QUERYTOOL: DESIGN, IMPLEMENTATION AND APPLICATIONS

Peter VAN OOSTEROM[(1)]**, Bart MAESSEN**[(2)]**, Wilko QUAK**[(1)]
[(1)] TU Delft, Department of Geodesy, Thijsseweg 11, 2629 JA Delft, The Netherlands
[(2)] Dutch Cadastre, P.O. Box 9046, 7300 GH Apeldoorn, The Netherlands
oosterom@geo.tudelft.nl, bart.maessen@kadaster.nl, quak@geo.tudelft.nl

KEY WORDS: Query, Spatio-temporal, Integration, Data aggregation, User interfaces

ABSTRACT

This article describes the Querytool, a platform for querying multiple geographic data sets and associated thematic (legal) data available at the Cadastre in one integrated system. The system is nationwide and available for analyzing and performing consistency checks on the Cadastral data. It is designed to support ad hoc queries covering both the spatial and the thematic part of the data. The system is based on two important components: a relational DBMS and a (geo)graphic user interface for comfortable query formulation and showing the results. The usefulness of the tool, is shown by three quite different (implemented) cases from the real world, each illustrating different aspects of the system. These applications are based, among others, on the following generic concepts: spatial aggregates, historic spatial data, spatial join, integrated geometric and thematic data, SQL and shell scripts, using external spatial data, and spatial joins involving topologically structured data. As the relational DBMS can not execute the spatial join with topologically structured data, this is implemented in the (geo)graphic user interface.

1 INTRODUCTION

In this article the integration of multiple geographic data sets and associated thematic (legal) data in one database is described. The geographic data sets consists of large scale topographic data and the cadastral maps of the different provinces. Associated with the cadastral maps is thematic (legal) data, also organized per province. The relationship between the parcels on the cadastral map and the thematic data is through the nationwide unique parcel numbers. The cadastral maps are based on a topological structured model and manipulating area features in such a model requires navigation using the references to the boundaries. In addition, the topographic maps and the cadastral maps contain the full (update) history since the introduction in 1997. This is not (yet) the case for the thematic (legal) data.

In the Dutch Cadastre such a database as described above has been set up. It is nationwide and available for analyzing and performing consistency checks on the Cadastral data. The goal being to improve the quality of the source data. The purpose is to create an environment with easy access to all the data. Therefore a user friendly interface is needed on top of this database. This was realized by using the standard GIS package GEO++ together with custom made add-on's. GEO++ can best be described as a general Querytool for relational databases with spatial extensions, similar to the OpenGIS SFS/SQL (Buehler and McKee, 1998, Open GIS Consortium, Inc., 1998). The custom made add-on's are used for: *easier access to data:* just one button, instead of querying 4 tables; *analysis not possible in a relational DBMS:* e.g. intersection of a topologically structured area feature with a polyline; and *introduction of new interface concepts:* e.g. the 'active set'. In summary the Querytool can be used for easy querying, analysis and visualization of data in one nation-wide database. Besides more or less complicated ad-hoc queries which can be dealt with in the (geo)graphic user interface, either the standard GEO++ part or one of the add-on parts, the Querytool database has been used in many applications.

Section 2 gives a short overview of the Querytool system including a description of the computed spatial aggregates, sizes of the geometric and thematic data sets, and some remarks how good performance is obtained. The first application 'producing bills based on changes in the topographic map' (Section 3) uses spatial aggregates (municipalities), historic data and the spatial join. The second application 'deriving zipcode map from cadastral data' integrates geometric data (cadastral map) and thematic data (addresses) and uses a combination of SQL and shell scripts to derive the zipcode map; see Section 4. The third application, described in Section 5, 'quality improvement of the registration of legal notifications' again integrates geometric and thematic data (legal notifications related to parcel), but also uses a spatial join between topologically structured parcels and linear pipe lines (data from third party). As the relational DBMS can not execute the spatial join with topologically structured data, this is implemented in the (geo)graphic user interface. The paper concludes with a short list of other implemented applications and some future developments in Section 6.

