March 1995

EUROPEAN ORGANIZATION FOR EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH.

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

Report edited by S. Gray
Ordnance Survey of Northern Ireland

Official Publication No 30

Wir bitten, die vorgedruckte Empfangsbestätigung auf der unteren Hälfte dieses Blattes herauszutrennen, Ihre Adresse, bei Änderung auch die frühere, einzutragen und die Karte zurückzusenden. Bei mehreren Heften in einer Lieferung genügt es, nur eine Karte zurückzusenden, wenn die Nummern der anderen Hefte auf dieser notiert sind. *Sollten die Bestätigungskarten ausbleiben, müssen wir annehmen, daß Sie am Tauschverkehr mit uns nicht mehr interessiert sind, und weitere Lieferungen einstellen.*

Please detach the acknowledgement of receipt attached below, enter your address, in case of change enter your former address, too, and return the card.
If a shipment comprises several volumes, please return one card only listing all numbers. *If the receipt is not acknowledged we understand that you are no longer interested in our exchange of publications and that you wish to be cancelled from our mailing list.*

Ayez la bonté de détacher l'accusé de réception que vous trouvez sur la partie inférieure de cette page, d'y inscrire votre adresse et, le cas échéant, votre ancienne adresse et de retourner la carte.
Si la livraison se compose de plusieurs fascicules, il suffit de ne retourner qu'une seule carte portant également les numéros des autres fascicules. *Le fait d'avoir expédié un envois sans avoir reçu la carte de réception nous donnera raison à supposer que vous n'êtes plus intéressés à continuer l'échange de publications et que vous approuverez l'arrêt de nos envois.*

Le rogamos sacar el acuse de recibo figurando en la parte inferior de esta página, inscribir su dirección y si ha cambiado, inscribir la dirección anterior, y retornar esta tarjeta.
Si el envío se compone de varios fascículos es bastante retornarnos una sola tarjeta indicando también los números de los otros fascículos. *Si quedamos sin acuses de recibo tenemos que suponer que ya no tiene Vd interés en nuestro cambio de publicaciones y que desea Vd suspender otros envíos.*

Absender:
Sender:
Expéditeur:
Expedidor:

Es sind hier eingegangen:
We received:
Nous avons reçu:
Han llegado en nuestras manos:

OEEPE
Publ. off. No. 30
25 / 2.10 / 1948

Institut für Angewandte Geodäsie
Richard-Strauss-Allee 11

60598 Frankfurt am Main

Germany · Allemagne · Alemania

March 1995

EUROPEAN ORGANIZATION FOR EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH.

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

Report edited by S. Gray
Ordnance Survey of Northern Ireland

Official Publication No 30

ISSN 0257-0505

Printed and published by the Institut für Angewandte Geodäsie, Frankfurt am Main

EUROPEAN ORGANIZATION
for
EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH

STEERING COMMITTEE

(composed of Representatives of the Governments of the Member Countries)

President:	Dipl.-Ing. R. KILGA Bundesamt für Eich- und Vermessungswesen Krotenthallergasse 3 A-1080 Wien	Austria
Members:	Administrateur-Général J. DE SMET Institut Géographique National 13, Abbaye de la Cambre B-1050 Bruxelles	Belgium
	Ir. M. VAN DEN HERREWEGEN Director of the Dept. of Photogrammetry Institut Géographique National 13, Abbaye de la Cambre B-1050 Bruxelles	
	Mr. O. BRANDE-LAVRIDSEN Laboratoriet for Fotogrammetri og Landmåling Ålborg Universitets Center Fibigerstraede 11 DK-9220 Ålborg Ø	Denmark
	Mr. O. JACOBI Institut for Surveying and Photogrammetry Technical University of Denmark Landmaalervej 7 DK-2800 Lyngby	
	Abt. Dir. Dr.-Ing. H. BAUER Niedersächsisches Landesverwaltungsamt – Landesvermessung – Warmbüchenkamp 2 D-30159 Hannover	Federal Republic of Germany
	Präsident und Prof. Dr.-Ing. H. SEEGER Institut für Angewandte Geodäsie Richard-Strauss-Allee 11 D-60598 Frankfurt am Main	
	Prof. Dr. B. WROBEL Technische Hochschule Darmstadt Institut für Photogrammetrie und Kartographie Petersenstraße 13 D-64287 Darmstadt	
	Mrs. P. NOUKKA National Land Survey of Finland Box 84 SF-00521 Helsinki 52	Finland

Prof. Dr. R. KUITTINEN Department of Photogrammetry and Remote Sensing Finnish Geodetic Institut Ilmalankatu 1A SF-00240 Helsinki	Finland
Mr. P. DENIS Ecole Nationale des Sciences Geographiques 2, Avenue Pasteur F-94160 Saint-Mande	France
Mr. A. BAUDOIN Centre National d'Etudes Spatiales 2, Place Maurice-Quentin F-75039 Paris Cedex 01	
Dr. Eng. L. SURACE Geographical Military Institute Via Cesare Battista 8–10 I-50100 Firence	Italy
Prof. R. GALETTO University of Pavia Dept. di Ingegneria del Territorio Via Abbiategrasso 209 I-27100 Pavia	
Prof. Dr. M. G. VOSSELMAN Delft University of Technology Thijsseweg 11 NL-2629 JA Delft	Netherlands
Ir. P. VAN DER MOLEN Dienst Kadaster en de Openbare Registers Waltersingel 1 NL-7314 NK Apeldoorn	
Mr. I. INDSET Statens Kartverk N-3500 Hønefoss	Norway
Prof. Ø. ANDERSEN Norges Landbrukshøgskole Institutt for Landmåling P. O. Box 5034 N-1432 Ås	
Prof. J. TALTS National Land Survey of Sweden S-80112 Gävle	Sweden
Prof. K. TORLEGÅRD Royal Institute of Technology Dept. of Photogrammetry S-10044 Stockholm 70	
Prof. Dr. O. KÖLBL Institut de Photogrammétrie, EPFL GR-Ecublens CH-1015 Lausanne	Switzerland
Mr. F. JEANRICHARD Bundesamt für Landestopographie Seftigenstrasse 264 CH-3084 Wabern	

Lt. Col. M. ÖNDER
Ministry of National Defence
General Command of Mapping
TR-06100 Ankara

Turkey

Col. S. FOÇALIGIL
Ministry of National Defence
General Command of Mapping
TR-06100 Ankara

MR. N. S. SMITH
Ordnance Survey
Romsey Road
Maybush
Southampton S016 4GU

United Kingdom

Prof. Dr. I. J. DOWMAN
Dept. of Photogrammetry and Surveying
University College London
Gower Street 6
London WC 1E BT

SCIENCE COMMITTEE

Prof. Dr. I. J. DOWMAN
Dept. of Photogrammetry and Surveying
University College London
Gower Street 6
London WC 1E BT

United Kingdom

EXECUTIVE BUREAU

Mr. C. PARESI
Secretary General of the OEEPE
International Institute for Aerospace Survey
and Earth Sciences
350 Boulevard 1945, P. O. Box 6
NL-7500 AA Enschede (Netherlands)

Ir. J. TIMMERMAN
Smaragdstraat 20
NL-7314 HG Apeldoorn

SCIENTIFIC COMMISSIONS

Commission A – **Aerotriangulation**

President: Prof. Dr. T. SARJAKOSKI
Geodetic Institute
Ilmalankatu 1 A
SF-00240 Helsinki

Commission B – **Digital Elevation Models**

President: Dr. G. ROBINSON
NUTIS
University of Reading
White Knights
GB-Reading

Commission C – **Large Scale Restitution**

President: Prof. Eng. S. DEQUAL
Dipartimento Georisorse e Territorio
Polytecnico di Torino
Corso Duca Degli Abbruzzi 24
I-10129 Torino

Commission D – **Photogrammetry and Cartography**

President: see Commission I

Commission E – **Topographic Interpretation**

President: Prof. Dr. B.-S. SCHULZ
Institut für Angewandte Geodäsie
Richard-Strauss-Allee 11
D-60598 Frankfurt am Main

Commission F – **Fundamental Problems of Photogrammetry**

President: Prof. Dr. W. FÖRSTNER
Institut für Photogrammetrie
Universität Bonn
Nußallee 15
D-53115 Bonn

APPLICATION COMMISSIONS

Commission I – **Topographic Mapping**

President:

Mr. M. J. D. BRAND
Director
Ordnance Survey of N. Ireland
Colby House, Stranmillis Court
Belfast BT 9 5BJ
United Kingdom

Commission II – **Cadastral Mapping**

President:

Ir. L. A. KOEN
Dienst
v. h. Kadaster en de Openbare Registers
Waltersingel 1
NL-7314 GH Apeldoorn

Commission III – **Engineering Surveys**

President:

Mr. A. FLOTRON
Gemeindemattenstrasse 4
CH-3860 Meiringen

Commission IV – **Environmental/Thematic Surveys**

Commission V – **Land Information Systems**

UPDATING
OF
COMPLEX DIGITAL TOPOGRAPHIC
DATABASES

(with 2 Figures and 12 Appendices)

Report edited by S. Gray

Ordnance Survey of Northern Ireland

Table of Contents

1 ABSTRACT

This project addresses what has to date been a largely underdeveloped aspect of digital technology. Complex topographic databases are needed to underpin today's GIS, but their usefulness can rapidly be degraded if maintenance is neglected. It is essential to have in place, from inception if possible, efficient and effective updating methodology which can handle incorporation of new information and satisfactorily cater for retention of the historical perspective if this is necessary.

The project seeks to achieve its objectives through pooling the resources and expertise available within the OEEPE membership. It is therefore a combined attempt to produce ideas which will act as a catalyst, both to pioneer in-house solutions to the complex issues involved and to promote essential development by GIS suppliers.

2 EXECUTIVE SUMMARY

This project was initiated to examine the problems and address the growing need for more user-friendly methods to update the complex digital topographic databases increasingly underpinning a range of GIS applications. A project proposal was prepared by OSNI in collaboration with other OEEPE members and approved by OEEPE in October 1992.

The proposal envisaged a two phase practical project:

Phase I – generation of ideas.

Phase II – validation of these using common digital data.

OEEPE partners representing both digital mapping/GIS and academic interests (see Appendices) undertook the work, the objective being to produce a range of individual approaches to an agreed set of key updating issues. Results from the work would then serve as a catalyst for necessary software development. Project objectives and key issues are set out in Sections 4 & 5.

An overall timescale of one year was set for project completion. The project has followed the agreed schedule with CERCO Working Group IX, also involved in digital update, being included in an end of Phase I workshop.

In general terms Phase I indicated that both the relational model approach and the object oriented approach merited further practical examination and this was followed through in Phase II.

The key issues were all successfully addressed with the project focusing the thinking of a number of leading edge organizations on a problem that few have tackled or even considered. Ideas were shared and potential working systems devised.

The main problem areas for the future were seen as:

- handling the time/historical dimension
- development of user friendly, cost effective procedures
- production of software to produce and accept change-only information
- data compatibility.

All require attention by software developers and users must ensure that their demands are recognized.

Although no major breakthroughs were achieved the project can be judged a success in that it has brought these important needs into sharper focus as a basis for on-going research and development.

3 INTRODUCTION

The November 1991 OEEPE Steering Committee meeting charged Commission 1 with producing proposals for a project addressing the problems of updating complex digital topographic databases. Despite increasing use of spatially referenced information generally there was little indication that this fundamental requirement was receiving other than scant attention from software developers.

From population and use of its large scale, fully structured, digital topographic database, Ordnance Survey of Northern Ireland (OSNI) was familiar with the problems posed by updating. Functional update procedures had been developed in-house, based on the existing software, but a fresh approach was necessary for the future. Importantly the issue was not unique to OSNI.

This background enabled a draft updating project to be presented to the May 1992 meeting of OEEPE Commission 1 by the President, Mr M J D Brand. This visualized a collaborative, multi-national project pooling the expertise of various OEEPE members in a detailed examination of the problems inherent in the maintenance of such databases. At Commission 1 request further details, to supplement the draft proposal were provided to members and an initial meeting was convened at OSNI to consider the proposal further.

3.1 Initial Meeting

This was held in Belfast on 21/22 September 1992. (See Appendix I for list of attendees).

While supportive of the proposal, the meeting expressed the following concerns:

- resource limitations given the one year duration
- the constraints of differing computer systems
- unfamiliarity with the complex 1:1250 OSNI data
- the potentially competitive nature of the project.

However concerns were overcome and a draft two phase action plan agreed.

Phase I – examination of the problems and production of ideas (ie a "thinking" phase).

Phase II – practical validation of Phase 1 theories using a common OSNI large-scale digital data set.

A four month Phase I, five month Phase II, followed by a three month period for final report compilation was provisionally accepted.

Representatives were shown both the OSNI system and the large-scale data. The nature of the problems to be addressed was demonstrated and full documentation to support this demonstration was provided.

The meeting concluded with all being requested to consider the draft proposal, and indicate, in principle, their willingness or otherwise to participate.

3.2 Final Proposal

Positive feedback from the initial meeting enabled the final proposal to be prepared by OSNI and presented by President OEEPE Commission 1 for approval of the Scientific & Steering Committee at their 27–29 October 1992 meeting in Brussels. (See Appendix II for content of the final proposal).

Approval was granted for official commencement on 1st January 1993.

3.3 Project Actions

Phase I commenced as planned and concluded with a Workshop at which each participant submitted and presented a report.

The usefulness of this Workshop was enhanced through inclusion of CERCO Working Group IX, also involved in digital updating.

Phase II proceeded with a reduced complement, due mainly to resource problems. Nearing the end of this phase some overrun was deemed desirable in the light of developments and completion was put back to 31 December 1993. This delayed project report completion but resulted in a more positive outcome.

3.4 Final Report Structure

Due to the nature of the project, the range of participants and the resultant diversity of material generated, this report has largely been organized chronologically. The main body covers the initiation and conduct of the project, outlining the main events and summarizing results. The full content of individual reports is incorporated in the appendices. The main project elements are listed in the Contents and where appendices relate this is indicated in the text.

The final report should be regarded as a record of the work carried out, with findings and recommendations, rather than a definitive treatise on updating. As such it forms the basis for further more detailed research.

4 PROJECT OBJECTIVES

Two main objectives were established:

(i) To examine the issues involved in updating a complex digital topographic database, using as a basis the key areas identified at 5 and a common digital dataset drawn from the OSNI large-scale digital topographic database.

(ii) To present findings as a collective report for publication through OEEPE.

5 KEY ISSUES

Seven aspects of update were identified as key issues.

5.1 How an updated, fully-structured and attributed digital graphic, conforming to the full OSNI database specification and free of map sheet-line constraints, can best be arrived at?

5.2 How existing textual records linked to the graphic should be modified, and how the detail dealt with in the course of updating should be tagged with date of addition/modification/suppression, held, maintained and presented?

5.3 How output of the revised topographic data, both digital and hard copy, should be handled in terms of replacement map sheet and much more importantly, "change only" information?

5.4 How archiving of topographic information no longer extant should be dealt with?

5.5 Assuming the requirement to hold such data, inclusive of all attributes, date of suppression, etc., as a layer within the topographic database, how should this be met?

5.6 How such archived data can be interrogated/output for specific historical points and/or periods?

5.7 How a continuing policy of such updating and archiving can be satisfactorily operated and how administration of the resultant database can be kept manageable, given a policy of continuous revision in line with ground change occurring?

6 OSNI TOPOGRAPHIC DATABASE

As the project areas to be addressed were identified within the OSNI general purpose, topographic database, and as the common digital dataset was drawn from it, it is appropriate to provide a brief synopsis of the large-scale element of this expanding database. Typical 1:1250 data is shown at Appendix III.

The database structure is complex, being designed to act as the geographical "hub" of the developing Northern Ireland Geographic Information System – NIGIS. This is a growing multi-partner initiative to develop a common infrastructure to radically improve information sharing, chiefly across the Northern Ireland public sector. Development is in the form of a distributed network of individually controlled digital databases each holding spatially referenced information of a specific partner organization.

The large scales element of the OSNI database has some 190 different levels, is of links and nodes form and is capable of handling points, line strings and polygons. Geometry, while held once only, is multi-attributed and all map sheets are fully edge-matched with the objective of providing homogeneous cover for all of Northern Ireland. (See Appendix IV for database structure).

Textual information is linked as associated data to selected points, vectors and polygons, the link being a system generated, Irish grid reference, representing the X and Y co-ordinates of these objects to one metre positional accuracy on the ground. Polygon integrity in particular is fundamental to current and future NIGIS user applications (eg roads, buildings, properties, etc) and considerable effort is required at updating to ensure that this is preserved. One typical example of this feature is linking of postal addresses to specific properties, a factor of obvious importance to NIGIS.

7 OSNI DATABASE UPDATE METHODOLOGY

Basic System : DEC/Sysdeco

OBJECTIVES

OSNI digital updating objectives are in direct response to database user needs. These range from the basic in-house requirement to maintain the currency of the survey archive, to the GIS demands of an increasing user base. The latter in particular requires:

- current information
- an historical perspective
- "change only" information

Fundamental to the whole updating process is the need to ensure that all changes made are fully compatible with the database specification so that key features such as polygon integrity/multi-attribution of the graphic and associated textual records are not degraded over successive updates.

The methodology devised by OSNI is an attempt to accommodate these requirements within the constraints of the existing digital mapping/database system. While it meets the first of the above requirements satisfactorily the latter two can only be achieved in a cumbersome manner with "change only" information supply in particular being a significant problem.

However the existing updating system is straightforward in its approach and lends itself to illustrating the need for improved methods of handling the key issues listed in Section 5. As such a brief outline is of benefit to this project.

INITIAL UPDATE STAGE

Taken in its simplest form the updating process begins with the current ground information being supplied to the digital updating group on a plastic survey document. This document is accompanied by a paper copy which clearly identifies the changes that have occurred since the last survey or update. New and obsolete survey information is highlighted on the paper copy in different colours to assist the operator to quickly and accurately identify change.

UPDATING PROCEDURES

STEP 1

A copy of the relevant mapsheet is extracted from the database and is then assigned with

(a) The date of present update.

(b) The date of the original digital capture.

The appropriate data files are then created for the new and obsolete textual and graphic information:

1. New addresses
2. New vegetation polygons
3. Linestrings
4. Deleted addresses
5. Deleted vegetation polygons.

The hierarchical structure of the database file is then checked to ensure that it is compatible with the latest specification and amended if necessary.

STEP 2

The hierarchy of the file is then temporarily changed to accommodate the additional levels of "Suppressed", "Structure Change" (feature code changes) and "Retained". During the graphic editing process all detail which has been highlighted for deletion is identified in the file and the level "Suppressed" added to it, and similarly, all detail with feature code change has the level "Structure Change" added to it.

At this stage for ease of identification the operator alters the on-screen colours for the "Suppressed" and "Structure Change" geometry.

STEP 3

"Structure Change" identifiers are normally created automatically, by a systems procedure, for all linestrings, and the date of change is recorded in the respective data files.

STEP 4

At this stage of graphic editing a new file is created and from this two copies are made to provide:

1. Interim Deletion File
2. Remainder File

The "Suppressed" and "Structure Change" information is transferred from the Interim Deletion File to create the Final Deletion File. Each element of "suppressed" geometry will have an identifier created which automatically records a date of deletion, and the "Structure Change" geometry will be linked to a single identifier recording the date of change. To update the textual data in the Deletions File all geo-references are selected and recorded with a date of deletion. On completion of graphic editing operations the file will be returned to normal database hierarchy with the removal of the additional levels of "Suppressed", "Structure Change" and "Retained.

The Remainder File, which contains all unchanged geometry, is created by deleting all "suppressed" geometry and returning the file hierarchy to normal database format.

STEP 5

The next procedure is to create an empty file into which all the new geometry will be digitised. New unique identifiers will be generated automatically for all relevant addressable properties and polygons and these will be stored in the respective data files awaiting input of textual data. Following this, identifiers will also be generated for all newly digitised geometry and a date of survey attached to the respective data files.

A file has now been created containing new geometry which can be merged with the Remainder file containing the unchanged geometry. This produces a new updated version which will be checked during graphic editing to ensure that all the amendments have been attended to, all feature code levels are correct and all polygons closed. The new or amended textual data is recorded in the relevant data files for each addressable property or vegetation polygon.

On completion of the necessary quality control procedures the updating process is completed by returning the graphic and textual files to the database ready for the next update.

The overall process can be represented as in Figure 1.

In general, while update of 1:1250 information presents significantly more detailed problems than might normally be expected with the smaller scales topographic databases more commonly encountered, nevertheless the same broad issues apply. Where GIS features large, then similar areas demand the same high degree of attention. The nature of data may differ but polygons must still be closed and appropriate linear features must still be linked if the topographic data is to successfully act as the GIS information carrier.

OSNI Digital Update Flowline

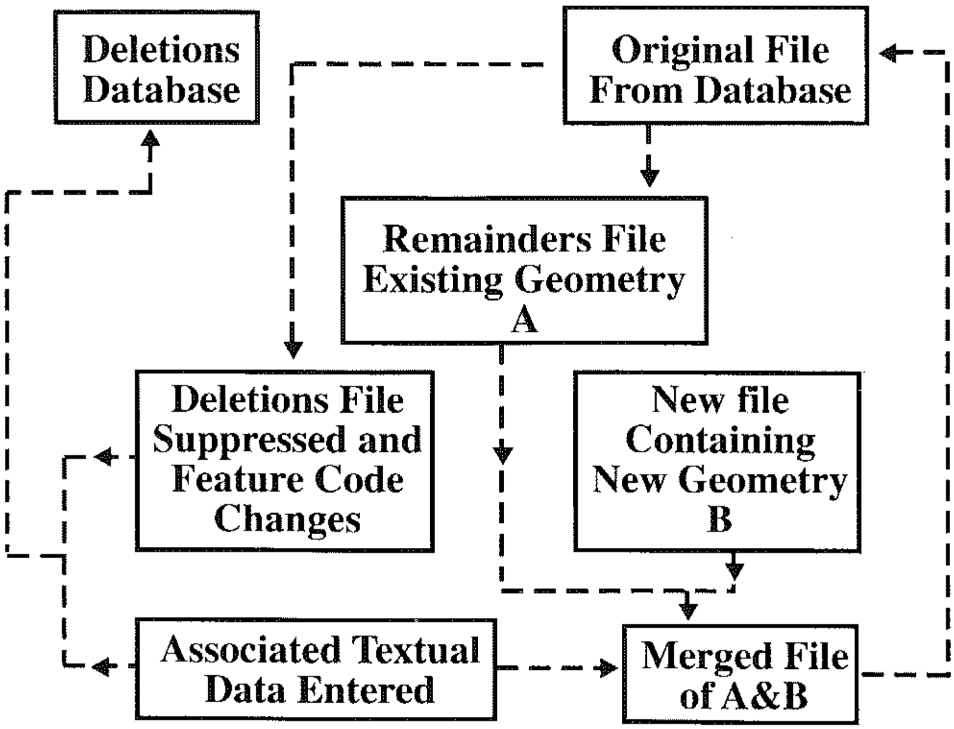


Figure 1

SUMMARY

This broadly summarises what is in effect a very complex operation requiring a high level of competency and concentration on the part of the operator. The OSNI large scale (1:1250, 1:2500) database is complex. There are therefore a large number of aspects to be dealt with in any updating session and a consequent penalty to be paid in respect of effort required to maintain its currency. Work is carried out at Digital VAX Station 3100's with a linked interactive digitising facility using Sysdeco software.

The system outlined involves a mix of interactive and batch processes and is kept under continual review in the light of experience and developing technology in order to refine the process and reduce the turnaround time between completion of the survey and the latest information being databased. At present a turnaround within seven days from take up is normal due to the extent of batch processing necessary for efficient system management. By increasing the interactive proportion this can be reduced considerably.

This very much emphasises the importance which must be attached to the whole maintenance mechanism in any GIS database. The more complex the database the greater the resultant updating commitment. While in the case of OSNI, updating policy is in response to identified NIGIS requirements, we are acutely aware that the effort involved in achieving the desired currency, particularly in a rapidly changing urban area such as Belfast, is approximately the equivalent of initial digital conversion.

A particularly relevant maintenance factor is the inclusion, as associated textual data, of the date and nature of survey for each successive update. OSNI policy is that at initial conversion a common date is applied to the total content of a map sheet (due to the fact that at that time there is no historical record) and as successive suppressions and additions are carried out the actual data involved has date and nature of survey linked through an automatically created 12 figure, Irish Grid geo-reference, to each geometric element or object. At this comparatively early stage in maintaining the topographic database this policy does not create any significant storage or database management problem, but as the digital conversion programme proceeds and the database grows, then constraints may appear. This reinforces the need, in a constantly changing scenario of user demand and technological development, for an organisation to continually monitor the performance required and what tools are available, in order to ensure that an unmanageable situation is not created.

The associated textual element is an important feature and one which has tremendous application within the developing NIGIS. At present only a small proportion of the potential application is being addressed in this way, (eg linking of postal addresses to building polygons, etc) and work is in hand to enhance the NIGIS potential of the data by, for example, linking road names and classification numbers to road centrelines, and a plethora of other similar functions is obvious. All this is ultimately of benefit and sensible for the future use of NIGIS, but in turn adds yet again to the maintenance overhead. There is no doubt that such enhancements will be desirable in any complex GIS database as it develops and new applications are identified, but the knock-on maintenance effect should never be under estimated.

8 ACTIONS TO COMMENCE PROJECT

As already noted, participants who attended the September 1992 meeting in Belfast had the opportunity to see the identified problem areas demonstrated on the OSNI system. However they were not shown how OSNI handled the updating issues thus ensuring that all came to Phase 1 without pre-conceived ideas influenced by the OSNI approach.

All participants were provided by OSNI with a standard set of documentation and system outputs indicating the problem areas, and this formed the basis for Phase 1 activity. The difficulties of communicating the issues became evident later in Phase 1 when new participants who had not had the benefit of screen demonstration and were thus forced to rely solely on the documentation, required considerable further clarification.

The common documentation has not been included in this report but copies can be made available from OSNI.

A further issue, commenced at this stage and essential to Phase II, also proved a considerable problem and reinforced the need for multi-national standards for digital data transfer. This was supply of the standard OSNI dataset in formats which the range of partner systems' could accommodate. Although eventually resolved it required much greater effort on OSNI's part than anticipated and is obviously an area where users should be pressurizing all bodies and suppliers concerned for standards and appropriate translation softwares.

9 PHASE I

This phase was formally commenced in January 1993 with an enhanced number of participants. All those not present at the initial (September 1992) meeting in Belfast were issued with copies of the documentation relating to the trial similar to that provided earlier to attendees at the original meeting. A full list of Phase I participants is at Appendix V.

All were asked to provide details of the data format requirements of their systems to enable the common OSNI digital data set to be loaded to each for Phase II work. Initial actions regarding data translation were put in hand.

Using the common information and explanatory documentation issued by OSNI, each participant sought individually to address the project areas on the basis of examination of the objectives outlined in the final project document.

Work on this phase proceeded with little interruption or further joint contact apart from the explanatory issues mentioned earlier. With the benefit of hindsight more contact would perhaps have been desirable as nearing the conclusion of this phase it became apparent that some misconceptions regarding the objectives still existed. Not surprising perhaps given the multi-national involvement, but nevertheless indicative of the latent communication problems which can exist.

In spite of this sufficient progress was evident to warrant the convening of an end of Phase I workshop in Belfast on 24/25 May 1993. Prior to this each participant prepared an interim report. These individual findings were collated by OSNI and where possible, issued to all prior to the workshop date.

10 END OF PHASE 1 WORKSHOP – PROCEEDINGS AND CONCLUSIONS

In view of the commonality of interests between this project and the activities of CERCO (Committee of the Heads of European National Mapping Organisations) Working Group IX on updating of digital mapping, this workshop took the form of a combined meeting, thus greatly enhancing the range of interest and expertise present. A list of workshop attendees is included at Appendix VI.

Each project participant gave an illustrated presentation of their Phase 1 theories to supplement intermediate reports and these acted as the basis for subsequent discussions. The diversity of these still reflected to some extent the misconceptions referred to earlier but nevertheless provoked extensive discussion.

Full documentation of the workshop presentations has been omitted in the interests of brevity but copies are available from OSNI if required.

The main debate resulting from the presentations centred on the difficulties encountered in trying to produce a single generic description of updating and the limitations imposed by the conceptual model of the system employed to manipulate the data. It was concluded that the former may not necessarily be an essential output from the project and that, in the case of the latter given the tight timescales, existing tools should be used in providing solutions, rather than trying to create a conceptual model which could not be implemented.

The feasibility of dividing the group into units and tasking each with addressing specific issues was considered, but rejected as impracticable as the areas are related and different approaches would have led to confusion.

Two main themes emerged from this phase:

- handling update using a relational model
- handling update using an object oriented approach.

The former has been around for some considerable time and tends, wrongly perhaps, to be thought of as outdated. It does however provide for rapid searches and is kind to disk space. While it enables relationships between data elements it usually does this by reference to graphic elements with links being made to the textual matter. The OSNI system is a typical example.

The latter offers a newer approach by treating the whole rather than linking parts. Using this the totality of information on a particular feature, for example a dwelling, would form one object – geometry, nature of feature, postal address, type of survey, date of survey, amendment, suppression, deletion, etc. The objects can in turn be part of other objects or classes of objects.

Following much deliberation on the options a reduced team agreed to go forward with Phase II using both the approaches outlined above to address the project objectives.

CONCLUSIONS AT END OF PHASE I

1. Both the relational model approach and the object oriented approach were valid and offered possibilities which should be examined further.
2. While conceptual updating models capable of implementation were necessary for Phase II of the project other alternatives should also be recorded.
3. Problems encountered with supply of the common digital dataset in formats appropriate to the range of participant systems reinforced the need for, ideally, all to be able to handle data in a neutral format.
4. Difficulties in retaining the integrity of the complex OSNI dataset, particularly in respect of multi-attributed features, when it was moved between systems emphasizes a weakness which some system suppliers need to address for the future.

