Geol. Jb.	A 122	91—100	4 fig.	Hannover 1992

An Experimental Geoscientific Database System

KARL NEUMANN & FRIEDRICH LOHMANN & HANS-DIETER EHRICH

Project, geo-information system, database language, data handling, computerized cartography, agricultural use, soil texture, soil assessment

Abstract: This paper presents a survey of the activities carried out within our research project on the design of a geoscientific database management system and outlines the results. The project has explored new ways of providing appropriate database support for geoscientific applications. We have proposed and partially implemented a novel, high-level database user interface which permits storing, managing, and retrieving geoscientific data in a convenient way and which also supports computerized cartography. A broad variety of geoscientific applications can be supported, as the user is enabled to define his own geo-object classes and even add new geometric data types. A sample application serves as an illustration of our approach to modelling and manipulating geoscientific data.

[Ein experimentelles geowissenschaftliches Datenbanksystem]

Kurzfassung: Diese Arbeit gibt einen Überblick über Verlauf und Ergebnisse des Forschungsprojektes «Geo-Datenbank-Entwurf». Inhalt des Projektes war die Erforschung neuer Möglichkeiten, geowissenschaftlichen Anwendungen geeignete Datenbankunterstützung zu bieten. Es wurde eine neuartige Benutzerschnittstelle für ein Datenbanksystem vorgeschlagen und teilweise implementiert, die es erlaubt, geowissenschaftliche Daten bequem zu speichern, zu manipulieren und wiederzufinden, und die auch automatische Kartografie unterstützt. Eine Vielzahl geowissenschaftlicher Anwendungen kann unterstützt werden, da der Benutzer die Möglichkeit hat, seine eigenen, anwendungsspezifischen Objektklassen zu definieren und sogar neue geometrische Datentypen in das System einzuführen. Eine Beispielanwendung soll den hier verfolgten Ansatz zur Modellierung und Handhabung geowissenschaftlicher Daten veranschaulichen.

Contents

Authors' address: Dr. K. NEUMANN, Dr. F. LOHMANN, Prof. Dr. H.-D. EHRICH, Informatik/Datenbanken, Technische Universität Braunschweig, Postfach 3329, D-3300 Braunschweig.

1 Introduction

In recent years, it has been recognized that traditional, commercially available database management systems such as IMS, SQL/DS, or ORACLE are not well suited for so-called non-standard applications (DITTRICH & DAYAL 1986; BANCILHON 1988; DITTRICH 1988). Therefore, many research groups have started to develop new data models (cf. survey in PECKHAM & MARYANSKI 1988), database languages (ANDERSEN et al. 1988; HOHENSTEIN & GOGOLLA 1988; GÜTING 1989), and storage structures (SCHEK & WATERFELD 1986; SAMET et al. 1988; HENRICH et al. 1989). These new concepts are intended to be appropriate for areas like business, manufacturing, engineering design, and, of course, geoscientific applications.

Within the «Digital Geoscientific Maps» priority project (VINKEN 1988a), it was our task to develop a new database management language that permits storing, managing, and retrieving geoscientific data, as well as support computerized cartography. Hence, this subproject, which was supported by the German Research Foundation (DFG), can be seen as a small part of the nonstandard database research activities sketched above.

We started our project in 1984/1985 by first designing a special nonstandard database language. We then implemented a partial prototype in order to be able to demonstrate the features of the language for realistic geoscientific applications. With these experiences in mind, we carefully designed a second, improved prototype. The following chapter gives a survey of the extended database language, while chapter 3 outlines the architecture of the first and second prototypes. Finally, in chapter 4, we give some conclusions concerning the whole project.

2 Database Language

In this chapter we present a brief survey of the geoscientific database language and the underlying object model. As the database language and the underlying concepts have been presented elsewhere (LIPECK & NEUMANN 1987, EHRICH et al. 1988, and NEUMANN 1988), we confine ourselves to summarizing the characteristic features and introducing a few typical language constructs in a sample application taken from the soil sciences.

