MULTI-SERVER INTERNET GIS: STANDARDIZATION AND PRACTICAL EXPERIENCES

Carel van den Berg, Frank Tuijnman, Tom Vijlbrief
Professional GEO Systems, Damrak 44, 1012 LK Amsterdam, The Netherlands
email {frank|tom}@pgs.nl, http://www.pgs.nl
and
Co Meijer, Peter van Oosterom and Harry Uitermark
Cadastre, P.O.Box 9046, 7300 GH Apeldoorn, The Netherlands
email {meijer|oosterom|uitermark}@kadaster.nl, http://ooa.kadaster.nl

1. INTRODUCTION

An approach to an open infrastructure for geographic information on the Internet is presented in this chapter. This infrastructure enables data providers to publish their data independently, while enabling end-users to access data from several providers simultaneously, and integrate the data locally in a geographic browser. Our goal is that an end-user finds accessing geographic information in this environment as easy as if he would be working with a state-of-the-art GIS package with all data that he is interested in on his own computer. The key elements that are required are: a common format for publishing meta-data on each server, a common SQL derived query protocol, standard transfer file formats, and standard certificate based authentication procedures, for access control and (optionally) billing. An experiment with this approach has been carried out, with three data providers in The Netherlands: The Dutch Cadastre, the municipality of Almere, and the cable-TV company Casema. This chapter presents the major design decisions, the choices for the prototype environment, and the relationship to ongoing specification and standardization processes for geographic data. In particular the relationships with the proposed European CEN standards, and the recently accepted specifications from the OpenGIS consortium are described.

The current wave of GIS software for Internet is based either on the file downloading paradigm, or on the picture paradigm (presenting a map as a JPEG picture), or on the client-server paradigm (creating a closed interaction between a client and a single server). Neither of these approaches can capitalize on the main potential of the Internet: integrated and easy access to a vast amount of geographic information on various servers. In addition to that the interaction protocol between client and server is typically proprietary, which means that someone who browses geographic information needs software from the same vendor as is used by the publisher.

To make GIS popular on the Internet one needs to create for geographic information the same level of uniformity as the World Wide Web has done for text. The brilliance of the World Wide Web lies in the combination of the hypertext model with the Internet, together with a formatting standard for text (HTML). The hypertext model, however, does not work for geographic data: it is not particularly useful to jump from one map to the next. So another basic metaphor has to be used.

The standard model for geographic data on the computer is the layer model. The layer model of geographic information systems relates to a paper map like the hypertext model relates to text on paper. To make geographic data on the Internet attractive one has to set a standard for the layer model, so that we can obtain a topographic layer from one source, a pollution layer from another, and a property layer from a third source, and dynamically merge them in a geo-web browser. To achieve this the following standard protocols have to be defined (in addition to support for authentication and billing): 1. a method to inquire which layers are available, and what data they contain; 2. a method to ask a particular layer from a particular source; and 3. a standard format for returning this information. Many aspects of these protocols are subject to ongoing standardization efforts. The format for returning geographic information is essentially a description of a transfer file format.

The method for querying the meta-data has a clear relationship with the meta-data standards (CEN-META 1996), and Clearinghouse related activities (Absil van de Kieft and Kok 1997). The protocol for querying geographic information is new. The CEN has acknowledged the need for something like this; see (CEN-QUERY 1997), which specifies names and semantics of required spatial operators. The closest relevant specification for the query protocol is the OGC specification for SQL with simple geometric features. Despite the fact that it is relatively easy to identify protocols that can be used to address part of the problem, no comprehensive proposal exists so far that can be used to achieve open access to geographic data publishers on the Internet.

2. ARCHITECTURE

Our approach to the design of the meta-data structure, the query formalism and the format for the returned data is based on the object-relational formalism, where we include geometric data types as attributes within a relational table. The motivation for this approach is based on the following considerations:

1) object relational database management systems with support for geographic data are now available from most major vendors (Informix, CA-OpenIngres, Oracle). Even if one wants to provide access to a file based collection of geographic data it is not difficult to implement a limited selective capability on top of it, though of course performing such selections will cost more time. So it is technically feasible for any organization to implement this functionality.
2) the object relational model is currently the only widely available formalism that can deal with geographic data, and in which all three required elements (a meta-data structure, a query formalism, and a format for returned data) are defined in an integrated manner. This is essential from a technical point of view: the meta-data does not just describes the data, it also has to provide the 'words' that can be used in queries, and it has to be clear which words can replace which syntactic elements in a query. The returned data has to be understood as a response to the query, so there has to be a well defined relationship between the semantics of queries, and the actual data that is returned.
3) It can be mapped easily to the stateless http protocol, because SQL is also stateless (meaning that any request can be handled independently from previous or subsequent requests).
4) It ensures that the browser requires only knowledge about which data is available, rather than detailed knowledge about file naming conventions, and tiles.

In mapping extended SQL to http, in such a way that it can be used effectively for geographic data publication over the Internet, a number of issues have to be tackled:

1) The geometric types and their representation.
2) The meta-data structure.
3) The query and the way it is encoded in a URL.
4) The format of the returned data and data compression (efficient data transfer over the Internet).
5) Authentication, and billing mechanisms to support commercial geographic data publication.

2.1 Geometric types

A common understanding of geometric types is essential for exchange of geographic information. All file transfer standards that currently exist define their own geometric constructs, and so do the proprietary data formats. Examples are the new CEN standards (CEN-SCHEMA 1996) and (CEN-TRANSF 1997), the OpenGIS Simple Features specification (Buehler and McKee 1996), and national standards, such as NEN1878 (1993) . The basic types that we choose to support are point, polyline, and polygon. A point is represented by its coordinates, a polyline and a polygon as a list of points. In the current system polylines and polygons consists of straight line segments only. The NEN 1878 standard, commonly used in The Netherlands for the exchange of geographic data, allows polylines and polygons that are defined as a sequence of segments, where each segment can have its own interpolation method. Due to the object-relational formalism in which these types are embedded, it is relatively straightforward to include topological information.

