PERGAMON

Computers, Environment and Urban Systems
25 (2001) 509–528

Computers,
Environment and
Urban Systems

www.elsevier.com/locate/compenvurbsys

Spatial data management on a very large cadastral database

P.J.M. van Oosterom[a,*], C.H.J. Lemmen[b]

[a]*Faculty of Civil Engineering and Geosciences, Department of Geodesy, Delft University of Technology, Thijsseweg 11, 2629 JA, Delft, The Netherlands*
[b]*Kadaster, Department of Information Policy and Projects, Hofstraat 110, 7311 KZ Apeldoorn, The Netherlands*

Abstract

An overview is presented of spatial data management in the Netherlands Cadastre. As a result of more than three years research a major improvement in the accessibility and maintenance of 50 Gb available geometric data has been introduced. Quality assurance of topological relationships, maintenance of historic spatial data and flexible data access by a query tool is supported. The 'Spatial Location Code' used for spatial clustering and spatial indexing guarantees excellent performance of the relational database. This opens the possibility to develop new products, which can be delivered efficiently via networks. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Spatial data management; Cadastral database; The Netherlands

1. Introduction

Cadastral systems usually include a database containing spatially referenced land data, a set of procedures and techniques for systematic collection, updating, processing and distribution of data and a uniform spatial reference system. The Netherlands Cadastre (Kadaster) maintains a database representing spatial cadastral data and topographic data and a separate database with data on real rights and subjects. Source data are deeds from the notary and terrestrial surveys.

The spatial database contains the graphic representations of the boundaries of about 7.0 million cadastral parcels in the Netherlands, parcel identifiers, buildings, street names etc. The accuracy of this data set depends on the type of region; the original cadastral data are related to the scales 1:1000 and 1:2000. The topographic

---

* Corresponding author. Tel.: +31-15-278-6950; fax: +31-15-278-2745.
E-mail address: oosterom@geo.tudelft.nl (P. van Oosterom).

0198-9715/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved.

data set has a 1:2000 accuracy for rural areas and a 1:1000 or 1:500 accuracy for urban areas. The data set covers the whole country and contains all the important topographic objects like buildings, roads, railways, watercourses, bridges, street names etc. Unlike the cadastral data, the Kadaster does not own the large-scale topographic data itself. There is a joint ownership of municipalities, public utility companies and Kadaster, where Kadaster plays an important role in producing and maintaining these topographic data. Though the cadastral and the topographic data sets are quite different, they are both maintained in the same environment as described in this paper.

Increasing user requirements have been the main motivation for a redesign of the spatial data management processes and the related database. Instead of repeated full deliveries of spatial data in a certain region, more and more users have taken a subscription for periodic updates with only the changes to their copies of spatial data from Kadaster. Further, network users expect efficient and flexible geographic data browsers. In the future this will increase even more due the 'National Clearinghouse for Geo-Information' (de Gunst & van Oosterom, 1997; Kamel & Faloutsos, 1994; Van de Kieft & Kok, 1997). To support these user demands more flexible software was needed. Therefore, an object/relational database with geometric data types and spatial indexing was chosen as basis of the renewed system, in our case CA OpenIngres (ASK OpenIngres, 1994). An important aspect is the use of the Object Management Extension/Spatial Object Library of this database.

The redesign of spatial data management processes and the spatial database can be considered as successful after being in production for more than three years now, that is, since 1997. The database (50 Gigabytes) has been implemented for the production branches of Kadaster and is performing very well. This paper highlights some innovative developments during the design of this database, e.g. the Spatial Location Code for spatial clustering and indexing. However, the main innovation is the combination of full history support, introduction of explicit topology, maintenance of nation-wide unique object identifiers, integration of geometric and thematic attributes in one table, spatial clustering, spatial indexing, SQL based product development and the use of a query tool combined in one environment.

The context and reasons for the redesign of the spatial database are described in Section 2. Usage of geometric data types is presented in Section 3 in relation to the data-model of the renewed database. A short overview of spatio-temporal modelling aspects is given in Section 4, the developed Spatial Location Code for spatial indexing is introduced in Section 5. Some flexibility of SQL in a spatial data environment is documented in Section 6. In Section 7 a query tool to access the spatial and administrative data in an integrated manner is described. Conclusions and future work are given in Section 8.

2. Spatial database

Automation of the land administration started 25 years ago. System developments to computerise the cadastral map and the cadastral registration were separated. The automated system supporting the administrative and legal cadastral registration and information supply is operational nation-wide since 1990. The cartographic/ geometric system contains all applications regarding surveying and mapping (van Osch, 1998). The first version of this system became operational in 1988 supporting the establishment of new cadastral parcels, the production of data sets with geometric updates, quality improvement ('renovation') of geometric cadastral data, maintenance of the large scale topographic data, geodetic calculations, etc. The second version of this system includes the renewed database and runs since 1997 in production environment.

Main reasons for redesign of the database are flexible access to the spatial data, easy implementation of extensions, topological quality assurance, introduction and maintenance of historic data to produce update files and a good performance in spatial data access (Lemmen, Oosterbroek, & van Oosterom, 1996). The advantages are clear: development of new cartographic and integrated cartographic/administrative products can be supported in an easy way. Sophisticated query tools using an SQL based interface can be used for this purpose.

For flexible product development, the object/relational database CA OpenIngres (ASK OpenIngres, 1994) has been selected. A Spatial Object Library is included in the functionality of this database-product. This extension concerns a library, which supports storage and manipulation of the geometric data types *point*, *line*, *polygon*, *box*, etc. Operators *like* '*intersect*', '*inside*', '*distance*', etc. can be applied directly in SQL. Such functionality allows the storage of representations of spatial objects in a relational environment and the integration with non-spatial data can be organised in a very natural way.

The total size of the 15 'provincial' databases with spatial data is about 50Gb (Status 1997). This size will be increasing in regular production about 25% per year, because of the maintenance of history in the database. This serious increase is not only caused by regular cadastral updating, but is also caused by an ongoing quality improvement process. The geometric accuracy of the cadastral boundaries will be improved using the high accurate large-scale topographic data (Salzman, Hoekstra & Schut, 1998). This implies that all cadastral boundaries will get a new version in the database with more accurate co-ordinate pairs. This quality improvement process is planned to be finished in 2003.

