MICROCOMPUTER BASED DATABASE MANAGEMENT SYSTEMS IN SUPPORT OF OFFICE AUTOMATION

Csaba J. Egyhazy
Computer Science Department
Virginia Polytechnic Institute & State University
Falls Church, VA 22042

CS830005

CS 820015

MICROCOMPUTER BASED DATABASE MANAGEMENT SYSTEMS IN SUPPORT OF OFFICE AUTOMATION

Csaba J. Egyhazy
Virginia Polytechnic Institute and State University

The evolutionary advancements in microprocessor technology as it relates to database management systems (DBMSs) are discussed. Practice and experience with five commercially available database management systems are reported, based mostly on data gathered from a series of interviews focusing on comparison among systems.

Several prototype systems specifically designed to meet the needs of office information systems are identified, their conceptual framework ascertained and capabilities described. Finally, remarks on the limitations and future of microcomputer based DBMSs are made.

Categories and Subject Descriptors: H.2.0 [Database Management], H.4.1 [Office Automation], B.7.1 [Microprocessors and Microcomputers], H.2.1 [Data Models]

General Terms: Database Management Systems, Microcomputers, Office Automation, Office Information Systems.

Additional Key Words and Phrases: Practice and experience with microcomputer based database management systems.

INTRODUCTION

Among the primary factors driving the development of the mostly non-textual component of Office Information Systems are microprocessors and database management systems [10]. The powerful marriage of cheap hardware and downscaled software we call database management

Author's address: V.P.I. & S.U. Graduate Center, 2990 Telestar Ct., Falls Church, Va. 22042

systems for microcomputers provides the office environment of tomorrow with the resource it needs for take off.

The first commercially available microcomputer based database management systems were downscaled versions of well known DBMSs, such as Ingres and Oracle, and customized DBMSs, such as dBase II and Condor. In addition, the literature continuous to report on more and more prototypes, [16], [18] attempting to explore new directions for the design and development of Office Information Systems.

The following sections will examine the incremental evolution of microcomputer based database management systems as an integral part of Office Information Systems. The next section provides a brief overview of microprocessor technology. Section 3 introduces some of the products on the market today, namely dBase II, Condor Series 20, MDBS III, MicroINGRES and Oracle Version 3. A summary of the research in the area, in terms of prototypes, is given in Section 4. Practice and experiences with existing micro-computer based DBMSs is reported in Section 5. Finally, the last section discusses some of the limitations of microcomputer based DBMSs, the trends in managing and distributing office data and the current research being conducted at VPI & SU.

2.0 MICROPROCESSORS

In 1971, INTEL Corporation introduced the first microprocessor, the INTEL 4004. Although microprocessors of varying sizes have been built and marketed, the more common microprocessors have an internal word size of 4, 8 or 16 bits. The earliest microprocessors, including the INTEL 4004, were 4-bit microprocessors. Recent advancements in semiconductor technology have resulted in 32 bit microprocessors. Although some existing microprocessors have an internal word size of 32 bits (i.e., the Motorola 68000), thev are classified as 16-bit microprocessors based on the size of their external data path. The 32-bit microprocessors listed below have both a 32-bit internal word size and a 32-bit external data path. Table 1 lists some of the most popular microprocessors.

8-bit	16-bit	32-bit
Intel 8080/8085	Intel 8086	Intel 80386 (late 1984)
Motorola 6800	Motorola 68000	Motorola 68020 (Dec. 1983)
Zilog Z80	Zilog Z800	Zilog Z80,000 (late 1983)
MOS technology 6502		

Table 1. Popular 8 and 16-bit microprocessors and announced 32-bit microprocessors

By the very nature of their size, microcomputers possess certain inherent limitations in their ability to support large software components, including DBMS packages. Most notable of these limitations are memory capacity and secondary storage.

Although recent technological advancements have resulted in 16-bit (and announced 32-bit) microprocessors, the microcomputer based on an 8-bit microprocessor is still the most common. As table 2 shows, an 8-bit address can only distinguish among 256 bytes of memory.

Address length (bits)	Memory Capacity (bytes)
4	16
6	64
8	256
10	1K
12	4K
16	64K

Table 2. Address Length vs. Memory Capacity.

Memory capacity constraints result in two limitations, the size of a program and the speed of program execution. The latter point requires further explanation.

Floppy disk drives are more commonly used than hard disks with microcomputers, mainly because of their relative low cost. Floppy disks are available in three sizes: 8 inch, 5 1/4 inch and the so-called sub 4 inch disk. The 5 1/4 inch disk is the most popular.