A practical problem that anyone implementing a system like this faces is whether the client software should be sufficiently intelligent to be able to handle also the most complex structures, or whether the data publisher simplifies his dataset, in order to enable more simple client software to use this data. We believe that the only practical approach here is that the data publisher (typically a large organization) takes the trouble to offer his information in a number of ways, targeted to both advanced software systems, and simple ones. For example, the data publisher has to make a decision whether or not the client software has to understand topology. Parcels can be represented as a table of lines with references to left and right polygons, and a chain topology. For which in The Netherlands NPR3611 (1997) is a commonly used standard. Parcels can also be presented by the data publisher as a table with polygons with holes. In the first case the client software needs to understand the meaning of the attributes that encode the topological structure, or simply view a line image. In the second case the client software can pass the polygons almost directly to a graphics library for drawing.

2.2 The meta-data

In our view the primary role of a data-publisher is to enable easy access to the information, and not to define a particular application. In particular visualization characteristics, and the structure of the legend, should be customizable (in our implementation: in the HTML page from which the applet is started), without having to change the server. In this way a single data server can support a wide range of applications, and applications may use several servers. On the other hand, someone without a customized browser should be able to view the data in an acceptable manner. For the meta-data that we use here we distinguish three levels:

1) The definition of the structure of the geographic data;
2) The definition of the visualization characteristics of a layer; and
3) The aggregation of a number of layers to a single layer, or map, from the user perspective.

The first level describes which tables are available, which attributes are defined and what their types are, the scale for which the information is intended, which projection has been used and which attributes can be used in the where clause of a query. It gives the minimal information that is necessary for someone to query the data, and to display it. The second level determines the use of colors, line styles, fill patterns and symbols. It is useful to have default values at the server-side, for those situations where someone just wants to take a look. In most practical cases, however, choice of symbols and colors will have to be defined at the client side, by the client application. One reason is that this makes it possible to define different visualization characteristics for different applications. The second reason is that applications that use data from several servers will almost always have to define their own visualization characteristics, to avoid conflicting or confusing use of colors, symbols and line styles. The third level, defining maps as composite layers, is also something that is required both at the server, and at the client side. At the server side a good hierarchical structure of the layers is needed to support browsing. At the client side one can define maps that are composed of layers from several sources, and that are specific to a particular application. It is worth noting that the meta-data that we require here is far less than is typically included in a geographic meta-data catalog (CEN-META 1996). The information included here is the minimal information necessary to be able to retrieve and visualize information.

2.3 The query model

The query model is essentially based on spatial SQL, without joins. The reason to exclude joins is that this makes it relatively easy for the server administrator to control the load on his database. If (spatial) joins are required, one can circumvent this limitation by including a view in the server database that constructs the desired join. So the client designer has the freedom to select a subset from a relation, and the designer of the server has to decide (by defining suitable views), which joins on his tables are allowed. We believe that this constitutes a fair compromise between flexibility at the client side, and the need to guarantee stable operation at the server side. To use the http protocol the queries have to be encoded in a URL. Below the syntax for that encoding is given (capitals indicate non-terminal symbols for which we have not included the remaining production rules):

<Query>:: <prefix> ? <query>
<prefix>:: http://<sitename>/<directory>/<program>
<query> :: coordsys=<COORDSYS> & database = <DATABASENAME> & format=<FILEFORMAT> &
& relation= <RELATIONNAME> & attributes=<attributelist> & where=<whereclause>
<whereclause> :: <spatialclause> | <spatialclause> & <ANDOR> & <conditionlist>
<spatialclause> :: WRectangle.intersects(<x1>,<y1>,<x2>,y2>)
<conditionlist> :: <condition> | <condition> & <ANDOR> & <conditionlist>
<condition> :: <ATTRIBUTENAME> <OPERATOR> <VALUE>

The example below shows a query which requests four named attributes (magma_oid, geo_bbox, geo_pgn, owners), for all parcels in ‘percelen’ relation in the ‘kad4’ database, within the selected region, where the owners names prefix is between ‘oost’ and ‘oostf’. The coordinates are given in RDM (plane state coordinates in The Netherlands). As a result a list of tuples will be returned, satisfying the where clause.

//ooa.kadaster.nl/cgi-bin/magma?
coordsys=rdm&format=javabin&database=kad4&relation=percelen&
attributes={magma_oid,geo_bbox,geo_pgn,owners}&
where=WRectangle.intersects(189000,485000,192000,488000)&and&
owners>=’oost’&and&owners<=’oostf’

2.4 The returned data format and data compression

The returned data format consists of a list of tuples, in a single file. Compared to other popular file formats, such as the shape file format by ESRI and the MapInfo exchange format, the main difference is that all information is in just one file: geometric data as well as the non-geometric attributes. An important aspect of the returned data format, not required by most standards, is that each object should have an object-identifier, or some other unique key. This is important to enable the browser to determine whether he has already received an object or not. Receiving objects twice can easily occur after panning, when objects that overlap the previous viewing region may be transferred again. The other reason is that we believe that each object on the map should be traceable to the supplier, and that it should be possible to make inquiries about the object. Geographic data area often based on 8 byte double precision floating point numbers. Due to the spatial cohesion a substantial reduction in size can be achieved. In large scale mapping, differences between points within a polyline can almost always be encoded with 16 bits, and usually with less. A differential encoding scheme has been used, which achieves good compression, with relatively little computing overhead.

