Automatic placement of text for objects that do not have a clear geographic definition

Rose Nyberg

Lantmäteriet, the Swedish mapping, cadastral and land registration authority; rose.nyberg@lm.se

Abstract*:* The Swedish mapping, cadastral and land registration authority Lantmäteriet has started a project to automatically generalise geographic information. Planned timespan for the project is 2015-2022. The project and the automatic generalisation work are presented in two separate sessions at ICC. This paper is written to raise a question about automatic placement of texts in a map. In the project this is planned to be developed at the end of the timespan, but since the data models for almost all geographic feature types will be affected the text modelling work has been started.

Keywords: Automatic map generalisation, ArcGIS, Flexible products, National mapping agency, Sweden, Maplex, Automatic labeling

1. The project objectives

The starting point for the project MTP (Modern Topographic Production) is geographic databases in different scales, 1:10 000 (SE10), 1:50 000 (SE50), 1:100 000 (SE100), 1:250 000 (SE250) and 1:1 million (SE1M). There are many project objectives, the first to be mentioned is to automate the production of the smaller scale maps starting from the base map in 1:10 000.

Another project objective is to gather the manual forces in the work on the 1:10 000 scale base map to increase the accuracy and quality, thus enhancing the possibilities to automate the production of the smaller scale maps. A necessary task is then to extend the basic geographic information in SE10 to cover the geographic information now only present in the smaller scales.

Other project objectives

- A more flexible production line.
- Production of databases and maps that are possible to use in a more flexible way.
- Increasing the efficiency in production so that updates are visible in all scales more quickly.
- Facilitation of the development of new products.
- Modernized and improved databases, harmonized and comprehensive with flexible structure, synchronized with other Swedish authorities' databases.

1.2 Technical information

The development environment consists of

- ArcGIS Desktop version 10.5 with ModelBuilder and FME Extension for ArcGIS
- FME Desktop 2016 Oracle Edition
- Proprietary tools developed with ArcObjects (C#) and Python (ArcPy)
- The database environment is ArcSDE on top of Oracle.
- For automatic labeling Maplex will be tested.

In the current databases, the texts are stored as stand-alone objects (in annotation classes) with only positional connections to the real-world objects they describe. The geometry types for the texts are points or lines.

1.3 The generalisation work

This paper will concentrate on the labeling of texts. The generalisation work of the other objects is more closely described in a paper submitted to the ICC, and will be presented there too. Here is just a very short example of how the generalisation of buildings, hydrography, land cover, roads and railroads can look like.

In *appendix 2* the generalisation methods used in ArcGIS ModelBuilder up till today are listed. The generalisation methods are put together in generalisation models and the tools are sometimes used many times to accomplish the desired result.

The map examples below describe the ongoing work and **not** the finished production model.