The geoscientific database language is based on a specific *geo-object model* which is an extension of the well-known Entity Relationship model (ER model, CHEN 1976). Hence, geoscientific worlds are modelled in terms of object classes and relationship classes. While objects have to be atomic in the classical ER model, the geo-object model also allows *complex objects* composed of a hierarchy of subobjects (or sets or lists of subobjects). Within the database, all objects are represented by their attribute values.

The geo-object model supports a conceptual distinction between information about geoscientific objects (*geo-objects*) and their graphic representation on maps (*map objects*). While the user may define arbitrary geo-object classes according to the requirements of his specific application, the attribute scheme of maps and map objects is predefined and fixed. The database language offers special constructs for transforming geo-objects into corresponding map objects.

A significant feature of geoscientific objects is that they usually have a spatial extent. In order to take this important characteristic into consideration, the geo-object model offers special *geometric data types* in addition to the conventional alphanumeric data types. Thus, geometric data can be manipulated just as conveniently as standard data. The following data types are defined in the database system: *points* (sets of points in a plane), *lines* (sets of lines in a plane), and *polygons* (sets of polygonal areas in a plane). A variety of operations, e.g. for computing intersections or for computing the size of a line or an area, are also offered by this database system. Moreover, the user can define additional geometric data types and introduce them into the database system, e.g. spline curves, rasters, or points in space. These new types can then be used in database statements in the same way as the predefined data types.

In its data definition part, the database language enables the user to model his view of the geoscientific world by defining (and deleting) atomic or complex geo-object classes and relationships; in its data manipulation part, the language includes statements for creating, retrieving, updating, and deleting geo-objects, relationships, maps and map objects, as well as statements for alphanumeric and graphic output. The database language shows some resemblance to the relational database language QUEL (STONEBRAKER et al. 1976). Like QUEL, our geodatabase language uses range variables for objects stored in the database. The semantics of the database statements can be precisely formulated by the underlying object calculus (NEUMANN 1988).

To illustrate the use of the database language, we now present a few typical statements taken from a sample application. We consider a database in which base data for computing soil erosion in a certain test area are to be stored (cf. BORK & HENSEL 1988; HEITLAND 1988). The geo-object scheme for this database may be defined as follows:

```
DEFINE GEOOBJ Field
    (Name               String,
    Geometry            Polygons,
    Crops per Year      LIST OF GEOOBJ Crop per Year
        (Year               Integer,
        Crop                String));

DEFINE GEOOBJ Drill Hole
    (Geometry           Points,
    Soil Texture        String,
    Layers              SET OF GEOOBJ Layers
        (Depth              Real,
        Organic Matter      Integer));

DEFINE GEOOBJ Soil Area
    (Name               String,
    Geometry            Polygons);

DEFINE REL Point in Area (Drill Hole, Soil Area);
```

Information about the fields in the test area and their agricultural use is assigned to the first object class (Fig. 1). Each field has a unique name and geometric shape (a polygon). A list of the crops grown on the field over the course of several years (e.g. grain, sugar beets, or rapeseed) is stored for each field.

The second object class contains information about drill holes. The geometric location of the drill hole and the soil texture found there (e.g. loamy sand, loam, clayey loam, or clay) is stored for each drill hole. Additionally, we store the value of the mass fraction of organic matter found at this point in the different layers. Each drill hole and its respective soil texture is considered to be representative for exactly one area (according to the Reichsbodenschätzung (soil assessment)), called «Soil Area» (Fig. 2). The association between drill holes and soil areas is given by the relationship «Point in Area».

