MODELING, COMPUTING AND CLASSIFYING TOPOGRAPHIC AREA FEATURES BASED ON TOPOLOGICALLY NON-STRUCTURED LINE INPUT DATA

Marlies DE GUNST*, Peter VAN OOSTEROM and Berry VAN OSCH*****
* Geodan Geodesie, Koningslaan 35, 1075 AB Amsterdam, The Netherlands,
m.degunst@geodan.nl
** Delft University of Technology, Department of Geodetic Engineering,
Section GIS-technology, Thijsseweg 11, 2629 JA Delft, The Netherlands,
oosterom@geo.tudelft.nl
*** Dutch Cadastre, P.O. Box 9046, 7300 GH Apeldoorn, The Netherlands,
berry.vanosch@kadaster.nl

KEY WORDS: classification, knowledge engineering, modelling, topology, spatial databases

ABSTRACT

In this paper the conversion of a spaghetti into a topologically structured topographic base map is discussed. The first step, called structuring, comprises node computation, handling overshoots, undershoots and overlapping line segments. Node computation is a computational geometry problem driven by tolerances and a weighting scheme. The next step is placing additional lines by hand to close certain areas. Finally, area features are classified using a rule-based system. Specifically the rule-based classification of the areas has been implemented, tested and compared to human classification. In these experiments automatic classification leads to a speed-up of a factor 2 while maintaining a similar classification performance when compared with manual classification.

1 INTRODUCTION

The large-scale topographic base map of the Netherlands (GBKN) at scale 1:1.000 in urban areas and at scale 1:2.000 in rural areas covers almost the complete country. It consists of line elements, point symbols an text labels. The current GBKN does not represent area features explicitly. However, human map readers are, most of the time, able to interpret the map and determine where the area features are located by looking at the lines (representing topographic boundaries), text labels and symbols. For the computer it is much harder to discover the topographic area features. This is the domain of database interpretation: deduce information that is not explicitly stored in the data model of a digital map (Anders and Fritsch, 1996).

Having area features explicitly in the map makes it possible to produce better cartographic output; for example fully colored maps. The topologically structured topographic map is better suited as basis for thematic maps by combining the basic areas to appropriate units. It is a better model of the world (Digital Landscape Model, DLM), because there is more explicit knowledge stored concerning the real world features. This makes it possible to relate administrative data, like cadastral information, to the topographic area features. Having area features explicitly available makes it also easier to reason about them; e.g. apply generalization (removing objects, combining objects, simplifying objects, etc.) or at least propagate updated data from the large-scale to the medium- and small-scale models (Uitermark et al., 1999a). Other examples are the computation of the center line of roads or waterways and 3D building reconstruction using a large-scale database and aerial photographs (Haala and Anders, 1996).

The goal of the described research is to convert at lowest possible costs the current spaghetti data set into a topologically structured data set with labelled area features. The developed method fits within the map production process of the Dutch Cadastre. As human labor is expensive, it is tried to automate the conversion process as much as possible. This is especially true for a very large data set. The estimated size of the Netherlands large scale line-oriented topographic map is: 5.000.000 text labels, 5.000.000 point symbols, and 180.000.000 topographic line segments (circular arcs or straight). In total the database requires about 25 Gb disk space. Some initial estimations state that 95% of the lines will be part of the area features and that for only 10% or less of these lines the position remains unchanged. Because labeling the area features is very time-consuming, a rule-based system is developed for this purpose. Figure 1 shows a fragment of the current topologically non-structured large-scale topographic base map of the Netherlands.

Before describing the conversion process steps, first the area-based model is discussed in some more detail in Section 2. The first step in the conversion, structuring, is described in Section 3. It comprises node computation, removing overshoots and handling double line segments. This is computational geometry driven by tolerances and the quality (accuracy)

