far scarcely led to operational systems (McMaster 1995).

The French IGN takes a bottom-up approach to achieve a comprehensive model for generalisation (Ruas and Lagrange 1995). They integrate their existing generalisation algorithms and routines - which are largely cartographic in nature - to create a comprehensive system. The desired end result is clear from the start of the procedure and resembles a traditional paper map. The operators they established are basically cartographic and geometric in nature; there are no operations based on attribute values such as class generalisation. The focus is on local operators and semantic rules are scarce. Bottom-up approaches like this lead to operational systems but as valuable as the developed operations are, it is questionable whether the general scheme is applicable in interactive, digital mapping.

2.5 Knowledge acquisition

Several authors have described knowledge bases for generalisation purposes and stressed the need of this information to enable automated generalisation (McMaster 1991, McMaster 1995, Shea 1991, Mark 1991). Knowledge-based systems require formalised expert know-how. This calls for knowledge acquisition; capturing human knowledge and structuring it into a computer-implementable form. In the case of map generalisation: this means research into the formalisation of expertise applied by cartographers.

Several procedures for knowledge acquisition have been mentioned, ranging from the analysis of text documents to sophisticated methods such as machine learning and neural networks. While the more advanced methods have not or hardly ever been implemented, the more straightforward methods invariably led to very specific knowledge or rule bases (Mark 1991) that lack meta-rules (Shea 1991) and hierarchical ordering, which is considered essential for effective expert systems (Hayes-Roth et al. 1983).

An alternative approach worth mentioning is the *amplified intelligence* method where the human expert initiates, controls and evaluates automated procedures that substitute labour-intensive tasks. Entire generalisations can be rerun and analysed to allow knowledge acquisition by logging the user's interactions (Weibel 1991).

Manually generalised maps are still considered *the* source of information for the creation of rule-bases. However, this is debatable because of the different requirements of digital mapping (Müller et al. 1995) and the questionable quality of material that has been generalised manually (João 1998). Rules for manual generalisation are generally not suitable for automated use either, being either too vague or too specific (Robinson 1995).

2.6 Rule-based systems

Rule-based systems use either a predetermined rule execution sequence or an inference engine to control the rule execution sequence. Many authors have mentioned the use of inference engines (Shea 1991, Armstrong 1991, Keller 1995), but we have not found any operational systems. This could be a consequence of problems with connecting the technology to geographic information systems but it could also arise from unpredictable results. Either way, actually creating generalisation expert systems based on inference engine technology seems difficult. Attempts to create generalisation systems based on predetermined sequences of generalisation rules were more successful. It is difficult to determine whether this is the result of intrinsic advantages of the method or whether such systems are simply easier to realise.

The OSGEN system, developed for the Ordnance Survey of Great Britain, is an expert- or rule-based system that uses a hierarchical approach with composite objects. It uses aggregation hierarchies for built-up areas containing object types like 'building', 'building group', 'block' etc. OSGEN was developed for the generalisation of large-scale data and uses rather specific rules, addressing particular situations and object classes. Such a rule base works fine in well-defined situations such as a typical large-scale Ordnance Survey map, but fails in less specific situations (Robinson 1995). Rulebases for generalisation are generally considered to be strongly application-dependent (Kilpelaïnen 1992).

Richardson developed a system for the generalisation of datasets for the National Atlas of Canada (Richardson 1993). Land cover, hydrography and populated places are generalised separately but are related through tables of 'necessity factors' for the different target scales. The base scale is 1:1 million, the target scales range from 1:2 million to 1:30 million. A necessity factor determines the percentage of objects in a certain object class to be retained at the specified scale. The necessity factor depends on the map's purpose (spatial analysis, map design), the subject of the map, the intended scale, and the functional requirements (orientation etc.). It is defined on the basis of interviews with cartographers as well as reviews of existing map series. An attenuation factor can be applied to the necessity factor to fine-tune the generalisation.

Map generalisation is always a battle between what should be maintained - based for the most part on the importance of the object class for the application - and what cannot be represented in the available presentation space, i.e. the reduction in the number of objects needed for the target 'scale'. These two aspects are typically translated into parameters (Richardson 1993, van Oosterom 1995).

2.7 Assessment of generalisation effects

Generalisation effects can be evaluated visually or automatically. Quantitative measures for the evaluation of generalisation results are global, geometrical, topological or software-related (Weibel 1995). Global measures include object density and distribution. Some parameters merely reflect the decrease in the number of objects while others imply that an error is introduced during the aggregation process. The error occurs when small polygons are merged with adjacent ones and reclassified. The attribute change index (Bregt and Bulens 1996) expresses this shift in attribute classes whereas the area reduction index (Bregt and Bulens 1996) and reduction factor (Richardson 1993) demonstrate a mere decrease in polygon numbers. Geometrical measures comprise objects that are too small or too close as well as changes in line length and sinuosity. Displacograms (João 1995) provide measures for the lateral displacement of map objects. Violations of topological relations can occur in the form of unintended intersections or broken connections. Software-related measures include labour-intensiveness, speed and equipment cost.

Manual generalisation often causes large inaccuracies. Manually generalised maps by national mapping agencies, for instance, show displacements that are significantly larger than what is considered acceptable. That is why for GIS analysis data of the largest possible scale must be used or else automatically generalised data, which are preferable to manually generalised material (João 1998). This raises the question whether manually generalised data should be used to assess the results of automated generalisation processes (Müller et al. 1995).

2.8 Multi-scale data models

Creating 'intelligent zoom' applications in which the information density increases while zooming in and decreases while zooming out (Timpf and Frank 1995). In the multi-scale approach, various cartographic representations of a single object are stored to allow for viewing at different levels of abstraction. The levels of abstraction are linked by a tree structure; the level of the tree to be displayed is dependent on the zooming level (Jones 1991, van Oosterom 1995, Timpf and Frank 1995). As one moves up and down the tree structure, objects are combined or split, drawn or omitted etc.

A method specifically aimed at area partitionings is the Generalised Area Partitioning (GAP)-tree (van Oosterom 1995). The GAP-tree ensures that no gaps appear where objects are left out, but that instead a higher-level object is returned. The tree is a pre-computed and stored aggregation hierarchy, based on a class generalisation hierarchy and the length of the common boundaries between the objects. Objects with low 'importance', a function based on object type and size, are aggregated first. No method is provided for determining these importance factors, the classification hierarchy is that of DLMS DFAD (DGIWG DIGEST 1992). The reactive tree (van Oosterom 1990) enables a fairly constant number of objects to be returned, irrespective of the zoom level.

Multi-scale research is aimed at connecting and accessing existing geographic databases (of the same area) of different levels of abstraction rather than creating the different levels. The multi-scale database can be created from a single detailed database or built from various existing datasets. The first method still requires generalisation methods to be developed. The main difficulty of the second option is in matching the objects in one dataset to objects in the other (Sester et al. 1998, Uitermark 2001). Some have connected the multi-scale database approach with the issue of update propagation from the detailed level to the more abstract levels (Kilpelaïnen and Sarjakoski 1995), which leads us back to the initial issue of defining generalisation operations.

2.9 Summary

The more advanced area aggregation methods (van Oosterom 1995, Bregt and Bulens 1996), i.e. methods that are not solely based on geometry, all use importance factors and compatibility measures for the object classes. It is commonly acknowledged that these factors are difficult to determine. There are virtually no examples of generalisation systems that use functional aggregation hierarchies.

The 'top-down' method of designing generalisation methods has yet to result in an operational system. The 'bottom-up' methods have led to working systems with shortcomings, especially strong application dependency. Nevertheless, the bottom-up approach did and does lead to operations that will be required once a universal generalisation framework has been defined.

The multi-scale database approaches add little to the actual generalisation process, i.e. creating the levels of abstraction. They do, however, contribute to the development of data models which support composite entity addressing.

Existing maps and rules for manual generalisation appear not to be suitable as a source of information for automated generalisation for digital mapping. Neither is scale a suitable unit of measurement for conceptual generalisation.

Chapter 3: Conceptual data model

3.1 Introduction

Data models for spatial information come in different shapes and sizes. Some data models are best suited for storing and retrieving large amounts of data. Others are designed to support certain spatial analyses. A data model that suits one purpose might be less appropriate for another. There is no ultimate data model, which is not a problem as long as we have the possibility to translate the information from one model to another without losing relevant information. Generalisation research has long suffered from a lack of appropriate data models for geographic information. The models were intended for drawing maps and were not meant to support complex spatial operations like generalisation (Grelot 1986). In the introduction we stated that generalisation is a form of context transformation, this means that the data model has to support context information. But before the matter of context can be explained, an introduction is in order, an introduction starting with the way reality is modelled in geographic information systems.

3.2 Object versus field approach

A dataset is the representation of a certain perception of reality in the computer, a model of reality. The representation of real-world phenomena in a geographic information system incorporates the choice of an appropriate conceptual model. Spatial phenomena with identifiable boundaries are generally modelled using the *object approach*. In the object approach, spatial units - the objects - are identified and attributes are attached to these units. In situations where the spatial variation is more gradual the *field approach* is commonly used. In the field approach continuity is the principal element; attribute values are directly linked to positions in space (Laurini and Thompson 1992). The discrete character of topographic data argues in favour of the object approach for this application.

3.3 Objects

Objects have spatial (geometric) and thematic attributes (Molenaar 1989, Figure 2). One example of a thematic attribute is 'colour', with the *attribute value* 'red'. A spatial attribute is the position of a node with values (x_n,y_n). These examples show that attributes can either be simple or atomic, or composite (Elmasri and Navathe 1989). In a geographic database comprising objects, the objects are distinguished by the values assigned to their spatial and thematic attributes. Objects can be distinguished if at least one of their attribute values differs. The fact that the spatial attributes also identify an object means that two thematically identical objects can be distinguished if their positions are different. But objects usually get a unique *object identifier* (Figure 2), often in the form of a number.

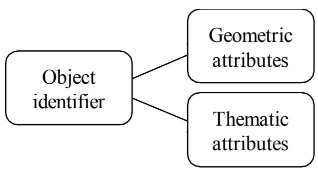


FIGURE 2. OBJECTS CONSIST OF A THEMATIC AND A GEOMETRIC DESCRIPTION

Objects are defined according to a certain context (Richardson 1993). A single object can be classified in different ways depending on the context. A road object can be classified based on its importance (motorway, secondary road etc.) or according to its management (e.g. provincial or national road). A building has a different meaning to a utility company than it has to a landscape planner. Both are interested in different properties of the building and might therefore employ different definitions of a building.

In section 1.1 the context of information transfer was described as: *what* is communicated *to whom* and *why*? But where do we find this information in a geographic dataset? According to Bishr (Bishr 1997), context information finds its expression in the combination of:

- classes
- class intension
- geometric representation.

These concepts are explained in the following sections.

3.4 Classes

In order to comprehend large amounts of data, people tend to categorise. We speak of houses, roads, trees etc. because we deal with them in different ways. These general notions refer to a number of actual objects that can be represented and operated in the same way although not even one is completely identical to another. This concept of categorising things into groups, or *classes*, based on similarity is called *classification*. The relationship between objects and classes is of the *'is a'* type.

3.4.1 Data class

A common way of indicating that an object is a *member* of a class is to store the class name to the object in the form of an attribute value. The objects are in this case assigned to *data classes* (Molenaar 1998). All objects reside in the same database table and therefore share the same attribute structure. Attributes that do not apply to certain classes are left empty. The database is therefore not fully *normalised*. A data class is a class for which the intension can be specified by means of attribute values.

3.4.2 Object class

Sets of objects that are so different that this cannot be expressed by their attribute values alone, should have their own sets of attributes (Molenaar 1998). Such a set of objects with a common attribute structure is called an *object class* (Molenaar 1993, Richardson 1993). The list of attributes of an object class is unique to that object class and distinguishes the class from other object classes. The individual objects belonging to the object class are called *instances* of the class. The collection of instances is referred to as the *extension* of the object class.

3.4.3 Taxonomies

Some classifications are more detailed than others. If an object is said to be a building, one roughly knows what it will look like, but it can still be many diverse kinds of building. If it is referred to as being a house, one can make a far more accurate estimation of its properties. The class of buildings is called a *superclass* of the class 'house', the class 'house' is a *subclass* of the class 'building'. The subclass presents a more detailed description of the object. Thus, classification may proceed from the bottom up, that is by grouping individual occurrences to make larger sets, or it may subdivide existing groups into subgroups (Laurini and Thompson 1992). Subclasses and superclass are related in the *generalisation plane* (Smith and Smith 1989), also called *classification hierarchy* or *taxonomy* (Figure 3). The basic *taxonomy relationship* or *taxon* is between a subclass C_{sub} and its superclass C_{super} (Uitermark 2001):

$$TAXON[C_{sub}, C_{super}] = 1$$

If such a taxonomy relationship exists, then the extension of C_{sub} is a subset of the extension of C_{super}:

$$Ext(C_{sub}) \subset Ext(C_{super})$$

This means that an object o_i that is a member of class C_{sub} is also a member of C_{super}:

$$MEMBER[o_i, C_{sub}] = 1 \Rightarrow MEMBER[o_i, C_{super}] = 1$$

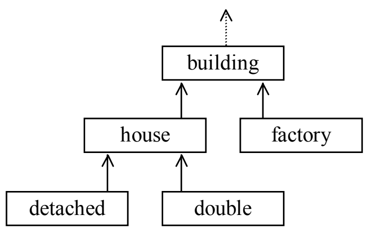


FIGURE 3. PART OF A TAXONOMY

Moving up in the taxonomy is called *class generalisation*, moving down *class specialisation*. As we move up and down in a taxonomy not only the class label changes, but more importantly the object is referenced in different ways, revealing different levels of detail (Figure 4). Class generalisation should not be confused with the broader use of the word generalisation in map generalisation, which encompasses many other operations.

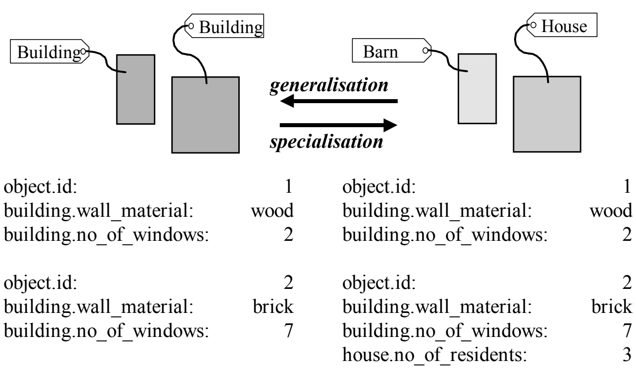


FIGURE 4 SUBCLASSES AND SUPERCLASSES REVEAL DIFFERENT LEVELS OF DETAIL

Taxonomies can also be implemented as *data classes*, either in the form of hierarchical attribute values, or attached to the basic classes in the form of a related table. In that case class generalisation and specialisation do not affect the number of attributes of an object, since no distinct object classes are discerned.

3.4.4 Inheritance

An important aspect of taxonomies in object-oriented modelling is the inheritance of attributes from the top down. An *object subclass* inherits the attributes of the object class it descends from and gets some additional attributes that are specific for that subclass. For example, houses and barns are both buildings and share properties such as ‘wall material’. The subclass ‘house’ restricts the degree of freedom of the object by adding attributes and limiting the domain of existing ones. For example, the attribute ‘no. of residents’ is part of the class description of a house but not of buildings in general (Figure 4). The attribute is inappropriate for some buildings, like barns and office buildings. The other way round: if an object is referenced as a member of a certain *object superclass* instead of its ‘normal’ object class, a number of attributes of the object becomes inapplicable.

3.5 Class intension

The *intension* of a class is the set of conditions that applies to all members of a class and not to any object that is not a member of the class (Richardson 1993). It is by this set of conditions that objects are associated with a class. It is important to bear in mind that no actual instances are captured; only rules and possible values of the attributes (i.e. their domains) are defined (Bishr 1997). A class intension often includes the class intension of another, higher-level class (superclass), incorporating additional conditions. For instance: 'village'; a 'populated place' with 100 to 10,000 inhabitants. 'Populated place' is in this example considered to be a class intension that was defined earlier.

3.6 Geometric representation of objects

3.6.1 Raster and vector representation

In geographic information systems two types of geometric representations are distinguished. Rasters are based on a regular grid of raster cells, every cell having a value for a specified attribute. Rasters are generally used to represent field data, but can also be used to store objects. In a vector representation, spatial objects are represented by points, lines and faces (and bodies in three-dimensional space). The vector representation is generally used in combination with the object model. It can, however, be used to describe field data, as in the direction and force vectors indicating wind in meteorological applications. Topographic datasets are typically stored in vector format.

Although physical geographical entities occupy space in three dimensions, most current GIS applications only support two-dimensional data storage and processing. This study is also limited to two-dimensional spaces.

3.6.2 Formal data structure

An important element in the determination of spatial inter-object relationships is topology. Topology can be stored or derived when needed. We adapt the Formal Data Schema (FDS) vector model as described by Molenaar (Molenaar 1998), a data model with stored topology. Objects in the FDS are composed of nodes, segments and faces. These are called the geometric primitives (Figure 5). The geometry of a node is expressed in the form of an x,y co-ordinate. Segments are lines or polylines between a begin node and an end node and are consequently directed.

Segment s_a has node n_c as the begin node	$\rightarrow BEGIN[s_a , n_c] = 1$
Segment s_a has node n_d as the end node	$\rightarrow END[s_a , n_d] = 1$
Node n_c is a node of segment s_a	$\rightarrow NODE[s_a , n_c]$ $= BEGIN[s_a , n_c] + END[s_a , n_c] \neq 0$

Segments do not cross; when this occurs they are split into four segments joining at a common node.

Faces are described by their boundary, a polygon consisting of one or more segments. References to the left and right adjoining faces are stored with each segment. Faces do not overlap, segments have one face on each side:

$LEFT[s_a , f_g] = 1$ and for any $f_i \neq f_g \Rightarrow LEFT[s_a , f_i] = 0$ and

$RIGHT[s_a , f_h] = 1$ and for any $f_j \neq f_h \Rightarrow RIGHT[s_a , f_j] = 0$

Segment S_a is part of the boundary of face f_g if:

$$BOUNDARY[s_a , f_g] = LEFT[s_a , f_g] + RIGHT[s_a , f_g] = 1$$

The spatial description of objects consists of one or more of these primitives. In this study it is assumed that nodes and segments do not represent objects - only faces do - for the reason that it significantly simplifies the definition of topological queries based on the segment-face model.

Face f_g is part of area object o_k if:

$$PART_{22} [f_g , o_k] = 1$$

If each face is part of an area object, segments have an area object on each side:

$LEFT[s_a , o_k] = 1$ and

$RIGHT[s_a , o_m] = 1$

Segment s_a is part of the boundary of area object o_k if:

$$BOUNDARY[s_a , o_k] = LEFT[s_a , o_k]+ RIGHT[s_a , o_k] = 1$$

The set of segments $S_{\partial ok}$ that are part of the boundary of area object o_k is defined by the function:

$$S_{\partial ok} = \{ s_n \mid BOUNDARY[s_n , o_k] \}$$

Two area objects o_k and o_m are adjacent at segment s_a if:

$$ADJACENT[o_k , o_m \mid s_a] = 1 \Leftrightarrow s_a \in S_{\partial Ok} \cap S_{\partial Om}$$

The value of this function can be found by means of the following expression:

$$ADJACENT[o_k , o_m \mid s_a] = Max(Min(LEFT[s_a , o_k], RIGHT[s_a , o_m]), Min(LEFT[s_a , o_m], RIGHT[s_a , o_k]))$$

In large-scale topographic datasets the objects are typically represented by faces. Points and segments *are* used in other applications to describe objects for the reason that this type of primitive enables operations, e.g. route calculations on a network of segments, which would be more difficult or impossible if the objects were represented by faces.

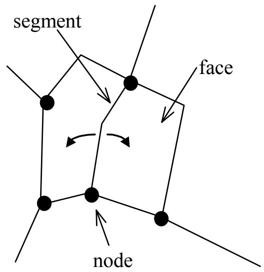


FIGURE 5. THE FORMAL DATA STRUCTURE (FDS)

A map[2] is the representation of a certain perception of reality, a model of reality. The set of objects in the map is called the *universe of discourse* of that map (Molenaar 1998). The *area of interest* (Molenaar 1998) is a spatially delimited region for which the identified universe of discourse is described.

The set of area objects covers the complete area of interest and objects do not overlap, i.e. the objects form a *geometric partition* (Molenaar 1998) or *area partitioning* (van Oosterom 1995). If, furthermore, any object in the dataset is a member of some class and each object is a member of one class only, then these classes from a *thematic partition* of the dataset (Molenaar 1998). If the objects form a geometric and a thematic partition we speak of a *single-valued vector map* (SVVM, Molenaar 1998).

Dichotomous or presence/absence maps contain objects belonging to only one class. Categorical or multi-class maps contain objects of several classes. Categorical themes form a thematic and geometric partition.

3.7 Inter-object relationships

Like real-world objects, the objects in a geographic dataset do not occur independently; they are related to other objects. In the previous sections two types of inter-object relationships have been implicitly introduced; objects could have common attribute values or belong to the same class. These are both examples of thematic relationships based on similarity of object properties. Here another type of thematic relationship between objects is introduced, the functional relationship, as well as the spatial relationships that can occur (Figure 6). Within the category of spatial relationships topological and geometric relationships are distinguished but only topological relationships will be discussed further. In the case of a thematic relationship the objects

[2] We refer here to the electronic, GIS form of a map

are related through their thematic attributes; a spatial relationship between objects means that they are related through their geometric attributes (see Figure 2).

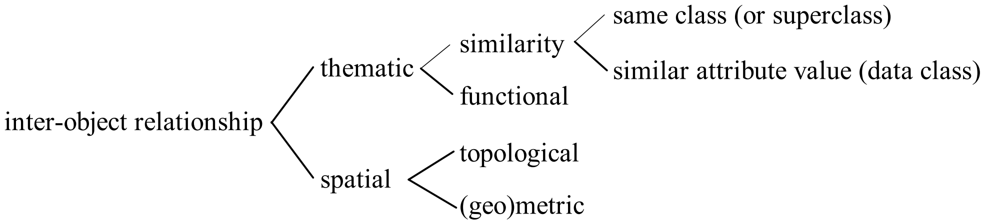


FIGURE 6. TYPES OF INTER-OBJECT RELATIONSHIPS

3.7.1 Thematic relationships

Objects can be thematically similar in the sense that they belong to the same class or share a common superclass. Objects can also have similar attribute values. The class relationship and the attribute value relationship are both based on similarity between the objects involved. But there is also another type of thematic relationship: the *functional relationship*. This relationship will become clearer if we look at a city. A number of different components can be distinguished: roads, buildings, parks etc. These components are not related in the sense that they are similar, but when put together a new object, the city, is formed. They are related in a functional sense, i.e. the city cannot function properly without any of these components. The components that make up a single city are also related in a spatial sense; only mutually connected components can create a functional city.

3.7.2 Spatial relationships

An important type of spatial relationship is topology. In a topological description the exact locations of the nodes and the paths that arcs follow are not important as long as the contiguity is represented correctly (Gatrell 1991). The two vector graphs in Figure 7, for example, are topologically identical, even though they have distinct geometric descriptions. The adjacency graph is identical for both situations (Figure 8). Topology can be described for 1) connected objects, e.g. Arc/Info vector format, FDS (Molenaar 1998), and 2) unconnected objects, using a Delauney triangulation to connect the objects in order to enable extended adjacency relationships, e.g. EFDS (Peng 1997), IEFDS (Liu 2002).

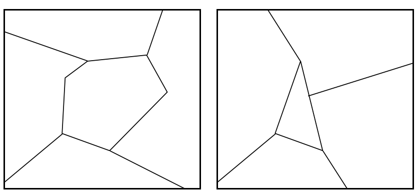


FIGURE 7. TWO GRAPHS WITH DISTINCT GEOMETRY BUT EQUAL TOPOLOGICAL DESCRIPTIONS

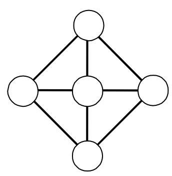


FIGURE 8. ADJACENCY GRAPH FOR BOTH GRAPHS IN FIGURE 7

A common use of topology is for the determination of an object's neighbours. In a single-valued vector map the connected neighbours of an area object are those objects that have an arc (first-degree neighbour) or node (second-degree neighbour) in common with the object. We will work only with first-degree neighbour relationships in this study.

An important aspect of spatial inter-object relationships is the determination of the level of abstraction at which the relationship occurs. Mixed forest is forest where individual coniferous trees and deciduous trees are found next to each other, not where plots of deciduous forest and plots of coniferous forest occur side by side.

3.8 Composite objects

Objects that are in some way, thematically or spatially, related and form a meaningful unit at a higher level of abstraction can be represented as *composite objects*. A composite object or *aggregate* is a relationship between two or more objects seen as a new object (Smith and Smith 1977, Hansen and Hansen 1992). Objects can therefore consist of other objects, thus creating an *object hierarchy* (Molenaar 1998). The relationship between a component and a composite object is of the *'part-of'* type.

The operation of creating a composite object from its components is called aggregation. Aggregation is explained further in chapter 4. Composite objects are not necessarily contiguous. They can consist of several spatially unconnected parts. Starting point for *this* study, however, is that the composite objects *are* contiguous.

Both the geometry and the thematic information of composite objects are normally derived from the geometric and thematic information of their component objects, although it is possible to append additional information (Figure 9). When objects are aggregated their thematic descriptions become one. If composite objects are derived completely through rules, they can only have derived attributes like area and the portions of original object classes in the composite object. At higher abstraction levels the derived attributes could be 'road density', 'fragmentation index' etc. The behaviour of attributes under aggregation depends on their domain: the type, units and scale type used to describe the attribute.

