Electronic Journal of Polish Agricultural Universities is the very first Polish scientific journal published exclusively on the Internet, founded on January 1, 1998 by the following agricultural universities and higher schools of agriculture: University of Technology and Agriculture of Bydgoszcz, Agricultural University of Cracow, Agricultural University of Lublin, Agricultural University of Poznan, Higher School of Agriculture and Teacher Training Siedlce, Agricultural University of Szczecin, and Agricultural University of Wroclaw.

ELECTRONIC
JOURNAL
OF POLISH
AGRICULTURAL
UNIVERSITIES

2004
Volume 7
Issue 1
Series
GEODESY AND CARTOGRAPHY

BARTONĚK D. 2004. DATABASE DESIGN FOR THE MEASURING IN FIELD **Electronic Journal of Polish Agricultural Universities**, Geodesy and Cartography, Volume 7, Issue 1.
Available Online http://www.ejpau.media.pl

DATABASE DESIGN FOR THE MEASURING IN FIELD

Dalibor Bartoněk
Institute of Geodesy, Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic

ABSTRACT

The measuring in field is the most commonly used activity in various industrial branches. To be the results useful the data must be stored in a suitable database. There is a problem: to use either one of most of commercial database systems or to create the special purpose designed database? This question is discussed and an own approach of the database design is presented in this paper. The goal is the designed database with minimum of memory capacity occupied and the most rapidly information provided. It is supposed that the basic database design rules e.g. to avoid redundant data (normalization), ensure that the relationships among attributes are represented or facilitate the checking of updates for violation of database integrity constraints are respected. A special optimization method for common attributes reduction, for storing strings of variable length and for database reorganization is used. The database model was implemented in the information system for anticorrosive protection of pipeline used in gas enterprises in the Czech Republic.

Key words: Database, object, measured values, quantities, relational model, E-R diagram, entities, common and individual attributes, equivalent classes, overlay table, optimization.

INTRODUCTION

The measuring of quantities on the object in field belongs to common activities in many spheres of individual usage. The gained values serve either to ensuring the run of certain devices or they inform the users e.g. in the frames of prophylactic check up. In order to take effective advantage of these values it is necessary to store it in optimal designed database. There are two possibilities: 1. to use a commercial database software product (dBASE, MS Access Oracle etc.), 2. to create own special purpose database in any programming language

(Borland Pascal, Delphi, C etc.). The commercial database systems, however, were designed for general purpose with the wide offer of functions which is connected with 2 main problems:

- a lot of functions lead to complications of the system and inconvenient usage (some functions would hardly be made use of, others the missing ones would have to be programmed by the experts themselves),
- by complications we mean tedious training of the operators who lack in experience with computers.

Having discussed all “pros and cons“ the authors together with a group of potential users decided to make their own solution in Borland Pascal programming language (MS DOS application), later in Borland Delphi (application for Windows). The results were used in the system for anticorrosive protection of gas-pipeline.

ANALYSIS OF THE MEASURING PROCESS

Lets ponder about measuring on the objects of various types. On each of them we measure different quantities in different time. The moment of measuring can be:

a. regularly (periodical e. g. monthly, quarterly, yearly etc.)
b. irregularly that means the measuring are done at random according to our needs.

The value of the measured quantity is under these hypotheses a function of two variables:

$$\boldsymbol{v = f(t, p),} \quad (1)$$

where
$\boldsymbol{v}$ is measured value,
$\boldsymbol{t}$ is the type of the object,
$\boldsymbol{p}$ is the period of the measuring.

The value of the measured quantity depends upon the type of the object and the period of the measuring. It means that of every type of the object various quantities depending upon the time of the measuring can be measured. In every period (monthly, yearly etc.) a different kind of quantities on the given type of the object of all types bat at the same time the required measuring plan which is the description of the individual period of the measuring further function relationship (1). There are several basic types of the object. Every object can be either of basic type or the combination of several basic types. This fact should be taken into consideration as well.