2 SYSTEM ARCHITECTURE

The Querytool system is based on the Ingres DBMS (ASK-OpenIngres, 1994, van Oosterom, 1997) and the GEO++ GIS package (IJsselstein and Kap, 1995, Professional Geo Systems (PGS), 1996, Vijlbrief and van Oosterom, 1992). The administrative data and geometric data models have been published before in (Lemmen et al., 1998, Lemmen and van Oosterom, 1995, van Oosterom, 1997). The integration of these models is realized through views as described in (van Oosterom et al., 2000) together with many other design and implementation aspects of the Querytool. Below we give some details on specific parts of the Querytool architecture. Thematic overview maps are created by aggregate views on the administrative data and combing the result with the spatial aggregates, again using views. Finally, views are also used to deal with the time aspect of the spatial-temporal data.

Spatial aggregates Spatial aggregations, i.e. larger spatial units derived form the cadastral parcels, are stored in the database. These aggregates include: sheets, sections, cadastral municipalities, legal municipalities and provinces. The spatial aggregates are used as a basis for visualizing aggregated thematic data and for orientation purposes. Spatial aggregations, that is, the larger spatial units derived from the parcels are stored in the database. Defining the thematics aggregates in a generic manner still has to be improved in the Querytool. This is quite a challenge as there are many degrees of freedom when specifying an aggregation: spatial unit (municipality, province); temporal unit (one moment in time or even a period); aggregate function (sum, min, max, avg); thematic attribute; and additional constraints to the selection

Size of the data size A few numbers describing the size the geometric data including history: 9.300.000 parcels, 25.200.000 boundaries, 31.200.000 topographic lines, 5.100.000 symbols and 5.200.000 text labels. The total number of different line segments in the database is over 250.000.000. The administrative (legal) databases contain the following amount of data (without history): 7.500.000 objects (that is, parcels or parts of parcels or apartments), 9.700.000 object addresses, 7.100.000 subjects (that is, persons or organizations 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.

Database Performance by Clustering The advantage of having all data integrated in one database should not be countered by degraded interactive response times. In the Querytool, database performance on the nationwide database is virtually the same as in a database containing only a smaller region, a province or a community. Also integrated, geometric and administrative, views (e.g. showing the prices of the objects color coded on the parcel) are at the same speed as pure geometric views. The same is true for the historic views. The key entries to the database are a region (usually a rectangle, sometimes just one point), parcel number, address and (subject) names. Whenever possible data is organized based on spatial location. This is obvious for the geometric data, but also applied to the administrative data; e.g. by clustering on parcel number, which contains a municipality and section code, or on postal code. This enables spatial range queries to perform well in all situations including the integrated views. The other entries are supported by secondary indices (btree or rtree), because they usually return one or a few results.

3 PRODUCING BILLS FOR THE MAINTENANCE OF THE LARGE SCALE TOPOGRAPHIC MAP

The organization of the maintenance of the large scale topographic map in the Netherlands is on a regional level. In each region there is an local institution, which is responsible for the maintenance of the large scale topographic map. These institutions are founded by several participants, who have a common interest in maintaining such a topographic map. Usually the participants are utility companies, water boards, the Cadastre, etc. This case is applicable to the province of 'Zuid-Holland'. The institution maintains the map per municipality. For each municipality the map is updated on a yearly basis by a selected partner. The partner is paid by the number of mutations in a certain period. The Cadastre participates in the maintenance of the large scale topographic map in several municipalities. The following kinds of mutations are distinguished:
Deletion A deletion of an element;
Semantic Mutation A change in a text string and symbols;
Building A new small building or changes to an existing building with the maximum of 8 coordinate points;
New hard topographic element New visible topographic element with a maximum of 10 coordinate points;
Main building New main building with a maximum of 10 coordinate points;
Big Building Change or a new complex building **'Soft' topographic, concentration** Mutations in so called 'soft' topographic elements with a maximum of 10 coordinates, in case they can be measured in combination with other mutations;
'Soft' topographic, not concentration Mutations in so called 'soft' topographic elements with a maximum of 10 coordinates, in case they can not be measured in combination with other mutations;