5. Resources available to participants were insufficient to allow all to proceed to Phase II. It was judged impractical to divide the key issues across the group to mitigate this and hence, regrettably, only a reduced team was able to continue with Phase II. The problems to be addressed were evident and regardless of Phase II outcome properly resourced further work would almost certainly be needed.

11 PHASE II

Based on the outcome of the Workshop a smaller team of participants (see Appendix VII) proceeded to Phase II. After clearing of some outstanding data transfer problems the common OSNI topographical dataset was successfully loaded on their respective systems.

Both relational and object oriented approaches were adopted and summarised versions of the individual teams' activities are included, under their respective titles, in parts 12–16 of this report. Full content of the various Phase II reports is incorporated in Appendices VIII to XII.

12 SUMMARY OF FINAL REPORT

Department of Computer Science,
Queen's University, Belfast, Northern Ireland

Basic System : IBM compatible PC

OSNI's complex large-scale topographic database can be characterised by the information it can provide namely:

- The storage and retrieval of detailed accurate geometrical data.
- Geometry structured with codes to indicate the nature of the features represented.
- Features represented with associated textual information describing non-graphical properties.

This paper proposes a solution based on an object-oriented (OO) data model.

It also shows how the OO model is especially suitable for topographic databases in general.

How is the updating carried out? First we need to examine some basic principles, with particular emphasis on OO.

The basic concept behind these databases is that the data are structured in terms of the real world objects being modelled. Therefore, each object of interest in the real world is represented directly by a single object in the database.

However these objects include more than the real world object, they will have methods or operations available to create, delete, manipulate and output this data. For example the dwelling house at 194 Bangor Road, Holywood has one object in the database which contains all information relevant to this property.

The feature (DWELL_HOUSE), 194 Bangor Road contains, for example, the fields making up the postal address (house number, street and city names etc.) and links to its geometry. The geometry in turn contains lines (composed of individual straight segments), symbology, text (in this case "194") and/or arcs, points etc.

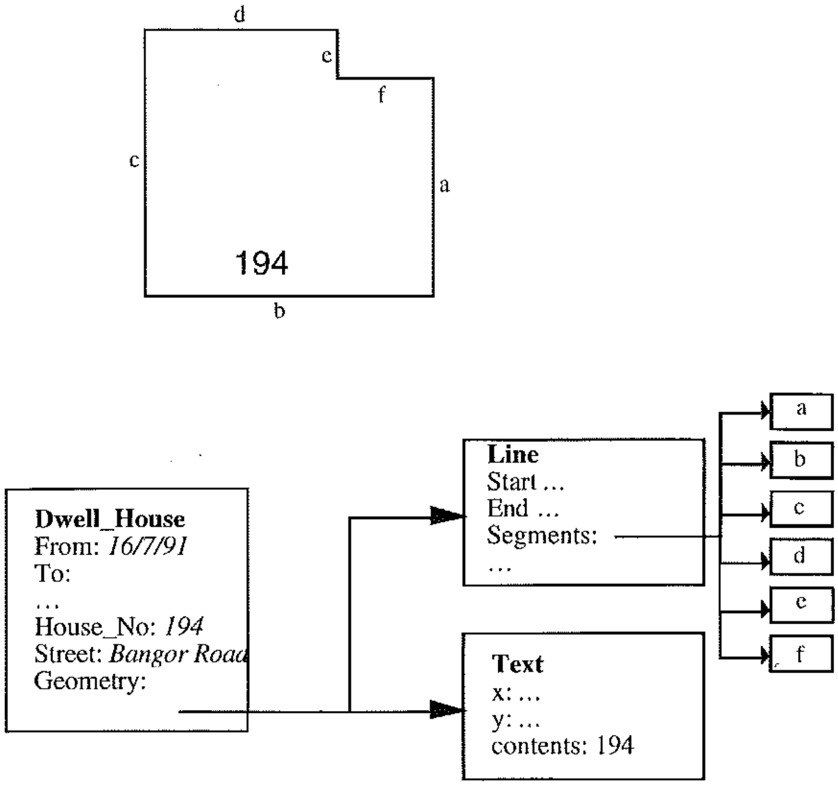


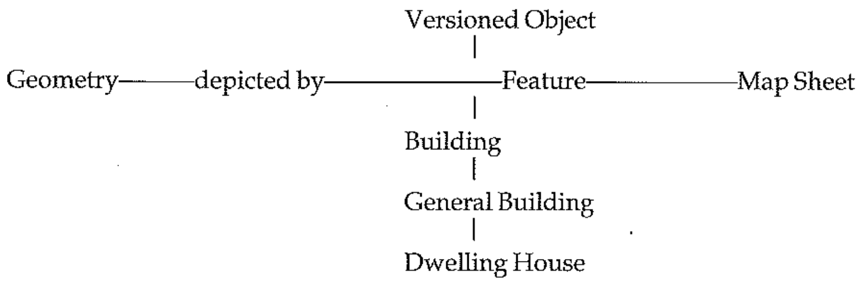
Figure 2 – Representation of a house in an OO topographic database

Objects are organised into homogeneous classes of similar real world objects. 194 Bangor Road would belong to the class DWELL_HOUSE which may inherit properties from its parent class, for example :

GENERAL_BUILDING
|
DWELL_HOUSE

This sub_class / super_class structure builds into a hierarchy similar to that of OSNI (a few alterations are necessary!).

Some OO-DBMS provide built in versioning capabilities for objects but if this is not the case, the proposal is to create a 'super' class called "versioned object" at the top of the structure, containing properties such as "Current From", "Current To" (dates). As this class sits at the top of the structure, those below can inherit these features.



Geometry and Feature are connected so that a particular map object is connected to the geometry objects that depict it including the lines showing its outline, associated features and its house number. The dividing wall between semi-detached houses would belong to two (dwelling house) objects.

The database is designed to be seamless so the link to 'map sheet' can be ignored.

During each change (update) the 'current from' date is added (automatically if required) to each new object and the 'current to' date to any deleted or changed object. By default only current detail is displayed. Through an operation called 'previous' earlier versions can be accessed, displayed or printed.

If an object is altered in any way, then first a copy is automatically made. This new version becomes the default object, although the older version is still available (it is marked with the 'current to' date). This process can be repeated many times and as we are dealing with complete objects, the data base management system maintains existing and generates new links to the geometry elements, therefore there is no duplication of the geometry. Changes to structure are inherent within this system.

Operations such as "Get Previous Version" and "Get Version At Date" can access older versions of the object. By applying these operations to all features a complete map-sheet for a particular date or change-only information for a particular period may be generated.

A pilot system is being developed on an IBM compatible PC, using the OO database system POET, which provides database extensions to the standard OO programming language C++.

Three separate programs will interact with the database:

- Conversion program to construct the database from a text file generated from the original OSNI database file.
- Editing – to permit revisions to the map and store them as new versions of objects.
- Display – to demonstrate time dependent queries.

This paper has proposed an object-oriented method for modelling changes to topographic databases. It improves on the data structure of the current OSNI model by maintaining all features as individual objects while providing a simple versioning mechanism for storing the history of a topographic object. It satisfies the original requirements for this project as described in the initiation document.

13 SUMMARY OF FINAL REPORT

Research Group
University of Pavia, Pavia, Italy

Basic system : PC based

This solution replaces the operations of 'addition', 'modification' and 'suppression', as described by OSNI, with 'addition' and 'disactivation'.

Addition refers to the introduction of something new into the database while disactivation refers to an instance of entity no longer existing; tagged as no longer extant in the real world, but still held within the database.

Both addition and disactivation can be 'simple' or 'complex' operations. Simple refers to instances of change where no other items in the database are effected by this change. Complex operations are changes that have an effect on the surrounding geometry, each operation is given a unique database number, this being necessary for future manipulations.

As geometry is entered into the database it is tagged with the date of introduction. Changed or deleted geometry is date stamped and continues to be held within the current database. Current data has a layer value ranging from 1 to 99; for detail no longer extant the layer value is increased by 100.

In order to assume time as the fourth dimension within the database all entities must be tagged with addition and disactivation dates, as required, and for this reason these dates are stored within the graphic files.

Because everything is tagged with the necessary date information, the file can be displayed at any point in time. As default views only display current data, introducing the concept of the 'fictitious current date' permits the operator to enter the date of their choice resulting in the data being displayed current at that time.

The database is structured on three types of file:

(a) Textual

- used to store the textual information of each of the entities, each class being stored in a different file. Links are made by means of an identification code. OSNI data has two classes having textual records; buildings and vegetation, and these are held in two DBIII files. These DBIII records are structured according to the group of attributes associated with these classes in the OSNI data.

(b) Graphic

- two types are used: description(D) and co-ordinates(C).

  The description file contains information such as colour, addition and disactivation dates, 'complex' number and various pointers to DBIII files and the co-ordinate file.

  The co-ordinate file contains the point's x, y and z co-ordinates and a pen up or down code. This file is used to output the geometry.

  Besides the description and co-ordinate files, a text file is also available but for reasons of simplicity was not created – names of rivers, roads etc, could be stored within this file.

(c) Graphic – DBIII link

– to link objects in the description file with the correct DBIII file in which textual data is stored.

The basic software used to carry out the project originates from existing software produced to manage digital cartography, some of the main features are :

- data capture
- data display, via a number of 'filters'
- editing
- updating
- data validation

These are performed by means of more than 200 dedicated menu driven functions. The source code is the 'C' language.

The hardware used is PC based (386 or 486) with both text and graphic (1280 X 1024 pixel resolution) monitors, matrox card 1281, mouse and digitiser – a cost effective solution.

The following modifications to the existing software have been carried out:

- the management of the addition and disactivation dates and the 'complex' operation number fields in the description file.
- a two way link to the DBIII files.

The following new features have been added:

- introduction of the addition/disactivation date and of the complex operation number in the dedicated fields.
- introduction of a different display representation for existing and disactivated geometry. Existing is drawn by a solid line and disactivated represented by a pecked line.
- a 'layer' filter, giving the possibility of displaying only current detail, disactivated or both.
- 'change only' filter, selecting detail between dates.
- 'complex operation filter' selecting detail involved in a complex addition or disactivation.
- 'fictitious current date' function giving the possibility of looking at the territory as it was in the past. Therefore detail having an addition date after the fictitious date will be considered as not existing. Detail having an addition date preceding the fictitious date and a disactivation date subsequent to the fictitious current date will be considered as still existing.
- complex operation group function : the entering of a complex operation number will result in that alone being drawn on the screen.
- DBIII query filter; only detail resulting from a DBIII query will be displayed.

The first six of the seven key areas are answered in the above text. Key area seven does not pose any particular problem in a production environment and indeed offers many advantages, for example it capitalises on cost effective hardware and software and lends itself to the setting up of many workstations to keep pace with continuous revision.

14 SUMMARY OF FINAL REPORT

National Geografisch Instituut (IGN) Belgium

Basic System : Intergraph

Due to time constraints and problems with the translator there was only a limited amount of practical work carried out for phase two of this project.

However, an adapted version of the proposal for phase one was created for testing. Some interesting differences between the OSNI and IGN concepts were brought to light. For example, consider a building in the middle of a garden/parcel; when the building is extended the garden gets smaller in the IGN model but in the OSNI model the parcel still retains the same area!

The data model used in the National Geographical Institute uses one database in which various tables, text and graphic files are held.

Geometry is mainly captured by stereoplotting, while attribute values are collected by field survey. The database structure allows geometry and attribute values to be updated at a different frequency. For each feature category (buildings etc.) four different kinds of vector files are used: a file for present geometry, one for history and a working file for each. For each (simple) feature type there is a geo-table and one or more tables with geometry independent attributes; the geo-table is permanent whereas the attribute tables are replaced when updating. Links to text and raster graphic files are also possible through attributes containing the full path.

If we consider the house/garden scenario, objects are created for the two area centroids and both geometry dependent and independent attributes are attached to them, resulting in at least two tables for each object.

When topology is being built, the area boundaries (and geometry) are linked to the area centroids meaning that a geometry change requires a local rebuilding of the topology.

The seven key areas as defined in the initiation document are answered by applying them to the IGN conceptual data model:

PROCEDURES FOR UPDATING

The data within the conceptual data model are not seamless but tiled. When geometry is altered we would normally store new, then delete old geometry.

All geometry versions of a house for example, are given the same unique identification and in this way a house becomes a complex object. For new detail a date of survey is added. Once completed the new working file is merged into a buildings_actual drawing file.

Deleted building's geometry is gathered into a working set from the buildings_actual drawing file and stored in a deletions file. This is eventually merged into a buildings_history file, the date of deletion is also completed in the relevant tables.

Geometry independent attributes are periodically captured in the field and the results stored as follows:

- unload the database tables.
- archive this ASCII dump.
- give the attribute tables a new name, for example HOUSE94 rather than HOUSE93 etc.
- load these into the database.
- delete all lines where the date of deletion was filled out the year before.
- perform the modifications.
- new objects have added the date of survey and are treated as for modified objects.
- modified objects will be given the date of change.
- deleted objects will be given the date of deletion.

Text files similar to those supplied by OSNI for dwellings and vegetation are considered too regular and are loaded into attribute tables and treated as above.

OUTPUT OF CHANGE ONLY SYSTEM

Dates used are in ISO8601 format. The supply is achieved on a table search by feature category basis. New geometry is in the current file whereas deleted detail is contained in the ***_hist file, but as all contain the relevant dates change only information is possible.

ARCHIVING OF HISTORICAL INFORMATION

All historical graphic information is kept in separate .dgn files. Except when needed for special applications these files only have to be loaded for updating purposes.

The database records which are directly linked to historical geometry are held within the single geo-table for every feature, whereas the real attribute tables are separately archived for each revision.

Interrogation of historical data does not pose any real problem as all the information is date stamped.

The concept described was designed for periodic revisions although it could be adapted for continuous revision. As more frequent versions would imply frequent archiving of the same unchanged data, perhaps the database should be archived once a year and modified data should be archived with a greater frequency.

THE APPLICATION OF THE IGN CONCEPTUAL MODEL ON THE OSNI DATA.

As previously stated, a number of problems arose during the translation process resulting in loss of links to both text and design files, it is also fair to say that we do not know how much extra functionality was lost during the translation.

One design file was received for each feature category resulting in a duplication of geometry in many instances. A more satisfactory solution is to hold vector data once and keep the structure in attribute tables.

On investigating the detail, it appears that OSNI geometry is fragmented. This can be avoided and the topology built automatically by changing to a structure that respects the principle of always choosing the highest possible dimension, for example:

- a point is not a point object if it belongs to a line,
- a line is not a line object if it belongs to an area etc...

We suggest that OSNI consider changing from line objects to area objects, having the centroids tagged with their functions(levels). This has the effect of reducing the number of objects 'active' at any given time.

Most of the OSNI data can easily be transformed into the conceptual data model that we have described.

In conclusion, it is possible to perform updating using the software at IGN although it is not really constructed for updating purposes; it should be improved with some real, user friendly updating functions.

15 SUMMARY OF FINAL REPORT

National Landsurvey of Sweden (NLS), Gavle, Sweden

Basic System: ARC/INFO

This is a four part entry to the updating project based on NLS experiences in developing the Geodatabank (GDB), being part of the NLS general software called Autoka. Parts one and two discuss generally:

- different geographic information systems.
- the importance of GIS conceptual models.
- features the ideal GIS might have, including how we may communicate with this system.
- complexity of structures and the operations that act on these and methods by which this complexity may be reduced.
- the various interfaces the ideal system can have in order to permit the most efficient use of human and computer resources.
- historical data and the possible fragmentation of the geometry.
- the storage of archival data, using new technology.

Part three is very much a practical exercise, manipulating the data as supplied by OSNI.

If we consider a dwelling house which has changes at the first update due to the addition of a glass building. The common geometry connecting the dwelling house to the glass building has been changed with respect to it having an additional level. The original structure was dwell_h and non_squ, the structure now is dwell_h,non_squ and glass_b. These codes are stored in tables containing the user_id and the feat_code, having a one-to-many relationship with no practical limits to how many codes an object can have.

The necessary commands are collected in 'macros' and form menus for more efficient updating operations. Dates of change are also held as a related table so that the historic viewpoint can be recorded. Each modification to an object will be treated in the same way.

Deletions to geometry also carry a date stamp and are held in a separate 'cover' or layer.

This paper concludes:

- most of the commands (probably all) needed to digitise, update, delete, multi-code and date handle objects such as lines, points, areas and text already exist in most established systems.
- in area handling it is important to have full topologic data structure to reap the benefits of analysis and control.
- the 'system' may also provide a macro language in order to produce user friendly menus for each application.

Although several software packages are used for handling topographic databases at NLS, Arc/Info (version 6) was chosen for this project; it is our opinion that it would have been impossible to use version 5. Version 6 introduces the 'cursor' concept; well known in SQL language.

With the help of the macro programming language (AML) and the cursor facility a small application program was implemented for updating the OSNI digital data, taking about three days. This application does not cover every issue although it is estimated to complete all the update cases would take about two weeks.

However, two items would still be missing, handling of historic data (versions) and change only updates. The latter could be introduced by an appropriate selection procedure, adding more time for the implementation. Historic data can be stored automatically in a separate layer but there is no easy method to handle time dependent versions of objects.

The data can be held in either a seamless or tiled system, depending on the user.

The prototype application could be further refined to create a production flowline and we estimate this would take a further two months.

Both textual and date information are handled using the macro language, although automatic tagging of date information has not been implemented.

Both hard copy and file export in a number of formats are possible.

No standard procedure for the export of change only information exists at present. In our opinion this feature should be built into the original system.

Traditionally the archival subsystem is separate from the main database resulting in archive data having to be handled differently; often manual routines are required.

NLS is presently investigating new optical mass storage technology and will use this in conjunction with the GDB. The advantage with this type of hardware being you don't have to treat historical data differently from current data: this mass storage is akin to an indefinite extension of conventional disks.

Conclusions

- it would appear that it is quite possible to use Arc/Info to perform the required updates – by writing some extensions in the macro language.
- archival storage problems could be solved by using new mass storage technology.
- to get beyond the limits of today's systems we have to investigate new conceptual models. However, there are many different opinions as to what the ideal conceptual model should look like!
- as far as possible the system must automatically handle the operators work to increase productivity. The new technology of object orientation seems to be moving in the right direction.

16 SUMMARY OF FINAL REPORT

General Command of Mapping, Ankara, Turkey

Basic System : Intergraph

Not all key areas as defined in the project initiation document have been answered due to limitations of the software used, although the exercise has given valuable experience in updating complex topographic databases.

KEY AREA NUMBER 1

The software used is Intergraph MGE and Informix is used as the DBMS. A special project was created along with related descriptive information and in order to have default relational database tables a project schema was also created.

A graphic data file (.dgn) was created for each feature category. Polygon feature classes are generated for dwelling houses and vegetation, including their related tables containing the necessary attributes.

The OSNI textual data files were loaded into the database using SQL statements. Using the MGE command "feature maker", OSNI graphic data was also loaded into the database.

The textual data was linked manually to each cartographic object containing the necessary centroid value.

KEY AREA NUMBER 2

The term 'updating' is assumed as addition, modification or deletion. To update the graphic data the OSNI map sheets are scanned via an Optronix 5040 raster scanner, followed by 'heads-up' digitising on screen of the changed information.

When new graphic data are entered the related textual data are also input. Modified graphic data also infers a change to the textual information and when geometry is deleted so also is the related textual data. If the graphical data is unchanged but changes to the textual are necessary, this can also be performed.

Both graphic and textual information can be printed and displayed.

KEY AREAS 3 TO 7

Due to limitations in the software it was not possible for us to answer these key areas.

17 CONCLUSIONS

Fully structured and attributed digital data from the OSNI topographic database was used for Phase II, although some project participants used only a small subset of this in order to help simplify matters. In these cases a scaling up of procedures and macros would be necessary to cater for the complete structure.

Some participants had initial difficulties in translating the necessary information into their systems and for at least one the problem wasn't fully resolved.

Whether the approach was an object oriented or a relational one, from the documented results the major areas of difficulty would appear three fold:

- how best to add the fourth dimension (time) to the database information ?
- assuming dates of change can be added, is the database management system able to manipulate the information based on time relevant queries?
- how should the suppressed data be archived?

From this it can be seen that the critical element is whether or not the database management system is capable of dealing with the fourth dimension. If the dates of addition and suppression are successfully incorporated with both graphical and textual elements and the DBMS is capable of utilising these dates then all the main project criteria can be satisfied.

This being the case then it is possible to provide the following to users:

- current information, being a straight copy of today's data
- 'change only' information, being only data marked as new or altered since the last version was supplied
- a 'snapshot' of the database at any given moment in time
- time sequenced extracts, similar to the above although they are taken over a period of time.

The solutions offered include the date stamp as the fourth dimension,whether added automatically or manually.There were a number of ways this feature was included in the database. For example, in one case the date 'field' was inherited from a 'super class' in the database hierarchy while others simply included the date as another related table.

Geometry changes were carried out in two main ways:

- make a complete copy of the geometry and its related tables, add the date of change to the original and modify the copy to agree with the current state.
- mark the related tables with the date of change and include a new table entry showing the current position – geometry is held only once.

Both work equally well although duplication of geometry may not be desirable in the long term.

As previously reported not all DBMS permit the addition of the fourth dimension as a standard function, although this situation is alleviated somewhat by the addition of macro functions and other utilities, permitting the addition of in-house 'workarounds'. This infers a thorough understanding of the various commands etc. available with these systems and could not be undertaken by a casual user.

It became apparent, from some of the papers, that considerable thought had been given to the computer/operator interface and how this may effect throughput of updates. For example, 'pull down' menus afford a greater degree of flexibility than does command line entry. If the operator selects an object on screen the system should present the relevant 'form' including the necessary default values, thus reducing input times. Automatic error checking should also be available thus reducing the possibility of mistakes by the operator during input.

Entry of changed or suppressed data has been mentioned only briefly but it would appear that this information can be stored by three main methods:

- held directly on line with the current data, although by default, only the current situation will be displayed.
- held indirectly, via the necessary tables to deletions layers or to files under the control of the DBMS, called only when necessary
- held on large optical storage devices, permitting rapid access to the user when required.

With regard to hardware, it was interesting to note that a number of the solutions offered were based on a PC platform, thus offering a cost effective route to updating, although this is typical of the ever narrowing gap in performance between PC's and the workstation environment.

The seven key areas:

1. "How an updated, fully structured and attributed digital graphic, conforming to the full OSNI database specification and free of map sheet line constraints, can be best arrived at."

   – a number of working solutions to this key area have been reported on, although at least one system operates on a tile by tile basis. However PC based systems may have difficulty in coping with the data volumes on a country-wide basis.

2. "How existing textual records linked to the graphic should be modified and how the data dealt with in the course of updating should be tagged with date of addition/modification /suppression, held, maintained and presented."

   – a number of solutions to this area were recorded, in the case of the OO approach the textual records are included with the 'object' being updated. Another solution replaces the add/mod/supp with 'addition' and 'disactivation'.

3. "How output of the revised topographic data, both digital and hard copy, should be handled in terms of 'replacement map sheet' and much more importantly, 'change only' information'"

   – it was reported that the time dimension can be added to all the necessary information and if the DBMS can utilise this then the required data can be extracted in the form required by the user.

4. "How archiving of topographic information no longer extant should be dealt with."

   – a number of solutions to this were recorded; the crucial factor being the addition of the date of deletion to the necessary graphical and textual elements.

5. "Assuming the requirement to hold such data, inclusive of all attributes, date of suppression, etc, as a layer within the topographic database, how should this be met"

   – several solutions to this key area were reported. It is possible to hold this information on line along with the existing data (tagged with the date of deletion). It is also possible to hold this data in a separate coverage or layer, linked via necessary tables.

6. "How such archived data can be interrogated/output for specific historical points and/or periods"

   – it was reported that if the time dimension can be added to the detail and the DBMS can utilise this information, then this key area can be answered. Queries such as "display detail with date of survey >date1 and date of survey <date2" are possible.

7. "How a continuing policy of such updating and archiving can be satisfactorily operated and how administration of the resultant database can be kept manageable, given a policy of continuous revision in line with ground change occurring."

   – this key area was answered in a number of papers. It was thought that the resultant data volume would be no greater whether updates were carried out, say, annually or continuously, the DBMS would record the necessary dates.

18 RECOMMENDATIONS

Although this project has succeeded in:

– bringing to light a problem that, perhaps, only a few organisations have tackled, or even considered.

– focusing the ideas and thoughts of a number of leading edge organisations in this area.

– sharing of ideas and an understanding of how this problem can be dealt with from different viewpoints.

– producing a number of potential working systems

a number of issues still remain.

These are identified as follows and the project recommends that the key areas which software development must concentrate on are:

1. Recording the time/historical dimension

   Not all GIS/DBMS cater directly for the fourth dimension and it is important that users, and user groups such as this, press this issue with software vendors so that their DBMS will have the necessary functionality. At least one of the project participants is currently doing this with their system supplier.

2. Production of user friendly/cost effective updating procedures

   The updating operations must be made as straightforward as the original data entry , with the DBMS logging and time-stamping all changes to the database and that the operator can achieve this in the most cost effective manner.

3. Production of software which can not only produce, but also accept, change only data

   These issues are important even for users who are not interested in the historical perspective, it may be more economical for them to receive change only information rather than complete tiles, for example. Therefore the end user's software must be able to utilise some of the features discussed in this report.

4. Data compatibility

   A number of data translation problems were apparent at the start of this project and it is important that these difficulties can be overcome by increased software house acceptance of GIS standards.

19. GENERAL COMMENTS ON PROJECT CONCLUSIONS BY PHASE II PARTICIPANTS

Following presentation of the draft report to the OEEPE Steering Committee in May 1994 it was recommended that general comments by participants would be a further worthwhile inclusion in the Final Report.

These are as follows:

Adam C. Winstanley, Queens University, Belfast BT7 1QP, Northern Ireland

This appraisal of the OEEPE project is divided into three sections starting with the overall results and then our own contribution. We then conclude with some suggestions of future directions.

1. The project

   The project involved diverse participants from many different countries across Europe using a variety of different GIS software systems based on relational, object-oriented and priority models. As such, diverse solutions to the problem were formulated, often drawing on the broad experience of the participants in using GIS systems professionally.

However, a number of drawbacks can be identified which hamper the usefulness of the results and mean real-world practical issues have not been addressed:

- Lack of resources for the project in terms of manpower and equipment.
- No participation from the GIS software development industry.
- The diversity of the approaches taken and solutions proposed, while being a strength, also means that any conclusions to be drawn from the project are not suitably cohesive or definitive.

2. Queen's University

Queen's entered the project at the beginning of Phase II, and consequently missed the valuable preliminary meetings and discussions and the Phase I workshop. Approaching the problem from a software engineering viewpoint, we decided to concentrate on basic models of time rather than adapting proprietary software as most other participants proposed. As such our system to input and display temporal topographic data is currently limited to a small demonstration running on a relatively slow personal computer. While necessarily restricted in application, we believe this has demonstrated the feasibility of our general conceptual model and emphasised the importance of the user-interface design in making the updating process fast and easy. We intend to continue our work by enhancing our system to address the current drawbacks by:

- Porting the implementation to a more powerful machine (probably a SUN workstation)
- Experimenting with the user interface in order to improve the updating procedure.

3. Conclusion and future development

The project was useful for experimenting with solutions but, of course with the limited resources available, it was never going to provide a fully workable solution to the problem. For this to come about, the following points would have to be addressed:

- problems due to the size and complexity of large-scale real-world datasets
- involvements by GIS software development industry
- external funding to provide resources.

Funding for a larger project from national governmental sources (such as one of the Research Councils in the UK) may be feasible. The pan-European nature of the participants, and of the OEEPE itself, would also possibly make it feasible to obtain support from one of the European Union's research funds (such as ESPRIT).

Prof. *Riccardo Galetto*, University of Pavia, Italy

We can identify many positive elements in the outcome of this project.

(a) From the methodological point of view we underline these aspects:

- efficient co-ordination which allowed the project to be contained within the predicted time span
- quick circulation of the papers documenting the activities carried out by the participants
- immediate presentation of the results of the finished work at an international conference (Athens – Georgia)
- utilization, by all of the participants, of the same data for the research project
- involvement of the CERCO Group IX in the research.

(b) Regarding the contents, the most revealing aspects are:

- in general, the attention placed on a problem concerning large-scale digital mapping. In fact, large-scale digital mapping still presents many unresolved aspects regarding its content; the way in which it best matches the cadastral maps; the degree of detail necessary to satisfy the needs of the land information systems, etc
- highlighting the fact that updating is a problem which does not yet have a definite solution and that it is therefore necessary to carry on with the research on this subject
- emphasizing the advantages of employing and running the time parameter in the GIS, which allows a dynamic vision of the evolution of the territory.
- motivating the participants to find new solutions concerning the data structure and more specifically, the investigation of solutions based on the object oriented theory.

(c) Conclusions

We believe that there could have been a more complete participation in the second phase of the project, which was more experimental and consequently crucial in verifying the theories created in the first phase.

Furthermore, we think that a final discussion session is necessary to analyze the results in depth and to evaluate and critique the solutions adopted by each participant.

Overall, our feeling is that the project was very successful and that it achieved the goals which were set in the beginning. Nonetheless, we consider this project to be unconcluded as yet and we think that the research must be continued to better develop the proposed solutions. Only in this way can we succeed in comparing the various methods on a practical level.

Jan Beyen, National Geographical Institute, Brussels, Belgium

This project was very useful because it forced us to find solutions to the updating problems at an early stage, the project coming right in time to influence our data structure. It also provided the opportunity to let (GIS software) vendors know what we expect, so that they might adapt their software by the time when updating will be our most important problem. I didn't succeed in giving a detailed evaluation of MGE within the deadline of the project, but my paper for ISPRS Commission IV (Athens; volume 30 part 4 pp 884–888) may also be considered as a final report to or a continuation of the OEEPE project.

