MTA SZTAKI Közlemények 32/1985 141-153

EXPERIENCES IN THE USE OF TWO DATABASE MANAGEMENT SYSTEMS FOR MICROCOMPUTERS

E. MUNIZ, M. FONFRIA, JORGE DE LA CANTERA
Central Research Institute for Computers
CUBA

INTRODUCTION

In the past few years, and as a result of the fast development in the microcomputer field and its introduction in almost all software areas, the general concept of database management systems (DBMS's) has suffered a little change. Until midseventies, database systems were conceived only for mainframes with fast I/O peripherals and large amounts of secondary storage (usually disk). This fact was dictated by the relative large resource requirements of such systems, given primarily for their ability to provide: a) high level of program and data independence, b) flexibility in the representation of data, c) high performance and efficiency in transaction-processing, d) supression of data redundancy, e) capability of searching through data according to its attributes, and f) procedures for data security and recovery, among others. In practice, only a few DBMS implementations include all of these features, however, some have come close.

While microcomputers are being introduced and gaining partidaries because of their interactive and simpler operation, increased the need for systems capable of integrate and compact rationally the currently available information. In this way, say "from bottom to the top" were approached the development of new systems featuring characteristics of DBMS's. In 1981, we could evaluate more than 10 of these systems.

However, even present-day DBMS's for microcomputers can not accomplish all of the objectives of a formal DBMS, although they are close according to the hardware currently available and offer the user a powerful tool for data processing.

In gerenal, these systems provide:

- an easy way of creating screens and database reports,
- facilities for data entry and its relationing with all other data,
- provisions for program and data independence,
- facilities for using data for many applications,
- tools for creating applications with minimum programming effort, leaving every time on the user free for concentrate on his logical design.

Inmersed in this context, the authors have had the opportunity of developing an application intended to be used as a management aid in the manufacturing of some equipmnet, and its implementation under two DBMS's for micros: dBASE II from Ashton-Tate and Sensible-Solution from O'Hanlon Computer Systems.

In this paper, we describe the application, the general features of both database systems, and offer the conclusions of thes realization, with the hope that they can help in giving some criterion about the use of these systems.

GENERAL FEATURES OF THE SYSTEM USED IN THE APPLICATION

dBASE II

dBASE II is a powerful tool for database management that allow very easy handling of small to medium-scale databases using English-like commands. The system has the following major features:

- complete creation of databases
- facilities for adding, deleting, editing, displyaing and printing database information
- data and program independence, i.e., changes to the program do not imply changes to the data and viceversa
- report generation from one or more databases
- use of the facilities of the terminal for editing data

dBASE II is typically interactive, having facilities for immediate correction of errors. A dBASE II program is a series of commands stored on a disk file which the user can execute by

means of the DO command. These command files do not need be previously compiled; commands are interpreted and immediately executed.

dBASE II allows the handling of relations (files) and memory variables. dBASE II files consist of a file header containing field descriptions, attributes and data all in compacted ASCII mode. The handling of files is accomplished in a manner transparent to the user. For example, for using a variable named COUNTER zero-initialized it is sufficient to utilize the command STORE 0 TO COUNTER without the need to declare or define COUNTER previously. The type of COUNTER (N: numeric, C: character, L: logical) is determined by the type of the source information, in this case numeric. Storing "ABC" to COUNTER will change its type to character.

If the user wants to modify the structure of a data file without loosing information, he can easily utilize a suitable combination of COPY, MODIFY STRUCTURE and APPEND commands without the need to modify the command files that reference the reestrructured file. An important feature of dBASE II is the ease of learning by non-specilaized people, the syntax of commands is very close to the human (English) language and also the documentation is clear and well presented.

dBASE II included a program named ZIP that allows the use of screens for data entry and also for displaying information. ZIP generates sequences of dBASE II commands that when executed perform the functions desired. Even though screen handling is not completely automatic in the sense that it has to be done by means of an external component, it has the advantage that the generated program can be edited and modified according to the user's needs.

