KUNGL TEKNISKA HÖGSKOLAN

TRITA-GEOFOTO 2001:18
ISSN 1400-3155
ISRN KTH/GEOFOTO/R--01/18-SE

Automatic generation of a view to a geographical database

Mats Dunkars

Licentiate Thesis

Royal Institute of Technology (KTH)
Division of Geodesy and Geoinformatics
100 44 Stockholm

September 2001

Abstract

This thesis concerns object oriented modelling and automatic generalisation of geographic information. The focus however is not on traditional paper maps, but on screen maps that are automatically generated from a geographical database. Object oriented modelling is used to design screen maps that are equipped with methods that automatically extracts information from a geographical database, generalises the information and displays it on a screen. The thesis consists of three parts: a theoretical background, an object oriented model that incorporates automatic generalisation of geographic information and a case study where parts of the model have been implemented.

An object oriented model is an abstraction of reality for a certain purpose. The theoretical background describes different aspects that have impact on how an object oriented model shall be designed for automatic generalisation. The following topics are described: category theory, the human ability to recognise visual patterns, previous work in automatic cartographic generalisation, and object oriented modelling.

A view is here defined to consist of several static levels, or maps, defined at different resolutions. As the user zooms the level that is appropriate for the particular resolution is shown. An object class belongs to one and only one level and has a certain symbolisation. The automatic creation of new objects in a level is discussed as well as the relation between objects in different levels. To preserve topological relations between objects in a level a network structure is formed between all linear objects in a level and objects that might cause conflicts are modelled using dependencies.

The model is designed for a set of typical geographical object classes such as road, railroad, lake, river, stream, building, built-up area etc. The model is designed to handle information in a scale-range from 1:10 000 to 1:100 000. The model has been implemented for a subset of these classes and tested for an area covering approximatley 60 km^2.

Key words

Digital cartography, automatic cartographic generalization, object oriented modelling

Acknowledgements

First I like to thank my supervisor Hans Hauska who patiently tried to understand the ideas presented in this thesis when they first evolved and who provided valuable guidance as the work matured.

I would also like to thank my co-supervisor Bengt Rystedt, and Liqiu Meng who was a supervisor during the initial part of this work. They introduced me to research in automatic generalisation of geographic information and have provided valuable comments on this work.

During my studies and work on this thesis I have had frequent contacts with Anna Bergman and Lars Harrie. Thanks for interesting discussions and valuable comments and also for proof-reading this thesis. Thanks also to David Douglas for valuable support and for having proof-red this thesis.

During my research I have been sitting in the midst of the rather hectic environment of a consultant company. I like to thank my colleagues, and especially Jan Zakariasson, for their support during this project and for giving me enough space to be able to focus on my research.

To try out the ideas presented in this thesis a few M.Sc. projects has been initiated. I would like to thank Maria Sjödin and Anna Strid who finished their M.Sc. project in February 2000 and Andreas Larsson and Guha Chandrasekaran who are currently working on their M.Sc. projects.

This thesis is a part of a joint research project between University of Gävle and VBB VIAK Company, sponsored by the Foundation of Knowledge and Competence Development, the Swedish Armed Forces, the Swedish Road Administration, the Swedish Railroad Administration, the Swedish Maritime Administration, the Geological Survey of Sweden, ESRI, Swedish Space Co-operation and the Swedish National Land Survey.

Finally, I would like to thank my family Isabel and Veronika, for love and encouragement during this work and for helping me avoid drowning in thoughts about research.

CONTENTS

1 Introduction

One current trend in Geographic Information Systems, GIS, is that an increasing number of geographical databases are made available by an increasing number of organisations. Traditionally, the National Mapping Agencies took a major role in the mapping activities even though there were several other organisations and companies such as the municipalities or the geological survey that produced and supplied maps. Today in Sweden the National Road Administration is building up a detailed database of the road network, the National Road Database (NVDB), the National Rail administration has a detailed database of the rail network, different energy companies maintain databases of the electricity-network, the municipalities maintain large scale databases etc. The list can be made much longer.

Another trend is the increasing use of maps in a variety of information systems. A characteristic of several of these maps is that they are part of a user interface that is designed for a particular user group and used for a particular purpose. Some examples of these kinds of maps are:

- Vehicle navigation, where the driver has a map showing roads, the best route, ongoing construction work that hinders the traffic and additional information that might be relevant such as petrol stations or restaurants.

- Within a municipality the geographical database has a rich set of data updated by several organisations. This data is accessed by several users such as urban planners, the environmental agency, maintenance personnel of the different utility departments, the fire department etc. All these different users want to access different subsets of the database.

From these two trends I have noted an emerging need to be able to extract data from several geographical databases and merge them together into a user interface that contains a screen map. However, there are difficulties with this approach and significant amount of work is put into the creation of standards for geographical information to facilitate the exchange of geographical data (STG 1998). The focus in this work is on another problem. Since each user interface has its own design where different symbols are chosen and different aspects of the data needs to be highlighted, there will be conflicts between the different map features. The data has to be generalised in a manner that is suitable for the purpose of this particular screen map.

Cartographic generalisation is a complex issue that will be further discussed in chapter 2. Intuitively it can be said that cartographic generalisation can be objectively defined to some extent and to a large extent is a matter of design. I have assumed that the design component can not be standardised. To achieve a map with good design that communicates relevant information to the user we need a cartographer. The aim of this work is to give the cartographer a tool to design user interfaces to geographical data. The cartographer specifies the screen map design and content. Furthermore the cartographer specifies from which data sets the user interface shall retrieve data and how the data shall be generalised. When the specification is done, the user interface automatically retrieves and generalises the data and the new screen map is created. If the result is unsatisfactory, details in the design such as symbolisation, choice of data to retrieve and how data is generalised can be modified to achieve a better result. The design of the screen map is an iterative process. When the design is finished, a screen map can be created automatically for all areas covered by the source databases. The databases can be continuously updated and the screen map is generated anew whenever there is a need.

The user interface is designed using the object oriented modelling method the Unified Modelling Language, UML. The intuitive approach in object oriented modelling of geographical information is to let different categories such as "*house"* or "*forest"* form different object classes. The member of an object class e.g. a house, correspond to the real world building, which has been surveyed as accurately as possible. The building object can be displayed differently in different screen maps defined at different scales and for different purposes. Using this approach and trying to automatically generalise and display the different object classes at different scales turned out to be very complex. This lead to the insight that geographical data models, are models of human concepts of the environment rather than models of reality. To be efficient the model ought to be designed in a manner that corresponds to human cognition. These thoughts lead into the study of human cognition, cartographic theory and theories in object oriented modelling and design. The parts of these studies that have impact on the design of the model are described in chapter two.

1.1 Thesis organisation

This work consists of three parts: a presentation of the theoretical background, the design of an object oriented model that incorporates automatic generalisation, and a case study where parts of the model have been implemented in an object oriented GIS. The theoretical background is given in chapter two and consists of a discussion about theories in cognitive science that has influenced the design of the model. The discussion is focused on categories, since categorisation is an important aspect in cartography as well as object oriented modelling of geographical information. It also contains a discussion about cartography, cartographic generalisation and different approaches that have been used to automate the generalisation process. The final part of chapter two contains a discussion about theories in object oriented modelling which have influenced the design of the model.

Chapter three contains a description and motivation of the different design choices that were made when the object oriented model was created.

Chapter four contains a detailed discussion about a model that creates different cartographic data sets by retrieving data from a cartographic data set defined at a scale of 1:10 000. The different data sets are displayed on the screen at the approximate scales of 1:50 000 and 1:100 000. Chapter four also presents results from the implementation of the model. The results are discussed in chapter five.

1.2 Terminology

In cartography, scale is defined as the relationship between distances on a map and distances in reality. In a geographical database, real world coordinates are used to describe the geometry of objects. The objects are visualised in a screen map with zooming capabilities. Thus the traditional concept of scale does not exist. However, the definition of the data model and the accuracy with which the geographical features are surveyed implies that the data set has a certain resolution. Since scale is a concept that is more familiar than resolution, it is used throughout this thesis to illustrate the resolution of a data set or a screen map.

Object and feature are two terms that can cause confusion. Throughout this thesis they have the following meaning:

Object - An object is always a database object, i.e. a chunk of information that exists in the database.

Feature - A feature is something that exists in the real world.

2 Theory

2.1 Human cognition

As has been described in the previous chapter this thesis presents a method to automatically generate screen maps at various scales from geographical databases. The screen maps are designed using object oriented modelling which will be described further in chapter 2.3. How the human mind conceives geographical information at various scales is an important aspect of how the model shall be designed and this chapter describes some research results from cognitive science that has influenced the design of the model. It is by no means a thorough description and the main influences for this chapter are from two books: George Lakoff's (1987)*, Women Fire and Dangerous Things: What Categories Reveal about the Mind,* and Alan MacEachren's (1995)*, How Maps Work.*

2.1.1 Categorisation

Categorisation is an important aspect in cartography as well as in object oriented modelling. Lakoff (1987, p. 6) states that: *Without the ability to categorise, we could not function at all*. MachEachren (1995, p151) takes this statement and formulates its equivalence in cartography: *Without categorisation, maps would not be possible.* How humans treat categories in general have implications on how categories in geographic information varies with scale. This have influenced how the object oriented model shall be designed.

Lakoff (1987) claims that there are two main views in category theory: the classical approach, which can be traced back to Aristotle, and the modern approach, called prototype theory. In the classical approach categories have the following characteristics:

- Categories are believed to exist independently of human beings. Since the categories already exist all we have to do is to discover and define them.

- Categories act as containers and a particular thing is either inside or outside a container.

- Things are assumed to be in the same category, if and only if they have certain properties in common.
- The properties, which the things have in common, define the category.
- All members of a category are considered to be equal members. There are no members that are better examples of the category than others.

As will be described in chapter 2.3 there are similarities between this view on categories and how object classes are defined in object oriented modelling. How the real world is abstracted into object classes is, however, a matter of design and does always depend on the application (Rumbaugh et al. 1992).

The classical view on categories was not questioned until the later work of Wittgenstein (1953). It was Elenore Rosch (1973, 1975, 1977, 1978; Rosch et al.1976) who made human categorisation a research issue. She was one of the primary forces behind the dramatic change in how we view human categorisation. George Lakoff calls the new theory that evolved prototype theory. In opposition to the classical view it takes the approach "…*that human categorisation is essentially a matter of both human experience and imagination*. Within this theory, categories have the following characteristics:

Family resemblance

A classical category has clear boundaries and is defined by common properties. Wittgenstein noted that the category game does not fit the classical mould since there are no common properties shared by all games. Some games involve mere amusement, without winning or loosing while others include an element of competition. Some involve luck, others skill, some involve both. The members of the category are united by what Wittgenstein calls family resemblance. Members of a family may resemble one another in various ways but there is no need for a single collection of properties shared by everyone in a family.

It is difficult to find a geographical category that is a typical example of family resemblance. However, the concept has interesting implications since it illustrates how the human mind can form categories from features that do not share a set of properties. A building that was used as farm until the 1960´s but is currently a dwelling house can still, in some sense, belong to the category farm, even though it does not fulfil the property that it shall be used for farming.

Extendable boundaries

Wittgenstein observed that there was no fixed boundary to the category game. The category could be extended and new kinds of games could be introduced if they resembled previous games in appropriate ways. When video games were introduced in the 1970s, for instance, the category game was extended to incorporate this new invention. Wittgenstein describes how the category number has evolved through history. First, the category number was taken to be integers and then it was gradually extended to rational numbers, real numbers, complex numbers, transfinite numbers and other kinds of numbers invented by mathematicians. One can, for some purpose, limit the category number to integers only, but the category number is not bounded in any natural way and it can be limited or extended depending on one's purposes. Wittgenstein's point is that different mathematicians give different definitions depending on their goals.

Categories in geographical information are extended as humans make new inventions. For instance, the category road evolved during the 20th century to incorporate highways.

Central and Non-central Members

In the classical theory, categories are uniform in the following respect. A category is defined by a collection of properties that the category members share. Thus no member should be more central than any other member. For the category numbers, however, it seems as if the integers are more central since every precise definition of numbers must include the integers, but not every definition must include the transfinite numbers. Rips (1975) shows an example where robins are considered to be more typical members of the category bird than ducks. During interviews, subjects inferred that if the robins on a certain island got a disease, then the ducks would, but not the converse.

Fuzzy categories

Some categories, like U.S. Senator, have crisp borders while other categories such as rich people or tall men are graded. The extent to which a certain feature is a member of the particular category depends on the context. Zadeh (1965) devised a form of set theory to model graded categories called fuzzy set theory.

In geographical information we can see that some categories, such as highway, are crisp, while the category forest is rather fuzzy. This will be elaborated on further in chapter 2.2 and in chapter 3.1

Conceptual embodiment

Conceptual embodiment is the idea, that human biological capacities and human experience of functioning in a physical and social environment, influences how categories are formed. Berlin and Kay (1969) describe how a language has a set of basic colour terms, like green, blue, red, etc. A basic colour term must consist of only one morpheme and the colour referred to by the term may not be contained within another colour. Some languages, like English, use eleven different basic colour categories, while other languages use as few as two. When speakers of different languages were asked to pick out the portion of the spectrum their colour terms refer to, no regularities appeared. But when they were asked to pick out the best examples of the basic colour terms, given a standardised chart of 320 small colour chips, virtually the same best examples are chosen for the basic colour terms by speakers in language after language. These best examples are called focal colours. If a language has a basic colour term that covers both green and blue, the best example of this colour term will not be turquoise but either focal green or focal blue. Kay and McDaniel (1978) were able to explain these results by studying the human visual system. The different receptor cells of the human eye interact in such a manner that the highest sensitivity is for the wavelengths that correspond to the different focal colours. This is the reason why humans conceive these as more primary.

If the idea of conceptual embodiment is applied to geographical information, we realise that the categories have been formed through several different experiences. In the case of the category forest it can consist of: interactions with the environment such as walking in different forests, studies in biology, and through looking at different maps that depict the forest.

Basic-level categorisation

Basic-level categorisation is a concept that further illustrates how human experience influences how categories are formed and organised. Lakoff refers to Brown (1958, 1965) and to Berlin et al. (1974), and describes how categories are organised, not only in a hierarchy from the most general to the most specific, but also so that the categories that are cognitively basic are "in the middle" of a general-to-specific hierarchy. An example of such a hierarchy is: vehicle - car - Mercedes, where car is cognitively basic. Generalisation proceeds "upward" from the basic level and specialisation proceeds "downward". According to MacEachren (1995) the basic level categories are basic in at least four respects:

- Perception – Basic level categories are the highest level categories having similar overall perceived shape, a single mental image and fast identification. Apple is a basic level category while fruit is a generalisation.

- Function – They are the highest level categories for which a person uses similar motor activities to interact with them (e.g. sitting on chairs vs. on furniture).

- Communication – Basic category labels are the shortest, most commonly used, and most contextually neutral words; they are the first learned by children; and they are the first to enter the lexicon.

- Knowledge organisation – The basic level is the level at which most of our knowledge is organised and for which the largest number of attributes is stored.

Basic level categorisation does not have direct implications on the modelling of geographical. It is included since it illuminates how human experience impacts on how categories are organised.

Multiple Representation

Classical categorisation assumes that there is always a single correct way to categorise any phenomenon. Prototype theory on the other hand presents a more flexible view that allows for multiple representations of individual concepts. An individual often holds more than one kind of representation of a concept suited to different applications. A cartographer, for example, may accept digital databases as being "maps" at a conceptual level, but, when looking in a bookstore for tourist maps for a trip abroad she would be rather disappointed finding a rack of CD's containing Digital Data Bank of the World files. MacEachren (1995) argues that: *“There is a need to explore the possibility of varying levels of categorisation for different goals, applications, and perspectives, and to explore how our maps might incorporate some of the less precisely defined (but no less truthful) ways of categorising the world.”*

2.1.2 Human pattern recognition

The human visual system is very efficient at recognising shapes and bringing up knowledge into the consciousness about what is seen from the “long time storage” in the brain. Consider for instance the case where we incidentally meet a friend from high school who we have not seen for ten years. We are usually able to recognise the person, which implies that there is some form of visual memory. The brain then immediately retrieves all kinds of information about: the school we went to, friends, teachers etc. This knowledge may not have been in the consciousness for years.

The human ability to recognise image patterns seems particularly interesting for the field of cartography and geographic information science. Map reading is to a large extent a matter of interpreting shapes displayed on a 2D surface to extract knowledge about the environment. According to MacEachren (1995) : “*human vision and visual cognition is incompletely understood”.* Marr (1982) presents a theory about vision, which is based on the idea that it is more important to understand what vision is for than understanding the neurophysiological mechanisms by which it works. Marr sees vision as an information processing task that has to be addressed at three different levels to be understood completely: the level of computational theory, the level of representation and algorithms, and the level of processing device or hardware implementation. The level of computational theory deals with what a process must do and why, along with a logical strategy by which the process might be carried out. The level of representation and algorithms deals with how the theory might be implemented while the level of processing device and hardware implementation considers how a particular implementation might be implemented in a particular device.

Vision, as an information processing system, begins with the image that is displayed on the retina in the eye, the retinal image, and ends with a three dimensional object centred model of reality that appears in the consciousness. Marr proposes two intermediate steps between the retinal image and the 3D model representation, see figure 2.1, for two reasons: One is the idea that information processing has to be addressed at different levels. The other is the evidence that mental representations of object shapes are stored in a different place of the brain than representations of use and purpose. From the retinal image a primal sketch is created that makes information in the retinal image explicit. The primal sketch is envisioned as an array of cells that contains "symbols" indicating the presence of edges, bars, blobs and so on, and their orientations.

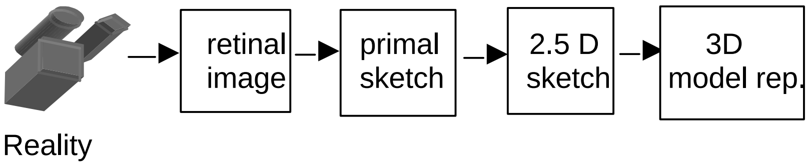

Figure 2.1 Marr's stages of vision. Derived from Marr (1982).

Pinker (1984) notes in a summary of Marrs theory that the features symbolised in the primal sketch are extracted separately for various scales. This allows major features to be distinguished from details and leads to a hierarchical model for storage of shape categories in memory, against which information from visual scenes is compared.

The next level of processing produces a 2.5 D sketch, a "*representation of properties of the visible surfaces in a viewer- centred coordinate system, such as surface orientation, distance from the viewer, discontinuities in these qualities; surface reflectance and some coarse* *description of the prevailing illumination*" (Marr, 1985, p. 125). Marr claims that so far the process is completely precognitive and has no input from the consciousness. Finally, the processing achieves the 3D model representation, which is what the consciousness is actually experiencing. The 3D model is object centred, and the objects that are seen are associated with all kinds of knowledge about what they are. This knowledge is retrieved from the brain and matched with what appears in the visual scene. The ability to recognise patterns and to associate knowledge with these patterns seems to be an unconscious activity in many cases. Through training it is possible to acquire an ability to recognise new patterns and as Schneider and Shiffrin (1977) point out, when someone repeatedly assigns particular visual patterns to specific categories, recognising these patterns becomes automatic (i.e., preconscious). A simple example comes from the popular Swedish habit to pick mushrooms in the autumn. Since some species of mushroom are poisonous, it is important to be able to recognise the different species, and to a novice this might seem rather difficult. But, if someone who is knowledgeable about mushroom points out the differences between different species to us, we gradually acquire the ability to recognise the edible mushrooms. Gradually that ability becomes automatic. Since some mushroom are poisonous, it is considered among mushroom pickers that it is not satisfactory to read about the characteristics of different species in a book. You have to be shown several different real samples to be able to acquire the knowledge to recognise the edible species. This example could be extended with similar discussions on how we learn to recognise different species of e.g. trees, flowers or dogs. Chase and Simon (1973) report that chess experts are able to organise information about the arrangement of the chess board into larger chunks, which gives them the ability to assess a particular arrangement more quickly than novices. This ability does only exist if the arrangement represents a likely arrangement of the chessboard and it seems as if the experts have developed the ability to recognise a set of different patterns for how a chessboard might be organised. Novices who do not have this ability have to process the visual scene at a more local level.

How knowledge is organised and stored in the human brain is, like vision, incompletely understood. Categorisation is an important aspect of how knowledge is organised and has been described above. Another approach is given through the different theories on how knowledge is organised into representations or knowledge schemata. MacEachren

(1995) describes a theory developed by Rumelhart and Norman (1985) that organises knowledge into three types: propositional, image and procedural. The propositional knowledge concerns declarative knowledge or knowledge about objects, attributes and places. Image knowledge seems suited to represent configurational knowledge, i.e. knowledge of patterns, and of spatial relationships among entities in space. Procedural knowledge concerns e.g. knowledge about the sequence of steps to get from one place to another. Different schemata can be embedded, one within another, representing knowledge at all levels of abstraction.

The different knowledge schemata can be elaborated on in much greater detail, but here we only discuss research by Golledge et al. (1992). They have noted that procedural knowledge obtained during route learning is difficult to transform into a configurational (image) representation. MachEachren (1995) draws the conclusion that it is equally difficult to transform knowledge in the opposite direction, from image to procedural. I believe that it is equally difficult to transform knowledge that is stored in an image form into propositional knowledge and the reverse. The mushroom picking example above illustrates this and another example can be constructed for different breeds of dogs. I have a detailed knowledge of what an alsatian dog looks like, since I'm able to immediately recognise an alsatian dog when I meet one in the street. This recognition process is fully automatic. It is, however, very difficult for me to describe an alsatian dog using propsitional knowledge, words, with such detail that someone who is not familiar with this particular breed should be able to recognise it. How this ability to recognise patterns influence modelling of geographical information will be elaborated on further in chapter 2.2.

2.2 Cartographic aspects

2.2.1 Categories in cartography

Robinson et al. (1995, pp10) describe the basic characteristics of a map:

- All maps are concerned with two elements of reality: locations and attributes where the attributes contain information about qualities and magnitudes.

- All maps are reductions and a map is smaller than the region it portrays.

- All maps involve geometrical transformations through a map projection

- All maps are abstractions of reality in such a way that maps only portray the information that has been chosen to fit the use of the map.

- All maps use signs to stands for elements of reality. These signs consist of various marks such as lines, dots, colours, tones, patterns, and so on.

This thesis focuses on how reality is abstracted into maps, and how the same reality is abstracted differently in different maps, even though the categories in the different maps have the same name.