As mentioned above in addition to databases with geometric data there is an administrative system for the registration of rights and subjects. On a yearly basis about 800,000 transactions in real estate are inscribed. About 150,000 transactions require topological manipulations on the cadastral data because of sub-division or consolidation of parcels. The spatial cadastral database is built up of about 25 million individual cadastral boundaries with topological attributes, each boundary containing 3.3 co-ordinate pairs on average. Each parcel and each boundary has a nationwide unique identifier, which also used in the topological references. The large-scale topographic data contain about 31 million line objects, 6.7 co-ordinate pairs on average. The database is prepared for inclusion of explicit topology in the topographic data. All together about 300 million co-ordinate pairs are represented in the spatial cadastral database.

3. Spatial data model

The spatial data are represented in the database using geometric data types such as 'point', 'line' and 'box'. In addition to the use of these data types, some other important new capabilities in the data model are storage of explicit topology and historic information. Furthermore, nation-wide unique identifiers have been introduced for *all* geographic objects, e.g. boundaries, topographic lines and unified surveying and mapping attributes.

The most important tables are boundary (cadastral boundaries), parcel (parcel identifiers), symbol (cartographic symbols), GCP (Geodetic Control Point), line (topographic lines) and text. The spatial extension of the objects in the tables boundary, parcel and line is indicated with a minimal bounding rectangle of type 'box'. The box covers a boundary or a topographic line or a complete parcel. The box can be spatially indexed and is useful for efficient retrieval purposes based on rectangle selections. There is no need for the geometric data type 'polygon', because the area features are stored topologically in the parcel table and boundary table using the so-called 'CHAIN-method'. The edges in boundary table contain references to other edges according to the winged edge structure (Baumgart, 1975), which are used to form the complete boundary chains. Currently, topology is only maintained for the cadastral data and not (yet) for the topographic data.

This approach allows calculations on correctness of topology after adjustment of the surveyed new boundaries to the cadastral data. Furthermore it opens the possibility to relate attributes to the boundaries between parcels, e.g. relation to the source documents of surveying, date of survey, names of persons locating the boundary, etc. If each parcel would be represented in the database by a closed polygon, it would be complicated to represent the basic object of cadastral surveying: one boundary between two neighbour parcels. Closed polygon representation would lead to double (or triple or even more) storage of all co-ordinates (except the territorial boundary), which complicates the data management in a substantial way. Closed polygon representation can result in the introduction of gaps and overlaps between parcels, which has no relation to reality. One more reason for the boundary-based approach is in the classification of boundaries: the administrative cadastral and political subdivision in sections (cadastral zones), municipalities and provinces is possible by classifying boundaries as 'section-boundary', 'municipal-boundary', 'province-boundary' or 'national boundary'. A 'national boundary' is by definition a 'province boundary' and a 'municipal-boundary' and a 'section-boundary' and a 'parcel-boundary'; etc.

The following attributes are included in the data model for all spatial features:

1. object_id, a nation-wide unique feature identifier for all objects represented in the database;
2. classif, classification code of the object, e.g. parcel boundary, parcel identifier, road (type), water, etc.;
3. location (of data type point) or shape is of data type line (50), a polyline up to 50 points, representing the cadastral boundaries, stored in a variable length way in the (object)/relational database;

4. sel_code, a selection code which indicates to which type(s) a geographic object belongs, e.g. cadastral data and/or large scale topographic data;
5. source of data, which is a reference to the field documents and files from total stations, or to the id of the photogrammetric project for large scale topographic mapping, etc.;
6. quality, which is the mode of data collection, e.g. terrestrial, photogrammetric and includes an accuracy code which denotes the deviation from the 'true' position;
7. vis_code, visibility code to classify less visible objects during photogrammetric data collection, e.g. because of trees;
8. l_area, official legal area, which is included in the official legal documents or deeds describing the transaction, in general this area is not equal to calculated area from the spatial cadastral boundary data; this attribute is introduced only for the parcel table.

A detailed overview of the parcel and boundary tables is given in Tables 1 and 2.

Table 1
Definition of the parcel table

Attribute name	Data type	Attribute description
Ogroup	Integer4	Group Id
Object_id	Integer4	Text/polygon object id (KEY together with t_max)
Slc	Integer4	Spatial location code, index
Classif	Integer4	Object classification
Location	Point	x,y co-ordinate pair of centroïde (parcel id)
D_location	Point	Delta x,y displacement of centroïde (parcel id)
Rotangle	Integer4	Rotation angle of centroïde (parcel id)
O_area	Float8	Cadastral boundary based calculated parcel-area
Bbox	Box	Bounding box covering the complete parcel
Object_dt	Date	Date of object creation; equal to t_min in this special case
T_min	Integer4	Date/time of parcel creation in the database
T_max	Integer4	Date/time of parcel deletion in the database
Sel_code	Char3	Selection code: object to be represented on cadastral and/or topographic data sets
Quality	Char2	Quality label: collection mode & accuracy
Vis_code	Char1	Visibility code; photogrammetric data collection
L_area	Float8	Official (legal) area of parcel
Municip	Char5	Municipality code, part of parcel identifier
Section	Char2	Cadastral section code, part of parcel identifier
Sheet	Char4	Sheet number of the original paper map, reference to paper archive
Parcel	Char5	Parcel id within cadastral section, part of parcel identifier
Pp_i_ltr	Char1	Part-parcel letter, in use to identify part of parcel before surveying
Pp_i_nr	Char1	Part-parcel number, in use to identify part of parcel before surveying
L_num	Integer4	Number of boundary references
Line_id1	Integer4	Reference to object_id of one surrounding boundary
Line_id2	Integer4	Reference to object_id of one boundary of the 1st enclave (if existing)