In case a topographic element has more than 10 coordinates it counts for more than one element. The Cadastre maintains the large scale topographic map in the LKI system. For each entity, like a topographic line, or text element, two time stamps are maintained. First, we have `tmin` which is the time of creation. Second, we have `tmax` which is the time of deletion. If an entity is current, then `tmax = MAX_INT`. Based on these time stamps it is easy to find the changes in the topographic map in a certain period. Further, we need to know which topographic lines are in which municipality. There are two tables `topographic_line` and `municipality` and their attributes look like [1]:

```
create table topographic_line (
   object_id integer,   -- the unique ID of the line
   line iline(50),      -- the coordinates which define the position
   tmin integer,        -- time of creation
   tmax integer);       -- time of deletion (if current tmax=0)

create table municipality (
   code char(5),        -- The code of the municipality
   pgon long polygon);  -- The coordinates which define the boundary of the municipality.
```

With the following SQL view command [2] we can determine which topographic lines have been deleted in the time frame from June 30th 1998 up to and including June 30th 1999:

```
create view deleted_topographic_line as
select l.object_id, municip=m.code, l.line, month = period(tmax), num_points=numpoints(l.line)
from topographic_line l, municipality m
where inside(first_point(l.line),m.pgon) and
  (l.tmax > lkidate2int('30-06-1998 23:59:00')) and (l.tmax <= lkidate2int('30-06-1999 23:59:00'))
```

If we also define SQL views for the new topographic lines and the current lines we can create maps where all lines that are changed in a certain time frame are highlighted, see Figure 1. The bold lines are the lines that have changed. The following view counts the total number of deleted lines per municipality per month:

```
create view deleted_per_municip_month as
select municip, month, number=count(*)
from deleted_topographic_line
group by municip, month
```

These changes as shown in the maps above are not equal to the mutations as described earlier. Usually, a group of changes is equal to a mutation. All depends on the manner which an operator has changed the topographic map. The map with changes is merely used as a means to find and recognize the mutations. Further a list of changes is produced from the database, which can be check marked, to make sure none of the mutations have been skipped. The maps with changes and the list form the appendix of the bill for the maintenance of the topographic map in a certain municipality.

4 DERIVING A ZIPCODE MAP FROM THE CADASTRAL DATA

A geo-marketing company is interested in a space filling zipcode map of the Netherlands. In this map infrastructure (roads, waterways etc.) should still be recognizable. The former zipcode map of the geo-marketing company was generated by unconstrained region growing around the center locations of mailboxes with the same zipcode. For large scale zipcode maps this process resulted in visually confusing maps.

As a basis for the new map, we use the `parcels` table in the Querytool. Every parcel has a unique id (`object_id`), and a land use code (`culture_code`) from which we can derive whether a parcel belongs to infrastructure or not. For parcels with buildings on them, the zipcode can be derived from the `addresses` table, the address is linked to the `parcels` table via the `parcel_id` key. The topological relationships between parcels are stored in the `boundaries` table. This table contains the boundaries between parcels. Every boundary line appears twice in this table, with negated `object_id` values. The `line_length` of the boundary, and a reference to the parcel to the right of this line is stored in `parcel_id` attribute. The parcel to the left of the boundary can be found by looking at which parcel lies to the right of the reverse of this boundary. In summary we use the following database tables for our algorithm [3]:

[1] For the sake of readability only the relevant attributes are shown.
[2] For the sake of readability the SQL-commands have been simplified.
[3] These tables are derived from the original tables in the Querytool database. Their structure is suitable for the zipcode map derivation.