In chapter 2.1.1 it has been stated that categorisation is fundamental in the map making process. Some characteristics of how humans form categories have also been described, such as fuzzy borders, conceptual embodiment and multiple representation. If this view on categorisation is adopted rather than the classical approach it has impacts on how the abstraction process can be seen. A map category, such as *forest* or *built-up area*, has been formed in the human mind through experience. For the category forest, this might consist of experience from walking in different forests, studies in ecology and physical geography, experience from reading various maps etc. The category forest is fuzzy and acquires different meanings in different contexts. For example: we visit Stockholm as tourists and ask the question: Are there any nice forests in Stockholm? A forest that comes to mind then is Liljansskogen, which is located close to KTH not far from the centre of Stockholm. A tourist can also ask the question: Are there any nice forests in Sweden? What comes to mind then are the national parks and other large forest areas that have not been logged. Here the category forest acquires a meaning in which Liljanskogen is barely a member. In this context Liljansskogen is more like a park. The reason for this change in meaning of the category is that the context has changed. In the first case the context is a tourist asking for a forest close to an urban centre. We know from experience that such a forest is kept for recreational purposes; a more famous example is the "bois de bologne" in Paris. Such a forest may contain man made lakes and mowed lawns suitable for playing frisbee or softball as well as areas covered with trees and shrubs. A nice forest in a country like Sweden, which is mainly covered with forest, is a different context. I think of beautiful nature untouched by man suitable for hiking.

It seems as if the category forest acquires a more narrow and specific definition when it is used in a particular context. In a similar manner a map provides the context for the categories that are members of the map. The categories acquire a more specific definition in the map that is suitable for the map purpose. In a tourist map of Stockholm the category forest is delineated differently than in a map of Sarek, a national park in northern Sweden. The meaning of a category like forest is different in different maps. It is perhaps not possible to define a general category forest that contains all the concepts humans have about what a forest is.

In chapter 2.1.2 the human ability to learn how to automatically recognise different patterns is described. The difficulties in transforming this “visual” knowledge into a propositional or procedural from is also elaborated on. It seems reasonable to assume, that cartographers and experienced map readers have developed such “visual” knowledge, and that it is used when interpreting different maps. The meaning a map category acquires in a particular map context is influenced by this visual knowledge, and the visual knowledge gives the category a more narrow definition. These ideas are speculative and more research is needed to prove their validity. Nevertheless, the idea will be illustrated with an example since it has influenced the approach to object oriented modelling of geographical information taken in this thesis.

When reasoning about the category built-up area without looking at a map it might seem reasonable to believe that the category has approximately the same meaning even though the scale changes from 1:50 000, 1:100 000 and 1:250 000. My impression is that the three different maps in Figure 2.2 through 2.4 show that the change in meaning of the different categories is larger than expected.

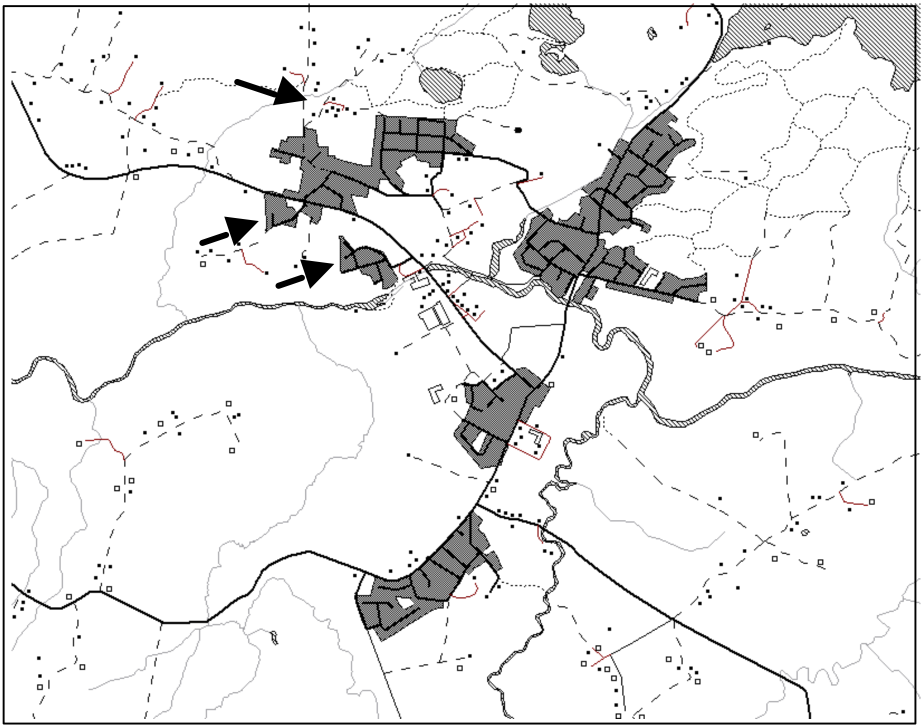

Figure 2.2 Swedish topographical map at a scale of 1:50 000. The original cartographic data in Figures 2.2-2.4 is provided by the National Land Survey of Sweden. (Copyright Lantmäteriverket 1998, Dnr: L2000/ 1415.)

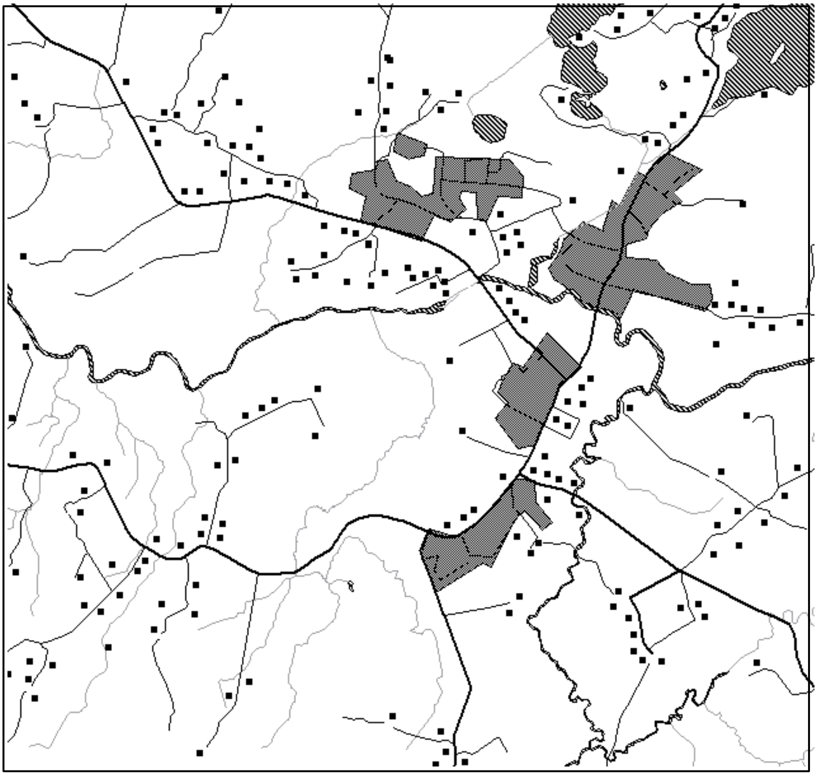

Figure 2.3 Swedish topographical map at a scale of 1:100 000

Figure 2.4 Swedish topographical map at a scale of 1:250 000

At a scale of 1:50 000 the built-up areas consist of several different small areas. It seems reasonable to assume that the cartographer considers a built-up area to be formed by a dense pattern of houses, streets, parking lots, etc. Through experience, the cartographer has developed a visual knowledge, that gives him the ability to recognise such patterns in larger scale maps or aerial photographs and classify these as built-up areas. This implies that there is a risk that different cartographers might develop different visual patterns to recognise the built-up areas. The same map series might then contain different concept of what a built-up area is since there are several cartographers working on different sheets of the map series. At the National Land Survey of Sweden, NLS, the cartographers co-operate and discuss difficult cases to avoid such differences. Through these discussions a consensus is developed about what the different categories in this map series mean. Another issue is, if the definition of built-up area is consistent over the map surface, i.e. does built-up area have the same meaning in densely populated regions as in sparsely populated? To be able to answer such a question we need a strict definition of what a built-up area is. Statistics Sweden gives such a definition, where built-area is defined as a cluster of buildings. Neighbouring house entities in the cluster should be located within a certain distance (e.g. 200m), and the cluster should contain a certain number of residents (e.g. 200 people) (Statistics Sweden 2000, Nordbeck 1969). But if the built-up area is defined through the cartographers ability to recognise visual patterns this question seems impossible to answer. To find an answer we would have to transform the visual knowledge of the cartographer into procedural form so that we can give the category a strict definition. As has been argued in chapter 2.1.2 this transformation is very difficult to perform.

At the scale of 1:100 000 the visual pattern that defines what a built-up area is, seems to have changed. If the definition of built-up area would be the same as in the previous map, but generalised to be readable at the smaller scale, the cartographer would: delete built-up areas that are to small to be seen at this scale, delete small islands within the built-up area which will not be visible, and simplify the outline of the built-up areas. In the map above other things have happened. Two of he built-up areas that are marked in Figure 2.2 are no longer considered to be built-up areas at the scale of 1:100 000, even though it is quite clear that they are large enough to be displayed at this scale. Instead, a group of individual buildings are portrayed to give the map-reader an impression of the area. In the 1:50 000 scale map a group of buildings is also marked. This group of buildings is represented as a built-up area in the 1:100 000 scale map.

At the scale of 1:250 000 all built-up areas in the map are aggregated into one big object. It might be possible to argue that the only motivation for this generalisation is to make the map readable while, the concept of built-up area is essentially the same as at the larger scale. However it is also quite clear that the landscape pattern that is interpreted into this feature has to be quite different from the patterns used at the two larger scales.

Kilpeläinen (1997) describes a multiple representation database, where objects in the database that represent the same features in reality, are connected to facilitate updates and analysis of the information. Figure 2.5

illustrates how different building objects in the database that represent the same real world building are connected.

Represen-tation levels	**Geogra-phic meaning**	**Reaso-ning processes**	**Cartographic representation**	**Geometric represen-tation**
Represen-tation Level 4	Built-up area	Aggregate buildings from level 3		Polygon
Represen-tation Level 3	Building	Replace the centerpoint of a building at level 2 by a point symbol		Point
Represen-tation level 2	Building	Simplify the outline of a building at level 1		Simple polygon
Represen-tation level 1	Building	Use the base level represen-tation		Complex polygon

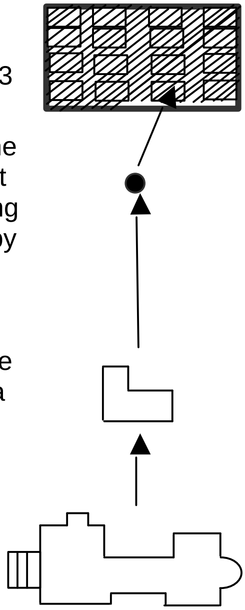

Figure 2.5 Representation levels for a building feature. (Redrawn from Kilpeläinen 1997, p.57)

Buildings are represented as individual entities at the larger scales and are aggregated into built-up areas in the smaller scale. Figure 2.6 shows three different cases that illustrate difficulties with how groups of buildings are aggregated into built-up areas.

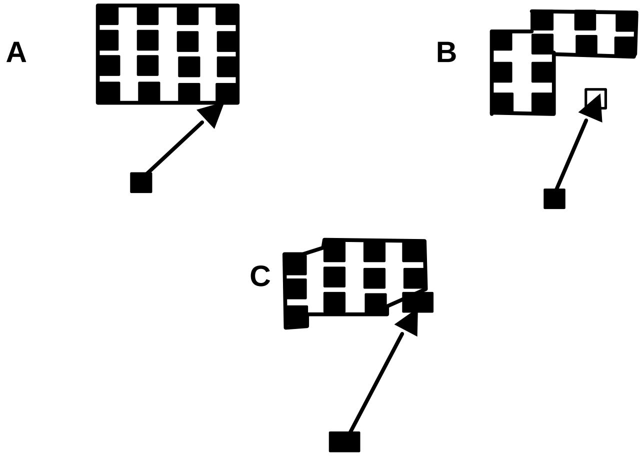

Figure 2.6 Different aggregation cases of buildings and built-up areas. (Redrawn from Kilpeläinen 1997, p.58)

Case A is uncomplicated, since the building is clearly a member of the built-up area and connectivity between the building and the built-up area is easy to implement. Case B is much more complicated, since it is uncertain whether this particular building is a member of the built-up area. It might be that the building is a part of the built-up area but the outline of the built-up area is simplified so that the building is outside the built-up area. It might also be that the building is not part of the built-up area but is not selected for display in this level. In case C the building is partly outside the built-up area since the outline of the built-up area has been generalised. Kilpeläinen argues that the building and the built-up-area must be connected in all three cases above to facilitate updates.

2.2.2 Human - map interaction

MacEachren (1995) describes how different research paradigms in cartography have evolved since the Second World War. During the war several U. S. geographers supported the military in their map making efforts and the emphasis of the discipline shifted from the artistic to the functional. The military need maps that communicate an unambiguous view of reality. Robinson was one of the principal players in the government's cartographic efforts during World War II. In his dissertation (Robinson, 1952) he argues that treating maps as art can lead to "*arbitrary and capricious*" decisions. According to MacEacren (1995)

Robinson “*saw only two alternatives: either standardise everything so that no confusion can result about the meaning of symbols, or study and analyse characteristics of perception as they apply to maps so that symbolisation and design decisions can be based on “objective*” *rules.”* Robinson advocated the second option, which was also taken by most academic cartographers.

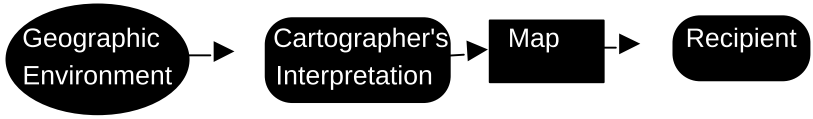

Figure 2.7 A schematic depiction of cartography as a process of communication. (Redrawn from MacEachren 1995, p 4)

Robinson’s dissertation pointed in a direction which, towards the end of the 1960s, was formulated as the cartographic communication paradigm, see figure 2.7. Different models to illustrate this paradigm were presented by, e.g., Board (1967) and Kolácný (1969). The cartographic communication paradigm claims that cartography is about communicating geographical knowledge. The knowledge about the geographic environment exists and is utilised by the cartographer to design a map. The knowledge portrayed in the map is acquired by a user through map reading. At each stage in this process there is a risk that knowledge might be lost and efforts have been made to measure this information loss.

MacEachren (1995) points out that the communication paradigm might be valid for certain kinds of maps, such as maps used by air traffic controllers or pilots. However, a great variety of maps have no predetermined message. The knowledge that can be retrieved from such maps depends on the previous experience and training of the map reader. It might be, that the cartographer who makes a large scale topographic map, can not retrieve the same knowledge from this map as an architect who uses the map for urban planning. Based on these thoughts and discussions about the role of art in cartography MacEachren (1995) states that. “*The map is examined here, then, not as a communication vehicle, but as one of many potential representations of phenomena in space that a user may draw upon as a source of information or as an aid to decision making and behaviour in space*”. This view on cartography is the one adopted here. In chapters 1 and 2.2.3 the role of the cartographer in the proposed system is discussed.

2.2.3 Generalisation of geographic information

MacEachren (1995, pp12) states: “*My position is that there is no single correct scientific, or non-scientific, approach to how maps work*”. I believe a similar statement could be made for generalisation of geographic information. Several different theoretical approaches can be found in the literature, which provide frameworks for different research efforts. This chapter provides a description of some of these frameworks and a discussion on how they are applicable to the approach taken in this thesis.

<u>Model and graphic generalisation</u>

A widely accepted approach among researchers in automatic generalisation is that generalisation of geographic information can be seen from two different perspectives: model oriented generalisation and graphic generalisation. Weibel (1995) argues: *“…there is a consensus in the research community that, apart from graphics oriented generalisation, there is also a need for model-oriented generalisation*”. The two concepts are described by Müller et al. (1995). Model generalisation is seen as the transformation of data between geographical data models defined at different spatial and semantic resolution. These transformations can be performed independently of the graphic representation. Model generalisation can be performed to facilitate data access in GIS and is also driven by analytical queries such as: What is the spatial average? Graphic generalisation can be viewed as transformation of objects in a graphic representation of spatial information, intended to improve data legibility and understanding. An example of graphic generalisation is the displacement of overlapping symbols. Müller et al. also suggest that model oriented generalisation can be precursor to graphic generalisation.

This thesis is concerned with how to create a user interface to a geographical database. This user interface is described in chapter 3.2. It can be used to perform different analyses using different functions and operators as well as to obtain information by looking at the screen map. The process that extracts, generalises, and inserts data into the data model of the user interface, is treated as a one-step process and not divided into model and graphic generalisation. The main reason for this is the difficulty to explicitly define where the model generalisation part of the process ends and graphic generalisation begins. As has been argued above, it is believed that visual knowledge has an important impact on how several of the categories or object classes in a particular map are defined. If the generalisation process is to be divided into model and graphic generalisation, the data model that defines the result after the model generalisation has to be defined neglecting the visual knowledge. The visual knowledge is only utilised during the graphic generalisation. However, the visual knowledge has impacts on how the model generalisation shall be performed. To divide the process into model and graphic generalisation introduces additional complexity to the problem. Kilpeläinen (1997) has tried to divide the generalisation process for topographic data into two steps and notes that the distinction between model and cartographic generalisation is not always clear.

The approach taken in this thesis is that the creation of the user interface is a matter of creating a view of the database that is optimal to the user. Some user interfaces are mainly used for visual analysis, others for analytical queries; but in the majority of cases, a user interface will include some categories suitable for both visual analysis and analytical queries, and some categories that only portray visual information. A typical example is a map displayed in a car for vehicle navigation, where the road network should be structured in such a manner that routes and distances can be computed. Lakes and streams, on the other hand, are shown mainly to give the driver an impression of the landscape, and are not used for analytical queries in this context. When the user interface is created compromises have to be made between different object classes and these compromises might involve both model and graphic generalisation, which makes it impractical to divide the process into two steps.

Conceptual Frameworks

There are several different conceptual frameworks that describe how an automatic generalisation system can be organised. A major issue seems to be the ordering of different actions. There is no consensus in the research community which framework to use (Harrie, 1998), and perhaps different frameworks are suitable in different contexts.

An often cited framework is the one proposed by Brassel and Weibel (1988). It is deterministic and consists of five steps:

- Structure recognition – This phase is an analysis of the source database based on the objectives of the target database. It aims "*…at the identification of objects or aggregates, their spatial relations and the establishment of measures of relative importance*".

- Process recognition – Based on the results of the structure recognition and the objectives of the target database the processes that will lead to the target database are recognised.

- Process modelling – This can be seen as a compilation of rules and procedures from a process library based on the structure and process recognition.

- Process execution – This step is the actual processing of the data where the target database is generated.

- Data display – This step converts the target database into a target map and is perhaps not part of the actual generalisation process.

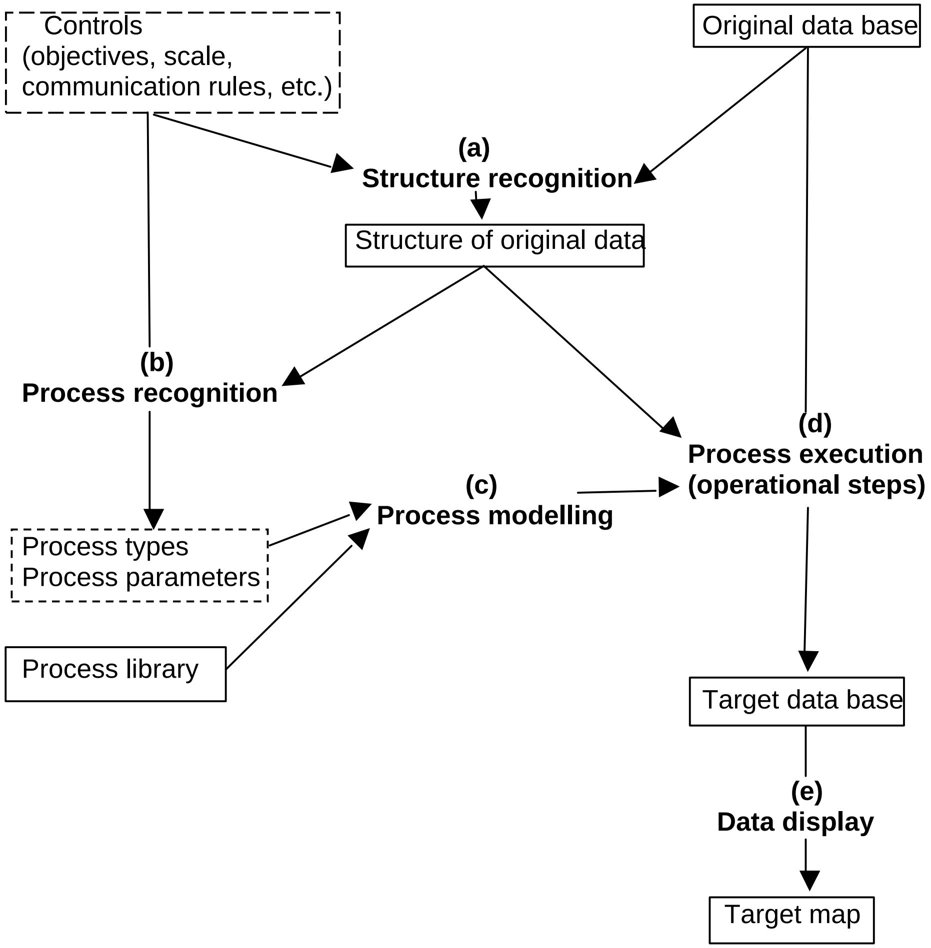

Figure 2.8 The Brassel and Weibel conceptual framework (Redrawn from Brassel and Weibel 1988, p.231)

Another framework is given by Mackaness (1995a), who argues that map design is a highly interactive process and suggests that: "…*we start with some hazy thumbnail sketch of what we want, we then source the data (in terms of its geographical extent and intended theme), apply some set of generalisation operators, view the result and repeat and refine subsequent application of generalisation operators in a cycle until a satisfactory solution is found*". In this approach the map designer starts with selecting the data to be displayed in the map and the map scale. The map designer then works his way through the different sections of the map applying different generalisation algorithms to different objects and groups of objects. To support the user Mackaness argues that different tools should be supplied to guide the map designer such as:

- An isoline map showing the density of the initially selected objects. If the density is too high in some area the user can choose to select a smaller set of objects.