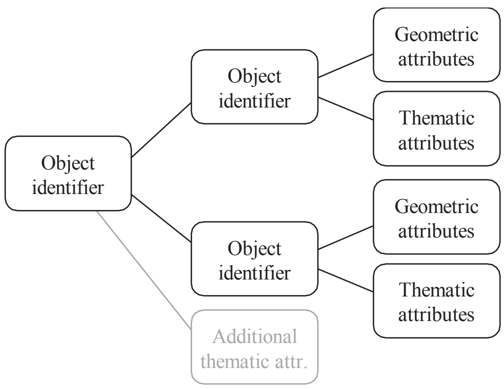


FIGURE 9. COMPOSITE OBJECT CONSISTING OF TWO ELEMENTARY OBJECTS

Composite objects are an important concept, not only for generalisation purposes - spatial abstraction depends highly on the possibility to combine objects - but for data and geo-data modelling in general. Datasets offer much more flexibility for querying and processing if multiple levels of object abstraction are distinguished. For example, road networks can be identified per segment (Figure 10a) or entire roads can be identified (Figure 10b). But it is often preferable to distinguish more than one object level (Figure 10c). This offers the most flexibility in constructing other objects different from the ones which were initially defined. It allows the identification of road segments as well as roads. It is easier to combine spatial objects than to break them apart, as it is necessary to add geometry to split up objects.

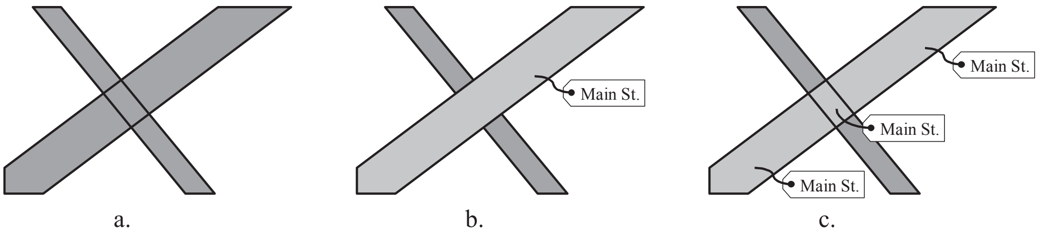


FIGURE 10. SINGLE OR MULTI-LEVEL CLASSIFICATION OF ROAD OBJECTS

3.9 Composite classes

Like a composite object, a *composite class* can be seen as a *relationship type* between two classes viewed as a new object class. In the case of Figure 11 the topological relationship type between the classes ‘building’ and ‘lot’ (buildings are on lots) is seen as the composite class ‘property’. The constitutive classes are called *component classes*. Component classes are non-exclusive, they can be shared by several composite classes. At the same time, composite classes are related to 1, 2 or more component classes. The relationship type between component and composite classes is therefore M:N.

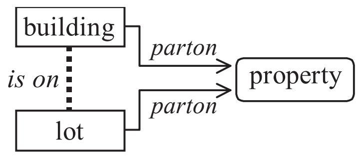


FIGURE 11. EXAMPLE OF A RELATIONSHIP BETWEEN TWO COMPONENT CLASSES BECOMING A COMPOSITE CLASS

The hierarchy of relationships between composite and component classes is called *aggregation hierarchy* or *partonomy* (Uitermark 2001). The basic *partonomy relationship* or *parton* is between a component class $C_{component}$ and its composite class $C_{composite}$ (Uitermark 2001):

$$PARTON[C_{component}\,,\, C_{composite}] = 1$$

The partonomy presents *possibilities* for object aggregation.

3.10 The basic conceptual abstraction operations

3.10.1 Aggregation

Grouping multiple individual objects to form a new composite object is called *aggregation* (Figure 12, Frank and Egenhofer 1988). A number of component objects create a single composite object at the next higher *aggregation level* in the *object hierarchy*. If the component objects are adjacent, their spatial descriptions are usually merged, but it is also possible to just assign a common identifier to the components. *Merging* and aggregation are therefore two distinct operations. Aggregation mainly involves combining the thematic information attached to the objects, whereas merging only concerns the spatial component.

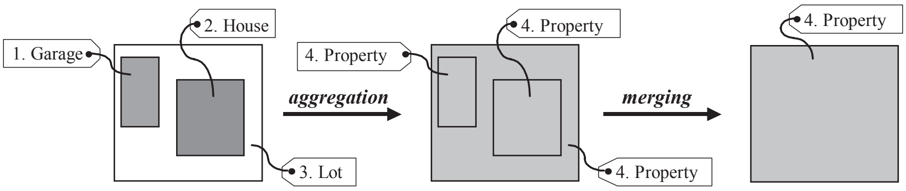


FIGURE 12. AGGREGATION FOLLOWED BY MERGING OF THE COMPONENTS' GEOMETRY

3.10.2 Elimination

If we assume a dataset with area objects to constitute a single-valued vector map, elimination can be considered a special case of aggregation. One of the consequences of a geometric partition is that objects cannot simply be removed, as this would create gaps. If an object is eliminated the released area has to be reclaimed by an adjacent object in order to maintain a geometric partition.

The set of faces $F = \{..., f_i , ...\}$ form an area partition; every face is part of an object:

$$\forall f \in F, \exists o \Rightarrow Part\,[\,f, o\,] = 1$$

Object o_k is removed but becomes part of adjoining object o_m:

$$PART[\, o_k , o_m \,] = 1$$

Then the faces of o_k become part of o_m , retaining the area partition:

$$PART[\, f, o_k \,] = PART[\, f, o_k \,] * PART[o_k , o_m \,] \Rightarrow PART[\, f, o_m \,] = 1$$

This means that objects have to be combined just as in the case of aggregation, but with this distinction that the properties of the eliminated object do not become part of the composite object's description (Figure 13, Figure 14). An example of elimination in a categorical area partitioning is the frequently used method of merging small area objects with one of their neighbours based on the largest common boundary.

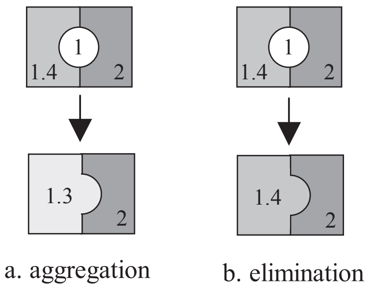


FIGURE 13. THE DIFFERENCE BETWEEN AGGREGATION AND ELIMINATION (QUANTITATIVE ATTRIBUTE)

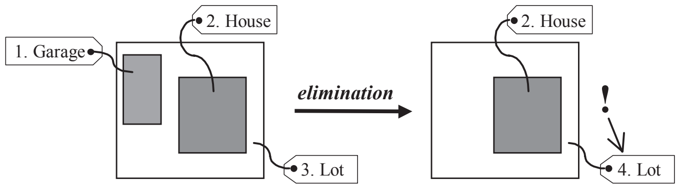


FIGURE 14. ELIMINATION (NOMINAL ATTRIBUTE)

A dataset is a description of reality and generalisation is just a way to make reality manageable; objects removed in a generalisation process still exist in reality. The fact that elimination is considered to be a kind of aggregation is related to the fact that all objects are more than one-dimensional. If an object is removed from the dataset, its area has to be merged with another, adjacent object, thus contaminating this adjacent object. So in fact a new, contaminated object is created.

Elimination can be predetermined, or be the result of what started as an aggregation operation. If one of the objects of the composite is insignificant compared to one or more of the others, it is possible that the composite object is not reclassified completely, but that instead the properties of the insignificant component are ignored. In this case the process which started as an aggregation ends up as an elimination. If the elimination is predetermined, the aggregation relationship is usually strictly spatial, i.e. the eliminated object is merged with one of its neighbours. There is generally no thematic relationship between the object and its neighbour.

Elimination is application-independent and can be used when shifts in the thematic content are not a problem. If a choice can be made for an object to be eliminated or aggregated, aggregation is generally to be preferred for its advantage of maintaining the most thematic information.

3.11 Summary

In this chapter a topological data model was introduced. This model contains the spatial relationships needed for aggregation. The aggregation operations are also supported by a partonomy or aggregation hierarchy of object classes and composite classes to determine aggregation options and by an object hierarchy of objects and composite objects to enable storage of the aggregation results. The difference between aggregation and elimination in a geometric partition was explained. The third type of hierarchy introduced is the taxonomy or classification hierarchy in which objects are members of classes and classes may be generalised to superclasses. In the next chapter the operations mentioned above - aggregation, elimination and class generalisation - will be the constituents of conceptual abstraction strategies.

Chapter 4: Abstraction process

Having described the data model, the abstraction process can be defined. It is obvious that the abstraction process depends highly on the possibilities offered by the data model, because the new, abstract descriptions are derived from and stored in the described model. In this chapter we zoom in on strategies and operations for the conceptual generalisation of a categorical area partitioning.

4.1 Thematic and spatial abstraction

Conceptual generalisation of geo-data has two aspects, thematic and spatial abstraction, which are - as we will see later - closely related. We speak of *thematic abstraction* if the number of distinct attribute values is reduced, i.e. the domain is limited. This reduction is achieved by classifying quantitative values. If the attribute values refer to classes already, the domain can be limited by moving to superclasses. Thematic abstraction is the conceptual GIS equivalent of combining legend items in traditional cartography. *Spatial abstraction* is achieved by reducing the number of objects by means of aggregation or elimination. Reducing the number of spatial dimensions of an object, for example, converting area objects to point objects, is not considered a conceptual generalisation operation within the framework of this study. Although it *is* an example of context transformation, not every context transformation is a generalisation. Spatial abstraction can be compared to simplifying the map image in map generalisation. Thematic abstraction triggers spatial abstraction when objects with an identical class value after classification or class generalisation are subsequently aggregated. Spatial abstraction frequently depends on a preceding thematic abstraction. In this study, the prime objective is spatial abstraction.

Abstraction is based on inter-object relationships. This means that we can draw a scheme of abstraction types which looks very similar to the diagram of relationship types in the previous chapter (Figure 15).

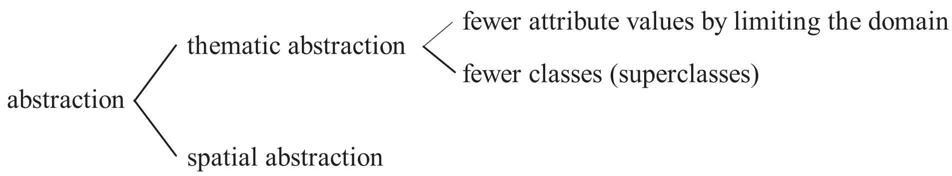


FIGURE 15. TYPES OF ABSTRACTION

4.2 Categorical maps need a holistic approach

It is possible to split up a categorical map into several dichotomous maps, i.e. to ‘lift out’ certain classes in order to generalise these maps independently and recombine them afterwards. Infrastructure, for example, can be part of a land cover map but may also create an independent map (Figure 16). In the first case the map is categorical, in the second case we have a dichotomous map.

FIGURE 16. 'LIFTING' A CLASS OUT OF A CATEGORICAL MAP TO CREATE A DICHOTOMOUS MAP

Generalising the different classes independently is only possible if the objects of these classes are represented by lines or points in the initial, categorical map. If area objects were eliminated, holes would appear after recombination of the maps. But even with point and line objects this method is problematic, because important relationships between objects belonging to different classes can be overlooked (railway bridge without a river). Categorical maps really demand a holistic approach, i.e. dealing with different types of objects and the interaction between them – not dividing the map up into layers containing only one type of objects and then attempting to recombine them after generalisation (Bundy et al. 1995).

Nevertheless, splitting up a categorical map into several dichotomous maps *can* be beneficial, even if the map only contains area objects. This is when we want to perform a database enrichment process (see next chapter). The objects can be classified using the dichotomous map, according to spatial structure, for instance, and this classification can subsequently be transferred to the original, categorical map to aid in its simplification.

4.3 Metaclasses

There are a number of problems in using existing rules. Rules for manual generalisation are generally only applicable within a limited resolution range because generalisation is usually carried out from one mapscale to the next. They also leave too much room for individual interpretation to be used in an automated environment. The additional knowledge used by the cartographer turns out to be difficult to identify and formalise, and the resulting rules are therefore either too general (e.g. if an object is too small it can be omitted) or too specific (e.g. if a lake is too small it should be omitted, except if it is part of a set of small lakes close to each other, in which case… (Ruas and Lagrange 1995)). Most rule-based systems described in literature are based on rules that are too specific (Shea 1991, McMaster 1991). The rules tend to be very detailed and address particular classes of specific datasets.

Because of these problems we opt for a different approach. In order to make the rule-base less dependent on the dataset a typology of classes would be beneficial. That way, rules can be defined for metaclasses instead of specific classes. Metaclasses are categories of classes that share some characteristics (Molenaar 1998). We distinguish three metaclasses: classes containing 'network-forming' objects like roads; classes with relatively small 'island' objects, e.g. buildings; and the remaining 'normal' area objects with parcels as a typical example. The properties of the objects in these classes offer possibilities for database enrichment and at the same time offer opportunities, but also set constraints, for aggregation.

Although this distinction may only appear to hold true for 'small-scale' datasets, where infrastructure is represented by line segments and settlements by points, this distinction also applies to 'large-scale' datasets where all objects are represented by faces. Although all objects are represented through faces, some objects can be characterised as 'linear', e.g. roads and waterways, while others are 'point-like', e.g. buildings.

Island and 'unimportant' network segments are ideal candidates for elimination whereas it is generally not advisable to merge adjacent objects with objects belonging to a network structure, because this might disturb network topology in case an accidental shortcut is created (Figure 17). However, small 'island' objects, e.g. refuges, which are fully enclosed by objects belonging to the network, could be merged into the network without consequences.

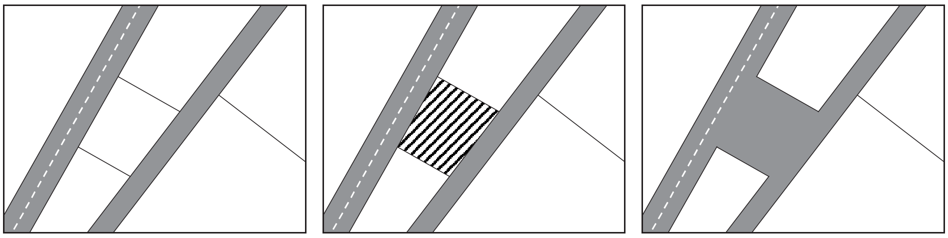


FIGURE 17. DISTURBING NETWORK TOPOLOGY BY AGGREGATION

Elimination based on simple properties like size is common in classes of 'island' objects. Objects that belong to network structures, like road and waterway systems, cannot simply be eliminated based on size. That way, structurally essential elements could accidentally be removed (Figure 18). Instead, elimination should be based on the importance or rather unimportance of the object within the network structure. This requires a classification of network segments based on the topological structure of the network. Usually, these more sophisticated classifications are not readily available. Pre-processing of the data or *database enrichment* is therefore required. Once importance-based classifications are available they can also be used for aggregation; with least important classes, such as blind alleys, as likely candidates for aggregation or elimination.

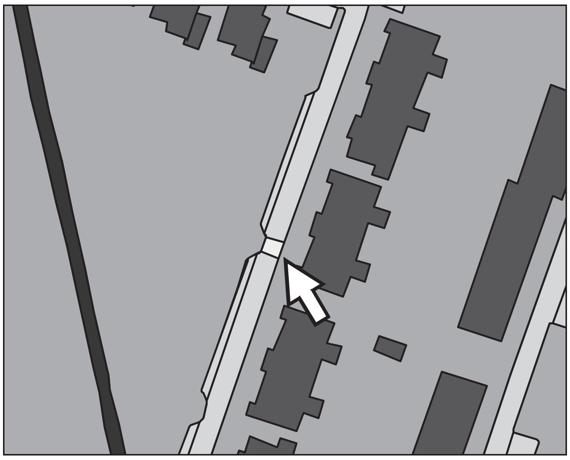


FIGURE 18. EXAMPLE OF SMALL ROAD SEGMENT THAT IS VITAL IN RETAINING CONNECTIVITY

4.4 Strategies for conceptual geo-generalisation (aggregation methods)

Molenaar distinguished four spatial generalisation strategies (Molenaar 1998). All are intended for the generalisation of single-valued vector maps, i.e. area partitions. The four types are:

- Geometry-driven generalisation
- Structural generalisation
- Similarity-driven generalisation
- Functional generalisation.

4.4.1 Geometry-driven generalisation

Geometry-driven generalisation is a rather crude method based on geometric relationships between the objects only. Objects, especially small ones, are eliminated by merging their geometry with a neighbouring object based on the largest shared boundary (Figure 19). Thematic compatibility between the objects plays no part in the process. In the case of nominal data, geometry-driven aggregation easily causes shifts in attribute values, enlarging some classes excessively while others may disappear.

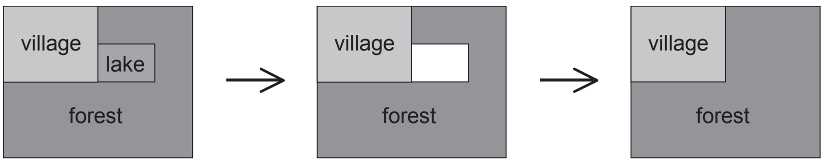


FIGURE 19. EXAMPLE OF GEOMETRY-DRIVEN GENERALISATION

4.4.2 Structural generalisation

Structural generalisation is based on the hierarchical relationships in a network structure. The elements of the network are usually represented by segments of a graph. Yet this network structure can be related to a geometric partition of area objects to direct the aggregation of the area objects, for example in the aggregation of catchment areas following the elimination of stream elements in order to retain a constant flow at the outlet (Figure 20, Martinez Casasnovas 1994). This approach was developed for quantitative information, such as catchment area and flow rate; whether the method can be applied to nominal, categorical data is as yet unclear.

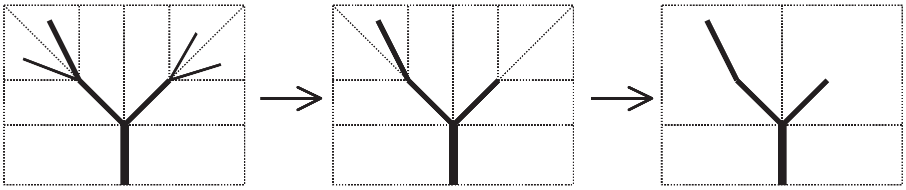


FIGURE 20. EXAMPLE OF STRUCTURAL GENERALISATION

In the case of geometry-driven generalisation and structural generalisation, reclassification becomes necessary after merging the units. This contrasts with similarity-driven and functional generalisation where reclassification drives the generalisation process. We will now go more deeply into the last two, being two of the more advanced methods for object aggregation in a *categorical* area partition.

4.4.3 Similarity-driven generalisation

Evaluating similarity relationships between objects requires the comparison of attribute values. Attributes with a ratio scale cause few problems, as the ‘distance’ between two attribute values can be measured objectively. Problems occur when the attributes have a nominal scale. These attributes often refer to classes, the values being class identifiers. It is obvious that different classes cannot readily be compared. These nominal data raise questions like: what is similar and which combinations are more similar than others? Bregt and Bulens encountered this problem and introduced the similarity matrix, a matrix in which every possible combination of class values is assigned a similarity measure. An important drawback of this approach is the labour-intensiveness of creating a similarity matrix (Bregt and Bulens 1996), and they are therefore hardly available. Moreover, similarity matrices are application-specific and can therefore not often be reused.

A special case of similarity-driven generalisation is class-driven generalisation (Molenaar 1998). Class-driven generalisation is based on the use of an existing taxonomy to establish whether the class values are related to a common superclass. In Figure 21, for example, we can see that ‘coniferous forest’ and ‘deciduous forest’ are both related to the superclass ‘forest’. These relationships can be used as a basis for aggregation. Since ‘deciduous forest’ and ‘coniferous forest’ are both types of forests, adjacent lots of both types can be aggregated, resulting in composite objects of the type ‘forest’ (Figure 22). This approach can only be followed if this kind of taxonomy is available. Like similarity matrices, taxonomies are context-dependent but reusability is better because the number of choices is smaller. The relationship is binary; there is a relationship or there is not, with, in contrast to similarity matrices, nothing in between.

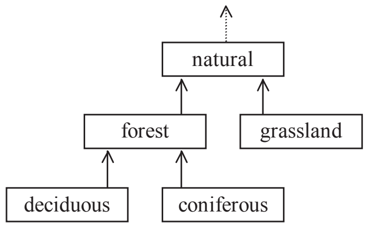


FIGURE 21. PART OF A TAXONOMY OR CLASSIFICATION HIERARCHY

FIGURE 22. EXAMPLE OF CLASS-DRIVEN GENERALISATION, BASED ON THE TAXONOMY IN FIGURE 21

Class-driven generalisation can follow the strict rules of object modelling, as the superclasses have a less intricate attribute structure. But often the classes involved are mere data classes and the taxonomy takes the form of hierarchical attribute values or a related table. Richardson's (Richardson 1993) method for the generalisation of land cover data is based on this principle. Another example is Bregt and Bulens' attribute class method (Bregt and Bulens 1996). Many datasets contain hierarchical attribute values that will support this practice. The Dutch topographic 'TOP10vector' dataset, for example, shows an hierarchical classification for roads.

4.4.4 Functional generalisation

Aggregation based on classification hierarchies works well in some cases, as examples by Richardson and Bregt and Bulens have shown, and is therefore a very valid approach, but not necessarily the right one to follow. It is often required that non-similar objects are aggregated in order to create meaningful composites, because:

- Adjacency of objects belonging to the same superclass is more or less coincidental. It is therefore far from certain that the spatial complexity of the dataset will reduce significantly.
- The approach only works within a limited spatial range as the objects basically remain the same, it is just the label that changes. Whether an object is referred to as a 'house' or a 'building', it is still the same object. The spatial resolution does not change.

Classes can only be applied within a limited range of spatial resolutions. The class 'building' for example is only valid within a limited spatial resolutions domain, since there are generally no buildings larger than a few hectares. If we want to describe an area at a higher level of abstraction, we will have to switch to classes like 'built-up area', composed of objects that bear little similarity, like buildings and roads. Thus, small steps towards higher levels of abstraction can be accomplished by merging similar objects. The larger steps should be based on other relationships, such as functional ones.

Functional relationships can play an important part in a spatial abstraction process, but they are only occasionally mentioned in map generalisation literature. One example related to topographic data that comes close was described by Robinson (Robinson 1995). He described an *aggregation hierarchy* of buildings, aggregated buildings, building groups and blocks (Figure 23). Also, Ruas and Lagrange (Ruas and Lagrange 1995) observed that a hospital is composed of a set of buildings and areas. But these are examples of the few times that functional relationships appear in map generalisation literature. It appears that most literature describes generalisation processes comprising relatively small generalisation steps, which can still be realised by elimination and reclassification based on similarity.

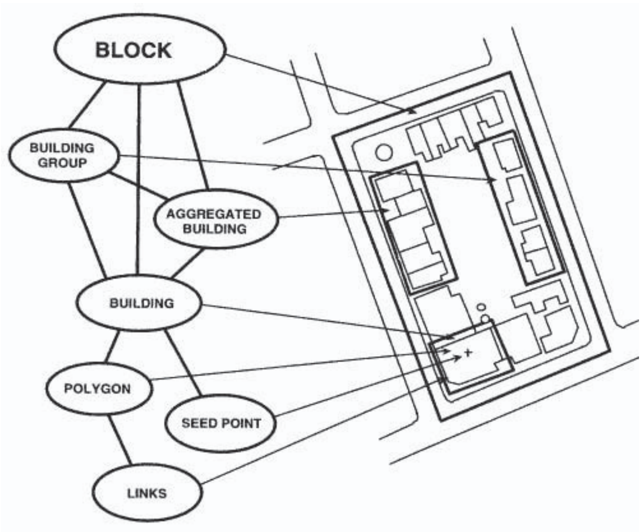


FIGURE 23. ROBINSON'S EXAMPLE OF AN AGGREGATION HIERARCHY

Functional generalisation is based on the functional coexistence of the objects involved. These functional relationships are generally much more interesting than relationships based on similarity. Take the example of a leopard. The leopard depends on a varied habitat with both forested areas and more open vegetation. An aerial image is available with a resolution of $5m^2$. In the image we will be able to recognise trees, shrubs, grassland etc., and we can aggregate adjoining trees and shrubs into larger areas of higher vegetation, keeping the larger areas that emerge and eliminating the smaller ones. However, in this process we will entirely lose the specific domain of the leopard which is in fact characterised by the variation in higher and lower, denser and more open vegetation. To find the leopard we will have to look at spatial units which contain both types of vegetation, high and low, and we have to look at the right level of abstraction; the open spaces must be large enough but not too large. To the leopard, the shrubs and open field are functionally related, the shrubs providing cover for the animal to stalk its prey grazing in the open.

But, just as with the methods based on similarity, the relationships need to be known beforehand. That is why we will now focus on the development of a method for automated determination of the relationships needed for functional aggregation.

4.5 Spatial co-occurrence of classes

How can we detect functional relationships between objects of different classes? Think of a workplace that consists of a chair and a desk. These items show less similarity than two chairs or two desks, but when put together they create a workplace, unlike two chairs or two desks. The fact that the items are found close to one another makes them a workplace. If we reverse this line of thought and look at an office without knowing the concept of workplaces we will probably only see chairs and desks. But, if we look closer we might observe that the spatial distribution shows that chairs and desks occur in pairs: one desk with one chair. There might be one chair too much or a chair away from its desk, but if we look at a test set that is statistically large enough, this relationship between instances of the class of chairs and instances of the class of desks will most likely show.

This approach is based on functional dependencies rather than similarity of the objects. It is assumed that the spatial correlation of the instances of classes can indicate an associative relationship, in conformity with Tobler's 'first law of geography': 'everything is related to everything else, but near things are more related than distant things' (Tobler 1970). It is also assumed that associative relationships can indicate functional dependencies.

4.5.1 Class adjacency index

The *class adjacency index (CAI)* is a global measure for the spatial adjacency of thematic classes; this index can be used to identify combinations of classes of which the mutually adjacent members might be aggregated (van Smaalen 1996b, van Smaalen 1999).

In the case of a single-valued vector map, the spatial adjacency or co-occurrence of classes is translated into cumulative adjacency for all members of the class. The class adjacency index of two classes will be evaluated by taking the sum of the lengths of shared boundary segments for all adjacent members of the two classes divided by the total boundary length of either of the two classes. By doing so for all combinations of classes in the dataset we can produce a list of values of class adjacency indices.

