MULTIPLE REPRESENTATIONS IN DBMS: TWO ALGORITHMS

Jantien Stoter[a] and Sisi Zlatanova[b]

[a] International Institute for Geo-Information Science and Earth Observation (ITC), Hengelosestraat 99, Enschede, The Netherlands, stoter@itc.nl

[b] Delft University of Technology, Jaffalaan 9, Delft, The Netherlands, s.zlatanova@otb.tudelft.nl

Commission IV/3

KEY WORDS: Databases, Generalization, Framework, Data structures, Representation, Retrieval, Modelling, Automation

ABSTRACT:

Spatial data sets are created to provide information for specific applications. These sets are representations of real world objects and are each collected by specific organisations for specific purposes. The objects of interests are modelled in a way appropriate for the application and therefore the data sets are a subjective selection of real world objects. Consequently, different representations of the same objects can be found in a variety of data sets. Dealing with several representations is related to two major research domains: efficient structuring and intelligent generalisation.

This paper addresses these two issues with respect to the functionality offered by spatial DBMS. Growing number of mainstream DBMS have been offering management of spatial objects. The number of implemented spatial operations increases as well. Many of these operations (or combinations of them) can be already successfully used to manage multi-resolution data. The paper discusses possibilities for data structuring (using the spatial data types offered by DBMS), algorithms for automatic linking of different representations and generation of new representations out existing ones. The algorithms are tested in a case study.

1 INTRODUCTION

A lot of research has been already conducted on multiple representations and generalisation related to spatial DBMS: Buttenfield and DeLotto, 1989; Devogele et al., 1996; Friis-Christensen et al., 2002; Grefen and Widom, 1997; Jones et al., 1996; Li and McLeod, 1992; Sheth and Larson, 1990; Spaccapietra et al., 1999), which is a clear indication for the significance of the issue and the need for support of multiple representations in DBMSs.

Two general principles for data organisation can be distinguished: single-resolution management (one real world object is translated into one instance in the database) and multiple resolution (one real world object has several representations in the database). Proposed frameworks by Frank and Timpf, 1994 and Vangenot, 1998 are examples of the first principle. This approach is very appropriate for modelling new data sets.

However, maintenance of several representations in one database is unavoidable in many cases. Multiple representations of real world objects already exist in many organisations. A typical example are the different representations of the same real world objects for different applications. Single representations are dependent on the subjective view of the user who has modelled the representation. This view is related to different aspects such as scale (also dependent on the amount of details to be represented), generalisation criteria, theme and time (Zhou and Jones, 2003). To be able to exchange such representations and to use the representations from one application in another application, a correspondence between the different data sets has to be established. Due to lack of efficient methods to establish a link between such representations (thematically and geometrically), very often a new process is initiated for data collection, modelling and updating of the same objects.

Here, we investigate the functionality presently offered by spatial DBMS to resolve two specific issues related to different geometries of one object existing in different data sets, i.e.:

- Possibilities for establishing a link between different representations of objects
- Automatic generation of low-resolution representations from high-resolution representations

In this paper both issues are discussed in the context of an object-relational DBMS. The paper is organised in four sections. Section 2 describes the functionalities offered by mainstream DBMSs to support multiple representations in DBMSs. Section 3 describes approaches for linking different geometries, while section 4 focuses on automatic generation of low-resolution representations. Section 5 reports the results of a case study utilising developed functions and scripts. The case study is carried out within Oracle Spatial 9i.

The paper concludes on the usability of spatial functions for multi-resolution management at DBMS level and outlines further research topics

2 DATA STRUCTURING OF MULTIPLE REPRESENTATIONS IN DBMS

DBMS plays an important role in the new generation GIS architecture. The algorithms to interrelate different data sets and to create low-resolution data from high-resolution data are based on core DBMS functionality. Therefore, the functionalities available in DBMSs that can support modelling of multiple representations are presented first (see also Stoter and Zlatanova, 2003).

2.1 Geometrical model

Mainstream DBMSs (Oracle, IBM DB2, Informix and Ingres) have implemented spatial data types and spatial operators (also called 'spatial functions') more or less similar to the Simple Features Specification for SQL of OGC (OGC, 1999). The implementation consists of an SQL extension using Abstract Data Types (ADTs) that supports storage, retrieval, query and updating of simple spatial features (points, lines and polygons). ADTs are used to be able to implement object-oriented technology in relational DBMSs. The spatial features are stored in geometrical primitives. Topological relationships between geometries can be retrieved by the use of spatial operators (see section 2.2).