- Thermometers that show to what extent different generalisation operators have been applied on the general level as well as for individual objects.

- Depending on the actions performed by the user he is informed about non-binding constraints that guides him to the next suitable operations. The user can navigate back and forth in the design process and choose different alternatives, which give different constraints further on in the design process.

Richardsson and Muller (1991) discuss how procedural methods and rule-based heuristics are applied in generalisation to handle conditional statements (IF-THEN statements). In procedural methods the sequence of statements must be executed in a predetermined order. In rule-based heuristics the conditional statements relate to symbolic matching, the rules may appear in a random order, and a search strategy for a solution may not always follow the same order. Richardsson and Muller (1991) argue that in cartographic generalisation the procedural and rule-based solutions are not mutually exclusive. *"A rule may call for another rule which in turn calls for a procedural routine. Conversely, a procedure may lead to a question that must be resolved by applying a rule-based strategy."*

Ruas and Plazanet (1996) propose a framework that is based on the one proposed by Brassel and Weibel (1988), but where the ideas of Mackaness (1995a) are incorporated. The cartographer has a similar role as in the framework proposed by Mackaness, who starts by selecting the data and symbolisation of the map, performs simple generalisation for the whole map, and then moves on to solve local generalisation problems. In the Ruas and Plazanet framework the generalisation process is described in The Global Master Plan. This is the deterministic part of the framework. It provides a general outline of the different tasks to be performed. It is noted that the initial tasks of the global master plan are well defined but when the generalisation needs to be performed for groups of objects in a certain area it becomes difficult to choose the objects and procedures. Generalisation in a certain area is called a situation. Ruas and Plazanet discuss how different constraints can be put on the resulting data, which guide the generalisation process. Constraints can be such things as a minimum space between objects, preservation of object shapes, maximum authorised displacement etc. The constraints do not impose an action but act as a guide to the cartographer.

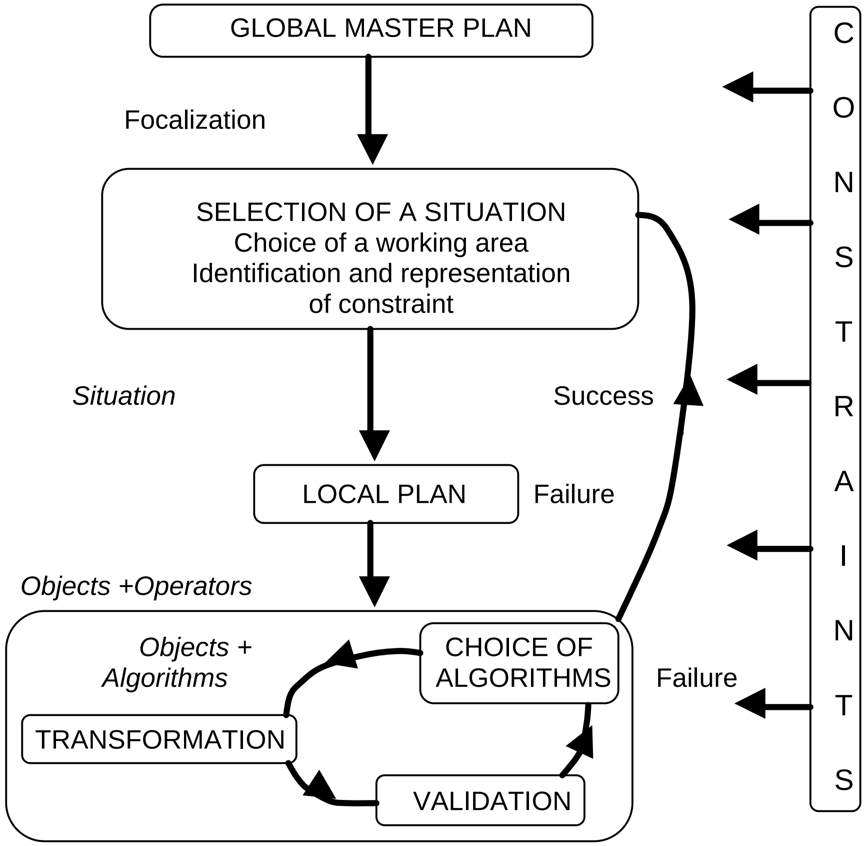

Figure 2.9 The Ruas and Plazanet framework. (Redrawn from Ruas and Plazanet 1996, p. 327)

The framework proposed in this thesis is based on the ideas described in the frameworks above but takes a slightly different approach: The generalisation process has to be fully automatic. This is currently impossible for topographical maps but, as has been described in chapter one, there is a growing need to be able to automatically generate simple single-purpose maps for different purposes such as the Internet. Another difference is that a clear-cut distinction is made between the source and target data-model. As will be described in more detail in chapter 3.3.3 the objects in the source data model and the target data model are considered to represent different aspects of reality, even though the object classes have the same names. A *built-up area* in a certain scale has a different meaning from a *built-up area* in a smaller scale, and in many cases it is impossible to define links between individual objects in the different scales.

The main similarities to the frameworks described above are: the idea that map design and generalisation is an interactive process, and the need for constraints to guide the generalisation process. The idea in this approach is that the cartographer shall be able to interactively choose: the data to be created, the symbolisation, and the generalisation parameters. These different choices can be modified until an acceptable solution is found. The constraints are expressed in the object classes of the target data model as will be described further in chapter 4.

2.3 Object oriented modelling

Object oriented modelling is now the dominating approach to design and development of new software. Several books have been written on the subject, but the main influences for this work are from: "*Object-Oriented Modeling and Design*" by Rumbaugh et al. (1991) and "*The Unified Modeling Language User Guide*" by Booch et al. (1999). This chapter describes some of the concepts in object oriented modelling that are of importance to the approach presented in this thesis. The modelling language that has been used is the Unified Modeling Language, UML.

Abstraction is a central concept in any kind of modelling. Rumbaugh et al. (1991) present the following discussion about the nature of abstraction within object oriented modelling:

" *Abstraction is the selective examination of certain aspects of a problem. The goal of abstraction is to isolate those aspects that are important for some purpose and suppress those aspects that are unimportant. Abstraction must always be for some purpose, because the purpose determines what is and is not important. Many different abstractions of the same thing are possible, depending on the purpose for which they are made.*
All abstractions are incomplete and inaccurate. Reality is a seamless web. Anything we say about it, any description of it, is an abridgement. All human words and language are abstractions - incomplete descriptions of the real world. This does not destroy their usefulness. The purpose of an abstraction is to limit the universe so we can do things. In building models, therefore, you must not search for absolute truth but for

adequacy for some purpose. There is no single "correct" model of a situation, only adequate and inadequate ones. A good model captures the crucial aspects of a problem and omits the others. Most computer languages, for example, are poor vehicles for modelling algorithms because they force the specification of implementation details that are irrelevant to the algorithm. A model that contains extraneous detail unnecessarily limits your choice of design decisions and diverts attention from the real issues."

This statement has had great impact on how the model presented in this thesis has been designed. The attempt has been to focus on the purpose, only to introduce concepts that are relevant for the task and keep it as simple as possible.

Booch et al. (1999) discuss different principles of modelling and state that " *It's best to have models that have a clear connection to reality, and where connection is weak, to know exactly how those models are divorced from the real world. All models simplify reality; the trick is to be sure that your simplification don't mask any important details.*" A discussion about how different geographical object classes in a GIS correspond to real world features will be given in chapter 3.1.

The most important building block in an object oriented system is the <u>object class.</u> An object class describes a set of <u>objects</u> with similar properties (attributes), common behaviour (operations), common relations to other objects and common semantics. If two objects are members of the same class depends entirely on the purpose. A horse and a barn may be members of the same class if they are viewed as financial assets only. If we take into consideration that a person feeds the horse and paints the barn they belong to separate classes. A well-structured class has crisp boundaries in the sense that there should be no ambiguity when determining which object class an individual feature belongs to. It is interesting to note the similarities between the definition of an object class and the different theories about categories described above. The definition of an object class seems to be very similar to the classical approach to categories. An object class has crisp boundaries and the members of an object class have certain properties in common. The discussion about abstraction above, on the other hand, seems to be very much in line with the prototype theory in the sense that categories and object classes are something that does not exist in the real world. Figure 2.10 shows the representation of an object class named *Bl_Church* that contains churches stored as symbols. The class has an attribute called *angle* which holds a value describing the turning angle of an individual symbol. *create(), select()* and *simplify()* are different methods that belong to the class and will be described further in chapter 4.

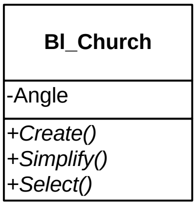

Figure 2.10 An object class in UML.

A link is a connection between objects, e.g. a building belongs to a certain parcel. Links are modelled as relationships between different object classes. In UML there are three main relationships: dependencies, generalisations and associations. A dependency is a relationship that states that a change in a specification of one thing may effect another thing that uses it. It is possible to introduce different flavours to the meaning of a dependency using stereotypes. This possibility has been used in the model presented below, where dependencies have been used to illustrate topological relations between different object classes. An example of this is given in Figure 2.11. A generalisation is a relation between a general object class (the parent) and a more specific object class (the child). Generalisation means that objects of the child may be used anywhere the parent may appear, but not the reverse. A child inherits the attributes and operations of the parent. An example of generalisation is given in Figure 2.12. An association describes a group of links with common structure and common semantics e.g. a person works for a specific company. Figure 2.13 shows an example of an association. Associations are inherently bi-directional. Associations have a crisp definition, while relations between individual objects in geographical data can be rather fuzzy. To define links between a *built-up area* and individual *house* objects is an example of an association that has been discussed in chapter 2.2.1. Whether links shall be formed between individual objects has to be decided within the application context.

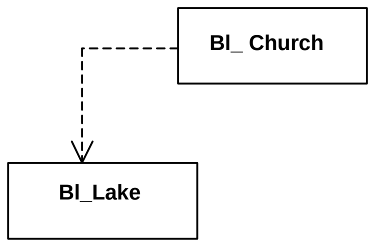

Figure 2.11 A dependency showing that church objects might be effected if lake objects are moved.

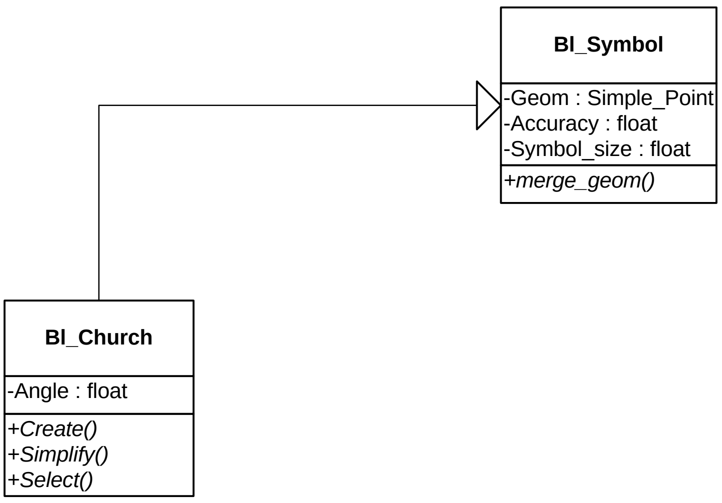

Figure 2.12 A generalisation showing how the church object class inherits from the symbol object class.

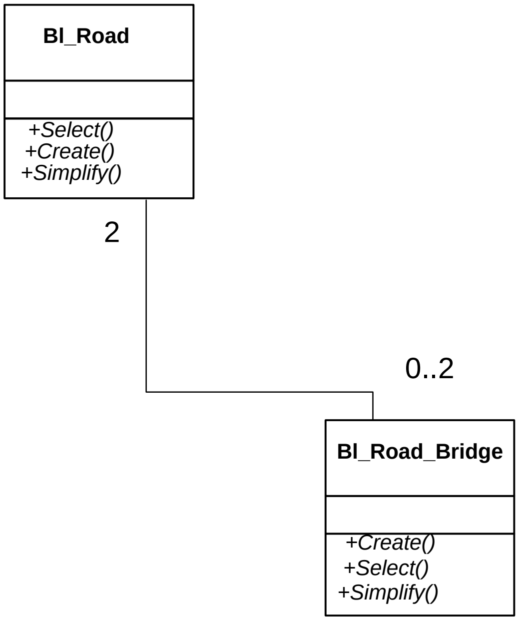

Figure 2.13 An example of an association showing how a road bridge must always be linked to a road

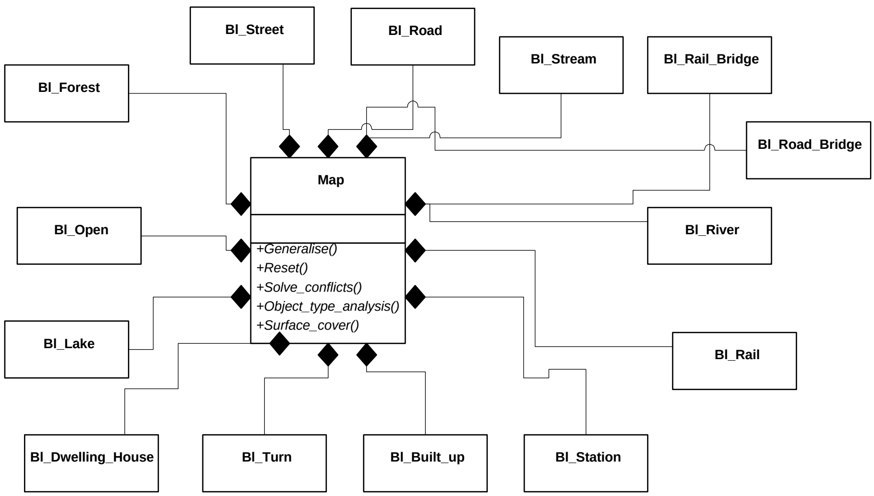

Figure 2.14 A map as an aggregation of set of different object classes representing geographical features.

Using <u>multiplicity</u> it is possible to model how many objects may be connected across an instance of an association. In Figure 2.13 a road bridge must always be connected to two road segments, while a road segment may be connected to 0, 1 or 2 road bridges. In Figure 2.14 the diamond shape symbol illustrates a special form of an association called an aggregation. An aggregation is a “whole/part” relationship that shows how a larger thing (“the whole”) consists of smaller things ("the parts”).

UML uses diagrams to visualise the design of an information system from different perspectives. A diagram only contains the things and relations, that are relevant to a specific view, all else is suppressed. An object class, for instance, may be shown with a box containing only its name in one diagram while another diagram contains all its methods and attributes. UML has nine kinds of diagrams, but the model presented in this thesis only uses two: the class diagram and the sequence diagram. A class diagram shows a set of classes and their relationships and illustrates the static design view of a system. A sequence diagram shows the dynamic view of a system with the emphasis on the time ordering of interactions between different objects.

3 An approach to map design by means of object oriented modelling

3.1 Object oriented modelling of geographical Information

A number of books and papers have been published on object oriented modelling of geographical information, see for instance Worboys (1995) or Molenaar (1998). Molenaar describes four different levels of modelling and states that these levels of modelling have evolved since the way most users of GIS understand information is rather remote from the way the information is handled by the computer. The different levels of modelling have evolved, to give users and system developers a comprehensible tool to reason and express their view on geographical information in such a manner that it can later be transformed into machine code. The four different levels are:

- Physical data modelling – This level concerns how data should be organised into bits, bytes, records and pages; structures that a machine can handle.

- Logical data modelling – This is the level of the database models. One of the best known models is the relational model described by Date (1990). The object oriented approaches are gaining popularity and is the main focus of this work, see Rumbaugh et al. (1992) and Booch et. al. (1999)

- Conceptual data modelling – concerns which terrain features should be represented in the logical data model, which thematic description they should have and how they should be represented geometrically.

- Spatial data modelling – Spatial data modelling concerns modelling within a certain discipline, such as soil mapping or vegetation mapping.

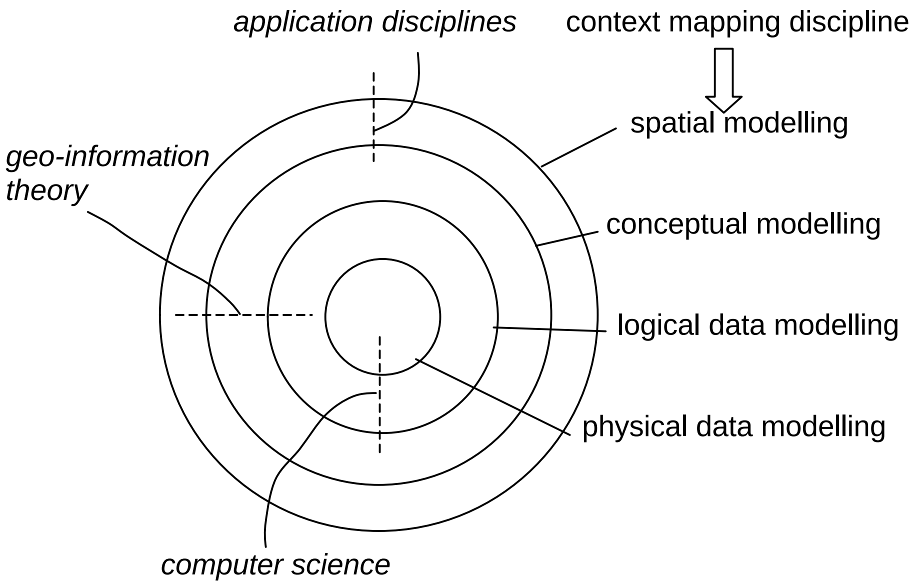

Figure 3.1 Different models according to Molenaar (1998).

The ideas presented in this work concern the levels of conceptual and logical data modelling. The main focus is on how geographical reality can be abstracted into an object oriented data model. The approach is based on the cognitive aspect of how humans reason about geographical information rather than from a specific application discipline.

The following assumptions form the basis for how the object oriented model presented in this thesis has been constructed:

- Several categories used in geographical information are rather fuzzy. Examples that illustrate this are given below. Since the categories are usually transformed into object classes in a GIS, there is a need to be able to handle fuzzy as well as crisp categories in a GIS.

- The categories have different meaning in different contexts. The context is set by the application discipline and the application within the GIS.

- The human pattern recognition ability has impacts on the meaning a particular category in a map acquires. The category in a map has a much narrower definition than when a category is discussed. It is the pattern the category forms in the map that influences the meaning that is given to the category.

- A map that is a view to a database should have the possibility to contain object classes suitable for analytical queries as well as object classes that are only used to visualise information in the map.

- Categories with the same name displayed in different maps, are defined within different contexts and thus have more or less different meaning. The categories are most likely overlapping, but whether associations between individual members of the two categories can be explicitly defined has to be decided in each individual case.

- The map presents a view of reality for a particular application. The meaning of the categories included in the map and the design of the map is an optimal compromise to convey the information required by the user.

Worboys (1995) gives the impression that object oriented modelling is focused on modelling individual objects which can later be organised into object classes. This might be true in other applications of object oriented modelling, such as designing software for an automatic teller machine. There is only one automatic teller machine, but there is a class of customers. When modelling geographical information, the focus is on the object classes rather than the individual objects. It is very rare that an individual geographical feature is modelled explicitly. This might seem a minor distinction. But I believe this is the reason why the discussion about the vagueness of individual objects in an object class is often neglected. A building is a well-defined feature with individual attribute values such as owner. Most people can agree on what an individual building is. A forest object on the other hand is geometrically represented by a surface. This surface can contain small open areas, which can be mapped as open areas or as part of the forest. How this is done depends on the application and the cartographer. If a group of people were asked to define the individual forest objects we would have an endless discussion. The main reason is probably that we are not interested in the individual forest objects. A forest object has individual attributes such as area. However, we are not interested in knowing the area of individual forest objects but rather how much land is covered by forest in a certain region. The forest is divided into individual objects because the object oriented model requires it. This example is valid for topographical mapping, in forestry applications, there is a need to handle individual forests objects. The extents of these objects are defined to suit the application and the objects have individual attributes, such as age and site quality class, that are relevant to the user. As Booch et al. (1999) point out, a successful model should have a good connection to reality, and it is important to know where there are flaws in this connection. Object classes that contain objects that do not have an easily comprehensible definition are such flaws and should be highlighted.

Based on these ideas all object classes defined in this model belong to a map. This implies that each object class is defined at a certain scale. Below we discuss the characteristics of different geographical categories as the map scale varies. The discussion is, to some extent, based on experiences from building and producing databases for the Swedish National Fortification Administration. As has been emphasised above, context is very important when defining object classes. In the discussion we illustrate how the meaning of a category can change with scale and context.

Road

A road that has been built by the National Road Administration is rather well defined, since the National Road Administration only builds a limited number of different road types. If we include all kinds of roads the division between different road types becomes fuzzier.

At a large scale (e.g. 1:1000) road is modelled in a manner that suits the needs for maintenance and construction purposes. If new buildings are to be constructed close to a road we need to know the exact location of the road. For maintenance purposes we want to know such things as the surface type of the road and when the road was last paved. Roads in a community form a continuous surface framed by distinct borders. This continuous surface has to be split into segments in some way, since it is impractical to have a single road object in a municipal geographical database. It seems reasonable to split the road object at crossings, but how each crossing shall be split is difficult to define and left to the surveyor, see Figure 3.2. As has been discussed with regard to the forest objects above, we are not interested in the individual road objects but rather such things as how much paved road there is in a certain area. The role of a particular road segment in the transportation network is hardly considered at this scale.

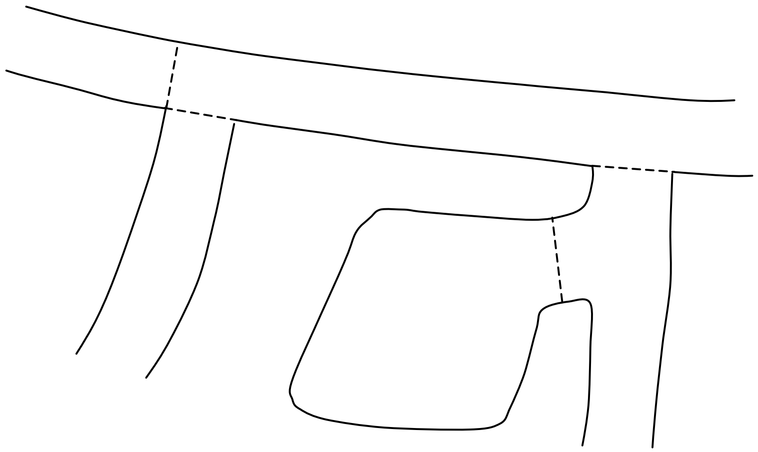

Figure 3.2 Examples of roads and a parking space treated as surfaces in a large scale database. The dotted lines show where the paved surface has been split up into objects.