Note that the class adjacency index is directional, i.e. the class adjacency index between 'building' and 'lot' is different from the one between 'lot' and 'building' (see Table 1). The class adjacency index is defined as:

$$CAI\ (C_a \mid C_b) = \frac{\sum_r Length(s_r \mid \mathrm{ADJACENT}[o_d, o_e \mid s_r] = 1)}{\sum_t Length(s_t \mid \mathrm{BOUNDARY}[s_t, o_d] = 1)}$$

with $\mathrm{MEMBER}[o_d, C_a]=1$ *(object* o_d *is a member of class* C_a*)*
and $\mathrm{MEMBER}[o_e, C_b]=1$ *(object* o_e *is a member of class* C_b*)*

Many aggregation studies use measures that express compatibility between classes (van Putten and van Oosterom 1999, Bregt and Bulens 1996). The class adjacency index can be seen as a type of compatibility measure based on topological information.

4.5.2 Cardinality

For efficiency reasons it is important that only class combinations that appear in a significant number of cases are reclassified. Each class contains a percentage of the total number of objects in the dataset. This percentage is taken into account along with the class adjacency index to select a class combination. The combination with the highest ratio is used for reclassification and aggregation of adjoining objects. The *significance ratio* is defined as:

$$SR\,(C_a \mid C_b) = CAI\,(C_a \mid C_b) * \frac{Card(C_a)}{Card(O_M)}$$

The aggregation process is performed repeatedly with other combinations of classes to produce subsequently higher levels of abstraction. Repetition of the procedure on the newly created composite classes may show new relationships previously undetected and create a new, next higher level in the object hierarchy. Combinations that have been used once are not used again. Moreover, combinations where C_b is a network-forming class are excluded from the process since this could create unwanted shortcuts that disturb network topology (see section 4.3).

4.6 Object aggregation factor

Once the combination of classes (C_a,C_b) with the highest *significance ratio* has been identified, the actual aggregation of objects starts. But not *all* adjoining objects of the two classes will be aggregated. The *object aggregation factor (OAF)* is used to determine whether a particular combination of two objects should be aggregated (this is determined for each of the two objects involved). The object aggregation factor is a local measure, in contrast to the class adjacency index which is a global measure. The object aggregation factor for area object o_d is:

- the sum of the lengths of the segments (s_r) that are part of the boundary of object o_d (of class C_a or C_b) and that are also part of the boundary of objects of the complementary class C_c of the selected class combination C_a,C_b

times

- the average area of all objects in the map: $\overline{Area}$

divided by the product of

- the perimeter of object o_d

and

- the area of object o_d

$$OAF\,(o_d) = \frac{\sum_r Length(s_r \mid ADJACENT[o_d, o_e \mid s_r] = 1) \; * \; \overline{Area}}{Perimeter(o_d) * Area(o_d)}$$

with $MEMBER[o_e, C_c]=1$ *(object* o_e *is a member of the complementary class* C_c*)*
and $C_c = C_b$ *if* o_d *is a member of* C_a *else if* o_d *is a member of* C_b *then* $C_c = C_a$

The threshold value for object aggregation will be determined empirically. Once set, this value should work for any categorical area partition since it is based on measures that are not dataset-specific. The threshold value evaluates the degree of inclusion of the object in the composite versus the size of the object. This means that objects that are only marginally connected, and also relatively large objects, have less chance of being aggregated. In the first case this is because the resulting objects should preferably be compact; in the second because it would create composite objects with a very coarse spatial distribution of the constituting classes, in which case the spatial relation between the classes in the composite is not very strong.

4.6.1 Merging the geometry

Once the new composite objects have been created, there will be faces that belong to the same object.

$PART_{22}[f_r, o_{ar}] = 1$
$PART_{22}[f_l, o_{ar}] = 1$

These faces can then be merged:

$Merge(f_r, f_l) = f_m$

- *The boundary of f_m is the union of existing boundaries minus their common segments; the latter are the segments where the original faces are adjacent:*
  $S_{\partial fm} = S_{\partial fr} \cup S_{\partial fl} - S_{\partial fr} \cap S_{\partial fl}$ *where*
  $S_{\partial fr} \cap S_{\partial fl} = \{ s_i \mid ADJACENT[f_r, f_l \mid s_i] = 1 \}$
- *The set of segments inside the new face contains all segments that have this face on both sides, this is:*
  $S_{Inside(fm)} = \{ s_i \mid BOUNDARY[s_i, f_m] = 2 \}$
  The segments in this set carry no semantic information as the study is limited to area objects
- *The set of segments related to the new face is then the union of the set of boundary segments and the set of segments inside the face:*
  $S_{fm} = S_{Inside(fm)} \cup S_{\partial fm}$

4.7 Solving specific scenes by general rules

In order to evade the common generalisation problem of rules that are too specific, it is necessary to develop a method that enables specific situations to be resolved by general rules. With the class adjacency index and a typology of classes we propose two mechanisms to avoid the problems related to using existing manual generalisation rules.

The class typification is used to define rules that are dataset-independent. Instead of the classes of a specific dataset at hand, metaclasses are addressed (section 4.3). This way it is possible to design a set of more comprehensive rules. Dataset-specificness is introduced with the class adjacency index. The class adjacency index is used to define aggregation rules automatically. Once the relationships at class level are clear, specific cases can be resolved quite easily. In general, generalisation rules are preferably not based on particular spatial configurations, but rather on statistical relationships like the class adjacency index.

Figure 24 illustrates the steps involved in the abstraction process (van Smaalen 1996a, van Smaalen 1996b). The bottom row illustrates a geographic dataset at different levels of abstraction. Four levels of an *object hierarchy* (Molenaar 1998) are shown, every level forming a single-valued vector map. The middle row shows the topological structure of the geometric descriptions in the bottom row in the form of adjacency graphs. In the adjacency graph every area object is represented by a node and the adjacency relationships between the objects are represented by edges connecting the nodes. As an addition to the plain adjacency graph as described by Molenaar (Molenaar 1998), a value representing the length of shared boundaries between the objects is attached to the edges.

In the implementation, this value is used to determine the significance of the adjacency relationship. The top row in Figure 24 shows the classes and the relationships between them. Taxonomy relationships are drawn vertically and partonomy relationships horizontally. The partonomy relationships connect the aggregation levels. The taxonomy relationships only play a part within the aggregation levels (see below).

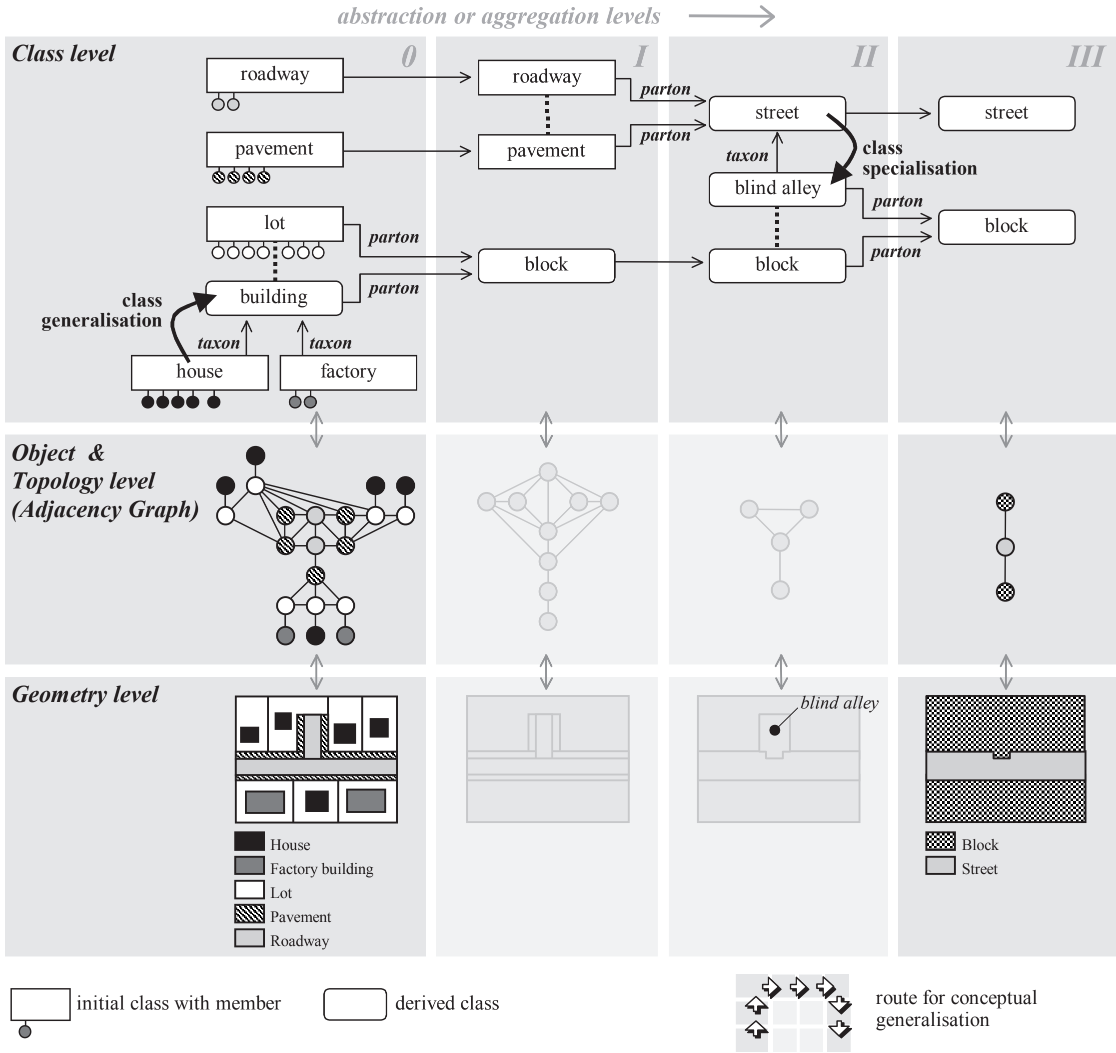


FIGURE 24. MULTI-LAYER MODEL FOR SPATIAL ABSTRACTION

TABLE 1. CLASS ADJACENCY INDICES FOR THE ABSTRACTION LEVELS IN FIGURE 24

Abstraction level 0

Combination	CAI
building-lot	1
roadway-pavement	0,96
lot-building	0,62
pavement-roadway	0,51
pavement-lot	0,49
lot-pavement	0,36
roadway-lot	0,04
lot-roadway	0,02

Abstraction level I

Combination	CAI
roadway-pavement	0,96
pavement-roadway	0,51
pavement-block	0,49
block-pavement	0,36
roadway-block	0,04
block-roadway	0,02

Abstraction level II

Combination	CAI
street-block	0,87
block-street	0,73
blind alley-block	0,7
blind alley-street	0,3
block-blind alley	0,27
street-blind alley	0,13

Abstraction level III

Combination	CAI
street-block	1
block-street	1

The whole abstraction process operates at the topological and class level. The class adjacency index is a measure at class level based on information that can be derived from the topological description. The aggregation process also works at the topological level; all information needed for the object aggregation factor is available in the adjacency graph. The geometrical level only plays a part at the beginning, using it to derive an adjacency graph, and at the end of the abstraction process when adjoining faces that belong to the same object after aggregation, are merged.

In the example of Figure 24 the distinction between houses and factory buildings was not considered relevant. These classes were therefore generalised to ‘building’. The class adjacency indices (Table 1) for abstraction level *0* show that the adjacency relationship between buildings and lots is the strongest of all combinations, with value 1 (the maximum value of the class adjacency index). This combination is therefore selected to be aggregated into ‘block’. The ‘property’ level of Figure 12 was omitted because it is very difficult in practice to determine the inter-object boundaries between adjacent objects of the same composite class[3]. The class adjacency indices for abstraction level *I* show the highest correlation between ‘roadway’ and ‘pavement’. This combination is aggregated into ‘street’ in abstraction level *II*. Level *II* would only have had two classes, and thus only one possibility for aggregation, if the class ‘street’ had not been subdivided into blind alleys and other streets. With more aggregation options, the highest correlations are now between ‘street’ and ‘block’ and ‘block’ and ‘street’. But now an additional rule comes into play which states that objects belonging to the metaclass of network-forming elements can only be aggregated in order of their importance within the network structure. This means that the blind alley is aggregated before other streets, resulting in the level *III* situation. Level *IV* consists of a single object.

Note that the class intension of ‘street’ changes between level *I* and level *II*. In level *I* the class intension includes blind alleys, whereas from level *II* upwards these are not part of the class. Strictly speaking, the name of the class should be changed in such a case, since the name is used to identify the class, but usually this is not done.

In an object-based geographic database each object instance is a member of a class, e.g. a certain object is a ‘house’. Sometimes this classification is not entirely suited for the application. In that case the classes may need to be generalised or specialised. Class generalisation and class specialisation are both types of database enrichment. Database enrichment will be treated more extensively in chapter 5. Database abstraction is often preceded by database enrichment because abstraction, quite contradictorily, often requires the introduction of more detail to begin with. Figure 25 shows an example of class generalisation preceding an aggregation operation: the object subclasses ‘house’ and ‘factory’ belong to the superclass ‘building’. In this case it is the superclass that holds the relationship to the lot, not the subclasses, as a distinction between ‘residential property’ and ‘industrial property’, by aggregating houses and lots, and factory buildings and lots separately, was not considered relevant.

[3] Although this is not particularly the case with objects of the ‘island’ metaclass, such as buildings, but more with objects that are alternately connected.

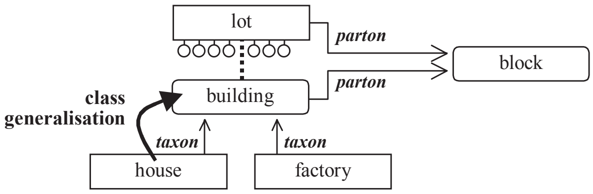


FIGURE 25. CLASS GENERALISATION PRECEDING THE COMBINATION OF CLASSES

It may also be necessary to specialise an existing classification prior to aggregation. Members of the subclasses may subsequently be aggregated with members of another class. It is not always necessary to create a real *object* subclass - with a more detailed class intension than the original class - a mere data class might also suffice (see section 3.4). Class specialisation is often based on spatial properties and relationships. The classification of the objects in a hydrological network, for instance, can be specialised based on the position of the object in the network structure, an example being the identification of blind alleys in Figure 26. Blind alleys may be found by searching for road objects that have only a single connection to another road object.

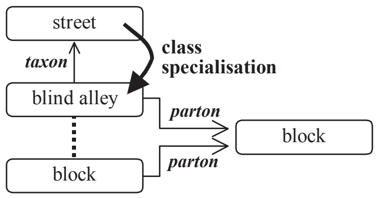


FIGURE 26. SPECIALISATION OF AN OBJECT CLASS TO ENABLE THE CREATION OF A COMPOSITE CLASS

4.8 Steering parameters

Having described the abstraction process, what are the parameters by which the user can influence this process? The full abstraction procedure is depicted in Figure 27. The system can be operated in *auto mode* or *step mode*. Auto mode means that the user sets a stopping criterion first after which the system will aggregate class combinations repeatedly until the criterion is met. A stopping criterion can be a predefined:

- number of aggregation cycles;
- number of objects (the number of objects decreases during the process);
- number of classes (increases);
- mean object area (increases).

In step mode the user evaluates each aggregation step before continuing. The evaluation is twofold. The user can decide to accept the last step and subsequently decide whether a next step is required or that the result resembles the desired end result. Or the user rejects the last step after which the system will revert to the situation that existed before the last step, exclude the current class combination and proceed with the next combination in line.

The last - or rather the first - manner in which the aggregation process can be influenced is by excluding a class from the aggregation process, in order to maintain the class as is.

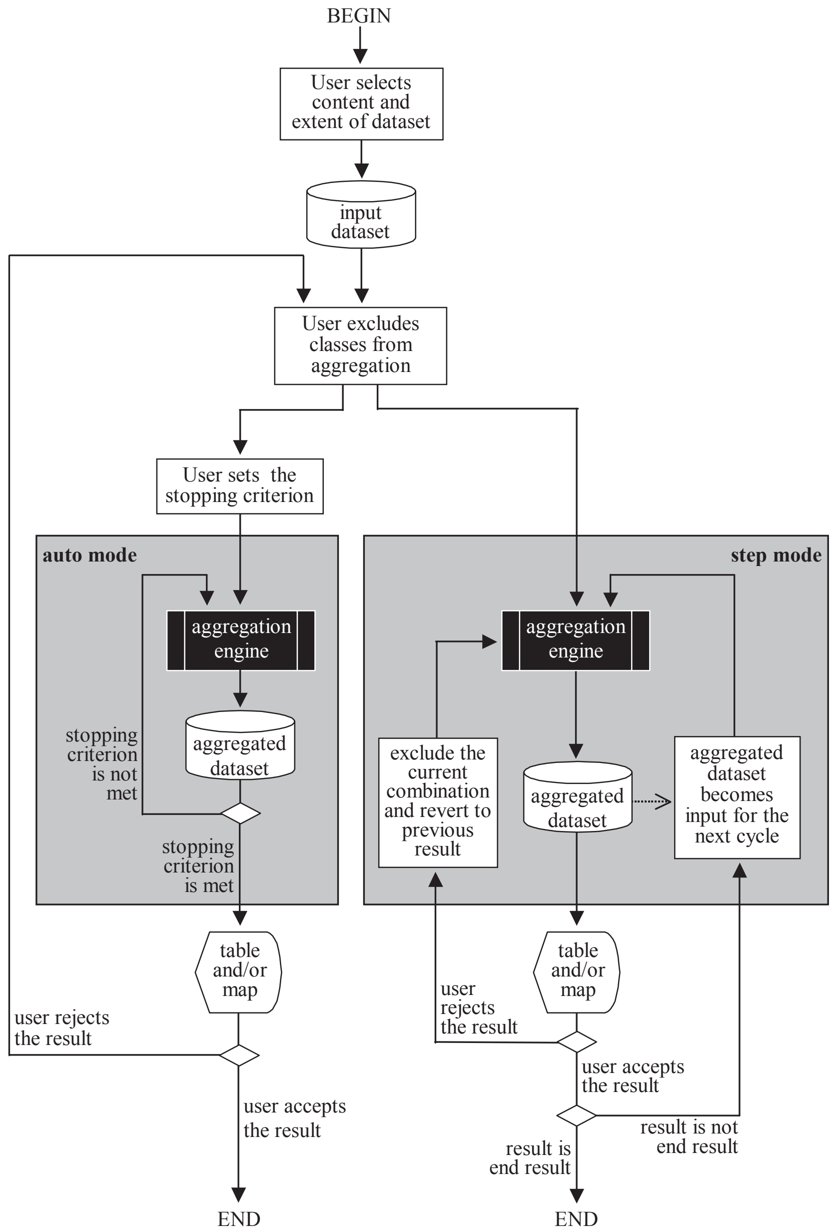


FIGURE 27. THE USER CONTEXT OF THE TWO ABSTRACTION PROCESS MODES

4.9 Summary

In a geometric area partition objects cannot simply be eliminated; aggregation is required. Aggregation requires compatibility measures for the objects. Aggregation of thematically similar objects is only applicable within a limited spatial range. Larger steps are only possible by combining dissimilar objects. Aggregation of dissimilar objects can be based on the class adjacency index (CAI). The class adjacency index is a global measure for the spatial adjacency of thematic classes; this index can be used to identify combinations of classes of which mutually adjacent members may be aggregated. A higher class adjacency index makes aggregation of the members more legitimate. The method based on the class adjacency index allows user-intervention at several points in the process but can also be automated completely.

Chapter 5: Datasets

5.1 Introduction

As shown in the previous chapters, the structure of the base data is very important for operations like generalisation. But it is not only the data model used which determines the possibilities for generalisation. The quality of the actual data is very important as well. This is why great care should be taken while determining the requirements that the datasets should meet.

A dataset is stored using a certain data model, but the data model should not be considered an integral part of the dataset. Datasets can be converted from one data model to another, without adding or losing information. The choice of a data model is determined by the operations one wants to apply to the dataset. A topologically structured data model is best suited for finding neighbouring objects whereas for retrieving objects within a user-defined rectangle, topology does not offer any advantages. An Arc/Info polygon coverage, for example, contains topological relationships between the objects, whereas an ArcView polygon shape file model does not contain any topology. But it is possible to convert an Arc/Info polygon coverage to an ArcView shape file. When converting it back the topology will be recalculated, resulting in an Arc/Info coverage that is identical to the initial one.

Dataset requirements for applying the aggregation method described in this study:

- The initial dataset needs to form a complete geometric and thematic partition, i.e. a single-valued vector map.
- The classification of the thematic partition should be sufficiently detailed. Strategically chosen ordinal class attribute values can play an important role in the abstraction process, in particular within network-forming classes. Database enrichment operations may be required to attain these classifications. The classification may either be derived automatically based on topological (e.g. blind alley) or geometric (e.g. narrow road) properties or can be assigned manually (e.g. road closed for motorised vehicles).
- The geometric primitives should be delimited accurately and be small enough, especially in network-forming classes, to leave sufficient flexibility in composing the desired generalised network topology. Figure 28 shows a situation where the choice of geometric primitives is arbitrary. A better solution is given in Figure 29. The transverse delineation of network segments needs to be consistent. This is most easily done by defining a separate primitive of the type ‘junction’ at every point where three or more segments connect.
- Area objects should not be described by their boundaries alone, but by faces.
- The dataset should be topologically correct. Object boundaries should be fully closed.

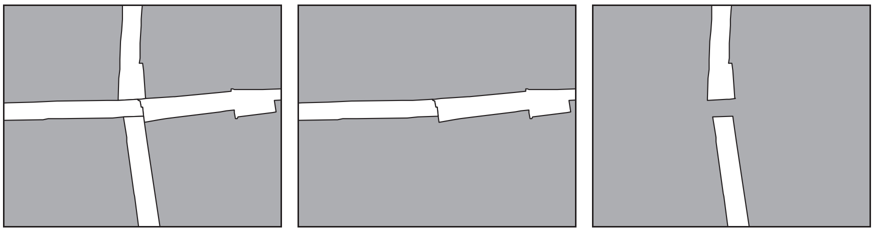


FIGURE 28. ARBITRARILY DELINEATED ROAD SEGMENTS CAUSE ELIMINATION PROBLEMS

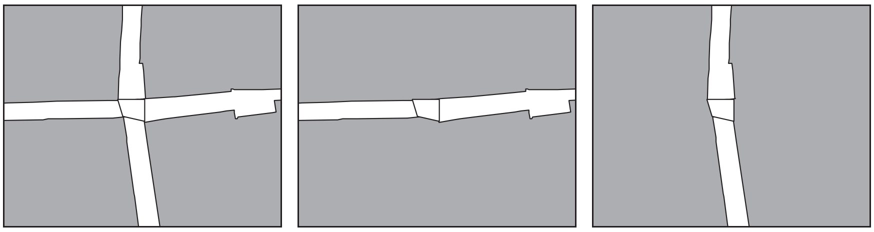


FIGURE 29. TREATING JUNCTIONS AS SEPARATE SEGMENTS ENABLES ELIMINATION OF ANY OF THE SEGMENTS

5.2 Datasets

Two topographic datasets are used in this study. The first one is the large-scale base map (Grootschalige Basiskaart or GBKN) in a prototype object form created by the Dutch Cadastre and Public Registers Agency (Kadaster). The second one is the TOP10vector by the Dutch Topographic Survey (TDN). Both datasets contain topographic information such as roads, waterways and buildings. The datasets are explained further in the following paragraphs. The Dutch town of Zevenaar and its immediate surroundings are used as a test area because of the availability of a prototype object-GBKN dataset for the area. The region is considered to offer a representative example of both types of datasets.

5.2.1 GBKN

Work on the GBKN started in 1975 with an initial government grant and was further financed and developed by an alliance of organisations: utilities, municipalities and land registry. Maintenance and distribution are in the hands of regional partnerships. The Dutch Cadastre and Public Registers Agency produces the dataset for a number of the regional partnerships. The *Landelijk Samenwerkingverband GBKN* (LSV-GBKN) is responsible for the co-ordination at a national level. The last unmapped area in the GBKN was completed in 2001 and GBKN data are now available for any location in the Netherlands.

The average spatial precision of GBKN data is 28 cm in built-up and 56 cm in rural areas. The level of detail is comparable with 1:500 to 1:2000 scale maps. The dataset is intended primarily for management and design purposes. The GBKN is generally considered a suitable geometric basis for additional information of the own organisation. The digital information was originally intended for drawing maps. As a result, the original GBKN only uses lines (roadside, building perimeter) and points to represent terrain features. Road boundaries are labelled ‘roadside’; the thematic information is attached to the boundary of the features as the dataset does not contain polygons. Although a human person can determine visually on which side of the line the road is situated, computers cannot deal with such incomplete information. Moreover, the data in the original GBKN are not topologically structured. Roadsides show gaps at entrances to buildings (Figure 30) and the transverse delineation of road segments is arbitrary. Overall, no objects are distinguished; the dataset only contains geometric primitives.

In the coming years the GBKN datasets will slowly move towards an object-based structure. The building layer will be converted first. The object-GBKN introduced in the next section is a test dataset developed for evaluation purposes.

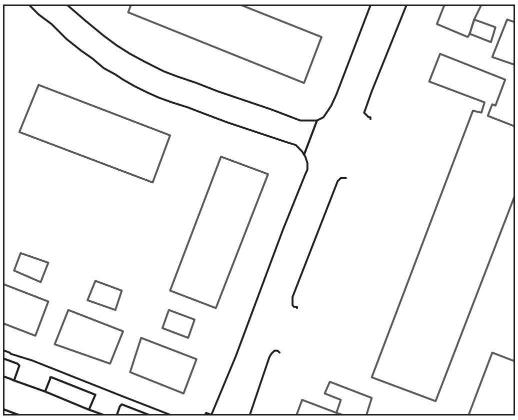


FIGURE 30 . EXAMPLES OF GAPS AT ENTRANCES AS THEY APPEAR IN THE GBKN DATASET

5.2.2 Object-GBKN

In this study a dataset called the object-GBKN is used. This dataset was created as a prototype by the Dutch Cadastre and Public Registers Agency (Kadaster). The thematic structure of this dataset is based on NEN3610 (NNI 1995, NNI 2001), an object-based standard. This, and the name object-GBKN, suggest that an object-based approach was used, but this is not really the case. As we can see in Figure 31, the division into road segments still seems rather arbitrary. This can cause problems during a generalisation process, as is shown in Figure 28 and Figure 29. There is no distinction between connections and junctions as defined in NEN3610.

Rather than object-GBKN, the dataset should preferably be called area-GBKN, since the main difference with the original GBKN is that it contains not just the boundaries of the terrain features, but also the features themselves, as areas.