To speed up spatial querying, spatial indexes can be built on a data set that is stored with geometrical primitives in the DBMS. Most DBMSs support the R-tree and the Quad-tree spatial index.

An important functionality that is supported in DBMSs, is validation of spatial features with respect to the Oracle object-relational model. Validation of spatial features checks if the stored spatial features are valid, e.g. polygons should have an area, outer boundaries of polygons should not be self-intersecting, there should be no repeated points in the sequence of coordinates etc. Validation is essential when supporting multiple representations in DBMSs. For example when a geometrical overlay is performed (see section 3 and 4), very small polygons can be created which do not refer to objects in reality. Using a validity function can check if the objects that were created are valid. Invalid object can then be further processed.

2.2 Spatial functions

Spatial functions when supporting multiple representations in DBMSs are needed both for determining interrelationships between different data sets and for deriving low-resolution data sets.

The OGC Simple Feature Specification for SQL (OGC, 1999) describes geometrical and topological functions that should be supported at DBMS level as part of the implementation of the geometrical primitive. Topological relationship operators between two geometries are implemented with respect to the nine-intersection model of Egenhofer. In the Egenhofer model each spatial object has an interior, a boundary, and an exterior. The boundary consists of points or lines that separate the interior from the exterior. The boundary of a line consists of its end points. The boundary of a polygon is the line that describes its perimeter. The interior consists of points that are in the object but not on its boundary, and the exterior consists of those points that are not in the object. Some of the topological relationships of the 9-intersection model have names associated with them that specify the type of relationship, e.g. ‘inside’ and ‘coveredby’. ‘Inside’ returns true if the first object is entirely within the second object and the object boundaries do not touch, otherwise, ‘inside’ returns false. ‘Coveredby’ returns true if the first object is entirely within the second object and the object boundaries touch at one or more points, otherwise, ‘coveredby’ returns false.

In Ingres the support for topological relationships is minimal. Oracle, IBM DB2, Informix and PostGIS support geometrical and topological functions defined by OGC and often more functions than these as reported in (Oosterom et al, 2002). Oracle Spatial 9i is used to illustrate the possibilities of spatial analysis using the geometrical primitive in DBMSs. Currently, Oracle Spatial supports three groups of selection operations, i.e. topological relationship operations, metric operations and specialisation operations. The names of the operations slightly differ from the ones suggested by OGC. In Oracle Spatial 9i all the topological relationships are implemented using one function (sdo_geom.relate) or operator (sdo_relate), where the type of relationship is passed as a text string (see table 1, right for the Oracle notations and table 1, left for the OGC notations). The spatial operator requires and utilises a spatial index and is therefore faster than the spatial function, which also work without a spatial index.

OGC	Oracle
Equals	equal
Disjoint	disjoint
Intersects	anyinteract
Touches	touch
Crosses	overlapbdydisjoint
Within	inside
Contains	contains
Overlaps	overlapbdyintersect
	Coveredby, covers, on

Table 1: Topological operations in the DBMS

Besides the relationship operations, many metric and specialisation operations are proposed by OGC that can take one (unary operations) or two geometries (binary operations), or other parameters (e.g. buffer size) and calculate new values or new geometries. The most important of them together with their Oracle equivalents are given in table 2.

OGC	Oracle
Unary metric operations	
Area	sdo_area
Length	sdo_length
Unary specialisation operations	
Buffer	sdo_buffer
Centroid	sdo_geomcentroid
Boundary	sdo_mbr
Convexhull	sdo_convexhull
Binary metric operations	
Distance	sdo_distance
Binary specialisation operations	
Intersection	sdo_intersection
Union	sdo_union
Difference	sdo_difference
Symdifference	sdo_xor

Table 2: Metric and specialisation operation in the DBMS

Another class of spatial operations in Oracle Spatial returns an aggregate of a collection of geometries. These are not defined within the OGC (see table 3).

```
SDO_AGGR_CENTROID: Returns the centroid (geometric object) of the specified objects.

SDO_AGGR_CONVEXHULL: Returns the convex hull of the specified objects.

SDO_AGGR_MBR: Returns the minimum bounding rectangle of the specified objects.

SDO_AGGR_UNION: Returns the topological union (OR operation) of the specified objects.
```

Table 3: Examples of aggregate functions in Oracle Spatial 9i.

