DESIGN AND IMPLEMENTING OF AUTOMATICAL GENERALISATION IN A NEW PRODUCTIONENVIRONMENT FOR DATASETS IN SCALE 1:50.000 (- and 1:100.000)

Marianne Bengtson
National Survey and Cadastre of Denmark
Rentemestervej 8, DK-2400 Copenhagen NV.
mjb@kms.dk

Abstract

At the National Survey and Cadastre of Denmark (KMS) the topographical map production have until the end of this year been based on a traditional basis – raster data from scanned map sheets. Parallel with the traditional map production KMS have during the last couple of years carried out a project, which was assigned to establish a new topographical vector database in scale 1:10.000 (TOP10DK), covering Denmark. The Database will be storing the entire Denmark at the end of year 2000. KMS wanted to base the future production of the topographical maps on the TOP10DK database, so for that reason KMS decided at the beginning of year 1999 to start-up a project, MTK – Modernisation of the Topographical Maps. The aim of the project were to have a new production system ready at the beginning of year 2001 for production of raster map sheets in scale 1:50.000 based on the topographical database TOP10DK.

The paper will go through the experiences we have gathered during the project with the work of designing and implementing a new production system for the topographical maps.

Introduction

By using TOP10DK as the basic dataset for the future production of respectively standard map series and custom products, KMS will get some effective production workflows, where the updating of the basic data for the different production lines only will take place in the TOP10DK. To be able to carry out the updating of the derived datasets on the basic of the updating of TOP10DK, KMS will have to implement updating processes to handle this. Development and implementation of new processes for the updating is expected to start up during this year.

Besides the requirement of only updating one basic dataset, there was another requirement to be handled in the MTK project. Development and implementation of automatically generalisations processes in the production system, so the work with the manual generalisation of the derived map products could be reduced.

TOP10DK, basic dataset for the production.
TOP10DK is a well-defined and well-structured topographical nationwide vector dataset in a resolution of 1 metre or better. All TOP10DK registrations are made on the basis of new aerial photographs and a subsequent field examination. TOP10DK have a very strict specification both for what is to be registered and how it is to be registered, and the registrations are all subject to a strict completeness check and topological check to ensure that the rules laid down are complied with.

The TOP10DK is updated every 5 years, which means that new aerial photographs will be taken and a new field examination will be performed on the basis of which the vector data set will be updated.

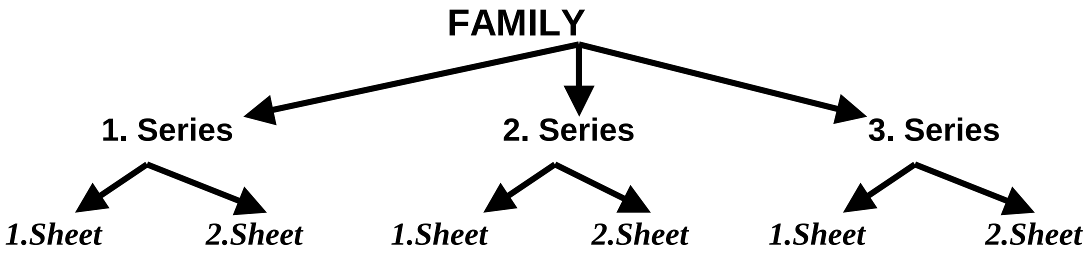
Establishment of Production system

When the design phase started up, 5 main requirements were defined for the new production system:

1. Requirement of using Oracle 8i to store the geometries for the derived datasets.
2. There should be an open interface between the databases and the software tools to be used for the production.
3. The central database for the derived datasets in 1:50.000 and 1:100.000 should be the *main key* of the production system – from this database the production lines shall receive data for the production.
4. The software tools we decided to use for the generalisation and the editing should be able to support selected part of the TOP10DK data model.
5. The software tools (Intergraph) and the production processes, which have been used for the production of paper maps in the earlier production system, should be retained.