Table 2
Definition of the boundary table

Attribute name	Data type	Attribute description
Ogroup	Integer4	Group Id
Object_id	Integer4	Line object Id (KEY together with t_max)
Slc	Integer4	Spatial location code, index
Classif	Integer4	Object classification
Interp_cd	Integer1	Line interpolation code, line/arc
Shape	Line(50)	Line co-ordinate pairs, compressed
Fl_line_id	Integer4	Object_id of first line to the left side in begin point→BOUNDARY
Fr_line_id	Integer4	Object_id of first line to the right side in begin point→BOUNDARY
Ll_line_id	Integer4	Object_id of first line to the left side in end point→BOUNDARY
Lr_line_id	Integer4	Object_id of first line to the right side in end point→BOUNDARY
	integer4	
L_obj_id	Integer4	Object_id of parcel on right side→PARCEL
R_obj_id	Integer4	Object_id of parcel on left side→PARCEL
Bbox	Box	Bounding box covering boundary between 2 parcels
Line_len	Integer4	Length of boundary
Object_dt	Date	Date of survey of the boundary in the terrain
T_min	Integer4	Date and time of boundary creation in the database
T_max	Integer4	Date and time of boundary deletion in the database
Sel_code	Char3	Selection code: object to be represented on cadastral and/or topographic data sets
Source_cd	Char2	Source of data, reference to field sketch, photogrammetric project
Quality	Char2	Quality label: collection mode & accuracy
Vis_code	Char1	Visibility code; photogrammetric data collection
L_municip	Char5	Municipality code, part of the parcel id to the left side→ALT-PARCEL.1
L_section	Char2	Cadastral section code, part of the parcel id to the left side→ALT-PARCEL.2
L_sheet	Char4	Sheet number of the original paper map, part of the parcel id to the left side→ALT-PARCEL.3
L_parcel	Char5	Parcel id within cadastral section, part of the parcel id to the left side→ALT-PARCEL.4
R_municip	Char5	Municipality code, part of the parcel id to the right side→ALT-PARCEL.1
R_section	Char2	Cadastral section code, part of the parcel id to the right side→ALT-PARCEL.2
R_sheet	Char4	Sheet number of the original paper map, part of the parcel id to the right side→ALT-PARCEL.3
R_parcel	Char5	Parcel id within cadastral section, part of the parcel id to the right side→ALT-PARCEL.4

The text/label location-attribute in the parcel table is represented with the data type 'point'. A parcel has at least one reference to one of the surrounding boundaries and one reference to a boundary of each enclaves. If a parcel has more then one enclave, then these additional references from parcel to boundary are stored in another table, called parcel_over, which can store up to 10 references. In case more references are needed, more parcel_over records are used. The structure of the topological references and the relationship between parcels and boundaries is visualised in Fig. 1.

Maintenance of topological relations is combined with updating the cadastral and topographic data in 'X-Fingis' work files (Karttakeskus, 1994), which are used for communication with the database via the check-out/check-in mechanism; see Section 4. User interpretation is required to correct most of the topological errors, which can occur halfway during editing or when processing data from external sources. Errors can be detected automatically in an X-Fingis work file. X-Fingis is the geographic editor used by the Kadaster, because of its strong topology support.

Possible errors like overshoots, undershoots, double identifiers, missing identifiers, double lines, overlapping lines, arcs with interpolation mode line, lines with only one vertex, very small parcels (less then 10×10 cm), 'crossing' lines (no node point), 'spike' lines, self intersecting lines or self overlapping lines, lines without node points, many (neighbour) lines smaller then tolerance (minimum line size, which is 10 cm), lines with equal 'next' co-ordinate, etc. are relatively easy to detect. But X-Fingis has been extended some years ago with the functionality to detect more special cases like parcels, which are a neighbour to itself. A condition for correction is the complete availability of each parcel in one of the work files where correction is done.

The cadastral data are and remain topologically consistent. The next step in improvement of quality concerns the co-ordinates of the cadastral data (Salzman et al., 1998; van Osch, 1991). The cadastral data will be renovated to meet strict quality specifications. At the same time the cadastral data will be harmonised with the large-scale topographic data of the Netherlands. Quality attributes related to cadastral boundaries will be set to correct realistic values. When this project is finished in 2003 the cadastral data will meet optimal quality specifications for topology, geometry and geometric attributes.

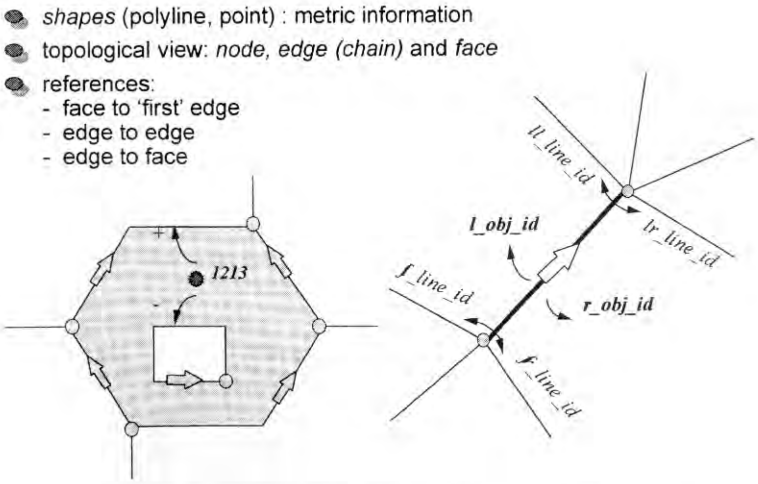


Fig. 1. Topology model in the spatial database of the Kadaster.

4. Spatio-temporal aspects

Recently, quite a lot of attention has been paid to methods of storing and manipulation spatial-temporal data (e.g. Al-Tana, Snodgrass, & Soo, 1994; Frank, 1994; Heres, 2000; Peuquet & Wentz, 1994; Worboys, 1994). The updates in the spatial database of the Kadaster are related to changes of a discrete type in contrast to more continuous changes such as natural phenomena or stock rates. It was therefore decided to implement history on record level. This in contrast to implementing history on attribute level, which requires specific database support or will complicate the data model significantly in a relational database. Further, instead of storing the old and new states, it would also have been possible to store the events. However, in that case it would not be easy to retrieve the situation at any given point in time. Therefore, it was decided to store the old and new states at the record level.

Though very complex solutions have been described in the spatio-temporal literature, a simple solution was chosen. Every object type is extended with two additional attributes: t_min and t_max. When a new object is inserted, it gets the current time as value for t_min during check-in, and t_max gets a special value (far in the future). When an attribute of an existing object changes, it is not updated, but the complete record is copied with the new attribute value. The old version gets current check-in time for its t_max value and the new version (record) gets this time value for t_min. Using this technique it is possible to retrieve the data from any given time in history including correct topology references and it is also possible to produce the changes over a given period efficiently; see Section 6. Those changes are delivered every month or quarter of a year as 'update files' to the users of the cadastral and/or topographic data. Of course, every new user first gets a full delivery.

4.1. Example

Table 3 shows the contents of a database, which contained on '12 jan' one line with object_id 1023. On '20 feb' this line was split into two parts: 1023 and 1268; see Fig. 2. Finally, the attribute quality of one of the lines was changed on '14 apr'.