The use of floppy disks have certain disadvantages. First, because the read/write head is in contact with the disk, some of the coating wears off. Eventually, the disk may become unusable, resulting in the loss of the information stored on the disk. Second, because the disk itself is pliable, a floppy disk can only rotate so fast before centrifugal force begins to misshape the disk. This, of course has an impact on the transfer rate of data from a floppy disk to the computer´s memory. Finally, floppy disks are quite limited in the amount of information that they can store. A 5/14 inch floppy disks has approximately 35 tracks (the number varies with manufacturer) per side; this is roughly equivalent to 30-35 pages of typed information.

Because very few programs could fit into 256 bytes of memory, (especially any program resembling a DBMS), the 8-bit microcomputer has complex instruction cycles which allow the processor to fetch addresses from program memory in 8-bit segments and reassemble them inside the CPU. Consequently, microcomputers are typically slower than larger computers because of this inherent inefficiency in addressing memory.

3.0 MICROCOMPUTER DATABASE MANAGEMENT SYSTEMS

There are many products on the market today whose manufacturers claim to be database management systems for microcomputers. One recent survey found 48 such advertisements in the trade journals and computer hoppyist magazines. Of these 48, the authors of the survey received responses to a request for information from 20 of the companies. The survey concluded that only two could properly be called database management systems; the others were really file management systems [2].

Rather than survey the entire list of available products, this paper attempts to focus on several of the "true" DMBS packages available for microcomputers and compare them not only with each other, but also with the concepts inherent in all database management systems. To that end; this paper focuses on five commercially available products:

Condor Series 20

dBase II

MDBS III

MicroINGRES

Oracle Version 3

The first two products are classified as single-user while the others are considered multiple user systems. The last two products are down scaled versions of commercial relational DBMS, running on minicomputers, for use on micros. This transformation is particularly significant for those familiar with the product, since the tasks of conversion and training become straight forward.

In addition to the above systems, the single-user Unify DBMS [11], running UNIX on Z800, 8086 and 68000 microprocessors, could be upgraded to support serveral users.

Condor Series 20 is a fully relational database system that includes multiple file capability. Table 3 lists the system requirements for Condor.

Microprocessor: 8080/8085, 8086/8088,Z80

Memory:	64K Bytes (IBM PC requires 80K bytes)
Operating Systems:	CP/M, MP/M, CODS, TURBODOS, MSDOS, CP/M-86, PCDOS
Secondary Storage:	2 to 4 floppy disks (At least 300k bytes total capacity). Hard disks may be used if supported by operating system.
Terminal:	Cursor addressable CRT with screen erase and line wrap.
Printer:	Line or character printer with form feed.

Table 3. Condor Series 20 System Requirements.

Condor Series 20 (hereafter called Condor) is written in Z80 assembly language and supports five data types: alphabetic, numeric, alphanumeric, data and dollar amounts. The system performs data type checking and range checking on a per field basis, if requested.

Condor includes an interactive data definition language, which allows the user to create tuples via the DEFINE <tuple name> command. The user enters field names (attributes) and data type. Further, Condor allows the user to specify attribute properties such as required field, minimum and maximum values, and a default value. As part of the definition of a tuple, Condor creates a "fill-in-the-blank" form to be used for future additions of new tuples. Condor also allows the user to dynamically add attributes to a tuple.

Condor has an English-like data manipulation language called Condorquery and a report writer called Condorwriter. However, it provides no programming language interface; all queries are performed interactively. Predefined queries can be placed in command files and executed by invoking the command file. Condor permits the passing of arguments to a command file.

Among the Condor commands of special interest are the select, project and join operators. These three operators result in the creation of a temporary file called RESULT. The user can then operate directly on RESULT. The join operation is limited to joins of two sets (tables).

Condor provides limited integrity checking -- it performs data type checking and (optionally) range checking. However, Condor provides no security mechanism and does not perform recovery after a failure.

dBase II was one of the first relational database management system for microcomputers on the market. Table 4 lists the system requirements of dBase II.

Microprocessor:	8080/8085, Z80
Memory:	48K bytes (dBase II users locations 5C to A400).
Operating Systems:	CP/M (version 1.4 or 2.x)
Secondary Storage:	One or more floppy disks or hard disks operating under CP/M.
Terminal:	Cursor addressable 24 line by 80 column terminal.
Printer:	Optional.

Table 4. dBase II System Requirements.

dBase II is written in 8080 assembly language and supports the character, numeric and logical data types.

The data definition language of dBase II allows a user to create, modify and delete files. All operations are performed interactively. Once a tuple is defined, dBase II allows for the addition of attributes and for changing the name of an attribute.

Like Condor, dBase II has no host language interface and instead provides an extensive set of English-like commands for performing queries. These queries are performed interactively at the terminal or through the use of command files. dBase II does not allow the passing of parameters to command files. However, the DML of dbase II contains programming constructs such as IF...THEN...ELSE, DO..WHILE and GOTO statements and the ability to declare and call procedures. Further, command files may be nested up to 16 levels deep. dBase II does not provide a programming language interface; however, because it provides programming like statements, its DML is itself a quasi-programming language.

dBase II allows the user to be operating on two files at a time. Through the invocation of the USE command, the user specifies the file on which he wishes to work. In order to work on two files simultaneously, the user specifies one as primary and the other as secondary. The join operator is thus restricted to two files.