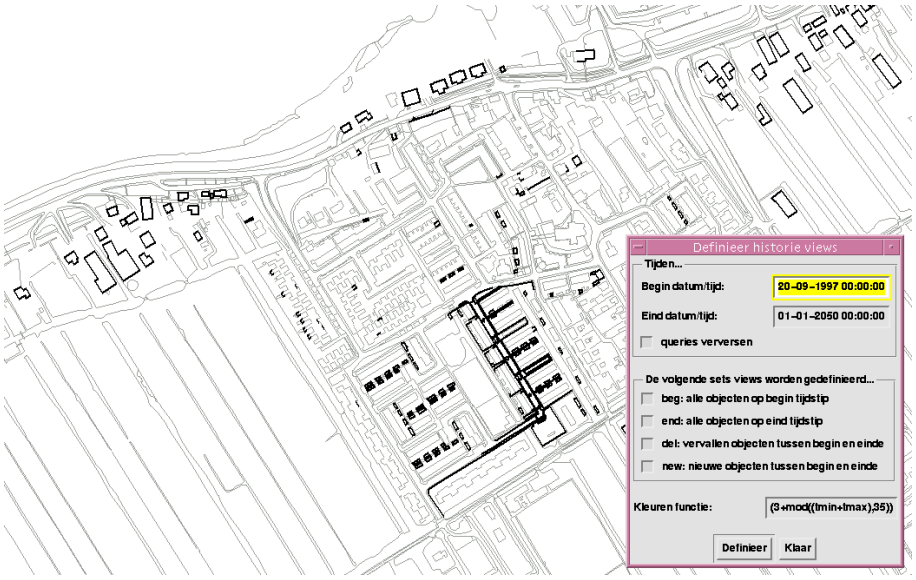

Figure 1: All changed objects

```
create table parcels (
    object_id integer,          -- Unique id of this face.
    culture_code char(2),       -- Code which indicates whether the parcel is infrastructure.
    zipcode varchar(256));      -- Zipcode of parcel (INITALLY EMPTY).

create table addresses (
   address char(100),           -- String which contains the address.
   zipcode char(6),             -- The zipcode of the address.
   parcel_id integer);          -- Reference to the parcel to which this address belongs.

create table boundaries (
   object_id integer,           -- ID of this line. every line appears twice, once
                                -- with oid > 0 and once with oid < 0.
   parcel_right integer,        -- id of parcel to the right of this line.
   line_length float);          -- Length of associated boundary line.
```

On these tables we run an algorithm that fills the `zipcode` field of the `parcels`. This field is initially empty. By first filling in the zipcodes derived from addresses, and then propagating these zipcodes to neighboring parcel across the map, a complete zipcode map will result. Below we give a more detailed description of the algorithm.

Initialize In the initial phase all parcels get an initial zipcode. We distinguish 3 cases:

1. In the database the parcel is marked as infrastructure: In this case the parcel gets the zipcode 'infrastructure'.
2. In the addresses table there is a reference to the parcel: The parcel gets the zipcode of the address. In some cases, a big apartment building for example, one parcel gets more than one zipcode. In this case the different zipcodes of this parcel are appended in the zipcode string of the parcel.
3. All other parcels get the zipcode 'unknown'.

Main Loop Now, we start assigning zipcodes to parcels which have been given the zipcode 'unknown'. This is done by looking at the neighboring parcels. If there is a neighbor which has a 'known' zipcode, this zipcode is propagated to this parcel. If a parcel has more than one neighbor with a zipcode we choose the zipcode of the parcel that has the longest boundary in common with this parcel. In every step of the algorithm we create a list of candidate zipcodes for all parcels with an 'unknown' zipcode that are adjacent to a parcel with a known zipcode. This is done with the following query:

```
create table candidates as
   select   rightboundary.parcel_right as parcel,
            leftparcel.zipcode as candidate_zipcode,
            sum(rightboundary.line_length) as weight
   from     boundaries rightboundary,
            boundaries leftboundary,
```