Kjell Degerstedt, National Landsurvey of Sweden, Gavle, Sweden

The findings of the OEEPE project addressing the above problem are of central and general interest for people concerned with updating of digital topographic databases because of the well known fact that the dominant part of the cost of building a working GIS is in data collection and updating.

The updating activity is in many contemporary systems very much dependent on the operators skill and decisions. The human factor is the main source of most of the data errors. The most cost effective way to rationalise the updating process is therefore to find better ways for the system to support the operator in the complex updating operations.

Progress in this direction is being made by all the main vendors in the GIS market. Incremental steps forward happen every day, but to get a more significant improvement it is perhaps necessary to look in new directions like searching for new data models and new paradigms. This is of course a complex problem demanding a large amount of personnel resource.

Unfortunately it has not been possible for the NLS to allocate enough resources to give a major contribution to the solution of the updating problem. However we have tried to convey our opinions about the topics based on earlier experiences of both a theoretical and empirical nature. We have also shown by updating the OSNI data, that extending a system possessing suitable features will make a real difference. To sum up we believe that the project has been successful in describing and pointing out important aspects of updating complex topographic data, which can be the starting point for further research.

Hayati Tastan (Capt. MSc.), General Command of Mapping, Ankara, Turkey.

As an active participant in the project, we believe that this project has contributed considerably to the activities in the GIS field. A digital geographic database which is not up-to-date is nothing but a modelled view of the real world at a definite time. A GIS database is supposed to be dynamic, ie up-to-date.

As for archiving of the historical geographic information in a digital database environment, the fact is that neither all geographic information users nor all the applications need this capability. On the other hand, if these historical data are needed and if they don't exist any longer it's water under the bridge. Taking this factor into consideration, it can be said that historical information should be archived when creating a GIS database in case any tiny possibility for making use of it is foreseen. Otherwise there is no point in wasting time and data storage space.

Another key area needing further discussion is the smallest unit used for updating of digital information. This unit should be a real world object (eg a house) but not an element of it (eg a wall of the house) since the descriptive data are linked to the object but not to any element of this object.

As a final technical remark it is believed that it would be useful if the results of the project could be presented as an appendix to the final report in some recognized statistical way such as line/bar graphs and/or pie charts which would cover the project as a whole and give the lead to future studies on this topic.

20 FINAL REMARKS

The project proved to be a most useful exercise in that a wide range of mapping and academic expertise was involved in using typical complex digital topographic data to examine an issue which, for various reasons, would appear to have received scant in-depth attention to date.

While no radical new theories or methodology have emerged what is particularly evident is that the process can be greatly simplified and speeded up given a properly facilitated approach. Here users must ensure that their demands are heard and influence the course of software development accordingly. This report and its appendices certainly highlight the significant areas for attention, and while not all will require to maintain data of this complexity the continuing expansion in use of geographic information will guarantee that the general principles must certainly apply.

As already indicated in the introduction the findings of this report can form the basis for more, properly resourced, research and development.

Only through keeping this update issue to the forefront can organizations be sure that their costly database investment of the 20th Century be safeguarded to service their demands in the 21st.

APPENDICES

Appendix I	Initial Meeting (21/22 September 1992) Attendees	
Appendix II	Agreed Project Proposal Document	
Appendix III	Typical OSNI 1 : 1250 data as used in the Project	
Appendix IV	OSNI Database Structure	
Appendix V	Participants Phase I	
Appendix VI	End of Phase I Workshop Attendees	
Appendix VII	Participants Phase II	
Appendix VIII	End of Phase II Report	– Queen's University, Belfast Northern Ireland
Appendix IX	End of Phase II Report	– University of Pavia, Pavia, Italy
Appendix X	End of Phase II Report	– National Geografisch Instituut, Brussels, Belgium
Appendix XI	End of Phase II Report	– National Land Survey, Gavle, Sweden
Appendix XII	End of Phase II Report	– General Command of Mapping, Ankara, Turkey

MAINTENANCE OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

INITIAL MEETING 21–22 SEPTEMBER 1992 AT ORDNANCE SURVEY OF NORTHERN IRELAND, COLBY HOUSE, BELFAST

Northern Ireland	– Mr *Michael Brand* OEEPE President Commission I Ordnance Survey of Northern Ireland
Germany	– Prof *Fritz Ackermann* OEEPE President
	– Dr *Wigand Weber* Institut für Angewandte Geodäsie
Sweden	– Mr *Kjell Degerstedt* Lantmteriet (National Land Survey)
Great Britain	– Mr *Keith Murray* Ordnance Survey of Great Britain
Italy	– Prof *Riccardo Galetto* University of Pavia
The Netherlands	– Ir *Auke Hoekstra* Dienst van het Kadaster en de Openbare Registers
Norway	– Mr *Odd Eriksen* Statens Kartverk (Norwegian Mapping Authority)
	Mr *Kasten Lein* Statens Kartverk (Norwegian Mapping Authority)
Northern Ireland	– Mr *Maurice Mulvenna* University of Ulster
	– Mr *Sam Gray* – Mr *Gerry Mitchell* – Mr *Raymond Clements* – Mr *Geoff Mahood* – Mr *Martin McVeigh*
	Ordnance Survey of Northern Ireland

TO: OEEPE SCIENTIFIC AND STEERING COMMITTEE

27TH – 29TH OCTOBER 1992 BRUSSELS

FINAL PROJECT PROPOSAL FOR APPROVAL

1. INTRODUCTION

The rapid development of geographic information systems (GIS) and the growing range of application areas identified, reinforces the need for increasingly sophisticated and up to date digital topographic databases.

The essential maintenance of such databases is an aspect which, to date, has received little or no attention.

This project seeks to address the issue in depth and support from a range of organisations has already been established.

OEEPE approval to proceed as outlined in the following agreed plan is hereby requested.

2. PROJECT TITLE

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES.

3. PROJECT OBJECTIVE

To examine the problems inherent in updating of complex digital topographic databases and identify and recommend effective procedures to ensure that database currency and maximum functionality is maintained. To present the findings as a co-ordinated report, published under the auspices of OEEPE, for the benefit of the GIS community in general, and as an indicator to software development houses of GIS demands which must be addressed.

4. PROJECT METHODOLOGY

The project will address the issues in depth through individual teams working, largely independently to a common objective and employing their own ideas and systems. No constraints will be imposed in respect of methods employed but each must produce a final report to an agreed common format.

5. PROJECT PARTICIPANTS

Ordnance Survey of Northern Ireland	NORTHERN IRELAND
Lantmateriet (National Land Survey)	SWEDEN
University of Pavia	ITALY
Ordnance Survey	GREAT BRITAIN

Dienst van het Kadaster en Openbare Registers (Cadastre and Public Registers Agency)	THE NETHERLANDS
Niedersächsisches Landesverwaltungsamt (Federal State Survey)	GERMANY
Statens Kartverk (Norwegian Mapping Authority)	NORWAY
University of Ulster	NORTHERN IRELAND

The possibility of an involvement from IGN FRANCE and Ministry of National Defence, TURKEY also exists.

6. ACTIONS TO DATE

At the Berne meeting of 7/8 November 1991 OEEPE Commission I was charged with production of proposals for a project concerned with updating of digital topographic databases. Resulting from this Mr *M J D Brand*, President Commission I, presented a proposal at meeting No 80 in Gavle, Sweden, on 13–15 May 1992. At the request of the Steering Committee further information was prepared and issued to relevant organisations, together with an invitation to participate.

A meeting of interested participants was duly held at Ordnance Survey of Northern Ireland on 21 and 22 September 1992, and following demonstrations and identification of the type of problems involved, using the OSNI database, agreement was reached on how the project could be handled within an acceptable timescale. The agreed proposal was subsequently circulated and the necessary support from the above listed participants confirmed.

7. SUMMARY PROJECT PLAN

The project will be handled in two consecutive phases:

Phase I

Each participating organisation will individually consider the updating issues as identified and demonstrated at OSNI, together with the principles underlying them, and produce proposals on various means of updating the topographic database as presented, in the form of a textual report. A workshop will be convened at the end of this Phase and results reviewed.

Phase II

Having formulated ideas and reached conclusions, organisations involved will, as far as possible, attempt to test and validate their individual theories on their own systems using a common dataset from the OSNI large scales topographic database. Results will be fully documented.

At the conclusion of these two phases OSNI will, with assistance as necessary from others, draw the individual findings together into a final report with conclusions and recommendations, for presentation to OEEPE.

8. PROJECT TIMETABLE

Phase I	– four months, commencing from date of OEEPE approval to proceed.
Phase II	– five months, following completion of Phase I.
Preparation of final report	– three months, from completion of Phase II
Total project duration	– 1 year.

9. PROJECT FUNDING

9.1 Costs of manpower and equipment within each participating organisation will be met by that organisation.

9.2 Travel and subsistence in respect of the end of Phase I workshop and any other full meetings necessary, will be the responsibility of each organisation concerned.

9.3 OEEPE funding is requested for any necessary travel and subsistence incurred by project co-ordinator staff in the course of the project.

9.4 OEEPE funding is requested to cover costs of magnetic media, documentation, postage/fax/telephones, incurred in the project.

OEEPE funding of £ 5,000 is requested to cover the issues at 9.3 and 9.4 above.

10. DETAILED PROJECT PLAN

Phase I

Using the information supplied by OSNI, derived from experiences to date with digital updating of the OSNI large scales database, and having seen the issues demonstrated on the OSNI system, each participant will individually examine the problem areas and the updating principles giving rise to them. This will take the form of an in-depth study and proposals will be produced on various methods of updating the topographic database as presented.

Further information will be provided by OSNI if and where necessary in the course of the study.

The key areas for examination and proposals will be:

- how an updated, fully structured and attributed digital graphic, conforming to the full OSNI database specification and free of map sheet line constraints, can best be arrived at.
- how existing textual records linked to the graphic should be modified and how the data dealt with in the course of updating should be tagged with date of addition/modification/suppression, held, maintained and presented.
- how output of the revised topographic data, both digital and hard copy, should be handled in terms of "replacement map sheet" and much more importantly, "change only" information.

- how archiving of topographic information no longer extant should be dealt with.
- assuming the requirement to hold such data, inclusive of all attributes, date of suppression, etc, as a layer within the topographic database, how should this be met.
- how such archived data can be interrogated/output for specific historical points and/or periods.
- how a continuing policy of such updating and archiving can be satisfactorily operated and how administration of the resultant database can be kept manageable, given a policy of continuous revision in line with ground change occurring.

At the end of this Phase each organisation will produce a report to a common format based upon the key areas outlined above. These will be the subject of discussion at an interim project workshop to be held at the close of this Phase.

All ideas and theories will be included in this report, even if only considered and rejected, as they may well prompt fresh lines of investigation by others.

Phase II

Having formulated ideas and reached conclusions on how the required actions should be carried out, and with the benefit of the workshop discussions, each organisation will attempt to test and validate, as far as possible, their individual results using their own equipment and systems.

To enable this OSNI will provide each with a common digital dataset of appropriate complexity from the large-scale topographic database. This will be supplied in a data format and on media which will, directly or indirectly, render it capable of reading into their respective systems. OSNI personnel will be available, on a limited basis, to visit a participant site if problems in relation to data reading or interpretation which cannot be otherwise resolved, are encountered.

Full statistical details derived from conduct of these practical trials, in respect of processing times, manpower and equipment will be recorded to enable cross system and procedural correlation. Statistics in relation to rejected options and processes will be recorded and untested alternatives detailed.

Results from these tests will be documented and included in the initial Phase I report.

11. POSITION AT END OF PHASES I AND II

Completion of the nine month period allowed for these phases will see a full set of individual participant reports prepared to a common format, ready for use in the compilation of a final report.

12. FINAL REPORT

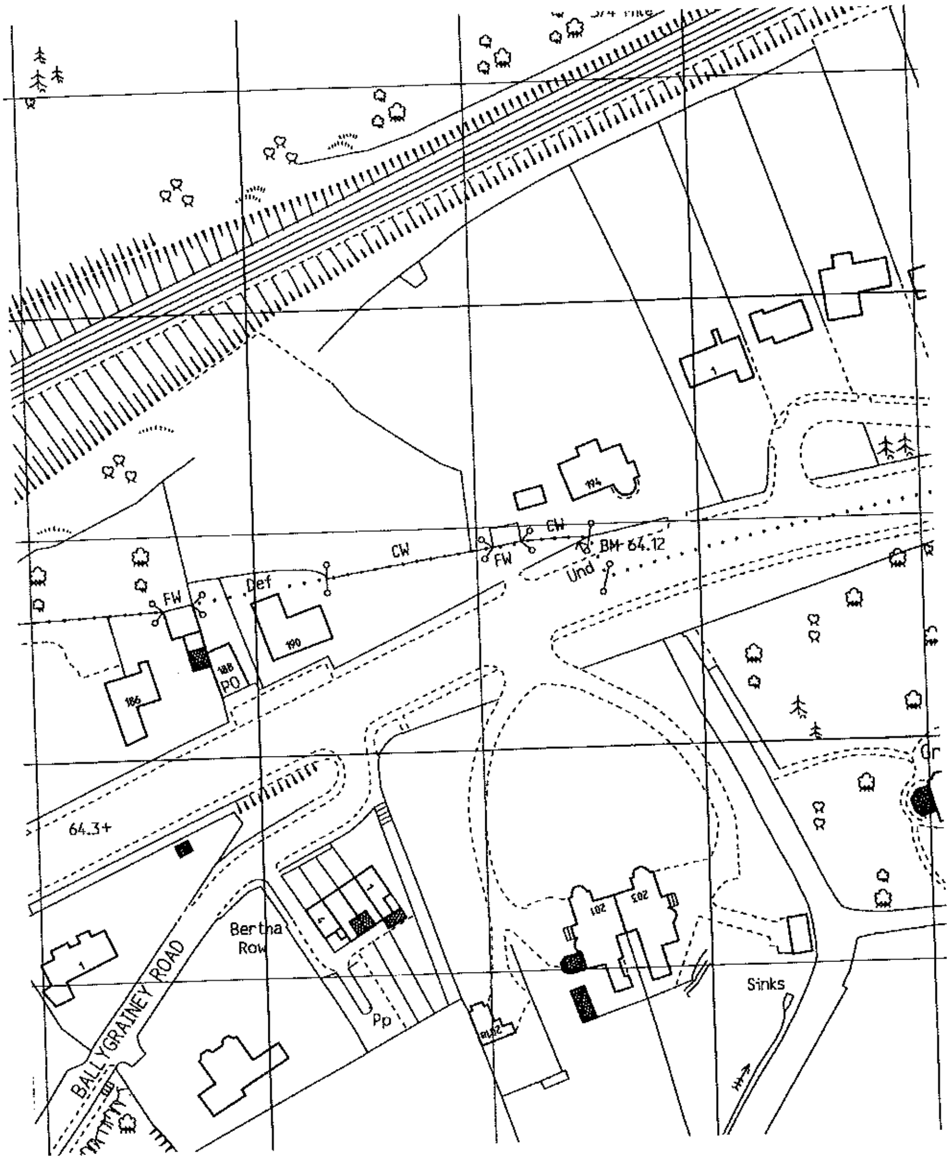
The remaining three months of the project will see collation of a final report document based on and drawing together the individual findings, complete with conclusions and recommendations.

OSNI will assume the overall responsibility for completion of the final report, with assistance from others as and when necessary.

As wide a range of options as possible will be incorporated in order to produce a report of real benefit. The resultant document will be of high quality, for presentation through OEEPE and eventual publication.

M J D BRAND BA FRICS
President OEEPE Commission I

TYPICAL OSNI 1:1250 DATA AS USED IN THE PROJECT



OSNI DATABASE STRUCTURE

```
MAINMASTER$1000
MASTER_OTH$200
MASTER_OS$100
    FRAMES$11
        CLIP$111
        LATCH$112
        FRAME_OTH$113
    FLAG$12
        FLATAREA$121
        AREAS$122
        FLAG_OTH$123
    AIRSURVEY$13
        UNSURE$131
        FIELD_TEXT$132
        AIRSUR_OTH$133
    TOPO_DATA$10
        ANTIQUITY$101
            ANTIQ_BLD$1011
            ANTIQ_FEAT$1012
            ANTIQ_SITE$1013
            ANTIQ_OTH$1014
        BUILDINGS$102
            COMMUNAL_B$1021
                LAW_ADMIN$10211
                HEALTH_B$10212
                EDUCATE_B$10213
                RELIGION_B$10214
                SERVICES_B$10215
                RECREAT_B$10216
                GOV_OFFICE$10217
                COMM_OTH$10218
            GENERAL_B$1022
                INDUSTRY_B$10221
                COMMERCE_B$10222
                DWELL_HOUSE_B$10224
                GENERAL_OT$10223
            BUILD_FURN$1023
                BLDFURN_OT$10231
            NON_SQU_B$1024
            GLASS_B$1025
            BUILD_LIN$1026
          — BUILD_OTH$1027
        COMMCATION$103
            COMM_AIR$1031
                AIRPORT$10311
                RUNWAY$10312
                AIRFURN$10313
                COMMAIR_OTH$10314
            COMM_RAIL$1032
                RAILWAY$10321
                RAIL_CNTR$10322
                RAIL_FURN$10323
                RAIL_OTH$10324
                RAIL_LINES$10325
                PERM_WAY$10326
            COMM_ROADS$1033
                MOTORWAY$10331
                DUAL_CARR$103313
                CLASS_A_RD$10332
                CLASS_B_RD$10333
                MINOR_RD$10334
                C_LINE_RD$10335
                TRACK$10336
                FOOTPATH$10337
                ROUNDABOUT$10338
                SIDEWALK$10339
                VERGE_RD$103310
                ROAD_FURN$103311
                ROAD_OTH$103312
            COMM_WATER$1034
                FERRY$10341
                FORD$10342
                STEP_STONES$10343
                CANAL$10344
                PORT_HARB$10345
                DOCKS$10346
                CHANNEL$10347
                WHARF_QUAY$10348
                WATER_OTH$10349
            COMM_OTHER$1035
        WATER_FEAT_WTR$104
            FRESH_WTR$1041
                RIVER$10411
                STREAM$10412
                DRAIN$10413
                CONDU_DUCT$10414
                C_LINE_W$10415
                WEIR_SLUIC$10416
                SOURCE_END$10417
                WELL$10418
                UNSHOWN_W$10419
                FRESH_OTH$10420
            TIDAL_WTR$1042
                HWMMT$10421
                LWMMT$10422
                TIDAL_OTH$10423
            WATER_FURN$1043
                WTRFN_OTH$10431
            W_FEAT_OTH$1044
        GEN_FEATUR$105
            ARTIF_FEAT$1051
                ARTIF_PNT$10511
                ARTIF_LINE$10512
                ARTIF_AREA$10513
                ARTIF_OTH$10514
            NATUR_FEAT$1052
                NATUR_PNT$10521
                NATUR_LINE$10522
                NATUR_AREA$10523
                NATUR_OTH$10524
            SLOPES$1053
                GRADIENTS$10531
                    GRA_UPPER$105311
                    GRA_LOWER$105312
                    GRA_CLOSE$105313
                GRA_GEOMS$10532
            GEN_F_OTH$1054
        TOPO_AREAS$106
            BOUNDARIES$1061
                LGD$10611
                WARD$10612
                PARL$10613
                COUNTY$10614
                CO_BOROUGH$10615
                TOWNLAND$10616
                BDY_LEFT$10618
                BDY_RIGHT$10619
                I_NATIONAL$10617
            EXTENTS$1062
                LAND_PAR$10621
                    PAR_PRINT$106211
                    PARCEL$106214
                    ISLAND$106212
                    LANDPC_OTH$106213
                ORNAMENT$10622
                WATER_PAR$10623
                    MOAT_LAKE$106231
                    RESERVOIR$106232
                    DAM_POND$106233
                    SEALOUGH$106234
                    BAY$106235
                    WATPC_OTH$106236
                VEGETATION$10624
                EXTENT_OTH$10625
            LIMIT_LINE$1063
            PERIMETER$1064
                FENCE$10641
                HEDGE$10642
                WALL$10643
                BANK$10644
                PECK_OTHER$10645
                PERIM_OTH$10646
            TOPO_A_OTH$1065
        CONTROL$107
            CONT_MAJOR$1071
                TRIG$10711
                BENCH_MARK$10712
                C_MAJ_OTH$10713
            CONT_MINOR$1072
                PLAN$10721
                    TRAV_STN$107211
                    REVIS_PNT$107212
                    AIR_PNT$107213
                    PLAN_OTH$107214
                HEIGHT$10722
                    SPOT_HT$107221
                    HEIGHT_PNT$107222
                    CONTOUR$107223
                    HEIGHT_OTH$107224
            CONT_OTHER$1073
        TEXT_INFO$108
            NAMES$1081
                DISTINCT_N$10811
                DESCRIPT_N$10812
                ANTIO_N$10813
                ROAD_NAME$10814
                ADMIN_NAME$10815
                NAMES_OTH$10816
            MEREINGS$1082
            HOUSE_NUMB$1083
            CONTROL_TE$1084
            TEXT_OTHER$1085
        SYMBOL_INF$109
            DUMBELLS$1091
            FLOW_DIR$1092
            ELEC_PYLON$1093
            SYMBOL_OTH$1094
        TOPO_OTH$110
    OTHER$14
    SPLINE$15
```

PHASE 1 – PARTICIPANTS

ITALY	–	*R Galetto* University of Pavia 1-27100 PAVIA
SWEDEN	–	*K Degerstedt* Lantmateriet (National Land Survey) S-80182 GAVLE
GERMANY	–	*P Creuzer* Niedersachsisches Landesverwaltungsamt D-30159 HANNOVER
	–	*A Illert* University of Bonn W-5300 BONN
GREAT BRITAIN	–	*D Sharman* Ordnance Survey of Great Britain SOUTHAMPTON SO9 4DH
SWITZERLAND	–	*O Kolbl* Ecole Polytechnique Federale de Lausanne CH-1015 LAUSANNE
TURKEY	–	*H Tastan* Ministry of National Defence TR-06100 ANKARA
FRANCE	–	*M C Combes* Institut Geographique National F-94160 ST MANDE
BELGIUM	–	*J Beyen* National Geografisch Instituut B-1050 BRUSSELS
NORTHERN IRELAND	–	*M Mulvenna* University of Ulster NEWTOWNABBEY BT37 0QB
	–	*R Clements* Ordnance Survey of Northern Ireland BELFAST BT9 5BJ

END OF PHASE I WORKSHOP (COMBINED MEETING WITH CERCO WG IX) 24 & 25 MAY 1994

AT ORDNANCE SURVEY OF NORTHERN IRELAND, COLBY HOUSE, STRANMILLIS COURT, BELFAST

PRESENT:	*M Brand*	President OEEPE Commission I	Northern Ireland
	V Zill	Bundesamt für Eich- und Vermessungswesen	Austria
	J Beyen	IGN	Belgium
	J Mousset	CERCO	Belgium
	M Combes	IGN	France
	A Jakobsson	National Survey Board	Finland
	P Creuzer	Federal State Survey	Germany
	M Endrullis	Institut für Angewandte Geodäsie	Germany
	A Illert	University of Bonn	Germany
	G Apagyi	Ministry of Agriculture	Hungary
	P Divenyi	Ministry of Agriculture	Hungary
	D Kirwan	Ordnance Survey (Ireland)	Ireland
	J Danaher	Ordnance Survey (Ireland)	Ireland
	A Arrighi	Instituto Geografice Militare Italiano	Italy
	R Galetto	University of Pavia	Italy
	J van der Linde	Topographfishe Dienst	Netherlands
	M Mulvenna	University of Ulster	Northern Ireland
	J Thompson	British Antartic Survey	Great Britain
	H Tastan	General Command of Mapping Ministry of National Defence	Turkey
	A Rodriques	IGN	Spain
	K Degerstedt	National Land Survey	Sweden
	P Lindberg	National Land Survey	Sweden
	C Eidenbenz	Federal Topographie Office	Switzerland
	O Kolbl	Institut de P'grammetrie University of Lausanne	Switzerland
	D Sharman	Ordnance Survey (Great Britain)	Great Britain
	A Wild	Intergraph	Great Britain
	B Hearn	Intergraph	Great Britain
	R Clements	Ordnance Survey (N Ireland)	Northern Ireland
	G Mitchell	Ordnance Survey (N Ireland)	Northern Ireland
	S Gray	Ordnance Survey (N Ireland)	Northern Ireland
	D Martin	Ordnance Survey (N Ireland)	Northern Ireland
	G Mahood	Ordnance Survey (N Ireland)	Northern Ireland

PHASE II – PARTICIPANTS

ITALY	–	*R Galetto* University of Pavia I-27100 PAVIA
SWEDEN	–	*K Degerstedt* Lantmateriet (National Land Survey) S-80182 GAVLE
TURKEY	–	*H Tastan* Ministry of National Defence TR-06100 ANKARA
BELGIUM	–	*J Beyen* National Geografisch Institut B-1050 BRUSSELS
NORTHERN IRELAND	–	*A C Winstanley* Queens University BELFAST BT7 1NN
	–	*R Clements* Ordnance Survey of Northern Ireland BELFAST BT9 5BJ

UPDATING COMPLEX DIGITAL TOPOGRAPHIC DATABASES AN OBJECT-ORIENTED SOLUTION

Final Report
1 February 1994

Adam C Winstanley
Department of Computer Science
Queen's University
BELFAST
BT7 1NN

1. INTRODUCTION

Complex large-scale topographic databases, such as that maintained by Ordnance Survey of Northern Ireland (OSNI), can be characterised as follows. They provide:

- The storage and retrieval of detailed accurate geometrical data
- Geometry structured with codes to indicate the nature of the features represented
- Features with associated textual information describing non-graphical properties

Currently topographical database management systems concentrate on support for the capture, maintenance and presentation of data representing an up-to-date view of the landscape. However, one of the additional tasks required by users of the data is the ability to reconstruct historical detail and to identify changes made over time. These issues have been largely ignored by the developers of commercial systems so far and have only recently been the subject of sustained academic interest (eg, as in [*Langran* 1993]).

OSNI have identified 7 key areas that need to be addressed by an updateable topographic database [OSNI 1992]:

1. It should provide an up-to-date, fully structured and attributed graphical representation.
2. Dates of the addition, modification and suppression of data should be held, maintained and presented.
3. Output of the revised data should be available in terms of replacement map sheet and change only information.
4. An archive should be maintained of topographic data no longer extant.
5. Archived data may be held as a layer in the database.
6. Archived data should be accessible for specific historical points and/or periods.
7. A continuous system of such updating and archiving should be operated and administered.

This paper proposes a solution to the problem of the continuous revision of large-scale topographical databases and the maintenance of a historical archive based on an object-oriented (OO) data model. It also shows that the OO model has characteristics that make it suitable for topographical databases in general and describes an experimental implementation of the ideas expressed. Section 2 introduces the main concepts behind object-oriented databases and illustrates them using examples taken from the OSNI data model. Section 3 discusses the support OO Database Management Systems provide for managing change and storing time-dependent information. Based on this material, a proposal for a temporal topographic database design follows in section 4 which addresses the requirements identified by OSNI. Section 5 describes the experimental implementation being developed at Queen's University, concentrating particularly on the user interface design. Section 6 concludes the paper by summarising the implemented model and evaluating it against the 7 key areas identified by OSNI.

2. OBJECT-ORIENTED DATABASES

The basic concept behind object-oriented databases is that the data are structured in terms of the real-world objects being modelled. That is, each object of interest in the real world is represented directly by a single object in the database. For example, the house at 194 Bangor Road, Holywood, has one object in the database which groups together or contains all the relevant information about that property such as its full address, ground area, date surveyed and graphical depiction (or possibly alternative depictions for different scales). This one-to-one relationship between objects in the database and those in the real world greatly eases the task, the creation and use of the database.

There are a number of properties of object-oriented systems that make their constituent objects easy to create, use and manage:

(a) Encapsulation:

Objects include not only the properties (data) of the real world object but also the methods or operations available to create, manipulate and output that data. Examples of methods for *194 Bangor Road* might include *create, display* and *delete.*

(b) Classification:

Objects are organised into homogeneous groups or *classes* of objects which represent similar real-world objects and have a similar structure and set of methods. *194 Bangor Road* would belong to a class *Dwell_House* representing all the dwellings in the database. A class definition gives the properties and methods that objects of that class possess. Using a notation based on that used in *Hughes* [1991], we can define the following structure for *Dwell_House:*

class Dwell_House
    properties
        House_No: cardinal;
        Street: string; Townland: string;
        City: string;
        County: County Type; {Antrim, Armagh, Down ...}
        Postcode: string;

methods
    create
    display
    delete...{other relevant methods}
end Dwell_House

(c) Class Hierarchy:

The different classes of objects in the database are arranged in a logical hierarchy. A portion of the class hierarchy for a topographic database is shown in Figure 1. The class *Dwell_House* is a sub-class of the class *General_Building* which in turn has a sub-class *Building*. This relationship is often indicated by the label IS_A (eg a *Dwell_House* is a *General_Building*; is a *General_Building* a *Building*).