THE DATA MODEL

For the storage of all data measured in field we have decided to choose the most commonly used the relational database. Relational data model will contain these entities:

a. Object and sub-objects with the ISA hierarchy,
b. Period of the measuring,
c. Measuring (with measured values),
d. List of quantities representing the equation (1).

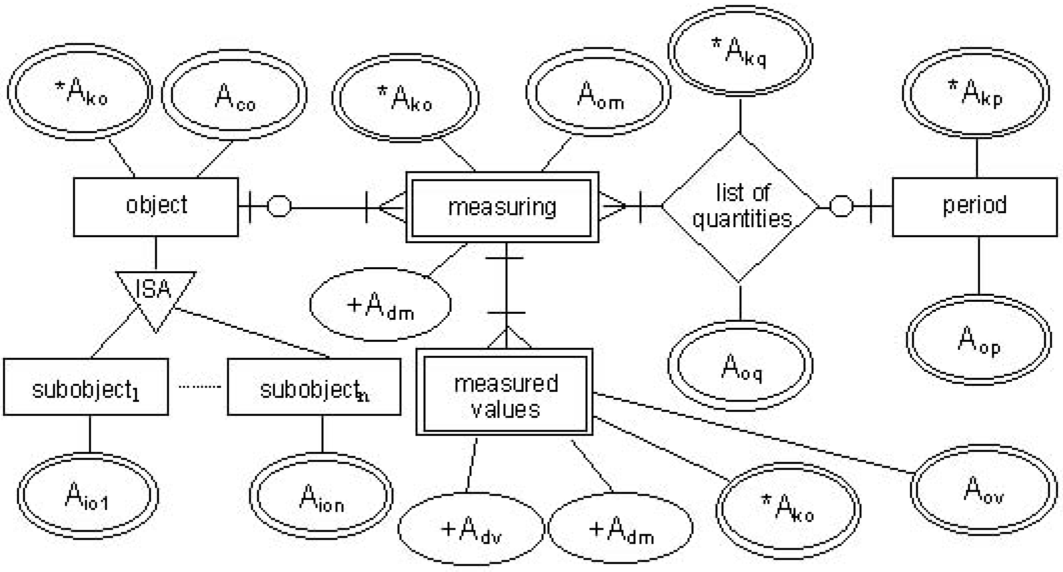
Object and period are the strong entity sets, whereas measuring and measured values are the weak entity sets, because they are dependent on the previous entities according to the equation (1). The list of quantities represents the relationship between period and measuring and realizes the function in the equation (1).
The objects attributes from the point of view of belonging to the objects can be divided into 3 classes:

a. key attributes - A_k,
b. common attributes for each type of the object - A_c,
c. individual attributes, which distinguish the individual types of the objects - A_i. These attributes can be from the point of view of data stored length divided into two groups:

- fixed length,
- variable length.

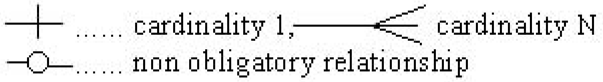
E-R diagram of the data model of the measuring in field shows figure 1.

Fig. 1. E-R diagram of the data model of the measuring in field



Explanations:

A_{ko}, A_{kp} and A_{kq} – primary key attributes of the object, period and list of quantities,
A_{co} – common attributes of the object, A_{io1} … A_{ion} – individual attributes of the sub-object No 1, … ,n,
A_{dm} – discriminator of the measuring entity, A_{dv} – discriminator of the measured values entity,
A_{om} – other attributes of the measuring entity, A_{ov} – other attributes of the measured values entity,
A_{op} – other attributes of the period entity, A_{oq} – other attributes of the list of quantities entity,



IMPLEMENTATION OF THE DATA MODEL

The data model on Fig. 1 was transformed into tables of relational database according to the rules described in (Korth, H. F., Silberschatz, A., 1996), (pokorný, J., 1998). The tables are of two types:

- Static, that means they have a fixed structure (tables of object, sub-objects, period, measuring and list of quantities),
- Dynamic – the structure of which is variable according to the equation (1) – table of measured values.