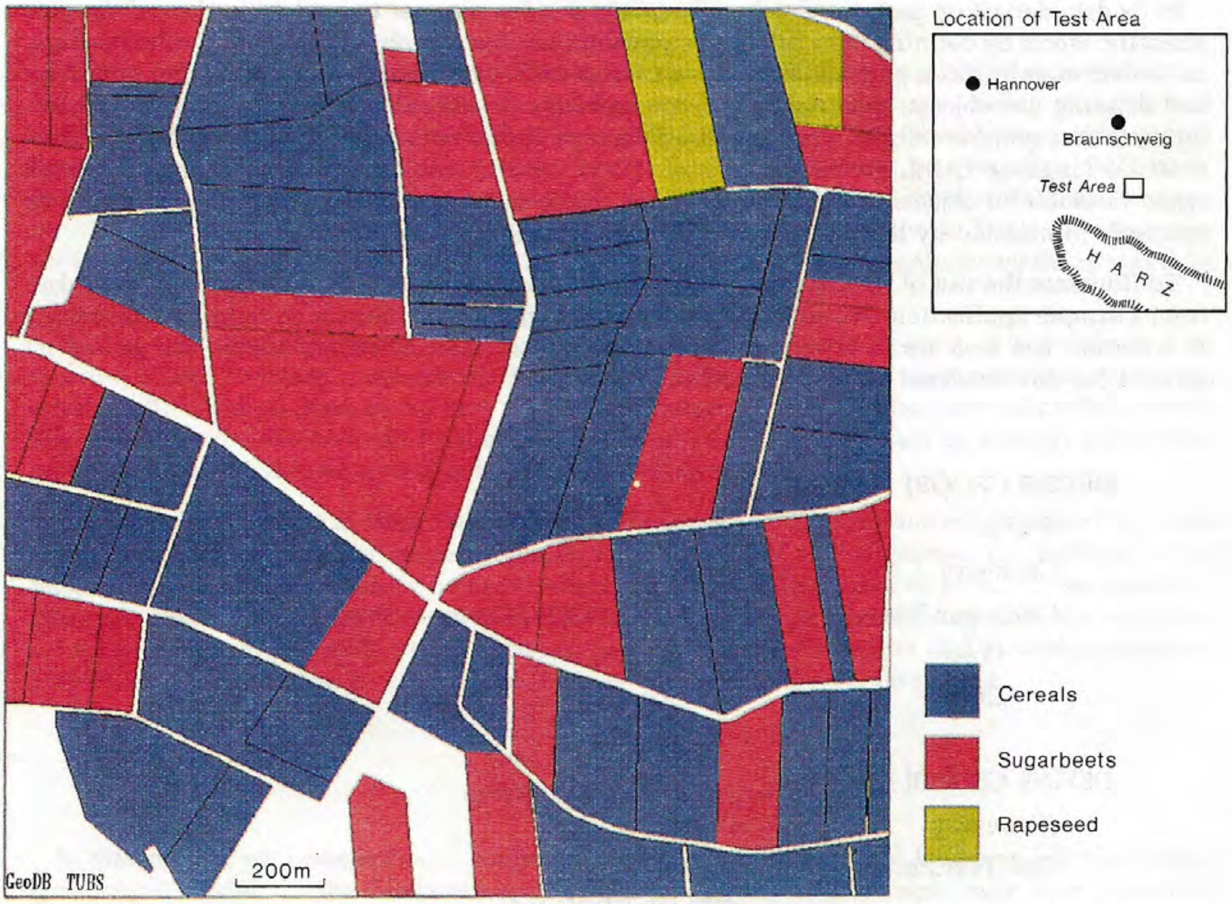

Fig. 1: Map of agricultural use in 1982.

When the geo-object scheme has been defined, data can be inserted into the database and retrieved for evaluation. The following query results in a temporary geo-object class named «Intersection Area»:

```
RANGE OF f IS Field;
RANGE OF c IS f.Crops per Year;
RANGE OF d IS Drill Hole;
RANGE OF s IS Soil Area;
RETRIEVE GEOOBJ INTO Intersection Area
    (Name           = f.Name CAT s.Name,
    Geometry        = Intersection (f.Geometry, s.Geometry),
    Soil Texture    = d.Soil Texture,
    Crop            = c.Crop)

WHERE Cut (f.Geometry, s.Geometry)
    AND Point in Area (d, s)
    AND c.Year = 1982;
```

The RETRIEVE statement is preceded by a declaration of range variables needed to refer to the objects stored in the respective geo-object classes. Each object of the temporary class resulting from the query represents an area with uniform soil texture and uniform agricultural use (Fig. 3). The geometries of these areas are computed by intersecting the fields «f» and soil areas «s» using the

Fig. 2: Map of soil texture according to the soil assessment (Reichsbodenschätzung).

geometric operator «Intersection»; the names are formed by concatenating the names of the respective field and soil area. The «cut» operation in the qualification part of the query specifies that only non-void intersections are to be stored. The soil texture of an area is given by the soil texture of drill hole «d» within soil area «s». The crop attribute is given by the crop grown in field «f» during 1982.