There are no provisions in dBase II for security and no recovery from failure procedures. dBase II does perform type checking on attributes.

MDBS III, from Micro Data Base Systems, Inc., is an extended network structured database management system. It supports the full CODASYL network architecture, plus provides many-to-many relationships. The system requirements for MDBS are shown in Table 5.

Microprocessor:	8080/8085, 8086/8088, Z80, Z8000, 6502, 68000
Memory:	20K bytes for Z80 version 22K bytes for 8080 version 30K bytes for 6502 version
Operating Systems:	CP/M, MP/M, PCDOS, TURBODOS, MSDOS, TRSDOS, OASIS-16, UNIX
Secondary Storage:	1 to 8 floppy disks or hard disks.
Terminal:	Cursor addressable CRT.
Printer:	Optional.

Table 5. MDBS System Requirement.

MDBS III contains five major components:

Data Description Language (DDL). The DDL allows the user to specify the logical structure of the database. The specification automatically creates an entry in the data dictionary.

Data Management System (DMS). The DMS carries out queries and provides application language interfaces for PASCAL, PL/1, COBOL, BASIC, C, FORTRAN and Assembler. In conjunction with the DDL, the DMS provides such features as data compression, data encryption, security checking, data type and range checking, elimination of unwanted data redundancy and locking protocols.

Query Retrieval System (QRS). QRS is a non-procedural query language, designed for ease-of-use by non-programmers. Among the features provided by QRS are selective retrieval, wildcard string matching, arithmetic expressions, sorted output, user-defined macros and tabular end-user views. The Interactive Data Manipulation Language (IDML) is a subset of the Query Retrieval System.

Recovery and Transaction Logging System (RTL). RTL prevents data loss resulting from media failures, power failures, and other abnormal system failures. It supports rollback and selective recovery. In the event of failure prior to completion of a transaction, MDBS automatically restores the database to the state that existed prior to that transaction. In the event of disk failure, the recovery utility (RCV) is invoked to reapply logged transactions to an old, backup copy of the database.

Design Modification Utility (DMU). DMU permits the user to make changes to a database, including changing physical file names, expanding the size of a database area, and obtaining statistics regarding disk space utilization.

MDBS III provides feasibility range checks on a per field basis. Security is provided in the following manner. For each user, the system stores a password and both a read and a write level in the user profile. Each record and each item can have associated with them separate read and write levels (range 0-255). Read (write/update) access is allowed only if the read (write) level of the user is greater than or equal to that of the record or item accessed.

MicroINGRES is a full function relational DBMS available on Motorola 6800-based microcomputers using the UNIX operating system. The system requirements are given in Table 6.

Microprocessor:	6800
Memory:	11 Megabyte
Operating Systems:	UNIX
Secondary Storage:	10 MB of hard disk storage
Terminal:	No particular requirement
Printer:	No particular requirement

Table 6. MicroINGRES System Requirements.

MicroINGRES maintains an integrated data dictionary, which contains information about the database schema. This information includes storage structures, secondary indices, integrity rules and protection information. Facilities for defining alternate views are also provided. MicroINGRES supports tables that are stored as hash files on up to five keys, as ISAM files, or as heaps (sorted or unsorted). Any number of secondary indices may be defined for a table. MicroINGRES supports the following data types:

(i) 1, 2 and 4-byte integer

(ii) 4 and 8-byte floating point

(iii) character (fixed length strings)

MicroINGRES provides a data manipulation language called QUEL (QUEry Language). Commonly used command sequences can be stored in command files. Queries may either be made interactively at a terminal or through a host language interface. Currently, there is only an interface for the C language; additional high-level language interfaces are planned. (INGRES provides FORTRAN, BASIC, PASCAL, PL/1 and COBOL interfaces in addition to the C interface; future MicroINGRES high-level language interfaces are likely to come form that group).

QUEL provides the following main data manipulation operations:

retrieve (finds data based on specified arguments)

append (adds rows to a table)

replace (updates columns in a row)

delete (deletes rows)

QUEL supports very complex queries; commands can span multiple tables (up to 10). As such, QUEL provides the select, project and join operations.

In addition to QUEL, MicroINGRES provides a subsystem called Query by Forms (QBF). QBF provides most of the power of QUEL, but allows beginning users to perform updates and retrievals without having to remember the QUEL syntax. QBF is one component of the MicroINGRES Forms Management System; the others are VIFRED, a forms editor, and Report by Forms, a form-based report specifier used in conjunction with the MicroINGRES Report Writer.