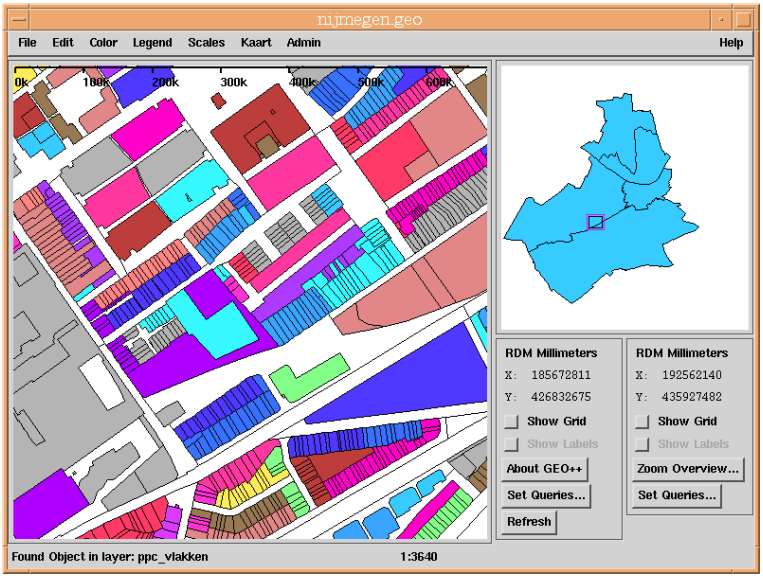

Figure 2: Zipcode map with recognizable infrastructure

```
            parcels rightparcel,
            parcels leftparcel
   where    rightboundary.parcel_id = rightparcel.object_id and
            leftboundary.parcel_id = leftparcel.object_id and
            -(rightboundary.object_id) = leftboundary.object_id and
            rightparcel.zipcode = 'unknown' and
            leftparcel.zipcode != 'unknown' and
            leftparcel.zipcode != 'infrastructure'
   group by rightparcel.object_id,leftparcel.zipcode;
```

For every candidate zipcode we also calculate the length of the boundary which the 'unknown' parcel and the candidate zipcode have in common. From these candidates we choose the one with the longest boundary in common. This is done with the following SQL statements:

```
create table maxes as
   select parcel,max(weight) as maxw
   from candidates
   group by parcel;

create table updatetable as
   select candidates.parcel,candidates.candidate_zipcode
   from candidates,maxes
   where (candidates.weight = maxes.maxw) and (candidates.parcel = maxes.parcel);

update ppc_vlakken
from updatetable
set zipcode = updatetable.candidate_zipcode
where ppc_vlakken.object_id = updatetable.parcel;
```

This main loop is repeated until there is an iteration where the `candidates` table is empty. Now all parcels have been assigned a zipcode. In Figure 2 the resulting map of an urban area is displayed. Different zipcodes have different colors. The requirement that infrastructure should still recognizable has been fulfilled. In this map the parcel boundaries within a zipcode area are still displayed, but these can be easily removed.

5 QUALITY IMPROVEMENT OF THE REGISTRATION LEGAL NOTIFICATIONS

In addition to the registration of the basic rights, such as ownership, related to parcels (cadastral objects), the Cadastre also registers many types of legal notifications. These legal notifications restrict the use of a parcel by the owner due to some reason. An important type of legal notification is related to pipelines usually below the surface; see Figure 3.