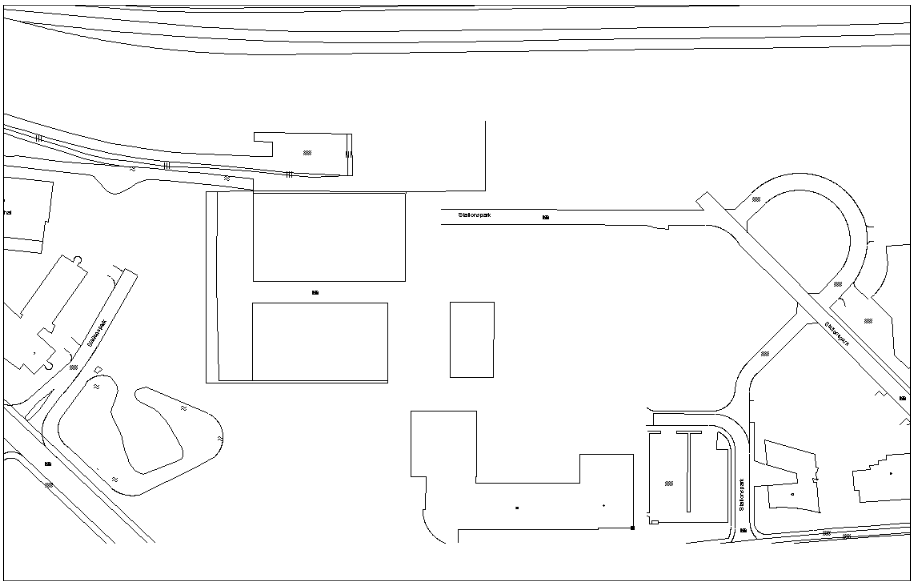

Figure 1: Current topologically non-structured topographic map

attributes related to all input lines. Spatial data organization (clustering and indexing) is important to perform this step efficiently. The next step is placing additional lines to close certain topographic areas, this is done by hand; see Section 4. The final step is then classifying the areas and for this a rule-based system has been used; see Section 5. Results of experiments on area classification are described in Section 6. The paper is concluded with some final remarks in Section 7.

2 MODELING

When designing a new model we had to decide how the different features are organized; e.g. Which are there different layers or themes (de Hoop and van Oosterom, 1992) both from the users and technical perspective? Further, we still have to be able to deliver the traditional line-based topographic data. This may cause some additional requirements for the model to be designed. Another feature of the current as well as the new model is that a line can only have one classification label (single-coding). However, a topographic line most of the time represents a boundary between two area features, e.g. a facade of a building and the side of the road. This means that a choice is made which classification label is assigned to a line, based on a hierarchy of classes. This complicates determining the classification label of the area and it has consequences for handling overlapping line segment when forming areas. Finally our model needs to be as much as possible consistent with NEN3610 (Ravi, 1995), the standard for the Dutch digital landscape model. In the next subsections the model is first discussed from a more logical and then from a technical point of view.

2.1 Logical data model

It was decided to store all area features in one topologically structured layer. The main motivation for this was that this would make the conversion process easier. In addition independent layers would introduce redundant data storage with inconsistency risks during editing. A classic discussion is how to model junctions in the 2D planar map. It can be the case that one road passes over another road (or waterway) and that there is no real connection. There is no common geometry in 3D, but overlap in the projected 2D situation. Of course, there are much more combinations, e.g. building over roads, and much more complicated situations; e.g. four 'layers' of elements overlapping in the 2D projection. There are seven possible classification labels for area features. In addition to the NEN3610 topographic objects, *roads, railroads, water, terrain, construction work* and *building*, it was decided in include a area feature type *junction*. It is also possible to assign sub-classifications that give a more detailed classification. For example, *main building* as refinement of a building and *road over water* as a description of the type of junction.

The real line and point features are stored in separate layers in order to avoid unnecessary intersections. Note that in a large-scale map there are not many line features. The traditional line features of the medium- and small-scale maps, such as roads, rivers, railroads are area features in the large-scale map. A few line features remain at large-scale: fences, (power) cables, etc. They are stored in the separate 'line layer' together with overlapping lines which may not be deleted, because these have to be delivered to customers. A third group of lines in this layer is formed by dangling lines. On the basis of their line classification they could very well belong to the planar partition layer, but for some reason they are unconnected at one or two endpoints. This could be caused by an error.

2.2 Technical model

It was decided, for practical reasons, to use the same topology model for the large-scale topographic map as the one used in the Cadastral map (Lemmen et al., 1998). So, we can reuse all our current tools: editor (KT-Datacenter Ltd., Riihimäki, Finland, 1994, Karttakeskus, Helsinki, Finland, 1994), querytool (van Oosterom and Maessen, 1997), mass data delivery software, etc. Note that, in this subsection only the layer with topologically structured area features will be discussed. The technical model is used to implement one planar partition, with a topology-based data structure called winged edge (Baumgart, 1975). Features have a Nationwide unique object identifier. In fact it was decided to allow two unique identifiers: one from the Cadastre and from an external organization; e.g. a municipality. The key element is the line element, in this case the topographic boundary, which can be either a polyline or a circular arc. The model also includes the thematic attributes. The history of objects is maintained by using time stamps (tmin/tmax). It is therefore a spatial-temporal model based on explicit topology. It is implemented in a relational database (CA-OpenIngres, 1995) using spatial data types very similar to the OpenGIS Simple Feature Implementation specification (Open GIS Consortium, Inc., 1998). In this environment both efficient delivery of update files and quickly browsing at different moments in time at the data set are possible.