As we move to a scale of 1:10 000 the road is mapped as a line. Maps at this scale can be used in the early phases when new constructions are planned so accuracy is still very important. Maps at this scale can be used for navigation. To facilitate route planning a node-link structure can be formed. When this is done complex traffic junctions are represented as nodes. Such simplifications can not be accepted if the map is to be used for construction planning.

At the scale of 1:100 000 the roads are used for e.g. navigation, route planning and to convey important aspects of the landscape. This implies that the accuracy as well as the need to portray all small paved areas that lead to parking lots etc is much less important. Roads at this scale can be used in two different perspectives: the local perspective where we sit in a car and try to navigate with a map, and the global perspective where we look at a map to see the characteristics of a road. Plazanet et al. (1995) describe how French cartographers generalise roads in mountainous areas. The first and last bends are maintained while other bends can be omitted or exaggerated. The first and last bends are kept to give a correct view when navigating using the map. When we turn into a new road we get confused if the first bend is in a different direction from what is shown in the map, but if one of several bends is omitted on a road segment this is hardly noticed by the driver. From the global perspective it is important to see that the road is winding even if some bends are neglected and others are exaggerated. Bertin (1983) has a similar discussion about map reading at three different levels: elementary, intermediate and overall. Map reading at the elementary level concerns individual map objects such as bends on a road. At the intermediate level we can isolate partial images to compare them with each other, such as comparing different road alternatives. At the overall level all objects in a map form a pattern that is easy to recognise and remember.

In urban areas the street network is shown differently in the Swedish topographical maps at the scale of 1:50 000 and 1:100 000. At the scale of 1:50 000 all streets are shown, while at the scale of 1:100 000 only a subset of the streets are shown. It is interesting to note that the 1:50 000 map does not contain enough information for the cartographers at the National Land Survey to construct the street network in the 1:100 000 scale map. At this scale the main streets are shown and additional information is needed to describe which are the main streets. Mackaness (1995b) presents an approach to generalisation of street networks using “alpha analysis”, which is a method to analyse urban structures. Based on the alpha-analysis it is possible to determine which streets ought to be most important within the street network. Based on this analysis it is possible to generalise the street pattern. If the purpose of the map is to convey the urban structure from a global view this might be satisfactory. If we are going to use the map for navigation there is a risk that a street that is suggested to be important by the alpha analysis turns out to be a dead end in reality.

The category road seems to be relatively well defined, even though the context of the road changes with scale. From a theoretical point of view it might be possible to include all information that is needed to create a small-scale road network in the large-scale database. At the large scale we would then have to classify the roads into parts that are part of the transportation network and paved areas leading to parking lots, bus stations etc. We would need a classification of the more or less important streets and the road objects would need to be split in such a manner that it is possible to form a node – link structure at the smaller scale. What information that needs to be included explicitly in the database and what can be computed by methods and algorithms to generate maps at different scales is difficult to know. As was illustrated with the discussion about alpha analysis above it depends on the purpose of the maps to be created. A reasonable approach seems to be to not try to find algorithms and methods that automate generalisation tasks a cartographer claims he can't solve.

If we aim at including all information that is needed to create maps at a large variety of scales and for a large variety of purposes in a large scale database, there is a risk that we burden the surveyor with a rather complex information acquisition task. What a practical approach is has to be decided in each individual case.

Buildings

Building is a relatively well defined concept. It is likely that most humans can agree on where the borders are between different buildings in a city block. A building has individual and relevant attributes such as owner, and explicit relations to other individual objects such as land parcel. As has been discussed in chapter 2.2.1 a building might also have a rather fuzzy relation to built-up area. If we divide the category buildings into subclasses such as farm and dwelling house the border between these two is fuzzy and depends on the context. A building that was used as a farm until e.g. the 1960's but is know used as a dwelling house, does still look like a farm and is perhaps portrayed as such on a tourist map. For taxation purposes, however, it is clearly a dwelling house and not a farm.

At the large scale the building is represented as a surface where the correct location of the building is very important for the same reasons as for roads. As the scale decreases more and more buildings are represented as symbols or merged into built-up areas. When the building is represented with a symbol, the exact location of the building becomes much less important. But it is very important preserve topological relations to neighbouring objects. Buildings has to be located on the right side of a road, even if the road has been moved 300 meters to solve a conflict with a railway. As the scale decreases a subset of the buildings is selected, since there is not enough map space to represent all the symbols. In a

Swedish topographical map this is the case at the scale of 1:100 000. The category buildings displayed in the map represents both the patterns formed by groups of buildings and the individual buildings. As the scale is decreased to 1:250 000 the buildings shown in the map only illustrate that there is a group of buildings, but there is no explicit connection between an individual building object and a real world building feature. It is for instance not possible to point at a building and obtain information about who the owner is etc. Buildings at this scale are good examples of an object class that is only used for display in a map and not used in any analysis.

As can be seen there is a change in emphasis as to how buildings are portrayed in a map, from the exact location of the building for construction purpose to the scale where buildings only illustrate groups of houses. However, since a building is a well-defined object it seems theoretically possible to automatically generate buildings in maps at different scales. As described above the location of a building symbol in a particular map depends to a very large extent on other objects displayed in the map. Therefore, the particular map imposes important constraints on how the objects are to be generalised in the map.

<u>Built-up area</u>

Built-up area is a category that is much more complex than buildings or roads. As described in chapter 2.2.1 Statistics Sweden has a definition that is based on the distance between different buildings. This might be sufficient for their needs but to a cartographer creating a topographical map a built-up area is a complex urban pattern consisting of buildings, gardens, streets, sidewalks, biking paths, playgrounds etc. How the built-up area is delineated depend on knowledge that the cartographer has in the form of image schemata (see chapter 2.1.2). It also depends on the purpose of the map. This gives the individual built-up area object a fuzzy definition in a similar way as has been described for forest objects in chapter 2.2.1. There are no attributes that are relevant to the individual built-up area object and one surveyor might delineate two objects while another surveyor merges them into one. One way to get well-defined objects could be to focus on categories that appear to consist of built-up areas such as villages or towns. A town is a clearly defined object in the sense that it has individual attributes such as name and population. It would be possible to construct links between all the built-up area patches and the town object. The town concept has a fuzzy delineation however and when looking at a topographical map showing two neighbouring towns it would require a rather theoretical definition to determine which town a particular built-up area belongs to. To define such links would require extra work, but it would add very little extra functionality to the system. A simpler approach is to treat the individual built-up areas as objects and accept their vagueness.

In a Swedish topographical map at the scale of 1:50 000 the built-up areas are classified into four different classes: dense built-up area, built-up area consisting of high buildings, built-up area consisting of low buildings, and built-up area consisting of summer cottages. At the scale of 1:100 000 these classes are merged into one built-up area class. To some extent we get a new set of objects, since neighbouring built-up areas are merged into larger objects. Small individual built-up areas can now be represented with a cluster of individual buildings in some cases. On the other hand, areas represented as a dense cluster of buildings at the larger scale may now be merged into built-up areas, see the example in chapter 2.2.1. At the scale of 1:250 000, we get a new set of built-up area objects when the objects at the scale of 1:100 000 are merged.

Fuzziness seems to be inherent in everything that concerns built-up area, which makes it difficult to abstract the category into a crisp, object oriented, data model. Since the category is vague in general, its delineation and use depends to a large extent on the application and the particular map in which it is displayed.

Land use

Land use is here discussed from the perspective of municipal and topographic maps. In other applications, such as forestry, the discussion about category crispness and how the category changes with scale will be quite different.

The land use categories in a Swedish topographic map at the scale of 1:50 000 are: forest, clearings, unmixed deciduous forest, arable land, open land that is not arable, swamp, and orchards. At the scale of 1:100 000 these classes are merged into forest, swamp, and open land. The problem here is the same as with built-up area: there are no distinct objects. The boundary between a forest and an open area is usually quite distinct. But, how large shall a group of trees in a field be to be mapped? How large shall a small open area in a forest, or a group of deciduous trees in a mixed forest, be to be mapped? If there is a small strip of open land between two forest objects when should these forest objects be merged into one object? As with the built-up areas this varies with scale, and when the scale is changed we get a new set of objects. In Sweden large parts of the country are covered with forest and when creating a seamless geographical database for a community we could end up with a huge forest object that stretches through the whole area covered by the database. It would be possible to divide the forest object where it is cut by roads, but this is a rather technical solution. Conceptually, a road leads through a forest and does not divide it. It seems as if the boundaries of the forest objects are more important than to define what an actual forest object is. Yet, there is an interest to know how much area is covered by forest, or arable land in a region so it seems reasonable to map land use as surfaces.

The land use categories seem to be as fuzzy as the built-up areas in topographical mapping. How they shall be forced into a crisp object oriented datamodel depends on the application.

Boulders

This example is taken from Ahlqvist and Arnberg (1998). Ahlqvist refers to Lind (1997) and discusses how the category boulder is mapped in a geomorphologic, a vegetation, and a geological map.

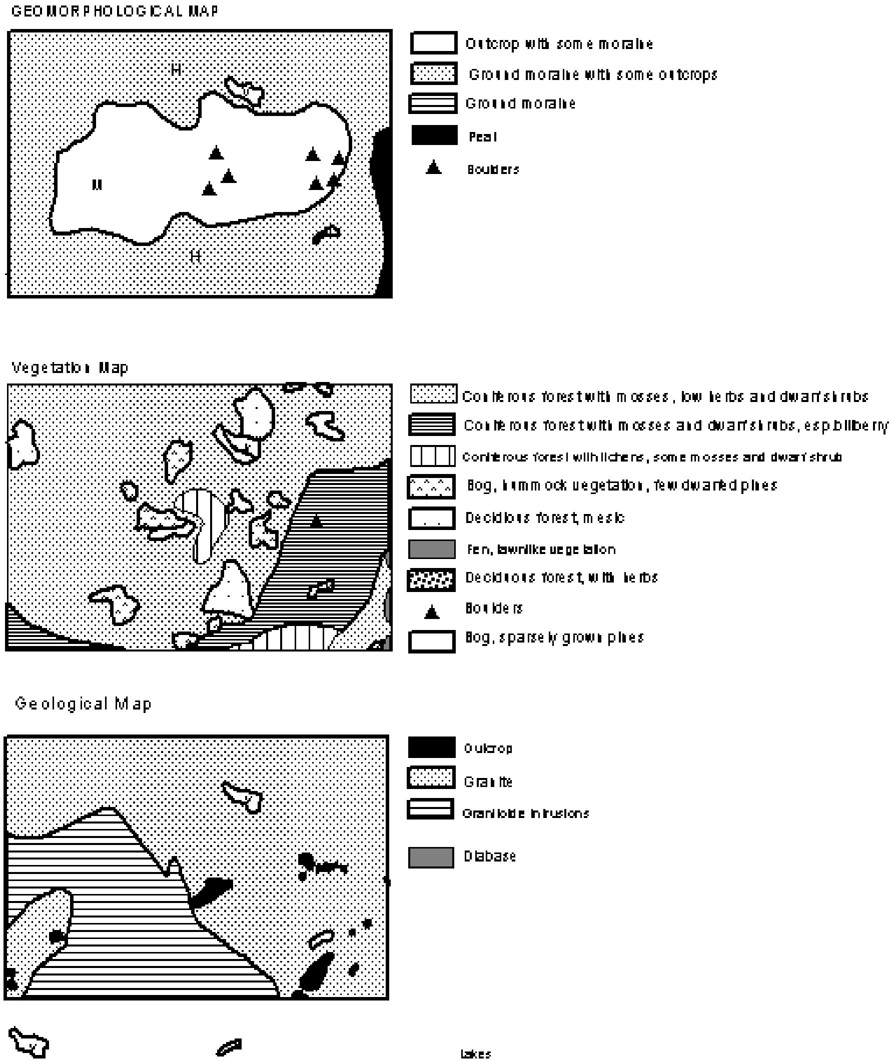

Figure 3.3 Thematic maps over the same area. (From Lind 1997)

As can be seen in figure 3.3 the category boulder is mapped quite differently in the different map series. Ahlqvist argues that this depends on the fact that the purpose for mapping boulders differs between the different map series. In vegetation mapping the boulders are significant, since they have impact on the vegetation types. In a geomorphologic map, boulders convey important information about the interpretation of the landforms and their genesis. In the geology map the boulders are used as an indicator of actual bedrock.

3.2 The user interface

As has been described in the introduction, we attempt to model a view to a database rather than the actual database. This chapter gives a more detailed description of the design and functionality of such a view or user interface.

The user interface is typically rather simple and designed for a particular user group. It gives the user the ability to view a map and zoom in and out to see details or to get an overview. The information that is displayed changes with scale. The user can retrieve additional information about the objects that are shown in a view, such as the owner of a house. It is also possible to perform different analyses depending on the application. In a car navigation system, for instance, it is possible to find the best route and to compute distances between locations. The view to the database is static in the sense that the user can not add additional information to the view. The user can choose to see subsets of the information that is available to a view however, by marking information in a legend. Sjödin and Strid (2000) give an example where the approach to modelling of geographical information presented in this thesis has been tested. They create a view for vehicle navigation where information about parking, petrol stations, and restaurants are displayed in base maps at two different scales. Theoretically, the symbols that are added to the base map can be retrieved from the Internet. The symbols are defined in the view in the sense that they are imported into classes equipped with methods that move the symbols when conflicts between different symbols occur in the view. Symbols are merged if a conflict can not be solved. The information hidden behind a combined symbol can be retrieved by clicking at the symbol.

3.3 Design decisions

In chapter 4 an object oriented model is described that creates a view defined at multiple scales. It is designed to automatically generalise geographical data between the different scales. During the design of the model certain decisions have been made which are discussed in this chapter.

3.3.1 Continuous generalisation

Above it has been stated that a map is an optimal compromise to display geographical information. A tempting thought is then to aim at building a system that incorporates a continuous generalisation and that finds the optimal presentation at every scale the user zooms to. There are two difficulties with such an approach: The obvious difficulty is the complexity in building a system that performs generalisation in real time. The other difficulty is that a continuously changing user interface might be confusing to the map-reader. Petzold et al. (1999) present an approach to automatic name placement that is continuously changing with scale. Since names can be placed in several different positions around a feature, even small changes in scale can result in drastic change of name placement due to conflicts and this is experienced as rather confusing. In our work a simpler approach has been taken. Each view consists of a set of maps, and each map is static and defined at a certain scale range. As the user zooms the map that corresponds to a certain scale is shown. Such a map is called a level and its characteristics are described more explicitly below.

3.3.2 Single object vs. Multiple objects

The obvious approach when modelling a geographical database is to state that a geographical feature, such as a house, forms one object in the database. This object can have several geometric attributes suitable for different scales and be equipped with methods that generalises the object when the scale changes. This is the approach taken by e.g. Jones et al. (1996). However, there are several difficulties with this approach. One is that several geographical objects have a vague definition; the examples land use and built-up area are discussed above. A technical solution to the problem of how two built-up areas shall be merged when the scale decreases is to claim, that one of the objects disappears and that the second object is enlarged with the area of the first object. Cognitively this is unsatisfactory since it is obviously not the same object. Another problem is that constraints on how an object shall be represented in a map depend to a large extent on the map purpose. If the object is to be represented in a large set of maps, different algorithms, parameters and perhaps geometries have to be tied to the object class and the individual objects for each map.

If an object, such as a road, is presented in several different maps, there is a risk that the choice of symbolisation creates conflicts in some of the maps. One approach to solve this problem could be to create a new geometry in every map where a conflict occurs. The difficulty with this approach is that when the database is edited on another occasion, there is a risk that conflicts occur in a map that is currently not viewed. The solution seems to be to check all maps where an object is represented to make sure that no conflicts occur. The advantage of letting one geographical feature be represented by one database object is that attribute information, such as house owner, is tied to one object. Furthermore, links to information about the house, stored in other databases, such as information about maintenance, are easier to maintain.

The approach taken here is to allow for a geographical feature to be represented by several objects. Each object belongs to an object class, and each object class is tied to a map. The object class is an abstraction of reality that is suitable for the particular map. For the remaining part of this thesis, such a map is called a level.

3.3.3 The level

A level has the following characteristics:

- A level is static and is defined to be viewed at a certain scale range.
- An object class is defined in one and only one level.
- The representation is defined for each object class, which means that an object class can only be represented in one way.

This implies that a category, such as building, will be represented by several object classes where each of these object classes is defined for a certain level. Each of the object classes that represent buildings has a more limited meaning than the general category building. A large scale object class might require that every building feature should be mapped while a small scale building class only contains larger buildings. In the large scale object class the buildings are represented as surfaces with a requirement for high geometrical accuracy. In the small scale object class the buildings are represented as symbols and the accuracy is of much lesser importance. At this scale it is more important to preserve topological relations between different objects, e.g. a house should be located on the right side of the road, than to maintain a high accuracy for individual objects.

Creation of new objects

When a new level is defined the cartographer creates a data model for the level. He defines the object classes that are members of the level, how they are represented in the level, the requirements for preservation of topological constraints between the different object classes etc. The object classes are then equipped with methods that retrieve and analyse information from other sources to instantiate new objects. When all definitions are made, objects that are members of the level can be instantiated. As the new objects are created the new level analysed to find a suitable location without conflicts with previously created objects.

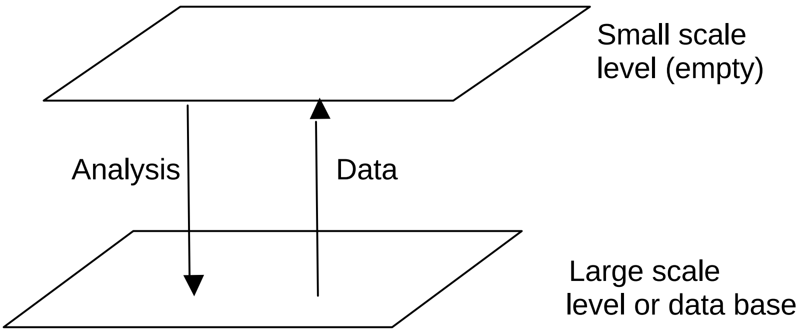

Figure 3.5   Generalisation

If the new objects create conflicts, which can not be solved, it can be decided not to create the object or to merge the new object with an existing object. What is done depends on the application and the object classes involved. The analysis is performed at two levels simultaneously: the level from which the data is retrieved, and the level that is being created. This approach makes a clear-cut distinction between the data source and the data that is created in the new level, since conflicts can only occur between objects that belong to the same level. In the case study presented below data is retrieved from only one level, but it is possible to fetch data from several sources when a new level is created. Sjödin and Strid (2000) present an approach where the base map is retrieved from one source and the symbols that are to be displayed in the map are retrieved from another source.

Relations between levels

A difficulty with the design that is proposed here might be the number of levels and object classes that will be created. This is especially significant if some object classes are connected to other databases. Buildings in a GIS might, for instance, be connected to a database that contains information about maintenance, tenancy agreements etc. It seems reasonable to maintain such connections between databases for as few object classes as possible. One approach could be to let the object class buildings, that is connected to another database, be defined only once in the database. This object class is a member of the initial level and can be called the main object class. Object classes that represent buildings in other, automatically created, levels can form links to the buildings in the initial level. Using these links it is possible for the user to retrieve information about the main object class and thus fetch information from other databases, to which the main object class is connected. The focus in this work is on levels that are automatically created. Levels that are manually updated or connected to other databases will not be discussed further.

Kilpeläinen (1997) presents a multiple representation database. In this database objects are defined at several representation levels. The objects that represent the same real world feature are connected with links. Kilpiläinen motivates these connections between different levels with two reasons: to enable the automatic propagation of updates between levels, and to facilitate a reasoning process between levels. Since the levels discussed in this work are automatically created, propagation is not a problem. The level can be recreated whenever there is a need. As has been described above there is a need to be able to navigate between objects defined at different levels.

Object classes and associations are crisp, while categories and relations between different categories in geographical information often are vague. Intuitively, it is simple to say that we see a cluster of individual buildings at the large scale. These buildings form built-up areas at the intermediate scale which are merged into a symbol that represents a town at the small scale. However, it is quite difficult to define in every case which buildings belong to which built-up area and which built-up areas are members of the town. And perhaps there is no need to model such relations. The approach taken here is that we should only model associations that are required by the application, and these associations ought to be clearly defined. At a large scale, for instance, every building is represented and at the intermediate scale a subset of the buildings is shown. In both the large scale and the intermediate scale it is clear which building the object refers to and links can be formed between the individual building objects. At the small scale buildings are shown only to convey to the map reader that there is a group of buildings in this area. Since the building object does not convey information about the individual building feature there is no need to form links to the objects in the intermediate and large scales and it does not seem likely that an application will require it. Land use objects have been discussed above and the vagueness of the individual objects has been highlighted. It is argued that we get a completely new set of objects as the scale decreases. To specify links between these objects in a clear and unambiguous way seems like an impossible task. If they are not required by the application such links shall be avoided.

Sequence of creation

It has been argued above that the sequence in which different tasks shall be done is an important issue in automatic generalisation. Baeijs et al. (1996) present interesting work in automatic generalisation using Multi-Agents. In their approach the cartographic objects or agents can communicate and effect each other until an equilibrium state is achieved which means that every agent is in a stable position.

The model presented here contains a simpler approach to the sequencing problem. The object classes are handled one at a time and all objects belonging to a class are created in a random sequence. The object classes are processed in an order similar to the one used in traditional map making. First, classes that give structure to the map such as lakes, rivers, railways, and roads are handled. Then object classes such as buildings (treated as symbols) and land use, are created. The order in which object classes are handled depends on the purpose of the particular map.

In chapter 4.1 sequencing between different operations is described for each object class. The sequence is expressed in the constructor of the object class and once an object has been created it is not modified.

Networks

When creating a level it is important to maintain crossings between linear features such as roads, railways, and streams, since these form important landmarks. Therefore, the choice was made to let all linear features in a level form a node-link structure, where each link is an object. A road object might for instance run between two road-stream crossings. This might seem a rather artificial way to split linear features into objects. If a system has support for topology, information about crossings can be retrieved through the topology without a need to treat every link as a single object. However, it is easier to solve conflicts when new objects are instantiated if a node-link structure is kept. This will be described further in chapter 4.1.

