D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

UPDATING TOPOGRAPHIC DATABASES WITH ARC INFO ; CLIENT-FITTED CREATION OF CHANGE-ONLY INFORMATION

Jan BEYEN and Jeanne HENRION, Brussels

National Geographical Institute
Abdij Ter Kameren 13, 1000 Brussels, Belgium
Tel. (+32) 2.629.82.11 - Fax (+32) 2.629.82.12
E-mail : jbe@ngi.be

ABSTRACT

Most of the existing change-only solutions for updating topologically structured, topographic databases only work in ideal conditions. We therefore decided to develop a new procedure that would also be suitable for life situations. The client-fitted change-only information is produced through an automatic comparison of the producer's new database with the customer's old database. The detection of entirely identical elements and partly identical elements, and the updating of NGI-attributes are completely operated in the tables.

The biggest advantage of our method, for updating our own database, is the fact that all modifications can be performed as if they were corrections, without worrying about creating change-only data for the customers : change-only data are created afterwards. For updating the client's database, there is no need for unique object-ids or dates and we do not need to archive all intermediate database versions. The fact that the National Mapping Agency is taking care of the integration into the client's database, allows the client to go on using a less expensive viewing software.

1 EXISTING APPROACHES TO THE PROBLEM OF UPDATING TOPOGRAPHIC DATABASES

In Oxford, 1991, some NMA's (National Mapping Agencies) shared their first thoughts about updating topographic databases. J. Farrow (O.S.-U.K.) intended to supply change-only information, while Ch. Faad (I.G.N.-France) wanted to replace the user's topographic layer completely. M. Brand (O.S.N.I.) stressed the importance of preserving a historical perspective.

In 1993, O.S.N.I. organized a very important workshop in Belfast. It was decided to look for solutions to the problems of supplying change-only updates and of preserving a historical perspective. A. Winstanley already proposed a very interesting object-oriented solution. None of the NMA's, however, used object-oriented databases at that time ; so this solution came perhaps a bit too early, but we're sure that it will be useful in the next decade. Prof. Galetto offered a solution for updating within object-based relational topographic databases. After that, all theoretical and practical approaches with relational models had in common that they used or intended to use both dates and unique object-ids (Combes 1993 ; Galetto 1994 ; Beyen 1994 ; Birth 1995).
The relational approach has some disadvantages : It works perfectly, but only if some conditions are fulfilled. It aims at customers who, like the data producer, work with a relational database, who do not modify the geometry by themselves and who have the discipline to keep the given unique object-ids. In practice we never met such a customer.

In Emmen, 1995, G. Mitchell mentioned problems occuring with the data-exchange, when data-suppliers and data-users use different dates. K. Birth reported that often, the users' hardware and software platforms are not prepared to manage the complex data-structures. In this respect, in 1997, O.S.N.I. admitted that "The potention to receive change-only information has not so far been exploited due to the tardiness of system vendors to produce software which is capable of accepting change-only data".

Besides, the relational approach overlooked the warning that the object-numbers of two objects, bought in two different companies, risk to be the same (Beyen, 1993). In his discussion of an object-oriented solution, P. Woodsford (1996) recalled that "The key technical challenge is to devise workable schemes for unique object- ids, particularly where there are multiple issuers/owners of data". And "the restrictions arising from such a centralised approach to the data model are onerous for a wider user community". Additional disadvantages are the fact that the use of a complex updating procedure requires more well trained staff (Birth, 1995) and that preserving a historical prespective means a considerable extra effort (Brand, 1991), concerning which P. Woodsford (1996) remarked : "It may of course be uneconomic or unnecessary to do so !".