As one of the sub goals for the MTK, was to have a 1.version of the production system ready for the production of a number of paper maps in 1:50.000 at spring time of year 2001, we focused on the specific requirements for the production of this map product.

Design of the Central Database (Oracle)
In the database the geometries for the topographical map products are structured according to the following object model:



The **Family** contains derived topographical geometries and geometries for name placement according to a defined main scale. These geometries can be used as basic for further manipulation/generalisation in the production of vector and raster products in this main scale.

The **Series** contains derived topographical geometries and geometries for name placement that either are geometries inherited directly from the Family or/and are geometries, which have been further manipulated/generalised for this Series.

The **Sheet** contains only topographical geometries inherited from the Family or the Series. It can contain it own geometries for name placement.

Data model

As basis for the work of defining a data model for the derived map products, we used the TOP10DK data model. Using this data model require that the database can support cut-outs in areas, comment geometry between area- and line geometries, the 3. and 4. Coordinate for storing z values and origin per point, start- and endpoint for object.

Origin per point in TOP10DK - lineage information is attached to each point in each object, which makes it possible to see when, with which instrument, with which accuracy, by which operator, etc. the point was measured. With this information it is possible to check which points have been newly measured and which stemmed from previous measurement.

As it is a problem to handle origin pr. point in any commercial system, we decided not to bring it with us into the derived dataset. A solution for the derived dataset could be to implement an origin per object, which tells what kind of generalisation the object has been through.

Establishment of references between TOP10DK and the derived datasets

For the possibility to update the derived datasets when the TOP10DK have changed, it can be expedient to have references between the derived object and the TOP10DK object(s). Therefore the database was designed to store these references attached to the derived objects during the generalisation.

Implementation of generalisation- and editing tools.

Tools for Generalisation

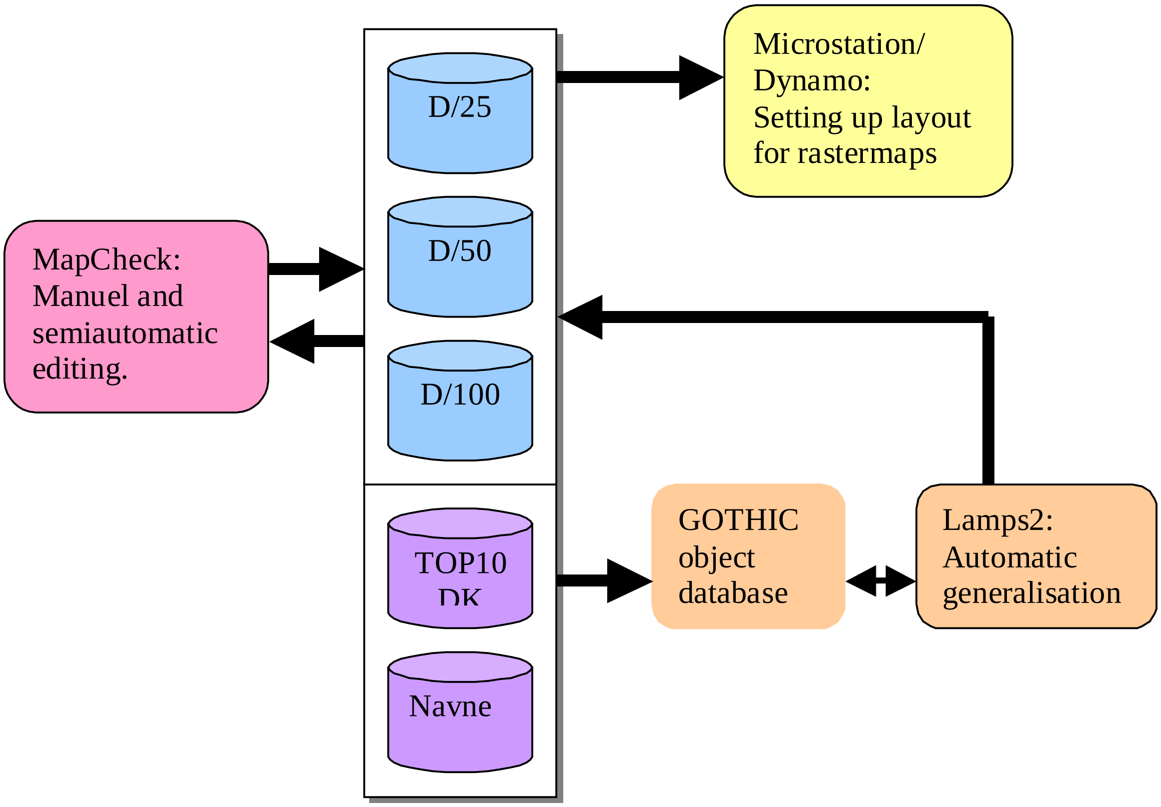
We decided early in the project to split the generalisation flow into a basic generalisation module and a product generalisation module.