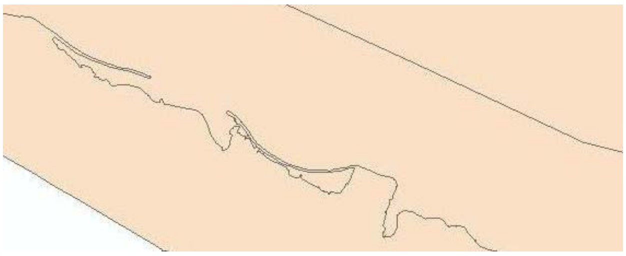
3 BUILDING INTERRELATIONS

As mentioned in the introduction, in many cases two or more representations of the same object may exist in different data sets. While attempting to define which object from the two data sets are the same, we have to define what kind of correspondence can be found in the two data sets. Since spatial objects can be characterised by their spatial and thematic characteristic, the differences and similarities can be related to theme and geometry. It may appear that one object from one data set can be linked to zero, one or many objects from the other data set. Such a multiplicity can occur in both the geometric and the thematic domain.

The following cases can be distinguished:

1. One object corresponds to only one object from the second data set. The relationship between the representations is then 1:1. In terms of spatial relationships this will mean 'object1 equals to object2'
2. An object exists only in one of two datasets but can be assumed that is part of conglomerate in the second data set. For example, the object 'bridge' cannot be found in the second data set because it is complete integrated within another object, e.g. 'road'. In such cases, we define relationship 1:0. Translated into spatial relationships, this case has to result in 'object1 inside/covered by object2'.
3. An object is an aggregation of several objects from the second data set. For example, the object 'road' in one data set may be confronted with 'primary road' and 'secondary road' in the second data. In this case, we define relationship 1:m. The spatial relationship will be 'object1 contains/covers object2'
4. The last most complex relationship is when two different themes are considered and the object cannot be thematically matched. For example, the object 'riverside' from a large-scale data set, which is only of interest for an organisation dealing with large-scale mapping and therefore not modelled in small-scale applications, cannot be related to a similar object in the small-scale data set. Instead it will overlap with other thematic objects in the small-scale data set, such as 'grass' or 'road'. In this case the spatial relationship will be 'object1 overlap/intersect object2'.

Considering only the theme of the objects, making the correspondence between two data sets looks a straightforward approach to be implemented at DBMS level (using the spatial operations, e.g. SDO_RELATE in Oracle Spatial 9i). However, scale and geometric resolutions (closely related to the scale) also influence the process of object referencing. When the scale is different, most commonly the outer rings of the polygons (in case of area objects) differ significantly. Although smaller, variations can be observed even in case of equal scales due to diverse data sources, data productions procedures and resolution (detail) used.



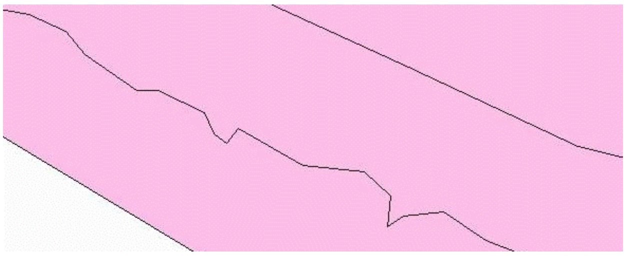


Figure 1: Scale differences (fragments): 1:1000 (up) and 1:5000 (down)

Bearing these geometric considerations, we have to expect that the boundaries of two objects will never completely 'coincide'. Thus, it will be rather problematic to execute spatial functions 'equals to' or 'covers' and obtain the unambiguous result 'TRUE'. In most of the cases, checking particular spatial relationships between two objects will be influenced by differences in the outer rings of the polygons. To avoid this problem, we have developed an algorithm, which does more than simply applying the topological operators.

The algorithm assumes that if two objects can be related, their geometries must intersect. The three general steps of the algorithm can be specified as follows:

- For each object from the first data set it is checked which are the objects from the second data set that 'interact'. The type of interaction is not important. Thus all the non-interacting objects are filtered out.
- For each object (from the second data set) that has been detected to interact with the object of interest (from data set 1), a new object 'intersection' is composed that represents the common (overlapping) area between the two. The area of the new object is computed.
- The areas of the object of interest and the 'intersection' object are compared. A decision if the two objects can be considered the same is taken using a threshold. For example, if the areas are more than 90% the same, the two objects are considered the same (see Section 5 for more details).