One difficulty with network-like structures is the transverse delineation of objects. If the network segments are represented by lines, this is generally taken care of automatically, because nodes are or can be inserted at intersection points. But when dealing with area objects we should have clear rules for handling intersections to avoid problems like the one shown in Figure 28, and peculiarities like the examples shown in Figure 31, where the road elements in the centre of the picture have an atypical shape. Currently, there are no such rules, whereas the rules for longitudinal delineation are generally very detailed.

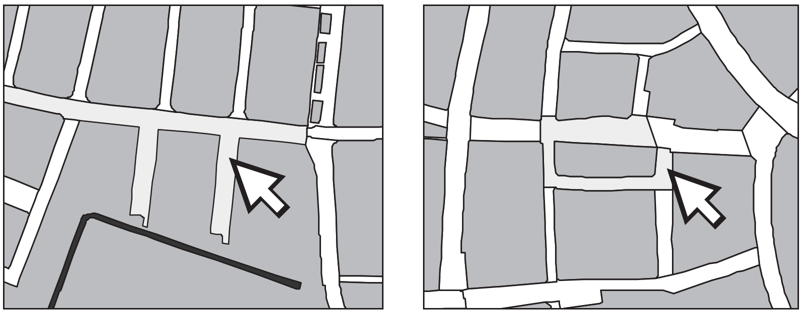


FIGURE 31. OBJECT-GBKN ROAD SEGMENTS APPEAR TO HAVE BEEN DELINEATED RATHER ARBITRARILY

The object-GBKN is an experimental dataset. It is topologically sound and consists of four layers:

- Division objects
- Furnishing objects
- Providing element*s (area)*
- Providing element*s (linear)*.

The linear elements are discarded since the application works on area objects only. The other three layers are described below.

The *division objects* in the object-GBKN constitute a single-valued vector map (Figure 32). Five feature classes are distinguished: road, railway, water, terrain and crossing. Roads, railways and water break up the spatial extent of the dataset. The remaining areas between the infrastructural objects are classified as 'terrain'. The use of the object type 'crossing' is not in accordance with NEN3610. 'Crossing' is used where two *distinct types* of division objects cross or two of the *same type* cross *at different levels.* NEN3610 does not identify an object type 'crossing', but it distinguishes between connections and junctions by means of an attribute. The crossings in the object-GBKN use the same code as the junctions in NEN3610. It seems that this was interpreted wrongly, because unlike the object-GBKN's crossings, junctions also appear where two objects of the same type connect or cross *at the same level*. This is not dealt with in the object-GBKN. As a result, transverse delineation of infrastructural objects becomes arbitrary. This applies to all types of infrastructure: roads, railways and waterways. *Furnishing objects* exist within the division objects. They belong to the classes 'building' and 'construction work' (bridge, tunnel, dam etc.). Main buildings and outbuildings or annexes are discerned. Figure 33 shows a map with the furnishing objects added to the layer of the division objects. Of the furnishing objects only the buildings are used. *Providing elements (area)* are objects that create a more detailed partitioning of the objects mentioned before (Figure 34). This results - at least in the case of road objects, which are furnished 100% - in component and composite objects. All objects in the object-GBKN have a unique object identifier.

FIGURE 32. GBKN DIVISION OBJECTS

FIGURE 33. THE FURNISHING OBJECTS ADDED TO THE MAP IN FIGURE 32

FIGURE 34. PROVIDING ELEMENTS ADDED TO THE MAP IN FIGURE 33

5.2.3 TOP10vector

TOP10vector is a dataset of the Dutch Topographic Survey (TDN) originally used to create 1:10,000 topographic maps for the military. Some of the classifications are still based on military requirements. The entire nation is mapped with a spatial precision of around 2 metres. TOP10vector is maintained and distributed by TDN. Recently TDN and the Dutch Cadastre and Public Registers Agency signed an agreement bringing the TDN under the responsibility of the Cadastre and Public Registers Agency. This more or less puts the GBKN and TOP10vector in the same hands, which will stimulate efforts to create both datasets on the basis of a single survey.

TOP10vector data consist of four layers. The first is a geometric partition of area features including water, roads, parcels, but not buildings, which are in a separate layer. Furthermore, there is a layer of point symbols (church symbols, individual trees etc.) and a layer containing linear objects (narrow waters, hedges etc.). The first layer contains objects that are based on land cover such as forests (deciduous, coniferous, mixed), grassland, arable land and water, but also infrastructural objects with functional classifications (motorway, street, bicycle path). The objects in this layer are therefore rather ambiguous, as function and appearance are mixed up. Moreover, hydrology cannot be used effectively as an infrastructure as it is separated into two layers; narrow water features are represented as lines, broad waters as areas. Roads are delineated in a less arbitrary way than in the object-GBKN; at crossings and connections the main road is generally uninterrupted. Although this may be true for the major roads, in residential quarters street delineation *is* rather arbitrary. The classification is also insufficiently discriminating in these areas.

The Dutch topographic survey, producer of TOP10vector, has been commissioned to develop the Dutch national 10,000 topographic dataset (TOP10NL, see section 5.4.2). TOP10vector will be developed towards an object-based model to serve as a basis for this new dataset.

5.3 Database enrichment

Classifications can either be assigned to the dataset by hand during construction, or automated database enrichment (Mackaness et al. 1997) processes can be used to create customised classifications whenever needed. Only certain types of classification can be derived automatically based on geometric or topological properties, a functional classification of road segments, for example. These derived classifications are not ‘the real thing’, however; errors can occur and sometimes additional information such as ‘one way traffic’ is needed. In other words, automated database enrichment is not the answer to everything.

In chapter 4, we discerned three metaclasses: ‘islands’, ‘network-forming’ objects and normal area objects. The first two types in particular offer opportunities for database enrichment.

Classes with many objects forming ‘islands’ within other objects can be used in a database enrichment process to classify areas based on the density of objects in the class, e.g. delineating built-up areas based on proximity of buildings (Figure 35) so that buildings in urban and rural areas can be distinguished and treated differently in the generalisation process.

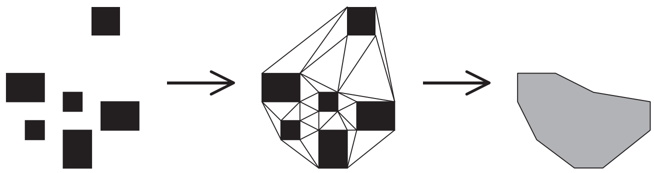


FIGURE 35. PROXIMITY OF BUILDINGS DENSITY FOR DELINEATION OF BUILT-UP AREAS

Network-forming classes allow elements to be classified based on their topological positions; i.e. based on their importance within the network structure. A subdivision into directed (e.g. stream networks) and non-directed (e.g. most road networks) networks is a valid basis for distinct classification algorithms. Several approaches have been presented for deriving classifications for roads (Peng 1995, Richardson and Thomson 1996), as well as for hydrological networks (Martinez Casasnovas 1994, Richardson 1993, Figure 36).

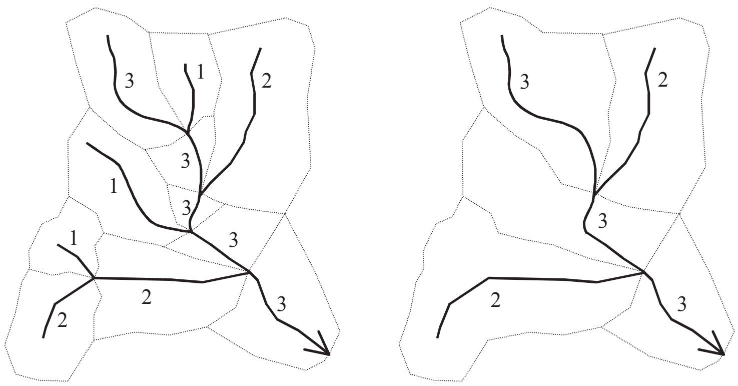


FIGURE 36. A CLASSIFIED HYDROLOGICAL NETWORK AND THE CORRESPONDING WATERSHEDS AT DIFFERENT LEVELS OF ABSTRACTION (HORTON CLASSIFICATION)

Besides thematic, e.g. creating subclasses and superclasses, data base enrichment can also be purely geometric. For example, during geometric database enrichment of a transportation network consisting of area objects, segments are added to subdivide existing objects (Figure 37).

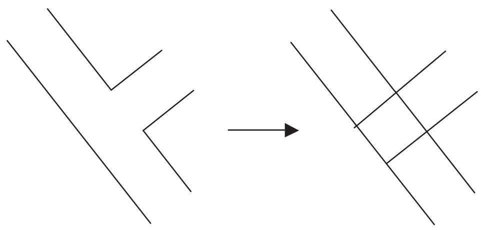


FIGURE 37. GEOMETRIC DATABASE ENRICHMENT

As generally the thematic database enrichment operations already exist and prove to work, it is assumed for this study that the attributes on which selection within a network structure is based, already exist. If not, they are added manually. In other words, the classification of the data is supposed to be detailed enough to serve as the basis for the generalisation procedure. The same applies to the spatial description, although operations for automated geometric database enrichment are rarer.

5.4 Recent developments

5.4.1 NGII

A debate has been going on for some time in the Netherlands about whether a number of 'base datasets' fall under the responsibility of the central government and whether this information should be made publicly available at little or no cost. In 1995 the Ravi described the Dutch National Geo-Information Infrastructure (Ravi 1995). The NGII comprises the aspects policy, datasets, technology, standards and knowledge to achieve maximum social and economic effect. Such a NGII depends highly on the availability of nationwide, standardised, mutually linkable base datasets. By the year 2002 availability of the base datasets was gradually considered a responsibility of the central government, although the actual realisation can be in the hands of local governments or private companies (Tweede Kamer 2000). Datasets regarded to be under this regime include: a topographic dataset (TOP10NL, level of detail comparable to 1:10,000 maps), a building and address registration and possibly the large-scale base map (GBKN, 1:1000). The address forms the link between the spatial information and non-spatial person and company registrations. The presence of no less than two topographic datasets in this selection seems rather odd. Object information is preferably maintained and made available by the actual owner; water boards keep hydrographical information up-to-date, municipalities maintain the address and building information etc. This is in contradiction with the idea of a single organisation collecting and maintaining information for a topographic map like the TOP10NL, since a topographic map is basically a compilation of information that is already maintained by other organisations.

5.4.2 TOP10NL

The thematic structure of the TOP10NL has been defined (Ravi 1998, Ravi 2002) but the dataset has not yet been realised. The Dutch topographic survey (TDN) recently acquired the assignment to produce the dataset. A development path has been defined to adapt the topographic survey's own TOP10vector to the specifications of this new dataset. TDN proposed a 5-year migration path to move from the current TOP10vector to the TOP10NL specifications.

5.4.3 NEN3610

The specifications of the TOP10NL are based on NEN3610 (NNI 1995, NNI 2001). NEN3610 was developed for the transfer of spatial information between different target groups[4], although the model is also used more and more to structure the internal information needs of organisations. NEN3610 is not a technical exchange format; it

[4] A group of similar organisations and/or institutions

describes the contents of the information exchanged. In others words: *what* is transferred, not *how* it is transferred. The model mainly comprises a classification, little is said about the spatial representation of the objects. It defines the object types that can be used and the associated attributes and attribute domains. As it is intended for exchanging information between sectors it is a generic model. It is up to specific sectors to define additional attributes and/or attributes values as needed for transfers within each sector.

The most important categorising attributes in NEN3610 are:

- type (connection/junction, main building/outbuilding);
- function.

For network-forming feature classes like roads and waterways, NEN3610 distinguishes connections and junctions. This means that a junction is required at every point where segments leave in three or more directions. That way it is possible to connect any combination of segments to create a route, and to eliminate any segment for generalisation purposes. But the lack of guidance in NEN3610 regarding the spatial organisation of the data makes that much is left to the interpretation of the topographer, leading to problems as shown in Figure 31.

NEN3610 does not provide a classification of network-forming features based on topological positions within the network structure. In the case of roads the functional groups are categories like ‘pedestrians’, ‘slow motorised traffic’, ‘fast traffic’ etc.; there is no classification based on importance such as main, secondary and tertiary roads. The *function* attribute in NEN3610 is rather ambiguous anyhow. The values show a mixture of function and manifestation/topography. On the one hand we see clearly functional values like ‘museum’, ‘wholesale trade’ and ‘school’ whereas there are also values like ‘forest’ and ‘water’.

A third important attribute in NEN3610 is the object identifier. Identifiers need to be unique within a dataset, and preferably be used across datasets. Object identifiers offer an important mechanism for integrating data from different sources. The address, for instance, is the link between people and buildings. If the address uses an identification such as the postal code and house number, the link can be made easier and more securely than when using the complete address consisting of municipality, town, street and number. This is due to the fact that the chance of any of these elements being spelled incorrectly or just differently in two related administrations is considerable. Take, for example, the two Dutch names for the city of the Hague: Den Haag and ‘s-Gravenhage. The drawback of using meaningless identifiers is that implicit verification by people accessing the information is impossible. Identifiers are used to relate objects, for identification by a system, not for humans to access the object. People use other characteristics of the object for identification. The identifier is strictly not a characteristic of the object.

If NEN3610 is implemented correctly, the spatial description offers sufficient detail for generalisation purposes. The thematic classes are not very suitable, however, mainly because of a lack of ordinal attributes.

5.4.4 Development of the existing datasets

At the same time the organisations behind the existent topographic datasets (TDN and LSV-GBKN) are busy restructuring their products. Anticipating decisions concerning the NGII they are already making their products more compliant with the ideas behind it.

Issues are whether the large-scale base dataset (GBKN) should be line- or area-based, and object-structured or not. However, some municipalities already work with an object-structured GBKN. Object-structuring is also one the prime issues for a new TOP10NL dataset. The proposed TOP10NL is based on a model with component and composite objects. It differentiates between entire roads and road segments, an important issue in the light of generalisation (see Figure 28, Figure 29). Further issues under consideration for TOP10NL are a shift from map sheets to a contiguous database and meta-information at object level.

The results of these discussions are very important as they will determine the structure of the Dutch topographic data for the next ten or fifteen years, a period in which we will undoubtedly see an increase in the interoperable use of geo-information. Which makes it even more important to ensure that the existing data become linkable, both thematically and spatially. In this respect, object identification and standardisation of spatial models will become even more important issues.

5.5 Pre-processing the data

5.5.1 TOP10vector

From the four TOP10vector layers only the two area layers (areas and buildings) are used, because the generalisation application works on area data only, and combined into a single layer (Figure 38).

In order to reduce the number of possible class combinations, the TOP10vector classification hierarchies were employed to generalise the classification somewhat. 'Built-up area' was incorporated into the class 'building'. 'Passage' and 'promenade' were combined into 'pedestrians'. Deciduous, coniferous and mixed forest were generalised to 'forest'. Arable land and grassland were combined. A few classes with a very small extension ('cemetery', 'sand') were incorporated into the class 'other land use'. For the roads an included classification of 6 distinct types was used. This resulted in the 14 distinct classes in Table 2 instead of the original 27.

5.5.2 GBKN

As for the object-GBKN, a single-valued vector map was created from the division objects in combination with the buildings from the furnishing objects layer (see Figure 33). The providing elements were not used. Although at first sight the geometry of this layer appears useful for generalisation purposes, the attributes do not offer much extra value. It does not provide additional transverse delineations, nor does it differentiate clearly between roadway and pavement or shoulder (Figure 40, Figure 41). The objects in the resulting dataset belong to 5 classes: 'Road', 'Building', 'Water', 'Railway' and 'Terrain' (Table 3).

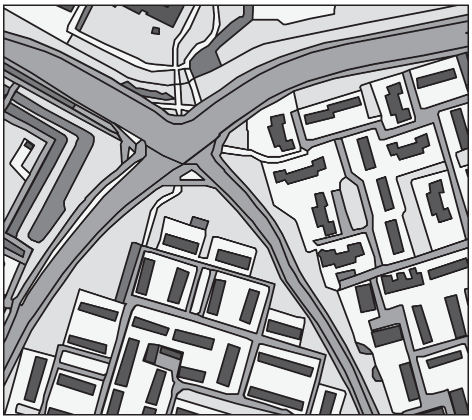


Figure 38. TOP10vector area layer and buildings combined (s.v.v.m.)

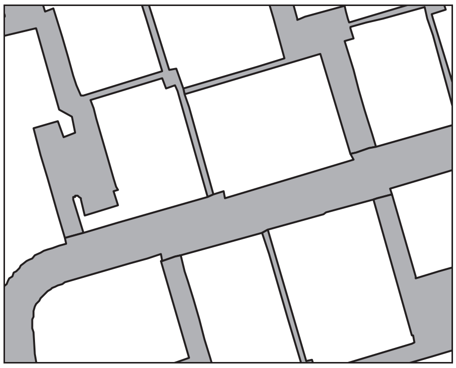


Figure 39. Division objects

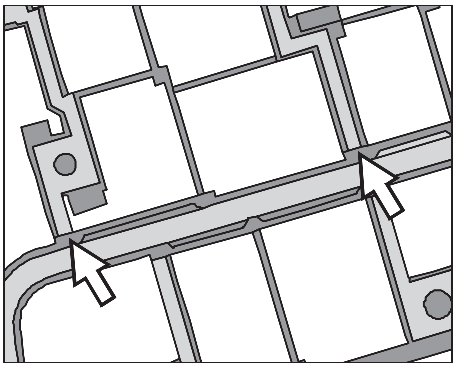


FIGURE 40. TYPE OF PAVING (ATTRIBUTE 2) IS NOT SUFFICIENT TO DISTINGUISH ROADWAYS

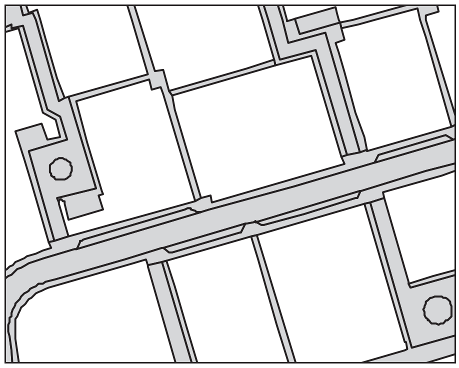


FIGURE 41. NEITHER IS ATTRIBUTE 1 (ROAD, SHOULDER)

TABLE 2. THE CLASSES IN THE TOP10VECTOR DATASET

Class
Building
Greenhouse
Motorway
Main Road
Secondary Road
Unpaved Road
Pedestrians
Street
Bicycle Track
Parking Space
Forest
Grass/Arable
Other land use
Water

TABLE 3. THE CLASSES IN THE GBKN DATASET

Class
Road
Building
Water
Railway
Terrain

5.6 Summary

The aggregation method described in the previous chapter is based on a data model with stored topology. The datasets should therefore be topologically correct. They must also consist of a single geometric area partition. The object-GBKN does not meet all further requirements, especially regarding the transverse delineation of road segments. The classification is also rather limited. The classification structure of the TOP10vector dataset is better suited, containing a functional classification of roads, for example. The TOP10vector is currently in the process of being restructured. This is a moment at which the requirements regarding automated generalisation should be taken into account, so that the future datasets permit automated generalisation procedures to be performed on them. The successor of the TOP10vector dataset, TOP10NL, is based on NEN 3610, an exchange standard for geographic information. Although NEN 3610 shows geometric features that are valuable for generalisation and aggregation purposes, such as consistent transverse delineation of network elements, the thematic classification of objects is not always very consistent nor is it detailed enough.

Chapter 6: Implementation

6.1 Implementing the topological model

6.1.1 Software

The implementation of the topological model described in section 3.6.2 is realised using Arc/Info to pre-process the data. Arc/Info employs a number of relational tables to store the information of an area partition - polygon coverage in Arc/Info terminology. Two of these tables are relevant for this application. One contains the faces (called *polygons* in Arc/Info) and related information, the other one the border segments (*arcs* in Arc/Info).

Besides the identifier and the length of the border segment, as well as its begin and end node, the *segment table* (Table 4) contains the identifiers of the faces on both sides of the segment. These references play a very important role in the implementation. The references are labelled *lobject#* (since every face represents an object) for the face on the left side of the segment, and *robject#*, for the one on the right. What is left and right in reality is not important, however; it is only important that the neighbour relationships of the faces are recorded. All face, segment and node identifiers are automatically assigned consecutive numbers and therefore bear no relation to any identifications used for the real-world features the dataset describes.

The data model allows objects to consist of more than one geometric primitive, but in the initial dataset objects do not comprise more than one primitive, i.e. every face is an object. Composite objects comprise more faces, but in this research they can, by definition, not overlap.

The object table (Table 5) contains information about the faces such as the identifier (object#), the area and perimeter and, very importantly, the feature class. The identifier is the number that appears under lobject# and robject# of the segment table. The combination of segment and object table is effectively an implementation of the adjacency graph as described in chapter 4.

TABLE 4. SEGMENT TABLE (# EQUALS NO.)

segment#	*fnode#*	*tnode#*	*lobject#*	*robject#*	*length*
1	14	26	3	24	8,974

TABLE 5. OBJECT TABLE (# EQUALS NO.)

object#	*area*	*perimeter*	*class*
3	234,564	122,355	5

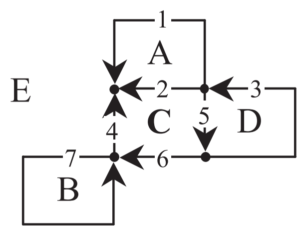
Both the segment table and the object table are transferred from Arc/Info to an Oracle database where all generalisation operations are carried out. This results in new segment and object tables for every abstraction level. The generalisation procedure is described in section 0. In the Oracle database, the tables are queried using SQL. SQL was chosen to keep the implementation as software-independent as possible. SQL offers no spatial operators, which means that the required spatial relationships need to be part of the data model. Once the objects of the new abstraction level have been created, Arc/Info is used to merge the geometry. ArcView is used to visualise the results.

6.1.2 Searching for neighbours in a topological data model

Searching for neighbours in a topological model like the formal data structure (FDS) requires the querying of the segment table that contains the relationships between the objects on the left side and the right side of the segment. In this way, we can find all objects connected to object C (Figure 42) by querying the segment table that contains references to the objects left and right of the segment. Because of the directionality of the segments we have to perform two queries:

1. select all segments that have a left reference to C (in this case 2) and return the right reference (here A), and;
2. select all segments that have a right reference to C (4, 5 and 6) and return the left reference (none, D and none respectively).

In Figure 42 we can see that only objects with a shared segment are considered neighbours, and not objects that only share a node, like B and C or A and D. Including objects that share a node in the neighbourhood would require additional queries (from object to segment, to node, to segment, to object respectively).



segment	left ref.	right ref.
1	A	E
2	C	A
3	D	E
4	E	C
5	D	C
6	E	C
7	B	E

FIGURE 42. A TOPOLOGICALLY STRUCTURED AREA DESCRIPTION

The segment table only contains the identifiers of the neighbouring objects, but the *properties* of the objects are in the object table. To find neighbours with certain characteristics in common we therefore have to perform a cross-table query. For example, if we want to search for segments where the class of the object to their left is identical to the class of the object to their right, a cross-table query involving the segment table and the object table (twice!) has to be performed (Figure 43). We see that segment 3 fits the requirement. While current GIS's, such as ArcView, usually require separate queries for spatial and thematic properties, the use of SQL and a topological model allows topological relationships and thematic properties to be queried simultaneously.

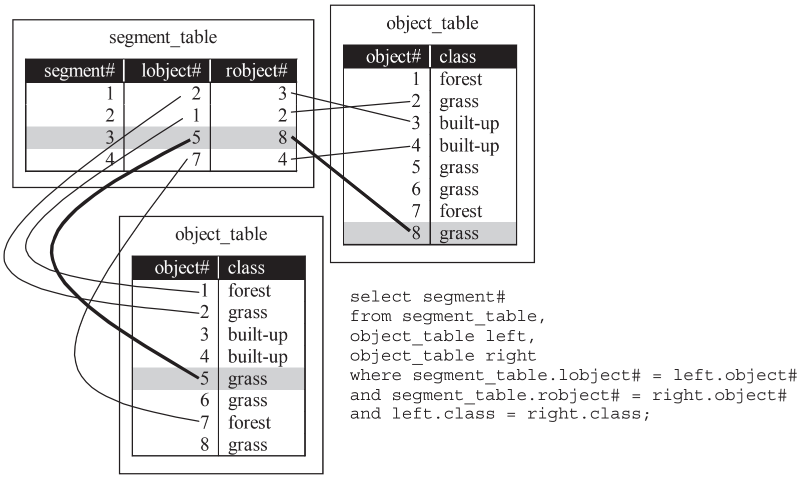


FIGURE 43. NEIGHBOUR RELATIONSHIPS BETWEEN OBJECTS SHARING A CERTAIN CHARACTERISTIC

The number of queries or the complexity of the queries required to find neighbours could be reduced by creating an array containing references to both the left- and the right-hand object. With SQL's 'IN' predicate it is then possible to find connected segments and neighbouring objects, respectively, in a more straightforward manner. Unfortunately, the version of Oracle used did not support the use of arrays. Although an array could be simulated by creating a string like '21,38', this was not done because of the additional programming required.

For the last couple of years there has been an interest in developing an enhanced version of the SQL data manipulation language that would allow the querying of spatial relationships. Although the implementation described in this study uses SQL to query topological relationships, it uses standard SQL without any real spatial functionality. Topological pre-processing of the data was done using Arc/Info, after which the Arc/Info tables containing the topological relationships were transferred to an Oracle database. The abstraction process was performed using SQL queries embedded in the PL/SQL programming language.

In spite of being the most widely-spread data manipulation language, SQL is rather user-unfriendly (Frank and Mark 1991). SQL queries quickly become very complicated, especially when several interrelated tables are involved, with the risk of ending up with the wrong answer unawares. SQL is therefore most suited as a programming environment; it is too complicated for end users to use as an interface.

6.2 Procedural and set-oriented data manipulation

SQL is a set-oriented data manipulation language. Some queries cannot be solved with set-at-a-time, or set-oriented, data manipulation alone, but require procedural or record-at-a-time data manipulation (Elmasri and Navathe 1989). In situations where repetitive queries have to be made, especially when the number of repetitions is not known beforehand, normal set-oriented queries do not suffice. SQL is highly suitable for selecting all forest objects in an entire dataset that are next to grassland, but not too suitable for finding all forest plots belonging to a contiguous forested area starting from one of the plots. The reason for this is that repetition is not a part of the SQL features. In order to overcome this problem, Oracle offers a procedural programming language, called PL/SQL, with its database management system. PL/SQL is an Oracle product and not a part of the official SQL89 specification. PL/SQL is a programming language that, like other traditional programming languages, enables procedural solutions like:

- control statements such as IF … THEN … ELSE, EXIT and GOTO;
- repetition statements such as FOR loops and WHILE loops;
- assignment statements such as X := Y + Z and;
- cursors, that enable query selections to be processed row by row.