3 AUTOMATIC STRUCTURING OF AREA FEATURES GEOMETRY

The automatic process to convert non-structured line input data into topologically correct area features is called *structuring*. From the original large-scale topographic map all lines are used in the process of structuring, perhaps with exception of some evidently real line features (power cables). A general limiting condition is that the original map changes as little as possible by the process of structuring. Note that no implementation has been made (yet) and therefore no practical testing has been done. Structuring covers: *node computation* at intersections and at geometries within tolerance distance (undershoots); solving *overshoot* problems; and handling *overlapping* (double) line segments.

3.1 Node computation

A *node* is a point were two[1] or more lines meet. The guiding principle of node determination is the use of a weighting scheme and a tolerance circle. The weighting discriminates our approach for node computation from work by other authors (Žalik, 1999, Laurini and Milleret, 1994). Weighting is based on quality attributes of lines, such as the identification precision and acquisition precision. Identification precision expresses how well a line can be determined in the terrain and is directly related to the classification of the line. Within specified tolerances lines may be changed in order to join in a common node. The determination of the position of this node is based on the presence of end-points and intermediate points of lines within the tolerance circle of a candidate node and their weights. Certain classification or quality attributes can be specified as not allowing to change the associated line geometrically by the node computation.

Two alternatives algorithms for the node computation were investigated during the project. The methods are called *point-by-point* and *cluster-based* node computation. The *point-by-point algorithm* determines the position of candidate nodes one after the other. First, nodes at intersecting lines are determined, next dangling lines are treated. The point within the tolerance circle with the highest weight becomes the new node for the handled intersection or dangling line. The *cluster-based algorithm* determines all possible candidate nodes at both intersections of lines and at (end and intermediate) points of lines close to other lines. Next contiguous areas of overlapping tolerance circles around the candidate nodes are clustered. For each cluster node locations are computed based on the weights of points and lines within the cluster. For both methods an efficient implementation based on spatial searching is needed.

Defining a node often implies that one of the involved lines has to chance a little (within tolerance). However, even this small change may cause a topology error somewhere else. So, after changing a line, this has to be checked. The advantage of the *point-by-point* method, is that only one line changes. However, the alternative *cluster-based* method may cause smaller changes, because the new node lies in between. A drawback of the *point-by-point* method is that the order of processing may influence the final result in case of clusters of points close together. In general the treatment of clusters of close polylines and points seams week. For the *cluster-based* method it may not be possible to find one node which lies within the tolerances of all points within the cluster. Therefore, sometimes more nodes within one cluster are needed. In the case that more nodes per cluster are needed, some algorithm has to be designed how to find these node locations. One approach could be to split the set of candidate nodes along a line orthogonal to the best-fit or eigenvector line-fit (Duda and Hart, 1973). This procedure to split a cluster into two parts can be repeated until a node can be computed, which is a good representation for all candidate nodes in the sub-cluster.

[1] Very often three or more lines meet in a node, because otherwise this node could just be an intermediate point of a polyline, unless the attributes of the lines are different.

3.2 Removing overshoots and handling overlapping lines

After node determination *overshoots* are handled. If dangling lines are shorter than a certain minimum size, these overshoots are removed. A threshold is used that can be tuned differently from the size of the tolerance circle used for node determination. Line segments that completely or partly *overlap* are handled. Principally, overlapping lines should not be present. In case a line represents a common boundary between two features, e.g. a road and a building, it should be represented by one line. The classification label is assigned to this line according to a certain hierarchy: buildings, roads, water, etc. However, overlapping lines turn out to be present in the spaghetti map. In case lines coincide, one element is maintained according to the hierarchy. In some cases both overlapping lines have to be present in the output in accordance to delivery agreements with certain customers. In this case one line element is chosen to be an area boundary according to the hierarchy and the other line elements are stored in the separate line layer in order to be able to still deliver the traditional line-based output.