This algorithm was implemented in Oracle Spatial 9i, using the high-level scripting language PL/SQL and the spatial functions SDO_RELATE with mask 'anyinteract' (for the first step), SDO_INTERSECTION (to compose the geometry of the 'intersection' object) and SDO_AREA (to compute the area).

4 DERIVING NEW REPRESENTATIONS

Creating new representations from existing ones always requires classification (based on theme) and aggregation algorithms (based on geometry). The issue can become extremely complicated. For example, considering only simplification of geometry (without theme differences), no ultimate set of algorithms exists at this moment, which can generate a dataset at a required small scale based on a data set at a larger scale.

Here, we follow a different approach: we make use of two existing data sets to derive a new data set. The underlying motivation is the existence of many different data sets (in topographic offices, municipalities, cadastre) that do have the same (or similar) themes or that contain slightly different themes but map the same resolution. An appropriate combination of such data sets may significantly reduce the efforts and time for delivering the requested data. Depending on what types of data are available, different algorithms can be developed. In our approach a low-resolution data set can be generated from a high-resolution data set by using a low-resolution data set containing a different theme.

For the scope of this paper we concentrate on a particular case (see also Section 5), i.e. two data sets in two different scales (high resolution and low resolution) and with slightly different themes. The goal is to obtain a new data set that has the resolution (scale) of the first data set but the theme (objects) of the second data set. We assume that the themes are similar, i.e. at certain classification level, the objects can be matched. The interest here was primarily on the applicability of spatial operations. We have established the cross-references between the themes of the two data sets manually, which is the most common way to link objects at the thematic level. The algorithm for deriving a new representation can be specified as follows:

- For each object with a particular theme, establish link with all the objects from the second data sets that interact (following the algorithm described in Section 3) AND have a similar theme. For example 'river' and 'bottom of river'
- Create an 'aggregation' of the objects of the second data set that fall in one object (from data set 1).
- Assign to the objects of data set 1, the theme properties of data set 2. Only matched objects have to be considered.

The algorithm was implemented in Oracle Spatial 9i (using PL/SQL) and the spatial functions SDO_AGGR_UNION (for the aggregation of objects).

5 CASE STUDY

The two algorithms were tested on three data sets named here DS1 (scale 1:1000), DS2 (scale 1:1000) and DS3 (scale 1:50000 with the following characteristics:

- DS1 and DS2 have different objects and different geometries, but the scale is the same. DS1 has been created on the basis of topographic boundaries (grass, river, forest, etc.), while the DS2 on the basis of maintenance characteristics (road, facilities, etc.). In principle, it is possible to have boundaries in DS2 that do not follow topographic boundaries.
- DS2 and DS3 are both designed for maintenance purposes. Consequently the theme of the objects is the same (given with a unique code) however the scale is different (see Figure 1). The polygons of the DS3 are defined with much less points compared to the polygons in DS2.

More details on the data sets and the tests can be found in Zlatanova et al 2003 and Binkhorst and Zlatanova, 2003.

The three datasets are produced and maintained by three different organisations within the Ministry of Transport and Public Works on different platforms and in different systems. This case study is part of a larger project on usability of Oracle Spatial for the support of infrastructure work-processes within the Ministry. The three data sets were imported in Oracle Spatial 9i in three different tables (using SDO_GEOMETRY data types), indexed (R-tree spatial index) and validated (as specified in Section 2).

We tested the fist algorithm for all the objects belonging to DS1&DS2 and DS2&DS3. Since the objects of the maintenance maps (DS2 and DS3) have the same theme, linking the objects was tested also considering thematic codes.

The results of the tests can be summarised as follows:

DS2&DS3: These two maps supposed to have objects that can be classified as having relations 1:1 (i.e. Group 1, Section 3). The statistics show that using the geometrical procedure about 85% of the objects of the two representations can be automatically linked, i.e. one object from the DS2 is matched with only one object from DS3. The tests have clearly shown that the threshold for overlapping areas can be very relaxed, all the objects with overlapping area larger that 60% can be considered as one object. In only few cases the overlapping area was smaller than 60% although the objects were the same (e.g. long, narrow objects). How many objects from the DS3 are not matched with any object was difficult to check, since the area covered by the DS3 was much larger. There were several reasons for not matching the remaining 15%, but they can be mostly considered as errors. For example, one of the objects in DS3 covers two objects in DS2, or missing objects (Figure 2). The results of this algorithm were better compared to the results from comparing theme codes (due to typing errors in the text string).