SQL statements can be embedded in the PL/SQL program.

SQL3, released in 1999 as the successor of SQL89, offers additional functionality within the official SQL specification. SQL3 incorporates procedural functionality similar to PL/SQL. Although the SQL3 standard does not yet offer spatial data types and operators, it allows the creation of user-defined, complex data types as well as user-defined functions and methods. This enables third parties to develop spatial information systems based on the SQL language. Since SQL is already widely spread in the non-spatial domain, SQL will most likely become a strong contender in the spatial domain when third party software companies develop systems employing the possibilities of user-defined data types and functions in SQL3. SQL3 was not used for the implementation in this study.

An exception to the rule that repetitive queries are not possible in a set-oriented data manipulation language is the hierarchical query. Oracle offers this feature in their version of the SQL language, SQL+. The hierarchical query is also part of the SQL3 specification. The hierarchical query offers unrestricted repetition in a table containing relationships of the parent-child type (Table 6). This means that the data should not contain any loops like in Table 7 where a destination value (in this case 2) is used earlier as a source value. In the implementation the hierarchical query is used to retrieve the consecutive component classes of a composite class. It can also be used to query the relationship between component and composite objects.

TABLE 6. EXAMPLE OF A QUERY TO SELECT THE NODES *DOWNSTREAM* OF NODE 2

```
SELECT  destination  FROM tree
CONNECT BY source = PRIOR destination
START WITH source =  2 ;
```

Table *tree*

SOURCE	DESTINATION
2	1
3	1
1	7
8	7
9	2
10	3
2	3

Query result

DESTINATION
1
3
1[5]
7

[5] Node 1 appears twice in the result because it is found twice, once through its link with node 3 and again through its link with node 2.

TABLE 7. TABLE WITH LOOP

SOURCE	DESTINATION
2	1
7	2
1	7
8	7
9	2
10	3
2	3

The loop in this table reads: 2 → 1 → 7 → 2

6.3 Procedure for functional aggregation

For the implementation we presuppose that every face in the original input data is an object, and that objects are always contiguous. The functional aggregation procedure we applied consists of two parts:

- Determining class adjacency indices for every class combination (section 6.3.1);
- Aggregating objects, i.e. the actual abstraction process (section 0).

6.3.1 Determining the class adjacency indices (global)

In order to determine how classes are spatially related we introduced the class adjacency index. In the implementation the class adjacency index is determined using the segments separating the objects.

Step 1: By joining the segment table and the object table using the numeric identifiers of the objects, a table (Table 8) is created containing two rows for every border segment; one row contains its length and the class of the object to its left, and the other row its length and the class of the object to its right. By grouping the information in the table we just created by class, we get a table (Table 9) containing the total length of borders for each class.

TABLE 8. CLASSES OF THE OBJECTS ON BOTH SIDES OF EVERY SEGMENT AND THE LENGTH OF THIS SEGMENT

class	*length*
segment1.classleft	segment1.length
segment1.classright	segment1.length
segment2.classleft	segment2.length
segment2.classright	segment2.length
segment3.classleft	segment3.length

TABLE 9. TOTAL BORDER LENGTH PER CLASS

class	*classlength*
class1	class1.length
class2	class2.length

Step 2: Based on the segment and object tables we create a table (Table 10) that contains the length of the segment and the classes of the objects on the left and the right. This

table contains two rows for every border segment; in the first row the class of the left object appears in the first column (class a), in the second row that of the one to the right. The length values in Table 10 are summed by grouping the values for each different combination of classes to create Table 11. Combining the information from Table 11 and Table 9, we can determine the *class adjacency index* (section 4.5.1) for every combination of classes (Table 12).

TABLE 10. LEFT AND RIGHT CLASS AND SEGMENT LENGTH FOR EACH SEGMENT IN THE DATASET

class a	*class b*	*length*
class1	class2	12.52
class1	class2	4.54
class2	class1	2.56
class2	class8	23.37
class9	class1	34.50

TABLE 11. THE INFORMATION OF TABLE 10 GROUPED BY COMBINATIONS OF CLASSES

class a	*class b*	*sumlength*
class1	class2	312.56
class1	class3	634.50
class2	class1	312.56
class2	class3	123.30
class3	class1	634.51

TABLE 12. THE CLASS ADJACENCY INDEX FOR EVERY COMBINATION OF CLASSES

class a	*class b*	*class adjacency index*
class1	class2	sumlength (class1-class2) / class1.length
class1	class3	sumlength (class1-class3) / class1.length
class2	class1	sumlength (class1-class2) / class2.length

From an efficiency point of view, it is important that only class combinations that appear in a significant number of cases are reclassified. Each class contains a percentage of the total number of objects in the dataset. This percentage is taken into account along with the class adjacency index to select a class combination for reclassification (Table 13). The combination with the highest ratio is used for reclassification and aggregation of adjoining member objects. The aggregation process is performed repeatedly with other combinations of classes to produce subsequently higher levels of abstraction. Combinations that have been used once are not used again. Moreover, combinations where class 'b' is a network-forming class are excluded from the process since these could lead to unwanted shortcuts that disturb network topology (see section 4.3).

TABLE 13. THE SIGNIFICANCE RATIO

class a	*class b*	*significance ratio*
class1	class2	class adjacency index (class1-class2) * (no. of obj. in class1 / total no. of obj.)
class1	class3	class adjacency index (class1-class3) * (no. of obj. in class1 / total no. of obj.)
class2	class1	class adjacency index (class1-class2) * (no. of obj. in class2 / total no. of obj.)

6.3.2 Object aggregation (local)

Once the class combination with the highest ratio has been identified, the actual object aggregation process is straightforward. Once more the segment and object tables are employed to find objects of class 'a' that have a neighbour of class 'b'. 'A' and 'b' are the two classes identified by the highest ratio. Both the selected objects of class 'a' and the neighbours of class 'b' are selected to be aggregated.

It is possible to evaluate the selected object combinations with an *object aggregation factor* (section 4.6). For the results presented here, the object aggregation factor threshold value was set to the lowest possible value, i.e. all objects were aggregated. The object aggregation factor *was* utilised for the second step in the aggregation process. This step involves objects that are adjacent to a composite object incorporating the class of the object. If they pass the evaluation they are incorporated into the composite (Figure 44, Figure 45). The threshold value was determined empirically and proved not very critical.

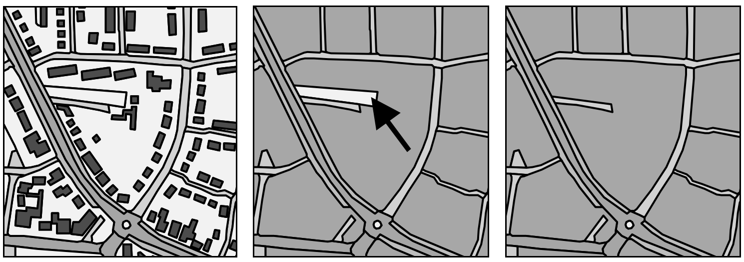


FIGURE 44. OBJECT ADJACENT TO A COMPOSITE OBJECT THAT INCORPORATES THE CLASS OF THE OBJECT

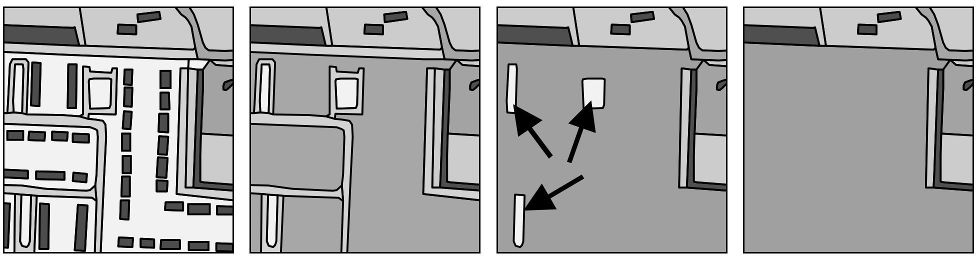


FIGURE 45. OBJECT ADJACENT TO AN OBJECT THAT INCORPORATES THE CLASS, DATING BACK TO A PREVIOUS AGGREGATION CYCLE

6.4 Storing the results

The results of each aggregation step can be retained in the form of full GIS datasets. The objects of the subsequent aggregation levels are connected through *part-of* relationships (Table 14), effectively building a multi-scale dataset that allows navigation through the levels of abstraction. This also makes it possible to store the aggregation results solely in the form of Table 14, containing the links between component and composite objects, and to create the complete datasets of the different abstraction levels only when needed.

Besides the object-to-composite relationships, Table 14 also contains the classes of the objects and the cycles in which the objects are created. The objects in the original dataset can be recognised by the value 0 (zero) for the cycle. Object 822 illustrates that composite objects can be part of other, higher-level composites.

TABLE 14. TABLE CONTAINING THE OBJECT-TO-COMPOSITE RELATIONSHIPS (PART-OF RELATIONSHIPS) AND OBJECT-TO-CLASS RELATIONSHIPS

object	*class*	*part-of*	*cycle*
25	12	824	0
34	1	822	0
140	14	823	0
234	3	822	0
333	6	823	0
623	1	822	0
822	27	824	1
823	28		2
824	36		2

6.5 Summary

The implementation depends on the availability of a data model with stored topology. These appear to become scarcer amongst recently developed GIS data models. An advantage of this approach is that, after pre-processing the data in dedicated GIS software to determine neighbour relationships, the entire procedure can be performed quickly using regular database software.

The process is characterised by a cyclic approach. During each aggregation cycle two classes are related through the class adjacency index (CAI) and their adjacent objects aggregated when the threshold value for the objects aggregation factor (OAF) is met. The component and composite objects are interconnected between the successive cycles, enabling navigation through the levels of abstraction. This way a multi-scale dataset is created in the process. The implementation was tested on the two datasets described in Chapter 5. The results of this exercise will be presented in the following chapter.

Chapter 7: Results and discussion

Evaluating the results of an aggregation process like this is difficult as there is no existing material for comparison. Two complementary methods will be applied, visual (qualitative) and numerical (quantitative) evaluation. Although the method described in this study is not meant for traditional mapping, visual evaluation can still be useful because of the human ability to assess the results quickly and quite thoroughly by means of visual inspection.

7.1 Qualitative assessment of the results

7.1.1 TOP10vector, auto mode

Figure 46 to Figure 49 show the results of generalisation on a TOP10vector dataset, using the application described in chapter 6, in the form of a map. The order in which classes are combined (see Table 15) is determined automatically based on the class adjacency indices and the number of objects involved, as explained in section 4.5 (theory) and 6.3.1 (implementation). The object aggregation factor threshold value was set to the lowest possible value so that adjacent objects belonging to the combination in progress were all really aggregated.

The higher levels of abstraction become clear. The north-west corner is more open with mainly arable land and grassland, whereas the north-east corner is more densely occupied with buildings and forest features, although buildings surrounded by grass and arable land are concentrated largely south of the town. The structure of the town also gets clearer at higher levels of abstraction, the town quarters showing clearly in Figure 48.

The road classification is hierarchical, but this hierarchy is only used implicitly. There is no dedicated mechanism to ensure that least important road classes are aggregated first, but as the lower-level classes are more numerous they will aggregate first. This simple mechanism works reasonably well, as the results show, but there is no guarantee that the network structure is not broken up.

7.1.2 GBKN, auto mode

Figure 50 to Figure 53 show the results of generalisation on a GBKN dataset. The system parameters used were identical to those applied for the TOP10vector dataset. The GBKN data have far fewer distinct classes than the TOP10vector dataset (see Table 17); the number of iterations needed to achieve a large decrease in the number of objects is therefore smaller than in the case of TOP10vector.

An interesting phenomenon can be observed during the generalisation of the GBKN dataset. The town of Zevenaar that appears to be lost in the first aggregation cycle (Figure 51), when buildings and terrain objects are aggregated, reappears after the second cycle. This is due to the fact that terrain in built-up area is usually delimited by roads whereas the terrain objects outside of the town are generally bounded by water.

7.1.3 TOP10vector, step mode

The method allows user interaction at several points in the aggregation process (see section 4.8). To illustrate this, the TOP10vector dataset was aggregated once more, but this time with a few user interventions in the otherwise fully automated process. The road classes 3, 4, 5 and 6 (‘motorway’, ‘main road’, ‘secondary road’ and ‘unpaved road’) were excluded from the aggregation process. Then the aggregation process was started. The first combination, ‘building’ and ‘other land use’, was accepted by the user but the results of the second aggregation cycle, that involved the classes 1 and 12 (‘building’ and ‘grass/arable’), was rejected. From cycle III onwards, the system’s decisions were accepted again. Results of this exercise can be found in Figure 54, Figure 55 and Figure 56. Compare these results with Figure 47, Figure 48 and Figure 49, respectively, which represent the results of the fully automated process after the corresponding number of aggregation cycles. In Figure 55 and Figure 56 it can be observed that, due to the exclusion of most road classes from the aggregation process, the areas enclosed by roads are aggregated further than in the fully automated mode. This is not the case in the top left-hand corner where a number of ‘grass/arable’ objects remain unaggregated. This is caused by the fact that only objects of different classes are aggregated, not objects of the same class (unless initiated by another aggregation process, see section 0). The objects in these areas are only enclosed by roads and have no other option than to be aggregated with objects of the same class and therefore remain unchanged.

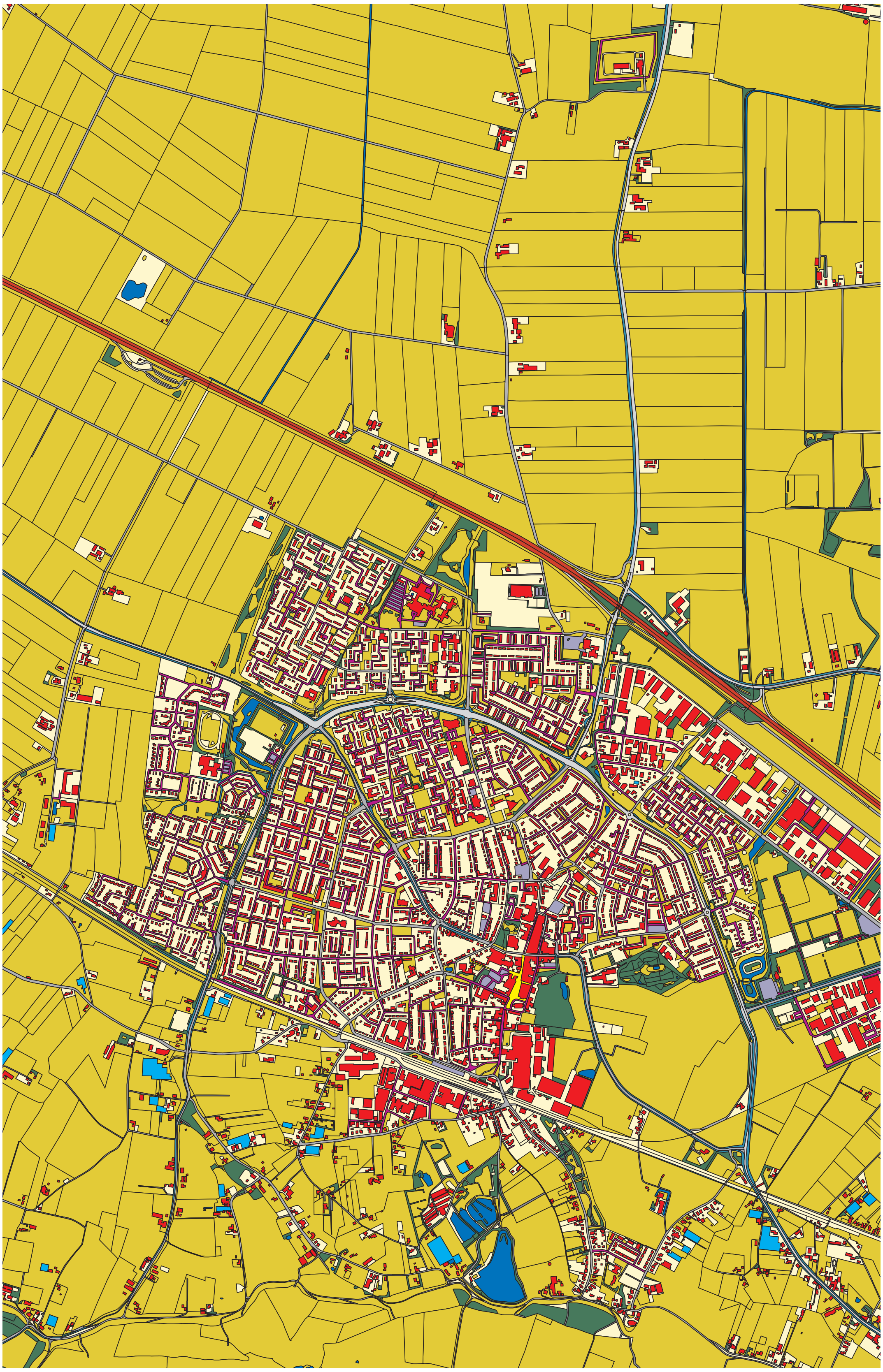


FIGURE 46. ORIGINAL TOP10VECTOR DATASET

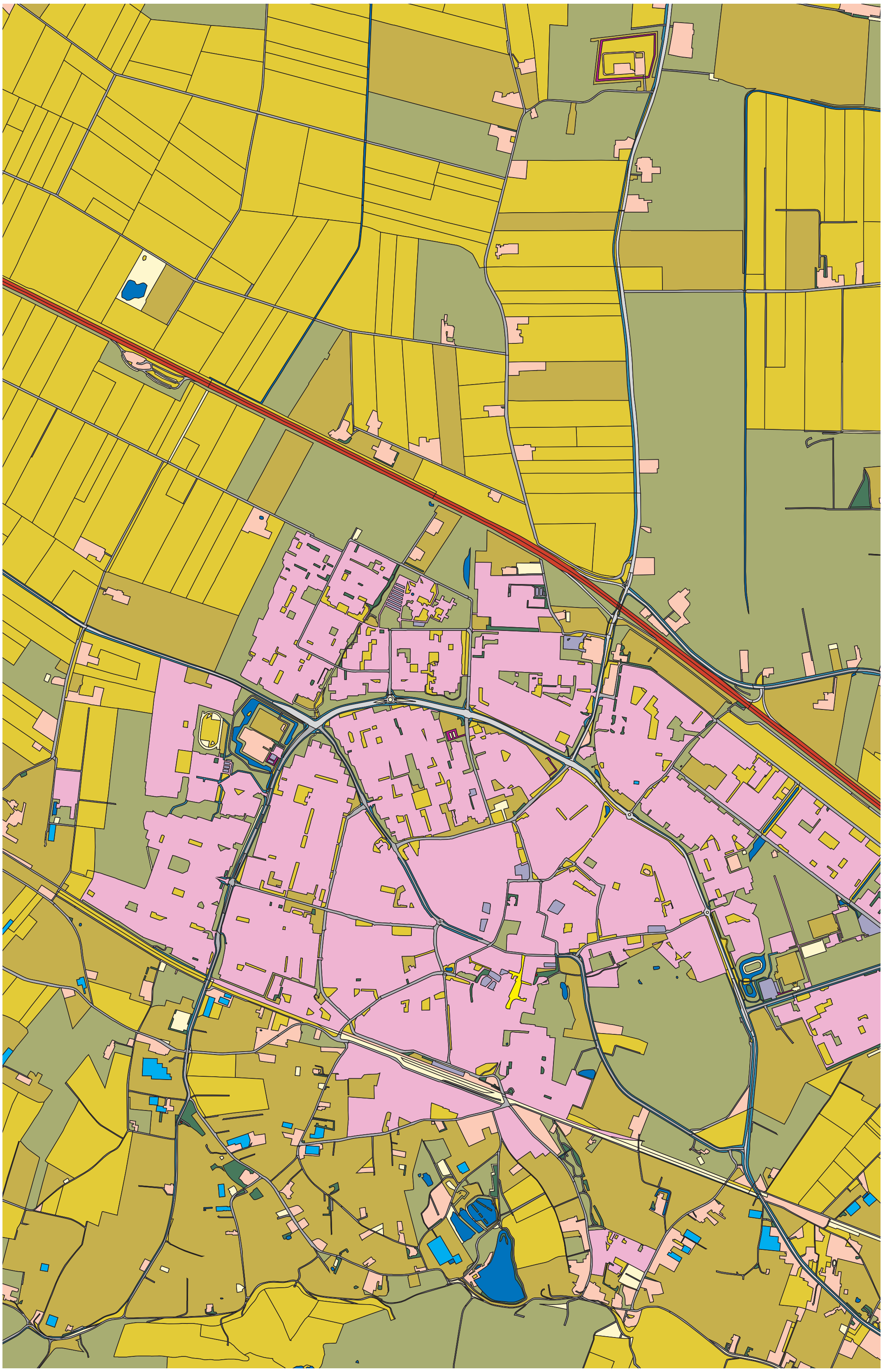


FIGURE 47. TOP10VECTOR AFTER AGGREGATION CYCLE IV IN AUTO MODE

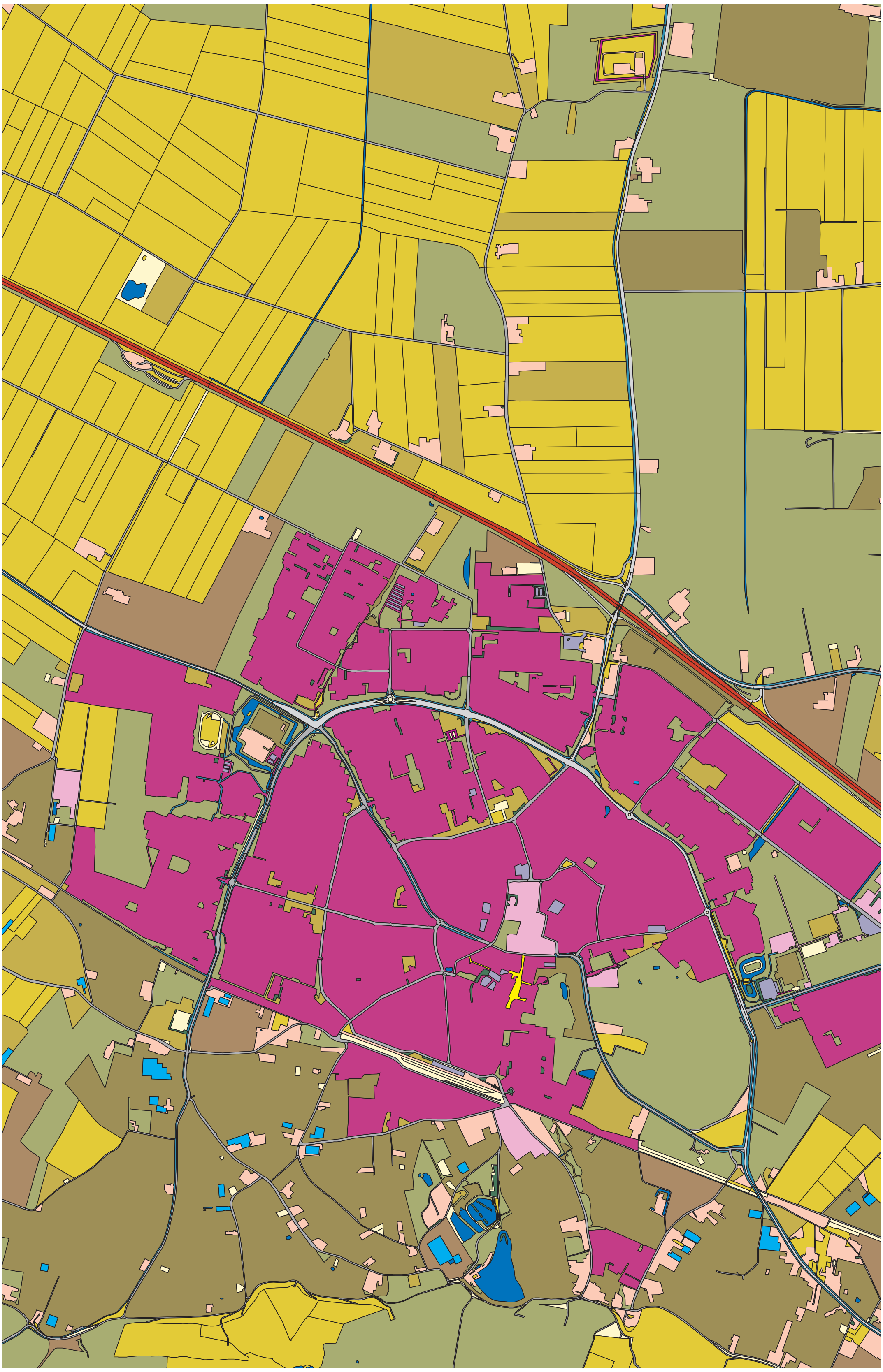


FIGURE 48. TOP10VECTOR AFTER AGGREGATION CYCLE VII IN AUTO MODE

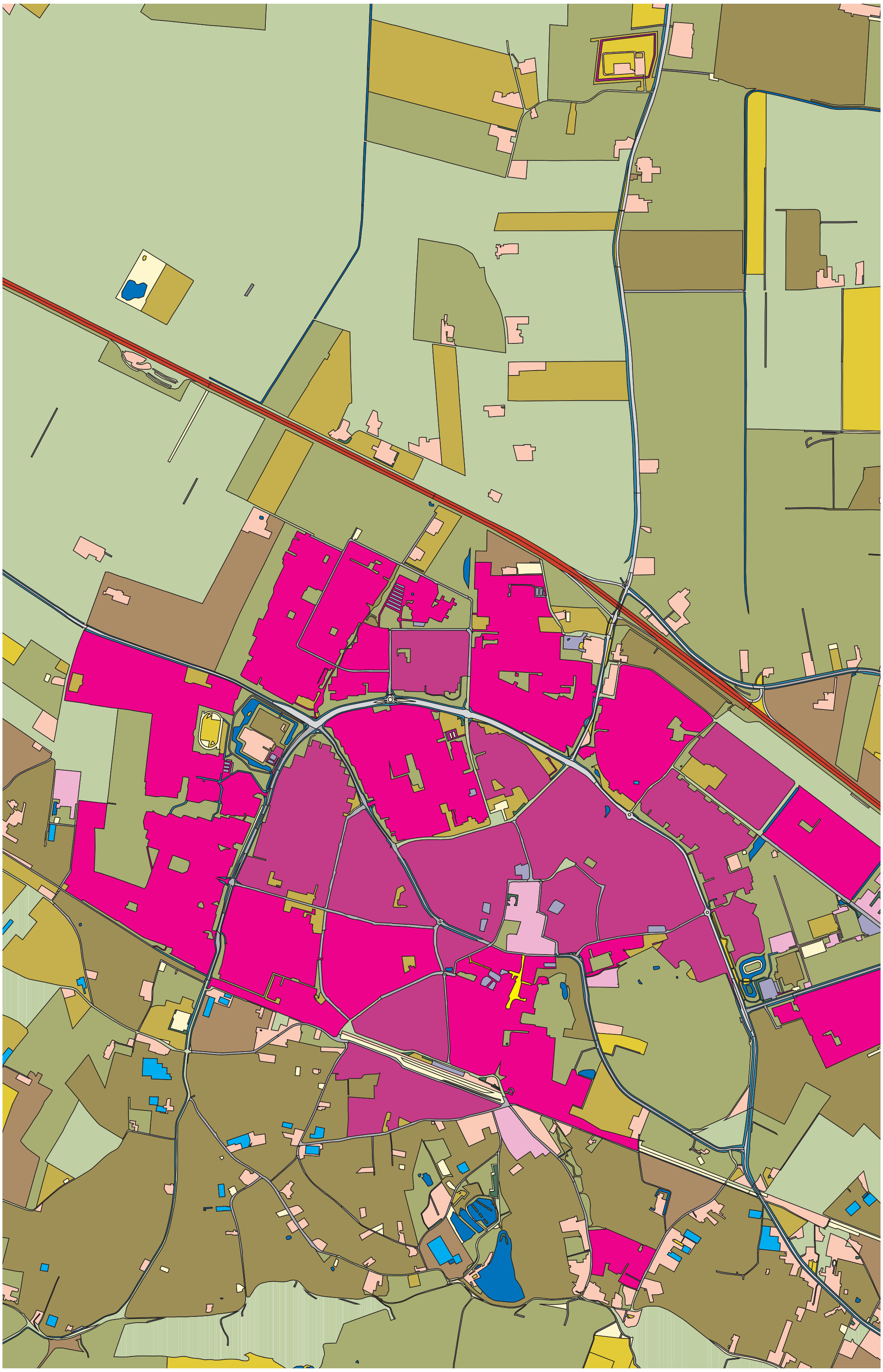


FIGURE 49. TOP10VECTOR AFTER AGGREGATION CYCLE IX IN AUTO MODE

LEGEND FOR FIGURE 46 TO FIGURE 49

1. building
2. greenhouse
3. motorway
4. main road
5. secondary road
6. unpaved road
7. pedestrians
8. street
9. bicycle track
10. parking space
11. forest
12. grass/arable
13. other land use
14. water
15. building & other land use
16. building & grass/arable
17. building, street & other land use
18. forest & grass/arable
19. building, street, grass/arable & other land use
20. building, unpaved road & grass/arable
21. building, unpaved road, forest & grass/arable
22. building, street, forest, grass/arable & other land use
23. secondary road & grass/arable