We didn’t have any experiences in how to set up the workflow for the generalisation and knew that this solution by splitting the processes into pre- and a post processes might fail during the project.

The idea was to create some almost full automatically processes in the MapCheck system which could be used on different scales by changing the parameters. The result of the pre-processes should be a derived basic dataset for further manipulation/generalisation during post processes in the Lamps2.

The advantage of using the MapCheck for the basic generalisation/pre-processes was that it could handle all the requirements of the TOP10DK data model – keeping the z-value during generalisation, keeping the origin pr. point during generalisation, handling multiple geometries. Beyond this it has already processes implemented for topology check, editing and updating processes for registration of old and new objects.

As the commercial systems have problems in handling some of the requirements of the TOP10DK data model, a solution was to implement one or several commercial systems for handling the generalisation related to the different product types. For these product generalisations it was only required that the systems could support a part of the TOP10DK data model.



After some test in using 2 systems for the generalisation, we found out that we needed more time to analyse and test the way of handling basic- and product generalisation. We decided that another project has to carry through such a test of workflow. So for that reason we decided for the production of maps in 1:50.000 in year2001 to set up a generalisation workflow, where only the Lamps2 generalisation tool would be included.

Tool for editing, topology check and updating
As MapCheck has processes for topology check, editing and updating processes for registration of old and new objects implemented, it was decided to use MapCheck as an editing tool between the different generalisation sequences in Lamps2.

Using the NT workstations the user can link himself directly from MapCheck to the database containing the TOP10DK data and the generalised data and start the editing process. During the automatically generalisation there have been objects the process couldn't find any solution for, these objects will be stored in the database as *mark-up objects* and it will be possible for the user to go through these mark-ups using a *Find_next* process implemented in MapCheck, which marks them in the view. During the editing in MapCheck the user can use orthophotos or scanned map sheets as background for the vector dataset.

As a preliminary work for starting the updating process for storing the changed data to the central database, all objects which have to be deleted or new objects which have to be measured during the manual generalisation/ editing will have to go through following process: The new objects, which were not present in the original registration will get a TOPxxDK object code attached + 10000 to indicate that it is a new object. The original object, which doesn't exist no longer after an editing will get a TOPxxDK object code + 20000 attached to indicate that this is an object to be removed. The database will identify the equivalent objects and have them removed.

Tool for interface communication
To be able to import and export data between the systems and the database it was decided to use FME Oracle and get plug-ins for GOTHIC and MapCheck developed and implemented in the FME. Using the functionality in FME to create test factories and – functions we were able to run different processes doing the export, for example data translation, amalgamation of areas, select objects of a specific size or with a specific attribute.

Establishment of Generalisation Workflow

Setting up the workflow for the automatically and manual generalisation of the dataset in 1:50.000 have been an iterative process. The works have been carried out with support from the cartographers from the production – manual workflows and cartographic rules have been analysed. Also the quality of the results from the automatically generalisation have been evaluated.