After computation of the class «Intersection Area», the results can be printed in alphanumeric form or they can be transformed into map objects for graphic display. Map construction is supported by the database language; the following CREATE MAPOBJ statement gives a idea of how it works. This statement creates map objects of all the areas where sugar beets are grown on loam and inserts them onto a map named «Soil Crops 82»:

```
RANGE OF i IS Intersection Area;
RANGE OF m IS MAP;
CREATE MAPOBJ FROM i
    INSERT INTO m.FACE
    (POLYGONS: Type        = SOLID,
               Colour      = 7 /* red*/)
WHERE m.Name = «Soil Crops 82»
    AND i.Soil Texture     = «loam»
    AND i.Crop             = «sugar beets»;
```

Note that the geometries of the map objects are computed implicitly by conversion of the coordinates of the underlying geo-objects from world coordinates into those of the map sheet.

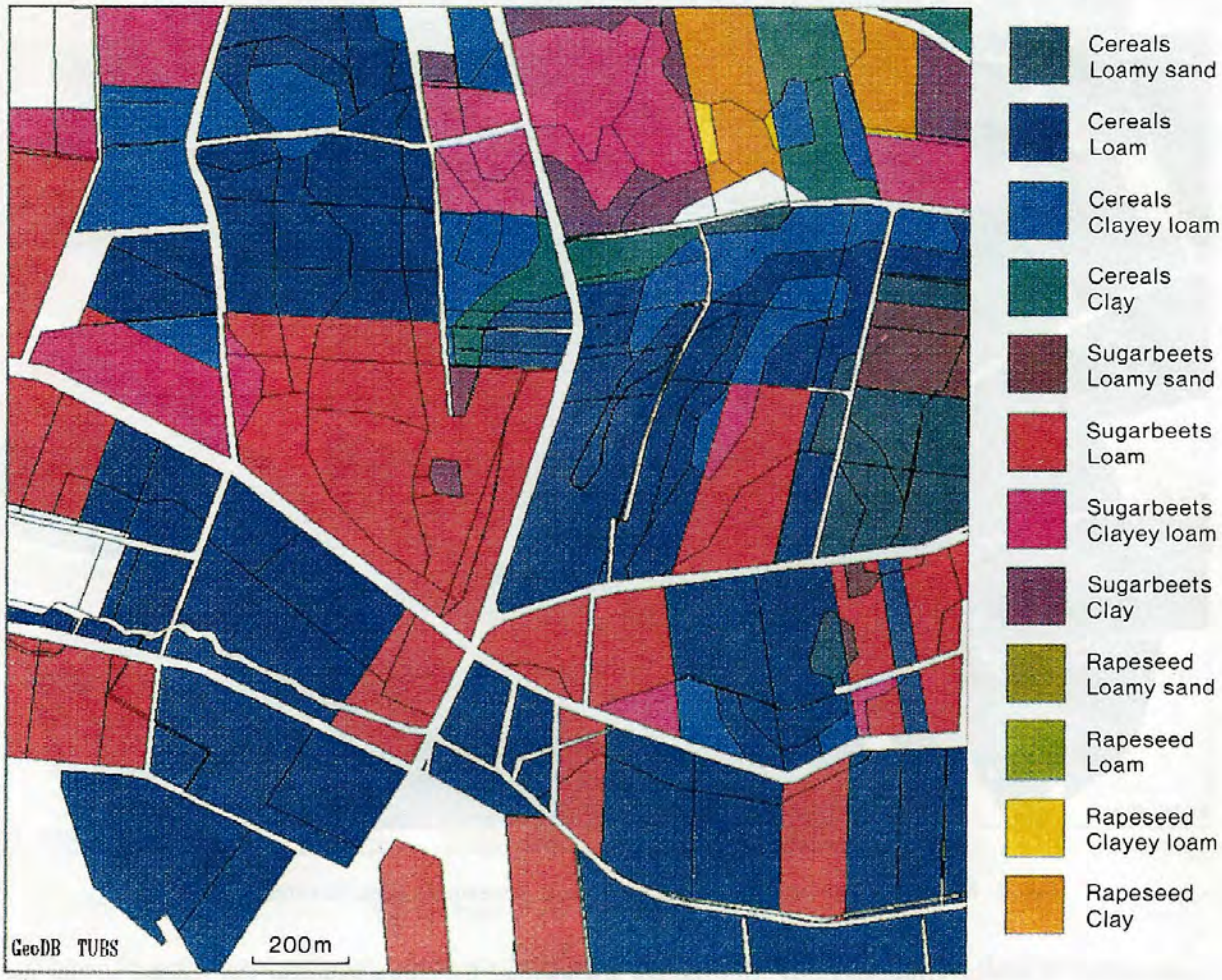

Fig. 3: Map of soil texture and agricultural use in 1982.

3 First and Second Prototypes

A subset of the database language presented in the previous chapter was implemented in a provisional prototype (JUNGCLAUS & NEUMANN 1988). It offers atomic geo-object classes and the most important two-dimensional geometric data types (points, lines, and polygons) with a variety of operations. It also supports the construction of maps.

This first prototype primarily offers a dialogue interface, which was later supplemented by a (still rather primitive) application programming interface (TEGETHOFF 1988). Programming was done in C (KERNIGHAN & RITCHIE 1977) on a VAX750 using the operating system ULTRIX. The choice of the IDM500 relational database machine (BRITTON LEE Inc. 1984) made it possible to implement the system in a relatively short time. On the other hand, it brought about rather unsatisfactory response times, as semantically related data are spread over several relations.