4 CLOSING AREA FEATURES BY HAND

After the automatic structuring of area features geometry, there may still be gaps larger than the tolerance for node computation that should be closed in order to separate area features. An example of a situation in which these gaps often occur is at the access to private property; see Figure 4 (left). It was agreed that the side of the access roads are only measured until a certain distance from the main road. Usually at that point there is no visible boundary between road and terrain, so no separation line is present. However, without the separation line, the classification of the area can never be correct and will result in a conflict during automatic classification of the areas; see Section 5. The user can correct the situation afterwards. Because these situations occur often, it is better to correct them by adding lines to separate topographic area features; see Figure 4 (right). The user may find these situations by inspecting the dangling lines after node computation. Another situation in which areas are separated is not because the classification should be different, but to subdivide the area into more manageable and meaningful parts. For example, the separation of road networks into junctions and connecting elements. In (Uitermark et al., 1999b) an automatic technique is described using a constrained Delauney triangulation; see Figure 4. The same technique can be applied to other linear feature types, railroad and waterways.

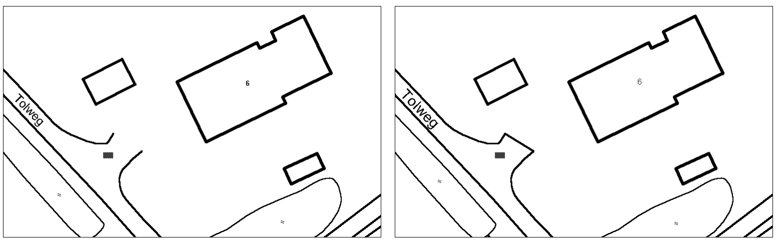

Figure 2: Situation without (left) and with (right) separation line between road and terrain near access to private property

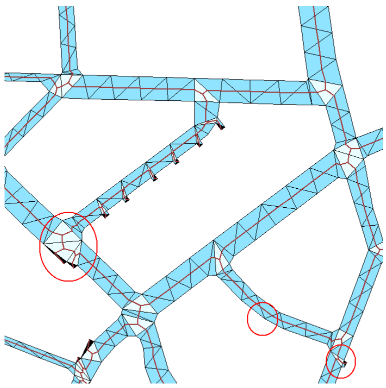

Figure 3: Automatic subdivision of large infra structure polygons

5 CLASSIFICATION OF AREAS

After the previous steps, structuring and closing area features by hand, the final areas can be created by computing the topology and storing the results in the database. The last step which has to be performed is classification of the areas. It was decided to use a rule-based approach, because it is easier to maintain, very flexible and can be adjusted to the provincial differences which exists within the input data in the Netherlands.

5.1 The knowledge

The knowledge a human experts uses when classifying the topographic area features resided mainly in the heads of the experts. Therefore, several interviews were conducted to make classification knowledge explicit and to obtain uniform classification rules. Main types of knowledge used for classification are:

- Labels of lines bounding an area feature;
- Presence of symbols and text within the area feature;
- Spatial relationships between area features;
- Shape of the area feature;
- Implicit contextual knowledge (e.g. expert recognizes a certain pattern such as parking lots alongside a road);
- Real world knowledge (e.g. expert lives in the represented area and can identify certain polygons).

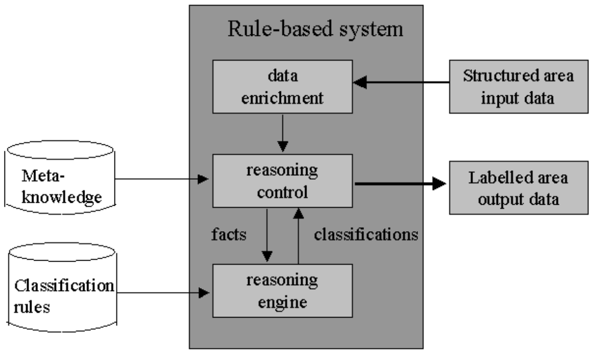

Figure 4: Architecture of system for rule-based classification

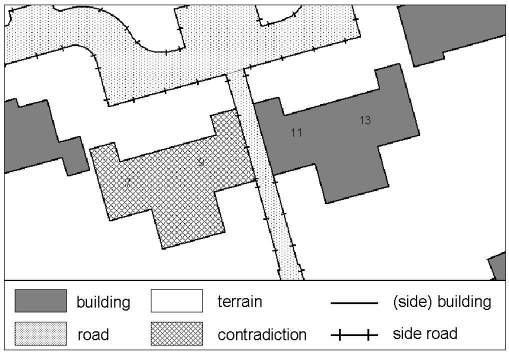

Figure 5: Contradiction detected by the rule-base system

The last two types of knowledge are respectively hard and impossible to model. Therefore, it was decided to use the first four types of knowledge to build the rule-based classification system. If there is evidence for more than one area class, it was decided not to choose for one of them, but to classify the area feature as a *contradiction*.