LEGEND FOR FIGURE 50 TO FIGURE 53

20. road
40. building
60. water
70. railway
80. terrain
81. building & terrain
82. building, water & terrain
83. road, building & terrain
84. water & terrain
85. road, building, water & terrain

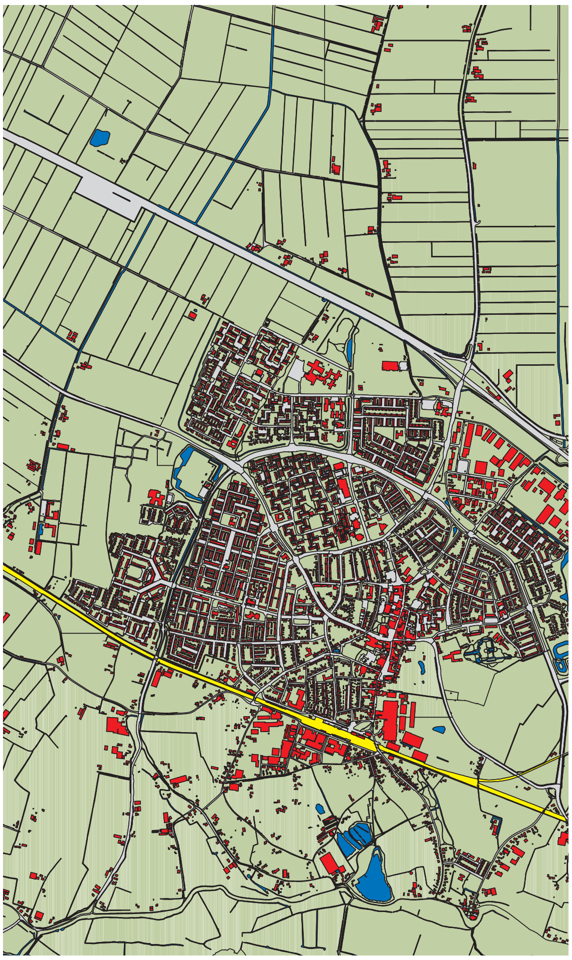


FIGURE 50. ORIGINAL GBKN DATASET

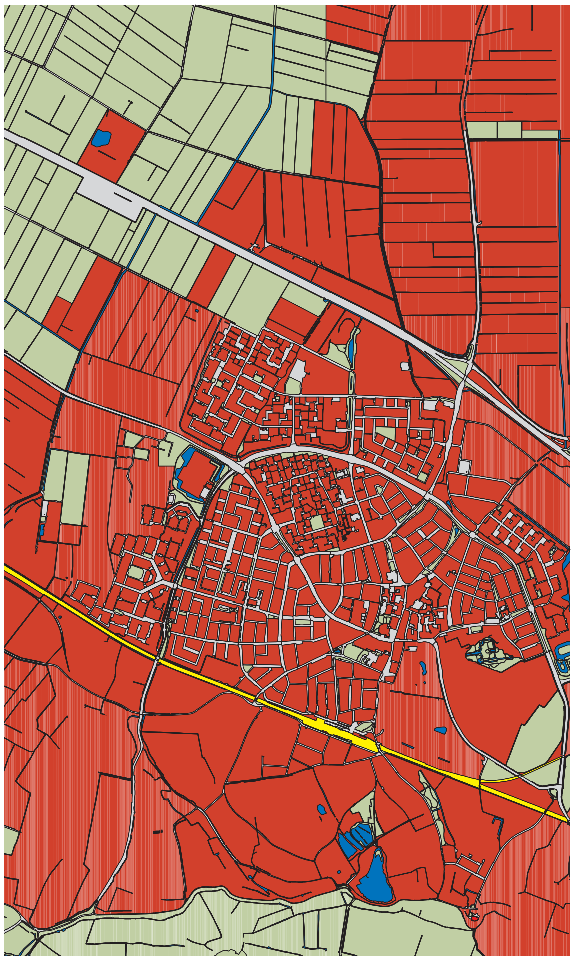


FIGURE 51. GBKN DATASET AFTER AGGREGATION CYCLE I IN AUTO MODE

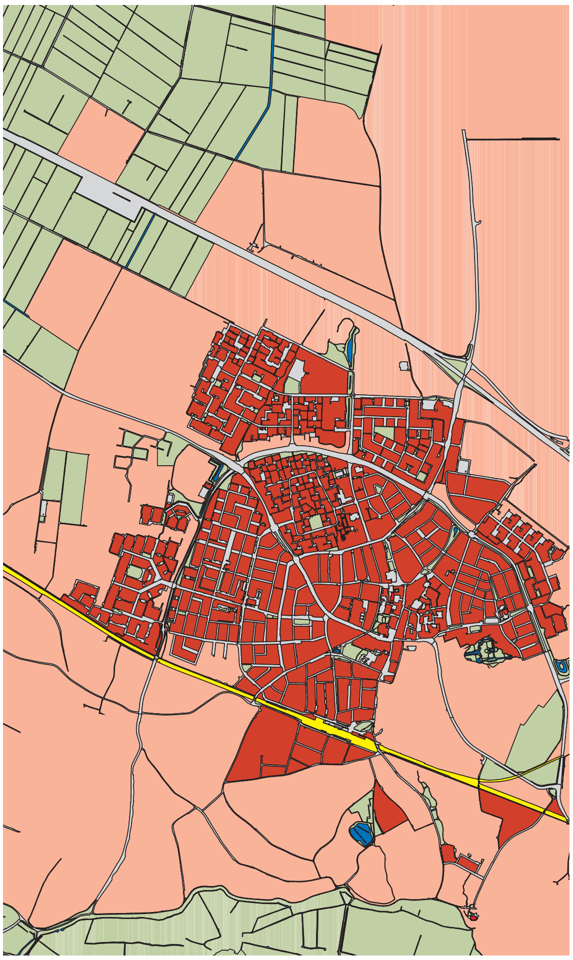


FIGURE 52. GBKN DATASET AFTER AGGREGATION CYCLE II IN AUTO MODE

FIGURE 53. GBKN DATASET AFTER AGGREGATION CYCLE V IN AUTO MODE

LEGEND FOR FIGURE 54 TO FIGURE 56 (FOLLOWING PAGES)

1. building
2. greenhouse
3. motorway
4. main road
5. secondary road
6. unpaved road
7. pedestrians
8. street
9. bicycle track
10. parking space
11. forest
12. grass/arable
13. other land use
14. water
15. building & other land use
16. building, street & other land use
17. forest & grass/arable
18. building, forest & grass/arable
19. building, street, grass/arable & other land use
20. building, grass/arable & other land use
21. building, forest, grass/arable & other land use
22. building, street, forest, grass/arable & other land use
23. building, street, forest, grass/arable & other land use

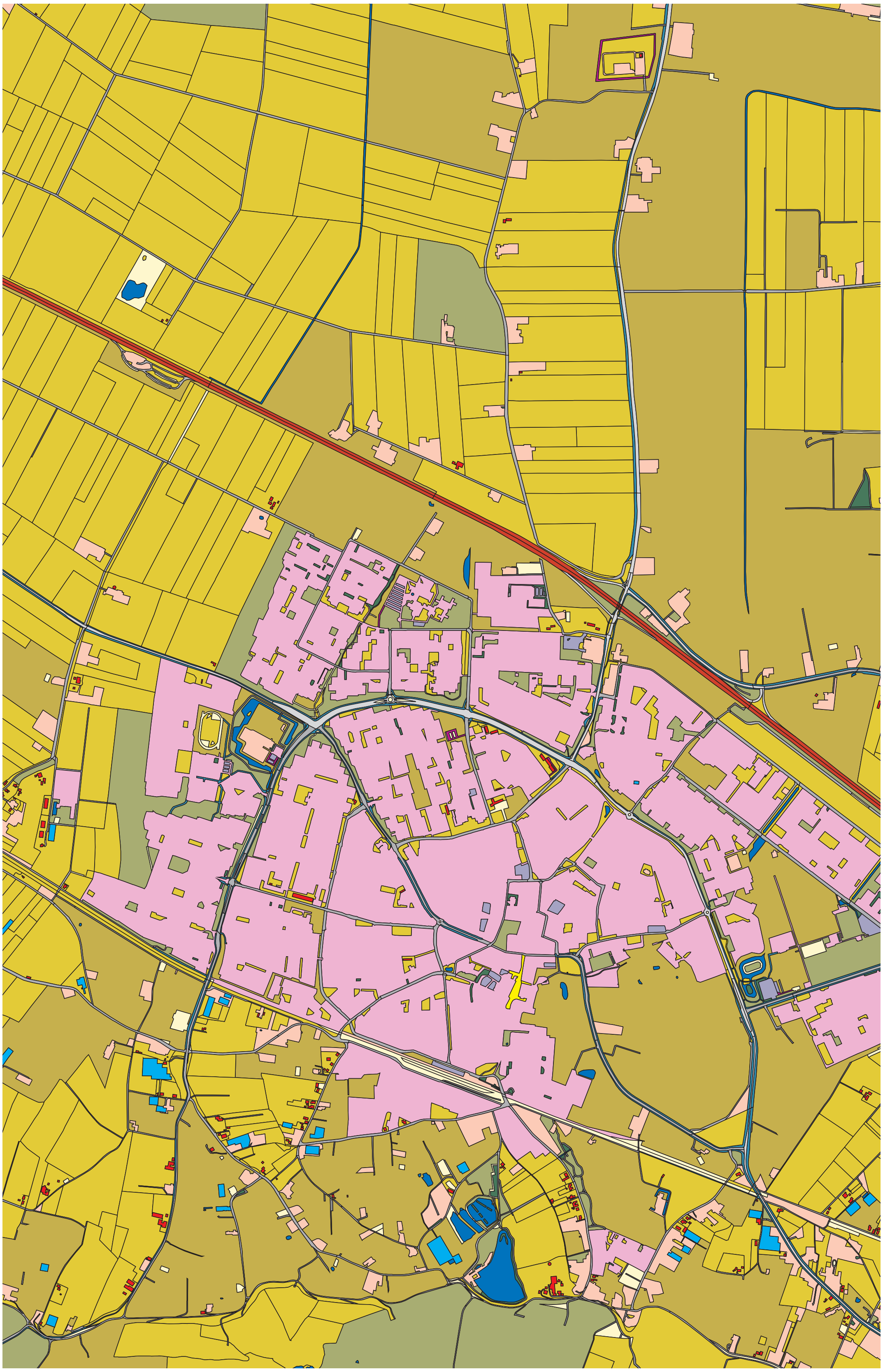


FIGURE 54. TOP10VECTOR AFTER AGGREGATION CYCLE IV IN STEP MODE

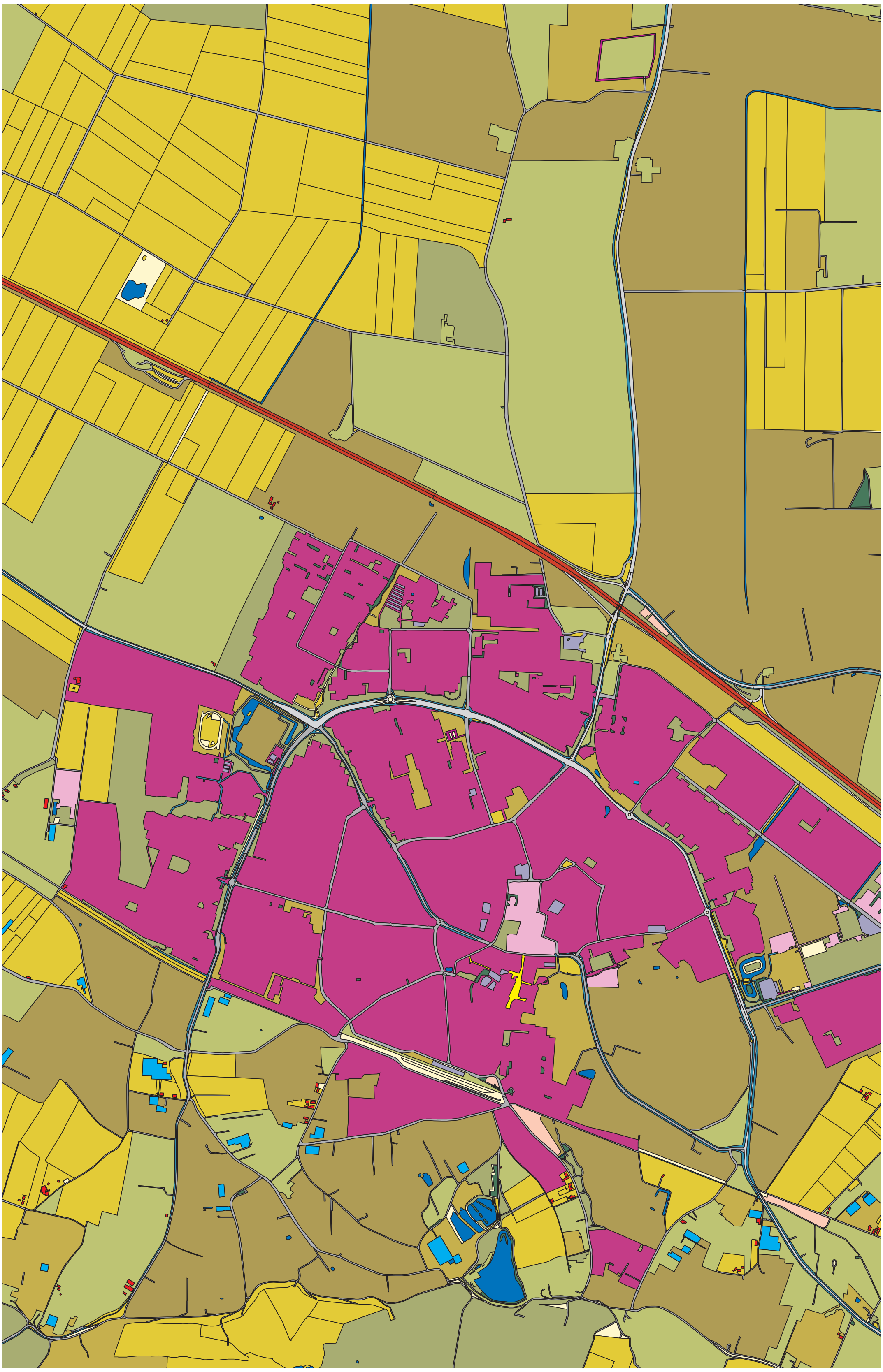


FIGURE 55. TOP10VECTOR AFTER AGGREGATION CYCLE VII IN STEP MODE

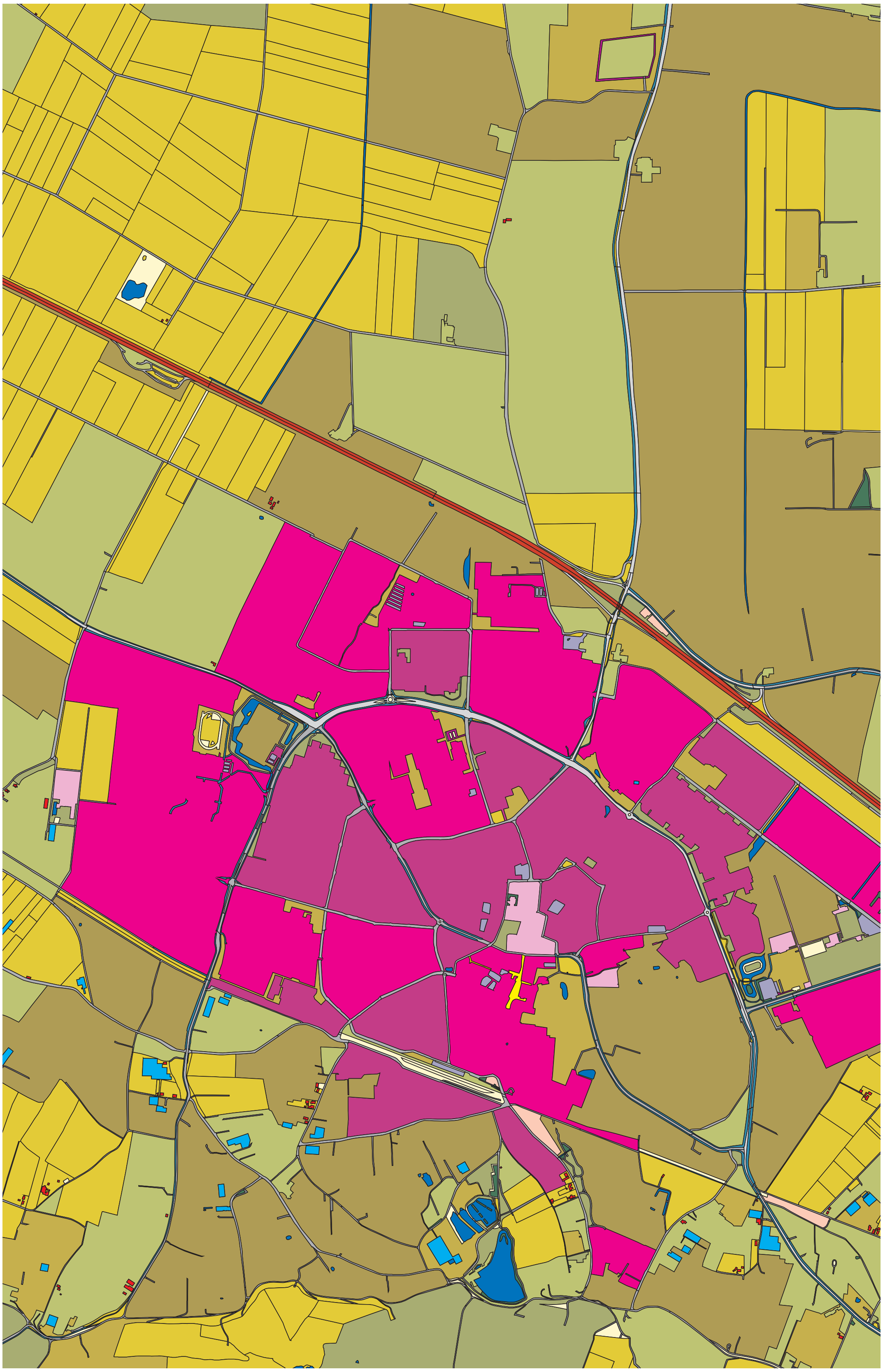


FIGURE 56. TOP10VECTOR AFTER AGGREGATION CYCLE IX IN STEP MODE

7.1.4 Anomalies due to data inconsistencies

Occasionally, the quality of the base data leads to unexpected aggregation results. An example emerges during the aggregation of TOP10vector objects belonging to the composite class consisting of objects of the classes 'building' and 'other land use' with objects of the class 'street'. As a rule in the TOP10vector dataset, the higher-level road is left uninterrupted at crossings. In this case this rule was violated, the result being an unintentionally interrupted main road after aggregation (Figure 57).

FIGURE 57. WRONGLY DELINEATED ROAD FEATURE CAUSING AGGREGATION PROBLEMS (TOP10VECTOR)

7.1.5 Using manually generalised maps as a reference

Comparing to existing, manually generalised products is a common method for evaluating the results of automated map generalisation systems. We do not compare the results with manually generalised maps because this study deals with model generalisation. The procedure does not aim to produce a dataset instantly suited for drawing maps. Consequently, there are no manually generalised maps produced following a comparable procedure (João 1998). Furthermore, often the results of manual generalisation themselves are not up to standard (João 1998)

7.2 Quantitative assessment of the results

7.2.1 Auto mode

Table 15 and Table 17 show the results of the mechanical aggregation processes for the TOP10vector dataset and the GBKN data regarding the reduction of the number of objects. Table 15 shows the quantitative data for the aggregation process that was described qualitatively in section 7.1.1. Table 17 is the quantitative representation of the results in section 7.1.2. The values represent the number of objects in a certain class (on the left) after every aggregation cycle (at the top). The last 4 rows provide some global measures for each aggregation cycle such as the total number of objects, reduction factor, mean object area and the time needed to complete the cycle. Numbers that remain unchanged compared to the previous cycle are shown in grey, so are classes that remain unchanged during the whole procedure. Classes involved in the class combination of a certain cycle - two component classes and one composite class - are shown in bold and other classes that are affected in regular black type.

The stopping criteria were 10 aggregation cycles in the case of TOP10vector and a maximum of 70 objects for the GBKN data.

In Table 15 we can see that the original classes ‘building’ and ‘street’ disappear almost completely from the TOP10vector dataset, whilst the classes ‘forest’, ‘grass/arable’ and ‘other land use’ are significantly reduced in number. In the case of the GBKN (Table 17) nearly all of the original objects have been aggregated after the fifth aggregation cycle, leaving less than one percent of the original number of objects. The rapid diminution of the reduction factor for the GBKN data is mainly due to the limited number of classes in the original dataset. It can also be observed that the classes 86 and 88 created in aggregation cycles VI and VIII are basically identical to 85 and 83, respectively; they contain the same original classes. They could therefore just as well be combined.