In 1995, P. van Asperen proposed a completely new way of updating topographic databases, clearly distinct from both the object-oriented approach and the relational model approach. On the other hand, he still thought about using unique ids and dates : "Change only datasets can be produced by subtracting the old dataset from the new dataset (...). To facilitate the change-exchange, a unique identifier for each element will be very useful (...). TDN is still busy to define (...) how to administer the delivered datasets". In 1996 : "Data-delivery through change-only datasets to users is still an issue to be looked into (...). Such update delivery requires an unchanged geometry at the user's side, or a system based on unique identifiers, maintained at both the producer's and the user's side (...). Change-only information can be produced through automatic graphical comparison of features or through a selection on

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

database entry date (...). Structural changes, due to splitting up or merging of features when connecting features are added or deleted, can be automatically filtered".

2 PRACTICAL UPDATING EXPERIENCE AT THE BELGIAN N.G.I.

In autumn 1996, we started an updating production line for our ARC INFO 7.0.2 basemap database and by spring 1997 we successfully printed our first series of 16 updated and upgraded maps at scale 1:10.000.
For updating our database, we interactively integrated the change-only information, that had been furnished by the Photogrammetry Section. At the same time we interactively created "clean and identified" change-only information for our customers, which should allow them to integrate our modifications without loosing their own attribute columns.

This method proved to have some disadvantages :
- interactive creation of change-only information risks to suffer from human errors ;
- the customers have to deal themselves with the integration into their database ;
- the customers are obliged to integrate all updating releases, in the right order and without skipping any release.

3 CHOICES MADE FOR THE DEVELOPMENT OF A NEW UPDATING PROCEDURE

- In order to allow the users to keep their specific attribute data, the updating should be "change only". This also means that the links to attribute data of both N.G.I. and customers should be kept.
- For every customer, we should create new, fitting change-only data.
- As we mentioned before (Beyen 1993, 1994), we should take care of the integration of the update into the customer's database. After this integration, the customer's database should have an optimal resemblance with ours, i.e. be a perfect copy, except where differences have specially been introduced by the client. The integration includes of course restoring topology (points, lines and polygones !). This way, the clients do not need to be able to restore the topology by themselves. Hence, they are not forced to purchase ARC INFO ; they can continue to use a less expensive viewing software, (combined with some ability to handle the attributes).
- In order to avoid human errors, change-only data should both be created and integrated in batch.
- The procedure should run on ARC INFO 7.0.2, without any supplementary RDBMS (e.g. ORACLE ...).
- Customers should be able to skip a release.
- Dates must not be needed as central parameters in the change-only procedure : the procedure should work as well for databases that did not keep dates; (but if necessary, like all attributes, dates can be kept and updated).
- Unique object-ids must not be needed either. This way, the procedure can also be used in countries where the NMA already produced and sold digital data without unique object-ids.

On basis of these specifications we decided to develop a series of AML's (ARC Macro Language programs), which automatically create client-fitted change-only information, by comparing our new database version with the customer's older version. We also developed a series of AML's, which together allow the customers to integrate this change-only information into their database.

Further advantages are that archiving of old versions is not really needed, since we compare our last version with the customer's last version, and last but not least, that, for updating our own database, we can perform all modifications as if they were corrections. The operators can do a job they're used to, without worrying about creating change-only data for the customers ; so we do not need to train the operators more than before.

Options :

- The procedure may be used for continuous revision as well as for cyclic revision at the NMA ; users can buy an update snapshot whenever they want to.
- It is possible to first cut out the same area, as purchased by the customer, before automatically comparing the data.
- If the customer does not want us to see his database, he can obtain our updated version of the complete database and produce his change only information by himself (after adding his items, with default values, to our coverages).