2.5 Commercial services

Commercial services for geographic data require a mechanism for either authentication, or one for direct payment. Authentication is feasible with current Internet technology. The most popular method is still the use of passwords and login names. For a system where one of the main attractions is the ability to access several providers simultaneously, it is cumbersome to have to use (and remember, and administrate) login names and passwords. The use of certificates is essential for three reasons:

1) The data publisher can be confident that data is transferred to the proper users;
2) The user doesn't have to type passwords each time that he obtains information from another server;
3) The user can trust that the information originates indeed from the source, and not from some impostor.

Though technically feasible, a major commercial problem is that the pricing models that are currently used for geographic data are not targeted to the infrequent and non-professional user. And it is not easy to see how the price structure should be changed. In The Netherlands, for example, a utility company will pay an annual fee to obtain updates of the large scale maps of the area in which he has cables. If the same map would be offered to infrequent users on Internet for a low fee per query, the utility companies would be tempted to stop their current annual subscription, and request the information on Internet instead.

3 RELATIONSHIP TO STANDARDIZATION ACTIVITIES

The architecture described here can be used to support various standardization approaches. We discuss here two of them: the CEN standards, and the OpenGIS consortium standards. The OpenGIS consortium has defined semantics for geometric types, and (among others) bindings to Extended SQL and to CORBA. To place the Lava/Magma architecture in the Extended/SQL framework all that has to be done is to define the format of the URL that encodes the query string, and to include a description of a compression mechanism (though this step could be omitted). The advantage of extending the current standards in this manner is that an easy to use Internet interface would be available to all geographic databases that conform to the open GIS standard. A second advantage is that it is possible to write simple interface software mapping the standard OpenGIS model, to the model that is actually implemented in the database. A third advantage is (provided that only an http mapping for the extended SQL is defined) that a uniform Internet interface can be created to geographic databases that implement the OpenGIS SQL interface, and those that implement the OpenGIS Extended SQL interface.

The main weaknesses of revision 0 of the OpenGIS specification for Corba are that the procedure to obtain the meta-data about a database is inadequately specified, and the absence of a compression mechanism. In addition, we believe that the Corba interface is unnecessarily complex for simple retrieval applications.

The new (CEN-SCHEMA 1996) standard proposals describe schema's in Express, an object-oriented specification language, for various types of geographic data sets, with a complete topology on one extreme, and 'spaghetti' on the other. Mapping these to the architecture described here requires a mapping from the object-oriented Express language, to the object-relational paradigm. For a non-topologic data set this is straight-forward: the express types describing the geometry become types in the object-relational paradigm. For topologic datasets the relationships that are expressed directly in the Express formalism, would have to be mapped to references through foreign keys. This can be done in a number of ways, and therefore would have to be explicitly standardized. For the returned data format the (CEN-TRANSF 1997) standard can be used. The query syntax and semantics would have to be described from scratch.

4 SYSTEM ARCHITECTURE

The Lava/Magma system, developed by PGS, implements the architecture with two main components: Lava, a Java based map viewer and presentation software, and Magma, a middleware program that interfaces with an http server on one side, and a geospatial database on the other side. Figure 1 depicts this architecture. The systems supports both raster and vector data. Raster images, such as aerial photographs, are stored as tiles in the database. Upon a request from Lava a single JPEG image is constructed without tiles, and with the right number of pixels for the display. Note that this requires that Lava requests the image for the actual screen size of the image. Currently, the images are not cached, so that each zoom or pan operation results in a request to the server.

[Fig. 1 here]

Vector data are cached by the Lava map viewer. The cache administration maintains a representation of the areas that have already been viewed. So once a particular area has been viewed, a second viewing (after panning or zooming), requires no request to the server.

5 TRIAL EXPERIENCES

A data integration trial with the PGS products Lava and Magma has been carried out in the province of Flevoland, where three organizations, the municipality of Almere, the cable-TV company Casema, and the Cadastre, have implemented a system that allows them to access the data of the other parties directly. Three servers are used, one operated by the Casema, one by the Cadastre, and one by PGS; see Figure 2. The data published by the Cadastre consists of the parcel boundaries. The data from the municipality consists of a large scale topographic map and large scale topographic plans for new neighborhoods. Almere builds about 3000 new houses annually. The Casema has published the cable locations, both planned and existing. For browsing, the Java based Lava GIS browser is used, and for the server the Magma GeoData publisher is used to interface between the http requests and various geographic datastores: CA-OpenIngres 2.0 (1995) for the Cadastre, Illustra for the topographic data, and flat DXF files for the Casema. The three partners have installed their own servers, connected to the internet, providing spatial data on request.

[Fig 2. here]

In this trial the feasibility of dynamic data integration over the Internet has been demonstrated, supporting both raster and vector data at the client side, and using a Java based GIS browser to give everyone direct access. The user interface offered by this system is comparable to a somewhat simplified version of GEO++ (Vijlbrief and van Oosterom 1992). Figure 3 shows the Lava user interface as it appears when invoked from http://ooa.kadaster.nl/. At the bottom of the screen a message line (left) and the X/Y position of the mouse (right) are displayed. The overview map provides orientation for the user, and allows selecting an area for viewing. From the icon bar above the main map zoom, pan, and object selection operations can be activated. Object selection results standard in a window that shows all the attributes, which are requested from the server that has that object; see Figure 4.