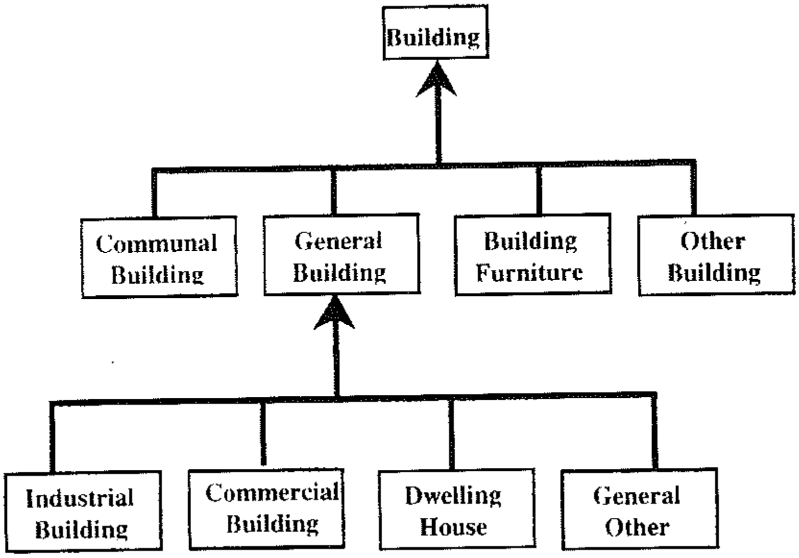


Figure 1 – Class hierarchy for buildings

(d) Inheritance:

Objects in classes lower down the hierarchy (*sub-classes*) inherit the properties and methods of those higher up (*super-classes*). Sub-classes can be seen as specialisations of their super-classes. They can have additional properties and methods only relevant to objects of the sub-class. Therefore properties relevant to all buildings, such as *area* or *geocode* or whether they have a glass roof, are declared for the class *Building* and inherited by all its sub-classes but properties only relevant to *Dwell_House* are declared in that class. In this way, the duplication of the data and methods for similar objects is avoided:

```
class Building
    properties
        Geocode:    string;
        Area:       real;
        NonSquare:  Boolean;
        Glass:      Boolean;
    methods
        ...{methods relevant to all buildings}
end Building,

class General_Building
    inherit Building
    properties
        ...{properties relevant to all general buildings}
    methods
        ...{methods relevant to all general buildings}
end General_Building,

class Dwell_House
    inherit General_Building
    properties
        House_No:   cardinal;
        Street:     string;
        Townland:   string;
        City:       string;
        County:     County Type;
        Postcode:   string;
    methods
        create
        display
        delete
        ...{other relevant methods}
end Dwell_House
```

(e) Extensibility:

The use of inheritance in the object-oriented data model facilitates the re-use and extension of existing objects. Re-use is through the formation of libraries of classes containing properties and methods for inclusion in a new program or database. New classes can extend those provided in class libraries. In this way, users of topographical data, such as telecommunications providers, can extend the class hierarchy to include the data relevant to them:

```
class HouseWithPhone
      inherit Dwell_House
      properties
            Householder:          string;
            Previous_Reading:     cardinal;
            Current_Reading:      cardinal;
            Last-Billed:          date;
      methods
            Create_Bill
            ...
end  HouseWithPhone
```

(f) Polymorphism:

In an object-oriented database, the same method name can mean different things to different classes of objects. For example, each individual class can have a separate method to *display* objects of that class with specific attributes. If a miscellaneous collection of different topographic features is assembled during the running of a program, they can each be sent the message to *display* themselves and each feature will display itself according to its own display method. In this way the software adapts automatically to the form of the objects selected.

(g) Composition:

Often an object is made up of several smaller objects collected together. For example, a line on a map is composed of a series of straight segments. Each segment is modelled as an object in its own right and each line object connects to the corresponding segments by a relationship labelled HAS_PART. Similarly each segment contains two points that represent its ends. Moreover the existence of these points depends on the existence of the appropriate segments which in turn depend on the line itself: the deletion of the line implies the deletion of the point.

Figure 2 shows a class hierarchy of geometrical objects suitable for a topographic database showing the relationship between lines, segments and points.

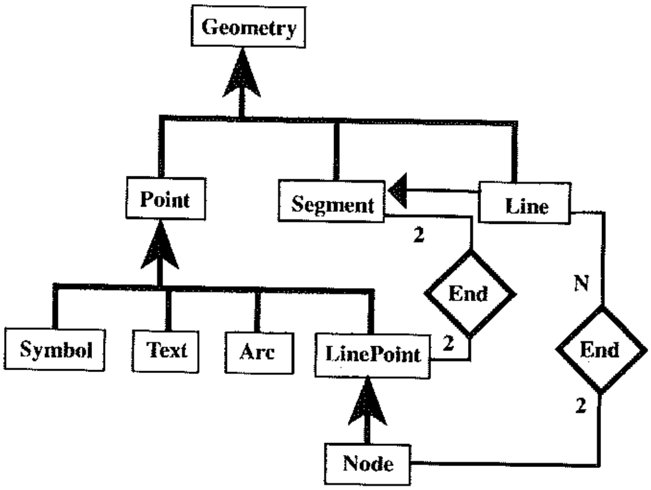


Figure 2 – Class hierarchy for geometry showing relationships.

(h) Aggregation:

Apart from composition, there are more general relationships between objects that need to be modelled. For example, the nodes or line-junctions are related to the individual lines. Each node can be the end-point of several lines but each line if related to only two nodes. Figure 2 shows the 2:Many node-line relationship. Using the notation introduced early, a data model for the node-line and point-line relationships can be specified as:

```
class Geometry
      properties
            colour:        colourtype;
            height:        real;
      methods
                           ...
end Geometry

class Point
      inherit Geometry
      properties
            x:             real;
            y:             real;
      methods
                           ...
end Point
```

```
class Line
      inherit Geometry
      properties
            start:          node;
            end:            node;
            segments:       sequence of Segment {internal segments are ordered}
            thickness:      cardinal;
            linetype:       cardinal;
      methods
                            ...
end  Line

class LinePoint
      inherit Point
      properties
            StartSegment:  Segment;
            EndSegment:    Segment;
      methods
                            ...
end  LinePoint

class Node
      inherit LinePoint
      properties
            lines:          set of Line;
      methods
                            ...
end  Node
```

Some OO database systems provide explicit support for inverse relationships such as that between *Line* and *Node*. When one side of an instance of the relationship is modified, the other automatically adjusts to maintain consistency. For example, if a line is deleted, the corresponding nodes will automatically update their properties to remove references to the deleted line. If the deleted line were the only one terminating at a particular node, the node itself would automatically be deleted as well.

This section has introduced various concepts used in object-oriented database design and illustrated them using topographic examples. OO databases, of course, can also support the more general facilities required of a database such as concurrency control, distribution, data integrity assurance, querying and error recovery. There are other advantages gained from using an object-oriented model for a topographical database:

- Knowledge-based techniques can be naturally mapped onto the object-oriented model. Many cartographic tasks such as object recognition, feature coding and map generalisation could be automated using such techniques.
- Graphical user interfaces, such as those required for topographic databases, are most easily modelled using objects. A one-to-one-one relationship between the real-world object, database object and user interface object should lead to easily created and maintained data and more intuitive interfaces.

3. UPDATING AND ARCHIVING OBJECT-ORIENTED DATABASES

Many application areas have to support multiple versions of objects. These may represent different copies of an object being modified by different people in large computer aided design (CAD) projects or alternative implementations of objects being evaluated to determine which satisfies certain criteria. Reviews of database versioning problems are presented in [*Snodgrass* and *Ahn* 1985] and [*Borhani* et al. 1992]. Recent papers specifically about GIS include [*Bofakos* et al. 1993], [*Kowalczyk* and *Kemp* 1993] and [*Wachowicz* and *Healey* 1993].

For this project we are interested in versions that store the evolution of a topographic object over time. *Hughes* [1991] has identified three requirements for versioned OO databases:

- Individual objects are versioned rather than classes
- Any object can have versions and there is no limit on the number of versions a particular object has
- Applications should be able to access either the current version (usually by default) or any of its older versions (which in general should be restricted to read-only access).

Some OODBMS provide built-in facilities for versioning. For example, ODE [*Agrawal* & *Gerani* 1989] provides an operation *newversion* that creates a new version (copy) of an existing object which can then be updated. This new version becomes the default object but the older version is still available through a call to a predefined operation *previous*. The operation *newversion* can be called many times for the same object and so a sequence of versions, linked through calls to *previous*, can be created over time to record the objects history.

When an OODBMS does not have in-built versioning, facilities can be programmed explicitly. A class of versioned object suitable for a topographic database can be created as follows:

```
class VersionedObject
      properties
            CurrentFrom:      date;
            CurrentTo:        date;
            Previous Version: VersionedObject;
      methods
            CreateNewVersion
            GetPreviousVersion
            GetVersionAtDate [returns the version current at a particular date]
end VersionedObject
```

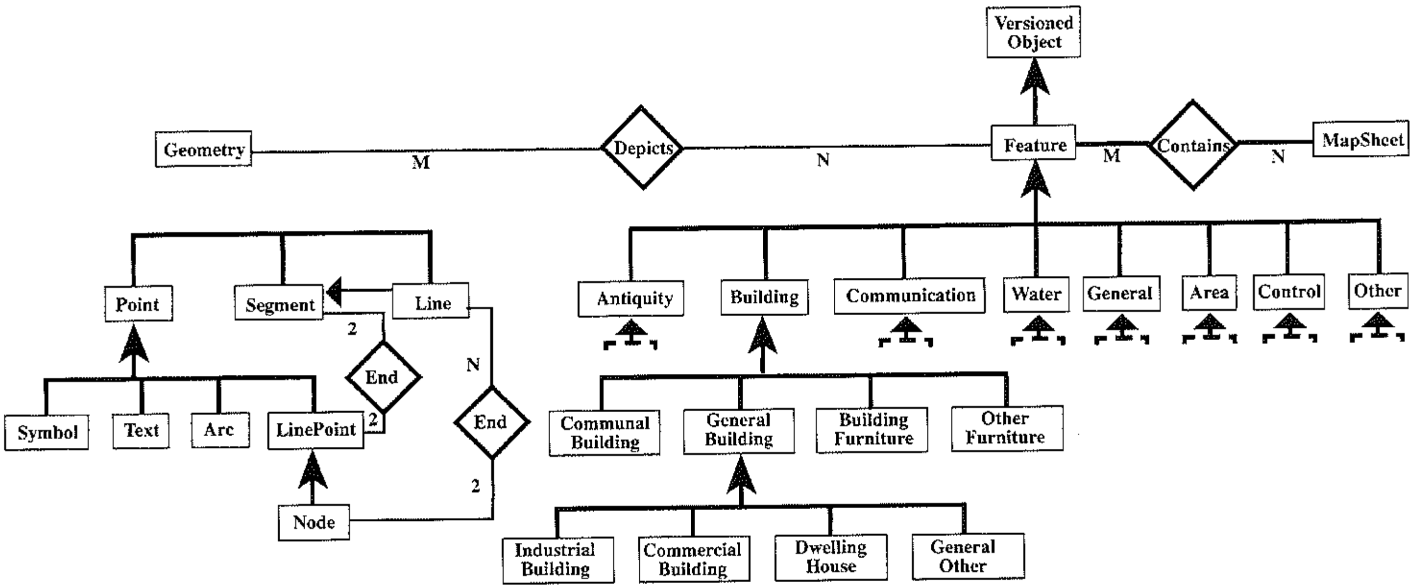


Figure 3 – Object-Oriented Topographic Database Design
(only Building sub-hierarchy is shown in full)

This class becomes the super-class of the topographic objects in the database which therefore inherit its facilities. If the dates to be recorded refer to the time when the database itself is updated (as opposed to the date when the change happened on the ground) then they can be recorded automatically by the database system.

4. A TEMPORAL OBJECT-ORIENTED TOPOGRAPHIC DATABASE

Based on the models for object-orientation and temporal data described in the previous sections, figure 3 presents a design for an object-oriented database to hold topographic data. The diagram incorporates the class hierarchy and the composite and aggregate relationships involved. There are three main classes of data: *Geometry, Feature* and *MapSheet.* To incorporate revisions to the database, the super-class of *Feature* is the *VersionedObject* described in section 3.

Geometry represents the graphical depiction of the map and has a sub-hierarchy to model the main geometrical object types: lines, arcs, symbols and text. The latter three are specialisations of the generic class *Point* which models a grid location. *Node* models the end-point or junction of Lines with which it has a 2:N relationship. *Line* also contains a composition of *Segments* representing its straight-line portions. Each segment shares its two end points with those of adjacent segments.

Feature represents the real-world objects depicted on the map. It's sub-classes reflect the object types of interest. Only the *Building* sub-classes are shown in full here but there will be similar sub-hierarchies under the other classes. Associated data are stored as properties in the appropriate classes. It can be seen that the *Feature* hierarchy is similar to the present OSNI hierarchical structure for feature codes. However, in this proposal these classes are used to model the actual features rather than for tagging geometry with feature codes. For this reason some of OSNI's categories are inappropriate as classes such as those tagging non-square and glass buildings. These instead become properties of the class *Building* to be inherited by all buildings. Similarly, there are no separate classes for text and symbols; these are contained in the objects they refer to. For example, a house number belongs to the appropriate house object, an antiquity name to the antiquity object and so on.

Geometry and *Feature* are connected by the many-to-many relationship *Depicts.* In this way a particular map object, for example, *194 Bangor Road,* is connected to the geometry objects that depict it (figure 4). This would include the lines depicting its outline, associated features and the house-number. Similarly, all geometry objects are connected to the features they are part of. The dividing wall between semi-detached houses, for example, would be connected to both house objects.

The database is designed to be seamless. However, to be consistent with existing map sheet series. *MapSheet* indicates those sheets in the series on which a particular feature appears. A map-sheet will contain many features: a particular feature may fall on more than one sheet.

There is therefore a N:M relationship between the two. In a large topographical database containing data captured at different scales, the *MapSheet* class is likely to be more complex than shown here.

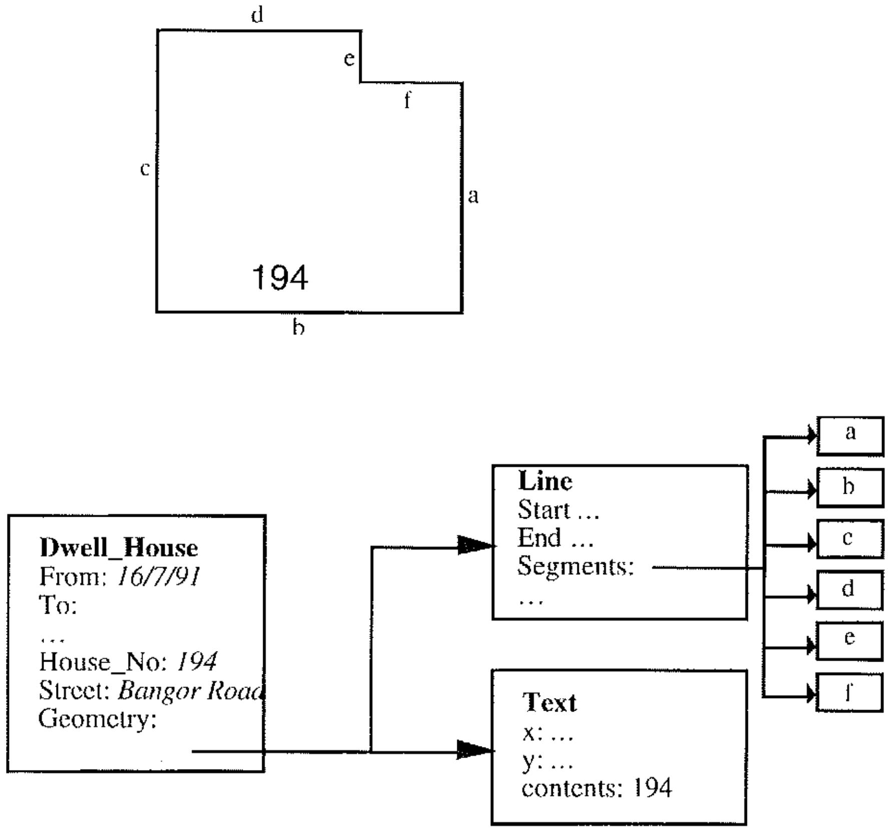


Figure 4 – Representation of a house in an OO topographic database

To illustrate the structure of the database, figure 4 shows the geometrical depiction of a typical building together with its representation. The building outline is stored as one line made up of six segments labelled *a* to *f*. The feature would be entered into the database by first indicating a new *Dwell_House* object is to be created. The date of creation of this initial version would be automatically recorded. The user would then be prompted to input the textual and graphical properties of the object. The different geometric objects involved and the links between them would be automatically maintained – as far as the user is concerned there is only one object.

The subsequent revision of the map causing, say, the addition of a new adjoining glass building would result in the configuration shown in figure 5.

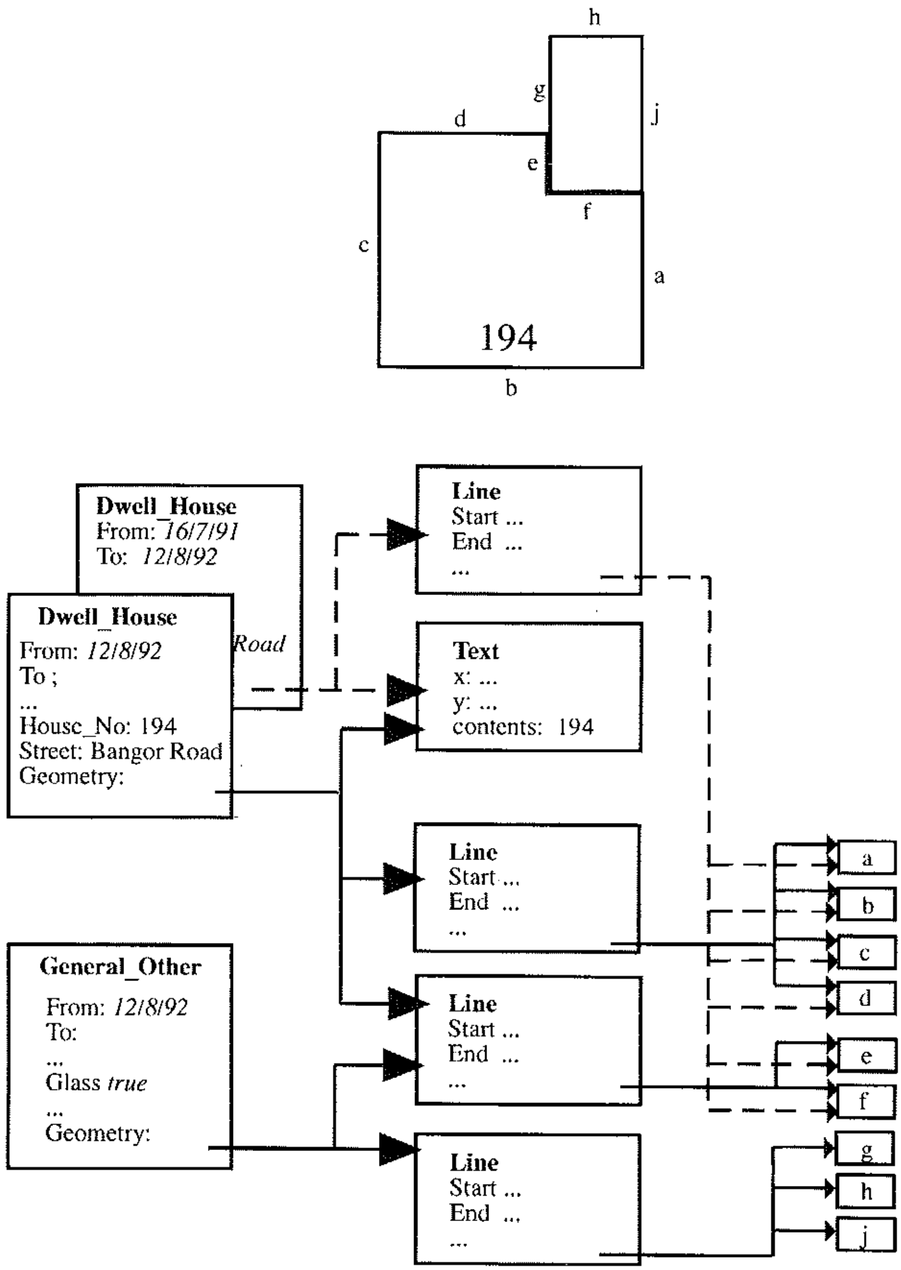


Figure 5 – Addition of an adjoining building

The new building is entered in the database. It's four sides are digitised but the system recognises that they partially coincide with existing geometry and so readjusts the existing lines to provide nodes at the appropriate points and share common elements. Notice how the line segments *a* to *f* remain unchanged even though the higher-level line structure has altered. This represents the real-world situation where the walls they represent are unchanged.

A further revision of the original building similarly causes the database to adjust as shown in figure 6.

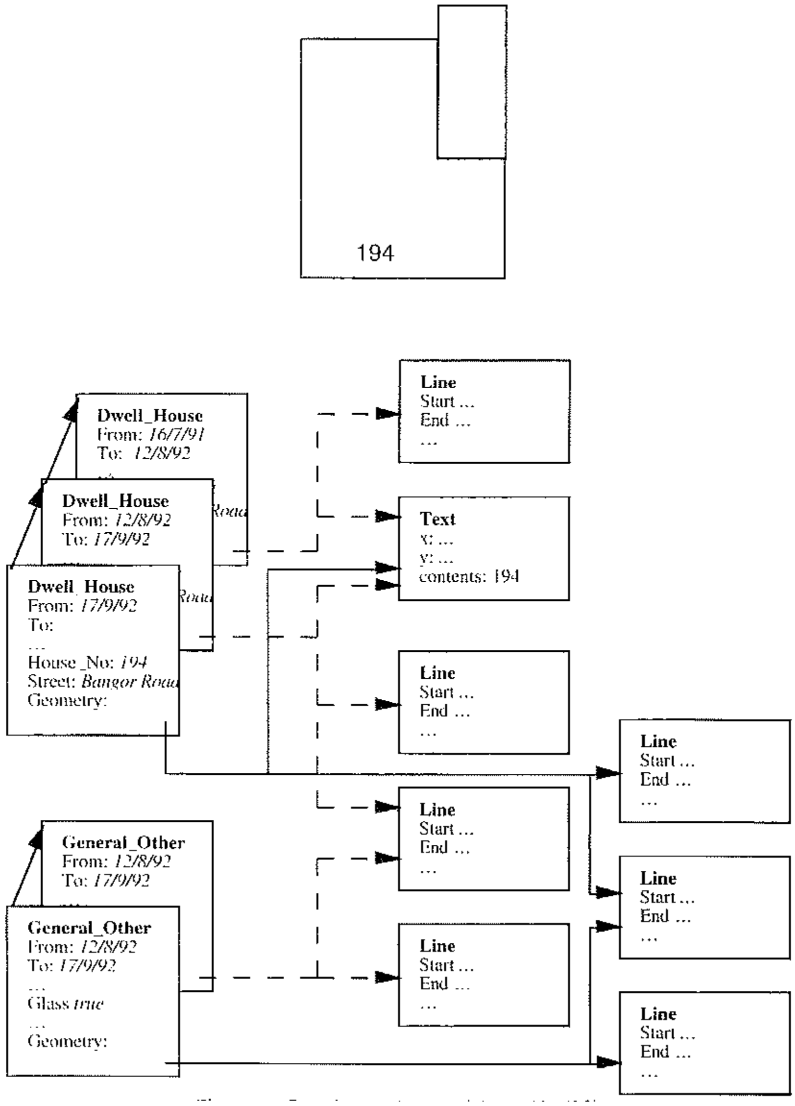


Figure 6 – Database after revision of building

Now there are three date-stamped versions of the house in the database. By default, all operations to display, print or query the house refer to the current version. However, by using the methods *GetPreviousVersion* and *GetVersionAtDate* the older versions can be accessed. The geometry records are maintained automatically by the database. Note how when geometry does not change between versions, as with the line segments *a* to *f* and text *194*, it is shared between them. The data model has been described using a building as an example. However, this versioning method is applicable to any kind of topographic object.

5. IMPLEMENTATION

To demonstrate and experiment with the conceptual model described above, a pilot system is being developed. The data model, provided as an appendix to this paper, has been implemented using the object-oriented database system POET [1993] on an IBM PC compatible machine. POET provides database extensions to the standard object-oriented programming language C++. In particular, it allows data class definitions to be prefixed by the key-word *persistent*. Storage is organised for these classes and a set of standard methods is provided so that objects belonging to these classes can be:

- stored and retrieved from disk
- organised into sets
- searched and queried based on values
- indexed
- locked and unlocked (for multi-user applications)

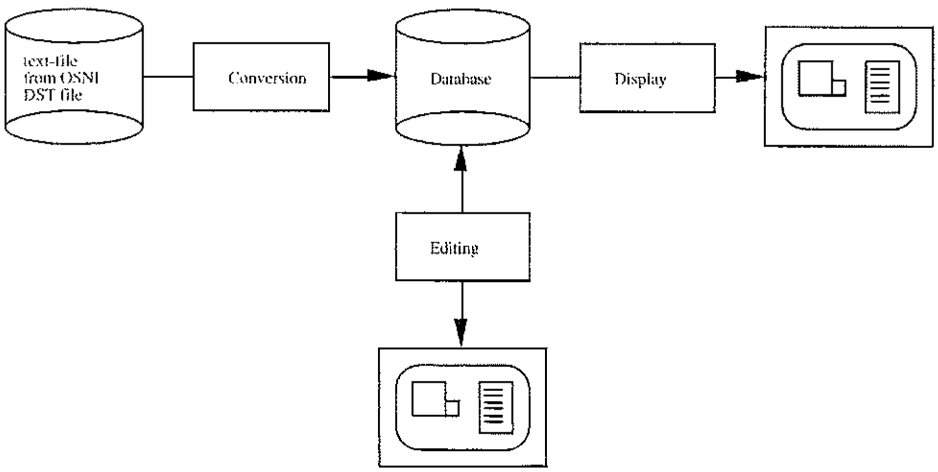


Figure 7 – Configuration of programs in demonstration system

The configuration of the system is shown in Figure 7. Three separate programs will interact with the database:

- A *conversion* program to construct the database from a text-file generated from an OSNI DST file. Because the OO database should store more information that the DST file (grouping geometry into real-world objects), for all feature types the conversion process would require rules for the construction of objects from geometry. The demonstration program only deals with buildings, however, and the geometry of each building is stored in the DST file as one object (area) already.
- An *editing* program that allows revisions to the map and stores them as new versions of objects.
- A *display* program that displays the results of time-dependent queries.

6. CONCLUSIONS

This paper has proposed an object-oriented method for modelling changes to topographic databases. It improves on the data structure of the current OSNI model while providing a simple versioning mechanism for storing the history of a topographic object. It satisfies the original requirements for this project set out by OSNI in the following ways:

1. *It should provide an up-to-date, fully structured and attributed graphical representation.* The data model proposed through the M:N *Geometry-MapFeature* relationship provides similar structuring of graphics to the current OSNI model. It goes further however by maintaining all features as individual objects, which can be selected and manipulated as such, rather than as individual feature-coded lines, symbols and text. At present the OSNI data model only treats the area of a *Dwell_House* as an object. The proposed model also integrates the associated data into the main topographical database.
2. *Dates of the addition, modification and suppression of data should be held, maintained and presented.* Using the *VersionedObject* super-class described in section 3 these dates are automatically attached to all topographic objects. They can be entered by the user or, if the dates concerned represent the date the change is entered in the database, be maintained automatically by the database system. This is the method adopted by the demonstration system.
3. *Output of the revised data should be available in terms of replacement map sheet and change only information.* The *replacement map sheet* is represented by all the current versions of the objects on the sheet; *change-only* information by those current versions whose *CurrentFrom* date is later than that of the previous map-sheet.
4. *An archive should be maintained of topographic data no longer extant.* The data no longer extant is represented by all object versions which have a *CurrentTo* date earlier than present (ie the *CurrentTo* date has been entered). The "archive", as such, is integrated with the current database. However it would be possible to separate the current information from the archive if required.

5. *Archived data may be held as a layer in the database.* The concept of a "layer" is not used in the proposed model. Separating the current data from the archive, as described in 4, would have an equivalent effect.

6. *Archived data should be accessible for specific historical points and/or periods.* Using the method *GetVersionAtDate* on versioned objects, the set of all object versions that were current at a particular date are returned. These can then be searched, displayed, printed, etc. in the same way as the current data.

7. *A continuous system of such updating and archiving should be operated and administrated.* The discussion in section 4 has indicated a possible procedure for updating the database. The demonstration system will provide a user interface that matches the object-oriented data model described, shielding the user from the underlying geometrical model of lines and nodes. The user will then enter, update and manipulate the data in terms of topographic objects rather than geometric elements.

REFERENCES

Agrawal, R and Gehani, N, [1989]: ODE: The Language and the Data Model. Proceedings ACM SIGMOD Conference on the Management of Data, Portland.

Bofakos, P, Story, P and Worboys, M F [1993]: Design and Implementation of a Temporal GIS, GIS Research UK, Keele, pp 69–77.

Borhani, M, Barthés, J P A, Anota, P and Gaillard, F [1992]: A Synthesis of the Versioning Problems in Object-Oriented Engineering Systems, 3rd International Conference on Data and Knowledge Systems for Manufacturing and Engineering, Lyon, pp 165–182.

Hughes, J G, [1991]: Object-Oriented Databases, Prentice Hall, London.

Kowalczyk, A and Kemp, Z [1993]: Incorporating the Temporal Dimension in a GIS Application, GIS Research UK, Keele, pp 46–67.

Langran, G, [1993]:Time in Geographic Information Systems, Taylor and Francis, London.

OSNI [1992]: Updating of Complex Topographic Databases, Final project proposal, Ordnance Survey of Northern Ireland.

POET [1993]: POET release 2.0 programmer's and reference guide, BKS Software Entwicklungs GmbH, Hamburg.

Snodgrass, R and Ahn, I [1985]: A Taxonomy of Time in Databases, SIGMOD '85 Proceedings, Keele, pp 236–246.