The workflow is divided into 2 main steps, which contains one to several generalisation sequences. Each sequence includes a number of generalisation processes. In between the main steps a manual editing of the automatically generalised data is carried through. We have chosen to divide the workflow into these steps to avoid too many errors being transfered from one sequence to another without any manual editing in between.

As basis for carrying out the generalisation workflow, data from the different databases are imported to the GOTHIC database. The data that will be imported are the TOP10DK, the cadastral data, name data and supplementary data needed for this map product. The automatically generalisation will be carried through in the GOTHIC/Lamps2 environment and when the manual editing has to be carried through afterwards, the data will be transfered from GOTHIC to MapCheck.

The raster maps in 1:50.000 have in the earlier manual map production been derived from the existing map product in 1:25.000 and therefore it’s a relatively detailed map. We have been concentrated the development of generalisation processes on the following five main topics: 3D processing on the object classes – Traffic, Building and Built-up areas, Selection of objects and reclassification, Generalisation of areas, Generalisation of roads, Generalisation of buildings.

Generalisation processes in the workflow.

Development in the Lamps2/C

As it was not possible to meet the requirements of handling origin per point and height during a generalisation without having to put in many resources to get it implemented, it was decided to give these requirements a low priority during the implementation of this first version of the production system.

For having the possibility in the future to carry through an updatingproces of the derived dataset by getting information of which TOP10dk objects have been changed since last update, it is important that all the derived objects have an attribute attached, which tells us which TOP10DK objects it was derived from. As a first try we implemented such functionality on the processes for area generalisation and road collapsing, where the relation between the generalised object and the TOP10DK object is a ”one to many” relation

The generalisation processes have been developed in the Lamps2 environment, where the programming language is lull. It is also possible to develop processes in the C language. We have been co-operated with Laserscan in the work with developing the processes. Laserscan has among other works implemented a special version of the Agent, which supports the requirements of how to handle the buildings inside/ outside the urban areas during simplification and displacement and how to handle a cluster of buildings during a displacement.

Implemented generalisation processes for the D/50
Selected processes will be examined in the following section.

Area selection and -generalisation

As TOP10DK have several area classes, which it will not be possible to display in a map in 1:50.000, the first thing to do was to specify which 1) TOP10DK area types could be reclassified to new derived area types and 2) TOP10DK area types, which were too small to be displayed could be reclassified and aggregated to another nearby placed area type.

Processes to do analysing, reclassification, amalgamation, collapsing to line/point and deletion were developed. The following analyses were carried out – checking: area size, area type, area inside another area and area touching another area.

In TOP10DK all areas are specified not to cross the road so they are actually divided along the roads. When carrying through a derivation to a smaller scale we need to change this rule so the areas no longer have to be divided along the roads but can form a common point with the road where road cross the area. This change of the specification resulted in development of a process, which is able to do an amalgamation of these areas.

Selection of Buildings and Roads

Buildings

For the map products in 1:50.000 most of the buildings inside the build-up areas have to be deleted, only selected buildings as named buildings, large buildings, buildings for specific purpose (Hotel, Church, School,) or buildings inside industrial areas are displayed.

A process to support the selection of these buildings by using the information of the attributes of the building objects was developed.

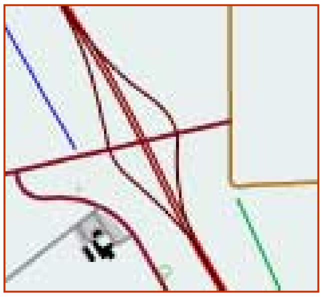
In TOP10DK the farms are registered as buildings so it is not possible to select them among the other buildings. As the farms have to be simplified and displaced in a specific way we need to make a reclassification of the buildings whish are farms. By using the cadastral areas as search polygons and the attributinformation of the building register we are able to identify and reclassify the farms.