[Fig 3. here]
[Fig 4. here]

An important part of the user interface is the legend, to the right of the main map panel. The legend indicates the layers that constitute the map, shows their visualization, and shows an icon to indicate the server from which the layer was obtained. The user can add layers, delete layers, or temporary switch off visualization of a layer. The ordering of layers can be changed with up and down arrows. After choosing the ‘Add’ button the screen shown in Figure 5 pops up. In this example a user has selected from the server ‘magma.pgs.nl’, the database ‘Almere’, and the layer ‘Gras’. When combining information from several servers, the user has to be able to configure locally colors, line styles, and symbols, to avoid conflicting usage. This is possible by pushing the ‘Edit’ button, which brings up a screen like in Figure 6 (for text labels), or Figure 7 (for line objects). This screen also enables a user to enter additional selection criteria, such as “source=’GEOGR’ and quality= ‘T1’”. This selects a subset from the parcel boundaries, which are shown as bold black lines in Figure 8.

[Fig. 5 here]
[Fig. 6 here]

[Fig. 7 here]

The trial has demonstrated that it is possible to work interactively with these maps. A typical picture, like the one in Figure 8, requires about 100Kb of vector data. With an ordinary telephone modem this takes only 40 seconds, and with an ISDN line about 15 seconds. This is already acceptable. In the near future this will be improved upon substantially because the Dutch utility companies are committed to offer cable modem Internet access to 70% of the country by the year 2000. By then the performance of the server will become more crucial than the transmission delay.

[Fig. 8 here]

Two non information technology aspects contributed to the success of this trial. The first was that all geographic data had a high positional accuracy, and was surveyed with respect to the same reference points (corners of buildings, etc.). As a result, the fences from the topographic map nicely overlap with the cadastral boundaries, and the relative distance between cables (on the Casema layer), and houses (on the topographic layer) can be trusted. So even though these organizations make their own maps, they already collaborate to ensure that these maps can be combined. If this is not the case, a dynamically combined map becomes far more difficult to interpret, and may easily mislead a casual user.

A second aspect that contributed to the success of the pilot was that there was no overlap in 'themes'. This was avoided by using from the Cadastral map only the property boundaries, and not the buildings, that are present on the Cadastral map on paper. If the houses from the Cadastral maps would have been included as well, they would have overlapped mostly with the buildings on the topographic map, but not completely, due to different classifications, and different criteria for deciding whether a house or a building should be on the map. This is something that does not represent a problem for the professional user, but would be confusing for the casual users, who would be confronted with subtly different meanings attributed to names or concepts that he would consider synonyms. If casual users are to make effective use of this type of maps the meaning of these maps should be well standardized at a national level. In The Netherlands a new national standard has been proposed NEN3610 (1995) for modeling terrain information, that will be used to address this problem.

6 CONCLUSION

To support open access to geographic data over the Internet three related protocols need to be defined: 1. a method to inquire which geographic data is available from a particular publisher; 2. a method to ask a particular layer from a particular publisher; and 3. a standard format for returning this information. The solution proposed in this article is based on the object relational model with geometric types. Simple SQL requests are encoded as URL's for http, and the result is returned as a list of tuples, with geometric attributes.

The trial, with the Lava/Magma software from PGS, has demonstrated that it is effectively possible to implement this approach, and to achieve in this manner open, easy and integrated access to data from different organizations. As such it can be the basis for a public infrastructure that allows each organization to publish her data independently, while at the same time it enables clients, both professionals and citizens, to have integrated access through the Internet to all available geographic information.

An effective usage of such an infrastructure by infrequent users requires that the maps are made with a well defined (and sufficiently high) accuracy, and that the meaning of the maps, relative to the actual situation in the terrain, is standardized as much as possible.

References

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

Buehler, K. and L. McKee (1996). The OpenGIS guide - introduction to interoperable geoprocessing. Technical Report 96-001, The Open GIS Consortium, Inc. (OpenGIS® Abstract Specification, http://www.opengis.org/).

CEN-META (1996). CEN/TC 287. Geographic information - data description - metadata. Technical Report draft prEN 12657, CEN, Brussels.

CEN-QUERY (1997). CEN/TC 287. Geographic information - processing - query and update. Technical Report draft prEN 12660, CEN, Brussels.

CEN-SCHEMA (1996). CEN/TC 287. Geographic information - data description - spatial schema. Technical Report final draft prENV 12160, CEN, Brussels.

CEN-TRANSF (1997). CEN/TC 287. Geographic information - data description - transfer. Technical Report draft prEN 12658, CEN, Brussels.

INGRES (1995). CA-OpenIngres. Object Management Extension User's Guide, release 1.1.

NEN1878 (1993). Exchange format for data about spatial objects that are related to the earth surface (in Dutch). Nederlands Normalisatie-instituut.