Wachowicz, M and Healey, R G [1993]: Towards Temporality in GIS, GIS Research UK, Keele, pp 58–68.

```
//gisdef.hcd
//Header file for versioned topographical database

#include <poet.hxx>

class TimeStamp             //stores temporal information
{
private:
      PtDate      Date;
      PtTime      Time;
public:
      TimeStamp();          //Constructs using current date and time
      void                  SetTimeStamp (PtDate, PtTime);
      PtDate GetDate();
      PtTime GetTime();
      int         operator > (TimeStamp &);          //later than
      int         operator < (TimeStamp &);          //earlier than
};

persistent class VersionedObject
{
private:
      TimeStamp                                      From;
      TimeStamp                                      To;
      ondemand<VersionedObject>                      Previous;
public:
      VersionedObject   (); //Constructs with From = now
      VersionedObject   *CreateNewVersion ();
      VersionedObject   *GetpreviousVersion ();
      VersionedObject   *GetVersionAtDate (PtDate Date);
      TimeStamp         GetFrom ();
      TimeStamp         GetTo ();
};

persistent class MapFeature;
// defined below

persistent class Geometry
{
private:
      1set<ondemand<MapFeature>>      Features:      //list of map objects
                                                        belongs to
      int         colour;
      double      height;
public:
Geometry();                 //constructor
            void            AddFeature(MapFeature);
            void            RemoveFeature (MapFeature);
      virtual  void Display ();
};
```

```
persistent class Point: public Geometry
{
private:
        double x,y;
public:
        Point (double,       double);                //constructor
        void                 SetXY(double, double);
        virtual              void Display();
};

persistent class Symbol: public Point
{
private:
        int        SymbolNo;
        int        Rotation;
public:
        Symbol     ();
        Symbol     (double, double, int);
        Symbol     (double, double, int, int);
        void       SetSymbolNo(int);
        void       SetRotation(int);
        void       Display();
}

persistent class Text: public Point
{
private:
        int        Size;
        int        Inclination;
        int        Underlining;
        int        Rotation;
        PtString   Contents;
public:
        Text( );
        Text (double, double, PtString);
        Text (double, double, int, int, int, int, PtString);
        void       SetSize (int);
        void       SetInclination (int);
        void       SetUnderlining (int);
        void       SetRotation (int);
        void       Display( );
};

persistent class Arc: public Point
{
private:
        int        Thickness;
        int        Linetype;
        int        Radius;
        int        StartAngle;
        int        AngleRange;
```

```
public:
        Arc( );
        Arc (double, double, int);
        Arc (double, double, int, int, int);
        void        SetThickness (int);
        void        SetLinetype (int);
        void        SetRadius (int);
        void        SetStartAngle (int);
        void        SetAngleRange (int);
        void        Display();
};

persistent class Linepoint;
persistent class Node;
// defined below

persistent class Line: public Geometry
{
private:
        ondemand<LinePoint>                 Start;
        ondemand<LinePoint>                 End;
        ondemand<PolyLine>                  PLine;
public:
        Line    ();
        Line    (LinePoint*);
        void    AddEnd (LinePoint*);
        void    Display();
};

persistent class PolyLine: public Geometry
{
private:
        ondemand<Node>                      Start;
        ondemand<Node>                      End;
        1set<ondemand<Line>>                Lines;
        int                                 Thickness;
        int                                 Linetype;
public:
        Polyline ();
        Line (Node*);
        void        AddLine (Line *);
        void        AddEnd (Node*);
        void        SetThickness (int);
        void        SetLineType (int);
        void        Display();
};

persistent class LinePoint: public Point
{
private
        ondemand<Line>              Line1, Line2;
public:
        LinePoint (double, double, Line*);
        LinePoint (Line*, Line*);
}
```

```
persistent class Node: public LinePoint
{
private:
        cset<ondemand<PolyLine>>            PLines;
public:
        Node (double, double);
        void         AddLine (Line*);
        void         RemoveLine (Line *);
        void         Display();
};

persistent class MapFeature: public VersionedObject
{
private:
        lset<ondemand<Geometry>>        Geom;
public:
        MapFeature ();
        void         AddGeometry (Geometry*);
        void         Display();
};

persistent class Building: public MapFeature
{
private:
        PtString     Geocode;
        int          NonSquare;
        int          Glass;
public:
        Building ();
        Building (PtString);
        void         SetNonSquare();
        void         SetGlass();
        void         Display();
};

enum CountyType {Antrim, Armagh, Down, Fermanagh, Londonderry, Tyrone };

persistent class DwellHouse: public Building
{
private:
        int          HouseNo;
        PtString     Street, Townland, City;
        enum CountyType                                 County;
        PtString     Postcode;
public:
        DwellHouse();
        DwellHouse (int, PtString, PtString, PtString, CountyType, PtString);
        void Display();
};
```

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES
RESEARCH GROUP: UNIVERSITY OF PAVIA

FINAL REPORT

R Galetto – F Viola

9 November 1993

1. ANSWER TO THE KEY AREA N.1

how an updated, fully structured and attributed digital graphic, conforming to the full OSNI database specification and free of map sheet line constraints, can best be arrived at.

The way envisaged to best cater for the aims expressed in the key area N.1 is based on the following steps.

1.1 *Restructuring of the graphic data*

The structure of the graphic data has been changed from the original topological to a non-topological structure.

This fact implies that the cartographic objects (for the terminology see the appendix) of each instance of entity are realized by means of simple or complex objects which have only topographic value; each instance of entity has its individual geometric description independent from that of all the others in the database.

Thus if we remove from the database the cartographic object which geometrically describes an instance of entity, the geometric descriptions of all the others remain unchanged.

From the point of view of the geometry three different kinds of entities are possible:

- area entities, whose corresponding cartographic objects are made up of simple or complex polygons
- line entities, whose corresponding cartographic objects are made up of strings
- point entities, whose corresponding cartographic objects are made up of points.

1.2 *Introduction of the concept of fictitious current date*

When the user examines the data in the database they are found in a given updated situation, which corresponds to the last updating. The consultation date can be called *the current date.*

To best achieve the ends of the project the concept of a *fictitious current date* has been introduced.

That is to say: the software which has been used to carry out the project can accept that the user chooses any date in the past, and it performs all the functions as through that were the real current date. This is the fictitious current date.

1.3 *Replacement of the additional/modification/suppression operations with addition/disactivation ones*

1.3.1 The concepts of *addition* and of *disactivation*

The basic concept of the updating procedure adopted is that we have replaced the operations of *addition, modification* and *suppression* of an instance of entity, and have substituted for them *addition* and *disactivation.*

By *addition,* we mean addition in the general sense, that is, the introduction of an instance of entity into the database.

By *disactivation,* we mean that when we find that an instance of entity no longer exists in the real world we do not suppress it, removing it from the database, but leave it *disactivated* in the database, tagged as *no longer extant* in the real world. This kind of operation will be called from now on *disactivation.*

Consequently, when we find that an instance of entity in the real world is different from its representation in the database, in regard to one or more of the values of its attributes or its cartographic description or both, we act in two ways:

- we *disactivate* it, that is, we save the situation as it appears in the database as *no longer extant*
- we *add* to the database a new instance of entity which corresponds to the new situation in the real world.

If an instance of entity changes several times it will be disactivated accordingly several times. That is, in the database the most recent situation will be retained as well as all the previous ones, each of them tagged with the date of its recognition as no longer extant in the real world.

All the instances of entity of the database will compulsorily have the introduction date; those *no longer extant* will have the disactivation date too.

1.3.2 The *simple addition* and *simple disactivation* procedures

When an instance of entity is introduced into the database it is tagged with the date of its introduction and it will never be removed. The introduction of an instance of entity into the database can be classified as *simple addition* or *addition.* We have a *simple addition* of an instance of entity when its introduction does not imply the disactivation of any other instance of entity already existing in the database.

When an instance of entity disappears in the real world it remains in the database as *no longer extant* and it is tagged with the date of the registration of this fact in the database; the fact that an instance of entity is tagged as no longer extant will be indicated as *disactivation* of the instance of entity, as previously said; this can then be subdivided into *simple disactivation* and *disactivation*. We have a *simple disactivation* of an instance of entity when its classification as *no longer extant* in the database does not imply the disactivation of any other instance of entity already existing in the database.

1.3.3 The complex operations

An addition, or a disactivation, is no longer *simple* when it is carried out within the context of other additions and/or disactivations, with which it has an interrelating link.

The link can be *objective* or *subjective* or both.

An *objective* link is when the addition, or the disactivation, necessarily implies the disactivation of at least one existing instance of entity.

A *subjective* link between a number of additions and disactivations is created by the operator when he judges that it is necessary as a consequence of interrelated changes to reality.

All the operations performed in the context of the same link represent a *complex operation*.

Each complex operation assumes in the database a sequential number.

All the cartographic objects of the instances of entity linked in the same complex operation are tagged with the number of the complex operation.

The record which describes the cartographic object of an instance of entity in the graphic file has two dedicated fields to the complex operation number. Indeed an instance of entity can be added to the database with a complex

operation and can be disactivated within another complex operation.

2. ANSWER TO THE KEY AREA N.2

how existing records linked to the graphic should be modified and how the data dealt within the course of updating should be tagged with date of addition/modification/suppression, held, maintained and presented.

The way envisaged to best cater for the aims expressed in the key area N.2 is based on the following steps.

2.1 *Structuring the database on three types of files*

The database structure is characterized by the following types of files:

- textual data files
- graphic files
- graphic-DBIII link file (auxiliary file in the following)

Even if in an up-and-running application it might be realistic to have several such groups of files, each of them relating to a part of the whole territory managed by the database. In this report we shall assume that we have only one in the database.

2.1.1 Textual data files

DBIII files are used to store the textual information of the entities; each class of entity is stored in a different file. The link between an instance of entity in the graphic file and the corresponding record in the DBIII file is performed by means of an identification code.

As in the data supplied by OSNI two classes of entities have textual records, buildings and vegetation, two DBIII files have been used.

For the classes *buildings* and *vegetation* the DBIII file records are structured according to the group of attributes associated with these classes in the OSNI data.

Each file contains all the instances of entity which belong to the class to which the file refers. According to the updating operations the records can refer to *existing* or *no longer extant* instances of entity. A dedicated field in the record will contain the layer value of the corresponding record in the graphic files (see the following description).

2.1.2 The graphic files

The cartographic objects of the instances of entity are stored in the database by means of two files:

- the description file (D file in the following)
- the co-ordinates file (C file in the following)

(a) D file

For each instance of entity we have one record in the description file and as many records in the co-ordinates file as are the points of its geometry.

For each instance of entity the record structure in the file D is the following:

- name: sequential number which identifies the instance of entity within the file; for instances of entity which belong to the buildings and vegetation classes this number is used as link to the corresponding record in the DBIII files.
- code: class code
- point_number: number of points which describe the geometry
- co-ord_pointer: pointer to the record of the file C in which the first point of the entity is stored.
- colour: number of the colour by which the instance of entity is drawn on the screen
- layer: the value is set to 1 for the instances of entity existing in reality; it is set to 101 for instances of entity no longer extant
- addition date: the date in which the instances of entity has been added into the database

- disactivation date: the date from which the instance of entity is no longer extant in reality
- complex addition: the value is set to zero for simple addition or to the number of the complex operation within which the addition has been performed
- complex disactivation: the value is set to zero for simple disactivation or to the number of the complex operation within which the disactivation has been performed

The layer is strictly interrelated with the addition or disactivation date: indeed if the disactivation date is set to 00/00/00 the layer must be de rigeur equal to 1 and if the disactivation date is not 00/00/00 it must equal to 101.

The redundancy aims to avoid errors in the updating operations.

In order to assume the time as the fourth dimension of the database not only the entities which have corresponding textual records must be tagged with the addition and disactivation dates, but all the entities. For this reason the addition and disactivation dates are stored in the graphic file instead of in the DBIII files.

(b) C file

The C file contains the co-ordinates of the points which describe the geometry of the instances of entity.

The record structure is:

- point code: pen up code (–1) or pen down code (1)
- x, y, z co-ordinates of the point.

(c) Text file

Besides the files D and C a text file also exists in which the names of rivers, roads and so on are stored; for reasons of simplicity for the OSNI data this file was not created.

2.1.3 The auxiliary file

This file has the function of linking the cartographic objects of the D file, corresponding to the instances of entity which have textual data, to the DBIII file in which the textual data are stored.

The record has 2 fields:

- a class code
- the DBIII file name which contains the textual data of the entities having the previous class code.

2.2 *Modification of existing software*

The basic software used to carry out the project originates from an existing software expressly produced to manage digital cartography; its main facilities are:

- acquisition
- visualization; a large number of filtering criteria is available (by code, by colour, by type, by layer)
- editing
- updating (by digitizer or other standard input devices)
- data validation

performed by means of more than 200 dedicated menu driven functions.

The source code is in C language and it runs under MS-DOS.

The hardware configuration used to run the software is:

- PC 80386 or 80486
- double monitor, a text monitor and a graphic one with 1280 x 1024 pixels resolution
- matrox card 1281
- mouse
- digitizer

For the realization of the new software derived from the existing one, these modifications have been performed:

2.2.1 the management of the addition date, disactivation date, complex operation numbers fields of the record of the D file.

2.2.2 linking to the DBIII files; the link allows two operations. First: when the cartography is visualized on the graphic screen, if an instance of entity is captured (that is, it has been pointed by the cursor on the graphic screen and enhanced) and it has a corresponding record in a DBIII file, this record is printed on the text screen. Second: when a selection is made on a class file using the DBIII capabilites, the names of the instances of entity matching the selection criteria are stored in an ASCII file which can be managed by the graphic software.

2.2.3 implementation of the following new capabilities:

2.2.3.1 introduction of the addition/disactivation date and of the complex operation number in the dedicated fields performed for an instance of entity or for a group of instances of entity enclosed in a given polygon.

2.2.3.2 the introduction of a different graphic representation on the graphic screen for existing and disactivated instances of entity; the existing ones are drawn with solid lines, the disactivated ones are drawn with dotted lines.

2.2.3.3 *layer* filter: this filter gives the possibility of visualising only the existing instances of entity, or the disactivated ones, or both.

2.2.3.4 *change only* filter: this filter selects the instances of entity which have the addition date or the disactivation date between two given dates. We have the more straightforward application of this filter when the two dates define an interval of time which includes only the date of the last updating; in this case the filter gives the classic *change only* situation.

2.2.3.5 *complex operation yes/no* filter: this filter selects the instances of entity according to whether they were, or not, involved in a complex operation, when they have been introduced into the database or in following updating operations.

2.2.3.6 *fictitious current date* function: the default situation that is taken into account when managing the database refers to the current date; this function gives the possibility of looking at the territory as it was in the past by setting up a fictitious current date, that is, a current date which corresponds to a date situated in the past. Indeed to fix such a fictitious current date will have these consequences:

- the instances of entity which have the addition date subsequent to the fictitious current date will be considered as not existing in reality
- the instances of entity which have the addition date preceding the fictitious current date and the disactivation date subsequent to the fictitious current date will be considered as still existing in reality.

2.2.3.7 *complex operation group* function: given a complex operation number, the instances of entity involved in this complex operation are selected, and they alone drawn on the graphic screen.

2.2.3.8. *DBIII query* filter: setting this filter the software takes into consideration only the instances of entity previously selected by a DBIII query and whose names are stored in an ASCII file produced in the DBIII environment.

2.2.3.9 many functions devoted to the updating procedure have been changed according to the needs of the project.

2.3 The addition and disactivation procedures

Now we take into consideration the main procedures which can be performed with the modified software.

2.3.1 The addition of an instance of entity

The cartographic object of the instance of entity is created adding a record in the D file and the corresponding co-ordinates records in the C file. The introduction date field is filled with the updating date and the layer value is set to 1 for simple addition or it is set to the number of the complex operation if it is the case.

If the instance of entity has textual attributes a record in the corresponding DBIII file is created and filled and linked to its corresponding cartographic object.

2.3.2 The disactivation of an instance of entity.

The disactivation procedure is applied to an instance of entity in two cases:

(a) when the instance of entity does not exist any more in reality

(b) when the instance of entity has had a change in reality which implies a change of its cartographic object or of its textual record (if it exists) or both.

Case (a)

In the record of the D file corresponding to the cartographic object of the instance of entity the value of the layer field is set to 101 and the disactivation date field is filled with the updating date. If the instance of entity has a textual record in a DBIII file, the layer field in the record is set to 101.

Case (b)

The records concerning the cartographic object descriptions in the D and C files are duplicated; a new name is automatically given to the cartographic object; the DBIII record, if it exists, is duplicated and linked with the duplicated cartographic object; the cartographic object is updated using the graphic software capabilities and the textual record is updated using the DBIII capabilities. Then the old situation is disactivated with the same operations as case (a).

In order to save time in the updating procedures, the introduction of the addition or disactivation dates and the value of the *existing/no longer extant* layer can be performed by a group of instances of entity by the function illustrated in 2.2.3.1.

2.4 The database data presentation

The data stored in the DBIII files can be consulted by the DBIII capabilities or using the software function described in 2.2.2.

The data concerning the cartographic objects stored in the graphic files can be displayed on the graphic screen, as illustrated in 2.2.3.2, according to a large number of filters already available in the original software and to those specifically produced for the project illustrated in 2.2.3.3, 2.2.3.5, 2.2.3.7 and 2.2.3.8.

3. ANSWER TO THE KEY AREA N.3

how output of revised topographic data, both digital and hard copy, should be handled in terms of replacement map sheet and much more importantly, change only information.

3.1 The replacement map sheet problem

Due to the fact that we hold in the database the present situation as well as all the previous ones, after any updating phase we do not have any replacements to carry out. As usual, what must be done is only a backup of the archives, for safety.

3.2 *The change only information*

The availability in the software of the *change only* filter, as illustrated in 2.2.3.4, is a very powerful tool to cater for any query of the user about this problem.

As has been previously said, we can have the *change only* situation between any two desired dates. If the user sets as the first date a date immediately preceding the last updating and as the second date the current date, he will have the change only situation in the usual sense. But many other queries are possible; just to give an example:

show the *change only* situation between date1-date2 (whichever) regarding only some classes (for instance buildings), and, within these classes, only the instances of entity involved in complex operations, and show not only the existing ones, but also the disactivated ones (existing and disactivated are drawn in different ways: solid and dotted lines).

4. ANSWER TO THE KEY AREA N.4

how archiving of topographic information no longer extant should be dealt with.

The topographic information no longer extant is maintained in the database (in the graphic file) tagged as no longer extant (value 101 in the layer field).

This solution allows the user to perform on the topographic information no longer extant, all the operations available for the existing ones.

5. ANSWER TO THE KEY AREA N.5

assuming the requirement to hold the topographic information no longer extant, inclusive of all attributes, date of suppression, etc, as a layer within the topographic database, how should this be met.

As previously said, in the structure of the database that we propose, the topographic information no longer extant is indeed a layer; and also the records which contain the textual information in the DBIII files have a field in which the value of this layer is registered.

6. ANSWER TO THE KEY AREA N.6

how the topographic information no longer extant can be interrogated/output for specific historical points and/or period.

To answer this question two functions are available in the software:

- the *current date* function
- the *change only* function

With the *current date* function (illustrated in 2.2.3.6) it is possible to have a flash back for whatever point in the past is wanted, and to see the situation as it was at that moment. When a date is set with this function all the capabilities of the software work, as all instances of entity having the addition date subsequent to the given date were not in the database, and all the instances of entity having the disactivation date subsequent to the given date were still existing.

Setting two dates, with the *change only* function it is possible to examine the evolution of the territory in this period, using any combination of the filters.

Just to give an example: show only the buildings of a given category introduced into the database between date1 and date2, which at present are no longer extant and whose area was less than 120 square metres.

7. ANSWER TO THE KEY AREA N.7

how a continuing policy of such updating and archiving can be satisfactorily operated and how administration of the resultant database can be kept manageable, given a policy of continuous revision in line with ground change occurring.

The proposed method does not pose any particular problems in becoming an up-and-running application; on the other hand it offers many advantages:

- it utilizes cheap hardware and software and so lends itself to the setting up of many workstations which can follow the continuous revision work without falling behind.
- the updating procedures are easily carried out as the non-topological structure of the data makes the geometric description of cartographic objects closer to reality.
- the capabilities of the software allow the user to verify the consistency of the different types of data, geometric, textual and temporal.

The manageability of the database can be assured by subdividing the territory into not too extensive zones.

class of entities: a class of real world phenomena which are not subdivisible into phenomena of the same kind (for example: buildings, roads, rivers...)

entity: a generic real world phenomenon which belongs to a class of entities (for example: a house, a road...)

instance of entity: an individual materialization of an entity of a class of entities (for example: the building located at number 62 Chestnut Road, London)

cartographic object: a digital representation of an instance of entity in the database

simple objects: the elements by means of which the cartographic objects are constructed; simple objects can have geometric value only or both geometric and topological value.

For example:

(i) Points, segments, strings and rings of strings are simple objects with geometric value only.

(ii) Nodes, links, chains and rings of chains are simple objects corresponding to the previous ones but with geometric and topologic value.

complex objects: a mix of simple objects, for instance a complex polygon.

attribute: a defined characteristic which indicates an aspect under which the entities of a class of entities are seen in the database.

attribute value: a specific quality or quantity assigned to an attribute for a given instance of entity.

group of attributes: all the attributes relating to a class of entities.

APPENDIX X

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

FINAL REPORT
28 OCTOBER 1993

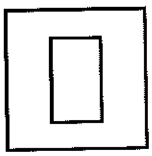
JAN BEYEN
NATIONAAL GEOGRAFISCH INSTITUUT
BRUSSELS

PREFACE

Since on the one hand the Intergraph translation to the MGE format of the OSNI dataset was very unsuccessful and on the other hand the documentation on the project only arrived in Brussels by 7 October 1993, we did not have enough time to carry out much of the practical work, which was to be done in phase II of the project.

Nevertheless, by assuming how the OSNI dataset would look like in a proper MGE translation, it became possible to test a slightly adapted version of the updating procedure that we proposed in phase 1. "Adapted" because of new views on the matter and because of the detected differences in OSNI and IGN concepts.

Our proposal for phase 1 was worked out regarding our own 1:10 000 – 1:20 000 scale topographic mapping, in which area objects are never part of other area objects. In the following figure we see a house surrounded by a garden.



When the house is getting bigger, the garden is getting smaller and vice-versa. OSNI though, also considers administrative data like land parcels. OSNI see the same figure as a land parcel containing a house. When the house is getting bigger, the land parcel's size does not change. Directly linked to this difference in points of view and to the difference in working scales there is a different approach in making objects of the real world features and in building topology.

The conceptual data model used in the National Geographical Institute

For practical reasons we do not use different databases as previously described, but one big database to which various tables, text files and graphical files may be loaded or not; depending on the information we need.

The geometry is mainly stored by stereoplotting, while the real (ie, geometry independent) attributes are surveyed in the field, and this with another frequency. The database structure keeps this distinction.

For every feature category (buildings/hydrography/...) we distinguish 4 kinds of files containing vector graphics:

- actual geometry
- former geometry (history)

- new geometry working file
- suppressed geometry working file

+ 1 everlasting geo table for every feature type (eg, house, roadsegment, but not a wall in a house...), containing

- mslink, ie, the objectnumber which is also written in the graphical element
- mapid = identification of the graphical file

  Together mslink and mapid form a (MGE) software dependent identifier for a graphical element
- dates of source/input/suppression
- geometry dependent attributes
- unique combination of pointer values to be used for all geometries of a certain object and to be repeated in all related tables.

This unique pointer combination is a software independent identifier for an object.

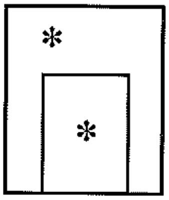
+ for every feature type eventually two kinds of tables with 'geometry independent' attributes:

- 1 table with general attributes (+ one single date for the lot)
- 1 or more tables with specific attributes which should only be filled out if a general attribute takes a certain value.

For simple objects both are directly related to the geo table.

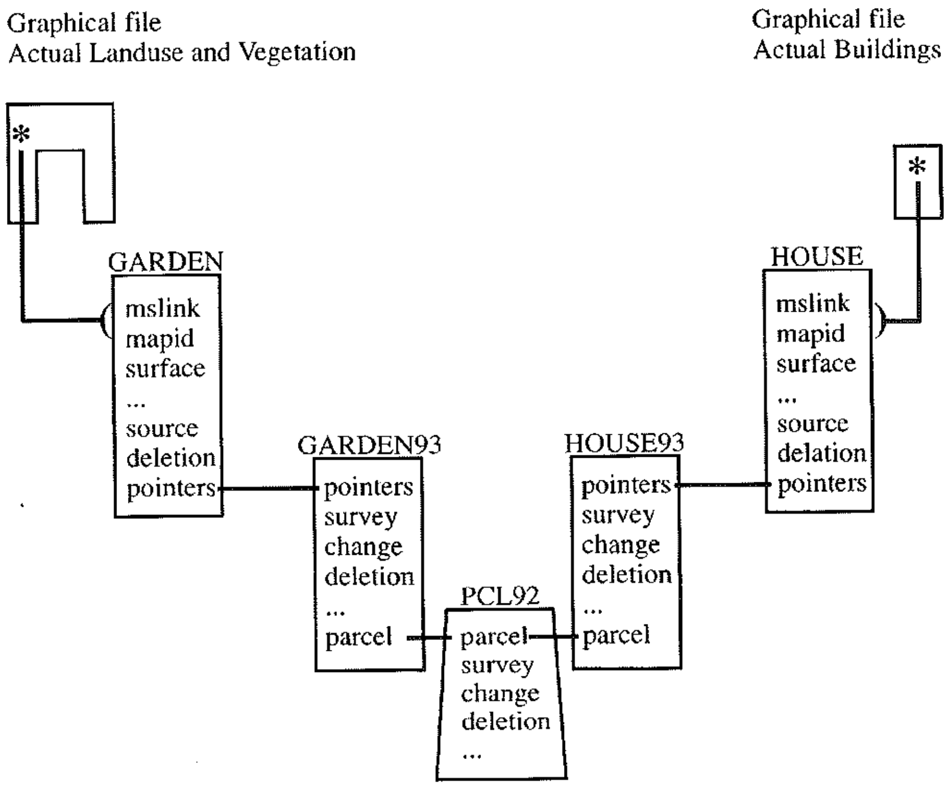
For complex objects both are related to the geo table through the table of their components.

+ Text files (and raster graphics) can be reached through attributes that have the full path of the file as a value.



For this figure we would make objects (a house and a garden) of the two area centroids and link geometry dependent and geometry independent attributes in separate tables to them. So there are at least two different tables per single object. Possible attribute values may be listed (or limited by range) in domain tables.

A land parcel might exist as a complex object (without direct graphical representation) linked to the two (simple) area centroid objects. So for complex objects we only have the second (geometry independent) attribute table.



Only when building topology, the area boundaries and their geometry are linked to the area centroids. This means that a change in the geometry asks for a local rebuilding of the topology, and this either afterwards in batch with MGE-SX or directly (automatically) with MGE_Dynamics, which we do not have at the moment.

Graphically we have two (.dgn) files per feature category: an actual and an historical one. We also have two temporary files (for new and deleted objects).

Answers to the 7 Key Areas, applied to the IGN conceptual data model

1. Procedures for Updating

1.1 Geometry

Unlike OSNI data, the data within the CDM used in IGN are not seamless, but tiled.

When for example, the old geometry of a number of houses is changed we would normally store new and delete old geometry.

1. By stereoplotting: store new geometry of the houses in new.dgn

2. Bulk: make features of all elements in new.dgn; the program centroidplacer adds lines (with new mslinks) in the table HOUSE.
3. Interactively: fill out the other columns (except from mapid and date of survey) of the new lines in the table HOUSE, giving the new geometry every time the same software independent object identification as the old geometry has. This way a house becomes a complex object with all its geometry versions linked to the same line with geometry independent attributes, (eg, in the table HOUSE93).

3′. Eventually (for new houses) add a line in HOUSE93 and fill out the date of survey.

4. Merge new.dgn into buildings_actual.dgn.

In table HOUSE for all elements appearing in new.dgn, bulk :

5. set mapid = mapid of buildings_actual.dgn.

and

6. fill out the date of survey.

6′. Remark: for new geometry of old houses no impact on HOUSE93.

7. Interactively add the old geometry of the houses from buildings_actual.dgn into a working set.
8. Build a graphical file del.dgn from the working set (instruction : ff = del.dgn)
9. Bulk: delete the old geometry of the houses from buildings_actual.dgn (all houses appearing in the working set; instruction: delete fence contents).
10. Merge del.dgn into buildings_hist.dgn

In table HOUSE. for all elements appearing in del.dgn:

11. Set mapid = mapid of buildings_hist.dgn

and

12. fill out the date of deletion
13. In table HOUSE93 fill out date of deletion (via join through the pointercombination).

For new houses we would apply (1) – (6), with the understanding that in (3) the object identifier must be new and unique.

For houses that disappeared we would apply (7) – (13).

1.2 Geometry independent attributes are periodically surveyed in the field.

For storage of the new survey results we would follow this sequence:

1. Risunload of the database tables.
2. Archive an asciidump of the database tables.
3. Give the (real) attribute tables a new name, eg, HOUSE94 instead of HOUSE93 etc.
4. Load these into the database.

5. Delete all lines where the dod (date of deletion) was filled out the year before.
6. New objects can be treated as modified objects, since at (3') in the geometry updating sequence: for every new object (not : geometry version of old object) we added a line with the data into the (real) attribute table, eg, HOUSE94 filling out the date of survey, but leaving the other columns on default values.
7. Modified objects: change attribute values, including date of change, where necessary (instructions : GDL – query on HOUSE94 review graphics – identify element – modify record).
8. Entirely deleted objects : fill out the date of deletion.

2. Textfiles

Textfiles like the OSNI dwellings and vegetation .lis descriptions that we received, are too regular to be considered as textfiles. They should be loaded into attribute tables and be treated as described above.

For a real, irregular ascii text, – as well as for a raster image – the full path to the file may be filled out as a value in the attribute table. The dates of addition and suppression are the same as those already indicated on the same line in the table. We cannot tag every single attribute value with dates without creating a huge number of tables. This does not really matter since all attributes are – periodically – revised together.

A timestamp within the asciifile gives us the date of an eventual supplementary modification (which should not occur in normal circumstances).

The textfile linked to a certain graphical element can be reached by: gdl query on the right table – review graphics – identify element – gdl execute command ! vi? text?, in which "text" is the name of the attribute that has the full path of the file as a value.

3. Output of Change Only Information

Using dates in ISO8601 format !