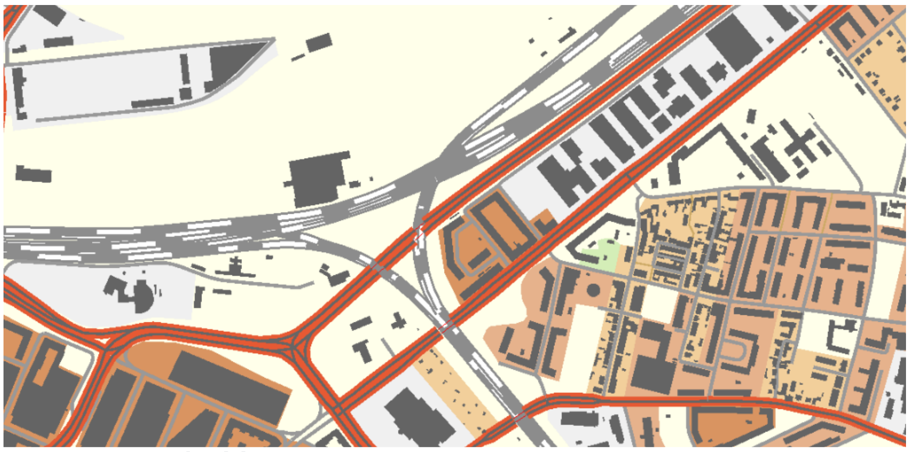


Fig.1 Ungeneralised data, 1:5 000

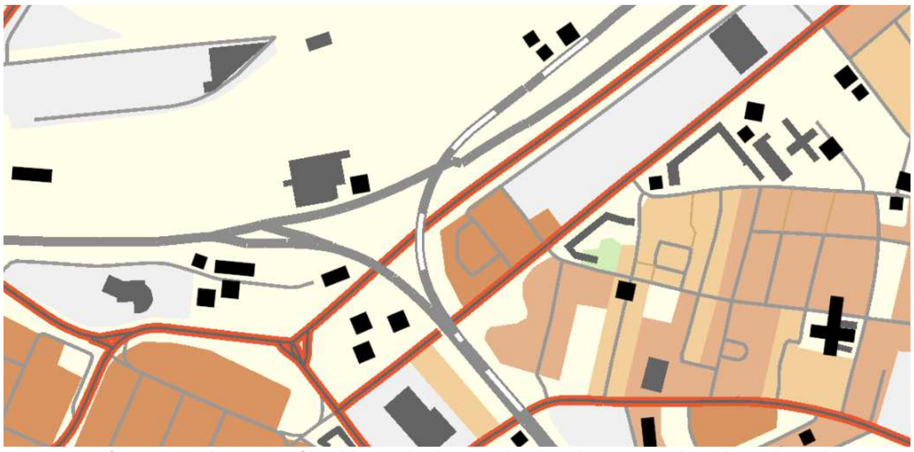


Fig.2 After generalisation of buildings, hydrography, land cover, railroads and roads

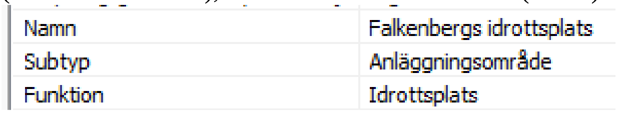
1.4 The planned production lines and performance requirements

The projects hypothesis for the generalisation production lines is that the smaller scales will be produced and stored in separate databases. Thus, the performance requirements are lower compared to on-the-fly-maps. But still the performance must be sufficient to manage the updating of these databases often enough.

2. Automatic placement of texts

Using Maplex the testing of automatically placement of texts has been started. There are lots of Maplex settings to explore when the texts are drawn in the map and the work has just begun. Below is an example where the Maplex setting “remove duplicates” has been completed with Python-code (see *appendix 1*).

The Python-script handles both attributes that can be displayed in the map for the class “Anläggningsområde” (Construction Zone), these attributes are “Namn” (Name) and “Funktion” (Function).