In dBASE II direct access to relations' attributes is accomplished through dense index files, one for each attribute required by the user. When "opening" the relation the index file is specified and searches are carried out by FIND commands over the index field associated to the index file. If it is necessary later to change the search for another index field, the corresponging index file must be opened. The search command

(FIND) is very little flexible and do not allow automatic processing of records with duplicate keys. Reports can be obtained by the REPORT command, which can generate a program ready for execution. This command simplifies the process of report generation without the need for porgramming, though these facilities are of low level.

dBASE II shows up some inconveniences when relationing one file to another. This has to be done by means of the JOIN command, which outputs a file that in many cases is not completely necessary, leading to waste time and disk memory.

dBASE II allows processing of data files generated by other processors i.e., BASIC, FORTRAN, PASCAL and can produce files compatible with these processors. Also, recent versions include means for calling programs segments written in machine code, though this feature is very specific and aside from the general framework of users.

One limitation present in dBASE II is that only two files can be opened simultaneously, restricting the development of medium to large-scale professional applications. This limitation has been overcome in later releases of the systems.

dBASE II attains an efficient use of the hardware resources, requiring little memory. Also, it is necessary to point out that along with the simplicity of its language and the documentation it is easy to install it.

SENSIBLE-SOLUTION

This system shows up general features similar to that of dBASE II, however, his implementation differs in many aspects.

Sensible-Solution groups conceptually its functions into "tasks" that can be requested via a functions menu. The most important are:

- execution of command files
- data dicitonary maintenance
- creation of screens
- command editing
- compile command files written in Sensible-Solution

- creation of programs
- database queries
- report generation
- restructuring files.

Command files or language statements are not processed interactively, rather, they have to be previously compiled.

In Sensible-Solution all of the data including auxiliary variables must pertain to files controlled by the data dictionary (files RECFLE. MS, RECFLE.KS, FLDFLE.MS and FLDFLe.KS). Database information is structured around files comprised physically of two files: one contains user data in compacted ASCII mode called Master Data File, and the other contains a binary tree of pointers to records in the Master Data File; this file is called index-key-file and do always exist from the declaration of the file. This index file allows indexing a file up to 9 index fields and facilitate direct searching of attributes by the FIND statement, which can be indistinctively performed over any of the index fields. The various formats of the FIND statement are powerful tools for processing relations based upon its attributes, and the handling of two related files is performed easily by one of its variants (31 FND.REL.RC, Find Related Record).

Data definition is rigid and not very transparent to the user, and is always controlled by the data dictionary. For example, for using an auxiliar variable the user must invoke the task that updates the data dictionary and define the variable in an auxiliar file together with its declaration, and initialize that file. Similarly, for modifying the structure of a file without loosing information, for example, modifying the length of a field, it is necessary to make changes in the data dictionary, reorganize the file and possibly reindexing it. Also, the changes are not "transparent" for the associated programs since all programs referencing the restructured file must be recompiled.

One powerful tool offered by Sensible-Solution is the definition and handling of screens. Screen definition includes labels

and windows to fields of a file that together with command files controls allow data entry, update and retrieve information from the specified file.

Sensible-Solution language is very peculiar and seems to be distant from the usual syntactical structures, that has two major disadvantages: one is the difficulty of learning by non-specialized users and the other is the necessity of a special task for editing command files, since the operating system's editor can not be used.

Sensible-Solution offers good tools for report generation. Through the definition of the report formats sophisticated reports can be obtained. This procedure adds great flexibility even though stands for the need of programming the report.

Sensible-Solution does not achieve automatic compatibility with other processors. One important feature of Sensible-Solution is its ability to open simultaneously up to 10 files, which is a great advantage when developing complex applications.

Sensible-Solution requires a large amount of primary and secondary storage, its components need to be distributed within 2 floppy disks each with at least 300K of free space, representing a constraint for its use on systems with 8 inch single density drives. Also, the installation procedure is not simple and requires a predetermined allocation of components within the two floppy disks.

DESCRIPTION OF THE APPLICATION

It is evident that all manufacturing processes involve the control of the different parts and elements that form the product, for example, it is necessary to know the list of elements that comprise every part of the product, the composition of the different models to be produced, and the stock of elements to carry out the process. The namual control of these aspects is a tedious and complex task, requiring generally one or more persons completely dedicated to this activity. For this reason, it was decided the implementation of a programming system on a microcomputer intended to provide an automated control of the

necessary operations supporting the manufacturing process, such as inventory control, allowing additionally that people unfamiliar with microcomputers could use the system, saving human and monetary resources, and obtaining reports with a high degree of reliability and in a small response time.