4 The case study

The approach to modelling presented here is mainly focused on creating views into a geographical database that show information customised for a specific user group. It is assumed that such views can be simpler than traditional maps. The case study, however, deals with generalisation of a subset of categories in a topographical map, which might seem as a contradiction. A topographical map is a multi-purpose map. The categories have been chosen, however, to get a wide representation of different types of geographical categories which interact in different ways. Jones et al. (1995) argue that there is a need for this kind of research efforts to solve generalisation problems in a holistic manner rather than focusing on individual object classes. A study of these categories illuminates the problems with this approach better than a simple map designed for a particular user group. Sjödin and Strid (2000) describe how this approach to modelling is used to create a simple map for a specific user group (see chapter 3.2).

Parts of the model have been implemented in the GIS software LAMPS2. LAMPS2 is built on an object oriented database and has support for topology. When the object classes are implemented in LAMPS2 they inherit functionality from the base classes already defined in LAMPS2 and topological relations between the different object classes are defined. The connections to the LAMPS2 data model are not described here but for a thorough description of the LAMPS2 data model see Laser-Scan (1999).

4.1 The model

The model consists of two levels: the *Green* and the *Blue level*. The *Green level* is defined at the scale of 1:10 000 and is the base level which is initially filled with data. In the case study, data from the NLS are imported into this level. The *Blue level* is automatically created by retrieving and analysing data from the *Green level*. Depending on the choice of symbolisation and generalisation parameters the *Blue level* can be instantiated at scales in the range from 1:30 000 to 1:100 000.

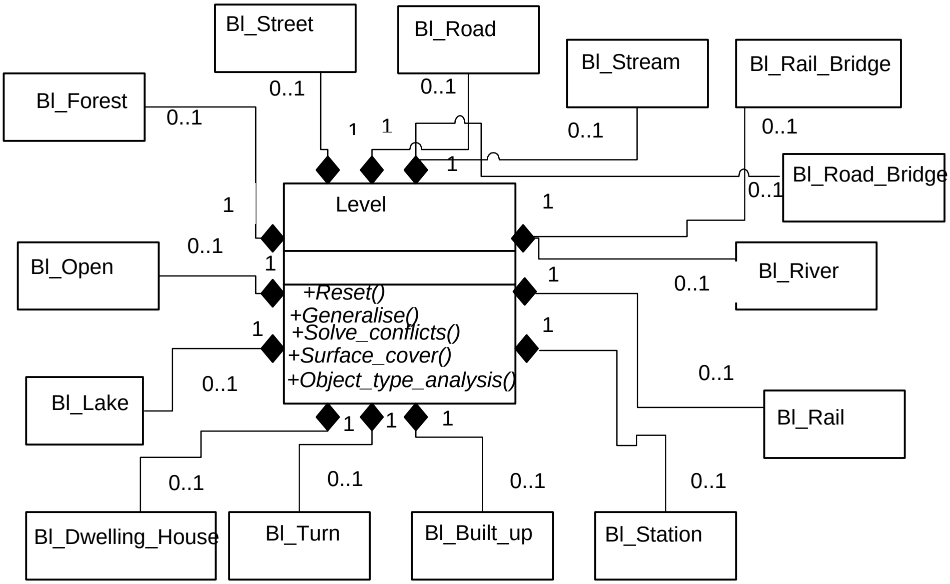

Figure 4.1 The Bl object classes are members of the Blue level.

<u>Levels</u>

Figure 4.1 shows all the geographical object classes that are members of the *Blue level*. The connection between the object classes is an aggregation, meaning that a level consists of geographical objects. Figure 4.2 shows the members of the *Green level*. The *Green level* contains the data, while the *Blue level* is empty at the start. A process is initiated in the *Blue level* and data is extracted from the *Green level*, analysed and inserted into the *Blue level*. The process ends when all objects in the *Green level* have been processed.

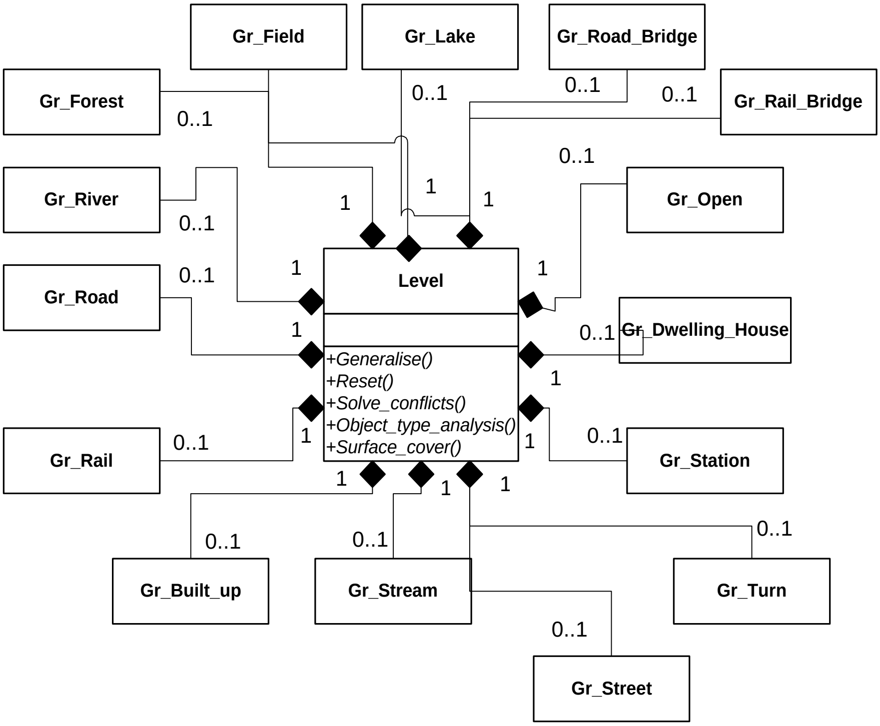

Figure 4.2 The Gr object classes are members of the Green level.

<u>Inheritance</u>

The model uses inheritance structures, which are shown in Figure 4.3 - 4.5. There are three different inheritance structures for objects represented as points, lines and areas. At the most general level there is a virtual object class called *Bl_Object*. In this class the three methods *select(), create()* and *simplify()* are defined. These methods are implemented further down in the object class hierarchy. All linear object classes inherit from the *Bl_Network* object class. There is an association within the class that describes how linear objects form node-link structures. Most of the attributes concerning symbolisation and general constraints, such as accuracy requirements or the minimum length an object should have to be displayed at this level, are defined in the *Bl_Network* object class. The values of the attributes are then assigned in the leaf object classes. Object classes such as *Bl_Station* and *Bl_Railroad_Bridge* have associations with *Bl_Railroad*. These associations illustrate topological relations between these object classes. A *Bl_Station* or a *Bl_Railroad_Bridge* has to be connected to a *Bl_Railroad*.

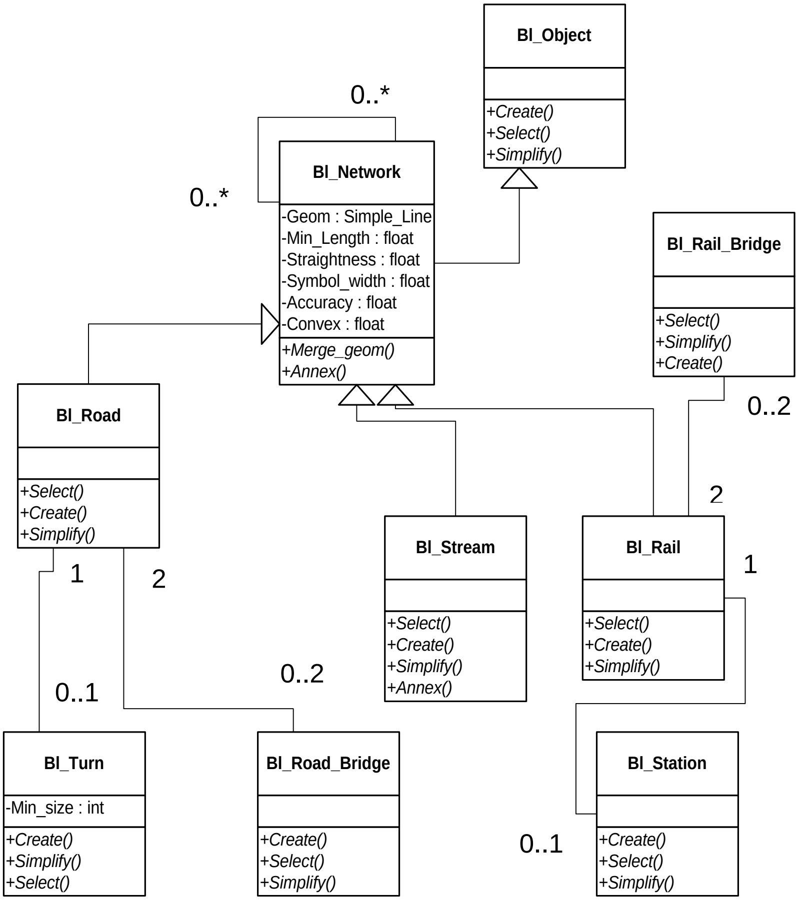

Figure 4.3 The inheritance structure of object classes in the Blue level that are represented with linear geometry.

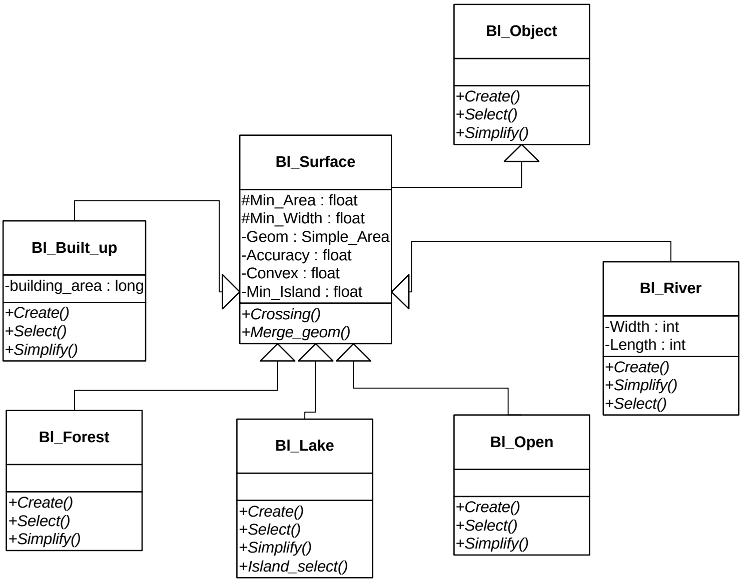

Figure 4.4 The inheritance structure of object classes in the Blue level that are represented with surface geometry.

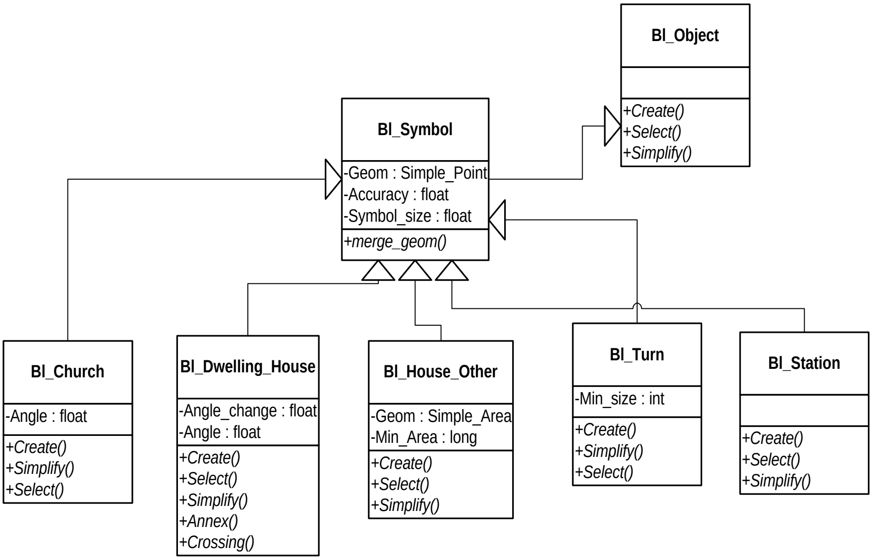

Figure 4.5 The inheritance structure of object classes in the Blue level that are represented with symbol geometry.

In the initial stages of this work an attempt was made to model the hydrographic network using multiple inheritance. The hydrographic network includes both surface objects, such as lakes and rivers, and linear objects such as streams. The idea was that the *Bl_River* object class to some extent has the characteristics of a lake and to some extent has the characteristics of a stream and thus should inherit functionality from both. As the functionality was designed in detail it turned out that functions which the *Bl_River* object needed to inherit where defined within the *Bl_Surface* object class. The characteristics of the *Bl_Network* object class, which the *Bl_River* class needs, are the requirements on how data should be structured. To make sure that crossings of linear features, e.g. roads, and rivers are maintained properly, the river is divided into a new object at every crossing. Figure 4.6 shows how a *Bl_River* object is formed between two *Bl_Road_Bridges*.

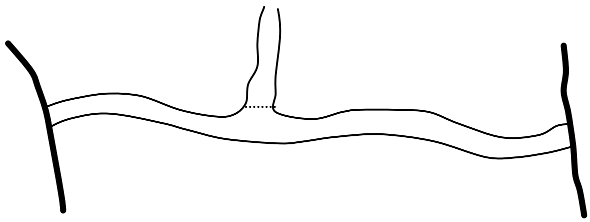

Figure 4.6 Example of a Bl_River object that ends where a road crosses the river. There is also a split into separate objects where the two Bl_River objects meet.

The part of the road that crosses the river, the bridge, has different requirements on how it shall be generalised compared to other parts of the road. A road bridge should be located "on" the river and other parts of the road should be located "on" land. Therefore, all road and railway bridges are treated in separate classes. Where a road crosses a stream there is no need to form a bridge for the purpose of generalisation. The node where the two objects cross is maintained by the network structure.

The different road types are treated as separate road classes since they have different symbolisation. As can be seen in figure 4.7, however, they have the *simplify()* method in common which is implemented in the general *Bl_Road* class. The discussion about where to implement generalisation functionality in the inheritance tree is deferred to chapter 4.1.1.

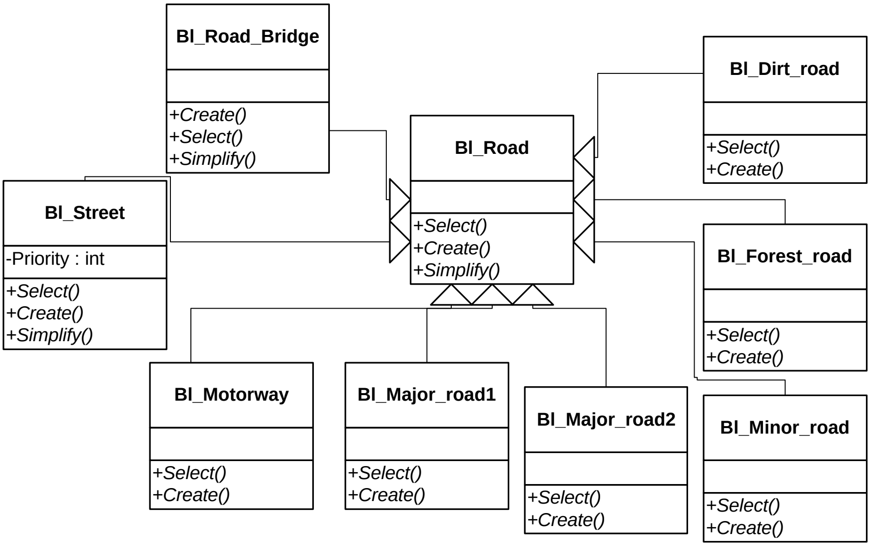

Figure 4.7 Different road types inherit from the general Bl_Road object class.

The object classes that inherit from the *Bl_Surface* object class forms a surface cover. There should be no gaps or slivers in this surface cover. In some cases an object class, such as *Bl_House_Other*, might be represented with area geometry that is not part of the surface cover. These object classes have the characteristics of symbols and are located "on" the surface cover rather than forming a part of it. In Figure 4.5 the class *Bl_House_Other* inherits from the symbol class even though it has a surface geometry. As can be seen in Figure 4.5 the attribute *geom* that stores the geometry, is defined as a point in the *Bl_Symbol* object class, but is redefined as an area in the *Bl_House_Other* class.

<u>Dependencies</u>

In the introduction to this chapter it was stated that topology will not be discussed. There is a part of the model however that defines which objects might be effected if an object is moved. This is illustrated using dependencies, which have a specific meaning in this model. The *Bl_Lake* class in Figure 4.8, for instance, has dependencies with all other object classes. This means that if a *Bl_Lake* object is moved, conflicts with objects of any other object class might occur. In Figure 4.9 describe how the move of a *Bl_House* object effects objects in other classes. The interesting information is whch object classes are left out. As can be seen the object classes *Bl_Forest* and *Bl_Open* are not considered to be effected when a *Bl_House* object is moved. This means that it is irrelevant in this level if a *Bl_House* object is located "on" a *Bl_Forest* object or "on" an *Bl_Open* object. This is a design choice for this particular level and in a different level it might be decided to let the *Bl_House* object class have dependencies to all other object classes.

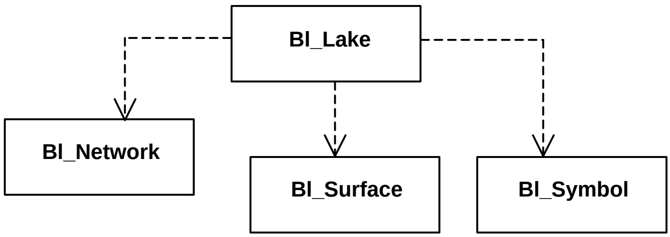

Figure 4.8 This figure illustrates which object classes that might be effected if a Bl_Lake object is moved.

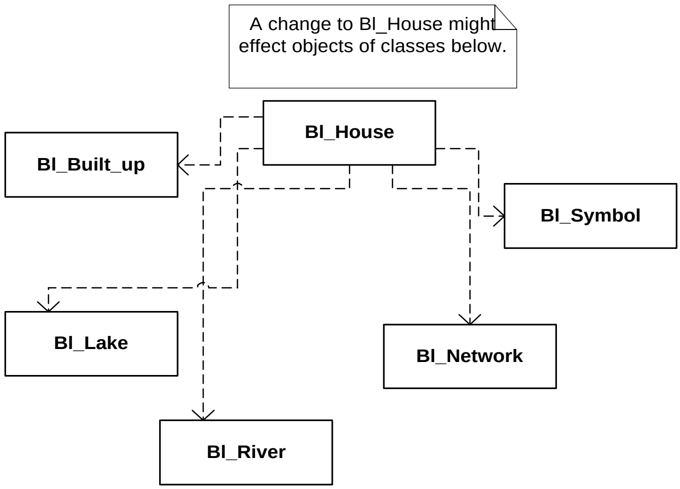

Figure 4.9 This figure illustrates which object classes that might be effected if a Bl_House object is moved. A Bl_House may not be located on water or on the wrong side of a road but it might be moved from a forest to an open area without severely deteriorating the quality of the map.

Similar relations are created for each object class. Even though the *Bl_House* and *Bl_Forest* object classes do not have dependencies, they can effect each other indirectly through other object classes. A *Bl_House* house object, for instance, can be moved in such a manner that it effects a *Bl_Lake* object, which in turn, effects the *Bl_Forest* object when it is moved. All object classes in the level have direct or indirect dependencies.

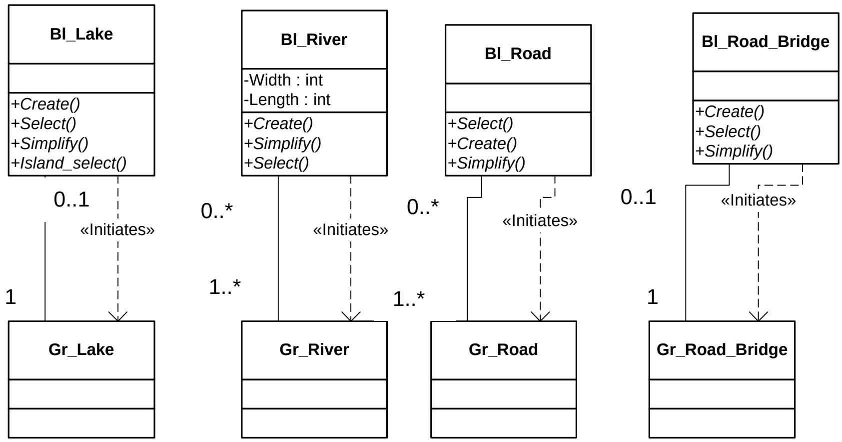

Figure 4.10 There are two kinds of links between object classes in different levels, association and dependency.

<u>Links between levels</u>

There are two types of links between object classes defined in different levels, association and dependency. This is illustrated in Figure 4.10. The association is formed for the object classes where the relation is intuitively simple or where the relation is needed when the data in the new level is created. The association between *Bl_Lake* and *Gr_Lake* is simple to form since it is considered to be obvious which objects in the two classes that refer to the same real world feature. The association between *Bl_River* and *Gr_River* is more vague and forms a many-to-many relation. According to Schylberg (1993) the production rule for rivers, at the National Land Survey of Sweden, is that a river feature that stretches longer than 500 m should be wider than 10m in scale 1:50 000 and 20m in scale 1:100 000. The *Bl_River* objects in the *Blue level* represent river features that have a width greater than 10 meters, while the *Gr_River* objects in the *Green level* represent river features with a width greater than 6 meters. This implies that a river feature that has a width of 7-8 meters for a certain length is a *Gr_River* object in the *Green level* and a *Bl_Stream* object in the *Blue level*. A *Bl_River* object is part of the node-link structure formed by all linear objects. This means that every *Bl_River* object begins or ends at a *Bl_Stream*, a *Bl_Lake*, or wherever it is cut off by a linear object such as a *Bl_Road* or a *Bl_Railway*. If a *Gl_Road* that crosses a *Gr_River* in the *Green level* is not represented in the *Blue level*, the two *Gr_River* objects in the *Green level* are merged into one in the

Blue level. This case is illustrated in Figure 4.11. On the other hand, if a long *Gr_River* object in the *Green level* contains a part that is narrower than 10 meters it shall be represented with three objects in the *Blue level* - a *Bl_Stream* object in the middle and a *Bl_River* object on either side of the *Bl_Stream* object where the river is wider than 10 meters. This case is illustrated in Figure 4.12. It is quite clear that this association is not intuitive, but it is needed later in the map creation process when buildings are generated to make sure that the buildings are located on the right side of the river. For built-up areas, forests, and open areas there is no need to form associations.