TABLE 15. NUMBER OF OBJECTS PER CLASS AFTER EACH AGGREGATION CYCLE (TOP10VECTOR, AUTO MODE)

	aggregation cycle	orig	I	II	III	IV	V	VI	VII	VIII	IX	X
	class / class combination		1 & 13	1 & 12	8 & 15	11 & 12	12 & 17	6 & 16	11 & 20	11 & 19	5 & 12	15 & 21
1	building	3918	**544**	**13**	**5**	5	5	5	**3**	3	3	3
2	greenhouse	28	28	28	28	28	28	28	28	28	28	28
3	motorway	18	18	18	18	18	18	18	18	18	18	18
4	main road	28	28	28	28	28	28	28	28	28	28	28
5	secondary road	192	192	192	192	192	192	192	192	192	**104**	104
6	unpaved road	236	236	236	236	236	236	**102**	**94**	94	94	**82**
7	pedestrians	4	4	4	4	4	4	4	4	4	4	4
8	street	406	406	406	**5**	5	**4**	4	4	4	4	4
9	bicycle track	111	111	111	111	111	111	111	111	111	111	111
10	parking space	34	34	34	34	34	34	34	34	34	34	34
11	forest	550	550	550	550	**179**	179	179	**105**	**37**	37	**35**
12	grass/arable	1371	1371	**962**	962	**579**	**336**	**304**	**298**	**295**	**62**	**57**
13	other land use	1096	**258**	258	**161**	161	**149**	149	149	**147**	147	**125**
14	water	67	67	67	67	67	67	67	67	67	67	67
15	1 & 13		**552**	552	**165**	165	**161**	161	161	161	161	**89**
16	1 & 12			**120**	120	120	120	**62**	**61**	61	61	61
17	8 & 15				**34**	34	**6**	6	6	6	6	6
18	11 & 12					**152**	152	152	152	152	152	152
19	12 & 17						**28**	28	28	**14**	14	14
20	6 & 16							**42**	**14**	14	14	14
21	11 & 20								**28**	28	28	**5**
22	11 & 19									**14**	14	14
23	5 & 12										**19**	19
24	15 & 21											**21**
	total number of objects	*8060*	*4400*	*3580*	*2721*	*2119*	*1859*	*1677*	*1586*	*1513*	*1211*	*1096*
	reduction factor (%)	*100*	*55*	*44*	*34*	*26*	*23*	*21*	*20*	*19*	*15*	*14*
	mean area (m^2)	*3172*	*5810*	*7062*	*9203*	*12071*	*13111*	*14477*	*16130*	*17742*	*21129*	*23348*
	processing time (sec.)		*105*	*53*	*52*	*38*	*30*	*29*	*27*	*26*	*28*	*22*

TABLE 16. COMPOSITION OF THE COMPOSITE CLASSES IN TABLE 15

composite class	class composition
15	building & other land use
16	building & grass/arable
17	building, street & other land use
18	forest & grass/arable
19	building, street, grass/arable & other land use
20	building, unpaved road & grass/arable
21	building, unpaved road, forest & grass/arable
22	building, street, forest, grass/arable & other land use
23	secondary road & grass/arable
24	building, unpaved road, forest, grass/arable & other land use

TABLE 17. NUMBER OF OBJECTS PER CLASS AFTER EACH AGGREGATION CYCLE (GBKN, AUTO MODE)

	aggregation cycle	**orig**	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**
class / class combination			40 & 80	60 & 81	81 & 20	60 & 80	20 & 82	84 & 85	40 & 70	40 & 83
20	Road	521	521	521	**109**	109	**7**	3	3	**2**
40	Building	9037	**49**	45	28	28	24	24	**21**	**20**
60	Water	740	740	**165**	165	**20**	12	12	12	12
70	Railway	8	8	8	8	8	8	8	**6**	6
80	Terrain	1281	**416**	386	112	**29**	5	5	5	4
81	40 & 80		**507**	**460**	**0**	0	0	0	0	0
82	60 & 81			**43**	43	43	**9**	9	9	9
83	81 & 20				**1**	1	1	1	1	**0**
84	60 & 80					**31**	31	**8**	8	8
85	20 & 82						**4**	**2**	2	2
86	84 & 85							**2**	2	2
87	40 & 70								**2**	2
88	40 & 83									**1**
	total number of objects	*11587*	*2241*	*1628*	*466*	*269*	*101*	*74*	*71*	*68*
	reduction factor (%)	*100*	*19*	*14*	*4.0*	*2.3*	*0.9*	*0.6*	*0.6*	*0.6*
	mean area (m^2)	*2003*	*10361*	*14265*	*49912*	*86600*	*232089*	*317930*	*346401*	*357059*
	processing time (sec.)		*191*	*17*	*20*	*5*	*4*	*2*	*2*	*2*

TABLE 18. COMPOSITION OF THE COMPOSITE CLASSES IN TABLE 17

composite class	class composition
81	building & terrain
82	building, water & terrain
83	road, building & terrain
84	water & terrain
85	road, building, water & terrain
86	road, building, water & terrain
87	building & railway
88	road, building & terrain

7.2.2 Step mode

The quantitative results of the process described qualitatively in section 7.1.3 are given in Table 19. Due to the exclusion of the classes 3 to 6 these remain unchanged. This contrasts with the results of the fully automated aggregation of the TOP10vector dataset (Table 15), where classes 5 and 6, secondary and unpaved road, *are* involved in the aggregation process and the number of objects in these classes therefore decreases. We also see that cycle II involves a class combination different from the mechanical approach of section 7.2.1. The combination originally suggested by the system - 1 & 12, ‘building’ and ‘grass/arable’ - was rejected by the user leaving the current combination (8 & 15) as the second choice. These adjustments lead to quite different combinations in the subsequent aggregation cycles as can be seen when we compare the class compositions in Table 20 with the compositions of the fully automated process in Table 16.

TABLE 19. NUMBER OF OBJECTS PER CLASS AFTER EACH AGGREGATION CYCLE (TOP10VECTOR, STEP MODE)

	aggregation cycle	orig	I	II	III	IV	V	VI	VII	VIII	IX	X
	class / class combination		1 & 13	8 & 15	11 & 12	1 & 17	12 & 16	12 & 15	15 & 18	11 & 19	17 & 22	1 & 20
1	building	3918	**544**	536	536	**233**	217	97	84	81	80	**59**
2	greenhouse	28	28	28	28	28	28	28	28	28	28	28
3	motorway	18	18	18	18	18	18	18	18	18	18	18
4	main road	28	28	28	28	28	28	28	28	28	28	28
5	secondary road	192	192	192	192	192	192	192	192	192	192	192
6	unpaved road	236	236	236	236	236	236	236	236	236	236	236
7	pedestrians	4	4	4	4	4	4	4	4	4	4	4
8	street	406	406	**6**	6	6	**3**	3	3	3	3	3
9	bicycle track	111	111	111	111	111	111	111	111	111	111	111
10	parking space	34	34	34	34	34	34	34	34	34	34	34
11	forest	550	550	550	**97**	96	96	96	95	**31**	31	31
12	grass/arable	1371	1371	1371	**782**	756	**491**	**271**	264	263	261	254
13	other land use	1096	**258**	161	161	161	148	135	106	105	101	101
14	water	67	67	67	67	67	67	67	67	67	67	67
15	1 & 13		**552**	**165**	165	165	161	**77**	**6**	6	6	6
16	8 & 15			**35**	35	35	**5**	5	5	5	4	4
17	11 & 12				**163**	**106**	106	106	104	104	**56**	56
18	1 & 17					**57**	57	57	**29**	29	29	29
19	12 & 16						**29**	29	29	**16**	16	16
20	12 & 15							**56**	56	56	56	**49**
21	15 & 18								**27**	27	27	27
22	11 & 19									**13**	**1**	1
23	17 & 22										**11**	11
24	1 & 20											**7**
	total number of objects	*8060*	*4400*	*3543*	*2664*	*2334*	*2032*	*1651*	*1527*	*1458*	*1401*	*1373*
	reduction factor (%)	*100*	*55*	*44*	*33*	*29*	*25*	*20*	*19*	*18*	*17*	*17*
	mean area (m^2)	*3172*	*5811*	*7216*	*9597*	*10954*	*12582*	*15485*	*16743*	*17535*	*18249*	*18621*
	processing time (sec.)		*117*	*71*	*57*	*37*	*59*	*40*	*24*	*40*	*42*	*25*

TABLE 20. COMPOSITION OF THE COMPOSITE CLASSES IN TABLE 19

composite class	class composition
15	building & other land use
16	building, street & other land use
17	forest & grass/arable
18	building, forest & grass/arable
19	building, street, grass/arable & other land use
20	building, grass/arable & other land use
21	building, forest, grass/arable & other land use
22	building, street, forest, grass/arable & other land use
23	building, street, forest, grass/arable & other land use
24	building, grass/arable & other land use

7.2.3 Generalisation effects, global

Generalisation effects are often expressed in terms of reduction factors and generalisation error measures. Reduction factors and indices (Richardson 1993, Bregt and Bulens 1996) are used to express a simple decrease in the number of objects, either per class or for the entire dataset. Other measures, like the attribute change index and error index (Bregt and Bulens 1996), assume a shift in thematic values. The area of a class before generalisation is compared to the area after generalisation. These measures are based on the assumption that the classes before and after generalisation are identical. However, the aggregation method described in this study introduces entirely new classes, so it is not possible to compare the situations before and after generalisation by means of these traditional measures. Using this method, generalisation errors in the form of incorrect classifications are not an issue. The new dataset is less detailed but correct. Only if the original categories are maintained after generalisation are misclassifications unavoidable. Misclassifications are the result of holding on to the same categories before and after generalisation.

But what effect measures *can* be used then? It is possible to define a global reduction factor (Richardson 1993), comparing the total number of objects in the generalised dataset to that of the original (Figure 58, Figure 59). The differences in the reduction factors of the two datasets can be explained - apart from the fact that the GBKN data contain far fewer classes - by the great number of 'outbuildings', often small garden sheds, present in the GBKN data. These annexes are aggregated into the built-up areas, accounting for a significant decrease during the first aggregation cycle.

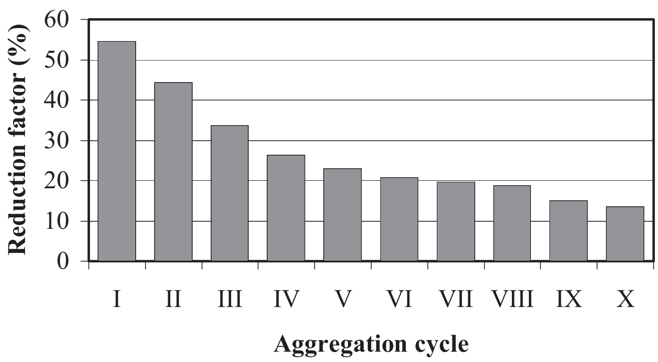


FIGURE 58. GLOBAL REDUCTION FACTORS FOR EACH AGGREGATION CYCLE (TOP10VECTOR, AUTO MODE)

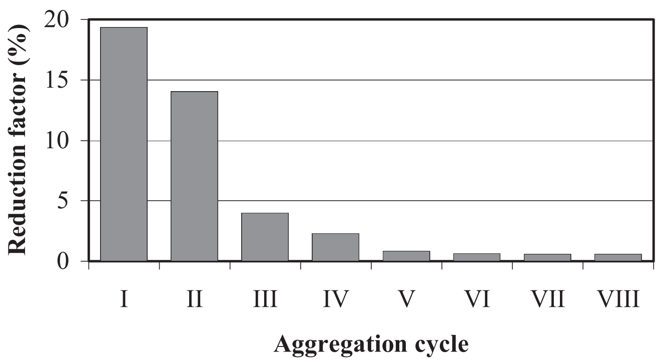


FIGURE 59. GLOBAL REDUCTION FACTORS FOR EACH AGGREGATION CYCLE (GBKN, AUTO MODE)

7.2.4 Other effect measures

Geometric measures such as minimal dimensions, for instance, objects that are too small or too close, are all related to the graphic representation. The 'too' in too small and too close is defined by the minimal resolution of the output medium, which is a map. These measures are therefore not evaluated. Violation of the network topology (roads, waterways) is prevented by never considering newly created composite objects part of any network structure. Existing connections can in these cases be eliminated just as in structural generalisation, but shortcuts cannot be created unintentionally. The procedure was tested on a 64-bit, 800-Mhz Digital Alpha DS10 with an internal memory of 1 Gb. Processing time varied from 105 to 22 seconds for the TOP10vector data and between 191 and 2 seconds for the GBKN dataset, both depending on the initial number of objects and the number of objects reclassified (see Table 15, Table 17). The time shown is for the aggregation process alone, the Arc/Info 'dissolve' operation to merge the geometry was not taken into account.

7.3 Discussion

The process described is completely automated, so it is not required that the class intensions be known. However, the operator *can* be given the possibility to reject certain class combinations and proceed with the next combination, which opens up the possibility of more application-specific aggregations.

Whereas the spatial structure of the dataset becomes more abstract, the thematic structure (the legend) gets more intricate, even though rarely all members of a class can be aggregated. In some cases the remnant of a class only consists of a handful of objects. In such cases it might be appropriate to eliminate those objects. Because of the small number of objects involved this would only cause few generalisation errors and may therefore be outweighed by the advantage of having a simplified legend.

The TOP10vector dataset contained 14 distinct classes, the GBKN dataset contained 5. The approach proved to work in both situations, although the larger number of classes of the TOP10vector data resulted in a more gradual decrease in the number of objects. An even larger number of classes was not tested, but this may be expected to give rise to an unmanageable number of composite classes.

The results are reproducible, providing the exact same dataset is used. Different spatial selections on the same type of dataset *could* lead to different results for a particular local scene, as the generalisation of a smaller selection of the TOP10vector dataset shows (Figure 60, Figure 61). This is due to the fact that in a different selection a certain class combination may appear more or less often, leading to different class adjacency indices and subsequently different class combinations (Table 21).

Legend for Figure 60 and Figure 61

1. building
2. greenhouse
3. motorway
4. main road
5. secondary road
6. unpaved road
7. pedestrians
8. street
9. bicycle track
10. parking space
11. forest
12. grass/arable
13. other land use
14. water
15. building & other land use
16. building, street & other land use
17. forest & grass/arable
18. building, street, grass/arable & other land use
19. building, forest & grass/arable
20. building, street, forest, grass/arable & other land use
21. secondary road & grass/arable
22. building, secondary road, grass/arable & other land use
23. building, street, forest, grass/arable & other land use

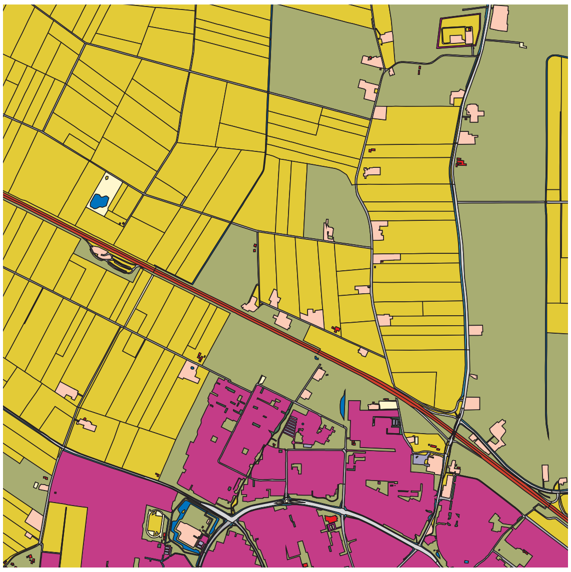


Figure 60. Smaller TOP10vector selection after aggregation cycle IV (compare to Figure 47)

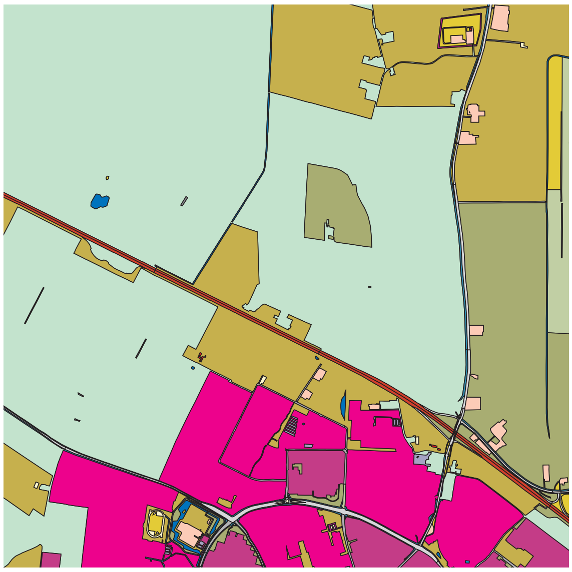


Figure 61. Smaller TOP10vector selection after aggregation cycle IX (compare to Figure 49)

Table 21. Identified class combinations using a smaller selection (Figure 60 and Figure 61), compare with Table 15 and Table 16

aggregation cycle	composite class	class combination	class composition
I	15	1 & 13	building & other land use
II	16	8 & 15	building, street & other land use
III	17	11 & 12	forest & grass/arable
IV	18	12 & 16	building, street, grass/arable & other land use
V	19	1 & 17	building, forest & grass/arable
VI	20	11 & 18	building, street, forest, grass/arable & other land use
VII	21	5 & 12	secondary road & grass/arable
VIII	22	15 & 21	building, secondary road, grass/arable& other land use
IX	23	17 & 20	building, street, forest, grass/arable & other land use
X	24	10 & 23	building, street, parking space, forest, grass/arable & other land use

A network topology can be broken up accidentally in two ways:

1. at class level: a more important class is selected for aggregation before the lower-level parts are selected;
2. at object level: certain objects in a class are aggregated while ‘upstream’ elements of the same class stay intact.

The first situation can be prevented by forcing network classes to be aggregated in hierarchical order. The second one is more complicated because certain segments cannot be aggregated for lack of suitable neighbours.

More and more, generalisation will become a tool for unlocking the underlying, more detailed data. This means that the generalised information has to aid in determining *where* to look into the data in more detail. For that reason the different levels of abstraction will have to be linked together, to enable switching from one level to another. The process as described here connects composite and component objects to achieve this.

In section 1.2, generalisation was referred to as a type of context transformation. Context transformations occur when we convert datasets to adapt them to other types of use. Context transformations are characterised by changes in any of the three components of context information: class, class intension and geometric description. The aggregation process described in this study affects the class part of contexts.

The current procedure can be enhanced in a number of ways:

1. A procedure for the *structural generalisation* of networks can ensure that the network topology of the roads is not corrupted.
   Structural generalisation requires the roads to be converted from area objects to a topological graph of nodes and segments to be able to determine the topological structure of the network, but such skeletonisation methods are currently part of mainstream GIS applications (Lee 1999, ESRI 2000, Lee 2001). By maintaining the object identifiers of the original area objects during the conversion, a classification created using the topological graph can be transferred back to the area objects in the single-valued vector map to allow for the structurally least important elements to be aggregated first. The object of this procedure is to prevent that instances of less important subclasses such as blind alleys outlive objects that belong to a higher-level road subclass in the aggregation process, causing road objects to remain that are no longer connected to the rest of the network.
   Several approaches for the classification of the elements of non-directed topological graphs can be found in literature. Peng described a method for urban road networks based on the form and size of the enclosed areas (Peng 1995). Other approaches use shortest-path spanning trees (Richardson and Thomson 1996) or the continuity of the elements (Thomson and Richardson 1999). The results of these automated classifications are encouraging.

2. By using *class-driven generalisation* - aggregation of objects of the same class or superclass - in addition to the current method it is possible to prevent situations as described in section 7.1.3, where some objects (‘grass/arable’) remain unaggregated for lack of suitable aggregation partners from a different class.

3. *Geometric generalisation* (Molenaar 1998) can be used to eliminate remnants of old classes. One has to be bear in mind, however, that geometric generalisation on

categorical data introduces generalisation errors, since the face of the eliminated object is merged with a neighbouring object of a different class.
Following aggregation, some classes have been almost completely included in composite classes, and only some instances of these classes have 'survived'. These instances can often not be aggregated any further because they all have neighbours of different classes, which would require a new composite class for every single object. Then another approach is needed and geometric generalisation comes into the picture again, also *because* only a few objects are involved and generalisation errors would therefore be kept to a minimum.

4. In this study, aggregation is only possible for connected objects. Peng and Liu, among others, described methods for the amalgamation of unconnected neighbours, based on extended adjacency relationships (Peng 1995, Liu 2002). The data models used were the extended formal data structure (EFDS) and the integrated and extended formal data structure (IEFDS), respectively. Both models are extensions of the formal data structure (FDS) used in this research. Amalgamation of area objects is currently possible with commercial GIS software (ESRI 2000).
   It is likely that this enhancement will also introduce generalisation errors since with the aggregation of unconnected objects, the areas in between - which are part of other objects in a single-valued vector map - are included in the composite.
5. The implementation currently works at a predetermined taxonomy level. Adjacency could be evaluated at several levels of a classification hierarchy, to determine at which level co-occurrence relationships occur. This level could subsequently be used for aggregation.
   For instance, the class adjacency index between roads and lots could be investigated, but also more specifically the class adjacency index between streets - 'street' being a subclass of 'road' - and lots. Since roads also occur in rural areas and streets are limited to urban surroundings, the relationship between streets and lots is presumably stronger than the one between roads and lots.

If the generalised dataset needs to be visualised, additional cartographic generalisation may be inevitable. Small and narrow objects may need to be exaggerated or eliminated. This is to be done after the conceptual generalisation has been performed and it would preferably not affect the underlying data.

Chapter 8: Conclusions and recommendations

8.1 Conclusions

8.1.1 Objective

The objective of this study was to develop a framework and a working prototype for the generalisation of object- and vector-based categorical maps - such as large-scale topographic data - based on inter-object relationships, thereby striving for a system that is to a large extent automated and can be operated by non-expert users.

Aggregation based on the class adjacency index is a generalisation method that appears suitable for regular geographic database generalisation as well as exploratory purposes, as described by the *private - high interaction - exploration of unknown* corner of MacEachren's map use cube (MacEachren and Kraak 1997). The method is quick, fully automated, and all thematic information is retained. The exact spatial positions are lost but these can be recovered by descending down the aggregation tree. Although here the process is based on the class attribute, the method can be employed on any nominal attribute attached to the objects.

The current implementation is meant to demonstrate the method based on the class adjacency index. It is not intended as a complete application for conceptual generalisation. This is especially noticeable in the way network-forming object classes have been handled. Extensions, however, have been suggested in the discussion section.

Compared to international map generalisation research this study is different because it does not involve any cartographic operations and it is based on the aggregation of *dissimilar* objects to create new, *composite* objects. The aggregation of objects that are not directly related through taxonomy relationships was examined earlier in national generalisation research (e.g. van Oosterom 1995, Bregt and Bulens 1996), but these approaches relied on manually assigned compatibility measures to identify aggregation options.

Using the method described here, spatial detail decreases while the classification becomes more intricate. This could result in complicated legends when used for traditional mapping. The method *can* be used for mapping purposes but additional cartographic generalisation is then required to remove insignificant remnants of the original classes and to remove or exaggerate remaining objects that are too small or narrow to be displayed at the target scale.

8.1.2 Research questions

Research Question 1: What are the consequences if we concentrate completely on conceptual, that is non-cartographical aspects of the generalisation process? What are the relevant operations in that case and how do we assess the result?

If we concentrate on non-cartographic issues the number of operations is reduced significantly, enabling us to concentrate on one of the most important conceptual data reduction processes: aggregation. It was shown that in an area partitioning elimination and aggregation are closely related but aggregation is generally to be preferred; elimination should only occur if aggregation is not possible and the object under evaluation is small or unimportant.

Creating composite objects by aggregation and merging the geometry of the components are considered two separate actions. During the abstraction process we only work with object referencing, the original topology is retained. This helps to ensure topological consistency during the operations.

Assessment of the results of conceptual generalisation is still problematic. The existing material is not suited for comparison and the existing generalisation assessment measures are oriented towards graphic generalisation, comprising measures for displacement, etc.

Research Question 2: How are the objects in a categorical map interrelated thematically and spatially and how can we use these relationships for the definition of generalisation rules?

Categorical data require a holistic approach, i.e. the spatial relationships between the classes are important. Many aggregation approaches are based on thematic similarity between the classes. These similarity measures are normally based on expert knowledge. Similarity is commonly expressed in a classification hierarchy but classification hierarchies are not a very suitable basis for abstraction. Since the objects remain the same - just the 'label' changes and only occasionally can adjacent objects belonging to the same superclass be merged - this only works within a limited spatial range.

The aggregation process in this study is based on spatial measures and is automated completely. No manually assigned compatibility measures for the classes are required. The class intension needs not even be known, as the process is entirely based on topology and geometry. Minimal requirements are basic classes like building, waterway, road etc., but more detailed classifications like main road, motorway and brook, river and canal are to be preferred. Network structures, in particular, require more elaborate classifications to enable elimination/aggregation of less important segments. If these classifications cannot be created automatically, by database enrichment operations, they have to be included in the initial dataset. Although existing classification hierarchies may not entirely suit the purpose of the generalisation, it is always better than nothing. It is important to include not only the most detailed classifications but the superclasses as well. However trivial these may seem, the superclasses can serve as a basis for database enrichment operations.

What topologic relationships can be queried depends on the data model used by the application to perform the queries. Topology does not need to be included in the input data, it is a property of the data model used to process the data. Topology can be stored as a part of the data model or derived when needed. A data model with stored topology like the FDS allows topological relationships to be queried in a set-at-a-time manner, like the neighbour relationships between the members of two classes in this research. In other words, an entire dataset can be queried in a statistical manner. Topological correctness or consistency, on the other hand, is a property of the data. A data model like the formal data structure (FDS) can enforce topological consistency.

In order to prevent the generalisation rules from being too application- and dataset-specific, these should not address specific classes but categories of classes such as networks.

Research Question 3: What parameters can be defined for the user to control the outcome of the generalisation process?

The system can be operated completely automatically or interactively. In auto mode the user starts with setting a stopping criterion, after which the system repeatedly aggregates class combinations until the criterion is met. In step mode the user evaluates each aggregation step before continuing. This enables the user to adjust the result for a certain purpose. The following steering parameters are available:

1. In both modes: excluding classes from the aggregation process. Some classes can be protected so that the aggregation process does not affect these classes.
2. In auto mode: setting the stopping criterion. This stopping criterion can be:
   a. the maximum number of objects in the resulting dataset,
   b. the maximum number of classes in the resulting dataset,
   c. the maximum number of aggregation cycles to be performed.
3. In step mode: accepting or rejecting an aggregation step. If an aggregation step is rejected, the current class combination is excluded and the system reverts to the result of the previous step and picks the next combination.

Research Question 4: How can we minimise errors, such as shifts in thematic values and topological inconsistencies?

Creating new, composite classes (of dissimilar component classes) is an unusual approach in map generalisation literature, but unlike most other approaches it does not lead to generalisation errors. The new data are more abstract, but correct. Generalisation errors only occur if the described method is complemented by other generalisation operations such as geometric generalisation and amalgamation.

8.1.3 Additional conclusions

Because of its ability to facilitate large abstraction steps and the absence of generalisation errors, aggregation can be seen as *the* central operation in a generalisation process.

The implementation in a (relational) database management system (DBMS) ensures that all information needed during the course of the entire generalisation process is accessible at any moment and that you can store all additional information needed. The database allows storage of topological and thematic information about the object and retrieval of this information at any given moment. The integration of all information in one database allows spatial and thematic properties to be queried simultaneously, as opposed to most current GIS's that require separate queries for spatial and thematic data. The use of a common relational DBMS also prevents the method from being too dependent on the possibilities of the current GIS software packages.

Object-orientated programming is not required to create a system capable of automated generalisation nor is an object-oriented DBMS needed. The data model should nevertheless show objects, features in reality that form identifiable objects in the database. This is called the object-*based* approach. Objects are uniquely identified by an object identifier. Objects can be defined at different levels of abstraction; not only can objects be part of composite objects, but objects can also be composed of several geometric primitives. Composite and component objects (including geometric primitives) are related through their object identifiers, this means that effectively a dataset with multiple representations is created.