There is a difference between the *system* (or *transaction*) time, when the recorded object changed in the database, and the *valid* (or *user*) time, when the observed object changed in reality. In the data model t_min/t_max are system times. Further, the model includes the user time attribute object_dt (or valid_tmin) when the object

Table 3
Database content and some changes through history

Object_id	Shape	Quality	T_min	T_max
1023	(0,0), (4,0), (6,2)	1	12 jan	20 feb
1023	(0,0), (4,0)	1	20 feb	14 apr
1268	(4,0), (6,2)	1	20 feb	–
1023	(0,0), (4,0)	2	14 apr	–

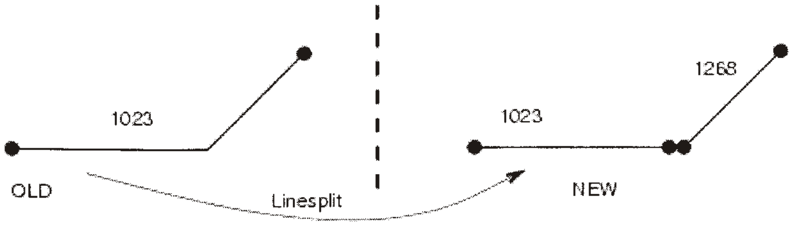


Fig. 2. Splitting a polyline (boundary) in two parts on '20 feb'.

was observed. Perhaps in the future also the attributes last_verification_dt and valid_tmax could be included, which would make it a *bitemporal* model. The *unique identifier* (key) is the pair (object_id, t_max) for every object version in space and time. It should be noted that using t_min in the key would have been more elegant as this value never changes. For the topological references, only the object_id is used to refer to another object and not t_max. This is extremely important as in the situation that a referred object is updated and keeps its object_id, then the reference (and therefore the current object) does not change. This avoids, in a topologically structured data set, the propagation of one changed object to all other objects as all objects are somehow connected to each other. Only in case the object_id of a referred object has changed (becomes a different object), the referring object is also updated and a new version of the referring object is created.

A query producing all historic versions of a given object only needs to specify the object_id and need not specify any restrictions based on the time attributes. This does work for simple object changes, but does not work for splits, joins, or more complicated spatial editing. However, this predecessor and successor information can always be obtained by using spatial overlap queries with respect to the given object over time, that is, again not specifying t_min/t_max restrictions.

As the database must always be in a consistent state, it should not be polluted with 'temporary' changes that are required during the topological edit operations. This is the motivation for the introduction of a *temporary work copy* for the spatial edit program; e.g. X-Fingis. The copy is made during *check-out*. This is only possible in case no other work areas overlap the requested region with respect to the themes to be edited. The database is brought from one (topologically) consistent state to another consistent state during a *check-in*. It is important that all changes within the same check-in get the same time stamps in t_min/t_max (system time as always). This architecture also has the advantage that it enables an easy implementation of a high-level 'cancel' operation (rollback).

What exactly should be locked when a user specifies a rectangular work area? Of course, everything that is completely inside the rectangle must be locked. This is achieved at the *application* level: check-out and check-in. Objects that cross work area boundaries could also be locked, but this may affect a large part of the database; e.g. very long roads. Other users may be surprised to see when they want to check-out a new non-overlapping part (rectangle); this is impossible due to elongated objects that are locked. Therefore, the concept of *partial locks* is introduced for these objects: the *coordinates* of the line segment crossing the boundary of the work area are not allowed to change. Together with the fact that the rectangular work areas can never overlap, this implies that the other changes to the edges and faces that cross the borders of two nearby work areas are *additional* and can be merged in the database. Therefore these objects do not have to be locked, but have to be checked in with some additional care. All check-ins are processed sequentially.

Errors in the past with respect to data collecting or entering pose a difficult problem: should these be corrected by changing the history (t_min/t_max)? Because of possible consistency problems it was decided not to do so. An alternative solution is to mark error objects by setting an additional attribute error_date.

5. Spatial clustering and spatial indexing

The Spatial Location Code (SLC) is developed in the Kadaster and is in use for spatial clustering and spatial indexing geographic objects in very large spatial databases (van Oosterom & Vijlbrief, 1996). It combines the strong aspects of several known spatial access methods [Quadtree (Samet, 1984, 1989), Morton code (Orenstein & Manola, 1991) and Field-tree (Frank, 1983; Frank & Barreta, 1989; Kleiner & Brassel, 1986)] into one SLC value per object (Fig. 3). The SLC technique has been developed because earlier versions of CA OpenIngres did not provide spatial clustering and spatial indexing.

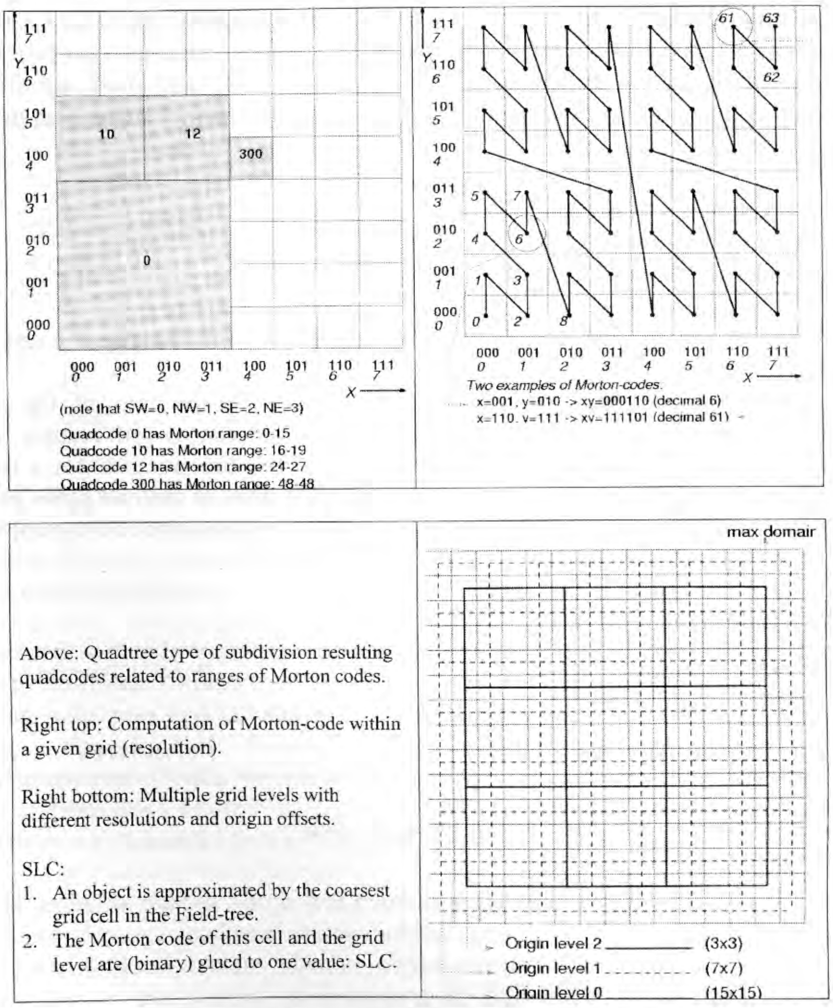
The unique aspect of the SLC is that both location and extent of possible non-zero-sized objects are approximated by a single value. These SLC values can then be used in combination with traditional access methods, such as the b-tree (Comer, 1979), available in every database. The typical query response time for spatial objects is reduced by orders of magnitude for a very large spatial data set. The SLC is used in two-dimensional space, but the SLC is quite general and can be applied in higher dimensions.