In order to protect these pipelines, the parcels crossed by a pipeline get a legal notification of the proper type. This is only done in the administrative part of the Cadastral registration. It has to be done in an official manner described by law: a

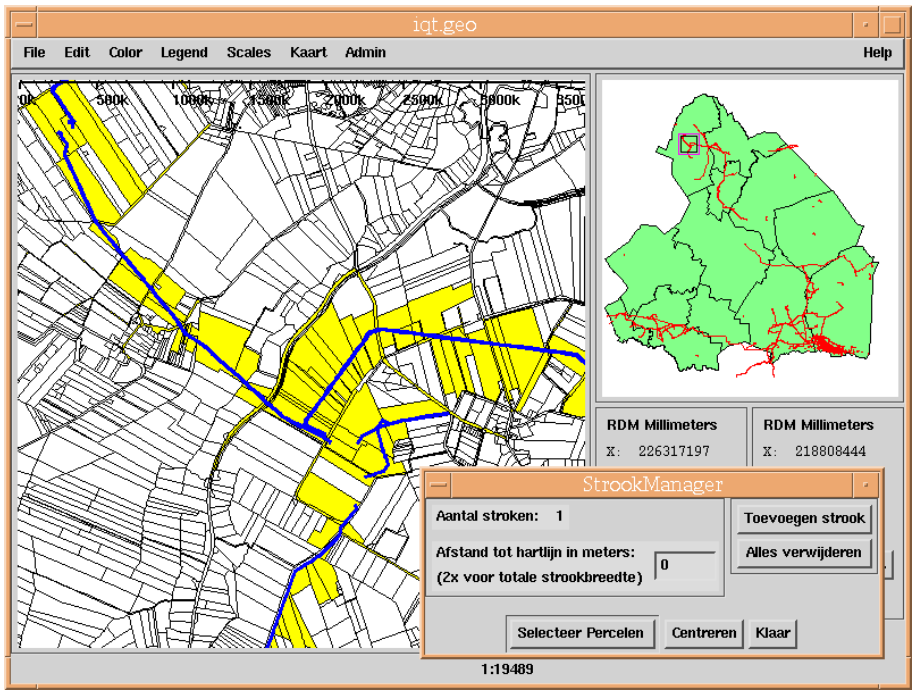

Figure 3: Pipelines of the NAM and the parcels they cross

dead has to be drawn up by the notary and submitted to the Cadastre for registration. The pipelines are not available in the geographic part of the cadastral registration. Several types of problems became more and more visible the last couple of years:

1. Whenever a parcel is split, all new parts inherit the legal notification. This is because the pipelines themselves are not registered at the Cadastre, so it is impossible to determine, which new parts are crossed by the pipeline. In order to be safe all new parts inherit the legal notification. This means that too many parcels have these legal notifications, which implies unnecessary costs for the owner of the pipeline.

2. It is very easy to forget a few parcels when trying to register the complete trace of a pipeline without the exact geometry of the parcels and the pipeline. This results in parcels without a legal notification. This is a dangerous (legal) situation as the pipeline crosses these parcels, but without the proper status.

3. The registration of basic rights always stores who (which subject) has a certain type of right on which parcel (object). In the early registration of legal notifications it was not registered who caused the specified type of legal notification. Only the fact that there were one or more (types of) legal notifications was associated with the parcel. This makes the maintenance of this registration very difficult. Imagine that for some reason a pipeline does not need the legal protection anymore, then it is dangerous to remove all the legal notifications because they are 'anonymous'. It could very well be the case that another utility company has a pipeline crossing the same parcels.

In order to solve the problems mentioned above it was decided to start a quality improvement process. Going back to all the paper deeds is just too much work, so the Querytool was used to select the parcels, which have these 'anonymous' legal notifications (these are of types BP, BG or OG). Using the list of selected parcels the paper deeds are now retrieved and the legal notification is associated with the proper organization ('owner' of the pipeline) and also the type of legal notification is changed to OL or BZ. This solves the third problem mentioned above. However, it does not solve the first two problems.

A pilot project was started with an important owner of pipelines in the Netherlands; the NAM, Nederlandse Aardolie Maatschappij, a company equally owned by Shell and Esso. The NAM delivered a digital version of their pipelines to the Cadastre, which were then entered into the Querytool database and confronted with the parcels; see Figure 3. This was not a simple query in the database, because the geographic data model of the Cadastre is based on topology (van Oosterom, 1997). Within a relational database an overlap or cross operation based on parcels modelled with topology is impossible. Therefor this operation was implemented in the interface (front-end) part of the Querytool; see the inset window of Figure 3.