Namn	Falkenbergs idrottsplats
Subtyp	Anläggningsområde
Funktion	Idrottsplats

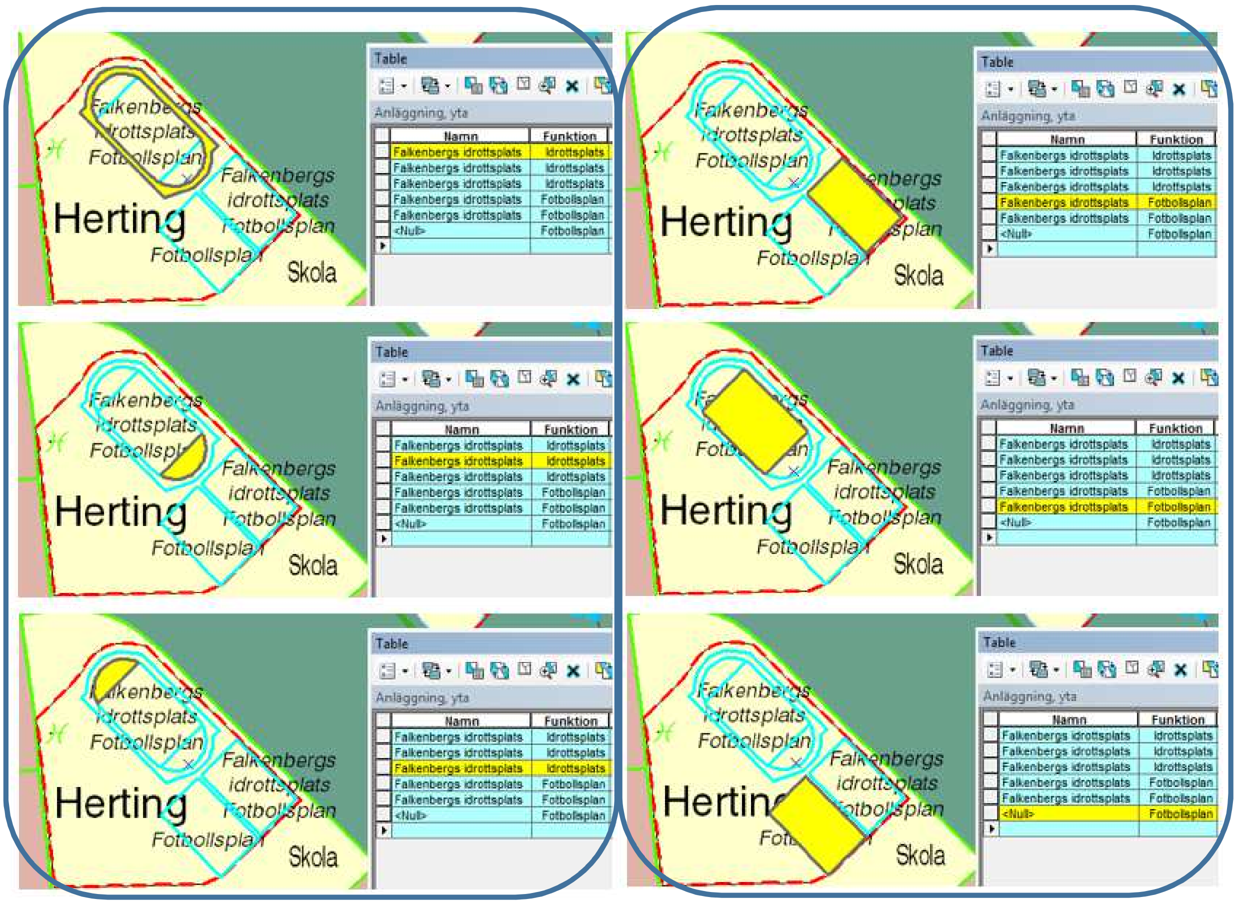
The Python-script chooses which text to draw like this:

1. If the object has a function attribute and a name attribute and the function is included in the name no labeling is done, e.g. name=“Falkenbergs idrottsplats”, function=”Idrottsplats”. This is OK for the example, but not for every object class.
2. If the object has a function attribute and a name attribute and the function is **not** included in the name, the name and function is put together to a label, e.g. name=“Falkenbergs idrottsplats”, function=”Fotbollsplan” → label=“Falkenbergs idrottsplats \newline Fotbollsplan”
3. If the object has a function attribute it is labeled, e.g. name=“”, function=”Fotbollsplan” → label=“Fotbollsplan”

The chosen area below displays a sports stadium in “Anläggningsområde”. A sports stadium that consists of many smaller areas and football fields, thus there are many objects with similar names and functions in the vicinity.

The objects in yellow below are never labeled, see point 1 above:

These objects in yellow below are labeled according to the Maplex settings, see point 2 and 3 above:


Falkenbergs idrottsplats Fotbollsplan
Falkenbergs idrottsplats Fotbollsplan
Herting
Fotbollsplan
Skola

Table

Anläggning, yta

Namn	Funktion
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Fotbollsplan
Falkenbergs idrottsplats	Fotbollsplan
<Null>	Fotbollsplan

Table

Anläggning, yta

Namn	Funktion
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Fotbollsplan
Falkenbergs idrottsplats	Fotbollsplan
<Null>	Fotbollsplan

Table

Anläggning, yta

Namn	Funktion
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Fotbollsplan
Falkenbergs idrottsplats	Fotbollsplan
<Null>	Fotbollsplan

Table

Anläggning, yta

Namn	Funktion
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Fotbollsplan
Falkenbergs idrottsplats	Fotbollsplan
<Null>	Fotbollsplan

Table

Anläggning, yta

Namn	Funktion
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Fotbollsplan
Falkenbergs idrottsplats	Fotbollsplan
<Null>	Fotbollsplan

Table

Anläggning, yta

Namn	Funktion
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Idrottsplats
Falkenbergs idrottsplats	Fotbollsplan
Falkenbergs idrottsplats	Fotbollsplan
<Null>	Fotbollsplan

In all the labeling samples below the same Python-script is used, see *appendix 1*. All settings are the default settings if not otherwise mentioned. The screen dumps are taken in the map scale 1:5 000.