The most important functions offered by the system are:

- list elements or parts involved in the whole manufacturing process,
- list the composition of every part that constitutes the device,
- checking for the posibility of produce a specified number of parts depending upon current stock of elements printing deficits,
- extraction of components necessary for manufacturing a specified quantity of the device or some parts,
- listings that facilitate contracting elements for manufacturing a specified quantity of the device printing suppliers and prices,
- automation of the extraction and reception of components.

The programming system was designed to be supported by a database management system.

All the information handled by the system is included in the relations named PART, STOCK, TYPE, TRANSAC, and MODEL1, MODEL2, etc. Following is the description of the relations.

Relation PART

ORDER	ELEMENT	EQUIV1	EQUIV2	EQUIV3	PART:CODE	QUANTITY

ATTRIBUTE	DOMAIN
ORDER	numeric values of the form xx.yy, where xx identifies a particular type of element and yy is a consecutive
ELEMENT	the name of all elements used in the production of the device (all models)

EQUIV1	up to 3 equivalents to the main element, these fields can be blanks
EQUIV2	in case the equivalents do not exist.
EQUIV3	The inclusion of these attributes adds more flexiblity in the extraction of elements, since the main element and its equivalents are handled indistinctively, when the stock of a component is exhausted
PART:CODE	code of all parts comprising the device. Every part is formed by one or more elements.
QUANTITY	numeric values indicating how many elements are used in a part.

This relation is interpreted as follows: ELEMENT (or EQUIV1, EQUIV2, EQUIV3) of type xx is used in PART:CODE in quantity QUANTITY.

Relation STOCK

ORDER	ELEMENT	MANUFACTURER	CODE	PRICE	QUANTITY:ON:HAND

RECEPTIONS	EXTRACTIONS	DELIVERY

ATTRIBUTE	DOMAIN
ORDER	the same values as the corresponding in the relation PART. The objective of this field in this relation is to help in the control of the completeness and consistency of data and to facilitate its updating
ELEMENT	the same values as ELEMENT, EQUIV1, etc., in the relation PART. The ORDER of an element and its equivalents is the same.

MANUFACTURER	name of the manufacturer of each element
CODE	codes for identifying elements in the warehouse
PRICE	unit prices of the elements
QUANTITY: ON:HAND	stock of elements
RECEPTIONS	the sum of all receptions of each element
EXTRACTIONS	the sum of all extractions of each elements
SHIPPING	quantities extracted of each element to be printed in promissory notes. These values are retained until the user specifies another function involving extraction of components; this allows to obtain as many copies of promissory notes as desired and also in case of a system failures while printing, to repeat the procedure.

The relation STOCK is interpreted as the stock of all elements in the warehouse and the transactions in which each element is involved.

Relation TYPE

CONSEC	TITLE

ATTRIBUTE	DOMAIN
CONSEC	the values xx of the attribute ORDER of the relation PART
TITLE	description of the category of component represented by CONSEC

For example, the tuple (02, cables) can be interpreted as that the elements of type 2 are cables. This title is used in the listings obtained from the system's functions.

Generally, database management systems for microcomputers do not offer facilities for automatic recovery of information in case of failures, for this reason it was designed a relation named TRANSAC that keeps a history of transactions performed (receptions and extractions), ensuring more data security.