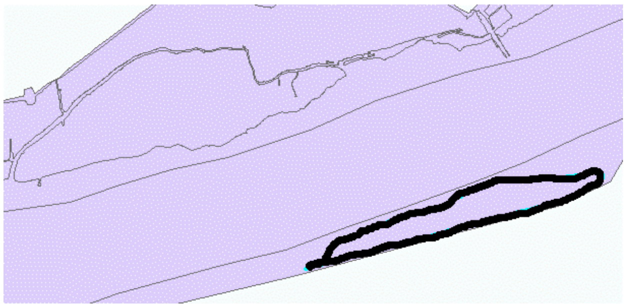


Figure 2: Missing object in one of the data sets (the thick polygon)

DS1&DS2: In this case, the thematic definition of objects was different and therefore there were more cases of the last three groups defined in Section 3. For example an object "river side" from DS2 may overlap with "grass" and "trees" from the DS1. Clearly, when a DS1 object has 100% overlap with DS2 object then the object is completely 'inside' the DS2 object (i.e. group 2) and can be linked to it. It has to be a multi-step process in which first objects that have a 100% match can be linked, then link 'obvious' objects (those which are e.g. 60% or 70% or more inside a matching object). The remaining objects will require some additional 'rules' (e.g. assign the object to the matching object with the largest overlap if this overlap is more than 50% of the input object). We expect some objects will remain which cannot be matched. This may require a match between the objects defined in a different way. The last step of the algorithm was not tested. We have matched all the objects that have an overlap of 100% (58%), larger than 80% (70%) and larger than 60% (74%).

Similarly to the second algorithm, a simplification of the objects match between DS1 and DS2 will be achieved applying an object aggregation with respect to theme classification. For example, the river in DS2 is subdivided into several additional parts with respect to the usage. All these parts (which interact) are first united (the thick polygon in Figure 3, up). DS1 does not have theme subdivision of rives, but it has subdivision with respect to data collection procedure (aerial stereo-pairs). Figure 3 (down) illustrates the resulting new object.

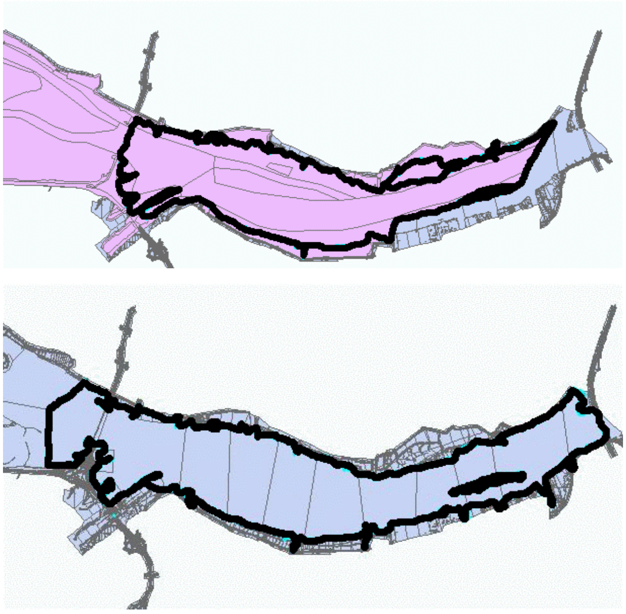


Figure 3: Union of objects according to a theme classification: DS2 (up) and DS1 (down)

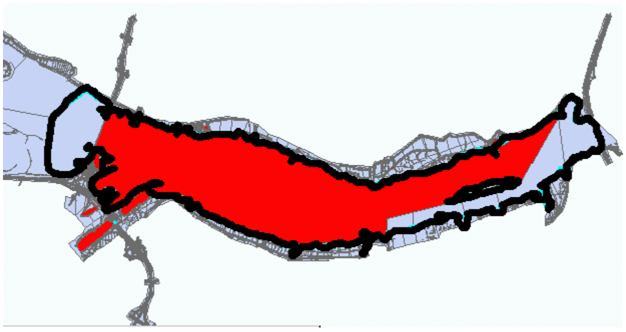


Figure 4: Match between the new objects of DS1 (the thick polygon) and DS2 (red-filled polygon)

Apparently, matching the newly created objects is much easier and intuitive (Figure 4).