The performance test of the SLC-based index proved that the required (pre-defined) time limits are obtained (Cadastre, 1995). 'Normal' spatial selections within a rectangular area resulting in a few thousand objects require a few seconds. The explanation is in the more optimal spatial clustering techniques which results in storage of spatial objects 'near to each other' on disk if the objects are 'near to each other' in reality. Spatial range queries now only need a few disks accesses, without clustering it could have been the case that every object would require a separate disk access, which would be very slow. Check-in times are really good because only new, changed, and deleted objects are checked-in in the database. Those test results have been the basis for the decision to introduce the relational database in production environment.

Since version 2.0 of CA OpenIngres, a spatial index is also available in the database itself. This index is an R-tree and the Kadaster has tested the performance; it is about two times faster then using the SLC only. The SLC is still used for spatial clustering. This creates good conditions for fast access to the very large spatial databases. Fast access is not only important for data maintenance but also for external access, e.g.

supply of a cadastral map via a network used by about 4000 external customers (notary, municipalities, real estate brokers, etc.) and in the future also for applications based on web servers to provide on-line cadastral information to the customers.

In CA OpenIngres, the primary storage structure has to take care of spatial clustering of the data. In the spatial database of the Kadaster, this is realised by using a b-tree structure on the slc, ogroup. The ogroup indicates to which group the record belongs, and the SLC is used for spatial data access (van Oosterom & Vijlbrief, 1996). Note that the data is not clustered on time. As mentioned above, CA-OpenIngres 2.0 also supports a native spatial index: the Hilbert R-tree (Kamel &



(note that SW=0, NW=1, SE=2, NE=3)

Quadcode 0 has Morton range: 0-15
Quadcode 10 has Morton range: 16-19
Quadcode 12 has Morton range: 24-27
Quadcode 300 has Morton range: 48-48

Two examples of Morton-codes.
x=001, y=010 -> xy=000110 (decimal 6)
x=110, y=111 -> xy=111101 (decimal 61)

Above: Quadtree type of subdivision resulting quadcodes related to ranges of Morton codes.

Right top: Computation of Morton-code within a given grid (resolution).

Right bottom: Multiple grid levels with different resolutions and origin offsets.

SLC:

1. An object is approximated by the coarsest grid cell in the Field-tree.
2. The Morton code of this cell and the grid level are (binary) glued to one value: SLC.

Fig. 3. The principles of the Spatial Location Code (SLC).

Faloutsos, 1994) which is put directly on the bbox. The SLC is still used for spatial clustering. Efficient selection based on object_id is supported by a unique secondary index (isam) on object_id, t_max. The secondary indices on left/right references to faces are also shown in Table 4.

6. Spatial and temporal queries in SQL

Some examples of SQL queries will show the flexibility of the extended relational database: e.g. find all parcels larger than 10,000 m^2 in a rectangular region

```
Select * from parcel
Where o_area > 10000 and
Inside(bbox,((12000,363000),(40000,382000))) = 1
```

or find all geodetic control points with quality code 'T1' within a given circle with radius of 2 km:

```
Select * from GCP /* Geodetic Control Point */
Where quality = T1 and
Distance(location,(45000,393000)) < 2000
```

A query tool can be used to formulate the SQL queries and visualise the results in a user-friendly way; see Section 7.

As explained in Section 4, after an initial full delivery of the data set, the customers receive periodic update files, which contain the differences with respect to the previous delivery (Lemmen & van Oosterom, 1995). The time interval for a typical update file starts at the begin point in time t_beg and stops at the end point in time t_end. There are two ways of interpreting the begin (t_beg) and the end (t_end) times

Table 4
Storage and index structures for the boundary table

```
Modify boundary to btree on slc,ogroup                          /* spatial clustering */
 With compression = (nokey,data);

Create index x_boundary0 on boundary (bbox)                     /* spatial index */
 With structure = rtree;

Create unique index x_boundary1 on boundary (object_id,t_max)
 With structure = isam;

Create index x_boundary2 on boundary (l_obj_id)
 With structure = isam;

Create index x_boundary3 on boundary (r_obj_id)
 With structure = isam;
```

related to an update file: as a complete time *interval* or as two individual *points* (*instants*) in time. In the second case, the customer is not interested in temporary versions of the objects between the two points in time t_beg and t_end. First, two queries to find all changes over time interval (t_beg, t_end] including t_end, with delivery of all temporary object versions.

```
/* deleted/updated objects, time interval */
Select * from line l where t_beg < l.t_max and l.t_max < = t_end;

/* new/updated objects, time interval */
Select * from line l where t_beg < l.t_min and l.t_min < = t_end;
```

In case an object is updated two times, two versions of old objects (old: x,t1 and x,t2) and two versions of new objects (new: x,t2 and x,-) will be included in the update file. Note that the object version (x,t2) is included in both the set of old objects and in the set of new objects; see the example below:

```
                                      Object_id=x,
                    Object_id=x,      t_max=-
Object_id=x,        t_max=t2       |------------->
t_max=t1     |---------------------|
-------------|                    t2
             t1

--O-------------------------------------X----->
t_beg      (time line)                 t_end
```

The next two SQL queries will produce only changes comparing the two points in time t_beg and t_end, excluding all temporary versions. This means that the object versions have to overlap in time either t_beg (deleted/updated objects) or t_end (new/updated objects):

```
/* deleted/updated objects, two points in time */
select * from line l where
 t_beg < l.t_max and l.t_max < = t_end and l.t_min < = t_beg;

/* new/updated objects, two points in time */
select * from line l where
 t_beg < l.t_min and l.t_min < = t_end and t_end < l.t_max;
```

The example queries above will produce only one version of the old object (old: x,t1) and only one version of the new object (new: x,-). This approach will also filter out the temporary versions when an object is really deleted (but also changes one or more times in the time interval) or when an object is really new; see the examples below:

```
                              Object_id=n
             Object_id=n,     t_max=-
 new         t_max=t3        |-------------->
 object     |----------------|
           t1                t3

                   Object_id=d
Object_id=d,       t_max=t4             deleted
T_max=t1      |-------------------|     object
-------------|                   t4
             t2

--O----------------------------------X----->
t_beg     (time line)              t_end
```

This way of selecting historical data was a big improvement in the way to produce update files for users of cadastral spatial data. Until the relational database was introduced the production of those update files was based on geometric comparison of cadastral and/or topographic data in the users area. This means a very large archive with digital spatial data sets had to be maintained and a lot of computations had to be executed to compare the actual situation with an earlier situation valid at the moment of deliverance of the last update file.