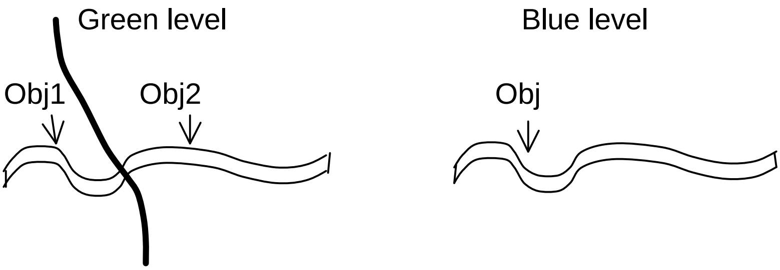

Figure 4.11 Two Gr_River objects in the Green level are merged into one object in the Blue level, since the Gr_Road object in the Green level is not represented in the Blue level.

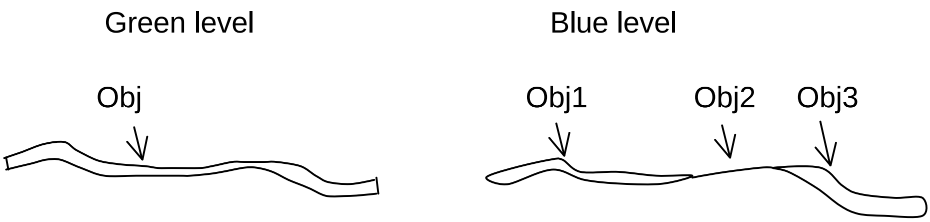

Figure 4.12 The Gr_River object in the Green level is split into three objects in the Blue level since, the middle part of the river is narrower than ten meters and is represented as a Bl_Stream.

The other relation that is formed between object classes is a dependency called “initiate”. It has a different meaning than the dependency in the former section that illustrates topological relations. During the creation process the object classes in the *Blue level* analyse the *Green level* to find objects and groups of objects that can form objects in the *Blue level*. The first part of this analysis is to locate an object that can be a seed for the creation of objects in the *Blue level*. The *Gr_Lake* objects are the seeds for creation of *Bl_Lakes*. The *Gr_House* and *Gr_Built_up* contains the seeds for the creation of the *Bl_Built_up* in the *Blue level*. This relation is illustrated with the "initiate" dependency. The actual creation process is further described below.

<u>Sequence</u>

As has been described in chapter 3.3.3 the object classes are stepped through one at a time when the objects of the *Blue level* are created. Figure 4.13 illustrates how the control is passed from the *Bl_Level* object to the different object classes.

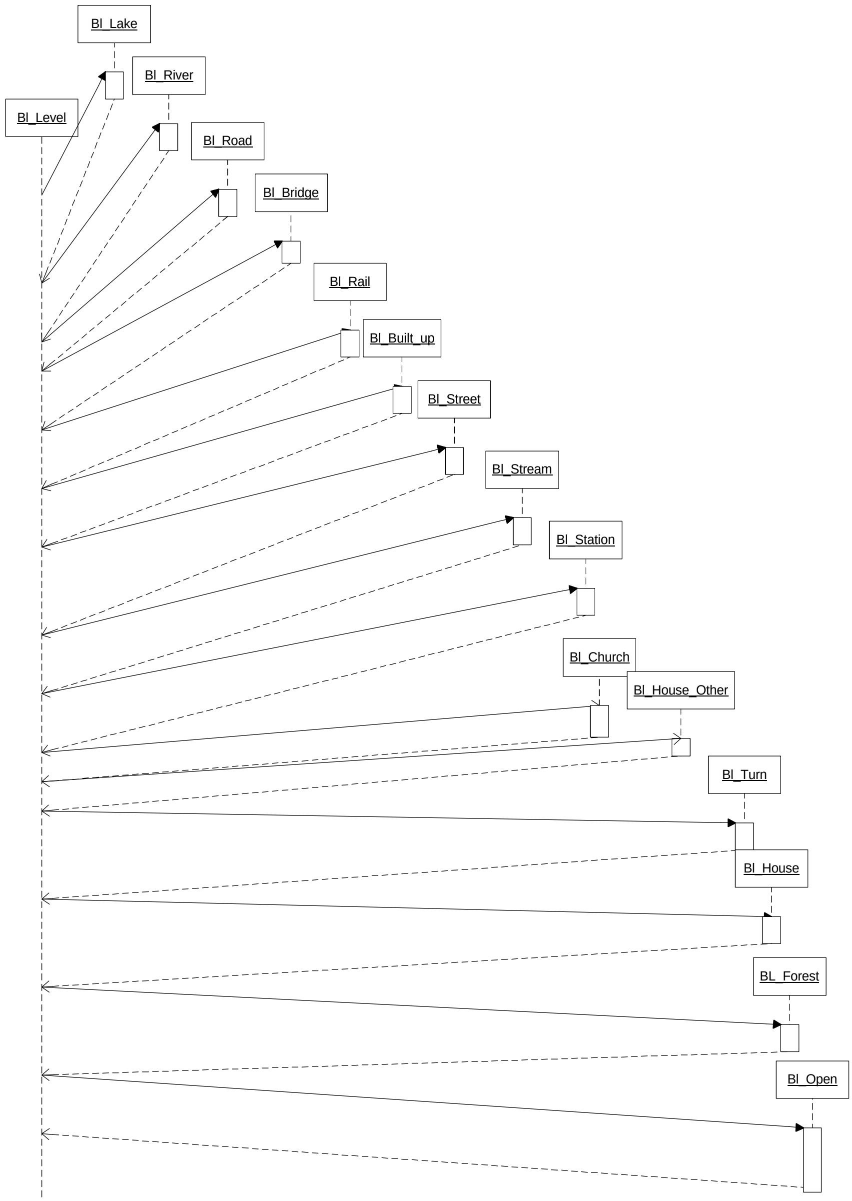

Figure 4.13 The sequence of the creation process.

The process starts with the object classes that are most important and give structure to the map and then works through the less important object classes. In this model objects can not be modified after they have been instantiated. Initially an attempt was made to equip every object class with a *move()* method to be called when conflicts occur during the creation of new objects. The *move()* method was supposed to move the whole object or a part of the object, whichever was most appropriate for the particular object class. This approach introduced complexities in the model that appeared too difficult to sort out. If an object was moved, for instance, new conflicts would occur and the *move()* method of these object classes would have to be called which would create new conflicts etc. It would be difficult to know when to end the process and what to do instead. Furthermore, it does not seem likely that a complex area with several conflicting geographical objects can be generalised properly by objects that are calling each others *move()* methods randomly. Sjödin and Strid (2000) have a slightly different approach to this problem. Objects that have been instantiated can not be moved, but they can be merged with other objects when conflicts occur. This approach is efficient, since the merge operation in this case simplifies the map without introducing any new conflicts. The objects that are merged are symbols with equal size representing parking, restaurants and petrol stations. If two symbols with different meaning are merged they are displayed as a symbol that represents more than one feature.

In some cases there is not enough space to create a new object in a level, but the model is set up in such a manner that the object shall be instantiated anyhow. A level that is too dense is created. The designer of the map can see the results when the level is finished and can then choose to modify symbolisation, the object classes that are part of the map or the constraints on how the generalisation is to be performed. After these modifications a new attempt can be made to create the new level.

4.1.1 The creation process.

The creation of new objects follows a similar pattern for all object classes. However, since different object classes have different characteristics and different relations to other object classes there are variations.

In general, the creation of all objects in a class consists of the following steps.

1. The objects that can act as seeds to form objects in the *Blue level* are selected and stored in an array. The seed relations are shown with the "initiate" dependency in Figure 4.10. The array is stepped through and the objects are analysed one at a time. Step 2-4 is done for an object at a time until the end of the array is reached.

2. First the *select()* method determines if this particular object is suitable to form an object in the *Blue level*. For *Bl_Lakes* this is a simple decision if the *Gr_Lake* is large enough. In the *select()* method other objects in the vicinity of the current object can also be analysed. An island, for instance, is selected if it is larger than a certain size and if there are no objects on the island that shall be members of the *Blue level*. An example of such an object could be an important building. This is determined by calling the *select()* method of the objects on the island. The select function does not give the complete answer to the question if an object shall be displayed or not. At a later stage in the construction process an analysis is made how much space is available for a particular object in the *Blue level*. If there is not sufficient space the object may not be instantiated even though the *select()* method has returned a true.

3. Based on the accuracy requirements of the object class a buffer is created around the object that shows where the new object may be located in the *Blue level*. If there are other objects in the *Blue level* within this buffer that the current object can be in conflict with the buffer is modified until no other objects are located within it, using the *merge_geom()* method. Which object classes a particular class can be in conflict with is modelled as dependencies, which was described above. The result is an area where the new object may be located.

4. The *simplify()* method computes the geometry of the new object using the geometry of the seed object and the area where the new object may be located.

5. When all objects are created there might be conflicts in the level due to symbolisation. Theoretically, these conflicts can be solved using the constraint method described by Harrie (1999).

Lakes

The creation of *Bl_Lake* objects follow the structure that has been described above to a large extent. When it has been decided that a *Bl_Lake* shall be created the *island_select()* method is called. This method steps through the islands in the *Gr_Lake* and analyses if they shall be selected for the *Blue level*.

Roads

Roads are more complex, since each *Bl_Road* object only stretches between two nodes in the *Bl_Network* structure. This means that a single *Bl_Road* object might stretch between a crossing with a *Bl_Stream* and a crossing with a *Bl_Railway*. Harrie (1998) describes the analysis made at the National Land Survey of Sweden, when determining if a dirt road is to be selected as a member of the green map (1:50 000). A simplified version of the selection process is to say that a dirt road that is a cul-de-sac is not created if it is shorter than 500 meters and does not lead to a lake or a house. This analysis has been implemented in the *select()* function for *Bl_Dirt_Road*. A recursive method called annex() is used to collect the *Bl_Road* objects that lead between two road crossings or form a cul-de-sac. This set of *Bl_Road* objects is then analysed according to the above criteria.

If a new *Bl_Road* object shall be created in the *Blue level*, the extent of this *Bl_Road* object has to be determined. Linear objects should have a node-link structure similar to the structure in the *Green level*. However, we do not know if the other *Gr_Network* objects, e.g. *Gr_Streams*, at the endpoints of the current object will be displayed in the *Blue level*. A collection of objects in the *Green level* that can form one object in the *Blue level* has to be formed. Another recursive function calls the *select()* functions of the *Bl_Network* objects that connect to the endpoints of the current object. If these objects shall not be displayed in the *Blue level* the *Gr_Road* object is added to the collection. When the function ends, we have a collection of *Gr_Road* objects that shall form one *Bl_Road* object in the *Blue level*.

A buffer is created where the new object may be located and modified in a similar manner to that described above. In this process it is an advantage to treat individual links in the *Gr_Network* structure as objects.

Figures 4.14 and 4.15 show how a stream crosses a road and then makes a turn and moves along the road. In Figure 4.14 the individual *Gr_Road* object stretches between two road crossings and thus runs over the crossing with the *Gr_Stream*. In Figure 4.15 the road is split at the stream and the individual *Gr_Road* object only leads between the crossing with the stream and the crossing with the road. As has been described above, the area where the new object may be located should not have any conflicting objects within it and is modified until all conflicting objects are outside the area. If an individual *Gr_Road* object would stretch over crossings with other linear objects, e.g. a *Gr_Stream*, we would have to accept the *Gr_Stream* object within the area where the new object may be located. This is the case shown in Figure 4.14, where we have to accept the part of the *Gr_Stream* that crosses the *Gr_Road* within the area, while the part that runs next to the road should be outside the area. In Figure 4.15 all linear objects are treated in a node link structure and this problem does not occur. At the end nodes the area is modified so that the end node is located on the border of the area.

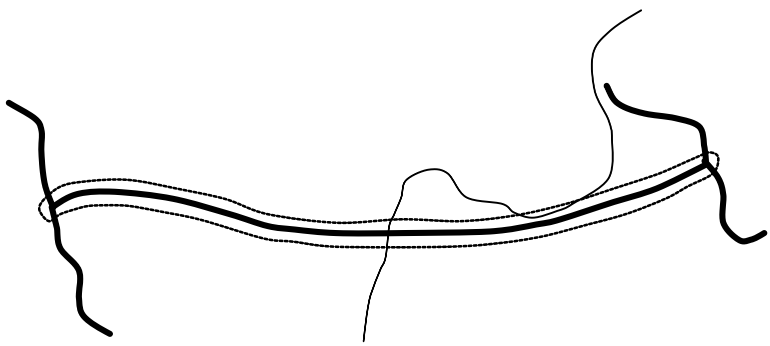

Figure 4.14, A stream crosses a road and then makes a turn and moves along the road. In this example the road object crosses the stream object. A buffer around the road shows where the new road object may be located. The stream is located within the buffer in two places, at the crossing with the stream and when the stream is close to the road.

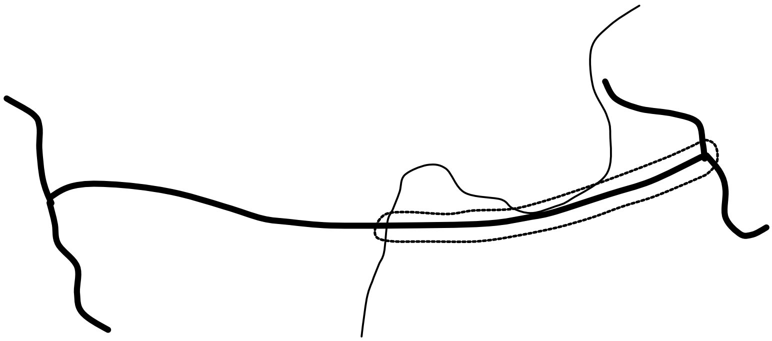

Figure 4.15, The same case as in Figure 4.14, but now the road object ends at the crossing with the stream.

Finally, a new *Bl_road* object is created in the *Blue level* based on the area and the geometry of the *Gr_Road* object(s).

Figure 4.7 illustrates how the model contains seven different road types that inherit from the general *Bl_road* class. It is interesting to note, however, that most of the methods and attributes for generalisation are not stored in the leaf object class. The *select()* method is implemented for each road type, while the *merge_geom()* as well as the *annex()* methods are implemented in the *Bl_Network* class. The *simplify()* method finally is implemented in the *Bl_Road* class. This illustrates an important advantage with using object oriented modelling for automatic generalisation. Object classes with similar behaviour can be organised in hierarchies in such a manner that methods and attributes can be shared.

River

As has been described above, the *Gr_River* objects are structured in such a way that they are part of the node-link structure of linear features in the map. To be able to analyse if a particular *Gr_Road* object in the *Green level* shall initiate a *Bl_Road* object in the *Blue level*, neighbouring *Gr_Road* segments had to be annexed recursively. Since every *Gr_River* object in the *Green level* will instantiate either a *Bl_River* or a *Bl_Stream* object in the *Blue level* this is not needed.

To determine which *Gr_River* objects in the *Green level* shall form a *Bl_River* object in the *Blue level* the annexation is needed in a similar manner as for *Bl_Road* objects. A set of *Gr_River* objects is collected, that stretch between two links in the network structure or between a link and a *Gr_Lake*. A *Bl_River* object in the *Blue level* is formed if the *Gr_River* is wider then a certain value, e.g. ten meters, for a certain distance. This implies that we might have to form two *Bl_River* objects with a *Bl_Stream* in the middle, see Figure 4.12. This is rather unsatisfactory, since the creation of streams ought to be handled by the *Bl_Stream* object class.

The processing of *Bl_River* needs to be thought out more in detail. Perhaps the requirements that determine if a *Bl_River* object should be formed, a certain width for a certain distance, are not detailed enough. When looking at some Swedish topographical maps it appears as if a stream turns into a river at a certain point in the hydrographic network and is treated as a river thereafter.

<u>Built-up area</u>

When *Bl_Built-up* objects are created the seed can be either a *Gr_Built-up* or a single *Gr_Dwelling_House* in the *Green level*. The surroundings of the seed object is analysed. If there are any *Gr_Built-up* or *Gr_Dwelling_House* objects that are located close enough they are annexed. However, they may not be located on the other side of a *Gr_River* or a *Gr_Lake*. This process continues recursively for the annexed objects until no more seed objects can be added. The select function determines if the objects cover an area that is large enough to be represented with a *Bl_Built_up* in the *Blue level*. If so, the new area is created using the *simplify(*) method. In this case there is no need to analyse if there are any conflicting objects since the only objects that have been created so far and that can have conflicts with this object are other surface objects. This knowledge can be obtained from the model by looking at the dependencies between different object classes and the creation sequence. New surface objects that overlap already existing surface objects are modified by the *simplify()* method until there is no overlap.

<u>Buildings</u>

There are three kinds of buildings in the *Green* as well as in the *Blue level*: Dwelling houses, churches, and other buildings. Churches and other buildings are rather uncomplicated and follow the general pattern for creation of new objects described above. Dwelling houses are more complicated. The seed to create new *Bl_Dwelling_House* objects in the *Blue level* are *Gr_Dwelling_House* in the *Green level*. The *select()* function determines if an individual *Gr_Dwelling_House* object is suitable to be created in the *Blue level*. However, a group of *Bl_Dwelling_House* objects in the *Blue level* represents both the individual dwelling house features and the pattern formed by a group of real world features. Thus we have to form a collection of *Gr_Dwelling_Houses* that can be analysed to determine which *Bl_Dwelling_House* objects that shall represent the pattern of buildings. This collection is created by analysing the surroundings of the seed *Gr_Dwelling_House* and adding neighbouring *Gr_Dwelling_Houses* to the collection. To be added to the collection a *Gr_Dwelling_House* should not be located on the other side of a linear object such as a *Gr_Road* or a *Gr_River*. When the surroundings of the seed *Gr_Dwelling_House* has been analysed the process continues recursively with the *Gr_Dwelling_House* that have been added to the collection until no more *Gr_Dwelling_Houses* are added. When we have a group of *Gr_Dwelling_Houses*, the area they can occupy in the *Blue level* is determined, which is shown by the dotted line in Figure 4.16. If the accuracy requirements for *Bl_Dwelling_Houses* are low, there is a risk that they can be located at the wrong side of a road. This can be avoided by using the links that are formed between linear objects in different levels. The roads in Figure 4.16, for instance, are connected with links between the levels. If, for instance, a *Gr_Road* object is located west of the *Gr_Dwelling_Houses* we can use the links to determine which *Bl_Road* object in the *Blue level* that has been formed from this *Gr_Road* and make sure that the *Bl_Dwelling_Houses* are located west of the *Bl_Road* object. When we know the area within which the *Bl_Dwelling_Houses* should be located, the *simplify()* method finds conflicts among the *Bl_Dwelling_Houses* due to symbolisation and selects a subset of the *Bl_Dwelling_Houses* to be represented in the map.

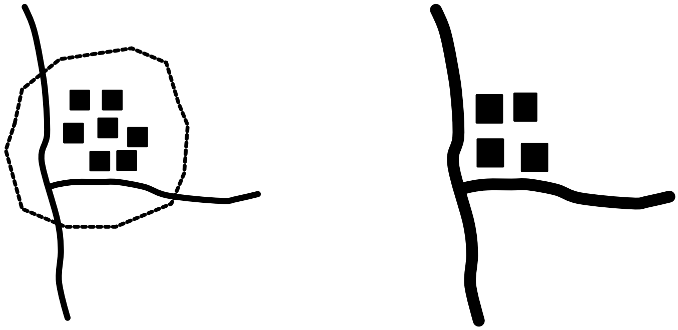

Figure 4.16 Left is group of buildings in the Green level that form a group of buildings in the Blue level to the right.

Final comments

The development of the model has to a large extent followed the steps suggested by Rumbaugh et al (1992). First, a general sketch of the model is created, which is gradually filled in with more and more details. One detail in the model, upon which very little effort has been spent, is the *simplify()* method which creates the geometry of a new object. In most cases the method simplifies the geometry of the seed object by picking out every second coordinate. The method also checks that every node in the new geometry is located within the area where it may be located and, if necessary, moves nodes to be within this area. Future work on this model is to evaluate different algorithms that might be suitable to implement within this model. One interesting algorithm is, for instance, the one proposed by Christensen (1999).

The analysis of the *Green level* often relies on recursive methods. The reason for this is that the *Green level* should not contain any knowledge about how the *Blue level* is generated. This requirement is necessary in an application where the *Blue level* retrieves data from a database that belongs to a different organisation. We can not expect this organisation to modify their database to suit our application. Furthermore this organisation is most likely supplying data to several different applications and we can not burden the database with all kinds of application specific information. This implies that the constructor of an object class receives a seed object and can analyse its surroundings to find patterns of objects that are suitable to form new objects in the *Blue level*. Since the surroundings of the seed object is unfamiliar, recursive methods are very suitable for this analysis.

It should be noted that some objects will be analysed several times during the creation of all objects in an object class and that the *Gr_Dwelling_house* and *Gr_River* classes in the *Green level* can form objects of more than one class in the *Blue level*. To avoid that an object in the *Green level* is used as a seed more than once to create objects in the *Blue level* it is marked with attributes.

4.2 Implementation and evaluation

The model described above has been implemented for three levels. The *Green* level contains information from the GGD database of the Swedish National Land Survey, which is defined at an approximate scale of 1:10 000. The data set covers an area of approximately 60 km^2. The *Blue level*, which is initially empty, exists in two different versions defined at different scales. The two different versions of the *Blue level*, contain essentially the same functionality but the parameters of the object classes are tuned differently for the different scales.

The *Green level* contains the object classes: *Gr_Lake*, *Gr_River*, *Gr_Road* divided into six separate classes, *Gr_Railroad*, *Gr_Road_Bridge*, *Gr_Railroad_Bridge*, *Gr_Stream*, *Gr_Dwelling_House*, *Gr_House_Other*, and *Gr_Church*. The *Blue level* contains the following object classes: *Bl_Lake*, *Bl_River*, *Bl_Road* divided into 5-6 classes, *Bl_Railroad*, *Bl_Road_Bridge*, *Bl_Railroad_Bridge*, *Bl_Built_up*, *Bl_Dwelling_House*, and *Bl_House_Other*. Figure 4.17 shows an example of the *Green level* at a scale that is appropriate to display this information. Figure 4.18 shows the same area in the *Blue level* at a resolution that corresponds to a scale of approximately 1:50 000. Figure 4.19 shows the *Blue level* at a resolution that corresponds to an approximate scale of 1:100 000.