New geometry

For graphics:

per category eg buildings eg since 27.10.1992 for all graphical elements appearing in buildings_actual.dgn database search on tables HOUSES, FACTORIES, etc, where date of survey > 19921027 ;_> build a graphical file new.dgn from this ; the client merges this file with his or takes it in reference.

For dbrecords:

per geometry independent attribute table eg, HOUSES, for all graphical elements appearing in new.dgn. select *from houses (ie implicitly where dos > 19921027) _> ascii-dump. The client unloads his version of the corresponding table, merges the two ascii-tables and loads the upgraded table.

For real attributes concerning the new geometry : no search in graphics ; eg select *from HOUSE93 where dos >...

Deleted Geometry

For graphics:

per category, for all graphical elements appearing in eg, buildings_hist.dgn search database tables HOUSES, FACTORIES, ... where dod > 19921027, and build a graphical file del.dgn from this.

The client has to delete from his buildings_actual.dgn all graphical elements appearing in del.dgn. To do so : merge del.dgn into buildings_actual.dgn – linecleaner merge duplicate linework to a free chosen level (attribute linkages are being kept) – either file fence + delete fence contents on this level if the client wants to keep historical data, or delete feature and attribute linkages on the elements of this level if he does not want to keep historical data.

For dbrecords:

per geometry dependent attribute table eg, HOUSES, for all graphical elements appearing in del.dgn (ie implicitly dod > ...) select * from HOUSES > asciidump. The client loads this table DEL_HOUSES to his database, updates his HOUSES by setting (in sql) HOUSES.dod = DEL_HOUSES.dod and he unloads DEL_HOUSES.

For real attributes concerning the deleted geometry : eg. select *from HOUSE93 where dod>...._> asciidump = DEL_HOUSE93. The client loads DEL_HOUSE93, updates dod in HOUSE93 and unloads DEL_HOUSE93.

Modified real attributes

select from HOUSE93 where doc>...._> asciidump = CH_HOUSE93

The client has to replace in his HOUSE93 all lines appearing in CH_HOUSE93:

– load CH_HOUSE93
– update doc (where = pointercombination) in HOUSE93
– delete from HOUSE93 all lines where doc >...
– unload both tables, merge the changes into the main and reload HOUSE93.

4. Archiving of Historical Information

Graphics : All historical information is kept in separate .dgn files (per category). Except when needed for special applications, these files only have to be loaded for updating purposes. This way the total amount of graphical data in the 'actual' .dgn files remains the same, so the system does not slow down.

Database : The records which are directly linked to historical geometry are being kept within the one and only 'geometry dependent attribute table' of every feature. This allows us to select very quickly (through their mapid's) all geometries an object ever had. The fact that this table is getting bigger does not really slow down the searches on actual data because these searches are being performed through the graphics, which are separate.

The asciidumps of real attribute tables are separately archived for every release/revision.

5. See 1 and 4

Whenever needed the graphical history files can be copied onto the disk and the old attribute tables can be loaded into the database. The link to these is permanently available in the 'geometry dependent' tables.

6. Interrogation of Historical Data

As for actual data, but interrogating other files/tables.

eg: Suppose that on 1.7.1994 we need the geometry and attributes of all houses demolished between 1.5.1986 and 1.5.1993 and suppose that we have revised our attributes in 1985, 1988, 1991 and 1994 (utopic):

First solution: for all elements appearing in buildings_hist.dgn,

```
select * from HOUSES where dod > 19860501 and dod < 19930501
select * from HOUSE88 where dod > 19860501
                            HOUSE91 !=0
                            HOUSE94 !=0 and < 19930501
```

Second solution: make joins (through identical pointercombinations)

```
1 between HOUSES and  HOUSE88
2                     HOUSE91
3                     HOUSE94
```

and make views on all columns VIEW_HOUSE88,91 and 94.
then with one single instruction :
select *from these views where HOUSES.dod > 19860501 and HOUSES.dod < 19930501

Therefore avoid giving columns in HOUSES and HOUSExx the same name!

7. Continuous Revision

Although the concept that we described above was developed for periodical revisions, it may be applied for continuous revision as well. In this case the same procedures can be applied without any change for the treatment of the graphical elements and the geometry dependent attributes.

For the real attributes though we would need to add to the concept the notion of 'releases' : in production the revision may be continuous, but for administrative reasons we would be forced to deliver the data in releases to the clients (every year/twice a year/...?)

As more frequent releases would imply more frequent archiving of often the same unchanged data, it might be interesting to maintain the principle that the total database is archived just once a year and that the modification data are output and archived more frequently.

Application of the IGN conceptual data model on the OSNI data

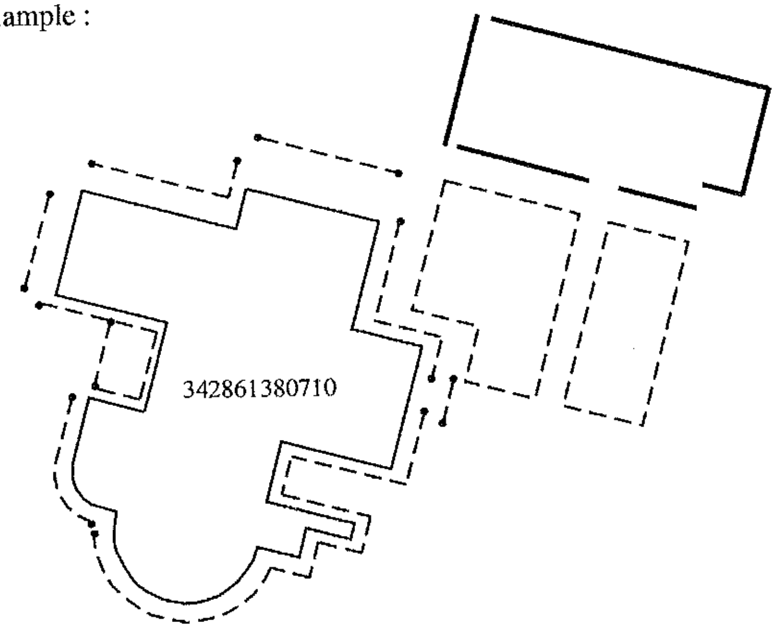
In the Intergraph translation of the OSNI dataset we found on the database side just a feature table ie a list of 292 possibilities for kinds of objects, eg, non_squ_b, dwell_house, comm_roads, road_oth, perim_oth, glass_b, etc, (+ a number, part of which indicates the feature category) and some .lis textfiles containing geocode, house_no, street, etc.

We did not receive any tables containing an enumeration of the existing features and their attributes and references to the graphical representation and to the .lis textfiles.

So on the database side we had no objects, no attributes, no links to the design files and no links to the textfiles!

On the graphical side we received a design file per feature category containing line-strings and complex chains tagged as a simple object and texts (for the geocodes) which were not tagged at all. Lines that have several functions were repeated on different levels and every different representation of the same line was tagged as a different object.

Dataset example :



Legend	Fcode	Type	Number of elements in this figure
(thick solid line)	general_ot	linestring	4
(dashed line)	glsb_b	complex chain	2
(dashed line with dot ends)	non_squ_b	linestring	10
(thin solid line)	dwel_b	complex chain	1
geocode	-	text	1

As we understood from the documentation to the project, in OSNI's original data and contrary to what we received, each line appears only once and is labelled with one single code (lingeo), which is serving as a pointer/key to the associated textual data. This label is considered as a single object. The textual data contain dos, doc, dod, sheet, plan and a multifeature code.

Both the labels and the associated texts were missing in the OEEPE dataset.

Apart from these OSNI data also contain geocode labels within dwelling houses and textual data associated to these (geocode, house_no, street, townland, city, county, post_code, ig_sh, plan, area, survey_date, level_no and deletion_date). As for lingeo labels we suppose that the geocode labels are considered as an object and that the code is serving as a pointer/key to the textual data.

We received both the geocode labels and the associated textual data!

Before even thinking of updating we first had to build a database from these remnants.

The realization of the logical links between

1. a line and its lingeo table
2. an area and its geocode table
3. a label and the text associated to it

depend on software and may be different within our system.

In MGE it is easy to establish the second link, ie, between the dwelling areas and their geocode labels. All it takes is making features of the labels and running the program topology builder. From the feature table we choose the feature dwel_c, which is typed as an area centroid.

It took more effort to associate the texts to the labels. As we already mentioned in our answer to key area 2 the dwellings textfile (13seld.lis) is too regular to be considered as a textfile; it should be loaded into an attribute table.

Therefore, before running the program featuremaker, we attached an attribute table DWEL_C1 to the feature definition of dwel_c, containing the columns occurring in the dwellings textfile + of course mslink and mapid. After making features dwel_c of all the concerned text elements, we used the program labelloader to load the geocode label into the column geocode of the table DWEL_C1.

On the other hand we put command separators between the different columns of the dwellings textfile, made a table DWEL_C2 in the database and loaded the textfile into the table.

This made it possible for us to fill out the other columns of the table DWEL_C1 one by one:

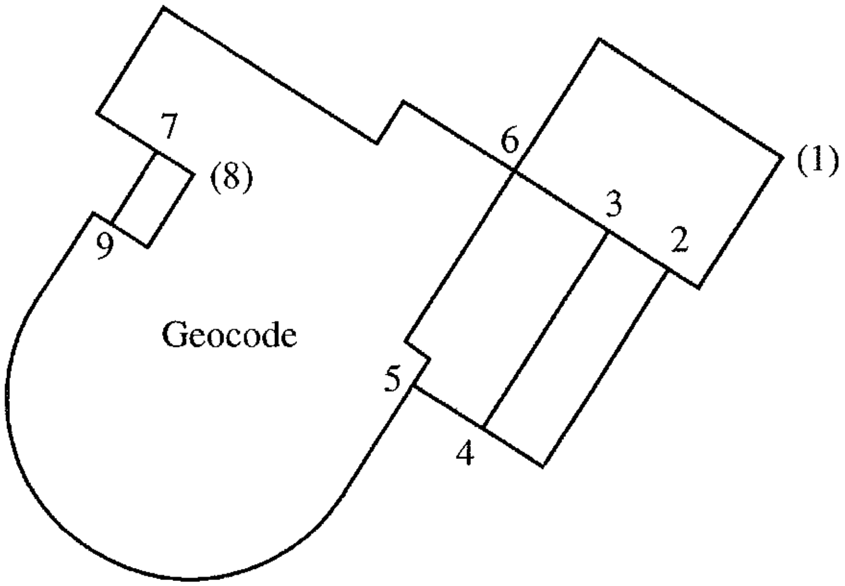
update DWEL_C1 set house_no = (select house_no from DWEL_C2 where DWEL_C1.geocode = DWEL_C2.geocode);

After having filled out all columns of DWEL_C1 we dropped the auxiliary table DWEL_C2.

We see that at this stage the complete textfile contents are linked to the label and can be edited through selection in the graphical file; so, for the areas we now have what we need : (centroid) objects, each with a link to a record in an attribute table and a link to a graphical (text) element.

For line objects though, Intergraph tried to replace the lingeo labels and the associated texts by duplicating the lines as many times as there were functions switched on in the multifeature code and by tagging each of these duplications with one different single (functional) object. This duplication of the graphics was not necessary; Intergraph could have given a multiple feature tagging to the lines. Furthermore, by raising the function of a line from an attribute/text value in the original format to a featurecode in the Intergraph format, they dropped/lost the other attributes of the line, viz dos, doc, dod, sheet and plan, well knowing that it does not make sense to repeat these data for every different function of the same wall. Even when using a multiple feature tagging we would have met the same problem: the geometry dependent data like dos have to be linked with one of the indicated features or with a special supplementary feature, eg, wall in order to avoid choosing between the different functions.

By using the tools linecleaner – merge duplicate linework and featuremaker, eg, wall (defined with an attribute table) we could still have built the following multiple tagged structure for the dataset example:



2–1–6 = wall, general_ot
2–3 = wall, glsb_b, general_ot
3–4 = wall, glsb_b, glsb_b (twice ?)
4–5 = wall, glsb_b
5–6 = wall, glsb_b, dwel_b, non_squ_b
6–7–8–9–5 = wall, dwel_b, non_squ_b
7–9 = wall, non_squ_b

with an attribute table (containing dos, dod and main geocode) attached to the feature wall. (Neither the dataset nor the documentation is showing this, but we suppose that the textual data associated with the lines also contain the geocode of the main dwelling in order to group the line objects per dwelling).

We feel that this structure is too far away from the original OSNI datastructure, which could easily have been reached by making all line elements the same kind of feature (eg wall) and by leaving the functions of the line in an attribute table.

On the other hand OSNI determined the structure of their database at a time when software did not allow automatic topology building. They did a good job by fixing the topology in their data and they disposed of a complete database when we still had to start.

But nowadays their database structure is pushing OSNI to do a lot of unnecessary work: The fragmentation, which implies more data to be filled out, can be avoided and the topology can be built automatically by changing to a structure that respects the principle to always choose the highest possible dimension:

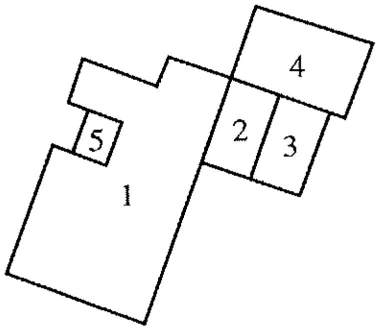
a point is not a point object if it belongs to a line

a line is not a line object if it belongs to an area

an area is not an area object if it belongs to a volume

a volume is not a volume object if it belongs to a?

We suggest that OSNI consider the dataset example like this:



5 area centroids tagged with their function(s)

1 = dwel_c, non_squ_c (2 objects)

2 = glsb_c

3 = glsb_c

4 = general_ot_c

5 = non_squ_c

in which all six objects are linked to their geometry dependent attribute tables (and dos and dod; no doc!) and either just the main object or all six objects are linked to the one and only geometry independent attribute table (or textfile in OSNI terminology ; + dos, doc, dod, geocode, house_no...).

In MGE the first link is inherent in the feature definition ; in Sysdeco the first link may be the geocode of the area + 2 digits for the version, the second link may be the main geocode.

The main attribute table may also have a link to the land parcel on which the house is built.

When the geometry of an outside wall is changed, one part of the dwelling (one object with its area centroid) is deleted and replaced. For an inside wall we need to replace two objects, but on the other hand the conceptual data model has the advantage of avoiding fragmentation : we end up with six 'active' objects whereas the original dataset example had 18 objects.

If ever the history of a part of a certain wall is needed, it can still be cut out afterwards.

Most of OSNI data can easily be transformed into the CDM that we described above. Only the replacement of a series of 'lingeo tables' every time by one single 'areageo' table might show some minor difficulties / choices between different options.

As for updating matters, we can then refer to our answers to the 7 key areas.

Conclusions

The IGN conceptual data model allows updating of a complex topographical database including tables, textfiles and both vector and raster graphics.

We also arrive at outputting all necessary change only information and the client is able to integrate this information into this database.

By "allows", "arrive at" and "is able to" we mean: it is possible to do so by using the Intergraph software packages microstation and MGE, but the software is not really constructed for updating purposes; it should be improved with some real user friendly updating functions.

Our CDM has no problems with archiving, holding and interrogating historical data. It was conceived for periodical revisions, but it can be used for continuous revision.

We recommend that OSNI adapt their database structure to a similar CDM in order to save a lot of work.

NATIONAL LAND SURVEY OF SWEDEN
MAP DEPARTMENT
COMPUTER SECTION

KJELL DEGERSTEDT

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

FINAL REPORT – Part 1 General Discussion 7 June 1993

1. INTRODUCTION

A project with the above title was initiated by the European Organisation for Photogrametric Research (OEEPE) in October 1992. The project had a first meeting at Ordnance Survey of Northern Ireland (OSNI), where the structure and goal of the project were laid out.

Among the many participants is the National Landsurvey of Sweden (NLS) from where this paper emanates. We are taking part in this project to a degree, which is not yet quite clear to us, because of our limited resources. We have however decided to try to contribute at least to the first phase of the project, which to some extent is of theoretical nature.

At the NLS we have during a couple of years developed a geographical information system, which we call the Geodatabank System (GDBS). The GDBS has as it's main purpose to store large quantities of complex digital geographical data. During this work we have encountered many problems which certainly fit into the scope of this project. We also believe that some of the addressed problems are in fact being solved by the GDBS. So in the following we will describe our experiences and ideas in connection with these matters.

This paper will be followed by a part 2 in which conceptual, structural and operational problems will be discussed in more depth.

2. DIFFERENT GIS SYSTEMS

There exist today many commercial GIS systems on the GIS market. (We will be using the term GIS in a broad way also including the production of conventional maps). These systems are all different in many ways, an evident effect of the market forces. What the perfect GIS system looks like is still an open question and will probably be for decades to come.

Because all systems are different a reasonable conclusion would be that some systems must be better than others. Unfortunately the truth is a bit more complicated. Some systems are better than others in some aspects but not in all. In order to be able to compare different systems, you must be well aware of how and for what you want to use the system. This applies to everyone trying to purchase a GIS.

Given a particular GIS there are consequently a limited number of ways to solve a given problem. If the problem is outside of or beyond the original purpose of the system, you cannot be certain of finding an acceptable way to solve your problem.

Many of the GIS on the market were primarily designed for data capture, geometrical editing and map output. These activities constitute of course a very large part of the effort in creating a geographic database, so a system which is good in these parts, will therefore be very useful. But, when it comes to complex manipulation of the data, it can be that these types of systems are not very appropriate many having their roots in the 1970s and early 1980s.

3. THE IMPORTANCE OF GIS CONCEPTUAL MODELS

Behind the visible outer features of a GIS there is always a more or less hidden conceptual model. The differences between separate models becomes evident when transferring data between the systems. Often there is no room in the receiving system for some types of data from the sending system. Especially when it comes to structure, that is, relations between different data items, it can be impossible to transfer the information to the receiving system. This is evident and well known from many practical cases.

The same problem also exists between a given GIS and the user problem space in the real world, but it is not always as obvious as in the case of transfer of files. For that reason this circumstance is sometimes not considered of much importance and is too theoretical to have any practical importance in everyday production.

It is our opinion and our claim that in fact this internal structure is the most important limiting factor in the GIS's of today. This limit becomes evident in two ways.

First, when the user wants to model the features of his real world there are very often obstacles to storing attributes in the desired way. For instance the identifier field may be too small, the precision of co-ordinates is not good enough; the number of attributes is too small and so on. Some of the most modern systems have solved this problem by letting the user add attribute data in separate tables, but then it is sometimes impossible for the user to operate on them inside the GIS in the way he wishes.

Also when it comes to relations, that is complex data structures, possibilities are sometimes too limited.

The second problem is the dynamic part of the system, that is the operations on the data offered by the system. The vendor often claims that there is a lot of flexibility by supplying a so called "macro language". This is of course partly true because the number of possible combinations of operations increases exponentially in this way. But this also increases the number of redundant operations to the same extent and in fact the proportion of redundant operations will also grow exponentially, while perhaps the needed functionality is still missing.

The reason for this short coming is that it has not been possible to map the users' problem space into the conceptual model of the system. A small example will perhaps make this a bit clearer.

If the system contains the elementary operations E1=a1+a2 and E2=a3+a4 it it not possible to execute the operations a1+a3, a1+a4, a2+a3 and a2+a4. The E-operations are the smallest visible operations accessible to the user. At a lower invisible level there are still smaller a-operations. The E-operations are the operations available in the macro language. If the user for instance wants to do the operation a1+a3, he has to write a program of his own in an traditional programming language, that is, he has to extend the system. This can only done if the system has open structure with an interface to a standard programming language and the systems internal datastructure. The limiting factor then becomes the datastructures. What data are stored and in which structures. For instance it is almost impossible to get a system to handle raster formatted data, if raster data is not among the basic datastructures of the system.

The conclusion is that a system's flexibility ultimately depends on what the elementary operations are and what the basic datastructures are, or expressed in another way, what do the dynamic and static conceptual models look like.

4. THE "IDEAL" SYSTEM MODEL

Of course it would be presumptuous of us to claim here that we know what the ideal conceptual model for a GIS system would look like, but there are some promising new ideas that very likely will be fundamental to such a system. We are referring to the "new paradigm" of object-oriented systems. (It is not possible to describe this concept in full here but for a good tutorial on this theme see reference (1)). These ideas have partly been around for a long time but have found a mature form in the object oriented technique, sometimes referred to as the OO-paradigm.

Many old (10 years or more) systems are built in a bottom up way. The construction of the system has started from the most obvious elementary structures, namely the geometric objects, which are well known since the days of Euclid. These elements are very central to the production of maps and especially digital maps. This is where the digital mapping technique started. Later came the need for connecting attributes to the geometric information and the digital map information databases gradually changed to be geographic databases with more general application areas. This is a well known development and is especially motivated by GIS applications.

To be a bit more precise consider the following example.

In the bottom up approach a line is described as the division between, for instance, a dwelling house and a glass building. This is usually described at attaching feature codes to the line. Sometimes it is also said that the dwelling house is to the left and the glass building is to the right. With this structure it can be difficult to answer some questions and perform certain operations like:

– Give me a particular object or all objects having this line as a border!

This problem becomes still more difficult if there are many objects having the line as a part of their geometry. It is also awkward to connect data to the line when they in fact describe properties of the corresponding objects.

In the OO-technique the approach for this type of problem is normally top down. That is, there are in this case two objects, a dwelling house and a glass house, each carrying some properties of their own and they happen to have a line as a common part of their geometries. This way of looking at the objects as independent entities with some geometric, topologic and semantic relations is a more effective way of describing the users' problem and creates a wealth of possibilities. In particular interrogation of the objects can be defined in user friendly terms having responses in keeping with the user's expectations.

5. A DIALOGUE WITH THE "IDEAL" SYSTEM

Here we shall try to imagine what an "ideal" dialogue would look like if we could forget about the constraints of today's commercial products and instead had access to a very flexible system, which presumably must be built on the OO-principle.

In the following the computer (the system) will ask and answer questions in different ways. This can be done actively, the computer prompts the user for an answer, or passively, the computer expects the user to pull down the right menu or view. In all cases the ideal system should let the user answer in the way most appropriate to the actual situation. For instance the user could choose to identify an object by pointing at it on a map, by giving it's identification in a pop-up menu, by picking a value from a pulled down list or by perhaps answering a voice recognition interface. These are all properties of the systems user-interface, which is a completely independent part of the system and which can change in line with ongoing technological development. In future we will expect all GIS-systems to have this isotropic type of interface. The interface however is only the means to interact with and carry out operations on the system. How such functions are performed is dependent on the systems conceptual model.

In the following we represent the computers passive or active questions in ***bold italic*** letters to be interpreted in the generalized way above. The users part of the dialogue is immediately following in *italics.*

Consider now the familiar problem of extending a dwelling house with a glass house. The dialogue could be something like this:

What do you want to do?

The user answers that he wants to do the operation "extend" by pulling down a menu.

What do you want to extend?

The user pulls down a list of object types and points at the object type dwelling house, writes the words "dwelling house" in a field in a pop up menu or points at the map in another window on the screen. When he has chosen this object a form appears on the screen with all the attributes of the picked object giving the user the opportunity to fill in or change the attributes of the object.

What do you want to extend the dwelling house with?

Once again the user chooses the right object, which in this case is a glass house, and is also given the possibility to indicate the attributes in the same way as above.

Are there common geometric parts?

The user points out the common part(s) on the map on the screen under the control of the system.

Now the whole operation is completed and the computer finally asks a question like:

OK? Cancel or accept?

Of course this dialogue could be modified in many ways, for instance the user should perhaps be able to create the glass house during the operation "extend", which amounts to allowing simultaneous actions, which perhaps is not so unreasonable to ask for in a multi-window environment. As extra advantages in an OO-system there are a lot of possibilities to check for consistency and integrity of the operations. For instance the operation "extend" would only be possible to perform on certain object types. This feature alone will eliminate a lot of possible mistakes.

The conclusion we want to present with the above reasoning is as follows. When looking for a best way to operate on complex geographic objects it is very convenient to free oneself from the restrictions of many of today's commercial systems and try to describe the operations and the objects in a conceptual way like above using the OO-paradigm. The solution found in this way will be much closer to the users application area and contain fewer, more natural and safer operations. The system will be easier to understand and more predictable for the user.

6. APPROXIMATING THE "IDEAL" SYSTEM WITH A GIVEN COMMERCIAL SYSTEM

Suppose we have according to the preceding chapter found a good conceptual model for how we want to do our complex operations. Then the questions arise:

- Can we use this information to find a better way to utilize the given system?
- Would it have been better to stay inside the given system's restrictions instead and try to optimize it's use?

For a pragmatist the talk about the ideal system can seem to be too utopic to be of any value. The right way is of course dependent on what you want to achieve.

Each participant should try to implement the operations in his own system. For us this is our own-developed system Autoka.

In our case we have the advantage of complete control of the source code and our system also includes a macro language. Autoka is also, if not object-oriented, at least to some extent object-based. We could therefore with some extra work come rather close to the ideal dialogue. But if we then later want to move our findings over to a specific given system, we will face a problem of similar nature to map the "ideal" system directly into that system. This "cognitive gap" or "system impedance gap" between two different systems is always there, and we believe that it is best handled by people that are experienced with the given system. We do not believe that a user completely new to a system can by luck or because of a fresh view of the system be able to find a solution that an experienced user has not been able to find. The reason is that in our experience it takes a lot of time to learn a modern GIS-system in such depth, as to be able to explore all it's hidden possibilities.

7. CONCLUSIONS

In modern GIS-systems complex datastructures have become more and more common. The problem of finding new and better ways of creating and updating such structures is one of the aims of this project. Each participant should experimentally look for new solutions by moving some data from the OSNI database into his own system. Operations on that data will in total involve a lot of work. We do not believe that this is a cost-effective way to work. This will perhaps only show, that some operations can be done in one of the systems but not in the other. The risk is also that the discoveries will only be of trivial nature and also perhaps obsolete, because we are working with second generation systems, and the third generation is waiting around the corner. This can be said even about such a famous system as Arc/Info which many people believe will be a market-leader forever.

Our opinion is that we should do with much less experimental and time consuming practical and use instead more theoretical effort, in trying to define in conceptual terms, how we would like our GIS-system to be in a unrestricted but still very precise way. This model could be the basis for a broad debate and also serve as a guideline for the GIS vendors to improve their systems for the benefit of GIS users. One common problem for many of the GIS vendors is that they have very good resources and experiences in software science, but for obvious reasons much less knowledge of the photogrammetric, cartographic, geodetic and other disciplines involved in map production. Therefore we believe what they need are good conceptual descriptions that can serve as specifications of requirements for new enhancements and developments.

Literature

Max J Egnehofer and *Andrew U Frank*: Object-oriented Modelling for GIS. URISA Journal nr 2, 1992

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

NLS – SWEDEN

Part 2: Complexity of structures and operations

1. INTRODUCTION

This article continues the discussion from part 1 and continues to claim that the conceptual model is of fundamental importance for the users perception of the system.

We will also try to describe what is meant by complex structures and complex operations in topographic databases. Everybody has intuitively a feeling of what complex structures and operations are, but if you try to define what they mean more precisely, you will find that it is not so evident. Perhaps it is necessary to have these precise definitions in order to understand how to simplify structures and operations. One of the most complex operations you can perform on a topographic database is updating, which is the topic of the OEEPE project. Therefore our discussion will hopefully shed some light on this problem.

We will also use terminology and concepts from the object oriented theory, because it is a very convenient way to describe these type of problems and also, it seems, a good way to describe the solutions. That does not mean that we believe that OO theory is the magic remedy which will make these problems disappear, but there are certainly strong implications that object orientation (OO) will be a good help both in designing and implementing future GIS systems.

Most existing software packages (here called GIS-systems) for handling and building topographic databases are not implemented with object oriented techniques. We will despite that, express our belief that these ideas can be partly applied to traditional GIS and also be used to enhance them. The ideas and experiences referred to in this article emanate partly from practical experience at the National Landsurvey of Sweden (NLS) with developing and implementing our own system for storing large geographic/topographic databases. The system is called the Geodatabank system (GDB) and is part of the NLS general software, Autoka, occasionally referred to in the following. GDB has been in use for over a year but is still developing using of some of the ideas in this paper.

2. COMPLEXITY OF STRUCTURES

We are using the words "simple" and "complex" when we talk about data and information structures. The question is, what is the exact meaning of these two notions. When does a structure cease to be simple and start to become complex? We have intuitively a feeling for that, but for understanding how to simplify complex operations, we might know more precisely what makes a structure become a complex one.

If there is a lot of labour involved in creating a certain data structure we do not necessarily say that the structure is complex. It is laborious but not complex. Complexity has to do with something else. If, on the other hand, it is hard to create the structure correctly, we are more likely to call it complex. This difficulty is due to it being hard to understand the structure. If a structure is hard to create and/or hard to understand, we say it is complex. This complexity mostly has to do with there being a lot of facts and rules to be kept in mind simultaneously for understanding of and for operating on the structure.

3. COMPLEXITY OF OPERATIONS

The operations of concern are primarily the updating, or often equally, the creation of datastructures in a topographic database.

The complexity of these operations is not necessarily associated with the complexity of the corresponding datastructures. You can have complex operations on simple structures and simple operations on complex structures. However, there is normally a strong correlation between the complexity of structures and the accompanying operations. Most complex operations are due to the complexity of the structures operated on. Other reasons are that there are many rules involved simultaneously in the operation and the one-shot nature of the operation. With one-shot we mean, that the user has to make the operations right first time. The situation then becomes very difficult, and the system becomes almost impossible to use. In reality there are not so many pure one-shot situations, but it can be very hard to repair a mistake, which in fact is an obstacle of the same nature.

4. REDUCTION OF COMPLEXITY

The complexity of topographic data and information structures is dependent on the complexity of the features in the real world being described by these structures and the amount of semantics about these features we want to keep in our database. Unfortunately this reality is in itself very complex. The level of ambition in current information systems is rising all the time. So topographic datastructures will of necessity be complex. That does not mean that we should not try to simplify them. On the contrary we must strive for simplification otherwise the complexity of the structures will grow out of control. A similar argument holds for the associated operations.