Although some users prefer on-line access to retrieve actual data there are still a lot of users who prefer to get copies of cadastral and topographic data to be maintained with update files.

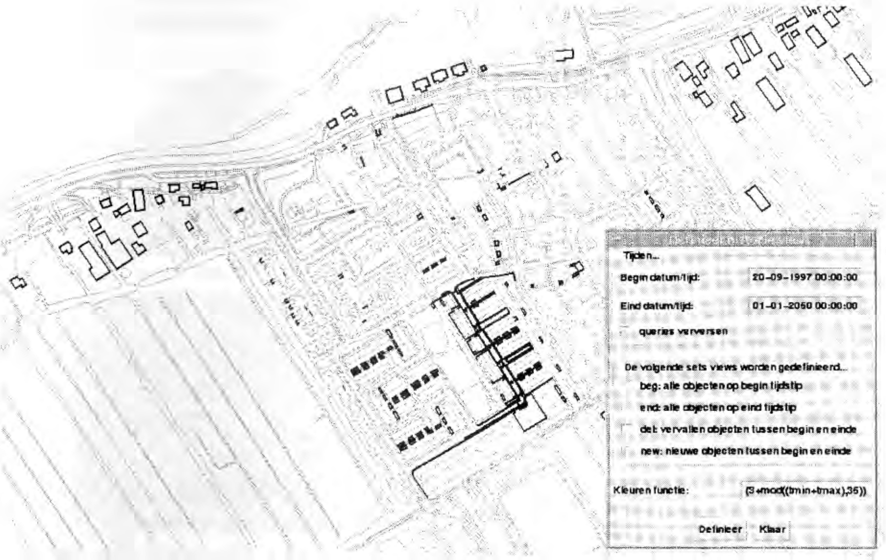
7. Cadastral query tool

Since the introduction in 1999 of the cadastral query tool based on GEO+ + (van Oosterom & Maessen, 1997; van Oosterom, Maessen, & Quak, 2000a, b; Vijlbrief & van Oosterom, 1992, 1993), all provincial spatial and administrative databases are periodically copied and combined into one query tool database available for cadastral data supply. A few numbers describing the size of the spatial cadastral and topographic data sets have already been given in Section 3. It should be noted that due to the maintenance of history in the database, these numbers grow every year. The administrative databases contain the following amount of data (without history): 7,500,000 objects (that is, ground parcels, part-of-parcels, or apartments), 9,700,000 object addresses, 7,100,000 subjects (that is, persons or organisations including their subject address), 10,100,000 right records (relationship between object and subject), 1,900,000 object limitations (legal notifications, restricting the use of the object due to some reason), etc. Though the spatial and administrative production databases are physically apart, they are logically related. The relationship between the parcels in the spatial cadastral data and the administrative data is through the nation-wide unique parcel numbers. In the query tool database the original data models (base tables) are not changed. Database *views* are used to present the data in a more appropriate manner:

1. integrate data from different tables in one view (e.g. parcel and object tables);
2. visualise historic data (Fig. 4);
3. different geometric visualisations of the same table (e.g. parcel as a point object);
4. derive default cartographic attributes, such as colour, width, symbol type, from other attributes;
5. derive thematic aggregates without storing the result (Fig. 5); and
6. present (encoded) attributes in a more clear way.

Examples of the last application of views are: time stamps encoded by integers can be visualised as readable strings such as '22-04-1998 09:52:50' or legal right codes, shortly coded by two characters, which can be represented better with a full string.

In one example application of the query tool, a digital version of pipelines was given to the Kadaster. These were then entered into the query tool database and confronted with the parcels; see Fig. 6. This was not a simple query in the relational database, because the geographic data model of the Kadaster is based on topology; see Section 3. Within a relational database an overlap or cross operation based on parcels modelled with topology is impossible. Therefore this operation was implemented in the interface (front-end) part of the query tool; see the inset window of Fig. 6.



Tijden...

Begin datum/tijd: 20-09-1997 00:00:00

Eind datum/tijd: 01-01-2050 00:00:00

queries verversen

De volgende sets views worden gedefinieerd...

beg: alle objecten op begin tijdstip

end: alle objecten op eind tijdstip

del: vervallen objecten tussen begin en einde

new: nieuwe objecten tussen begin en einde

Kleuren functie: (3+mod((tmin+tmax),35))

Fig. 4. All changed objects since '20 September 1997'.

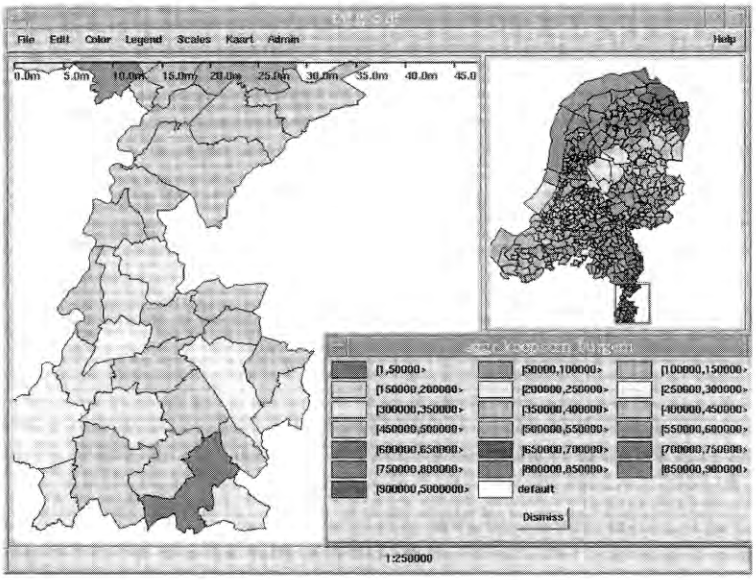
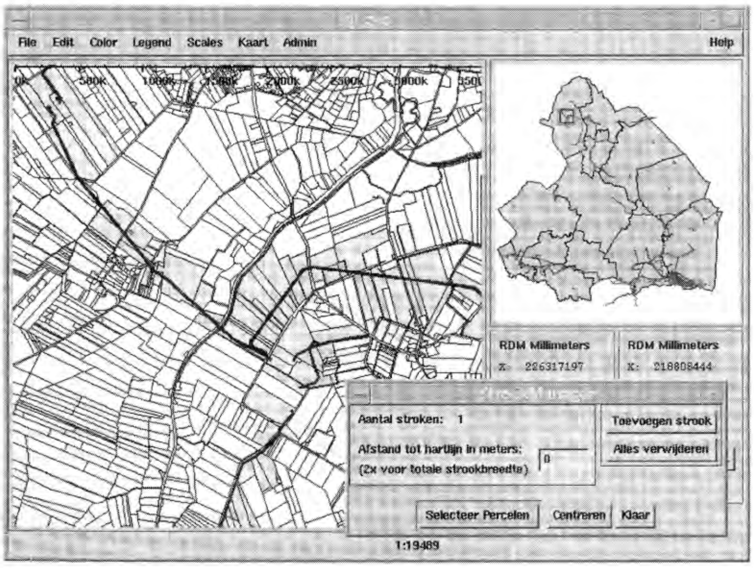


