TOPOLOGY UNDER THE MICROSCOPE

Jildou Louwsma, Theo Tijssen and Peter van Oosterom put Laser-Scan's Radius Toplogy v1 to the test in comparing topologically structured and 'plain' spatial data

Radius Topology is an advanced spatial processing environment, which extends the Oracle9i database and provides dramatic improvements to the speed and consistency of spatial data handling. Well that's the claim. So how does it measure-up in practice?

As part of Laser-Scan's Radius academic programme, Delft University of Technology in The Netherlands has been putting the company's Radius Topology 1.0 product through its paces. This article outlines some of the findings. But first, some background.

Extending roots

Database management systems (DBMSs) have their roots in managing administrative data. However, data that contain spatial components are not so easily accommodated. Over the past couple of years, several DBMSs have been extended to handle spatial data. The next step is the support of topological structures in the DBMS. A topological structure is an elegant and robust way to model relationships between geospatial features and confers a number of benefits:

- avoids redundant storage
- easier to maintain consistency after editing data
- more efficient for certain query operations

In extending the Oracle Spatial database, Laser-Scan's Radius Topology is the first commercial system able to manage topologically-structured data in a DBMS.

What's best?

The main question to be addressed by Delft University of Technology is: "Is a geo-DBMS better, both in terms of functionality and performance, if it uses topology to manage spatial data?"
The functional analysis needed to answer this question hinges mainly on running a number of typical spatial queries against test data set. And for this, the topological queryingpossibilities of Radius Topology are almost equivalent to those available in Oracle9i Spatial.
In practice, the user specifies rules that control how Radius Topology creates topological structures from the plain geometries. All subsequent editing will conform to these rules. Small errors are automatically rectified, and data that cannot be structured are rejected There is no functional equivalent of this in geo-DBMSs that do not support topology.

Testing time

For the studies at Delft, a 1:50,000 scale dataset from the Dutch Topographic Service (TDN) was employed that contained area features relating to the Netherlands region of Flevoland. The initial topological structuring process detected errors in a supposedly clean data set, thereby highlighting the importance of topology for data quality. After structuring, the Oracle table 'flevo_areas' has the following columns:

OID identifier	number(11)	unique object
SHAPE	sdo_geometry	plain geometry
		topological ID,
TOPO_ID	number(38)	reference to topological primitive
MANI_ID	number(38)	manifold ID

Radius Topology adds the last two columns to maintain the references to the topology structures, the SHAPE column contains the original, plain geometry.
In Table 1 below we can see that about 17 per cent fewer points are stored in the topology case (by avoiding storing 'common' boundaries twice). However the disk space required is 30 per cent bigger due to the increased number of topology primitives compared to the number of area features.

Plain geometry :	15.28 Mb (R-Tree index: 2.61 Mb)
edge geometry :	17.94 Mb
node geometry :	1.89 Mb

Table 1. Disk space requirements of plain geometry adn topology primitives

Table	Table (Mb)	Indexes (Mb)
node	2.34	7.41
edge	26.03	12.48
face	0.75	3.36
topo	1.69	1.26
edge_to_node	5.38	16.04
edge_to_edge	5.73	15.46
face_to_edge	5.69	16.37
line_to_edge	5.69	14.62
area_to_edge	0.69	2.71
	53.99	89.71

Table 2. Disk space requirements of topology (including primitives) and associated indexes

In addition, the total storage requirements for topology are increased by the ID's a
store the topological connectivity (see Table 2). The use of topology will only lowe
when the geometries of the features are complex (large number of vertices) and tl
linework between features.
Types of queries executed for the performance test are (see also Fig.1).
Query 1. find all features within a certain rectangular area (e.g. for display purpose
Query 2. find all features that overlap with a polygon supplied
Query 3. find and clip all features with a polygon supplied
Query 4. find all features that intersect with a (possibly long) query
Query 5. find all features that contain a given set of search points
Query 6. find all features that are adjacent to a certain feature (polygon)
Most of the queries to test performance compare against a geometry that is provic
geometry of a feature in the data. Only Query 6 is a test for topological relationshi
The area features to be used in the queries have to be constructed on the fly (exc
topologically-structured data is used. Using the plain geometry model the feature i
This explains why the spatial queries performed on the stored geometries were fa
performed on the topologically-derived geometries (see Table 3). However, Radiu
geometries to be retained, meaning that spatial queries can continue to run as qui
Only the topological relationship queries (such as query 6) will be much faster usi
of query at which topology excels, as it is obvious that looking up a few references
a number of relatively time-consuming geometric calculations.

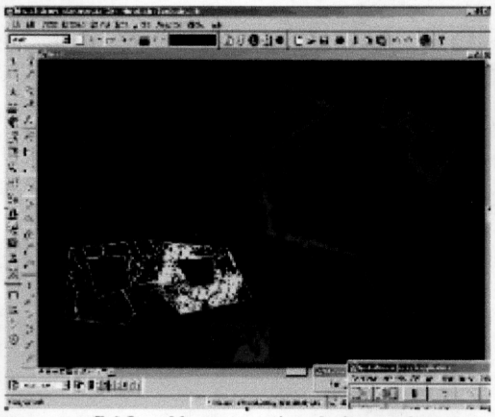
Fig.1: Some of the query geometries used and query result sets

Worst case scenario
The data used in this research constit
for topology if compared to plain geon
topological storage exceeds the simpl
requirements. The spatial relationship
when the geometries have to be deriv
If the geometries are stored as well, th
run at the same speed, although the s
increase.

Query	Query Type	# rows returned
1.1	rectangle	91
1.2	rectangle	958
2.1	polygon	296
2.2	polygon	573
3	clip	337
4.1	polyline	106
4.2	polyline	224
5.1	points	3
5.2	points	6
6.1	adjacency	5
6.2	adjacency	41

Table 3. Timing results of test queries (timings in seconds)

Conclusions
One conclusion is that queries for topological relationships are much faster when
The most important advantage of topology maintained within a geo-DBMS is the i
and consistency, and the ease with which errors can be detected and corrected.
Whether or not it is worthwhile to use topology in a geo-DBMS depends on the na
the data.
The experiences with Laser-Scan's Radius Topology in Delft have been interestin
made us more eager to continue the research, using other data sets, different que
(release 2 is now available!) and so on. In the years to come the advantages of m
geo-DBMS will become clearer with more efficient and enhanced implementations

showing already.

Peter van Oosterom is chair of theGeo-Database Management Center at Deft University of Technology (TU Delft) and can be contacted by email at: oosterom@geo.tudelft.nl while **Jildou Louwsma** and **Theo Tijssen** are with the GIS Technology Section of the Geodesy Department at TU Delft (www.gdmc.nl)