If we cannot reduce the intrinsic complexity of the structures and operations, then what can be done to help the user? The answer is the well known technique of *information hiding*, which means hiding the complexities from the user as far as possible. The system must take care of as much as possible of the (interdependence) rules and the facts. In the following we will propose some possible measures to be taken to attain simplification.

When the user creates or updates a topographic (compound) object he normally puts a lot of facts into the fields of a form. If there are many fields in the form the operation becomes laborious, a different aspect from complexity.

- The facts have to be filled into the form. There is no way around that. The only thing which can be done to simplify this operation is to have the system fill in as much as possible automatically. That is enable the system to make "intelligent guesses" about what these values can be. The standard way of doing that is to fill in as many default values as possible in the form. "Intelligent guesses" means that they should be done in a context sensitive way. The system must know as far as possible which default values are acceptable in the actual situation, eg, by having a default form for each object type at hand, which perhaps can be created automatically by the system by collecting information continuously. A simple variant is to keep the values from the nearest form of similar kind. Also the user can be relieved from typing by presenting selection menus. Of course, sometimes such simplifications are not possible at all, but it is a rare case. Normally a lot can be done that way.
- The fields of the form must be able to fill in any order. Of course it can be a disturbance for the context sensitiveness, but if some parts of the form must be completed in a certain sequence, the system must tell the user that. The system should take care of the ordering rules and other involved rules.
- The user must be able to change anything in the form as many times as required by moving back and forth (scrolling) the form. The form is not finalized (eg stored in the database) until the user explicitly gives the command "Store!". This is nothing more than the well known concept of a *transaction*. So the system must be able to do transaction handling.
- Even if a transaction is completed the user should be allowed to cancel the transaction. It can be very hard, perhaps sometimes impossible, to achieve that, but the user must have this possibility as often as reasonable, at least with necessary restrictions.
- If the operation concerns several related objects that have to be updated simultaneously, the system should make it look like one single compound object. That is, the whole operation should be done in one single form obeying the rules mentioned above. The user should be allowed to scroll back and forth in this form in a seamless way not having to bother about which of the involved objects he is operating on for the moment. To enable that, the system will probably have to keep a working copy of all involved objects throughout the whole transaction. In this way compound objects can be handled in a similar way to simple objects, which is probably as far as you can go in simplification of this type of operation. In this situation it becomes particularly important that the systems know all the rules concerned to prevent the user from making intricate mistakes.

Of course all these demands together raise the level of ambition a lot if you compare with many of the contemporary systems in use. This is however what is needed if there is to be a decisive simplification of the complex operations.

5. EXPERT INTERFACES

In addition to what had been said in Chapter 4 there is still another demand which is a part of the systems context sensitiveness.

There must be different interfaces for different types of applications. A modern way to express this is by referring to expert or knowledge based systems. As an example suppose you are updating a large scale cadastral topographic database. Then the system should "know" what is meant by all concepts involved like plot, real estate, border etc. If somebody else works with a small scale topographic database they should have another interface. Of course there are many similarities between these two interfaces. The fundamental differences are that some rules and some concepts are different. For the user to be allowed to work comfortably, the system will respond to the different demands by offering dedicated interfaces.

In this way the user will also have a simplified view of the database, which is much more adapted to his own problem area. If the user is working with dwelling houses, there should be an object called dwelling house in their view of the database behaving in accordance with the rules applicable to dwelling houses.

6. IMPLEMENTING THE INTERFACES

Of course the implementation of interfaces of the type mentioned will consume a lot of human resources with accompanying costs. But this investment is what we expect the vendors in the GIS market to make, to stay in business and keep up with competition and the constantly growing demands of the user community. The users expectations are of course well grounded. In the users own disciplines the collection and updating of the necessary databases are very time-consuming and very costly. There is a real need for simplification and the vendors must be made aware of that.

There are fortunately some factors that make these goals more attainable for today's vendors. We are thinking of the new software development platforms with object oriented techniques (OOA, OOD, OOP) and case tools for developing graphical user interfaces (GUI's). With these new tools the effort is greatly reduced compared to the 3rd generation procedural techniques.

OOP especially seems to be suitable. By putting both the behaviour and attributes into typed objects (object classes) and using inheritance to model similarities between different contexts the system will become very flexible. An object will follow all the rules and have the right type dependent behaviour in whatever context it is used or in whatever form it appears. It is much harder, perhaps overly ambitious, to accomplish this with ordinary 3rd generation languages like C or Fortran.

It can be contended that this type of extension has always been possible with the macro languages contained in commercial GIS-systems. If that is the case, then why has somebody not done it? The need is urgent and the macro languages have been around for a long time. There have been a lot of things done with macro languages to date, but a really good solution for the complex update operation has not yet appeared. We believe there are certain reasons for this.

Macro languages are powerful but not powerful enough because of the following reasons:

- Macros are mostly embedded into macro scripts. If you want to reuse a macro in another macro you normally have to copy the code into the new script. Sometimes it is possible to make a call to the other script, but in such a case you can have a conflict between local variables. In short, the reuse of macros is generally not so easy.
- Some systems do not have macro systems, or the macro feature is very limited, which means you have to involve third generation procedural languages like C or Fortran. Then you are caught in the trap again!
- Macros are perhaps good for creating new operations, but if you want to introduce new datastructures it is not just as easy. So it can be hard to create a new user view of the system.

The OO-technique can reduce these obstacles in the following ways:

- Reuse is much simpler. Every object is carrying it is own information which is completely independent of other objects and therefore always callable. The reusability is also strengthened through the inheritance feature. For instance you could create an object class called building and two other classes, dwelling house and glass conservatory, having some features inherited from the building class but also some unique features of its own. You can also introduce an operation (method) called "extend building" which behaves differently if you extend the dwelling house with a conservatory or with a garage. This type of reuse is one of the most important advantages with the OO-technique.
- With OO it is possible to build a new conceptual shell around an old system, at least to some extent. End users can have their own expert interfaces with the concepts and the terminology they are used to. Perhaps this is a way to upgrade the existing 2nd generation systems. This kind of upgrade can perhaps also survive a future change to another commercial system because the user can stay with the expert interface he is used to. The economic and other advantages with that are obvious.

But who is going to create this new "marvellous" system? The difficulty has moved from being one of implementation to being one of analysis and design. There is a lot of knowledge involved in this creation process and and it is not likely that one single person has got it all. A project with a well composed collection of people must be created with the accompanying problem of fundings, etc. Also somebody or something must make these persons work well together. This is the difficult problem of group dynamics. Making things like that happen is what we expect the vendors to do, and perhaps it is also what they are trying to do all the time. So once more, why has that not been done already?

(Sometimes this is a truism that can be said about anything, but nevertheless it is relevant, when you are looking for the main hindering reasons in this case). One reason can be that there is a lot of experience and knowledge on the user side which nobody has been able to communicate to the vendors. This is why it would be important to try to describe the ideal system in a way that is liberated from the straight jacket of 3rd generation procedural thinking.

7. HISTORICAL DATA

There has emerged in recent years a new interest in historical data in the GIS/LIS application, especially in certain application areas like cadastral systems. Most of the current GIS and mapping systems are not constructed to handle historic data. A common opinion is that the systems can always be made to store historic data by adding a user attribute carrying a time stamp. Here we are claiming that for the system to have complete support for handling historic data, it has to have a time dimension, or in other words, be based on a temporal conceptual model and a temporal database because of the following arguments.

The spatial co-ordinates x, y and sometimes also z, are always implemented as implicit attributes of the system's basic model in contemporary GIS. The reason for that is that the system needs to have full control of these attributes to be able to perform certain operations in an effective way, operations like spatial retrieval and overlay. If x and y had been user chosen attributes, it would have become very difficult, if not impossible, to implement the GIS.

For exactly the same reason it is most convenient that the database is temporal. This means that time should be one of the basic intrinsic attributes in the datamodel like the x,y co-ordinates. All operations involving time, like keeping historical data, change only updates and optimistic locking procedures, will be very much simplified and can also be handled in a more rigorous and consistent way.

Another thing that is necessary for keeping the history of an object is a labelling system that gives every object an unique time persistent identifier. Again, for almost the same reasoning as for the time co-ordinate, the labelling system should be one of the basic automatic features of the system. This type of labelling system can also solve other problems, like the need for pointers in compound objects and other relations.

Both these features are implemented in NLS's GDB system and have been found to work very well.

8. FRAGMENTATION OF GEOMETRY

When keeping historical data of complex compound objects, the geometrical parts present a particular problem. The question is, when a line is partly updated should the history tell exactly which part of the line was changed at a particular moment? Things become still more complicated if (multiple) object codes and attributes are also taken into account. The structure can become immensely complicated, as time goes on. Is there really a good cost/benefit case for this extreme fragmentation of the geometry? Does the user need this information? The problem can be solved with the same pointer technique as for other types of relations but the resulting structure will be more complex, error prone and will degrade the performance of the system.

In the GDB system we have stayed at a lower ambition level, because we believe that the benefit of this feature is very low. We are keeping the history of every object and can, for any point in time, tell how the geometry of a certain object looked, but not the relation to the earlier version of the geometry. This technique also overcomes the problem of geometrical properties common to several objects. The compromise lessens fragmentation of geometry and makes it easier to cope with.

9. ARCHIVAL STORAGE

The problem of keeping a large archive for the topographic databases has got a new solution recently, which at least to us at NLS, seems to solve this requirement satisfactorily for the foreseeable future. In our case it entails using a new type of hardware, optical jukebox storage, which in the version we have is capable of storing 1 Terabyte. The total economic map of Sweden, which is estimated of about 30Gb, and is the biggest topographic database in vector form we will build, will thus occupy only 3% of that storage.

However volume capacity is not the only favourable aspect. Another very good feature of our "terabyte" storage is that it works as an ordinary Unix filesystem, which means the NLS's GDB-system runs on it without any change and that this external storage is completely transparent to the user. The access time is of course larger than on ordinary disks, but on the other hand very much faster than a traditional magnetic tape archive. For these reasons we consider the choice of archival storage is no longer a problem. The system has not been used extensively because we have not yet moved our software to the Unix operating system.

10. CONCLUSIONS

There has been a considerable spread of GIS and other computerized mapping systems around the world during the current decade. Many institutions have managed to build very large digital topographic databases with these systems. Most of these systems can be considered to be of the second generation and contain features that are beginning to become archaic. Because of the large amounts of data collected in the formats of these systems the users often consider it almost impossible to change over to other systems. At the same time they observe that there are many new tools and techniques they would like to use.

In this paper we have discussed this problem and proposed a way to migrate smoothly into new techniques. The proposed route is to build new interfaces with the aid of the new techniques. By using OO methods and tools these interfaces will look more like expert systems, with a more powerful support to the user taking care of a large part of the complexities in the system by hiding them. The expert interfaces will give the user a new view of the database, which is a more natural conceptual model and thus easier to work with. Another advantage is that, because this model is a better description of the user's application field, the model, the user's knowledge of the model and the tools can be reused, even if the underlying commercial database system is eventually changed, which certainly is going to happen in the long run.

UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

NLS – SWEDEN

Part 3: Summary of results

1. INTRODUCTION

The topics of the OEEPE project addressing the above problem are of central and general interest for people concerned with updating of digital topographic databases. It has been of interest to the National Landsurvey of Sweden to follow and also to participate in the OEEPE project. Unfortunately NLS has not been able to allocate adequate resources to make a deeper study.

However, NLS has during recent years pursued a large project, the development of a system for storing and updating topographic and cadastral digital databases, in which there are many similar issues.

We have reported our findings in the three referenced papers. Part of the material is perhaps not strictly within the scope of this project, but we believe that our contribution will at least be of interest and perhaps give some new views on the problems. Our experimental efforts have also been very limited but will hopefully answer some questions.

In the following text we will try to give answers to the specific points in the project initiation document using our experiences and the material we have collected during the project.

2. UPDATING THE OSNI DATA

We are using several software packages for handling topographic databases at NLS. For several reasons we chose Arc/Info for this project. These were as follows:

Arc/Info has a very broad userbase, perhaps the largest of all GIS-systems, which among other things means that it is comparatively easy to get support. Inside NLS there are about 20 people familiar with Arc/Info. ESRI/UK translated the data from SYSDECO format into Arc Export format.

An important prerequisite for using Arc/Info was the existence of Version 6 of the system which has been available for about a year. In our opinion it would have been impossible to use Version 5. The reason is that in Version 6 ESRI has introduced the "cursor" concept. This concept is well known in SQL language, but it has not been available in Arc/Info until Version 6. It is not an ESRI invention. With the aid of cursors it was easy to handle the multiple feature codes associated with the line and polygon objects in the OSNI data.

This illustrates one of our main points. By adding a new concept to a known system, the system is given powerful new possibilities which are impossible or very timeconsuming to implement with the original system. Arc/Info has a very powerful and flexible macro-language but that would not have been enough without the cursor concept. So the conclusion is that finding the ideal system is very much a search for suitable concepts.

With the help of the AML language, the Arc/Edit program and cursors we implemented a small application program for updating the OSNI data. This took us about three days. The application uses menus and "radiobuttons". It does not cover all the issues, but we estimate that a complete application taking care of all the update cases, could be implemented in about two weeks.

However two things would still be missing, the handling of historic data (historic versions) and the change only updates. The latter could be introduced by an appropriate selection procedure. This would add some more days to the implementation. We have found a way to store historic data automatically in a separate layer during the update process, but no easy and practical way to handle time dependent versions of objects. This also confirms our theory that the only way handle historic data easily is to introduce the concept of a time dimension in the system.

Arc/Info has a built-in seamless database. The data are divided into smaller sections, so called tiles, which can be map sheets if desired. The user does not have to be aware of that and the data can be used in a completely seamless way. The AML – language fully supports handling attribute data and consequently this was no problem for us.

Of course our prototype application can be refined in many ways to be more modular and more directly relevant to the users application. The time needed for such implementation is hard to estimate without actually doing it and also very much dependent on the level of ambition. We believe that a couple of months would be reasonable for creating a routine acceptable in daily work.

With our prototype AML-program it takes about two minutes for a user to perform the update in the example. We do not know if this is acceptable for practical use. With a more sophisticated routine this time could perhaps be reduced to one minute. Is this is a sufficient benefit to make the development effort worth while? There is certainly a lot more to gain in a more complicated case. To find such a solution however demands a lot more analysis and ergonomic studies.

We have not used the topological features of Arc/Info in our prototype AML-program. It would certainly be more convenient and more effective to use these features, which are one of the strong points of the Arc/Info software. By using topology the multiple line feature coding could have been eliminated reducing the users manual work considerably. Also more automatic snapping procedures could have been utilized avoiding the user having to divide lines into segments.

All textual data and also all date data were handled as ordinary attributes in AML. Arc/Info has a special data type for dates which makes things a bit easier. Our routine does not contain any automated date tagging. The user has to do this interactively, which of course is not ideal. As stated previously time should be a dimension automatically handled by the system.

3. DATA OUTPUT

There are a lot of options for drawing complete map sheets in Arc/Info. These are probably satisfactory for most purposes. We have not looked closely at this problem. There are also many file transfer formats, so it is a routine matter to output data to users of these formats.

There is no standard procedure for change only data in Arc/Info. Such a routine has to be implemented. Because there are no general identifiers retained in the system it can be a difficult problem to provide a customer database with repeated change only updates. Our belief is that this feature should be built into the original system.

4. ARCHIVAL SUBSYSTEM

Traditionally the archival subsystem is a separate system from the topographic database management system. This is not an ideal situation because retrievals from the archive then have to be handled in a different way from ordinary current data queries. Often a manual routine is involved. To integrate these two routines into a seamless system, transparent to the user, is a very large implementation problem.

Fortunately a new way to solve this problem seems to have appeared. At NLS we are presently looking at the new optical mass storage technology, the "terabyte" memories. The one we are testing is called "Storage Server" from Digital. We are going to use this hardware in connection with our geodatabank system.

The great advantage with this type of hardware is that you do not have to treat your historic data differently from current data. The mass-storage is just an "indefinite" extension of your ordinary disks. To retrieve historic data is just asking for another time interval in your query.

Of course there can be a limit to how much data you want to keep on line. Even that situation can be handled with the same method. Obsolescent data are moved from magnetic memory through magneto optic memory to end up finally in optical write once memory. This last copy can be removed from the online system and handled manually on demand from the retrieval routines.

5. CONCLUSIONS

To summarize our findings from the above and from the references, these are the main conclusions:

- It seems to us that it is quite possible to use Arc/Info to perform the required updates by writing some extensions in the macro language.
- Archival storage problems could advantageously be solved by using new mass storage technology.
- To get beyond the limits of today's systems we have to look for new conceptual models.

- A difficulty with that policy is that there are a lot of different opinions on what the ideal conceptual model should look like. There are many different proposals in the literature and in the many commercial systems from different vendors. It seems necessary to formulate criteria which can be used to evaluate different models.
- The way to enhance the operator's performance when updating and collecting digital topographic data, is to let the software automatically handle as much as possible of the operator's work. The new technology of object orientation seems to be promising in this respect because it is semantically more powerful than traditional third generation techniques. That is our belief at NLS in spite of the not very impressive performances of commercial object-oriented GIS systems to date. However in the very latest of these systems (Smallworld) we find ideas in conformance with our beliefs. We are going to test these ideas by building an OO-interface to our geodatabank system with the object oriented language Eiffel.

REFERENCES

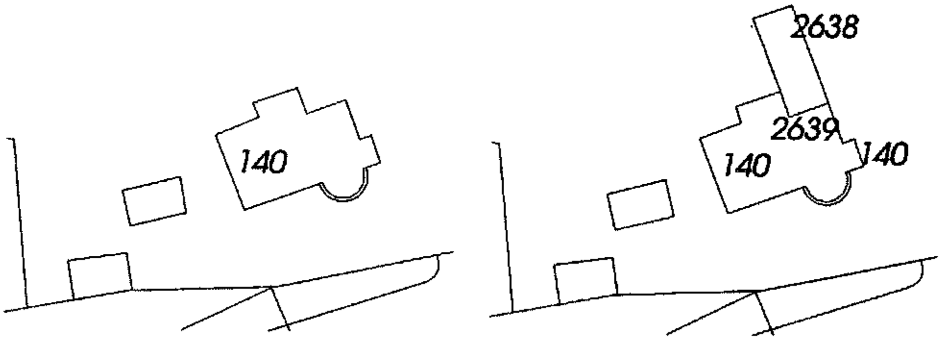
1. *Degerstedt, Kjell*: Updating of Complex Digital Topographic Databases, Part I.
2. *Degerstedt, Kjell*: Updating of Complex Digital Topographic Databases, Part II, Conceptual models.
3. *Westgrds, Ingemar*: Updating the OEEPE Dataset with Arc/Info.

Lantmateriet
Marknadsavdelningen
80182 Gavle
Sweden

ADDITIONS AND DELETIONS WITH ARC/INFO

First Update

The following illustrates the code for the dwell_house before and after updating with a glass_building.



1. Digitize a line with the code for *glass_building* and connect it to a line with the *dwell_house* code. The line between the connections gets a new user-id and the codes of both lines (multi-coding).

   The codes are stored in a table, which contains the user-id and the feat_code. It is a one-to-many-relationship with no practical limits to how many codes an object may get.

   This is also how the testdata we got was organized.

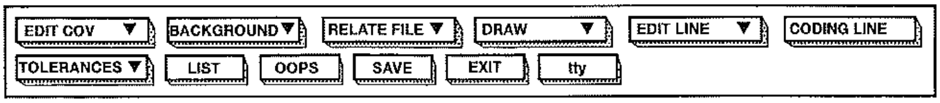
   (a) The related code-table for the current building before updating:

   Arcedit: list cov3d-id feat_code long_code
   Record COV3D-ID FEAT_CODE LONG_CODE
   4921 140 1022 *dwell_h*
   4922 140 1024 *non_squ_*

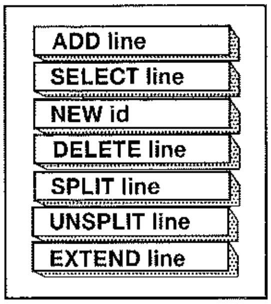
   (b) The part of the building, which is common to the glass-building gets a new cov3d calculated into a copy of the "old records" and the glass_b-code.

   Arcedit: list cov3d-id feat_code long_code
   Record COV3D-ID FEAT_CODE LONG_CODE
   4923 2639 1022 *dwell_h*
   4924 2639 1024 *non_squ*
   4925 2639 1025 *glass_b_*

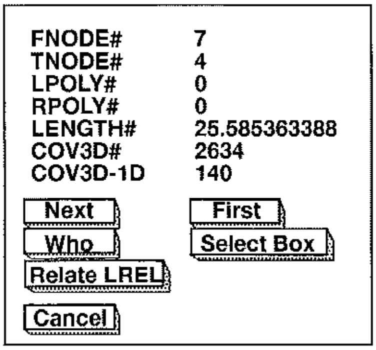
The requested commands are collected in macros and form-menus for effective handling. The following are some examples of menus created for this test.

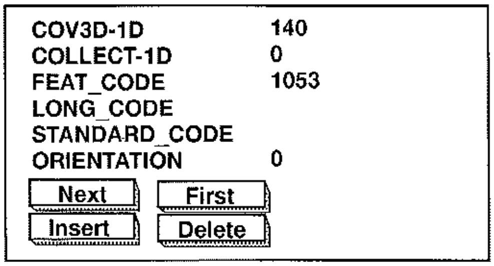


EDIT LINE



CODING LINE





These menus contain the macros you need

When you press the *"NEWid-button"*, the following macro is executed:

```
/*nyid.aml
/*
&s.cov[entryname %.ecov%]
&if[show editf] = ARC &then
&s.maxid [show maximum id]
&s.maxid = %.maxid% + 1
&ty Select one ARC...
SEL
&if[show number select] = 1 &then
CALC %.cov%-id=%.maxid%
&else
&ty Ingen träff...Ändra söktoleransen..
&return
```

The previous version of the "cov3d-id"-object may be moved into a "CHANGEREG" with a "date of change" (DOC) to save the history. Every modification to an object may follow in the same way.

Arcedit: select one
Point to the feature to select
Enter point
Arc 2660 User-ID: 140 with 4 points
1 element(s) now selected
Arcedit:put CHANGEREG
Copying the arc(s) into /HOME/SIGMA/CHANGEREG...
1 arc(s) copied
Arcedit: edit CHANGEREG
The edit coverage is now /HOME/SIGMA/CHANGEREG
Arcedit:editf arc
1 element(s) for edit feature ARC
Coverage has no COGO attributes
Arcedit: sel DOC = "
1 element(s) now selected

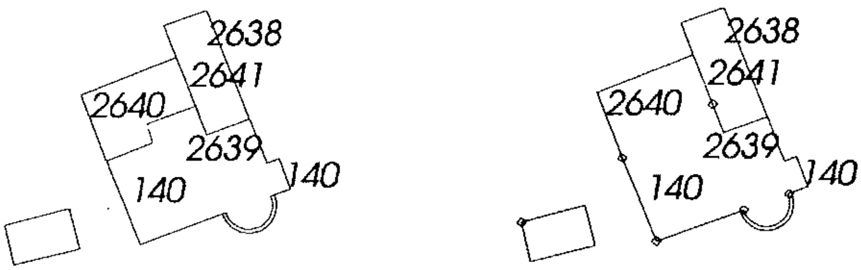
Put in the "change-date" in the attribut-item DOC

Arcedit: moveitem [quote [date – vmsdale]] to DOC
Arcedit: list

FNODE#	= 23	
TNODE#	= 29	
LPOLY#	= 0	
RPOLY#	= 0	
LENGTH	=	11.960
CHANGEREG#	=	1
CHANGEREG-ID	=	140
DOC	= 30–AUG–1993	

Second Update

The dwell_house is extended, as following example shows, with the new objects, nr 2640 and 2641. The old line, a part of nr 140, should be deleted.



The main difference to the previous example is that an old line should be deleted, but the date of deletion should be saved.

A way of doing this is to save the deleted object in a separated "cover", named KILLREG.

Following commands are executed when you press the button: DELETE line

Arcedit:select one
Point to the feature to select
Enter point
Arc 2660 User-ID: 140 with 4 points
1 element(s) now selected
Arcedit: put KILLREG
Copying the arc(s) into/HOME/SIGMA/KILLREG...
1 arc(s) copied
Arcedit: delete
1 arc(s) deleted
Arcedit: edit killreg
The edit coverage is now /HOME/SIGMA/KILLREG
Arcedit: editf arc
1 element(s) for edit feature ARC
Coverage has no COGO attributes
Arcedit: sel DOD = "
1 element(s) now selected

Put in the deletion-date in the attribut-item DOD

1 element(s) now selected
Arcedit: moveitem [quote [date-vmsdate]] to dod
Arcedit: list

FNODE#	= 23	
TNODE#	= 29	
LPOLY#	= 0	
RPOLY#	= 0	
LENGTH	=	11.960
KILLREG#	=	1
KILLREG-ID	=	140
DOD	= 30–AUG–1993	

CONCLUSION

Most of the commands (probably all) you need to digitize, update, delete, multi-code and date-handle of objects like lines, points, areas and text, already exist in most established systems.

In Area-handling it is important to have a topologic structure of data to get benefits of data-analyzing and control.

The system may also have a built in macro-language, as AML in Arc/Info eg, a flexible way of making user-friendly menus for every needed purpose.

If Arc/Info can save the history of the updates as well as an object oriented system is hard to say. The coming version of ESRI's librarian, called ArcStorm promises to support version-handling. Perhaps it is an answer?

NLS of Sweden
Ingmar Westgrds

PROJECT – UPDATING OF COMPLEX DIGITAL TOPOGRAPHIC DATABASES

FINAL REPORT OF TURKEY

HAYATI TASTAN (CAPT MSc Eng)

28 JANUARY 1994

1. ANSWER TO THE KEY AREA NUMBER 1

"How an updated, fully structured and attributed digital graphic, conforming to the full OSNI database specification and free of map sheet line constraints, can be arrived at".

The method developed in answer to key area number 1 via Intergraph MGE software is based on the following steps:

1.1 Project Creation

A project called %OEEPE is created and related descriptive information written.

1.2 Project Schema Creation

In order to have default relational database tables such as feature table, category table, mscatalog table, etc, required for the project a project schema with the name %OEEPE_s is created.

1.3 RIS Schema Creation

Since MGE communicates with the database management system (DBMS) via the relational interface RIS, an RIS schema is created. Informix is used as the DBMS.

1.4 Category Building

In MGE each graphic data file (eg dgn file) is supposed to be in a category. Therefore for each graphic data file a category is built.

1.5 Feature Building

Two features of boundary type are created which are "dwel_b" for a building and "vege_b" for vegetation. Related DBMS tables "dwel" and "vege" are built, having required attributes, and then linked to the features dwel_b and vege_b, respectively.

1.6 Introduction of Textual OSNI Data into the Database

Via an SQL statement, textual OSNI data files for each feature are loaded into the database.

1.7 Introduction of Graphic OSNI Data into the Database

Having first been converted to geographic features via the MGE command "feature maker", graphic OSNI data are loaded into the database.

1.8 Linkage between Graphics and Textual Data

Using MGE Geo Database Locate function each cartographic object having the text centroid value equal to the geocode field value in the database record, is found by the operator on the graphic screen, pointed to by the cursor and linked to the database record with the "link to graphics" function of the MGE software.

2. ANSWER TO THE KEY AREA NUMBER 2

"How existing textual records linked to the graphic should be modified and how the data dealt with in the course of updating should be tagged with the date of addition/modification/suppression, held, maintained and presented".

The answer to the key area number 2 is explained as follows:

2.1 Updating of OSNI Graphic Data

The term "updating" is assumed as addition, modification or deletion. Therefore if we update graphic data we either add, modify or delete them. In order to update OSNI graphic data we scanned OSNI map sheets via our OPTRONIX 5040 raster scanner in RLE format and then heads-up digitized "change only" data on the screen. In other words we add new data, modify existing data or delete data which are longer extant.

2.2 Updating of OSNI Textual Data

When we add new graphic data we also add related textual data. In case of modification of graphic data we also modify textual data if necessary. If we delete graphic data we delete textual data linked to the graphics as well. If there is no change in graphic data but changes in textual data exist, then textual data only are updated.

2.3 Presentation of Graphic and Textual Data

Both graphic and textual data are presented on the graphic screen. Textual data can be written via line printers whereas hardcopy of graphic data can be obtained using electrostatic plotters and pen plotters.

As for the terms "deletion/suppression date, change only information, replacement map sheet, archiving of information no longer extant and historical points", it has not been possible for us to answer these key areas because the Intergraph MGE software is not able to roll back the geographic data (graphic and textual).

LIST OF THE OEEPE PUBLICATIONS

State – March 1995

A. Official publications

1 *Trombetti, C.:* „Activité de la Commission A de l'OEEPE de 1960 à 1964" – *Cunietti, M.:* „Activité de la Commission B de l'OEEPE pendant la période septembre 1960 –j anvier 1964" – *Förstner, R.:* „Rapport sur les travaux et les résultats de la Commission C de l'OEEPE (1960–1964)" – *Neumaier, K.:* „Rapport de la Commission E pour Lisbonne" – *Weele, A. J. v. d.:* „Report of Commission F." – Frankfurt a. M. 1964, 50 pages with 7 tables and 9 annexes.

2 *Neumaier, K.:* „Essais d'interprétation de »Bedford« et de »Waterbury«. Rapport commun établi par les Centres de la Commission E de l'OEEPE ayant participé aux tests" – „The Interpretation Tests of »Bedford« and »Waterbury«. Common Report Established by all Participating Centres of Commission E of OEEPE" – „Essais de restitution »Bloc Suisse«. Rapport commun établi par les Centres de la Commission E de l'OEEPE ayant participé aux tests" – „Test »Schweizer Block«. Joint Report of all Centres of Commission E of OEEPE." – Frankfurt a. M. 1966, 60 pages with 44 annexes.