Fig. 5. The thematic aggregate view 'average parcel price per municipality' (note: spatial aggregate 'municipality' is derived from the parcels and stored in the query tool database).



RDM Millimeters
X: 226317197

RDM Millimeters
X: 218808444

Aantal stroken: 1

Afstand tot hartlijn in meters:
(2x voor totale strookbreedte) 0

Fig. 6. Parcels crossed by a pipeline are selected (note: this is a difficult query within a relational database, because the parcels are based on topology).

Besides the three displayed example applications (Figs. 4–6) and many small ad-hoc queries, the query tool has been used for several other projects during the last couple of years. A few will be mentioned here: collecting statistics with respect to 'akte posten' (that is, parcels which have to be surveyed because of changes such as splitting or reorganising), finding potential parcels owned by farmers which may be used for land exchange (lots at a large distance from the farm), finding parcels which may be merged because they have equal legal status (e.g. same owner), finding all parcels of interest to the Ministry of Agriculture which have to be outside given built-up area polygons, finding all parcels and their owners on which a protected monument is located, deriving a postal code map based on the cadastral parcel and the addresses from the administrative databases, deriving the type of house (free-standing house, corner house, middle house in a row, two under one roof house, apartments) by overlaying the topographic buildings (which are not classified) and the cadastral data, deriving aggregate information (e.g. number of changed topographic objects per month) and visualising the result on a geometric aggregation of the same level (e.g. municipality), and so on.

The query tool has proven to be a very useful tool within the Kadaster. Further enhancements will improve the usefulness of the query tool even more. Of course the users also have new functional wishes after using the system for a while. Their first wish, is to be able to specify his/her own views, based on joins, and having attributes from multiple tables in the result. Their second wish is to have more import and export functionality. Their third wish is to have a more up to data query tool database (now updated two times per year). Therefore, instead of loading full data sets two times per year to the query tool database, in the future the data will have to be replicated more frequently from spatial and administrative production databases. Instead of using full data set copies, this can be done more efficiently by only transferring the changes. These update files are standard products in the source systems and can be obtained every month. Summarising, the cadastral query tool gives easy interactive access to those data with a very good performance, most queries are managed within a few seconds.

8. Conclusions and future work

A renewed very large spatial database has been implemented within the Kadaster. This means a fully integrated database with all data (geometry, topology, history, and surveying/mapping attributes) in one relational database is available now. Spatial clustering and spatial indexing is included in the functionality of the database. The combination of all these aspects can be considered as a real innovation. Due to the open architecture, it is also possible to integrate this spatial database with other (geographic or administrative) data sets.

The database is the basis for a sustainable and flexible geo-information supply for internal (Kadaster) and external (users of cadastral data) purposes. For external purposes, the production of the update files based on the temporal attributes is a great improvement allowing more customers to be served. Further, the customers of the 'Kadaster network' (notary, municipalities, real estate brokers) do receive their requested A4-size parcel maps much quicker than before due the spatial data clustering and indexing. Internally there is already an improvement in the efficiency because of more efficient selections in the database for spatial editing, benchmarks indicate 10 times faster check-out times than before the renewal. The editing itself is less complicated because of the guaranteed topological quality of the cadastral map as there are no more topology errors from the past. Ongoing activities like a nation-wide cadastral data geometric quality improvement and the future introduction of explicit topology in the large scale topographic data will be based on the same model. Furthermore, the database is the first step in a complete redesign of the systems and infrastructure in the Kadaster. The integration of all provincial databases to a nation-wide database with cadastral and large scale topographic data in a seamless database is under development. The performance of the database is optimal to support this. Query tools will be used for an easy, efficient, and flexible definition of new special products to meet customer's ad-hoc wishes.

Externally one of the major developments is the National Clearinghouse for Geo Information. The operational version of the National Clearinghouse supports the supply of meta information on available geographical data sets for a specified area. A next step could be the (on-line) access to selected cadastral data via the Clearinghouse, e.g. by applying GeoShop (de Gunst & van Oosterom, 1997; van den Berg, Tuijnman, Vijlbrief, Meijer, Uitermark & van Oosterom, 1997). The GeoShop pilot allowed Internet access to several Geo-information sources at the same time. The GeoShop client has been completely implemented on platform independent Java technology and can manage both vector and raster data. On GeoShop servers several databases can be accessed (Informix, CA OpenIngres) and a number of file-based formats like DXF. The communication protocol has been standardised by the Open GIS Consortium on the basis of the 'web mapping test bed' from 1999 (Open GIS Consortium, 2000).

Extremely efficient execution of queries can be supported by a Very Large Memory (VLM) system, where the database is contained completely in the computer system main memory. Especially the (Internet) geo-data supply can be performed very well in a VLM environment, because one server can handle many requests per second. Pure VLM is *not* acceptable for the edit environment, because the back up on disk is missing in case of calamities. This means separation of edit- and supply environment, which is also desirable from the data protection and security policy.

Acknowledgements

Important contributions to (parts of) this paper have been given by the following group of people: Henri Aalders, Tapio Keisteri, Tjeu Lemmens, Marcel van de Lustgraaf, Joep Mathijssen, Esa Mononen, Ernst Peter Oosterbroek, Berry van Osch, Bertus Padberg, Tero Pietila, Johan Smit, Jantien Stoter, Theo Tijssen, Herman Welleweerd, Kees Westerlaken, and two anonymous reviewers for CEUS.