After the quality improvement of the legal notifications, the parcels with a legal notification of OL or BZ associated with the NAM can be displayed on top of the parcels crossed by a pipeline of the NAM. A few things can then be observed.

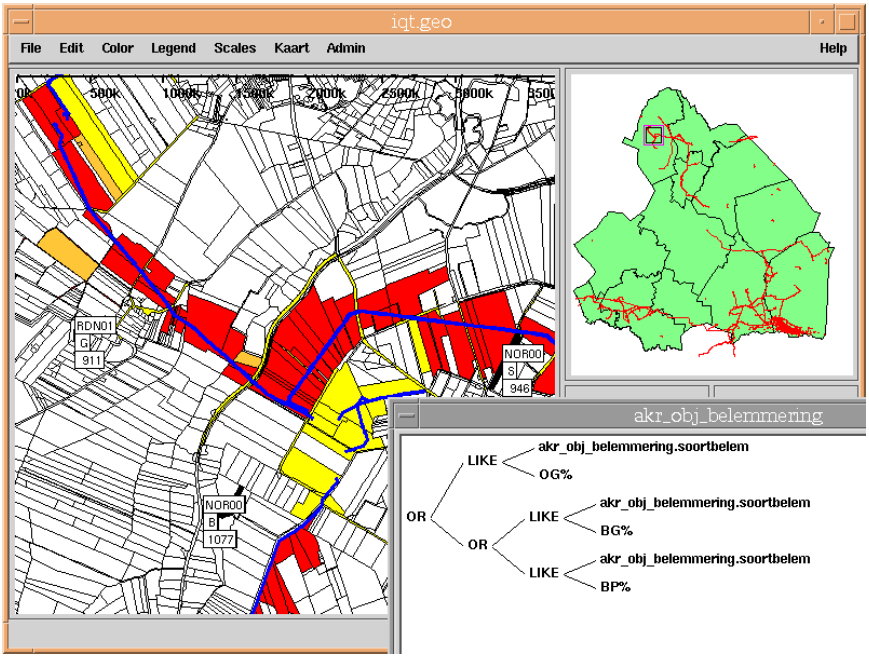

Figure 4: Parcels with legal notification of type BP, BG or OG are marked with a label

First, not all parcels crossed by a pipeline have the legal notification. This can be correct in case the parcel owned by the government; e.g. roads, in this situation a 'permit' is sufficient, but this is not registered at the Cadastre. However, there are several parcels without a legal notification and clearly not road parcels are crossed by a pipeline. This could be an old pipeline and has to be checked by the NAM. *Second*, there are parcels with a NAM legal notification which are not crossed by a pipeline. Again, this has to be checked by the NAM. It could be correct; e.g. the parcel might contain a NAM access road or some type of NAM location.

Finally, it is interesting to check if all 'anonymous' legal notifications of type BP, BG or OG are resolved by the quality improvement process. Therefore all legal notifications of these types are selected and displayed with a label, indicating the parcel number, in Figure 4. In the project of quality improvement of legal notifications and the pilot project with the NAM, the Querytool turned out to be a very useful tool. As described, the Querytool is used during several stages: before the process to inspect the situation and select the 'problem' parcels. During quality improvement to find the parcels crossed by pipelines, after quality improvement to check if indeed the results is correct; e.g. there are no more anonymous legal notifications. One final remark: there are many more types of legal notifications than the ones mentioned in this section. These were also quality improved, but not discussed in this section.

6 CONCLUSIONS

Besides the three example applications and many small ad-hoc queries, the Querytool has been used for several other projects. 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 reorganizing), 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. owner and so own), 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 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 in the way described above) and the cadastral map, aggregate thematic information (e.g. average price) and visualize the result on a geometric aggregation of the same level (e.g. municipality), and so on.