Limitations :

- The procedure is meant for delivering change-only updates to customers, who can read our Arc Info coverages. It was never intended as a universal solution for exchange between different conceptual data models, through translation to and from some standard exchange model, nor through an exchange format.
  For delivery to clients who modified the data structure, we would have to ask them to first adapt our new data the same way ; then run a slightly different procedure, which is specially developed for the occasion.
  For delivery to customers, who use a software, which cannot directly work with Arc Info coverages, we can of course produce and translate the change-only data, but the integration procedure should be re-written for the customer's environment.
- The procedure is not meant for interrogating a database with a historical perspective.
- Neither semantic relations, nor complex objects have been studied yet, because there are not any of these in our data structure.
- The programs were developed for our 1:10.000 map sheets of 8 km by 5 km. Hence, they use cleaning values between 4 cm and 15 cm, which are totally acceptable at our working scale, but which might be a problem at very large scales. For smaller scales, often with bigger map sheets, one should not overlook that, for single precision coverages, the cleaning values would automatically be adapted in function of the map sheet size. If not looked at, this might disturb the balance between some components, especially in the case of our first (graphical) solution.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

4 A FIRST, GRAPHICAL SOLUTION

Since ARC INFO does not support line-to-line topology, nor point-to-point topology, we first wrote some functions for graphical subtraction of database coverages: point coverages, line coverages or polygone coverages. If we define two objects to be different, when they have a significant difference in their positions, the output of a subtraction can be seen as a collection of objects which are different e.g. between the old and the new coverage. We therefore consider that the subtraction functions allowed us to "update the objects" for each coverage (after adding supplementary items with default values of the customer's "old" coverage to our "new" coverage).

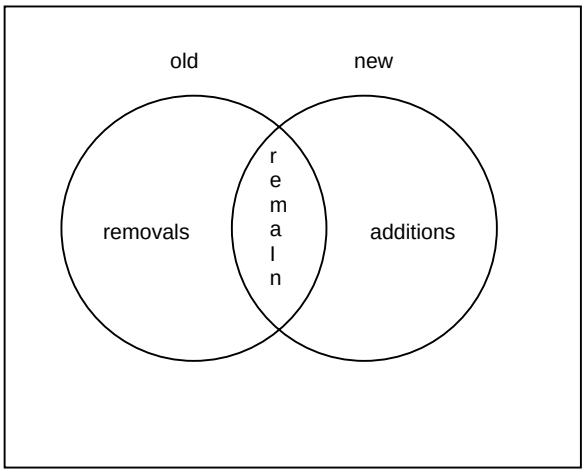

Figure 1. Relation between data sets.

The main operations for updating the objects are :

old – new(∩old) = removals
new - old(∩new) = additions
old – removals = remaining
remaining + additions = updated

followed by checking that new - updated = 0
and updated - new = 0.

At the + sign, we used the ARC instruction APPEND with the appropriate parameters. The - signs stand for subtraction functions, which are slightly different for every line above : for a certain data type (e.g. lines) all subtraction functions contain the same sequence of (buffer, clip, erase, append) ARC instructions, but each one uses some of these ARC instructions with specific parameters. The ∩ operation in the first subtractions is implicit. In between these operations, some snapping and cleaning is done.

Apart from this, we developed a series of AML's for updating the NGI-attributes in the "remaining objects" of the customer's database ; i.e. the objects whose position had not been modified significantly. Here again, we distinguished the point, line and polygone cases.
In practice the final checks "new-updated" and "updated-new" did not result in empty files, but in files containing about 1 element per 1.000 elements in "new". After solving these problems manually, we obtained a nice and clean result which was visually perfect.

However, "updated" contained much more line elements than "new", because the lines had been segmented due to the clipping. For oriented lines, this segmentation cannot be undone by a simple unsplit, because of the risk of flipping. For the other lines, an unsplit might only be performed if we take into account both the NGI-attributes and the customer's attributes.
Apart from the segmentation, this solution has the disadvantage of being rather slow for updating the objects and extremely slow for updating the attributes of the "remaining objects". On the other hand, it has the advantage of considering two nearly coinciding elements (= within the cleaning distance) as an identical object, so the customer does not loose his specific attributes, unless the difference in geometry makes it worth-while.
A second advantage is that we are able to update NGI-attributes for lines which have been split by the customer.