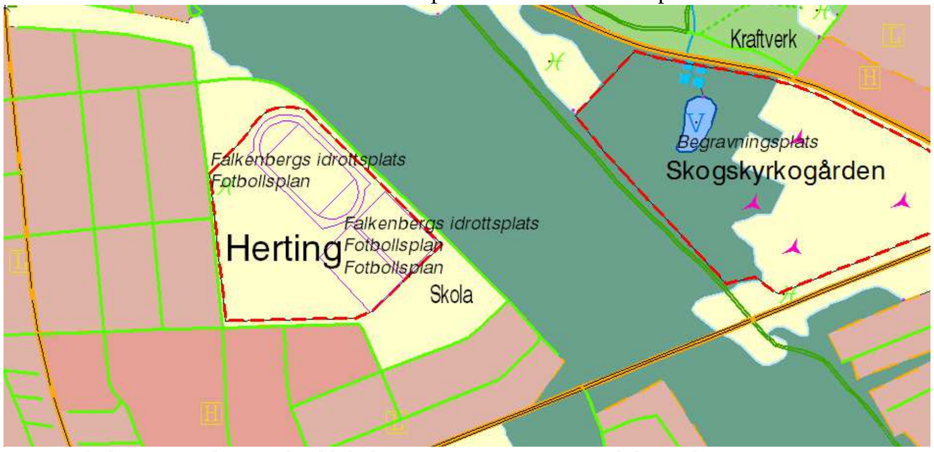


Fig.3 Labeling using the standard labeling engine in ArcMap and the Python-script

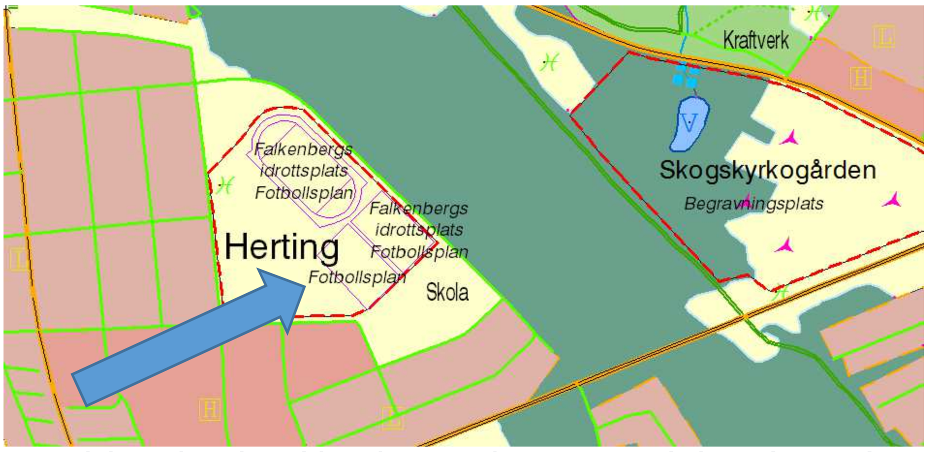


Fig.4 Labeling with Maplex and the Python-script, the setting remove duplicates deactivated