5.2 The system for rule-based classification

The rule-based classification system is designed to be separate component of the Geographic Information System (LKI[2]) of the Cadastre. Within the classification architecture three modules can be distinguished (Figure 4): 1. *Data enrichment*, 2. *Reasoning control*, and 3. *Reasoning engine*. The closed areas are input for the *data enrichment* module, which adds explicit information about the spatial relationships between the areas to the input data. For example, the fact that the same line is part of the boundary of two areas can be used to derive that these areas are neighbors. Another type of relationship that is derived are areas completely enclosed by other areas. This type of information is needed for classification. For example, a road will never be enclosed completely by a meadow. By early determination of all spatial relationships it is prevented that the reasoning engine, which is not suitable for this task, should perform it. Also the detailed line classifications of LKI are put together in more general classes. The reason for this is that rules will get clearer and will become smaller and easier to maintain.

The second module controls the *reasoning process*. Its main task is to give the third module, the reasoning engine, a limited classification task, together with the appropriate facts about areas (i.e. enriched data) and classification rules. An example of such a task is the classification of buildings (see Subsection 5.3). When the reasoning engine completes its classification task, the reasoning control module formulates a new task. This iteration continues until all classification tasks are completed. So-called meta-knowledge is defined in order to establish the classification order of the seven possible area classifications. It is based on the hierarchy of line classifications. The order is important because neighboring areas can help to determine the class of a certain area. In our situation it was decided to first find the contradictions (input data errors), then buildings, roads, railroads, water, junctions and terrain. If the order needs to be changed, only the meta-knowledge needs to be adapted which improves the maintainability of the rule-base. The rule-base is organized such that the rules are grouped per area feature type and are stored in one file. There are separate files for rules with changes or additions to the main rules that model provincial differences in the GBKN of the Netherlands.

The *reasoning engine* is implemented by means of the rule-based system CLIPS (Giarratano, 1991). CLIPS is a public domain expert system developed by the NASA, Software Technology Branch. CLIPS proved its reliability world wide in other environments while it is a completely open system that enables us to encapsulate it in a reasoning process. Within our application CLIPS gets from the reasoning control module facts about areas and *if-condition-then-action* rules. The CLIPS inference engine tries to match the pattern in the facts with the condition part in the rules. If both match for an area, then the action is executed; e.g. area is classified. This approach is called forward reasoning. After rule-based classification, visual inspection needs be done in order to resolve conflicting situations, which are detected automatically. Further the result of automatic classification should be checked.

5.3 An example

In this subsection an example will be given of a classification task: the classification of buildings. For this task the reasoning control module selects rules concerning classification of buildings. One of the rules[3] used for this purpose is:

[2] LKI stands for *Landmeetkundig Kartografisch Informatiesyteem* (in Dutch): 'Information System for Surveying and Mapping'.
[3] Pseudo-code is used to express the rules and facts because the use of real CLIPS-notation would be difficult to read.

Figure 6: Part of the GBKN of Zeeland classified by the rule-based system

```
(DEFRULE find_a_building
   IF ((area_class_is: UNKNOWN) AND (bounding_lines_have_ class BUILDING) AND
      (text_of_class: HOUSE NUMBER))
   THEN ((assign class BUILDING) (assign sub-class MAIN BUILDING))
```

This rule classifies areas as *building* with a sub-class *main building* if the area has not yet been classified, is bounded by lines that all have the classification building and has a text inside that represents a house number. Examples of facts that are tried to be matched with this rule by the reasoning engine are:

```
(AREA_FACT (identification_number: 12) (area_class: UNKNOWN)(area_subclass: UNKNOWN)
   (bouding_line_classes: BUILDING) (text: HOUSE NUMBER)

(AREA_FACT (identification_number: 13) (area_class: UNKNOWN)(area_subclass: UNKNOWN)
   bouding_line_classes : BUILDING, ROADSIDE) (text: HOUSE NUMBER)
```