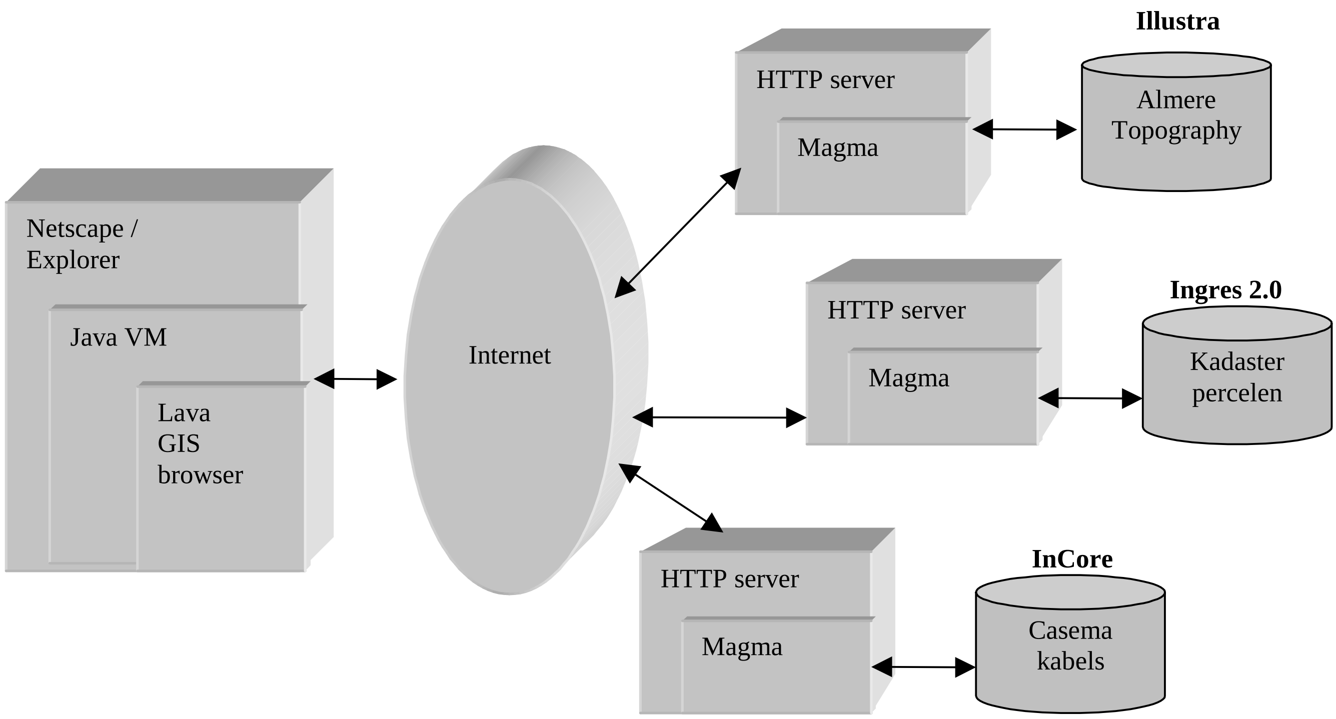
NEN3610 (1995). Terreinmodel vastgoed. Terms, definitions and general rules for the classification and coding of spatial objects related to the earth surface (in Dutch). Nederlands Normalisatie-instituut.

NPR3611 (1997). Practical guidance for the of NEN 1878 en NEN 3610 (in Dutch). Bijlage B - Ketting-topologie. Nederlands Normalisatie-instituut.

Vijlbrief, T. and P. van Oosterom (1992). The GEO++ system: An extensible GIS. In Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, South Carolina, pages 40-50.

Captions

Figure 1: Overview of the GeoShop Magma/Lava architecture
Figure 2: The GeoShop data providers (PGS on behalf of the Almere)
Figure 3: The standard GeoShop interface as started from the Kadaster
Figure 4: Getting more information of a certain object
Figure 5: Adding a layer ‘Gras’ from the ‘Almere’ database
Figure 6: Adjusting visualization properties of parcel labels (for parcels larger than 300 m2)
Figure 7: Adjusting visualization properties of boundaries (satisfying specified where-clause)
Figure 8: Selected parcel boundaries (source=’GEOGR’ and quality=’T1’) in bold black lines


Netscape /
Explorer
Java VM
Lava
GIS
browser
Internet
HTTP server
Magma
Illustra
Almere
Topography
HTTP server
Magma
Ingres 2.0
Kadaster
percelen
HTTP server
Magma
InCore
Casema
kabels