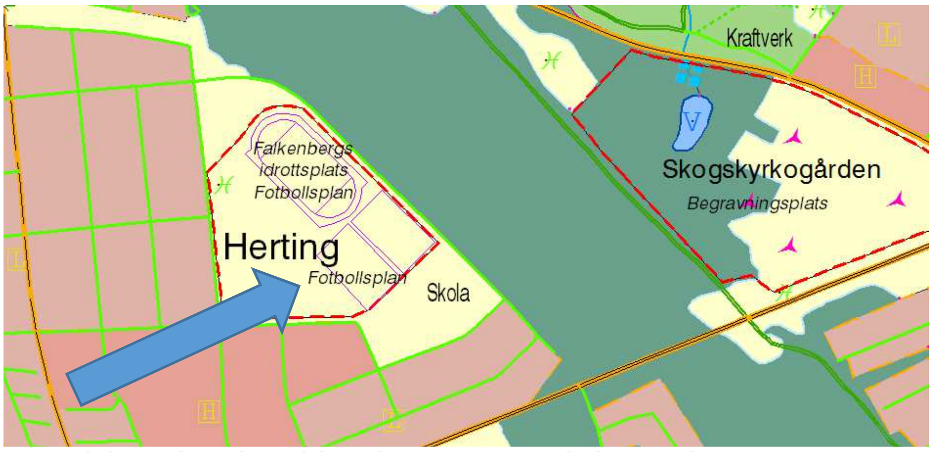


Fig.5 Labeling with Maplex and the Python-script, remove duplicates radius set to 20 mm

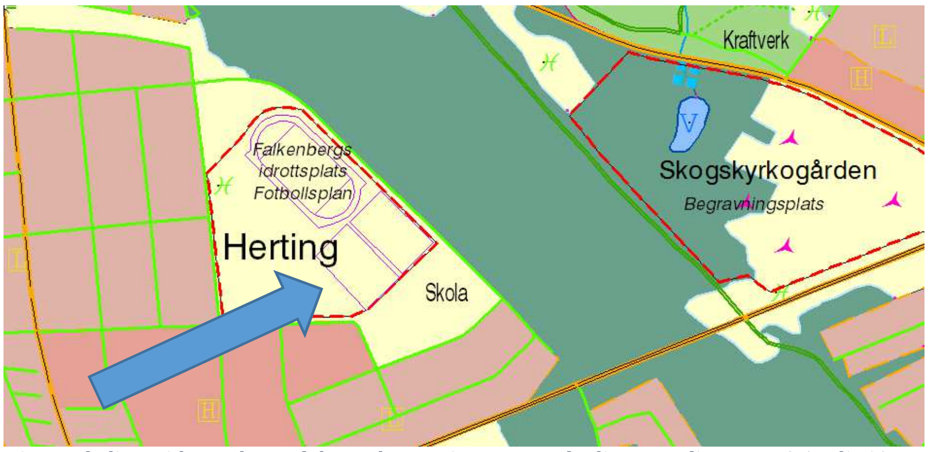


Fig.6 Labeling with Maplex and the Python-script, remove duplicates radius set to 0 (no limit), 1:5 000

When changing the Maplex setting “remove duplicates” first from deactivated to “radius” = 20 mm and then to “radius” = 0 (no limit) the labeling of the adjoining football fields is removed, as pointed out with blue arrows above.

The labeling results above show that with Maplex it is possible to steer the labeling in detail using different “remove duplicates” settings. There are many more Maplex settings to explore and test before a final solution is chosen.

3. Data modelling of texts

To be able to place the texts automatically using a labeling engine the understanding that all texts must belong to a real-world object has grown. This is the reason for starting the text modeling work early in the automatic generalisation project, since the data models for almost all other feature types must include texts.

There are texts where no real-world features exist in our current data. For example, the mountains, which are described with contour lines and names, but not with separate mountain features. Another example is the sea and the bays, the shore line does not mirror these objects.

This fact has lead us to try a solution where so-called *invisible features* that can be used for automatic labeling will be created. Invisible features refer to features whose geometry is not shown in the map, but contains the texts to be placed automatically. The ambition is that the creation of invisible features will be automated but it will be necessary with manual edits also. At the lowest ambition level the invisible text features will be created with point or line geometries, preserving the current geographical information. At a higher ambition level polygon features will be created where suitable.

Creating polygons for texts belonging to real world objects with a geographic extent the placement of the text will have more alternatives if colliding with other texts or features in the map.