When matching all features of the fact representing area *12* with the previous rule, this result in the classification of area *12* as a main building. However, area *13* will not be classified as a building when matched with the rule, because the area is bounded by one or more lines of class roadside. Actually, this line (Figure 5) should be classified as building, so there is an error in the map. This contradiction is signalized by the system because of the presence of a house number, using the rule like the rule below:

```
(DEFRULE find_a_contradiction_in_building
   IF ((area_class_is: UNKNOWN) AND (contains_text_of_class :HOUSE NUMBER) AND
      (one_or_more_bounding_line_have_NOT_class :BUILDING OR MANUALLY_CLOSED))
   THEN (assign class CONTRADICTION)
```

6 EXPERIMENTS AND RESULTS

In this section the classification performance and the prospects for the speed-up of the manual process of line to area conversion are discussed. The rule-base is developed using a training set of 11 spaghetti maps, each with an area of about 1 km^2. The performance of the system has been tested on an independent test set of 5 maps. The maps originate from different provinces of the Netherlands. The set of maps contains rural, suburban and urban regions. They were structured and both classified manually by an expert and automatically by the system; see Figure 6. The results of the comparison of the manual with the automatic classification on the train and test set are very promising: 88% of the areas in the train set and even 93% of the areas in the independent test set are classified identically by the human expert and the system (Table 1 and 2). When analyzing the 10% of the areas that were classified wrong, three causes of classification errors can be identified:

1. A situation was not covered in the rule-base (4-5%). The basic assumption when developing the rule base was that it contains rules for common situations and that unusual situations are corrected manually. Table 3 shows that common object types like building and water are better recognized than uncommon object types like railroad.

Region type (province)	Number of areas	% correct
Rural (Friesland)	150	99%
Suburban (Friesland)	4826	90%
Urban (Friesland)	1194	75%
Rural (Zeeland)	615	96%
Suburban (Zeeland)	3637	89%
Urban (Zeeland)	929	89%
Rural (Utrecht)	2148	84%
Urban (Utrecht)	2133	89%
Total	15632	88%

Table 1: Results on training set per region type

Region type (province)	Number of areas	% correct
Rural (Friesland)	197	97%
Rural (Gelderland)	548	96%
Rural (Flevoland)	524	93%
Rural (Groningen)	281	96%
Suburban (Groningen)	472	87%
Total	2022	93%

Table 2: Results on test set region type

Classification	Number of areas	% correct
Building	4954	97%
Construction work	105	55%
Road	4012	81%
Terrain	7345	88%
Water	1045	97%
Railroad	33	79%
Junction	160	20%

Table 3: Results on training and test set split up per type of object

2. The manual classification of the areas in the training and test set contains errors (4-5%). A human operator appears to perform badly on small areas in densely build-up regions and is often not consistent with the classification specifications in ambiguous situations.
3. The line input data contains errors that make it impossible to assign a correct area classification (0.5-1%). This is for example due to the fact that symbols are not exactly placed in the area they belong to or that areas are not correctly closed by hand. In these cases the areas get the special classification contradiction. In this way errors can be fou easily and corrected manually. This results in improvement of quality of the original map.

The time required for manual classification varies a lot, depending mainly on the type of region (rural of urban) and the quality of the input data, i.e. if some work has been done before to structure the spaghetti data. The average time in the experiment needed to calculate nodes and close area by hand for an area of 1 km^2 was 3 hours. Classification of the areas took about 6 hours per 1 km^2, followed by 3 hours checking and correction of errors in area formation and classification. In total it takes 12 hour per km^2. Automatic classification of an area of 1 km^2 takes about 6 minutes on a SUN UltraSparc. This means that the rule-based system can halve the time needed for conversion.

7 CONCLUSION

As described in the article we have designed a data model, both at the logical level and at the technical level, for a topologically structured topographic map. Based on the experience with the Cadastral map this model can be used both for standard products (update files) and ad hoc querying very well. Conversion of the input of spaghetti (non-structured) topographic lines, symbols, and text into well-structured and classified area features takes three steps: automatic structuring of area features geometry (not yet implemented), closing the area features by hand and rule-based classifying the area features. One could imagine specific interactive edit tools (for area features), but these have also not yet been implemented. However, the automatic area classification has been implemented and compared to human area classification.