Nevertheless, the first prototype enabled us to try out and review our concepts, and it has thus fully served its intended purpose. Several geoscientific applications were processed successfully

(OSTERHOLD 1987; HEITLAND 1988; TIETJEN 1988) and proved our concepts to be appropriate. The maps shown in Figures 1, 2 and 3 were constructed solely by using this first prototype.

Our experience with the first prototype, as well as discussions with geoscientists, influenced the design of a second prototype (LOHMANN 1987; LOHMANN et al. 1989), which is currently being implemented. It offers substantial improvements and extensions to the first prototype:

— The second prototype supports the definition and manipulation of arbitrarily complex geo-object classes.
— The user can define his own geometric data types and introduce them into the database system.
— A convenient, high-level application programming interface is offered.
— A special database kernel is used as the basis of implementation.

The overall architecture of the second prototype is sketched in Figure 4. To support programming of database applications, the geodatabase language has been embedded in the Modula 2 programming language (WIRTH 1985). Hence, application programs consist of a mixture of Modula statements and database statements. Database statements may be paramaterized by program variables to convey data from the program to the database.

Special tools, an object-type generator and an object-type compiler, are provided to generate abstract-data-type modules (ADT modules) for all geo-object classes the user wishes to process in an application program. The types and operations offered by the ADT modules enable convenient manipulation of complex geo-objects within application programs, while the programmer does not have to concern himself with their implementation. Similarly, a data-type compiler supports the introduction of new geometric data types.