One big question is if it is possible to automatically create invisible polygon features. For the mountain names the contour lines can be used but there are many cases where the contour lines are of little or no help at all as in the example below where two nature names, marked with red circles, are placed in the same area. “Storliden” can be placed from the contour lines, but not “Ladumyrberget”.

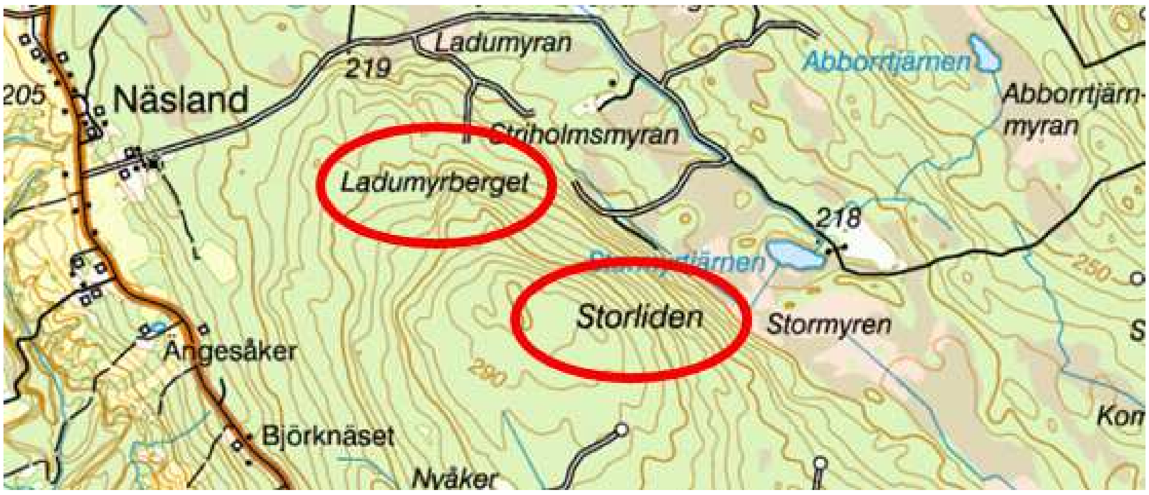


Fig. 7 Topographic map 1:14 000

Another example where automatic generation of invisible real-world objects is difficult to achieve automatically is seas and bays. In the example below from the Stockholm archipelago there are many names with the ending of “fjärd” (bay), none of which has a borderline in the map. And all of which are parts of “Östersjön”, the Baltic Sea.

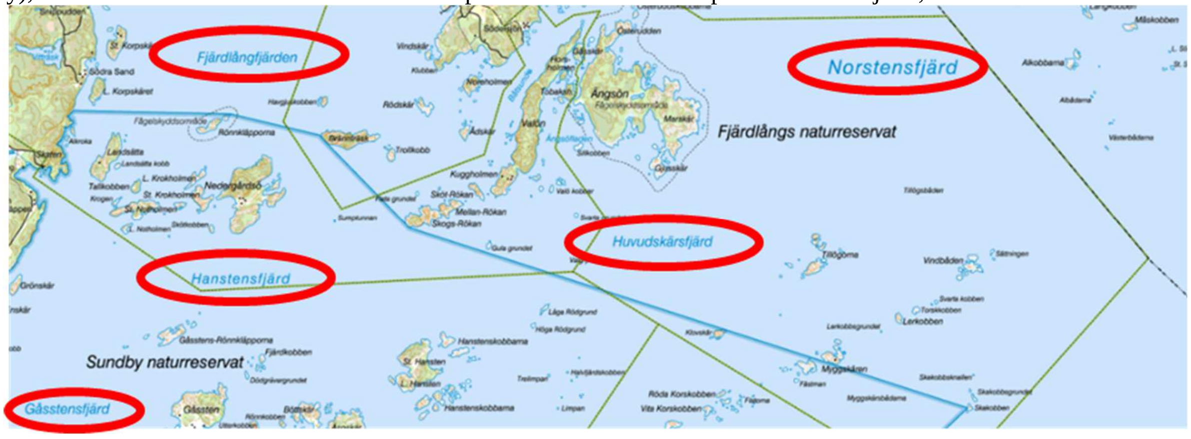


Fig. 8 Topographic map 1:50 000

4. Questions to discuss

This pre-conference has given us an opportunity to get information of other efforts in the field of automatically placed texts in map production. Below is a list of questions that can be a basis for the discussions.