Roads

A process has been developed, which can do a thinning of the roads in the build-up areas. Small roads with dead ends are deleted, entrances to farms are reduced and selected road objects of loops are deleted.

Before making a collapse of the larger roads we have to identify the slip roads and reclassify them. In TOP10DK they are classified as road objects.

Generalisation of the roads



For the larger roads in TOP10DK each lane has been registered as a road centreline. These parallel road centrelines have to be collapsed to one centreline during a derivation to 1:50.000. After the collapse all the endpoints for the roads, which have been connected to the original road centrelines have to be moved to the derived road centreline

The developed processes support the collapse and try to re-establish the connections between the roads connected to the original centreline and the derived centreline. It is necessary to do some manual editing to get a full reestablishment of the road network.

Generalisation of the buildings

Laserscan has implemented a version of the Agent, which support our requirement about how to do:

- Simplification of the building shape
- Scaling and symbolisation of buildings with a area size between 25 and 400m2
- Displacement of a cluster of buildings
- Displacement – priority of the actions: displace, orientate, merge, delete

The Agent functionality in Lamps2:
An agent is an active object within the database, capable of acting autonomously to bring about a satisfactory generalisation of its shape. Two different types of agents are used during the generalisation process – Mesoagent and Micoagent.

The Mesoagent are used to control groups of buildings. It’s responsible for ensuring, for example, that buildings do not overlap with each other or with nearby road objects, that buildings maintain their relative positions and that the buildings are deleted if necessary.

The Microagents represent the individual buildings. The microagent is responsible for itself only. It has constraints, which control aspects of its own geometry and plans, which can alter its own geometry. Microagents cannot affect any other objects and cannot destroy themselves.

In the earlier map production of 1:50.000 templates were used, when doing simplification of the buildings. Each farm was simplified to one of these possible farm templates. The farm templates could be combined, if necessary by using the farm template ’I’ together with one of the letters ’U’, ’O’, ’L’, ’T’, ’F’, ’H’. By using these farm templates during the simplification the maps got the same homogeneous look.

Did we have to implement these farm templates in the automatically generalisation workflow or could we reach the same results by using other methods?

We chose not to implement any farm template process as we would like to see how fare we could get by using the following processes on farms: simplification, scaling, deletion of small buildings, merging of buildings lying close to each other, displacement.

The displacement of buildings have been divided into 2 procedures:
Buildings in build-up areas – As most of the buildings have been removed inside build-up areas, only the buildings inside industrial area will have to be processed. The agent process will be run on the selected industrial areas.

Buildings in rural areas – As each farm is a cluster of buildings, a Mesoagent will be run on these. For other buildings a Microagent will be used. By selecting one or several partition area(s) defined by the user the process will be run on all the buildings inside that/these partition area(s).

Challenges in the Future

In the future work we will have to focus on the following topics:

1. *Improvement of specification for the small scale maps.*
   Set up a specification for each of the small scale map series, in which following topics will be described: Topological relations between the object types and – classes, How the objects have to be selected – which "if statements" should be used, How to generalise – when to smooth, when to delete, when to merge, How to do reclassification of objects.

2. *Implementation of updating processes for the derived datasets.*
   Methods for carry through an updating process of the derived objects in the light of the updates there have been carried through on the TOP10DK dataset, have to be developed and implemented.

3. *Improvement of generalisation- and editing tools.*
   Have to do further improvements of the processes used for the area generalisation and the road generalisation.

4. *Further test of implementing a pre- and post generalisation.*
   A model for how to carry through the generalisation in 2 steps – 1) from TOP10DK to a basic derived main dataset (TOPxxDK), where the processes are full automatically and 2) from TOPxxDK to a specific product, where the processes will be carried out in different tools where there will be both automatically and manually processes.

5. *Setting up generalisation workflow for maps in 1:100.000.*
   This work has not yet been started up. The plan is to start up a generalisation-project in autumn 2001.