The second algorithm was tested with data sets DS1 and DS3, i.e. they were used to create data set DS2. Recall the beginning of this Section about the description of the data sets. DS1 and DS2 have the same scale but different theme and DS2 and DS3 have the same theme, but different scale. Thus the aim is to create objects that have the higher resolution of DS1 (scale 1:1000) and the theme classification of DS2. As mentioned before, the three data sets are maintained separately. From company point of view, this process will greatly reduce efforts and time to create DS2 (suppose DS1 and DS3 already exist and DS2 does not yet exist).

Figure 5 shows a new object as a result of the executed script. Shown are the set of objects of DS1 that intersect with one object of DS3 and have similar themes. After aggregating the entire set of selected objects, a new object is created that follows the outer boundary of the aggregation. The difference between the original object of DS2 (the thick polygon polygon) and the newly created object is minimal.

Figure 5: Object of DS2 (the thick polygon) generated from DS1 (the objects inside) and DS3.

The tests have shown that the procedure is very successful. Several factors have to be taken into consideration while generating new objects: ontology schema, precision and accuracy of maps and gross errors. We expect that in some cases manual intervention will still be necessary.

The complexity of the process as mentioned above varies with respect to the data sets used. For example several more iterations are needed for the link between DS1 and DS2 compared to the link between DS2 and DS3. The number of objects in the DS1 is rather large and there are still quite many other errors (e.g. self-overlapping polygons, gaps, wrong layers, etc). As it was shown, many objects of DS1 (parts of rivers, or other objects) can be merged first (using appropriate conditions) and after that linked to DS2 objects. The link can be further controlled with respect to the classification of the two types of objects.

6 CONCLUSIONS

Although the optimal case would be to have one DBMS representation of a real world object, multiple representations (based on different themes or scale) exist and will continue to exist. Support for multiple representations in DBMSs is indispensable with respect to the growing role of DBMSs in the new generation GIS architecture.

In this context, the functionality of DBMS is critical. Two aspects of multiple representations in DBMS were explored in this paper: 1. automatically establish links between different data sets and 2. automatically generation of low-resolution data sets from high-resolution data sets. Two algorithms utilising Oracle Spatial functions were developed and tested with three different data sets.

The functionality currently offered by Oracle Spatial allows establishing a link between the three data sets from the case study. From this study it can be concluded that the results obtained from overlapping geometries are better than comparing themes. The complexity of the process also varies with respect to the data sets used.

The aggregation of several objects into a new object according to thematic characteristics is a relatively simple and straightforward process, which can be easily completed utilising available spatial functions in the DBMS. This can be used as a practical solution for comparing objects from different applications. The geometries of the objects can be aggregated according to a hierarchical classification up to a level at which the aggregated objects have the same meaning. Combinations of theme aggregations with geometry overlap gives the freedom to develop procedures with a diverse complexity. The creation of new data sets is an example of such an elaborated procedure.

Data consistency and integrity would be significantly improved if multiple representations were organised in one virtual DBMS. Different representations of the same real-world object can be stored in a distributed DBMS environment together with the integrity rules as well as rules for transformation between the representations. In this context both users and the system should be aware of the multiple representations. Data consistency of different representations is enforced through appropriate, automatic update propagation.

Further research is needed in many areas and will focus on the following aspects:

- Functions offered by DBMSs can be readily used for obtaining new geometries. Next step is consistent organisation of new objects in the structures used for multi-resolution management, e.g. DBMS-views could be used for representing data sets using generalisation procedures for geometry and theme properties once these procedures have been implemented in the DBMS.
- Data structures for management of multiple representations. The correspondence between spatial objects in a multi-representation environment has to be further organised in appropriate data structures.
- Functions for consistency checks and functions for propagating updates between different datasets are needed.
- Generalisation procedures for geometry – simplifications of boundaries, changes in dimensions of objects, aggregation of several small objects, selection of significant objects need to be further developed.
- Generalisation procedures with respect to the theme: aggregations of geometry based on theme properties and computation of a new object need improvements.