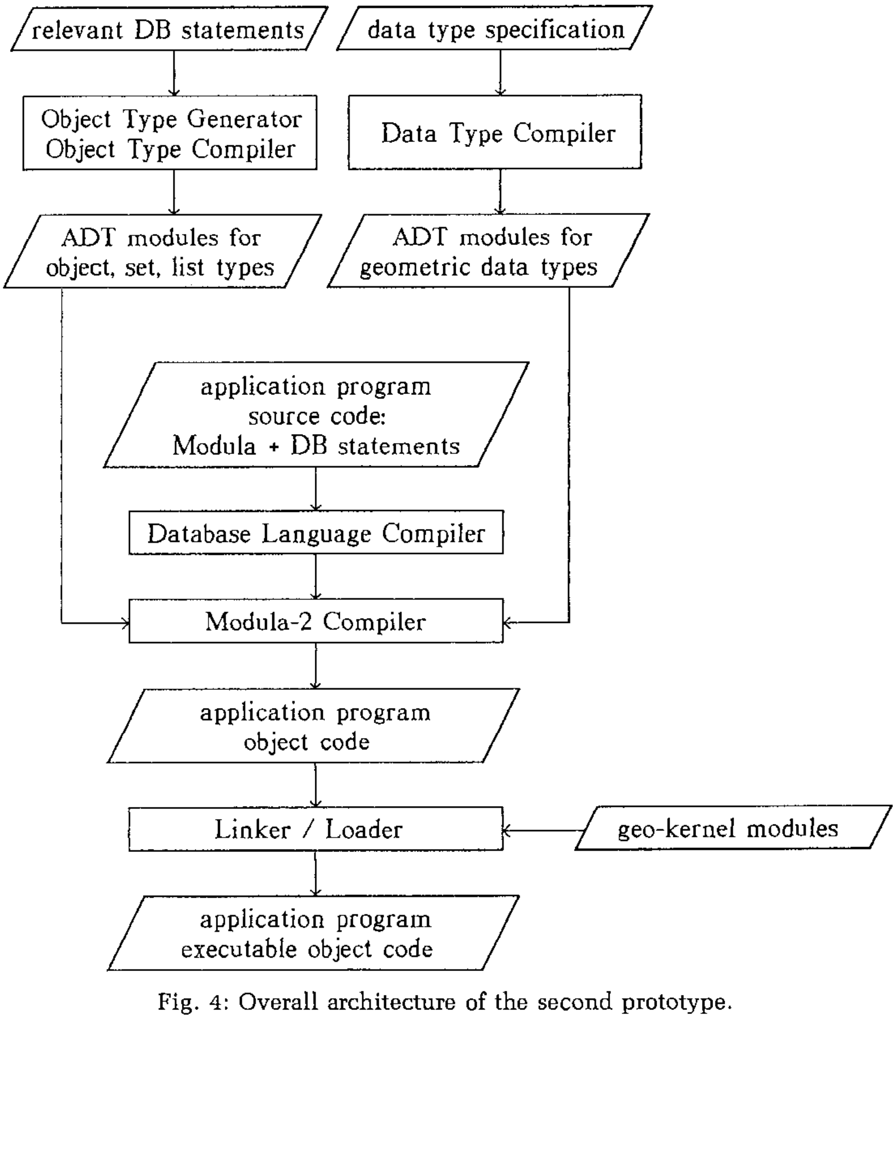

Fig. 4: Overall architecture of the second prototype.

Application programs with embedded database statements are preprocessed by a data-base language compiler, which searches the program for database statements, checks their syntax and semantics and also performs interlanguage type checking. For each database statement, the database language compiler generates a module with the «implementation» of the statement. Within the application program, the database language compiler replaces every database statement with a call to the respective module, in a way that the resulting program can be subsequently compiled by a standard Modula 2 compiler. More details about the application programming interface can be found in LOHMANN (1988a—c).

The second prototype is being implemented on a SUN3 workstation with UNIX; programming is done in Modula 2. The implementation is based on a special geodatabase kernel (SCHEK & WATERFELD 1986; HORN et al. 1988; WATERFELD et al. 1988). The database language compiler transforms, in several steps, the descriptive statements of the geodatabase language to the level of the call interface offered by the geokernel (NEUMANN 1988). As the geokernel supports nested relations with geometric attributes, we expect a considerable improvement of response times over those of the first prototype.

4 Summary

In this paper, we have outlined a geoscientific database language and two prototypes which make the language operational. While the first prototype was finished two years ago, the second one is currently being implemented. Most of its components will be completed in the near future.

The cooperation with the geoscientific research groups brought some interesting challenges for us. Thus, for instance, the realization of geometric data types point, line, and polygon and the associated operations, such as the intersection of polygons, seemed to be one of our hardest tasks in the beginning. Then the users asked for arbitrary geometric data types, including graphic representations of three-dimensional surfaces. On the other hand, the geoscientists also had to learn some «difficult» things, like modelling their worlds in terms of our database definition language and understanding the mechanisms of retrieving relevant data. So the interdisciplinary project was of interest for all participating groups.

Acknowledgments

We are very grateful to all the students who have worked and are still working within the project. During the last four years, about 25 of them prepared their term papers or «Diploma» theses mainly by implementing the first and second prototypes. Without their contributions, the geoscientific database system would have remained a «paper project».