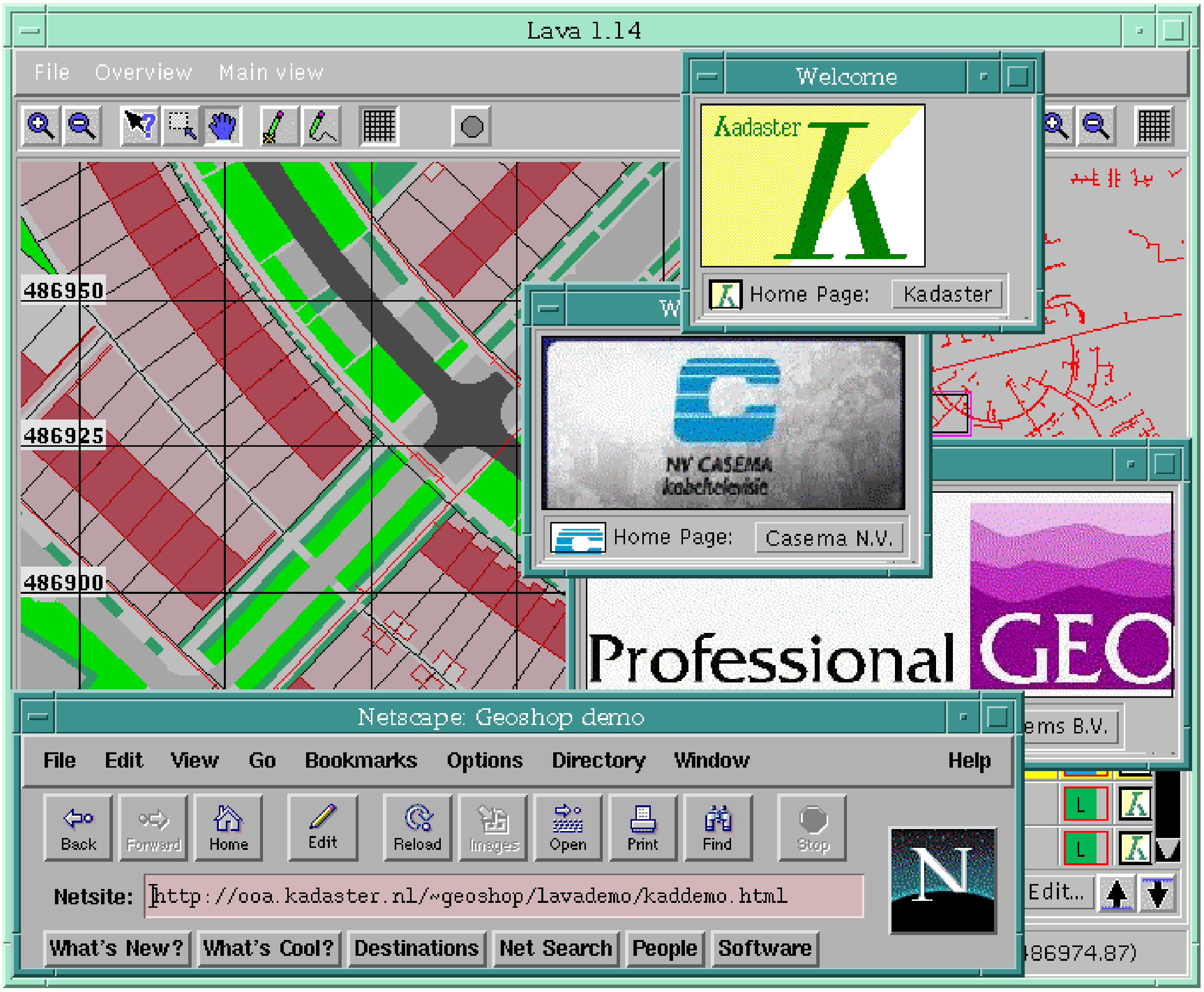


Figure 2: The GeoShop data providers (PGS on behalf of Almere)