References

Al-Taha, K. K., Snodgrass, R. T., & Soo, M. D. (1994). Bibliography on spatio-temporal databases. *International Journal of Geographical Information Systems, 8*(1), 95–103.

ASK OpenIngres (1994). INGRES/Object Management Extension Users Guide. Release 6.5.

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

Berg, C. van den, Tuijnman, F., Vijlbrief, T., Meijer, Co., Uitermark, H., & Oosterom, P. J. M. van (1997). Multi-server internet GIS: standardization and practical experiences. In *International Conference and Workshop on Interoperating Geographic Information Systems, Santa Barbara, California, USA, December 3–4 and 5–6, 1997, interop'97* (pp. 365–378). Boston: Kluwer Academic Publishers.

Cadastre. Testgevallen LKI 3.26. Technical report, Kadaster, July 1995. Internal report in Dutch.

Comer, D. (1979). The ubiquitous B-Tree. *ACM Computing Surveys, 11*(2), 121–137.

Gunst, M. de, & Oosterom, P. J. M. van (1997). Network Computers at the Dutch Cadastre. 46th Photogrammetric Week, Stuttgart, September 1997.

Frank, A. U. (1994). Qualitative temporal reasoning in GIS – ordered time scales. In *Proceedings of the 6th International Symposium on Spatial Data Handling* (pp. 410–430). September 1994. Edinburgh, Scotland.

Frank, A. (1983). Storage methods for space related data: the Field-tree (Technical Report Bericht No. 71), Eidgenossische Technische Hochschule Zurich, June 1983.

Frank, A. U., & Barrera, R. (1989). The Field-tree: a data structure for geographic information system. In *Symposium on the Design and Implementation of Large Spatial Databases, Santa Barbara, California* (pp. 29–44). July 1989. Berlin: Springer-Verlag.

Heres (2000). Time in GIS: *Issues in spatio-temporal modelling*. Netherlands Geodetic Commission, Publications on Geodesy, No. 47, May 2000.

Kamel, I., & Faloutsos, C. (1994). *Hilbert R-tree: an improved R-tree using fractals*. VLDB Conference 1994.

Karttakeskus (Map Centre), Helsinki, X-Fingis Software Documentation.

Kieft, A. van de, & Kok, B. (1997). The development of geo data meta service for the Netherlands. In *Third Joint European Conference & Exhibition of Geographical Information (JEC-GI'97)*. April 1997. Vienna.

Kleiner, A., & Brassel, K. E. (1986). Hierarchical grid structures for static geographic data bases. In *Auto-Carto London* (pp. 485–496). London: Auto Carto.

Lemmen, C. H. J., & Oosterom, P. J. M. van (1995). Efficient and automatic production of periodic updates of cadastral maps. In *JEC95, Joint European Conference and Exhibition on Geographical Information The Hague, The Netherlands* (pp. 137–142). March 1995.

Lemmen, C. H. J., Oosterbroek, E. P., & Oosterom, P. J. M. van (1996). New spatial data management developments in the Netherlands Cadastre. In J. Ryttersgaard, *GIS/LIS for sustainable developments, GIS/LIS in the future* (pp. 39–49). October 1996. Copenhagen.

Oosterom, P. van (1997). *Spatial Data processing with CA-OpenIngres 2.0*. Paper presented to CA World, New Orleans, July 1997. (Published on CD by Computer Associates).

Oosterom, P. van (1997). *Maintaining consistent topology including historical data in a very large spatial database*. Seattle, WA: Auto Carto 13. April 1997.

Oosterom, P. van, & Vijlbrief, T. (1996). The spatial location code. In *Proceedings of International Symposium on Spatial Data Handling*, Delft, The Netherlands, August 1996.

Oosterom, P. van, & Maessen, B. (1997). *Geographic query tool*. Presented to JEC-GI'97, Vienna, April 1997 (pp. 177–186).

Oosterom, P. van, Maessen, B., & Quak, W. (2000a). Querytool: design, implementation and applications. In *Proceedings SPRS2000*. Amsterdam, July 2000.

Oosterom, P. van, Maessen, B., & Quak, W. (2000b). Spatial, thematic, and temporal views. In *Proceedings SDH2000, 9th International Symposium on Spatial Data Handling* (pp. 8b. 37–52). August 2000.

Open GIS Consortium (2000). Open GIS Web Map Server Interfaces Implementation Specification, reversion 1.0.0 (report no. 00-028), April 2000.

Orenstein, J. A., & Manola, F. A. (1988). PROBE spatial data modelling and query processing in an image database application. *IEEE Transactions on Software Engineering, 14*(5), 611–629.

Osch, G. M. van (1991). Cadastral LIS in the Netherlands. In *Proceedings of the OEEPE-workshop on data quality in land information systems* (OEEPE official publication No. 28; pp. 71–90). Frankfurt: IFAG.

Osch, G. M. van (1998). *LKI: Surveying and Mapping in the Dutch Cadastre* (Version 1998). Kadaster Apeldoorn.

Peuquet, D. J., & Wentz, E. (1994). An approach for time-based analysis of spatio-temporal data. In *Proceedings of the 6th International Symposium on Spatial Data Handling*, Edinburgh, Scotland (pp. 489–504). September 1994.

Samet, H. (1984). The quadtree and related hierarchical data structures. *Computing Surveys, 16*(2), 187–260.

Samet, H. (1989). *The design and analysis of spatial data structures*. Reading, MA: Addison-Wesley.

Salzman, M., Hoekstra, A., & Schut, T. (1998). Cadastral map renovation, paper presented at the FIG XXI International Congres, Brighton UK, 19–25 July 1998.

Vijlbrief, T., & Oosterom, P. van (1993). GEO+ + systeem: een uitbreidbaar GIS met consistentieregels. *Nederlands Geodetisch Tijdschrift Geodesia, 35*(6), 274–281 (in Dutch).

Vijlbrief, T., & Oosterom, P. van (1992). The GEO+ + system: an extensible GIS. In *Proceedings of the 5th International Symposium on Spatial Data Handling Charleston, South Carolina* (pp. 40–50). Columbus, OH: International Geographical Union IGU. August 1992.

Worboys, M. F. (1994). Unifying the spatial and temporal components of geographical information. In *Proceedings of the 6th International Symposium on Spatial Data Handling*, Edinburgh, Scotland (pp. 505–517). September 1994.