It is very important that the rules for data acquisition are clear, both regarding the classification of objects and the spatial delineation of the objects. A good example is the delineation of road segments to prevent problems as shown in section 5.2.2. It is generally preferable to define geometric primitives at a low level of spatial abstraction.

This is particularly important in network structures: network segments and junctions should be individually identifiable. The geometry of an object is assembled from one or more of these geometric primitives. This offers the most flexibility in constructing other objects that are different from the ones that were defined initially. It allows the identification of road segments as well as roads. It is easier to combine spatial objects than to break them apart, since it is necessary to add geometry to split up objects. In NEN3610, the spatial description is detailed enough, providing the model is correctly implemented. The thematic structure of NEN3610, however, is not very suited for generalisation purposes, mainly because of a lack of functional classifications of network elements.

8.2 Future research

The real issue of conceptual generalisation is reclassification; what are the higher-level objects we are looking for? Detection of functional inter-class relationships is therefore one of the important topics for future research. Detecting spatial co-occurrence of classes based on neighbour relationships of the member objects is one way of identifying functional relationships but other methods should be investigated. Future research should therefore concentrate on:

- Other methods for detecting relationships between classes.
- Other methods for detecting spatial co-occurrence of classes.

The current method can also be *refined* further, especially regarding the local object aggregation factor as well as the fine-tuning of this parameter. The object aggregation factor can be enhanced by using additional measures such as object orientation so that parallel objects such as roadways and the accompanying pavements are even more likely to be aggregated, for instance. Also, in the discussion section a number of possible *extensions* of the current method were presented. These are:

1. Integration of structural generalisation of network data to ensure the topological consistency of the network.
2. Integration of the class-driven approach and the current method.
3. Addition of geometric generalisation operations (elimination) to remove remnants of old classes.
4. The use of extended adjacency relationships to enable the aggregation of non-connected neighbours (amalgamation).
5. Evaluation of spatial relationships at different levels of a taxonomy, to determine at which level significant relationships occur.

The method described here cannot be evaluated using conventional generalisation effect measures. New generalisation effect parameters are needed for conceptual generalisation.

Since the possibilities for generalisation depend on the spatial and thematic structure of the input data, database enrichment operations continue to be an important field of research, e.g. into the classification of undirectional networks such as roads.

References

Armstrong, M.P., 1991. Knowledge classification and organization. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 86-102.

Bader, M. and Barrault, M., 2001. Cartographic Displacement in Generalization: Introducing Elastic Beams. Fourth ICA Workshop on Progress in Automated Map Generalization, Beijing.

Barrault, M., Bader, M., and Weibel, R., 2000. Topology Preserving Conflict Removal between Symbolized Roads in Cartographic Generalization: Extending Snakes Methods. GIScience2000.

Bishr, Y., 1997. Semantic Aspects of Interoperable GIS. PhD thesis, Wageningen Agricultural University and ITC.

Brassel, K. and Weibel, R., 1988. A review and conceptual framework of automated map generalization. International Journal of Geographical Information Systems, 2(3), pp. 229-244.

Bregt, A. and Bulens J., 1996. Application-Oriented Generalisation of Area Objects. In: Molenaar, M. (ed.), Methods for the Generalisation of Geo- Databases, NCG, Delft, pp. 57-64.

Bundy, G.L., Jones, C.B. and Furse, E., 1995. Holistic generalization of large-scale cartographic data. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Burghardt, D. and Meier, S., 1997. Cartographic Displacement using the Snakes Concept. In: Foerstner, W. and Pluemer, L. (eds.), Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, Birkhauser-Verlag, Basel.

Chithambaram, R., Beard, K. and Barrera, R., 1991. Skeletonizing polygons for map generalization. Proceedings ACSM-ASPRS, Baltimore, pp. 44-55.

DGIWG DIGEST, 1992. Digital geographic information - exchange standards - edition 1.1. Digital Geographic Information Working Group, Defence Mapping Agency, USA.

Elmasri, R. and Navathe, S.B., 1989. Fundamentals of Database Systems. The Benjamin/Cummings Publishing Company, Redwood City, California.

ESRI, 1995. Understanding geographic information systems: the ARC/INFO method.

ESRI, 2000. Map Generalisation in GIS: Practical Solutions with Workstation Arc/Info Software. ESRI Technical Paper, ESRI, Redlands, CA.

Frank, A. and Egenhofer, M., 1988. Object-Oriented Database Technology for GIS. Seminar Workbook, San Antonio, Texas.

Frank, A.U. and Mark, D.M., 1991. Language Issues for GIS. In: Maguire, D.J., M.F. Goodchild and D.W. Rhind (eds.), Geographical Information Systems. Volume 1: Principles, Longman, England.

Galanda, M., 2001. Optimization Techniques for Polygon Generalization. Fourth ICA Workshop on Progress in Automated Map Generalization, Beijing.

Gatrell, A.C., 1991. Concepts of Space and Geographical Data. In: Maguire, D.J., M.F. Goodchild and D.W. Rhind (eds.), Geographical Information Systems. Volume 1: Principles, Longman, England.

Grelot, J-P., 1986. Archaic data models or hardware as a concept killer. Proceedings Auto-Carto, London, pp. 572-577.

Grünreich, D., 1985. Ein Vorschlag zum Aufbau einer großmaßstäbigen topographisch-kartographischen Datenbank unter besonderer Berücksichtigung der Grundrißdatei des ALK-Systems. Nachrichten aus dem Karten- und vermessungswesen, Series I, pp. 53-62.

Hansen, G.W. and Hansen, J.V., 1992. Database Management and Design. Prentice Hall, Englewood Cliffs, New Jersey.

Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., 1983. Building expert systems. Addison-Wesley, Reading.

Herbert, G., João, E.M. and Rhind, D., 1992. Use of an artificial intelligence approach to increase user control of automatic line generalisation. Proceedings of the 3rd European Conference on Geographical Information Systems (EGIS), Munich, pp. 554-563.

Højholt, P., 1998. Solving Local and Global Space Conflicts in Map Generalisation Using a Finite Element Method Adapted from Structural Mechanics. Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, pp. 679-689.

João, E.M., 1995. The importance of quantifying the effects of generalization. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

João, E.M., 1998. Causes and Consequences of Map Generalisation. Taylor & Francis.

João, E.M., Herbert, G., Rhind, D.W., Openshaw, S. and Raper, J., 1993. Towards a generalization machine to minimize generalization effects within a GIS. In: Mather, P.M. (ed.), Geographical Information Handling - Research and Applications, John Wiley and Sons Ltd.

Jones, C.B., 1991. Database architecture for multi-scale GIS. Technical Papers, 1991 ACSM-APRS Annual Convention, Baltimore, pp. 1-14.

Jones, C.B. and Ware J.M., 1998. Nearest Neighbour Search for Linear and Polygonal Object with Constrained Triangulations. Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, pp. 13-21.

Jones, C.B. and Ware, J.M., 1998. Proximity relations with triangulated spatial models. The Computer Journal, 41(2), pp. 71-83.

Jones, C.B., Bundy, L. and Ware, J.M., 1995. Map Generalization with Triangulated Data Structure. Cartography and Geographic Information Systems, 22(4), pp. 317-331.

Keller, S.F., 1995. Potentials and limitations of artificial intelligence techniques applied to generalization. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Kilpelaïnen, T., 1992. Multiple representations and knowledge-based generalization of topographical data. Proceedings ISPRS, Washington, pp. 954-964.

Kilpelaïnen, T., and Sarjakoski, T., 1995. Incremental generalization for multiple representations of geographical objects. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Langran, G., 1991. Generalization and parallel computation. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 204-217.

Laurini, R. and Thompson, D., 1992. Fundamentals of Spatial Information Systems. Academic Press.

Lee, D., 1999. Practical Solutions for Specific Generalisation Tasks. Third ICA Workshop on Progress in Automated Map Generalization, Ottawa.

Lee, D., 2001. Moving towards new technology for generalization. Fourth ICA Workshop on Progress in Automated Map Generalization, Beijing.

Liu, Y., 2002. Categorical Database Generalization in GIS. Dissertation number 88. ITC, the Netherlands.

MacEachren, A.M. and Kraak, M.J., 1997. Exploratory Cartographic Visualization: Advancing the Agenda. Computers & Geosciences, 23(4), pp. 335-343.

Mackaness, W.A., Weibel, R. and Buttenfield, B.P., 1997. Report of the ICA Workshop on Map Generalization. Gaevle, Sweden.

Mark, D.M., 1991. Object modelling and phenomenon-based generalization. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 103-118.

Martinez Casasnovas, J.A., 1994. Hydrographic Information Abstraction for Erosion Modelling at a Regional Level. MSc thesis, Wageningen Agricultural University.

McMaster, R.B., 1991. Conceptual frameworks for geographic knowledge. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 21-39.

McMaster, R.B., 1995. Knowledge acquisition for cartographic generalization experimental methods. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Molenaar, M., 1989. Single-valued Vector Maps - A Concept in GIS. Geo-Informationssysteme, 2(1), pp. 18-26.

Molenaar, M., 1993. Object hierarchies and uncertainty in GIS or why standardisation is so difficult. Geo-Informationssysteme, 6(3).

Molenaar, M., 1998. An introduction into the theory of topologic and hierarchical object modelling for geo-information systems. Taylor & Francis.

Müller, J-C. and Wang, Z-S., 1993. Complex coastline generalization. Cartography and Geographic Information Systems, 20(2), pp. 96-106.

Müller, J-C., Lagrange, J-P. , Weibel, R. and Salgé, F., 1995. Generalization: state of the art and issues. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Müller, J-C., Lagrange, J-P. and Weibel, R., 1995. GIS and Generalization: Methodology and Practice. Taylor & Francis.

Nederlands Normalisatie-instituut, 1995. Terreinmodel Vastgoed: Termen, definities en algemene regels voor de classificatie en codering van de aan het aardoppervlak gerelateerde ruimtelijke objecten. Nederlands Normalisatie-instituut.

Nederlands Normalisatie-instituut, 2001. Terreinmodel Vastgoed: Termen, definities en algemene regels voor de classificatie en codering van de aan het aardoppervlak gerelateerde ruimtelijke objecten (wijzigingsblad A1). Nederlands Normalisatie-instituut.

Nyerges, T.L., 1991. Representing geographical meaning. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 59-85.

Oosterom, P. van, 1990. Reactive data structures for Geographic Information Systems. Leiden University.

Oosterom, P. van, 1995. The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Peng, W., 1995. Object-oriented automated generalization and supporting data model. ITC - PhD Research Report.

Peng, W., 1997. Automated Generalization in GIS. PhD thesis, Wageningen Agricultural University and ITC.

Peter, B., 2001. Measures for the Generalization of Polygonal Maps with Categorical Data. Fourth ICA Workshop on Progress in Automated Map Generalization, Beijing.

Peter, B. and Weibel, R., 1999. Using Vector and Raster-Based Techniques in Categorical Map Generalization. Third ICA Workshop on Progress in Automated Map Generalization, Ottawa.

Plazanet, C., Bigolin, N.M. and Ruas, A., 1998. Experiments with Learning Techniques for Spatial Model Enrichment and Line Generalization. Geoinformatica, 2(4), pp. 315-334.

Putten, J. van and Oosterom, P. van, 1999. New results with Generalized Area Partitioning. Generalisatie met de computer, NVK Publicatiereeks nummer 28. Nederlandse Vereniging voor Kartografie.

Ravi, 1995. Nationale geo-informatie infrastructuur (NGII). Ravi, Amersfoort.

Ravi, 1998. Gebruikerseisen structuur en distributie Kernbestand. Ravi publicatie 98-5.

Ravi, 2002. Troefkaarten in de informatie-infrastructuur. Ravi publicatie 01-02.

Richardson, D.E., 1993. Automated Spatial and Thematic Generalization using a Context Transformation Model. PhD thesis, Wageningen Agricultural University, Wageningen, the Netherlands.

Richardson, D.E. and Thomson, R.C., 1996. Integrating thematic, geometric and topological information in the generalization of road networks. Cartographica Monograph 47, 33(1), pp. 41-54.

Robinson, G.J., 1995. A hierarchical top-down bottom-up approach to topographic map generalization. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Ruas, A. and Lagrange, J-P., 1995. Data and knowledge modelling for generalization. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Sester, M., Anders, K.H. and Walter V., 1998. Linking Objects of Different Spatial Data Sets by Integration and Aggregation. Geoinformatica 2(4), pp. 335-358.

Shea, K.S., 1991. Design criteria for an artificially intelligent system. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 3-20.

Smaalen, J.W.N. van, 1996(a). A hierarchic rule model for geographic information abstraction. Proceedings of the 7th International Symposium on Spatial Data Handling, Delft.

Smaalen, J.W.N. van, 1996(b). Spatial Abstraction Based on Hierarchical Re-classification. Cartographica, Monograph 47. 33(1), pp. 65-73.

Smaalen, J.W.N. van, 1999. Generalization based on topology, Generalisatie met de computer. NVK Publicatiereeks nummer 28. Nederlandse Vereniging voor Kartografie.

Smith, J.M. and Smith, D.C.P., 1977. Database Abstractions: Aggregation. Communications of the ACM, 20(6), pp. 405-413.

Smith, J.M. and Smith, D.C.P., 1989. Database Abstractions: Aggregation and Generalization. In: Mylopoulos, J. and Brodie, M., Readings in Artificial Intelligence and databases, Morgan Kaufmann Publishers, San Mateo, California.

Thomson, R.C. and Richardson, D.E., 1999. The ‘good continuation’ principle of perceptual organization applied to the generalization of road networks. Proceedings of the 19th International Cartographic Conference, Ottawa, pp. 1215-1223.

Timpf, S. and Frank, A.U., 1995. A multi-scale DAG for cartographic objects. Proceedings Auto-Carto 12. pp. 157-163.

Tobler, W.R., 1970. A computer movie simulating urban growth in the Detroit region. Economic Geography, pp. 234-240.

Uitermark, H., 2001. Ontology-Based Geographic Data Set Integration. PhD thesis, Universiteit Twente, the Netherlands.

Weibel, R., 1991. Amplified intelligence and rule-based systems. In: Buttenfield, B.P. and McMaster, R.B. (eds.), Map Generalization, Longman Scientific & Technical, pp. 172-186.

Weibel, R., 1995. Three essential building blocks for automated generalization. In: Müller, J.C., Lagrange, J.P. and Weibel, R. (eds.), GIS and Generalization: Methodology and Practice, Taylor & Francis.

Werschlein, T. and Weibel, R., 1994. Use of neural networks in line generalization. Proceedings of the 5th European Conference on Geographical Information Systems (EGIS), pp. 76-85.

List of symbols

s_r .... segment r

n_c .... node *c*

$BEGIN[s_r, n_c] = 1$ .... node *c* is the begin node of segment r

$END[s_r, n_d] = 1$ .... node *d* is the end node of segment r

$NODE[s_r, n_c] = 1$ .... node *c* is a node of segment r

f_g .... face *g*

o_k .... area object *k*

$LEFT[s_r, f_g] = 1$ .... segment r has face *g* on its left side

$RIGHT[s_r, f_h] = 1$ .... segment r has face *h* on its right side

$BOUNDARY[s_r, f_g] =1$ .... segment r is part of the boundary of face *g*

$PART_{22} [f_g, o_k] = 1$ .... face *g* is a part of area object *k*

$LEFT[s_r, o_k] = 1$ .... segment r has area object *k* on its left side

$RIGHT[s_r, o_m] = 1$ .... segment r has area object *m* on its right side

$BOUNDARY[s_r, o_k] = 1$ .... segment r is part of the boundary of area object *k*

$ADJACENT[o_k, o_m \mid s_r] = 1$ .... area objects *k* and *m* are adjacent at segment r

$MEMBER[o_k, C_p] = 1$ .... object *k* is a member of class *p*

$Length(s_r)$ .... the *length* of segment r

$Area(o_k)$ .... the *area* of object *k*

$Perimeter(o_k)$ .... the *perimeter* of object *k*

$Card(S_t)$ .... the cardinality of set *t* , i.e. the number of objects in set *t*

O_M .... the set of all area objects in map M

$Card(O_M)$ .... the number of objects in map M

$Ext(C_p)$ .... the extension of class *p*, i.e. the collection of members of class *p*

$CAI\ (C_a \mid C_b)$ .... the *class adjacency index* for the combination of class *a* and class *b*

$SR\ (C_a \mid C_b)$ .... the *significance ratio* for the combination of class *a* and class *b*

$OAF\ (o_d)$ .... the *object aggregation factor* for object *d*

Abstract

Since the late 1960's automated methods for map generalisation have been studied, but thus far no comprehensive system has been achieved. This is due to the general complexity of the matter, part of which is caused by the inability to separate the conceptual and the graphic issues. These aspects of map generalisation are considered separate issues ever since the advent of GIS but in practice it has been difficult to disconnect the conceptual issues from the impediments of graphic representation, either in the form of a paper map or on a computer screen. Current research into automated map generalisation generally appears to be in a cul-de-sac for this reason.

This study therefore aims to concentrate on strictly non-graphic operations and large generalisation steps, i.e. big scale changes. Whereas most existing methods work towards a clear end result, this approach does not. Instead, it is entirely based on the input data. Minimizing generalisation errors is a priority and assessment of the generalisation results is also an issue to consider. The goal is to develop a system for the generalisation of object- and vector-based categorical maps, such as large-scale topographic data, that is to a large extent automated and can be operated by non-expert users. In the past, several generalisation procedures have been developed for individual objects and dichotomous maps, but the number of procedures for categorical maps is still limited and the methods that do exist rely on similarity and importance factors that are hard to determine.

Large-scale categorical data mostly form an area partition, i.e. the whole spatial extent of the dataset is covered by objects and the objects do not overlap. This implies that objects cannot simply be removed - since this would cause 'holes' - but have to be combined or aggregated.

Objects can be aggregated based on taxonomy or partonomy relationships. Taxonomy relationships are based on similarity between the objects or classes. Aggregation based on taxonomy relationships has been described extensively in map generalisation literature, but only works within a limited spatial range. Since this study is aimed at large-scale changes it is based on the much less described partonomy relationships. Inter-object and inter-class relationships are used to determine functionally related classes in order to aggregate the object instances of the class. It is assumed that spatial correlations indicate functional relationships. The class adjacency index is used as a measure of spatial correlation between classes. Combinations of classes with a high class adjacency index are likely candidates for the creation of composite classes. Adjacent objects of these classes can subsequently be aggregated and reclassified to create composite objects.

The class adjacency index is determined based on adjacency measures of the member objects. The input dataset must therefore form a topologically correct, object-based area partition. The implementation is based on a stored adjacency graph and uses regular relational database management software. The data model is object-based and supports the concept of composite objects. In the process a multiple representations dataset is produced by connecting the composite objects created in every aggregation cycle to the constituent parts in the previous level.

The process can be fully automated but it is also possible to allow user interaction at several points in the process without compromising the approach. Since it is entirely based on characteristics of the input dataset, the method is also suited for exploratory

purposes. To a certain degree, the meaning of the classes is not even relevant, although in interactive mode the user naturally has to be aware of the classes.

The method was applied to two Dutch topographic datasets: TOP10vector and GBKN. The results show that this is a very promising method for conceptual generalisation. The concept of composite classes makes that generalisation errors are not an issue. Therefore, it cannot be evaluated using conventional generalisation effect measures. The output of the aggregation process is not readily suitable for mapping purposes, and additional cartographic generalisation is in that case required. The current implementation is not intended as a complete solution for conceptual generalisation. But since it is set in an environment of other conceptual generalisation operations, such as structural generalisation and extended adjacency graphs, it can be extended to create such a comprehensive system.

Samenvatting

Sinds het einde van de zestiger jaren wordt onderzoek gedaan naar geautomatiseerde methoden voor kaartgeneralisatie. Tot dusverre zijn geen alomvattende systemen gerealiseerd. Dit omdat het een zeer ingewikkeld onderwerp betreft, deels veroorzaakt doordat de conceptuele en de cartografische aspecten niet goed (kunnen) worden gescheiden. Deze beide aspecten worden als afzonderlijke onderwerpen gezien sinds de opkomst van de geografische informatiesystemen, maar in de praktijk blijkt het erg moeilijk de conceptuele zaken te scheiden van de beperkingen die een grafische weergave in de vorm van een kaart met zich meebrengen. Hierbij maakt het niet uit of het een papieren kaart betreft of een afbeelding op een beeldscherm. Het onderzoek naar geautomatiseerde kaartgeneralisatie lijkt hierdoor grotendeels op een dood spoor te zijn beland.

Dit onderzoek richt zich daarom strikt op niet-cartografische operaties en daarnaast op grote stappen in het generalisatieproces, i.e. grote schaalovergangen. Waar de meeste bestaande methoden naar een bekend eindresultaat toewerken, is dat in dit onderzoek niet het geval. In plaats daarvan is het volledig gebaseerd op de uitgangsgegevens. Het minimaliseren van generalisatiefouten is een prioriteit en ook wordt er aandacht besteed aan verschillende manieren om de resultaten te beoordelen. Doel is te komen tot een systeem voor generalisatie van object- en vectorgebaseerde, geclassificeerde gegevenssets - zoals grootschalige topografische data - dat voor een zo groot mogelijk deel is geautomatiseerd en kan worden toegepast door niet-ingewijde gebruikers. In het verleden zijn diverse generalisatiemethoden ontwikkeld voor individuele objecten en zogenaamde aan- en afwezigheidskaarten, maar methoden voor geclassificeerde kaarten zijn schaars en de methoden die er zijn, zijn gebaseerd op moeilijk vast te stellen factoren die overeenkomst en importantie weergeven.

Geclassificeerde grootschalige gegevens zijn meestal opgebouwd uit een aaneengesloten bedekking van vlakken. Dat wil zeggen dat de hele ruimte is bedekt met objecten die elkaar niet overlappen. Dit betekent echter dat objecten niet simpelweg kunnen worden verwijderd - omdat er dan gaten in de beschrijving ontstaan - maar moeten worden samengevoegd met aangrenzende objecten.

Objecten kunnen worden samengevoegd op basis van taxonomische relaties (classificatierelaties) of op zogenaamde ‘deel-van’-relaties. Taxonomische relaties zijn gebaseerd op overeenkomsten tussen de objecten of de klassen. Samenvoeging (aggregatie) op basis van deze relaties komt veel voor in de literatuur over kaartgeneralisatie, maar werkt slechts binnen een beperkt ruimtelijk bereik. Omdat dit onderzoek zich richt op grote aggregatiestappen is het in plaats daarvan gebaseerd op de veel minder beschreven ‘deel-van’-relaties. Relaties tussen objecten onderling, objecten en klassen, en klassen onderling worden gebruikt om functioneel gerelateerde klassen te vinden. Hierbij wordt verondersteld dat ruimtelijke correlatie wijst op functionele relaties. Als maat voor de ruimtelijke correlatie tussen klassen wordt de *class adjacency index* gebruikt. Combinaties van klassen met een hoge waarde voor de *class adjacency index* zijn het meest geschikt om samengestelde klassen te vormen. Aangrenzende objecten van deze klassen kunnen vervolgens worden samengevoegd en geherclassificeerd tot samengestelde objecten.

De *class adjacency index* wordt bepaald op basis van buurrelaties tussen de objecten in de klassen. De uitgangsgegevens dienen daarom te bestaan uit een topologisch correcte, aaneengesloten bedekking van objecten. De implementatie is gebaseerd op een opgeslagen topologische graaf van buurrelaties en maakt gebruik van gangbare relationele database management software. Het datamodel is objectgebaseerd en ondersteunt het gebruik van samengestelde objecten. Tijdens het proces wordt een meerschalige dataset gecreëerd doordat de samengestelde objecten uit elke aggregatiecyclus worden gekoppeld aan de objecten uit de voorgaande cyclus waaruit ze zijn samengesteld.

Het proces kan volledig worden geautomatiseerd, maar het is ook mogelijk om de gebruiker het proces op diverse punten te laten beïnvloeden zonder inbreuk te doen op de aanpak. Omdat de methode volledig is gebaseerd op kenmerken van de uitgangsgegevens is zij ook geschikt voor het verkennen van een dataset. De betekenis van de klassen is tot op zekere hoogte niet van belang. Hoewel de gebruiker in de interactieve modus natuurlijk wel op de hoogte moet zijn van de betekenis.

De methode is toegepast op twee topografische datasets: TOP10vector en GBKN. De resultaten tonen aan dat dit een veelbelovende methode voor conceptuele generalisatie is. Door het concept van samengestelde klassen is er geen sprake van generalisatiefouten. Evaluatie met de gebruikelijke generalisatie-effectmaten is daardoor niet mogelijk. De output van het aggregatieproces is niet zonder meer te gebruiken voor het maken van kaarten. In dat geval is aanvullende cartografische generalisatie noodzakelijk. De huidige implementatie beoogt ook niet een volledige oplossing voor conceptuele generalisatie te zijn, maar omdat zij is ingebed in een omgeving met andere conceptuele generalisatie-operaties kan zij worden uitgebreid om een dergelijk volledig systeem te creëren.

Curriculum Vitae

John van Smaalen was born on 27 October 1965 in Ede, the Netherlands. In 1993 he received his degree in landscape planning from Wageningen University. Main subjects during his studies were Geographic Information Systems and Information Systems Management. During his studies he conducted a traineeship at the Canada Centre for Remote Sensing in Ottawa.

From 1993 to 1994 he worked at Wageningen University on GIS projects and as a lecturer in GIS. In 1994 he started his PhD research on the conceptual generalisation of geographic information, which, after an intermission, he finished recently. In the meantime, from 1999 to 2002, he worked as a project manager for Ravi, focusing in particular on the development and promotion of a model for digital information in urban planning. This was done in close cooperation with governmental organisations at a local, provincial and national level, urban planning offices and software companies.

He was invited to give guest lectures at ITC, Enschede, within the framework of the MSc course Geographic Information Systems and for the TEMPUS Structural Joint European Project 'Joint Curricula Development for Soil and Water Resources Protection' (SWARP), Warsaw, Poland.

Recently he joined the Institute for Biodiversity and Ecosystem Dynamics (IBED) at the University of Amsterdam where he is responsible for the development of the master courses spatial modelling, GIS, and remote sensing.

Acknowledgements

This research was realised with the support of the Dutch Cadastre and Public Registers Agency (Kadaster), the Survey Department of the Directorate-General of Public Works and Water Management (Meetkundige Dienst van Rijkswaterstaat) and the Geodetic Commission of the Netherlands (Nederlandse Commissie voor Geodesie).