The features are generated essentially in the manner that has been described above. A detailed description is given for each object class to highlight how the different parameters are selected. The object classes are described in the sequence that they are treated in the creation process.

Figure 4.17 A part of the test area for the Green level defined at a scale of 1:10 000.

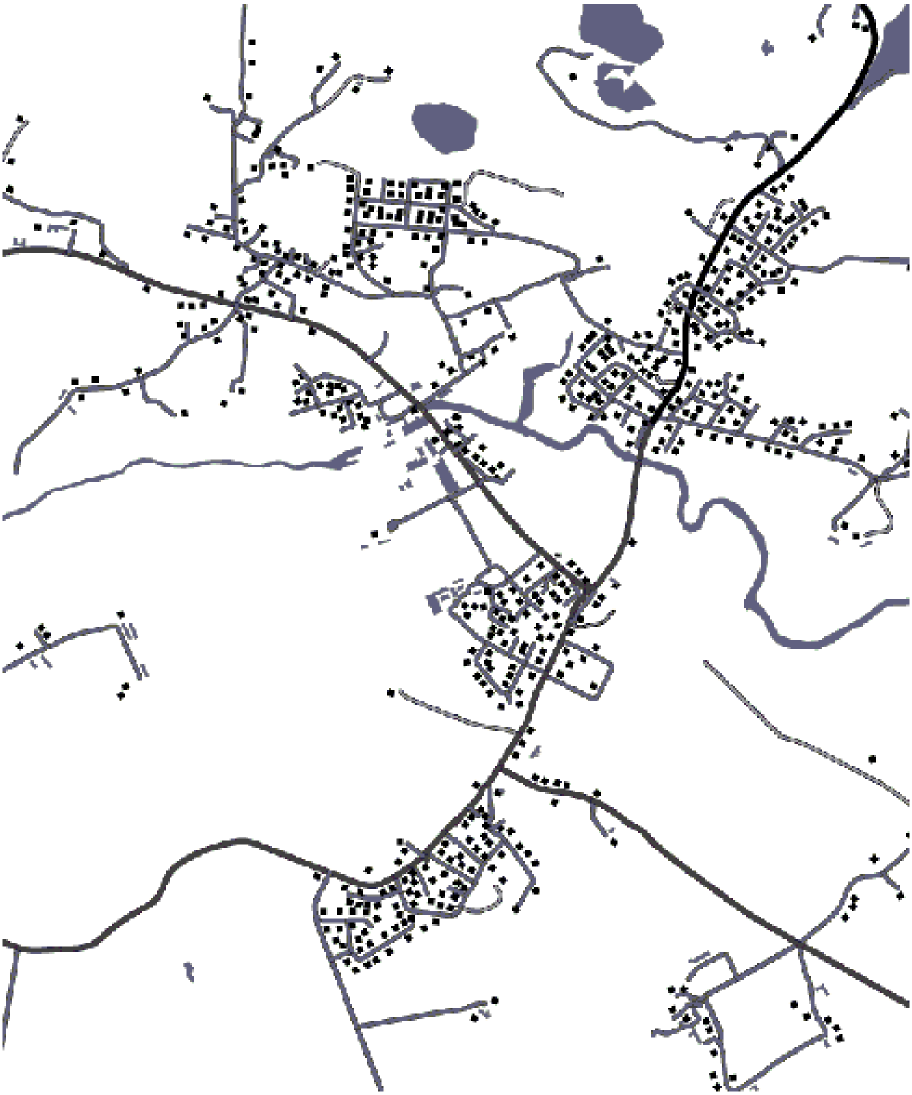

Figure 4.18 A part of the test area for the Blue level defined at a scale of approximately 1:50 000.

Figure 4.19 A part of the test area for the Blue level defined at a scale of approximately 1:100 000.

<u>Lake</u>

Bl_Lakes are created in the *Blue level* if the size of the *Gr_Lake* in the *Green level* is above a certain threshold. At the scale 1:50 000 this threshold is set to 400 m^2 and at scale 1:100 000 it is set to 800 m^2. The simplification of the *Bl_Lake* geometry is achieved by deleting every second vertex of the geometry of the *Gr_Lake*. As has been described above an analysis is made to find an area where a new object may be located in the *Blue level*. Since the simplification of the geometry of the *Bl_Lake* geometry is done in such a simple manner and *Bl_Lakes* are the first objects to be created in the *Blue level* no conflicts occur between different *Bl_Lake* objects.

When simplifying the geometry of the *Gr_Lake* we want to make sure that *Gr_Streams* and *Gr_Rivers* that connect to the *Gr_Lake* are also attached to the *Bl_Lake* in the *Blue level*, if they are members of the *Blue level*. In this implementation it has been chosen to use the topological functionality in LAMPS2 to determine if a particular node or line in the geometry of the *Gr_Lake* is shared with a *Gr_Stream* or *Gr_Rriver*. If it is, this node or line is not modified. A similar analysis is made for *Gr_Rivers*. This assures that *Bl_Streams*, *Bl_Rivers*, and *Bl_Lakes* are connected in the *Blue level*. Figure 4.20 shows how the links of a *Bl_Lake*, a *Bl_Stream* and a *Bl_River* object are connected.

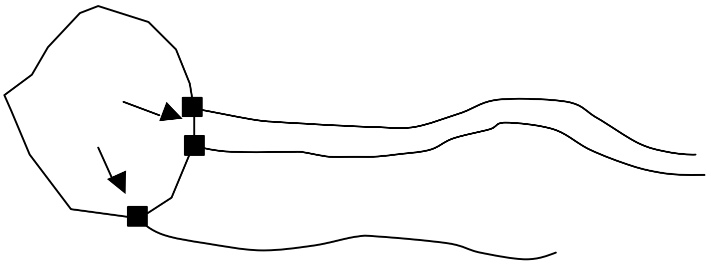

Figure 4.20 Links of a Bl_Lake, a Bl_Stream and a Bl_River object. The squares mark the nodes that divide the geometry into different links. The arrows point at the link that is shared by the Bl_River and the Bl_Lake and the node that is shared by the Bl_Stream and the Bl_Lake.

<u>River</u>

The complexity of annexing and analysing *Gr_River* objects in the *Green level* when creating *Bl_River* objects in the *Blue level* has been described above. The annexation of *Gr_River* objects has not been implemented and each *Gr_River* object is analysed separately. The width of the *Gr_River* object is not measured, and for each *Gr_River* object in the *Green level* a *Bl_River* object in the *Blue level* is created.

As has been described for lakes the separate links of a *Bl_River* object are simplified separately and if a link is shared with a *Bl_Lake* it is not simplified to make sure that the *Bl_Lake* and *Bl_River* objects are connected. In a similar manner a link that is shared a with a *Bl_Railroad_Bridge* or a *Bl_Road_Bridge* is not simplified.

<u>Railroads</u>

The creation of *Bl_Railroads* in the *Blue level* is done in a manner that is similar to how *Bl_Roads* are created, which is described above. In the test area all *Gr_Railroad* objects in the *Green level* are selected to form *Bl_Railroad* objects in the *Blue level*. Since some *Gr_Road* objects that cross the *Gr_Railroad* in the *Green level* do not have a corresponding *Bl_Road* object in the *Blue level*, some *Gr_Railroad* objects have been merged. This will be illustrated further in the next section about roads.

The geometry of the new *Bl_Railroad* object is computed using the area that describes where the new *Bl_Railroad* object may be located in the *Blue level* and the line geometry of the *Gr_Railroad* object(s) in the *Green level*. The line is simplified by picking every second coordinate of the line except the first and the last point. These are kept to make sure that the network structure of linear objects in the map is maintained. For each node, except the first and the last, it is checked if the node is located within the accepted area. If it is located outside the area it is moved to the nearest point of the area. When *Bl_Railroads* are created the *Blue level* only contains *Bl_River* and *Bl_Lake* objects, and in the test area there are no conflicts with these objects.

<u>Roads</u>

Roads are treated as one object class that is divided into 5-6 different types using different attribute values. Each road type is given a different symbolisation. This does not comply with what has been said in chapter 3.3.3 about how object classes shall be formed. Each object class should only be symbolised in one way and therefore each road type should be treated as an object class. However, when *Bl_Roads* and *Bl_Railroads* are created, the topological methods within LAMPS2 are used to find the objects that meet at the end nodes of the objects. Since the topological methods in LAMPS2 do not support inheritance, this means that treating the road types as separate object classes is rather cumbersome. For simplicity I have chosen to treat them as one class.

The creation process for *Bl_Roads* has been described in chapter 4.1 and the simplification of the geometry is the same as for railroads above. However, since the *Blue level* now is becoming more populated, we can see how conflicts occur with other *Bl_Road* objects as well as with *Bl_Railroad* objects. Figure 4.21 shows an example where the method works quite well. A *Bl_Road* runs along a *Bl_Railroad* and then crosses the *Bl_Railroad*. There are no other objects in the area that can create conflicts. The *Bl_Road* has been moved so that the symbols are not overlapping each other.

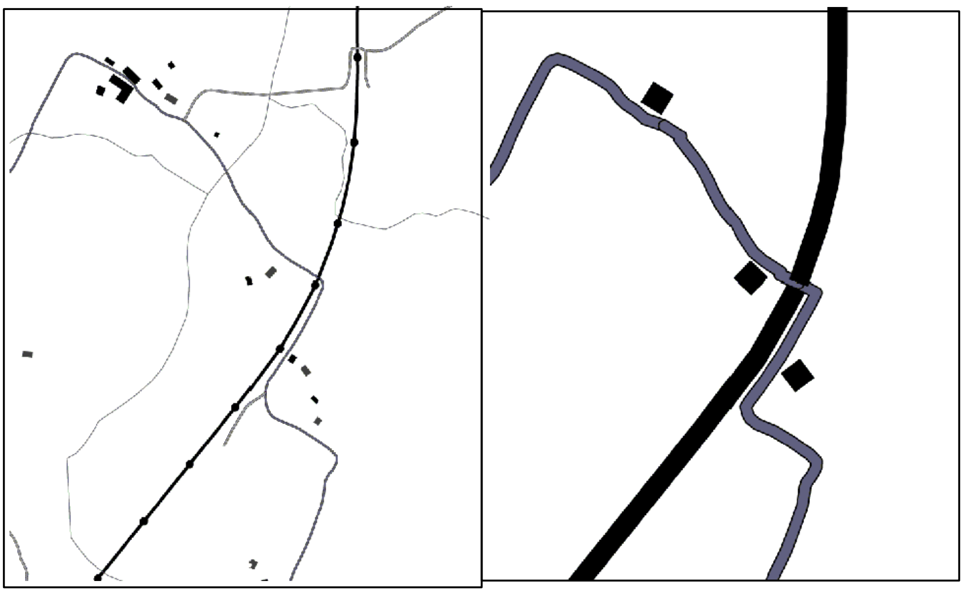

Figure 4.21 The top map shows the Green level and the map below shows the small scale version of the Blue level. The Bl_Road that runs along the Bl_Railway has been moved so that the symbols are not overlapping. The node where the road crosses the railway has been maintained properly. The roads that are not shown in the blue level are dirt tracks which are not shown at all at this scale. The building next to the road and the railway has also been moved. The choice of buildings for the Blue level will be described further below.

Figure 4.22 shows a similar case where a *Bl_Road* has been moved to not have conflicts with another *Bl_Road*. Here we can also see an example where the *simplify()* method is not sophisticated enough. The symbol size of the first object infringes on the area where the new object may be located to such an extent that the second point of the geometry is displaced irregularly.

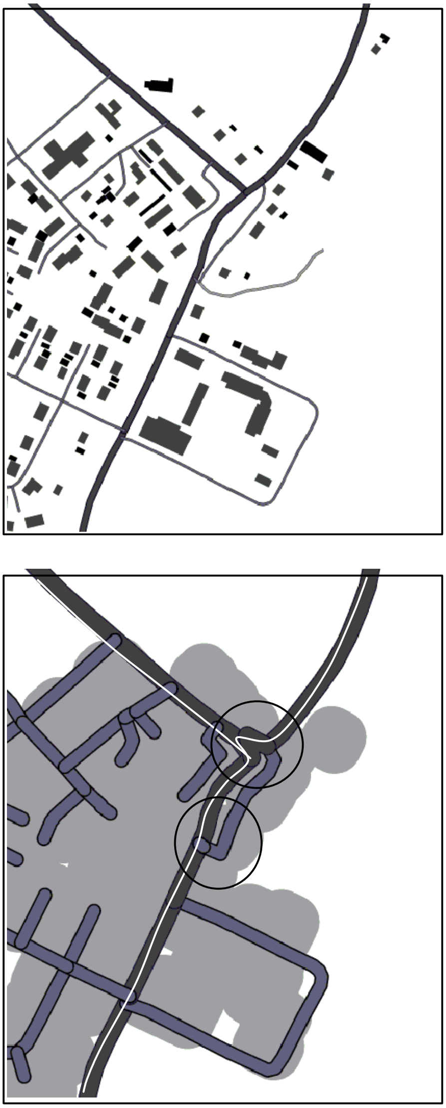

Figure 4.22 The top map shows the Green level and the map below shows the small scale version of the Blue level. The lower circle illustrates how the Bl_Road has been moved to avoid conflicting symbols. The

upper circle illustrates a how the second point of a Bl_Road link has been moved irregularly due to a conflict with a symbol of the neighbouring Bl_Road.

Another problem with *Bl_Robjects* is that junctions are not simplified. Figure 4.23 shows an example of a junction that is rather messy.

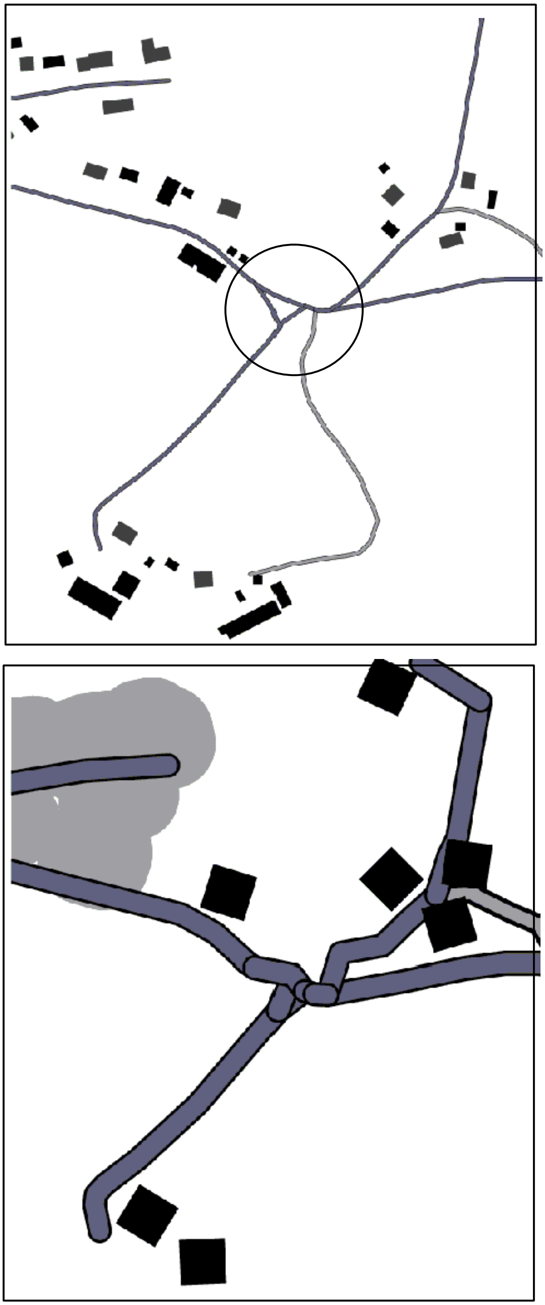

Figure 4.23 The top map shows the Green level and the map below shows the small scale version of the Blue level. There is no proper method to treat junctions like the one in the circle above. All links in the junction are maintained but are stacked on top of each other.

The parameters of the *select()* method are tuned differently at the small scale version and the large scale version of the *Blue level*. In the large scale version the dirt track road type is analysed according to the criteria described above. If it is a cul-de-sac shorter than 2000m that does not lead to a *Bl_Lake* or a *Bl_Dwelling_House*, no *Bl_Road* object is formed in the *Blue level*. For all other road types in the *Green level Bl_Road* objects are formed in the *Blue level*. In the small-scale version of the *Blue level* dirt tracks are not shown at all. The analysis described above is done for forest roads and all other roads are represented in the *Blue level*.

In the test data set there is no road type that correspond to the street class described in chapter 4.1 above. The sample maps shown above contain streets, but they belong to a road type that also contains minor roads.

Built up area

In the test area the *Green level* does not contain any *Gr_Built_up* objects and the large scale version of the *Blue level* is designed in such a manner that no *Bl_Built_up* objects are created. In the small scale version of the *Blue level*, however, the building symbols have to have such a size that they can not represent densely populated areas. A *Bl_Built_up* object has to be created instead. Since there are no built up areas in the test data set the method is rather simple. *Gr_Dwelling_House* objects are annexed recursively in the manner that is described above. The parameter for the annexation have been set so that the size of a *Gr_Dwelling_House* has to be larger than $90m^2$ to be selected for the *Blue level*. The distance between two *Gr_Dwelling_Houses* to be annexed has to be shorter than 40m. When the annexation is finished we have to have a group of at least 20 *Gr_Dwelling_Hous*es to form a *Bl_Built_up* object. The *Bl_Built_up* is formed by first creating a *Bl_Built_up* around each *Gr_Dwelling_House* using a buffer method. When buffers have been created for all selected *Gr_Dwelling_Houses*, all buffers that are overlapping are merged into one *Bl_Built_up* object and the small buffer objects are deleted. It would also be possible to delete the "islands" within the *Bl_Built_up* objects but this has not been implemented. Figure 4.19 shows how the built-up areas are shown in the small scale version of the *Blue level*. Of course this is far from the quality of a topographic map.

Buildings not used for dwelling

This object class will be called *Bl_House_Other*. If a *Gr_House_Other* in the *Green level* is suitable to form a *Bl_House_Other* in the *Blue level* depends on two requirements. The *Gr_House_Other* has to be larger than a certain area, 600m^2 in the small scale version of the *Blue level* and 200 m^2 in the large scale version. Furthermore, the *Gr_House_Other* should not be located within a *Bl_Built_up* object in the *Blue level*. If this is fulfilled an area where the *Bl_House_Other* object may be located in the *Blue level* is computed using the buffer method. The area is then modified so that all objects that can have conflicts with the *Bl_House_Other* are located outside the buffer. The geometry of the *Gr_House_Other* in the *Green level* is copied and the method tries to find a position for the geometry that is completely within the allowed area. However, since a building might be surrounded by roads this could be impossible. The method tries to move the geometry in eight different directions (North, Northeast, West etc.) and tests if the geometry is inside the area. If none of the alternatives is accepted the geometry is created at its original position anyhow. The actual geometry of the *Bl_House_Other* is not modified. Figure 4.24 shows an example of *Bl_House_Other* objects. Some *Bl_House_Other* objects have been moved while others stay put since they have no conflicts.

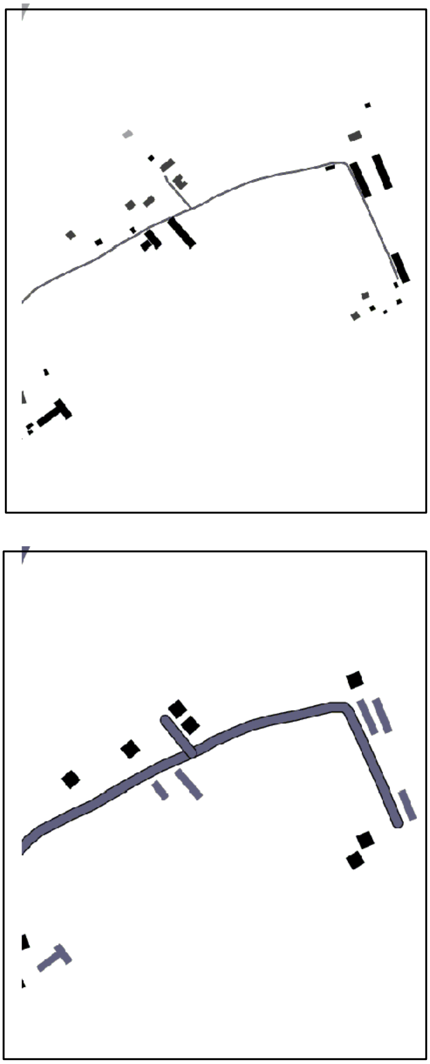

Figure 4.24 The top map shows the Green level and the map below shows the large scale version of the Blue level. The Bl_House_Other objects that are close to the Bl_Road have been moved to avoid conflicts.

In Figure 4.21 it is possible to see how a subset of buildings is selected. All the *Gr_House_Other* objects in this map are too small to be selected for the *Blue level*. The *Bl_Dwelling_Houses* however are represented with symbols if they are larger than 90 m^2.

Dwelling-houses

The creation of dwelling houses complies with what has been described in chapter 4.1. When the small scale version of the *Blue level* is created the *Gr_Dwelling_Houses* in the *Green level* that have been used as seeds to create *Bl_Built_up* objects are now marked and can not be used to create *Bl_Dwelling_Houses*.

When we reach the *simplification()* method we have a group of *Gr_Dwelling_Houses* from the *Green level* that shall be used to form a group of *Bl_Dwelling_Houses* in the *Blue level* and an area where the *Bl_Dwelling_Houses* may be located. First we move the *Bl_Dwelling_Houses* in a similar manner as has been described for *Bl_House_Other* above to try to get it within the accepted area. Then we check if there is a conflict with any *Bl_Dwelling_House* in the *Blue level* that has already been created. If not, the *Bl_Dwelling_House* in the *Blue level* is created. Figure 4.25 shows examples of *Bl_Dwelling_Houses* in the different versions of the *Blue level*.

Figure 4.25 The top map shows three Green level and the map below shows the large scale version of the Blue level. A subset of the houses has been selected but the displacement algorithm is obviously not sophisticated enough in dense areas. In the area marked by a circle the buildings form a regular pattern. This pattern is lost since one of the buildings is displaced.