The Querytool has proven to be a very useful tool within the Cadastral organization. Further enhancements will improve the usefulness of the Querytool even more. Of course the users also have new functional wishes after using the system. The first wish, is specifying his/her own views, based on joins, and having attributes from multiple tables in the result. The second wish is more import and export functionality. The third wish is a more up to data Querytool 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 our geometric and administrative 'production' databases. Instead of using full data set copies, this can be done more efficiently by only transferring the changes. These mutation files are standard products in the source systems and can be obtained every month.

Further, external users should be able to query the data over the internet. Limited functionality of the geographic Querytool is implemented in Java (Arnold and Gosling, 1996) as a prototype for the digital Geoshop (van den Berg et al., 1997). This must be coupled to the NCGI (National Clearinghouse Geo-information) (de Gunst and van Oosterom, 1997, Jacobi and Lind, 1997, van de Kieft and Kok, 1997) and based on OpenGIS standards (Buehler and McKee, 1998), specifically the standards currently being developed in the web-mapping testbed.

ACKNOWLEDGMENTS

We would like to thank the persons involved in the implementation of the Querytool. In total 30 to 40 persons have been involved with this project. These are just too many to mention. We will make a few exceptions. First, Herman Welleweerd, the project leader, who had the difficult task to implement and introduce a new system in a complex organization. Second, Bertus Padberg, the DBA who was responsible for loading the huge data sets in the database. Finally, Joep Mathijssen, who developed large parts of the cadastral add-on's.

REFERENCES

Arnold, K. and Gosling, J., 1996. The Java Programming Language. Addison-Wesley Publishing Comapny, Inc., Reading, Massachusetts.

ASK-OpenIngres, 1994. INGRES/Object Magemenent Extention User's Guide, Release 6.5. Technical report.

Buehler, K. and McKee, L., 1998. The opengis guide – introduction to interoperable geoprocessing. Technical Report Third edition, The Open GIS Consortium, Inc.

de Gunst, M. and van Oosterom, P., 1997. Network computers at the dutch cadastre. In: 46th Photogrammetric Week, Stuttgart, September 22-26, 1997.

IJsselstein, J. A. and Kap, A. P., 1995. Het kadastraal perceel: een stevig fundament! Nederlands Geodetisch Tijdschrift Geodesia 37(7/8), pp. 343–349. (In Dutch).

Jacobi, O. and Lind, M., 1997. Metadata: From european standard to user service. In: Third Joint European Conference & Exhibition on Geographical Information (JEC-GI'97), Vol. 2, pp. 1155–1164.

Lemmen, C. H. and van Oosterom, P. J., 1995. Efficient and automatic production of periodic updates of cadastral maps. In: JEC'95, Joint European Conference and Exhibition on Geographical Information, The Hague, The Netherlands, pp. 137–142.

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.

Professional Geo Systems (PGS), 1996. The GEO++ system, version 2.80, Reference manual. Technical report.

van de Kieft, I. A. and Kok, B., 1997. The development of a geo metadata service for the netherlands. In: Third Joint European Conference & Exhibition on Geographical Information (JEC-GI'97), pp. 1165–1176.

van den Berg, C., Tuijnman, F., Vijlbrief, T., Meijer, C., Uitermark, H. and van Oosterom, P., 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, Kluwer Academic Publishers, Boston, pp. 365–378.

van Oosterom, P., 1997. Maintaining consistent topology including historical data in a large spatial database. In: Auto-Carto 13, pp. 327–336.

van Oosterom, P., Maessen, B. and Quak, W., 2000. Spatial, thematic, and temporal views. In: Proceedings of SDH2000. (Submitted).

Vijlbrief, T. and van Oosterom, P., 1992. The GEO++ system: An extensible GIS. In: 5th SDH, pp. 40–50.