As for the static tables, the most complicated problem was with the creation of tables of object and sub-objects (ISA hierarchy). To avoid data redundancy the method of attributes overlay was used. The principle consists in dividing the sub-objects into basic types so that any sub-object can be composed by combination of one or some objects of these basic types. Thus the number of sub-objects is very (approximately by one order) reduced. Then is the overlay table created the structure of his is shown in Fig. 2. Rows in overlay table consists attributes A_{io1} – A_{ion}, in columns are basic types of sub-objects. In the individual cell we write either symbol "*" (string date type) or "#" (other date type). Another reduction of the database structure is by creation of overlay table achieved. The process consists in 2 phases:

1. Deleting all of the columns in the table 1, which have the null values in all own rows. Objects of these types are sufficiency covered by attributes in the basic table of object (common attributes). We obtain a new table.
2. In the new table we step by step join columns according these rules:
   a. We join these columns that have in his rows the some symbols.
   b. Further we join columns that is different in *n* symbols only according these criterion:

$$n\sum_{i=1}^{n} c_i t_{Oj} < m\sum_{i=1}^{m} d_i\, t_{Ok} \qquad (2)$$

where
n is number of attributes, in which are lower-level entity sets O_j and O_k (sub-objects) different,
m is number of attributes, in which are lower-level entity sets O_j and O_k (sub-objects) the same,
c_i is length of i. attribute (common for lower-level entity sets O_j and O_k) in Byte,
d_i is length of i. attribute (in which are lower-level entity sets O_j and O_k different) in Byte,
t_{Oj} is estimation of frequency of occurrence lower-level entity sets O_j in database,
t_{Ok} is estimation of frequency of occurrence lower-level entity sets O_k in database,

Table 1. Attributes overlay table

Attributes/ Subobjects	O_1	O_2		O_n
A_1	*			*
A_2				#
A_m	#			#

After the reduction process every row in the table 1 must contain at least one character * or # (completely overlay). Columns in the table 1 represent equivalent classes of sub-objects. We can define *2n* tables in the database, where *n* is number of columns in the table 1. *N* tables are for attributes of string date type (there are a special compression method for string saving used on the internal database level) – see Fig. 3, *n* tables for attributes of non-string attributes) – see Fig. 4. Internal structure of the database for saving records of objects is in Fig. 2 and. We use two files; first for the basic data e.g. common attributes A_{co1}, that must have a value (not null), second file contains two groups of attributes: A_{co2} (common attributes, that may be null value), A_{io}, (individual attributes of object type). The field "Info object" uses trigger for record resize. The measuring data are saved also in two files. First file for all attributes of measuring $A_m = A_{ko} + A_{dm} + A_{om}$, second file for measured values. This file has 3 parts: 1. "info-measuring" serves for trigger to record resize, 2. "bitmap" is sequence of bits (flags) corresponding to measured values. If the flag bit is zero, the correspondent value is null, if it is "1", the correspondent value is valid. D_1 is length of bitmap, D_2 is length of array of measured values.

Dynamic table for measured values is created with the help of the table of quantities list part of which is shown in the table 2. This table realizes the dependency in equation (1). The quantity Q_i is loaded into the dynamic table of measured quantities if is measured on one of the sub-objects types $<T_1, \dots T_m>$ and in one of the periods $<P_1, \dots P_n>$ written in the row - see table 4.