Relation TRANSAC

SPECIFICATION	QUANTITY	DATE	TYPE

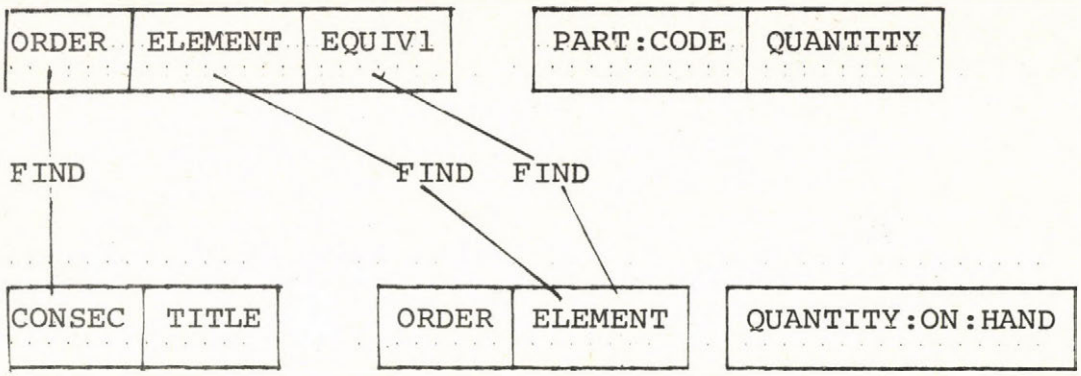
ATTRIBUTE	DOMAIN
SPECIFICATION	element or part handled in a transaction
QUANTITY	amount of elements or parts handled in the transaction
DATE	date when the transaction was performed
TYPE	the type of the transaction:(e)xtraction or (r)eception

There are other relations named MODEL1, MODEL2,... MODELn, that keeps the composition of each of the different models.

Relation MODELJ

PART:CODE

The following diagram shows operations that are performed among relations in the major part of the system's functions.



Another operation that is performed when handling a model of the device is the following:

(description modelj)

obtaining the subset of description that comprises the modelj

CONCLUSIONS

In general, Sensible-Solution performed the generality of the functions of the application faster then dBASE II due particularly to the command files have been already compiled, all related files can be opened simultaneously and the power and diversity of the search statements.

About Sensible-Solution, many of the functions could be integrated given the possibilities of screens and command file controls.

The programming and debugging resulted easier in dBASE II, particularly taking into account that a simple modification in the length of a field causes in Sensible-Solution modifications to the data dictionary, restructuring the file, etc. By the other hand, dBASE II's programming language is more mnemonic, consequently, it is learned easier and faster.

For simple applications handling a few files, dBASE II is a good choice. If the number of relations involved increases and should perform frequent and/or complex operations on databases, dBASE II can be ineffective, being advisalbe in this case the implementation of the application using Sensible-Solution.

REFERENCES

1. J.MARTIN, "Computer Database Organization".
2. C.J. DATE, "An introduction to Database Systems."
3. E. MUNIZ, and J. de la CANTERA "Design of a Control System under dBASE II". (to appear).
4. K.S.BARLEY, and J.R.DRISCOLL "A Survey of Database Management Systems for Microcomputers", BYTE, Nov 1981.
5. A.S.MICHAELS, B.MITTMAN, and C.R. CARLSON "A comparison of the Relational and CODASYL approaches to Database Management", Computer Surveys, Vol. 8, No. 1, March 1976.
6. J.DEMETROVICS, L.HANNAK and L.RONYAI "On functionally completeness of prime-element algebras", MTA-SZTAKI, Közlemények 25/1982.
7. M.W.BLASSEN, and K.P.ESWARAN "Storage and access in relational databases" IBM Systems Journal Vol. 16 num. 4/1977.
8. E.F.CODD, "A Relational Model of DATA for Large Shared Data Banks", Communications of the ACM Vol. 13/num. 6/June 1970.
9. E. LOWENTHAL, "Database systems for local nets", Datamation, august/1982 Vol. 28 num.9.
10. G.FLOAM, "Putting a Database on a mini", Datamation Vol. 22 num. 6. June 1976.

Tapasztalatok két adatbázis kezelő rendszer használatáról mikroszámítógépeken

E. MUNIZ, M. FONFRIA, J. DE LA CANTERA

Összefoglaló

A szerzők az Ashton-Tate-féle dBASE II és az O'Hanlo Computer Systems-féle Sensible-Solution adatbáziskezelő rendszerekkel kapcsolatos tapasztalataikról számolnak be.

Опыты использования двух систем обработки данных на микро-ЭВМ

Е. Миниз, М. Фонфрия, Й. Де Ла Кантера

Резюме

Авторы описывают опыты проведенные с системами dBASE II /Ashton-Tate/ и Sensible-Solution /O'Hanlo Computer Systems/.