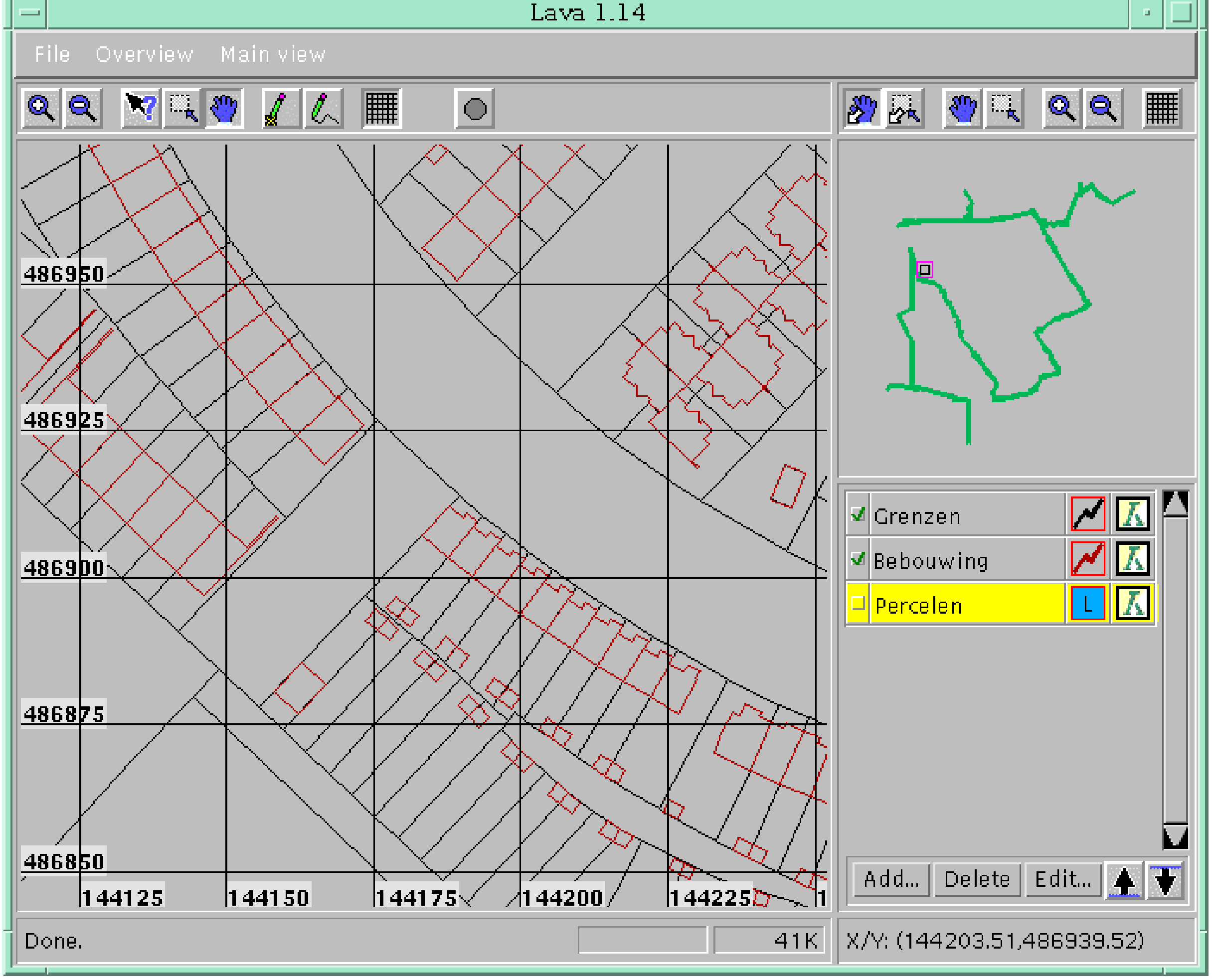


Figure 3: The standard GeoShop interface as started from the Kadaster

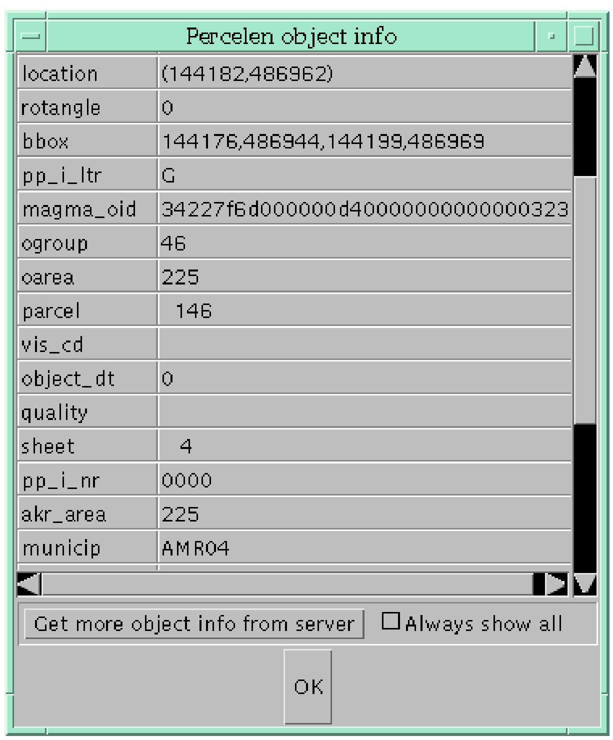


Figure 4: Getting more information of a certain object

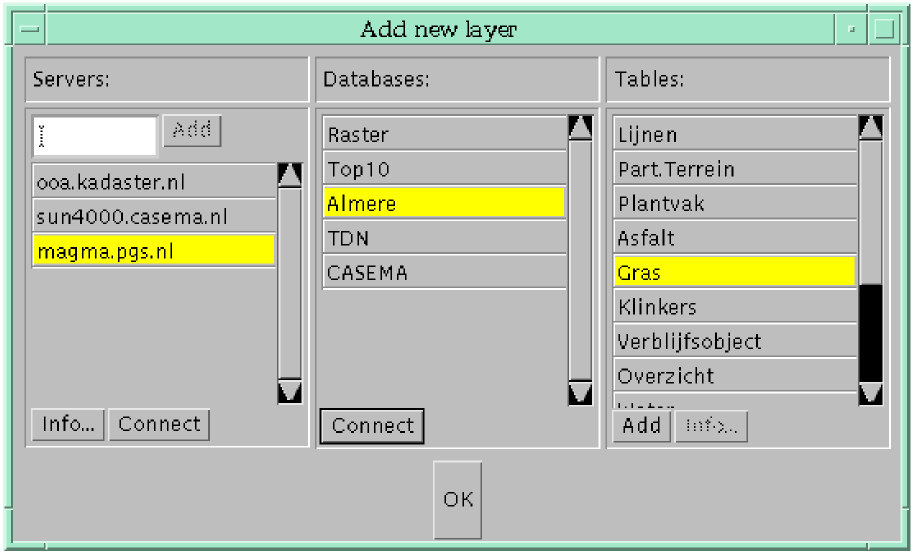


Figure 5: Adding a layer 'Gras' from the 'Almere' database

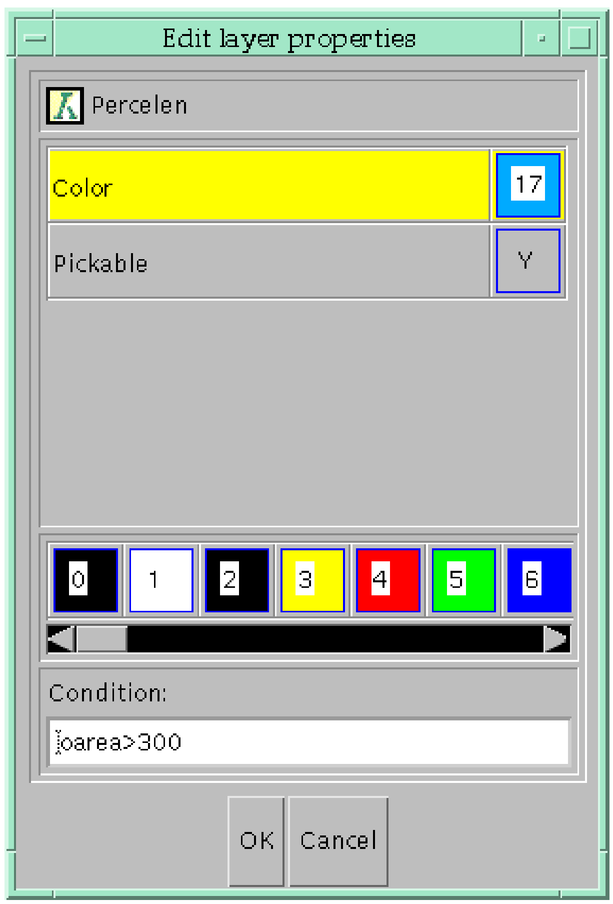


Figure 6: Adjusting visualization properties of parcel labels (for parcels larger than 300 m2)