ACKNOWLEDGEMENTS

This research is sponsored by the Rijkswaterstaat, Ministry of Transport and Public Works and the research programme Sustainable Urban Areas at OTB, Delft University of Technology, The Netherlands

REFERENCES

Binkhorst, V. and S. Zlatanova, 2003, De mogelijgeden van Oracle Spatial voor de natte sector, *GeoNieuws 2003*, No.2, pp. 6-9, 2003 (in Dutch)

Frank, A., and S. Timpf, 1994, Multiple representations for cartographic objects in a multi-scale tree: an intelligent graphical zoom, *Computers&Graphics*, 18(6), 1994.

OGC, 2001, OpenGIS Consortium. The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema), Version 5. *Technical Report OpenGIS Project Document Number 01-101*, Wayland, Mass., VS, 2001.

OGC, 1999, OpenGIS Consortium. Opengis simple features specification for SQL. *Technical Report Revision 1.1, OpenGIS Project Document 99-049*, OpenGIS Consortium, Wayland, Mass., VS, 2001.

Friis-Christensen A., and C.S. Jensen, 2003, Object-Relational Management of Multiply Represented Geographic Entities.

Buttenfield, B.P. and J. S. DeLotto, 1989, Multiple Representations – Scientific Report for the SpecialistMeeting. *Report 89-3, NCGIA, Department of Geography*, SUNY Buffalo, USA

Devogele, T., J. Trevisan, and L. Raynal 1996, Building a Multiscale, Database with Scale-transition Relationships, in: *International Symposium on Spatial Data Handling*, pp. 337–351

Friis-Christensen, A., D. Skogan, C. S. Jensen, G. Skagestein, and N. Tryfona, 2002, Management of Multiply Represented Geographic Entities, in: *IDEAS*, pp. 150–159

Grefen, P. W. P. J., and J. Widom, 1997, Protocols for Integrity Constraint Checking in Federated Databases, *Distributed and Parallel Databases*, 5(4):327–355

Jones, C. and D. Kidner, L. Luo, G. Bundy, and J. Ware. Database, 1996, Design for a Multi-scale Spatial Information System, *International Journal of Geographic Information Systems*, 10(8), pp. 901–920

Li, Q. and D. McLeod. Managing Interdependencies among Objects in Federated Databases, 1992, in: *IFIP Database Semantics Conference on Interoperable Database Systems*, pp. 331–347

Oosterom, van, P.J.M., C.W. Quak, and T.P.M.Tijssen (2002). Testing current DBMS products with real spatial data. In Proceedings UDMS 2002, Prague, Czech Republic, October 2002.

Sheth, A.P. and J. A. Larson, 1990, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, in: *ACM Computing Surveys*, 22(3), pp. 183–236

Spaccapietra S., C. Parent and C. Vangenot, 2000, GIS Databases: From Multiscale to MultiRepresentation, in *Abstraction, Reformulation, and Approximation,* B.Y.Choueiry and T.Walsh (Eds.), LNAI 1864, Springer

Stoter, J.E. and S. Zlatanova, Visualisation and editing of 3D objects organised in a DBMS, EUROSDR workshop: Rendering and visualisation, Januari 2003, Enschede, The Netherlands.

Vangenot, C. 1998, Multiresolution representation. Concept fo rthe description of multi-resolution spatial representation databases, *International Journal of GIS and Spatial Analysis*, Hermes, 8(1-2), p. 121-248

Zhou Sh. and C.B. Jones, 2001. Design and Implementation of Multi-Scale Databases. 7th International Symposium on Spatial and Temporal Databases SSTD01, *Lecture Notes in Computer Science 2121*, pp. 365-384.

Zhou, Sh. and C. B. Jones: 2003 A Multi-representation Spatial Data Model. In T. Hadzilacos et al. (Eds.): Advances in Spatial and Temporal Databases, in: *Proceedings of the 8th International Symposium on Spatial and Temporal Databases*(SSTD 2003), pp. 394-411, LNCS 2750

Zlatanova, S., T.P.M. Tijssen, P.J.M. van Oosterom and W. C. Quak, Research on usability of Oracle Spatial within RWS Organisation, *GISt No.21, ISSN: 1569-0245, ISBN: 90-77029-07-9, AGI-GAG-2003-21*, Delft, The Netherlands, 75 p. 2003.