5 A SECOND SOLUTION, THROUGH TABLES

In order to pace up and in order to avoid segmentation for the line elements, we looked for a second solution, using the tables.

For point symbols and label points, it is obvious that, after ADDXY, the co-ordinates can easily be compared, using a relation between the .PAT tables. This way, we can select points, which remained in the same position and update their NGI-attributes. The rest of "old" are points to be removed ; the rest of "new" are points to be added.

For the line elements, we used both the .AAT and the .NAT tables. After having established the right relations, we were able to isolate the line elements, which were geometrically identical in the old and the new data : oldcom and newcom (see fig. 2). We also immediately adapted the NGI-attributes in "oldcom" (Using "oldcom" allows to keep the client's specific attributes).
After this, we intersected the remaining "old" and "new" into "xold" and "xnew", and we repeated the first series of operations in order to isolate the parts of line elements, which were geometrically identical in the old and the new data : xoldcom and xnewcom. This way, we were able to both adapt NGI-attributes and keep client-attributes as well for the parts of lines that remained in the same position.

For the integration into the client's data-base : append oldcom + xoldcom + xnew. The attribute "upda" was added at the beginning of the program and it was adapted at every stage. It shows which intermediate coverage the element is coming from and it indicates whether the element's NGI-attributes have been modified or not.
Since the addition contains line-parts, resulting from the intersection, it holds more elements than our original "new". In order to diminish this difference, we intend to execute a conditional unsplit on the non-oriented lines, taking into account both our and the client's attribute values.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

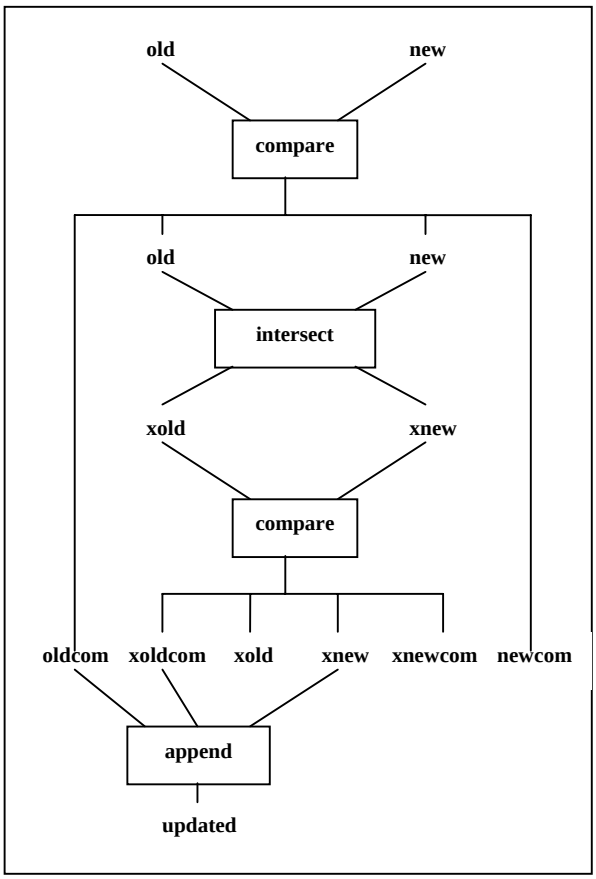

Figure 2. Producing line update information through TABLES.

The TABLES-solution for creating client-fitted change-only information works rather quickly and it gives a pure result, without any uncontrolable segmentation, but it does not offer the advantages mentioned for the first solution, viz. of keeping the client's specific attributes, where the difference in geometry is smaller than the cleaning distance, and of being able to update NGI-attributes and to keep the client's attributes for lines which have been split by the customer.