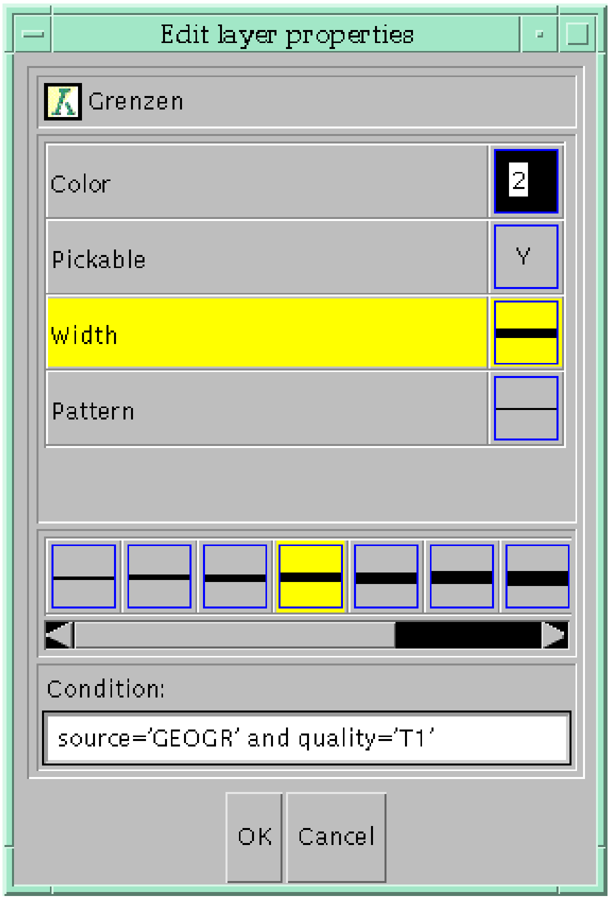


Figure 7: Adjusting visualization properties of boundaries (satisfying specified where-clause)

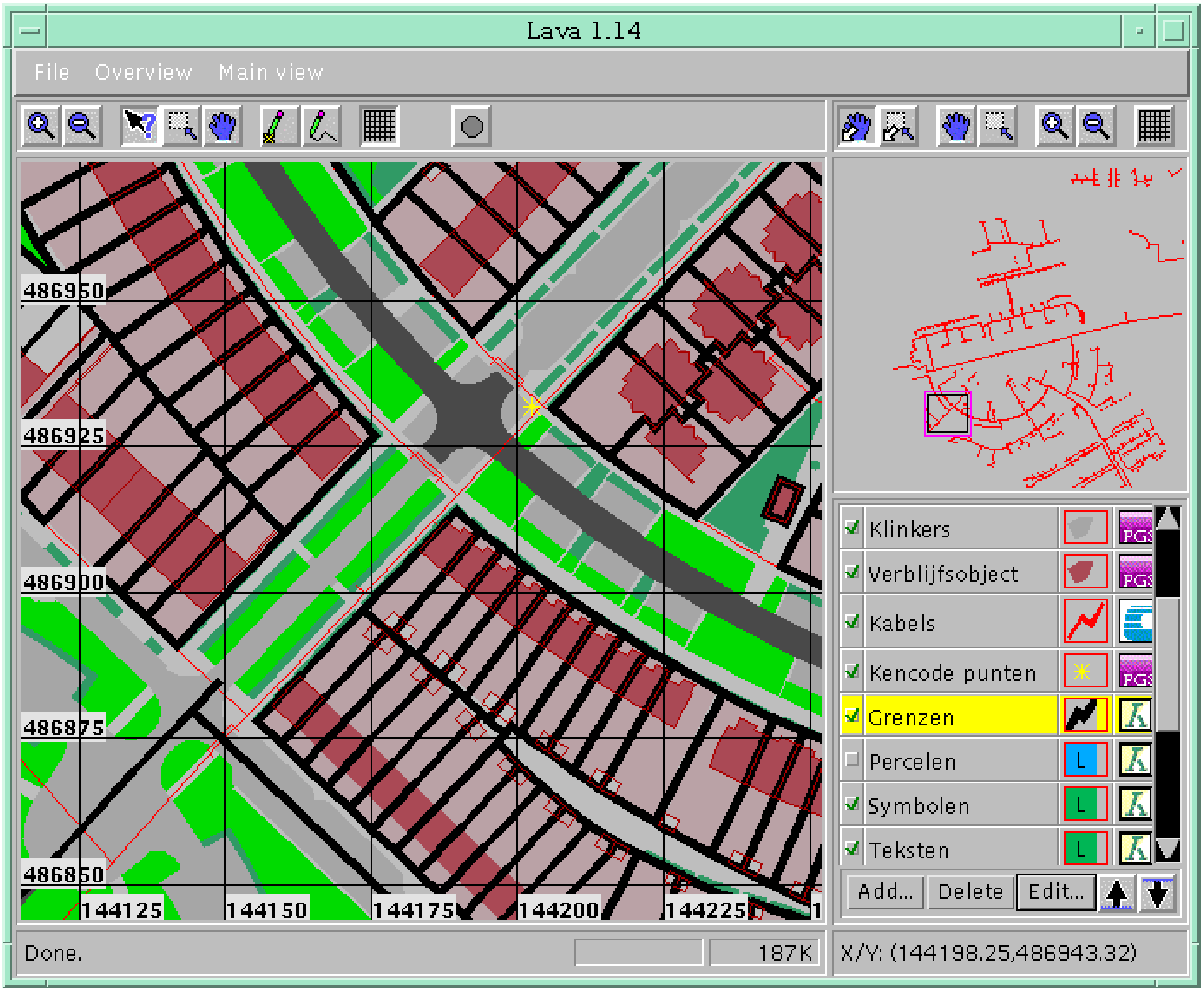


Figure 8: Selected parcel boundaries (source='GEOGR' en quality='T1') in bold black lines