3 *Cunietti, M.:* „Emploi des blocs de bandes pour la cartographie à grande échelle – Résultats des recherches expérimentales organisées par la Commission B de l'O.E.E.P.E. au cours de la période 1959–1966" – „Use of Strips Connected to Blocks for Large Scale Mapping – Results of Experimental Research Organized by Commission B of the O.E.E.P.E. from 1959 through 1966." – Frankfurt a. M. 1968, 157 pages with 50 figures and 24 tables.

4 *Förstner, R.:* „Sur la précision de mesures photogrammétriques de coordonnées en terrain montagneux. Rapport sur les résultats de l'essai de Reichenbach de la Commission C de l'OEEPE" – „The Accuracy of Photogrammetric Co-ordinate Measurements in Mountainous Terrain. Report on the Results of the Reichenbach Test Commission C of the OEEPE." – Frankfurt a. M. 1968, Part I: 145 pages with 9 figures; Part II: 23 pages with 65 tables.

5 *Trombetti, C.:* „Les recherches expérimentales exécutées sur de longues bandes par la Commission A de l'OEEPE." – Frankfurt a. M. 1972, 41 pages with 1 figure, 2 tables, 96 annexes and 19 plates.

6 *Neumaier, K.:* „Essai d'interprétation. Rapports des Centres de la Commission E de l'OEEPE." – Frankfurt a. M. 1972, 38 pages with 12 tables and 5 annexes.

7 *Wiser, P.:* „Etude expérimentale de l'aérotiangulation semi-analytique. Rapport sur l'essai »Gramastetten«." – Frankfurt a. M. 1972, 36 pages with 6 figures and 8 tables.

8 „Proceedings of the OEEPE Symposium on Experimental Research on Accuracy of Aerial Triangulation (Results of Oberschwaben Tests)"
Ackermann, F.: „On Statistical Investigation into the Accuracy of Aerial Triangulation. The Test Project Oberschwaben" – „Recherches statistiques sur la précision de l'aérotriangulation. Le champ d'essai Oberschwaben" – *Belzner, H.:* „The Planning. Establishing and Flying of the Test Field Oberschwaben" – *Stark, E.:* Testblock Oberschwaben, Programme I. Results of Strip Adjustments" – *Ackermann, F.:* „Testblock Oberschwaben, Program I. Results of Block-Adjustment by Independent Models" – *Ebner, H.:* Comparison of Different Methods of Block Adjustment" – *Wiser, P.:* „Propositions pour le traitement des erreurs non-accidentelles" – *Camps, F.:* „Résultats obtenus dans le cadre du project Oberschwaben 2A" – *Cunietti, M.; Vanossi, A.:* „Etude statistique expérimentale des erreurs d'enchaînement des photogrammes" – *Kupfer, G.:* „Image Geometry as Obtained from Rheidt Test Area Photography" – *Förstner, R.:* „The Signal-Field of Baustetten. A Short Report" – *Visser, J.; Leberl, F.; Kure, J.:* „OEEPE Oberschwaben Reseau Investigations" – *Bauer, H.:* „Compensation of Systematic Errors by Analytical Block Adjustment with Common Image Deformation Parameters." – Frankfurt a. M. 1973, 350 pages with 119 figures, 68 tables and 1 annex.

9 *Beck, W.:* „The Production of Topographic Maps at 1 : 10,000 by Photogrammetric Methods. – With statistical evaluations, reproductions, style sheet and sample fragments by Landesvermessungsamt Baden-Württemberg Stuttgart." – Frankfurt a. M. 1976, 89 pages with 10 figures, 20 tables and 20 annexes.

10 „Résultats complémentaires de l'essai d'«Oberriet» of the Commission C de l'OEEPE – Further Results of the Photogrammetric Tests of «Oberriet» of the Commission C of the OEEPE"
Hárry, H.: „Mesure de points de terrain non signalisés dans le champ d'essai d'«Oberriet» – Measurements of Non-Signalized Points in the Test Field «Oberriet» (Abstract)" – *Stickler, A.; Waldhäusl, P.:* „Restitution graphique des points et des lignes non signalisés et leur comparaison avec des résultats de mesures sur le terrain dans le champ d'essai d'«Oberriet» – Graphical Plotting of Non-Signalized Points and Lines, and Comparison with Terrestrial Surveys in the Test Field «Oberriet»" – *Förstner, R.:* „Résultats complémentaires des transformations de coordonnées de l'essai d'«Oberriet» de la Commission C de l'OEEPE – Further Results from Co-ordinate Transformations of the Test «Oberriet» of Commission C of the OEEPE" – *Schürer, K.:* „Comparaison des distances d'«Oberriet» – Comparison of Distances of «Oberriet» (Abstract)." – Frankfurt a. M. 1975, 158 pages with 22 figures and 26 tables.

11 „25 années de l'OEEPE"
Verlaine, R.: „25 années d'activité de l'OEEPE" – „25 Years of OEEPE (Summary)" – *Baarda, W.:* „Mathematical Models." – Frankfurt a. M. 1979, 104 pages with 22 figures.

12 *Spiess, E.:* „Revision of 1 : 25,000 Topographic Maps by Photogrammetric Methods." – Frankfurt a. M. 1985, 228 pages with 102 figures and 30 tables.

13 *Timmerman, J.; Roos, P. A.; Schürer, K.; Förstner, R.:* On the Accuracy of Photogrammetric Measurements of Buildings – Report on the Results of the Test "Dordrecht", Carried out by Commission C of the OEEPE. – Frankfurt a. M. 1982, 144 pages with 14 figures and 36 tables.

14 *Thompson C. N.:* Test of Digitising Methods. – Frankfurt a. M. 1984, 120 pages with 38 figures and 18 tables.

15 *Jaakkola, M.; Brindöpke, W.; Kölbl, O.; Noukka, P.:* Optimal Emulsions for Large-Scale Mapping – Test of "Steinwedel" – Commission C of the OEEPE 1981–84. – Frankfurt a. M. 1985, 102 pages with 53 figures.

16 *Waldhäusl, P.:* Results of the Vienna Test of OEEPE Commission C. – *Kölbl, O.:* Photogrammetric Versus Terrestrial Town Survey. – Frankfurt a. M. 1986, 57 pages with 16 figures, 10 tables and 7 annexes.

17 *Commission E of the OEEPE:* Influences of Reproduction Techniques on the Identification of Topographic Details on Orthophotomaps. – Frankfurt a. M. 1986, 138 pages with 51 figures, 25 tables and 6 appendices.

18 *Förstner, W.:* Final Report on the Joint Test on Gross Error Detection of OEEPE and ISP WG III/1. – Frankfurt a. M. 1986, 97 pages with 27 tables and 20 figures.

19 *Dowman, I. J.; Ducher, G.:* Spacelab Metric Camera Experiment – Test of Image Accuracy. – Frankfurt a. M. 1987, 112 pages with 13 figures, 25 tables and 7 appendices.

20 *Eichhorn, G.:* Summary of Replies to Questionnaire on Land Information Systems – Commission V – Land Information Systems. – Frankfurt a. M. 1988, 129 pages with 49 tables and 1 annex.

21 *Kölbl, O.:* Proceedings of the Workshop on Cadastral Renovation – Ecole polytechnique fédérale, Lausanne, 9–11 September, 1987. – Frankfurt a. M. 1988, 337 pages with figures, tables and appendices.

22 *Rollin, J.; Dowman, I. J.:* Map Compilation and Revision in Developing Areas – Test of Large Format Camera Imagery. – Frankfurt a. M. 1988, 35 pages with 3 figures, 9 tables and 3 appendices.

23 *Drummond, J.* (ed.): Automatic Digitizing – A Report Submitted by a Working Group of Commission D (Photogrammetry and Cartography). – Frankfurt a. M. 1990, 224 pages with 85 figures, 6 tables and 6 appendices.

24 *Ahokas, E.; Jaakkola, J.; Sotkas, P.:* Interpretability of SPOT data for General Mapping. – Frankfurt a. M. 1990, 120 pages with 11 figures, 7 tables and 10 appendices.

25 *Ducher, G.:* Test on Orthophoto and Stereo-Orthophoto Accuracy. – Frankfurt a. M. 1991, 227 pages with 16 figures and 44 tables.

26 *Dowman, I. J.* (ed.): Test of Triangulation of SPOT Data – Frankfurt a. M. 1991, 206 pages with 67 figures, 52 tables and 3 appendices.

27 *Newby, P. R. T.; Thompson, C. N.* (ed.): Proceedings of the ISPRS and OEEPE Joint Workshop on Updating Digital Data by Photogrammetric Methods. – Frankfurt a. M. 1992, 278 pages with 79 figures, 10 tables and 2 appendices.

28 *Koen, L. A.; Kölbl, O.* (ed.): Proceedings of the OEEPE-Workshop on Data Quality in Land Information Systems, Apeldoorn, Netherlands, 4–6 September 1991. – Frankfurt a. M. 1992, 243 pages with 62 figures, 14 tables and 2 appendices.

29 *Burman, H.; Torlegård, K.:* Empirical Results of GPS – Supported Block Triangulation. – Frankfurt a. M. 1994, 86 pages with 5 figures, 3 tables and 8 appendices.

B. Special publications

– Special Publications O.E.E.P.E. – Number I

Solaini, L.; Trombetti, C.: Relation sur les travaux préliminaires de la Commission A (Triangulation aérienne aux petites et aux moyennes échelles) de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (O.E.E.P.E.). I^{ère} Partie: Programme et organisation du travail. – *Solaini, L.; Belfiore, P.:* Travaux préliminaires de la Commission B de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (O.E.E.P.E.) (Triangulations aériennes aux grandes échelles). – *Solaini, L.; Trombetti, C.; Belfiore, P.:* Rapport sur les travaux expérimentaux de triangulation aérienne exécutés par l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (Commission A et B). – *Lehmann, G.:* Compte rendu des travaux de la Commission C de l'O.E.E.P.E. effectués jusqu'à présent. – *Gotthardt, E.:* O.E.E.P.E. Commission C. Compte-rendu de la restitution à la Technischen Hochschule, Stuttgart, des vols d'essai du groupe I du terrain d'Oberriet. – *Brucklacher, W.:* Compte-rendu du centre «Zeiss-Aerotopograph» sur les restitutions pour la Commission C de l'O.E.E.P.E. (Restitution de la bande de vol, groupe I, vol. No. 5). – *Förstner, R.:* O.E.E.P.E. Commission C. Rapport sur la restitution effectuée dans l'Institut für Angewandte Geodäsie, Francfort sur le Main. Terrain d'essai d'Oberriet les vols No. 1 et 3 (groupe I). – I.T.C., Delft: Commission C, O.E.E.P.E. Déroulement chronologique des observations. – Photogrammetria XII (1955–1956) 3, Amsterdam 1956, pp. 79–199 with 12 figures and 11 tables.

– Publications spéciales de L'O.E.E.P.E. – Numéro II

Solaini, L.; Trombetti, C.: Relations sur les travaux préliminaires de la Commission A (Triangulation aérienne aux petites et aux moyennes échelles) de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (O.E.E.P.E.). 2^{e} partie. Prises de vues et points de contrôle. – *Gotthardt, E.:* Rapport sur les premiers résultats de l'essai d'«Oberriet» de la Commission C de l'O.E.E.P.E. – Photogrammetria XV (1958–1959) 3, Amsterdam 1959, pp. 77–148 with 15 figures and 12 tables.

– *Trombetti, C.:* Travaux de prises de vues et préparation sur le terrain effectuées dans le 1958 sur le nouveau polygone italien pour la Commission A de l'OEEPE. – Florence 1959, 16 pages with 109 tables.

– *Trombetti, C.; Fondelli, M.:* Aérotriangulation analogique solaire. – Firence 1961, 111 pages, with 14 figures and 43 tables.

– Publications spéciales de l'O.E.E.P.E. – Numéro III

Solaini, L.; Trombetti, C.: Rapport sur les résultats des travaux d'enchaînement et de compensation exécutés pour la Commission A de l'O.E.E.P.E. jusqu'au mois de Janvier 1960. Tome 1: Tableaux et texte. Tome 2: Atlas. – Photogrammetria XVII (1960–1961) 4, Amsterdam 1961, pp. 119–326 with 69 figures and 18 tables.

– „OEEPE – Sonderveröffentlichung Nr. 1"

Gigas, E.: „Beitrag zur Geschichte der Europäischen Organisation für photogrammetrische experimentelle Untersuchungen" – *N. N.:* „Vereinbarung über die Gründung einer Europäischen Organisation für photogrammetrische experimentelle Untersuchungen" – „Zusatzprotokoll" – *Gigas, E.:* „Der Sechserausschuß" – *Brucklacher, W.:* „Kurzbericht über die Arbeiten in der Kommission A der OEEPE" – *Cunietti, M.:* „Kurzbericht des Präsidenten der Kommission B über die gegenwärtigen Versuche und Untersuchungen" – *Förstner, R.:* „Kurzbericht über die Arbeiten in der Kommission B der OEEPE" – „Kurzbericht über die Arbeiten in der Kommission C der OEEPE" – *Belzner, H.:* „Kurzbericht über die Arbeiten in der Kommission E der OEEPE" – *Schwidefsky, K.:* „Kurzbericht über die Arbeiten in der Kommission F der OEEPE" – *Meier, H.-K.:* „Kurzbericht über die Tätigkeit der Untergruppe „Numerische Verfahren" in der Kommission F der OEEPE" – *Belzner, H.:* „Versuchsfelder für internationale Versuchs- und Forschungsarbeiten." – Nachr. Kt.- u. Vermesss.-wes., R. V, Nr. 2, Frankfurt a. M. 1962, 41 pages with 3 tables and 7 annexes.

– *Rinner, K.:* Analytisch-photogrammetrische Triangulation eines Teststreifens der OEEPE. – Österr. Z. Vermess.-wes., OEEPE-Sonderveröff. Nr. 1, Wien 1992, 31 pages.

– *Neumaier, K.; Kasper, H.:* Untersuchungen zur Aerotriangulation von Überweitwinkelaufnahmen. – Österr. Z. Vermess.-wes., OEEPE-Sonderveröff. Nr. 2, Wien 1965, 4 pages with 4 annexes.

– „OEEPE – Sonderveröffentlichung Nr. 2"

Gotthardt, E.: „Erfahrungen mit analytischer Einpassung von Bildstreifen." – Nachr. Kt.- u. Vermess.-wes., R. V, Nr. 12, Frankfurt a. M. 1965, 14 pages with 2 figures and 7 tables.

– „OEEPE – Sonderveröffentlichung Nr. 3"

Neumaier, K.: „Versuch »Bedford« und »Waterbury«. Gemeinsamer Bericht aller Zentren der Kommission E der OEEPE" – „Versuch »Schweizer Block«. Gemeinsamer Bericht aller Zentren der Kommission E der OEEPE." – Nachr. Kt.- u. Vermess.-wes., R.V, Nr. 13, Frankfurt a. M. 1966, 30 pages with 44 annexes.

– *Stickler, A.; Waldhäusl, P.:* Interpretation der vorläufigen Ergebnisse der Versuche der Kommission C der OEEPE aus der Sicht des Zentrums Wien. – Österr. Z. Vermess.-wes., OEEPE-Sonderveröff. (Publ. Spéc.) Nr. 3, Wien 1967, 4 pages with 2 figures and 9 tables.

– „OEEPE – Sonderveröffentlichung Nr. 4"

Schürer, K.: „Die Höhenmeßgenauigkeit einfacher photogrammetrischer Kartiergeräte. Bemerkungen zum Versuch »Schweizer Block« der Kommission E der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M.,1968, 25 pages with 7 figures and 3 tables.

- „OEEPE – Sonderveröffentlichung Nr. 5"

  Förstner, R.: „Über die Genauigkeit der photogrammetrischen Koordinatenmessung in bergigem Gelände. Bericht über die Ergebnisse des Versuchs Reichenbach der Kommission C der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1969, Part I: 74 pages with 9 figures; Part II: 65 tables.

- „OEEPE – Sonderveröffentlichung Nr. 6"

  Knorr, H.: „Die Europäische Organisation für experimentelle photogrammetrische Untersuchungen – OEEPE – in den Jahren 1962 bis 1970." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1971, 44 pages with 1 figure and 3 tables.

- „OEEPE – Sonderveröffentlichung Nr. D–7"

  Förstner, R.: „Das Versuchsfeld Reichenbach der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1972, 191 pages with 49 figures and 38 tables.

- „OEEPE – Sonderveröffentlichung Nr. D–8"

  Neumaier, K.: „Interpretationsversuch. Berichte der Zentren der Kommission E der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1972, 33 pages with 12 tables and 5 annexes.

- „OEEPE – Sonderveröffentlichung Nr. D–9"

  Beck, W.: „Herstellung topographischer Karten 1 : 10 000 auf photogrammetrischem Weg. Mit statistischen Auswertungen, Reproduktionen, Musterblatt und Kartenmustern des Landesvermessungsamts Baden-Württemberg, Stuttgart." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1976, 65 pages with 10 figures, 20 tables and 20 annexes.

- „OEEPE – Sonderveröffentlichung Nr. D–10"

  Weitere Ergebnisse des Meßversuchs „Oberriet" der Kommission C der OEEPE. *Hârry, H.:* „Messungen an nicht signalisierten Geländepunkten im Versuchsfeld «Oberriet»" – *Stickler, A.; Waldhäusl, P.:* „Graphische Auswertung nicht signalisierter Punkte und Linien und deren Vergleich mit Feldmessungsergebnissen im Versuchsfeld «Oberriet»" – *Förstner, R.:* „Weitere Ergebnisse aus Koordinatentransformationen des Versuchs «Oberriet» der Kommission C der OEEPE" – *Schürer, K.:* „Streckenvergleich «Oberriet»." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1975, 116 pages with 22 figures and 26 tables.

- „OEEPE – Sonderveröffentlichung Nr. D–11"

  Schulz, B.-S.: „Vorschlag einer Methode zur analytischen Behandlung von Reseauaufnahmen." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1976, 34 pages with 16 tables.

- „OEEPE – Sonderveröffentlichung Nr. D–12"

  Verlaine, R.: „25 Jahre OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1980, 53 pages.

- „OEEPE – Sonderveröffentlichung Nr. D–13"

  Haug, G.: „Bestimmung und Korrektur systematischer Bild- und Modelldeformationen in der Aerotriangulation am Beispiel des Testfeldes „Oberschwaben." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1980, 136 pages with 25 figures and 51 tables.

- „OEEPE – Sonderveröffentlichung Nr. D–14"

  Spiess, E.: „Fortführung der Topographischen Karte 1 : 25 000 mittels Photogrammetrie" (not published, see English version in OEEPE official publication No. 12)

- „OEEPE – Sonderveröffentlichung Nr. D–15"

  Timmerman, J.; Roos, P. A.; Schürer, K.; Förstner, R.: „Über die Genauigkeit der photogrammetrischen Gebäudevermessung. Bericht über die Ergebnisse des Versuchs Dordrecht der Kommission C der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1983, 131 pages with 14 figures and 36 tables.

- „OEEPE – Sonderveröffentlichung Nr. D–16"

  Kommission E der OEEPE: „Einflüsse der Reproduktionstechnik auf die Erkennbarkeit von Details in Orthophotokarten." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1986, 130 pages with 51 figures, 25 tables and 6 annexes.

- „OEEPE – Sonderveröffentlichung Nr. D–17"

  Schürer, K.: „Über die Genauigkeit der Koordinaten signalisierter Punkte bei großen Bildmaßstäben. Ergebnisse des Versuchs „Wien" der Kommission C der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1987, 84 pages with 3 figures, 10 tables and 42 annexes.

C. Congress reports and publications in scientific reviews

- *Stickler, A.:* Interpretation of the Results of the O.E.E.P.E. Commission C. – Photogrammetria XVI (1959–1960) 1, pp. 8–12, 3 figures, 1 annexe (en langue allemande: pp. 12–16).
- *Solaini, L.; Trombetti, C.:* Results of Bridging and Adjustment Works of the Commission A of the O.E.E.P.E. from 1956 to 1959. – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 340–345, 2 tables.
- *N. N.:* Report on the Work Carried out by Commission B of the O.E.E.P.E. During the Period of September 1956–August 1960. – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 346–351, 2 tables.
- *Förstner, R.:* Bericht über die Tätigkeit und Ergebnisse der Kommission C der O.E.E.P.E. (1956–1960). – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 352–357, 1 table.
- *Bachmann, W. K.:* Essais sur la précision de la mesure des parallaxes verticales dans les appareils de restitution du I^er ordre. – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 358–360).
- *Wiser, P.:* Sur la reproductibilité des erreurs du cheminement aérien. – Bull. Soc. Belge Photogramm., No. 60, Juin 1960, pp. 3–11, 2 figures, 2 tables.
- *Cunietti, M.:* L'erreur de mesure des parallaxes transversales dans les appareils de restitution. – Bull. Trimestr. Soc. Belge Photogramm., No. 66, Décembre 1961, pp. 3–50, 12 figures, 22 tables.

- „OEEPE – Arbeitsberichte 1960/64 der Kommissionen A, B, C, E, F"

  Trombetti, C.: „Activité de la Commission A de l'OEEPE de 1960 à 1964" – *Cunietti, M.:* „Activité de la Commission B de l'OEEPE pendant la période septembre 1960–janvier 1964" – *Förstner, R.:* „Rapport sur les travaux et les résultats de la Commission C de l'OEEPE (1960–1964)" – *Neumaier, K.:* „Rapport de la Commission E pour Lisbonne" – *Weele, A. J. van der:* „Report of Commission F." – Nachr. Kt.- u. Vermess.-wes., R. V. Nr. 11, Frankfurt a. M. 1964, 50 pages with 7 tables and 9 annexes.
- *Cunietti, M.; Inghilleri, G.; Puliti, M.; Togliatti, G.:* Participation aux recherches sur les blocs de bandes pour la cartographie à grande échelle organisées par la Commission B de l'OEEPE. Milano, Centre CASF du Politecnico. – Boll. Geod. e Sc. affini (XXVI) 1, Firenze 1967, 104 pages.
- *Gotthardt E.:* Die Tätigkeit der Kommission B der OEEPE. – Bildmess. u. Luftbildwes. 36 (1968) 1, pp. 35–37.
- *Cunietti, M.:* Résultats des recherches expérimentales organisées par la Commission B de l'OEEPE au cours de la période 1959–1966. Résumé du Rapport final. – Présenté à l'XI^e Congrès International de Photogrammétrie, Lausanne 1968, Comm. III (en langues française et anglaise), 9 pages.

- *Förstner, R.:* Résumé du Rapport sur les résultats de l'essai de Reichenbach de la Commission C de l'OEEPE. – Présenté à l'XI^e Congrès International de Photogrammétrie, Lausanne 1968, Comm. IV (en langues française, anglaise et allemande), 28 pages, 2 figures, 2 tables.
- *Timmerman, J.:* Proef „OEEPE-Dordrecht". – ngt 74, 4. Jg., Nr. 6, Juni 1974, S. 143–154 (Kurzfassung: Versuch „OEEPE-Dordrecht". Genauigkeit photogrammetrischer Gebäudevermessung. Vorgelegt auf dem Symposium der Kommission IV der I.G.P., Paris, 24.–26. September 1974).
- *Timmerman, J.:* Report on the Commission C. "OEEPE-Dordrecht" Experiment. – Presented Paper for Comm. IV, XIII^th Congress of ISP, Helsinki 1976.
- *Beck, W.:* Rapport de la Commission D de l'OEEPE sur l'établissement de cartes topographiques au 1/10 000 selon le procédé photogrammétrique. – Presented Paper for Comm. IV, XIII^th Congress of ISP, Helsinki 1976.
- *Verlaine, R.:* La naissance et le développement de l'OEEPE – Festschrift – Dr. h. c. *Hans Härry*, 80 Jahre – Schweizerische Gesellschaft für Photogrammetrie und Wild Heerbrugg AG, Bern 1976.
- *Förstner, R.:* Internationale Versuche (Essais contrôlés) – Festschrift – Dr. h. c. *Hans Härry*, 80 Jahre – Schweizerische Gesellschaft für Photogrammetrie und Wild Heerbrugg AG, Bern 1976.
- *Bay, E.; Cunietti, M.; Vanossi, A.:* Détermination Expérimentale des Erreurs Systématiques des Faisceaux Perspectifs. – Société Belge de Photogrammétrie, Bulletin trimestriel, Brüssel 1977, pp 21–49.
- *Timmerman, J.:* Fotogrammetrische stadskaartering de OEEPE-proef Dordrecht. – Geodesia 19, Amsterdam 1977, pp. 291–298.
- *Waldhäusl, P.:* The Vienna Experiment of the OEEPE/C. Proceedings – Standards and Specifications for Integrated Surveying and Mapping Systems. – Schriftenreihe HSBw, Heft 2, München 1978.
- *Bachmann, W. K.:* Recherches sur la stabilité des appareils de restitution photogrammétriques analogiques. – Vermessung, Photogrammetrie, Kulturtechnik, Zürich 1978, pp. 265–268.
- *Parsic, Z.:* Untersuchungen über den Einfluß signalisierter und künstlicher Verknüpfungspunkte auf die Genauigkeit der Blocktriangulation. – Vermessung, Photogrammetrie, Kulturtechnik, Zürich 1978, pp. 269–278.
- *Waldhäusl, P.:* Der Versuch Wien der OEEPE/C. – Geowissenschaftliche Mitteilungen der Studienrichtung Vermessungswesen der TU Wien; Heft 13, Wien 1978, pp. 101–124.
- *Waldhäusl, P.:* Ergebnisse des Versuches Wien der OEEPE/C. – Presented Paper for Comm. IV, XIV^th Congress of ISP, Hamburg 1980.
- *Timmerman, J.; Förstner, R.:* Kurzbericht über die Ergebnisse des Versuchs Dordrecht der Kommission C der OEEPE. – Presented Paper for Comm. IV, XIV^th Congress of ISP, Hamburg 1980.

- *Bernhard, J.; Schmidt-Falkenberg, H.:* OEEPE – Die Arbeiten der Kommission E „Interpretation". – Presented Paper for Comm. IV, XIVth Congress of ISP, Hamburg 1980.
- *Bachmann, W. K.:* Elimination des valeurs erronées dans un ensemble de mesures controlées. – Papers written in honor of the 70th birthday of Professor *Luigi Solaini* – Richerche di Geodesia Topografia e Fotogrammetria, Milano 1979, pp. 27–39.
- *Visser, J.:* The European Organisation for Experimental Photogrammetric Research (OEEPE) – The Thompson Symposium 1982. – The Photogrammetric Record, London 1982, pp. 654–668.
- *Spiess, E.:* Revision of Topographic Maps: Photogrammetric and Cartographic Methods of the Fribourg Test. – The Photogrammetric Record, London 1983, pp. 29–42.
- *Jerie, H. G. and Holland, E. W.:* Cost model project for photogrammetric processes: a progress report. – ITC Journal, Enschede 1983, pp. 154–159.
- *Ackermann, F. E.* (Editor): Seminar – Mathematical Models of Geodetic/Photogrammetric Point Determination with Regard to Outliers and Systematic Errors – Working Group III/1 of ISP – Commission A of OEEPE. – Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften, Reihe A, Heft Nr. 98, München 1983.
- *Brindöpke, W., Jaakkola, M., Noukka, P., Kölbl, O.:* Optimal Emulsions for Large Scale Mapping – OEEPE–Commission C. – Presented Paper for Comm. I, XVth Congress of ISPRS, Rio de Janeiro 1984.
- *Ackermann, F.:* Report on the Activities of Working Group III/1 During 1980–84. – Comm. III, XVth Congress of ISPRS, Rio de Janeiro 1984.
- *Förstner, W.:* Results of Test 2 on Gross Error Detection of ISP WG III/1 and OEEPE. – Comm. III, XVth Congress of ISPRS, Rio de Janeiro 1984.
- *Gros, G.:* Modèles Numériques Altimétriques – Lignes Characteristiques – OEEPE Commission B. – Comm. III, XVth Congress of ISPRS, Rio de Janeiro 1984.
- *Ducher, G.:* Preparation d'un Essai sur les Ortho- et Stereo-Orthophotos. – Comm. IV, XVth Congress of ISPRS, Rio de Janeiro 1984.
- *van Zuylen, L.:* The influence of reproduction methods on the identification of details in orthophoto maps. – ITC Journal, Enschede 1984, pp. 219–226.
- *Visser, J.:* OEEPE- News – The European Organization for Experimental Photogrammetric Research. – ITC Journal, Enschede 1984, pp. 330–332.
- *Brindöpke, W., Jaakkola, M., Noukka, P., Kölbl, O.:* Optimale Emulsionen für großmaßstäbige Auswertungen. – Bildmess. u. Luftbildw. 53 (1985) 1, pp. 23–35.
- *Thompson, C. N.:* Some New Horizons for OEEPE. Presented Paper to the Symposium of Commission IV, ISPRS in Edinburgh, 8.–12. September 1986, pp. 299–306.

- *Dowman, I.:* The Restitution of Metric Photography Taken From Space – Comm. II, XVIth Congress of ISPRS, Kyoto 1988.
- *Kilpelä, E.:* Statistical Data on Aerial Triangulation – Comm. III, XVIth Congress of ISPRS, Kyoto 1988.
- *de Haan, A.:* An Analysis of the Precision of a DEM Obtained from SPOT Data – Comm. IV, XVIIth Congress of ISPRS, Washington 1992.
- *Ducher, G.:* OEEPE Test on Orthophoto and Stereo-Orthophoto Accuracy – Comm. IV, XVIIth Congress of ISPRS, Washington 1992.
- *Veillet, I.:* Accuracy of SPOT Triangulation With Very Few or no Ground Control Point – Comm. IV., XVIIth Congress of ISPRS, Washington 1992.

The official publications and the special publications issued in Frankfurt am Main are for sale at the

Institut für Angewandte Geodäsie
– Außenstelle Berlin –

Stauffenbergstraße 13, D-10785 Berlin

Organisation Européenne d'Etudes Photogrammétriques Expérimentales
Publications officielles

Content