6 PERFORMANCE AND STATISTICS

The modules were tested on 10 different map sheets of 40 km², using a HP 712-60 workstation with 64 Mb RAM and about 200 Mb of free disk space.
Every map sheet contains about ten coverages, but almost all modifications appear in four coverages :

VEG = polygone coverage with arcs for roads, railroads, hydrography and natural limits, and labels for land use.
COPAR = line coverage with treerows, hedges etc. parallel to the line elements of VEG, but which do not delimit any premise (cf. E. Bayers, 1994).
BATI = polygone coverage with buildings.
PUNT = label coverage with point symbols.

The arcs of COPAR do not intersect. In BATI, all attribute information is attached to the polygone labels ; so, for updating the arcs, the intersection and the second comparison do not make any sense.

At the moment, we still need an average of more than 3 hours for producing the complete client-fitted update information (see table 1). In exceptionally dense areas the runtime goes up to about 5 hours. Having in mind some possibilities for optimization, we hope to reduce the average total runtime significantly. A RAM-extension would of course be a great help.

7 PLANNED DEVELOPMENTS

In order to combine the advantages of both solutions, without having their disadvantages, we are studying some possibilities for distinguishing possible causes of differences between "xold" and "xnew" ; especially where the elements of both coverages are situated close to each other : has there been a merging, a splitting, or a very small displacement ?

At the deadline for submitting this paper, we already succeeded in selecting the elements of the remaining "xnew", which are situated within cleaning distance of the remaining "xold". Between half May and September 1998, we hope to diversify this selection and to recover the client's specific attributes. This should be possible, partly through TABLES, partly with a module, similar to a module of the first, graphical solution (cf. 4).
The execution time should not be a problem any more, since now we only have to treat very small numbers of elements.

The main flow chart for lines would then evolve towards figure 3 :

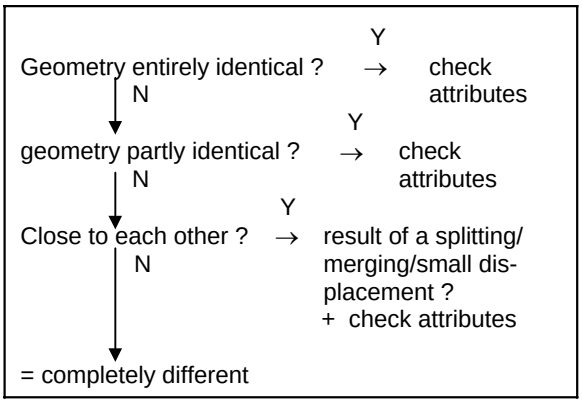

Figure 3. Expected main flow chart for lines

Independent from the planned developments mentioned above, we also intend to compare the results of our practical updating experience (cf. 2) with our new results. This should allow us to evaluate the human errors, made during our first updating experience.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

	ARCS			LABELS		
	min	max	average	min	max	average
VEG						
original old	8.223	17.760	12.403	2.183	5.341	3.511
original new	8.397	17.663	12.013	2.149	5.200	3.316
old/new-com	7.166	14.116	9.727	1.668	4.403	2.603
old (rest)	1.057	3.644	2.676	515	938	908
new (rest)	1.231	3.547	2.286	481	797	713
xold/xnew-com	722	2.900	1.832			
xold (rest)	561	2.524	1.761			
xnew (rest)	918	2.527	1.534			
updated	8.806	19.543	13.093			
COPAR						
original old	308	912	981			
original new	309	852	940			
old/new-com	304	744	881			
old (rest)	4	168	100			
new (rest)	5	108	59			
BATI						
original old	3.206	12.242	6.665	3.057	11.063	6.178
original new	3.691	12.322	6.879	3.465	11.601	6.617
old/new-com	2.996	10.315	5.758	2.911	10.132	5.701
old (rest)	210	1.927	907	146	931	477
new (rest)	695	2.007	1.121	554	1.469	916
PUNT						
original old				2.254	5.209	2.760
original new				2.115	5.098	2.631
old/new-com				2.026	5.007	2.560
old (rest)				227	202	200
new (rest)				89	91	71
TOTAL RUNTIME		ARCS + LABELS :		2h12'	5h06'	3h22'

Table 1. Statistics. Creation of change-only information through TABLES ; number of elements for the lightest mapsheet (rural area), the densest map sheet (urban area) and the average values for 10 map sheets.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

8 CONCLUSIONS

Updating a topologically structured, topographic database and integrating the produced change-only information into the client's database needs to be possible in all situations. We therefore developed two ways for automatically comparing the producer's new ARC INFO database with the customer's old ARC INFO database.