- Has anyone tried automatically placement of texts and can supply more information on how to handle texts that don’t belong to existing database features?

- To what extent is the labeling automated and what types of products are handled?

    - Is the result satisfactory or is it necessary to accept big deviations from the meticulous requirements for manually placed texts?

- What software has been used?
- Is the performance adequate for dynamic web maps?

Acknowledgements

We are grateful to Dutch Kadaster for all help at the initial project phase.

References

Rose Nyberg, Mikael Johansson and Yang Zhang (ICC 2017). Automatic map generalisation from research to production

Vincent van Altena, Jan Bakermans, Ben Bruns, Hedwig Kupers, Ron Nijhuis, Marc Post (2013). An Overview of the Dutch Approach to Automatic Generalization

Mote, K, Fast Point-feature label placement for dynamic visualizations.
https://arxiv.org/ftp/arxiv/papers/1209/1209.5765.pdf

Cynthia A. Brewer , Lawrence V. Stanislawski , Barbara P. Buttenfield , Kevin A. Sparks , Jason McGilloway & Michael A. Howard (2013) Automated thinning of road networks and road labels for multiscale design of The National Map of the United States, Cartography and Geographic Information Science, 40:4, 259-270, DOI:
10.1080/15230406.2013.799735

Raposo, P., Brewer, C.A., and Stanislawski, L.V., Label and attribute based topographic point thinning: Proceedings, 16th ICA Workshop on Generalization and Multiple Representations, August 23-24, 2013, Dresden, Germany, 8 p.

Appendix 1
Python-script for labeling of two attributes

```
def FindLabel ( [NAMN], [FUNKTION] ):
    import arcpy, os, string, re, codecs
    f = [FUNKTION]
    if f != "Ingen information":
        n = [NAMN]
        if n and len(n)>0:
            nLower = n.lower()
            fLower = f.lower()
            if nLower.find(fLower)==-1:
                nykarttext = n + '\r\n' + f
            else:
                nykarttext = ''
        else:
            nykarttext = f
    else:
        nykarttext = ''
    return nykarttext
```

Appendix 2

Lists of generalisation methods used, the list is not exhausted but more an example

Generalisation methods used for land cover

- Exaggerate part or all
- Delete narrow parts
- Dissolve adjacent land cover features of the same type
- Eliminate small land cover surfaces
- Aggregate small land cover surfaces
- Simplify land cover surfaces
- Eliminate small islands in some types of land cover surfaces
- Delete or exaggerate narrow parts of land cover surfaces

Generalisation methods used for hydrography - polygons

- Collapse water surface
- Eliminate small water surfaces and marshes
- Collapse narrow rivers
- Aggregate small water surfaces and marshes
- Simplify water surfaces, marshes and water lines
- Eliminate small islands in water surfaces and marshes
- Aggregate small islands in water surfaces and marshes
- Exaggerate small details in the shoreline (piers and forelands)

Generalisation methods used for hydrography - polylines

- Remove all water courses with short dead dangles
- Remove shortest water courses if they don't belong to water network
- Simplify and smooth the water courses
- Maintain the water network during the generalisation

Generalisation methods used for roads

- Reclassify roads
- Dissolve roads
- Merge Divided Roads
- Collapse road details
- Thin road network
- Resolve Road Conflicts
- Transfer Attributes between different input datasets

Generalisation methods used for railroads

- Reclassify railroads
- Eliminate short tracks with dangle nodes
- Remove unimportant railroads
- Dissolve
- Merge Divided Roads

Generalisation methods used for buildings

- Eliminate small buildings
- Dissolve buildings
- Eliminate buildings in built up areas
- Remove small, not isolated buildings
- Remove small holes in buildings
- Aggregate buildings
- Simplify building
- Collapse buildings as polygon features to point features
- Resolve building conflicts (displacement, selection, etc.)