Fig. 2. Internal structure of object records saved in the file

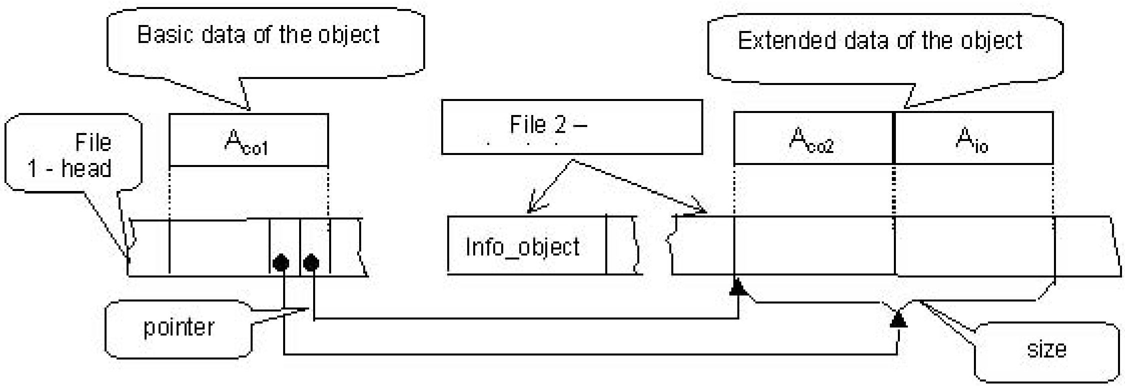


Table 2. Part of text attributes of sub-object type

Attribute i	Attribute i+1	…	Attribute m
…	…	…	…
string i	string i+1	…	string m
…	…	…	…