The automatic comparison avoids the risk of human errors and it has the advantage that, for updating our own database, we can perform all modifications as if they were corrections. For updating the client's database, there is no need for unique object-ids or dates and we do not need to archive all intermediate database versions. The customers can continue to use a less expensive viewing software. They are also allowed to skip releases : customers can ask for an update snapshot whenever they want to.

The second of our two solutions, viz. through TABLES, is clearly the quickest and it gives the best result. We are still working at some methods for even recovering the client's specific attributes in the different cases where the old and the new geometry are close to each other. This way, we intend to combine the advantages of our two methods, without having their disadvantages.

9 REFERENCES

Conferences :

[A] Updating digital data by photogrammetric methods, ISPRS and OEEPE joint workshop, Oxford 1991 (OEEPE publication 27, 1992).

[B] Updating of complex digital topographic databases, OEEPE and CERCO joint workshop, Belfast 1993.

[C] Mapping and Geographic Information Systems, ISPRS symposium, Athens (U.S.A.) 1994 (International Archives of Photogrammetry and Remote Sensing - Volume 30 part 4, 1994).

[D] Revision of digital topographic databases, CERCO WG IX workshop, Emmen 1995.

[E] Mapping and Geographic Information Systems, ISPRS commission IV, Vienna 1996 (International Archives of Photogrammetry and Remote Sensing - Volume 31 part B4, 1996).

[F] Revision of digital topographic databases, CERCO WG IX workshop, Emmen 1998.

Publications :

Bayers, E., 1994. S.I.G. et cartographie topographique, in Bulletin Trimestriel de la Société Belge de Photogrammétrie, Télédétection et Cartographie, 195, pp.41-53.

Beyen, J., 1993. How to destroy a client's data and still charge him for it ? (summarized in [B], part 2, appendix 7, paper 11).

Beyen, J., 1994. An "updating friendly" conceptual data model for topographic databases and its associated procedures, in [C] pp. 281-288.

Birth, K., 1995. ATKIS database revision in Nordrhein-Westfalen, in [D], session 2.

Brand, M., 1991. Updating a complex GIS database - The Northern Ireland experience, in [A] pp. 81-87.

Clements, R., 1993. Updating of complex digital topographic databases. (summarized in [B], part 2, appendix 7, paper 1).

Combes, M.C., 1993. BDCARTO road informations and updating strategies. (summarized in [B], part 2, appendix 7, paper 9).

Faad, Ch., 1991. Updating the French Topographic Database, in [A] pp. 59-65.

Farrow, J., 1991. Ordnance Survey Revision Problems and Solutions, in [A] pp. 73-79.

Galetto, R. and Viola, F., 1993. Conceptual model and database structure (summarized in [B], part 2, appendix 7, paper 2).

Galetto, R. and Viola, F., 1994. Time as the fourth dimension of the topographic databases, in [C] pp. 297-303.

O.S.N.I., 1997. Maintenance of a complex largescale topographical database at Ordnance Survey of Northern Ireland (LC/649/LMCM).

van Asperen, P., 1995. Updating TOP10 vector, in [D], session 2.

van Asperen, P., 1996. Digital updates at the Dutch topographic service, in [E] pp. 891-900.

Winstanley, A. and Mulvenna, M., 1993. Updating complex digital topographic databases - An object-oriented solution (summarized in [B], part 2, appendix 7, paper 6).

Woodsford, P., 1996. Spatial database update - a key to effective automation, in [E] pp. 955-961.