5 References

ANDERSEN, F. & LINNEMANN, V. & PISTOR, P. & SÜDKAMP, N. (1988): Advanced Information Management Prototype — User Manual for the On-line Interface of the Heidelberg Data Base Language (HDBL) Prototype Implementation. — Technical Note TN 86.01, IBM Heidelberg Scientific Center; Heidelberg.

APERS, P.M.G. & WIEDERHOLD, G. [Eds.] (1989): Proc. 15th Internat. Conf. on Very Large Data Bases, Amsterdam; Palo Alto (Morgan Kaufman Publishers).

BANCILHON, F. (1988): Object-Oriented Database Systems. — Proc. Principles of Database Systems, Assoc. for Computing Machinery, New York: 152—162; New York.

BORK, H.-R. & HENSEL, H. (1988): Computer-aided Construction of Soil Erosion and Deposition Maps. — In: VINKEN, R. [Ed.]: Construction and Display of Geoscientific Maps derived from Databases. — Geol. Jb., **A 104:** 357—371; Hannover.

BRITTON LEE Inc. (1984): IDM Software Reference Manual, Version 1.7.; Los Gatos (CA).

CHEN, P. P. (1976): The Entity-Relationship Model — Toward a Unified View of Data. — ACM Transactions on Database Systems, **1:** 9—36; New York.

DITTRICH, K. [Ed.] (1988): Advances in Object-Oriented Database Systems. — Proc. 2nd Internat. Workshop on Object-Oriented Database Systems, Bad Münster am Stein; Berlin (Springer).

— & DAYAL, U. [Eds.] (1986): Proc. Internat. Workshop on Object-Oriented Database Systems, Pacific Grove (CA), 1986; Washington D.C. (IEEE Computer Society Press).

EHRICH, H.-D. & LOHMANN, F. & NEUMANN, K. & RAMM, I. (1988): A Database Language for Scientific Map Data. — In: VINKEN, R. [Ed.]: Construction and Display of Geoscientific Maps derived from Databases. — Geol. Jb., **A 104:** 139—152; Hannover.

GÜTING, R.H. (1989): Gral: An Extensible Relational Database System for Geometric Applications. — In: APERS, P.M.G. & WIEDERHOLD, G. [Eds.]: Proc. 15th Internat. Conf. on Very Large Data Bases, Amsterdam: 33—44; Palo Alto (Morgan Kaufman Publishers).

HEITLAND, M. (1988): Der Einsatz eines Geo-Datenbanksystems auf dem Gebiet der Bodenerosion. — Studienarbeit, TU Braunschweig; Braunschweig. — [unpubl.]

HENRICH, A. & SIX, H.-W. & WIDMAYER, P. (1989): The LSD-Tree: Spatial Access to Multidimensional Point and Non-point Objects. — In: APERS, P.M.G. & WIEDERHOLD, G. [Eds.]: Proc. 15th Internat. Conf. on Very Large Data Bases, Amsterdam: 45—54; Palo Alto (Morgan Kaufman Publishers).

HOHENSTEIN, U. & GOGOLLA, M. (1988): A Calculus for an Extended Entity-Relationship Model Incorporating Arbitrary Data Operations and Aggregate Functions. — In: BATANI, C. [Ed.]: Proc. 7th Internat. Conf. on Entity-Relationship Approach; Amsterdam: 1—20; Amsterdam (North-Holland).

HORN, D. & SCHEK, H.-J. & WATERFELD, W. & WOLF, A. (1988): Spatial Access Paths and Physical Clustering in a Low-Level Geo-Database System. — In: VINKEN, R. [Ed.]: Construction and Display of Geoscientific Maps derived from Databases. — Geol. Jb., **A 104:** 123—138; Hannover.

JUNGCLAUS, R. & NEUMANN, K. (1988): Benutzerhandbuch zum ersten Prototypen des Braunschweiger Geo-Datenbanksystems. — Informatik-Ber. Nr. 88-01, TU Braunschweig; Braunschweig. — [unpubl.]

KERNIGHAN, B.W. & RITCHIE, D.M. (1977): The C Programming Language; Englewood Cliffs (Prentice-Hall).