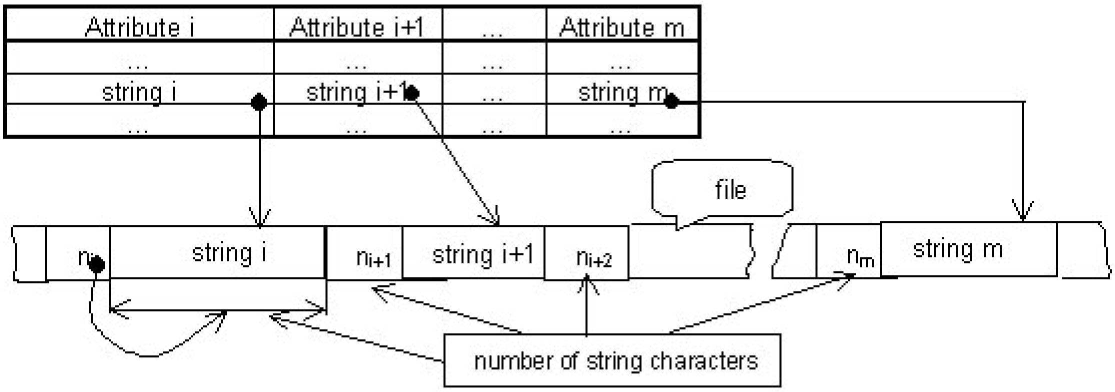


Fig. 3. Compression method of saving the text attributes into the file

Table 3. Measured quantities

Q_1	Q_2	…	Q_k
…	…	…	…
1.45	12.61	…	4587
…	…	…	…

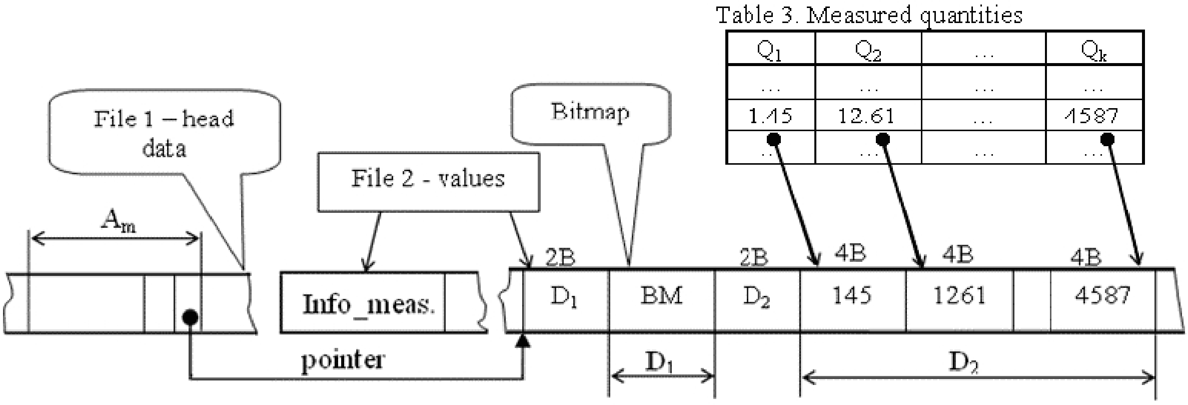


Fig. 4. Method of saving non-text attributes and measured values into the file

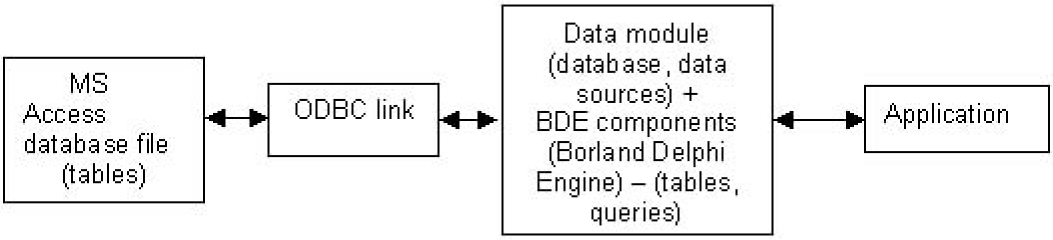
Table 4. Measured quantities

Quantities	Sub-object types			Periods of measuring		
Q_i	T_1	.....	T_m	P_1	.....	P_n

UPDATE OF THE IMPLEMENTATION OF THE DATA MODEL

At present the database model was updated to run uder the Windows operating system. The MS Access was used as the base relational database system. The source code of the application is written in Borland Delphi and it is connected with the MS Access by ODBC interface through the data modul. The conception is shown in Fig. 5.

Fig. 5. Conception of the current database solution


MS Access database file (tables)
ODBC link
Data module (database, data sources) + BDE components (Borland Delphi Engine) – (tables, queries)
Application

The advantage of this conception consists in simple data manipulation (queries, filters etc.), whereas the insert database operation has to be in code source ensured against the primary key collision. If the primary key collision occurs, the MS Access causes exception error in the application.

CONCLUSION

The database model was used in the information system for anticorrosive gas-pipeline protection in gas-works in the Czech Republic. This system with the name GAS-ACOR has already been running for more then 10 years and it had been upgraded according to the users needs. First version was made in Borland Pascal for MS-DOS. The last upgrade consisted in introducing module for universal file print in the network environment. This module was created in Borland Delphi. The authors provide the users with help in the form of hot line or personally.

Last year it has been worked on the converting of the GAS-ACOR system from the MS-DOS system into Windows. Source modules are made in Borland Delphi. The testing run of the new system is operated in South-Moravia gasworks in Brno. The introduction of GAS-ACOR into practice will mean a considerable time saving and the increase of the efficiency of the anticorrosive protection.

REFERENCES

1. Korth H.F, Silberschatz, A., (1996), Database System Concepts. Third Edition. McGRAW-HILL.
2. Pokorný, J., (1998), Databázová abeceda. SCIENCE, Veletiny, [in Czech].
3. Sedláček, S., Bartoněk, D., (1996), GAS – ACOR. User manual. SHINE, Brno, [in Czech].

Dalibor Bartoněk
Institute of Geodesy, Faculty of Civil Engineering
University of Technology
Veveří 95 662 37 Brno Czech Republic
tel. 541147204, fax: 541147218
e-mail: bartonek.d@fce.vutbr.cz

Responses to this article, comments are invited and should be submitted within three months of the publication of the article. If accepted for publication, they will be published in the chapter headed ‘Discussions’ in each series and hyperlinked to the article.

[BACK] [MAIN] [HOW TO SUBMIT] [SUBSCRIPTION] [ISSUES] [SEARCH]