5 Discussion and Concluding Remarks

5.1 Summary

Much of the research in automatic cartographic generalisation focus on issues such as the development of algorithms, or generalisation of a certain feature type. The aim of the research is to replicate maps created by cartographers automatically. The work presented in this thesis takes a different viewpoint and aims at creating a method that automatically generates views to a geographical database. In most cases these views are simpler than traditional maps since they are created for a particular application or user group. However, there is a growing need for this kind of maps, for instance when information from different databases is combined and distributed over the Internet to respond to different queries. The generation of these maps has to be fully automatic.

This thesis presents how such views, that are automatically created, can be designed using object oriented modelling. The design of the model is not based on any particular application discipline but rather on the following:

- <u>Human cognition</u> – A map consists of categories. The categories used by humans are fuzzy and change meaning with context while the object classes used in object oriented modelling are crisp. The same is true for relations between different categories. When creating a view we have to allow for both crisp and fuzzy object classes. Secondly, humans have a great capability to extract knowledge from visual patterns. There is a risk that this knowledge is neglected in object oriented modelling. The aim here has been to keep this knowledge in mind when creating the object oriented model.

- <u>Theory in object oriented modelling</u> – An object oriented model is created to highlight certain aspects of reality that are important for an application. When doing this, other aspects of reality are neglected. When a view is designed using object oriented modelling it highlights what is relevant for a particular user group and ignores other aspects of the geographical reality.

- General requirements of a view. – A view can be used for both visual analysis and to perform different queries such as route finding or distance measuring. In a particular view we can have some object classes, such as roads, that are suitable for analytical queries, and other classes such as houses, that are merely displayed to show housing patterns in the landscape. Since a view has zooming capabilities the information displayed in the small scale can be less detailed than in a paper map. If the user needs more details of a particular area he zooms in. It is possible to point at objects in a view and retrieve their attributes. This gives us the ability to merge conflicting symbols of different classes in a manner that is not suitable in a paper map. By clicking at the symbol that represents different classes, or zooming in, we can see what it represents.

- View design – A view consists of several different levels defined at different scales. To design a level is an iterative procedure. A choice of: object classes, symbolisation, scale and generalisation method and parameters for each object class are made. These different parameters are modified until an acceptable solution is acquired. From this perspective, generalisation is an inseparable part of map design.

The functionality to fill an empty level with geographical information is built into the constructors of the object classes of the level. The constructors retrieve and analyse information from different sources and create new objects in the level. The model presented in the case study contains a set of typical categories in geographical information. Relations between the different object classes in the level as well as relations to the object classes of the data source, are modelled. The model contains descriptions of how the constructors of the different object classes work. The constructors focus on retrieving and analysing information as well as locating new objects in the level topologically correct and where there is enough space. Little effort has been put into how the geometry of the new objects shall be simplified.

5.2 Contributions of the study

A method is presented to create object oriented models of geographical information that contain functionality for automatic generalisation. Object oriented models of geographical information can be created in many different ways and the approach taken here is new in the sense that the design of the model is based on the prototype theory about categories rather than theories from an application discipline. Furthermore the modelling method is focused on the information presented to the user rather than how a database ought to be designed.

5.3 Future Research

In chapter one it is stated that his method is mainly to be used to create simple views into a geographical database. What simple means is not elaborated on further, and it is not known when the complexity of a map makes this method unsuitable. To evaluate if the modelling method is suitable, it has to be tested in different applications. When this is done details of the creation process will have to be modified and refined to suit the needs of the different applications.

When a new level is filled with information the object classes are worked through one at a time and all objects of the class are created. Once an object is created it can not be modified. This approach is not sophisticated enough to solve complex generalisation problems and there is a need to develop methods that generalise an area that is dense with objects.

The generalisation procedure implemented in the model is focused on propositional knowledge. The knowledge a cartographer can acquire when he sees patterns of geographical features is not used in the model. Further research is needed on how to acquire such knowledge and use it in object oriented modelling.

When new objects are created in the model, the area where a new object may be located is calculated. This geometry and the geometry of the seed object are used to calculate the geometry of the new object. So far, very little effort has been put into finding which algorithms are suitable to perform this calculation. Additional research is needed to study how different algorithms can be used within this model and if the model needs to be modified to suit the algorithms.

References

Ahlgvist, O., and Arnberg, W., (1998) A general theory of spatial organisation in a GIS. In *Context Sensitive Digitisation of Geographical Information.* Licentiate Thesis, Stockholm University.

Baeijs, C., Demazeau, Y., and Alvares, L. (1996) SIGMS: Application of Multi-Agent Systems to Cartographic Generalization. *Proceeding of 7th European Workshop on Modelling Autonomous Agents in a Multi Agent World,* Springer, pp. 163-176.

Berlin, B., Breedlove, D. E., Raven, P. H. (1974). *Principles of Tzeltal Plant Classification.* New York: Academic.

Berlin, B. and Kay, P. (1969). *Basic Color Terms Their Universality and Evolution.* Berkely: University of California Press.

Bertin, J. (1983) *Semiology of Graphics,* Madison: The University of Wisconsin Press.

Board, C. (1967). Maps as models. In R. J. Chorley and P. Hagger (Eds.), *Models in geography* (pp. 671-725). London: Methuen.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide.* Reading, Massachusetts: Addison-Wesley Longman.

Brassel, K. E., and Weibel, R. (1988). A review and Conceptual Framework of Automated Map Generalization. *International Journal of Geographical Information Systems,* Vol. 2, No. 3, pp.229-244.

Brown, R. (1958). How shall a thing be called? *Psychological Review* 65:14-21.—— . (1965). *Social Psychology.* New York: Free Press.

Chase, W. G., and Simon, H. A. (1973). The mind's eye in chess. In W. Chase (Ed.), *Visual Information Processing* (pp. 215-281). New York: Academic Press.

Christensen, A., H., J., (1999). Cartographic Line Generalization with Waterlines and Medial-Axes. *Cartography and Geographic Information Science,* Vol. 26, No1, pp19-32.

Date, C. J., (1990). *An introduction to Database Systems,* Vol. 1. 5th edn., Reading, MA: Addison-Wesley Publishing Company.

Golledge, R. G., Gale, N., Pellegrino, J. W., and Doherty, S. (1992). Spatial knowledge acquisition by children: Route learning and relational distances. *Annals of the Association of American Geographers,* 82(2), 223-244.

Harrie, L. (1998) *Generalisation Methods for Propagating Updates between Cartographic Data Sets*. Licentiate Thesis. Lund institute of technology, Lund.

Harrie, L. (1999) The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. *Cartography and Geographic Information Science,* Vol. 26, No1, pp55-69.

Jones, C. B., Bundy, G. L., and Ware, M. J. (1995) Map generalization with a Triangulated Data Structure, *Cartography and Geographic Information Systems,* Vol. 22, No 4, 1995, pp. 317-331.

Jones, C. B., Kidner, D. B., Luo L. Q., Bundy G. Ll., and Ware M. J. (1996) Database design for a multi-scale spatial information system, *International Journal of Geographical Information Systems, vol. 10, No. 8, pp. 901-920.*

Kay, P. and McDaniel, C. (1978) The Linguistic Significance of the Meanings of Basic Color Terms. *Language* 54, no 3, 610-46.

Kilpeläinen, T., (1997). *Multiple Representation and Generalization of Geo-Databases for Topographic Maps.* Doctorate thesis, Publications of the Finnish Geodetic Institute, No. 124

Kolácný, A. (1969). Cartographic information— a fundamental concept and term in modern cartography. *Cartographic Journal,* 6, 47-49.

Lakoff, G. (1987). *Woman, Fire and Dangerous Things: What Categories Reveal about the Mind*. Chicago: University of Chicago Press.

Laserscan (1999) *Gothic release 3.2 Online Documentation.* Cambridge, UK.

Lind, J., (1997), *Länken mellan skapare och användare av kartor i GIS (in Swedish)*, M.Sc. Thesis, Dept of Physical Geography, Stockholm University.

Lounsbury, F. (1964). A Formal Account of the Crow- and Omaha-Type Kinship Terminologies. In W. H. Goodenough, ed., *Explorations in Cultural Anthropology,* pp. 212-54. New York: Holt, Reinhart & Winston, 1969.

MacEachren, A. M. (1995). *How Maps Work*. New York: The Guilfoord Press.

Mackaness, W. A. (1995a). A Constraint Based Approach to Human Computer Interaction in Automated Cartography. *Proceedings of the 17th International Cartographic Conference,* Barcelona, pp1423-1432.

Mackaness, W. (1995b). Analysis of Urban Road Networks to Support Cartographic Generalization. *Cartography and Geographic Information Systems,* Vol. 22, No. 4, pp. 306-316.

Marr, D. (1985). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* San Francisco: W. H. Freeman.

Molenaar, M (1998). *An Introduction to the Theory of Spatial Object Modelling for GIS.* London: Taylor & Francis.

Müller, J. C., Weibel, R., Lagrange, J. P. 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,* Gisdata 1, ESF, London: Taylor & Francis, pp. 56-69.

Nordbeck, S. (1969): *Koordinatsatta data och automatisk tätortsavgränsning (In Swedish)*. Rapport från Byggforskningen 31/69. Stockholm 1969.

Petzold, I., Plümer, L., and Heber, M. (1999) Label placement for dynamically generated screen maps. *Proceedings of the 19th International Cartographic Conference,* Ottawa, Session 25 B.

Pinker, S. (1984). Visual cognition: An introduction. *Cognition,* 18, 1-63.

Plazanet, C., Affholder, J-G., and Fritsch, E. (1995). The Importance of Geometric Modeling in Linear Feature Generalization. *Cartography and Geographic Information Systems,* Vol. 22, No. 4, pp291-305.

Richardsson, E. D., and Muller, J-C. (1991) Rule selection for small-scale map generalization. In Buttenfield, B. P., and R.B. McMaster, *Map Generalization: Making Rules for Knowledge Representation,* Longman Group, pp 136-149.

Robinson, H. A. (1952). *The Look of Maps.* Madison: University of Wisconsin Press.

Robinson, H. A., Morrison, J. L., Muehrcke, P. C., Kimerling, A. J., and Guptil, S. C. (1995). *Elements of Cartography, 6th ed.* Wiley, New York.

Rosch, E. (1973). Natural Categories. *Cognitive Psychology,* 4, 328-350.

Rosch, E. (1975). Cognitive reference points. *Cognitive Psychology*, 7, 532-547.

Rosch, E. (1977). Human categroization. In N. Warren (Ed.), *Studies in Crosscultural Psychology* (pp. 27-48). London: Academic Press.

Rosch, E. (1978). Principles of categorization. In E. Rosch and B.B. Lloyd (Eds.), *Cognition and categorization* (pp. 27-48). Hillsdale, NJ: Erlbaum.

Rosch, E., Mervis, C., Gray, W., Johnson, D., and Boyes-Brae, P. (1976). Basic objects in natural categories. *Cognitive Psychology,* 8, 382-439.

Ruas, A., 1998. And Plazanet, C. (1996). Strategies for Automated Generalization. In Kraak, M.J., and M. Molenaar (eds), *Advances in GIS Research II,* Taylor and Francis, London, pp. 319-336.

Rumbaugh J., Blaha M., Premerlani W., Eddy F. and Lorenson W. (1991) *Object oriented modelling and design* London: Prentice Hall.

Rumelhart, D. E., and Norman, D. A. (1985). Representation of knowledge. In A. M. Aitkenhead and J. M. Slack (Eds.), *Issues in Cognitive Modeling* (pp. 15-62). London: Erlbaum.

Schylberg, L. (1993). *Computational Methods for Generalization of Cartographic Data in a Raster Environment.* Doctorate thesis, Department of Geodesy and Photogrammetry, Royal Institute of Technology, Stockholm, Sweden, TRITA-FMI Report !993:7.

Schneider, W., and Shiffrin, R. M. (1977). Controlled and automatic human information processing: 1. Detection, search, and attention. *Psychological Review,* 84, 1-66.

Sjödin, M. and Strid, A. (2000) *Automatisk generalisering med objektorienterad kartdatabas.* M.Sc. Thesis. Royal Institute of Technology, Stockholm.

Statistics Sweden, (2000). *Tätortsdefinitionen* (in Swedish). Internet: http://www.scb.se/statinfo/pbesk/mi0801.asp (25 January 2000).

STG, (1998). *Tekniskt ramverk för geografisk information (*in Swedish). Stockholm: SIS Förlag.

Weibel, R., (1995) Three Essential Building Blocks for Automated Generalisation. In Müller, J. –C., Lagrange, J. –P., and Weibel, R., (eds.), *GIS and Generalization,* Gisdata 1, ESF, Taylor & Francis, London, pp. 56-69.

Wittgenstein, L. (1953). *Philosophical Investigations.* New York: Macmillan.

Worboys, M. F. (1995). *GIS: A Computing Perspective.* London: Taylor&Francis.

Zadeh, L. (1965). Fuzzy sets. *Information and Control* 8:338-53.

TRITA-GEOFOTO serien utges av:

Kungl Tekniska Högskolan (KTH)
Avdelningen för Geodesi och Geoinformatik
100 44 STOCKHOLM
Drottning Kristinas väg 30
Telefon: 08-790 7342
Telefax: 08-790 7343
URL: http://www.geomatics.kth.se/

The TRITA-GEOFOTO series is published by:

Royal Institute of Technology (KTH)
Division of Geodesy and Geoinformatics
SE-100 44 STOCKHOLM
Sweden
Phone: +46 8 790 7342
Fax: +46 8 790 7343
URL: http://www.geomatics.kth.se/

TRITA-GEOFOTO-serien 2000 - The TRITA-GEOFOTO series, 2000

2000:1 Carballo, Gabriel. Statistically-based multiresolution network flow phase unwrapping in SAR interferometry. January 2000. Photogrammetric Reports No 68. (Doctoral Dissertation)

2000:2 Jonas Andersson och Jens Hedlund. Geografiska analyser över Internet. Januari 2000. (Examensarbete i geoinformatik. Handledare: Roslund)

2000:3 Thomas Rehders. Noggrannhetsstudier vid RTK-mätning och kvalitetsundersökning av GPS-mottagare. Februari 2000. (Examensarbete i geodesi nr 3065. Handledare: Horemuz)

2000:4 Åsa Hesson. Publicering av historiska kartor på Internet. Februari 2000. (Examensarbete i geoinformatik. Handledare: Hauska)

2000:5 Malin Skagerström och Karin Wiklander. Extension of a control network in dar es salaam using GPS-technology. Februari 2000. (Examensarbete i geodesi nr 3066. Handledare: Stoimenov)

2000:6 Maria Sjödin och Anna Strid. Automatisk generalisering med objektorienterad kartdatabas. Februari 2000. (Examensarbete i geoinformatik. Handledare: Hauska)

2000:7 Per Koldestam. IT-projektet Ortnamnslinjen. Mars 2000. (Examensarbete i geoinformatik. Handledare: Hauska).

2000:8 Anna Källander och Elin Söderström. Deformationsanalys i horisontal- och vertikalled på Slussen-konstruktionen. Mars 2000. (Examensarbete i geodesi nr 3067. Handledare: Horemuz och Lewen)

2000:9 Anna Haglund. Towards soft classification of satellite data - A case study based upon Resurs MSU-SK satellite data and land cover classification within the Baltic Sea Region. Mars 2000. (Examensarbete i geoinformatik. Handledare: Roslund och Langaas).

2000:10 Wenke Sun and Lars Sjöberg. Tidal Effects on Sea Bottom Positioning by Combining GPS and Sonar Observations. March 2000.

2000:11 Helene Burman. Calibration and Orientation of Airborne Image and Laser Scanned Data Using GPS and INS. April 2000. Photogrammetry Reports No 69. (Doktorsavhandling)

2000:12 Martin Larbo. Vehicle Navigation - Market Research and Evaluation of Existing Systems.
April 2000. (Examensarbete i geoinformatik. Handledare: Roslund och Höglund).

2000:13 Patrik Simonsson. Internet som medium för tillgång till geografisk information. Tillämpning inom skogsbranschen. En förstudie till en interaktiv marknadsplats för GIS-produkter. (Examensarbete i geoinformatik. Handledare: Hauska).

2000:14 Fredrik Ahlman. Automatisk styrning av vägmaskiner med GPS, laser och totalstation. Maj 2000. (Examensarbete i geodesi nr 3068. Handledare: Horemuz)

2000:15 Niklas Svanholm. Jämförelse av olika metoder att föra över kartdetaljer till ett nytt koordinatsystem. Maj 2000. (Examensarbete i geodesi nr 3069. Handledare: Engberg och Fan)

2000:16 Jerker Boman. RTK at long distances with a dual frequency GPS/GLONASS receiver. Maj 2000.
(Examensarbete i geodesi nr 3070. Handledare: Horemuz)

2000:17 Susanne Reinicke. GPS deformation measurement in controlled environment. May 2000. (Examensarbete i geodesi nr 3071. Handledare: Sjöberg och Horemuz)
2000:18 Anders Hedelius. The Swedish Space Industry in a European Context. (Examensarbete i fotogrammetri nr 116. Handledare: Torlegård)
2000:19 Anna Widström. Uppbyggnad av markdatabaser vid topografisk kartframställning. Metodutveckling, erfarenheter och användningsområden. (Examensarbete i fotogrammetri nr 118. Handledare: Boberg).
2000:20 Camilla Mahlander. Detection of geographic changes and location of sites supported by fuzzy set theory. September 2000. Photogrammetry Reports No 70. (Licentiatavhandling)
2000:21 Birgitta Lindqvist. Size, compression rate and quality of remote sensing data for analyses and Internet presentation. September 2000. (Examensarbete i fotogrammetri nr 117. Handledare: Torlegård)
2000:22 Jeanette Öhman. GIS inom transportbranschen. Oktober 2000. (Examensarbete i geoinformatik. Handledare: Roslund)
2000:23 Huaan Fan. Theoretical Geodesy. Division of Geodesy Report No 2016. October 2000. (compendium).
2000:24 Niklas Ringdahl, Susanne Roman. Analys av metoder för GIS på Internet. November 2000. (Examensarbete i geoinformatik. Handledare: Hauska).
2000:25 Lars Nyhed och Fredrik Jansson. Bearbetning av geografisk information från olika källor. November 2000. (Examensarbete i geoinformatik. Handledare: Hauska).
2000:26 Jesper Svedberg. Autonom mätalgoritm för DGPS-mätning. November 2000. (Examensarbete i geodesi nr 3073. Handledare: Horemuz och Johnson)
2000:27 Andreas Larsson. Automatisk generalisering av samlingskarta. December 2000. (Examensarbete i geoinformatik. Handledare: Hauska).
2000:28 Malin Wallin. Interpolation med bèzier kurvor. December 2000. (Examensarbete i geoinformatik. Handledare: Hauska).
2000:29 Johan Svensson. Analys och utveckling av en geokodningsmodul. December 2000. (Examensarbete i geoinformatik. Handledare: Hauska).

TRITA-GEOFOTO-serien 2001 - The TRITA-GEOFOTO series, 2001

2001:1 Herbert Henkel. Remote Sensing and Image Analysis (2nd edition). January 20001. (kompendium)
2001:2 Anders Boberg. Introduktion till fotogrammetrin (Femte omarbetade upplagan). January 2001. (kompendium)
2001:3 Kerstin Hegner och Karin Pettersson. Utformning av kartor på mobila terminaler. Januari 2001. (Examensarbete i geoinformatik. Handledare: Hauska).
2001:4 Maria Gelinder och Helena Nyman. GIS på intranät. Februari 2001. (Examensarbete i geoinformatik. Handledare: Roslund).

2001:5 Kennert Torlegård (ed.). Proceedings of OEEPE workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Elevation Models, 1-3 March 2001.

2001:6 Martin Ehrlén och Magnus Lindqvist. Transformation av koordinatbestämda objekt till linjärt referenssystem med kvalitetssäkrad metodik för Vägverkets Vägdatabank. (Examensarbete i geodesi nr 3074. Handledare: Sjöberg och Horemuz).

2001:7 Petronella Westling. Interoperable Location Types within the Area of Location-Based Services. Mars 2001. (Examensarbete i geoinformatik. Handledare: Hauska).

2001:8 Ivar Andersson. Geodetiska mätningar vid station Wasa, Antarktis 1999 – 2000. Mars 2001. (Examensarbete i geodesi nr 3075. Handledare: Sjöberg och Asenjo).

2001:9 Carina Borg. Geografiskt informationssystem i en handdator. Mars 2001. (Examensarbete i geoinformatik. Handledare: Hauska).

2001:10 Annika Pettersson Björnström. Automatiserad koordinatmätning av fornminnen. Mars 2001. (Examensarbete i geoinformatik. Handledare: Hauska).

2001:11 Patrik Öhrström. Investigating the Feasibility of An OpenGis GeoBox Prototype. Mars 2001. (Examensarbete i geoinformatik. Handledare: Hauska).

2001:12 Stig-Göran Mårtensson. Height Determination by GPS – a Practical Experiment in Central Sweden. Maj 2001. Division of Geodesy Report 1054. (Doktorsavhandling)

2001:13 Jonas Ågren. Processing of the 1992, 1994 and 1997 Campaigns on the Northern GPS Deformation Traverse. Maj 2001. Division of Geodesy Report 1055. (Licentiatavhandling)

2001:14 Karin Andersson. Kvalitetsbestämning av Mobile Mapping teknik för insamling av grunddata till GIS- och vägdatabaser. Juni 2001. (Examensarbete i fotogrammetri nr 119. Handledare: Torlegård).

2001:15 Charlotte Geijer och Camilla Kristiansson. Development of a location-based WAP guide with text and voice interface. Juni 2001. (Examensarbete i geoinformatik. Handledare: Hauska).

2001:16 Per-Anders Olsson. Hitta optimala placeringar av ambulanser med ett GIS. Augusti 2001. (Examensarbete i geoinformatik. Handledare: Hauska).

2001:17 Patrik.Ottoson. Geographic Indexing and Data Management for 3D-Visualisation. Doctoral dissertation. September 2001.

2001:18 Mats Dunkars. Automatic generation of a view to a geographical database. Licentiate thesis. September 2001.

2001:19 Martin Werner. Pattern recognition: object detection and localization using multiple pan-tilt-zoom cameras. Master of Science Thesis in photogrammetry. September 2001. (Examensarbete i fotogrammetri nr 120. Handledare: Jan-Olof Eklundh och Takahito Kawanishi)

2001:20 Anders Åkerberg. Analys av amplituddata från ELAC multibeamekolod. September 2001. (Examensarbete i geodesi nr 3076. Handledare: Lars Sjöberg)