LIPECK, U.W. & NEUMANN, K. (1987): Modelling and Manipulating Objects in Geoscientific Databases. — In: SPACCAPIETRA, S. [Ed.]: Entity-Relationship Approach: Ten Years of Experience in Information Modelling (Proc. 5th Internat. Conf. on the Entity-Relationship Approach, Dijon 1986): 67—86; Amsterdam.

LOHMANN, F. (1987): Vorüberlegungen für den zweiten Prototyp eines geowissenschaftlichen Datenbanksystems. — Informatik-Ber. 87-04, TU Braunschweig; Braunschweig. — [unpubl.]

— (1988a): Processing Non-Standard Database Objects in a Higher Level Programming Language — An Abstract Data Type Approach. — Proc. Internat. Workshop on Software Eng. and its Applications, Toulouse: 1141—1159; Toulouse.

— (1988b): Abstrakte Datentypen fuer Objekte, Objektmengen nnd Objektlisten. — Interner Ber. GEO-88-02, TU Braunschweig; Braunschweig. — [unpubl.]

— (1988c): Abstrakte Datentypen für Geometrien. — Interner Ber. GEO-88-03, TU Braunschweig; Braunschweig. — [unpubl.]

— & NEUMANN, K. & EHRICH, H.-D. (1989): Entwurf eines Datenbank-Prototyps für geowissenschaftliche Anwendungen. — In: HÄRDER, T. [Ed.]: Datenbanksysteme in Büro, Technik und Wissenschaft. — Proc. GI/SI-Fachtagung, Zürich: 43—57; Berlin (Springer).

NEUMANN, K. (1988): Eine geowissenschaftliche Datenbanksprache mit benutzerdefinierbaren geometrischen Datentypen. — Diss., TU Braunschweig; Braunschweig. — [unpubl.]

OSTERHOLD, A. (1987): Der Einsatz eines geowissenschaftlichen Datenbanksystems im Bereich der Ökologie. — Studienarbeit, TU Braunschweig; Braunschweig. — [unpubl.]

PECKHAM, J. & MARYANSKI, F. (1988): Semantic Data Models. — ACM Computing Surveys, 20: 153—189; New York.

SAMET, H. & SHAFFER, C.A. & NELSON, R. & HUANG, Y.-G. & FUJIMARA, K. & ROSENFELD, A. (1988): Hierarchical Data Structures for Geographic Information. — In: VINKEN, R. [Ed.]: Construction and Display of Geoscientific Maps derived from Databases. — Geol. Jb., A **104:** 97—98; Hannover.

SCHEK, H.-J. & WATERFELD, W. (1986): A Database Kernel System for Geoscientific Applications.— Proc. 2nd Internat. Symp. on Spatial Data Handling, Seattle: 273—288; Williamsville.

STONEBRAKER, M. & WONG, E. & KREPS, P. & HELD, G. (1976): The Design and Implementation of INGRES. — ACM Transactions on Database Systems, **1:** 198—222; New York.

TEGETHOFF, R. (1988): Die Kommunikation zwischen Anwendungsprogrammen und einer geowissenschaftlichen Datenbank. — Diplomarbeit, TU Braunschweig; Braunschweig. — [unpubl.]

TIETJEN, S. (1988): Einrichtung von virtuellen Basiskarten auf einem geowissenschaftlichen Datenbanksystem am Beispiel von Island. — Studienarbeit, TU Braunschweig; Braunschweig. — [unpubl.]

VINKEN, R. (1988a): Digital Geoscientific Maps — A Research Project of the Deutsche Forschungsgemeinschaft (German Research Foundation). — In: VINKEN, R. [Ed.]: Construction and Display of Geoscientific Maps derived from Databases. — Geol. Jb., A **104:** 7—20; Hannover.

— [Ed.] (1988b): Construction and Display of Geoscientific Maps derived from Databases. — Geol. Jb., A **104:** 475 p.; Hannover.

WATERFELD, W. & WOLF, A. & HORN, DOE. (1988): How to Make Spatial Access Methods Extensible — Proc. 3rd Internat. Symp. on Spatial Data Handling, Sydney: 321—335; Columbus, Ohio.

WIRTH, N. (1985): Programming in Modula 2. — Third corrected Ed.; Berlin (Springer).