The first results of the area classification are very promising. Different test regions were classified by hand and compared to the automatic classification. About 90 % of the areas were classified equal. The amount of errors made by the rule-based system is about equal to the number of errors made by a human operator, but the type of errors differ. Therefore, a combination of automatic classification followed by a visual inspection leads to a more reliable and consistent result than manual classification alone. The time required by manual classification varies a lot, but is on the average about six hours per test region. As the time required for the rule based classification is on the average six minutes, the gain is quite significant.

ACKNOWLEDGMENTS

At the time of this project the authors Marlies de Gunst and Peter van Oosterom worked at the Cadastre and participated in the project. In this project many persons, both inside and outside the Cadastre, have participated. First of all, Jurgen den Hartog and Bernardus Holtrop, both of the TNO Institute of Applied Physics, implemented the rule-based classification system. The method for the node computation is based on discussions with Martin Salzman, Peter Jansen and Louis de Bruijne of the Cadastre. Other aspects in this project were covered by Bart Maessen, Ernst-Peter Oosterbroek and Harry Uitermark of the Cadastre. Without the work of all mentioned persons it would not have been possible to write this paper.

REFERENCES

Anders, K. and Fritsch, D., 1996. Automatic interpretation of maps for data revision. ISPRS, Int. Archiv. of Photogr., and Remote Sensing 31(4), pp. 90–94.

Baumgart, B. G., 1975. A polyhedron representation for computer vision. In: National Computer Conference, pp. 589–596.

CA-OpenIngres, 1995. Object Magemenent Extention User's Guide, release 1.1. Technical report, Computer Associates.

de Hoop, S. and van Oosterom, P., 1992. Storage and manipulation of topology in Postgres. In: Proceedings EGIS'92: Third European Conference on Geographical Information Systems, EGIS Foundation, pp. 1324–1336.

Duda, R. O. and Hart, P. E., 1973. Pattern Classification and Scene Analysis. John Wiley, New York.

Giarratano, J. C., 1991. The CLIPS user's guide, CLIPS version 6.0. Technical report, NASA Lyndon B¿ Johnson Space Center, Software Technology Branch.

Haala, N. and Anders, K., 1996. Fusion of 2d-gis and image data for 3d building reconstruction. ISPRS, Int. Archiv. of Photogr., and Remote Sensing 31(3), pp. 285–292.

Karttakeskus, Helsinki, Finland, 1994. Fingis User Manual, version 3.85. Technical report, Karttakeskus.

KT-Datacenter Ltd., Riihimäki, Finland, 1994. X-Fingis Software V1.1, INGRES version. Technical report, KT-Datacenter.

Laurini, R. and Milleret, F., 1994. Topological reorganization of inconsistent geographical databases: a step towards their certification. Computers & Graphics 18(6), pp. 803–813.

Lemmen, C. H., Oosterbroek, E.-P. and Oosterom, P. J., 1998. New spatial data management developments in the netherlands cadastre. In: proceedings of the FIG XXI interenational congres, Brighton UK, commission 3, Land Information Systems, pp. 398–409.

Open GIS Consortium, Inc., 1998. Opengis simple features specification for sql. Technical Report Revision 1.0, OGC.

Ravi, 1995. Geo-information terrain model. A Dutch standard for: Terms, definitions and general rules for the classification and coding of objects related to the earths surface (NEN-3610). Technical report, Ravi Netherlands Council for Geographic Information, Amersfoort.

Uitermark, H. T., van Oosterom, P. J. M., Mars, N. J. I. and Molenaar, M., 1999a. Ontology-based geographic data set integration. In: M. H. Böhlen, C. S. Jensen and M. O. Scholl (eds), International Workshop on Spatio-Temporal Database Management STDBM'99, pp. 60–78.

Uitermark, H., Vogels, A. and van Oosterom, P., 1999b. Sematic and geometric aspects of integrating road networks. In: A. Vckovski, K. E. Brassel and H. Schek (eds), Second International Conference , Interoperating Geographic Information Systems, Interop99, pp. 177–188.

van Oosterom, P. and Maessen, B., 1997. Geographic query tool. In: JEC-GI'97, Joint European Conference and Exhibition on Geographical Information, Vienna, Austria, Vol. 1, pp. 177–186.

Žalik, B., 1999. A topology construction from line drawings using a uniform plane subdivision technique. Computer-aided design 31, pp. 335–348.