MicroINGRES QUEL is identical to the VAX-based INGRES QUEL. Thus it contains features for asserting integrity constraints and protection predicates. It also allows a user to propagate access rights. Further, MicroINGRES provides utilities for loading and unloading data, as well as transaction logging, rollback, and crash recovery. In addition, the MicroINGRES journaling facility automatically saves rows to a second disk; journaling can be enabled for an individual table or an entire database.

Oracle Version 3 is a full function DBMS available on Motorola 6800-based microcomputers., the Altos 68000 based system, the Texas Instrument Professional Computer and the Fortune 32:16. The system requirements are given in in Table 7.

Microprocessor: 6800,68000, TIPC and Fortune 32:16

Memory: 64k bytes

Operating System: 98% operating-system independent

Secondary Storage: 5MB of hard disk storage

Terminal: none

Printer: none

Table 7: Oracle V3 System Requirements

Oracle Version 3 maintains an integrated data dictionary, which contains information about the database schema and facilities for defining alternate views. The system supports host language interface and it asserts integrity constraints.

Oracle Version 3 provides SQL Plus a query language, a data manipulation language, a data definition language and a data control language.

Traditional data processing systems have, until recently, been designed making little use of the interactive nature of the computer environment. Facilities provided by large DBMSs are much more suited for batch processing and applications development environments rather than an end user interactive environment [10]. Therefore, as microcomputers with DBMSs become the centers of vital data processing systems, users will create and manage their own local databases. In distributed systems, microcomputer based DBMSs will complement mainframe systems, making local databases accessible to a larger number of people.

4.0 PRACTICE AND EXPERIENCE WITH MICROCOMPUTER BASED DATABASE MANAGEMENT SYSTEMS

In general, reported experiences with microcomputer based DBMSs in an office environment are almost nonexistant, limited to perhaps a half a dozen isolated paragraphs in trade journals. One of the principal reasons for this is the lack of office information systems with a reasonable track record on microcomputer based DBMS. Most of todays Office Information Systems (OIS) with DBMS capability run on mainframes or minicomputers, in some instances still in batch mode [10]. There is, therefore, not a large enough sample size of commercial users to warrant a formal survey at this time. Our research group at VPI & SU is planning to conduct such a survey in the near future, as part of the second phase of the microcomputer based DBMSs study. Although experiences with these systems in the office environment are few, there were a number of noteworthy observations made to us by users and researchers alike during the first phase of the study. The following section highlights the findings of an intensive six month effort togather data on experiences with microcomputer based DBMS in office environments.

Condor Series 20 and dBase II are DBMSs developed specifically for microcomputers. dBase II, being the first relational DBMS for microcomputers in the market, has enjoyed wide acceptance among the single user population. Although, the large majority of these users operate in the home/small business environment as opposed to an Office Information System. Among the positive features of dBase II, users cited its ease of use, its single integrated language and its interactive/full screen orientation. On the negative side they cited its lack of sophistication, the lack of recovery mechanism and the limitations on the number and size of the relations. In a controlled testing environment Badal and Reive [24] found that the excessive number of command language statements required for some simple operations was a hindrance to users.

Condor Series 20, and in particular Condor III, has been developed to be run on IBM personal computers running under PC DOS operating system. The result of some testing performed at the National Bureau of Standards showed high marks for the ease of use of its interactive data definition language, data manipulation and report writer.

MDBS I, the single-user version of MDBS III, enabled the personnel department of a medium size corporation to operate more efficiently. General satisfaction with the incorporation of this DBMS for microcomputers in the office environment was acknowledged by professionals and office workers alike [23].

MDBS III is a multi-user DBMS based on the network data model. It was, until recently, the only multi-user system available. While considered by users a fairly sophisticated DBMS it is down graded because of the need of separate languages (DDL, DMS and QRS), each with it's own syntax and semantics and the excessive length of its linking tables for microprocessing implementation.

A striped down version of the MDBS III database management system was compared, in terms of performance, to a dBase II at the Naval Postgraduate School [24]. The results of the test showed the stripped down version of the MDBS III outperforming the full fledged dBase II.

MDBS III, MicroINGRES and Oracle Version 3 are probably the three most complete and powerful microcomputer database management systems in the market today. They all support a multi-user environment and necessary locking mechanism. In addition, they provide integrity constraints, security, transaction logging and crash recovery. These products are certainly prime candidates for inclusion in office information system. Both MicroINGRES and Oracle Version 3 have just been announced and the only experience with the systems we can report on is our own. MicroINGRES, a downscaled version of INGRES database management system running under UNIX on minicomputers, has on interface to C language which facilitates the tailoring of applications. The most striking feature of MicroINGRES is the subsystem called Query-By-Forms (QBF). In QBF each relation is mapped into a form instead of being presented to the end user as a relation. The user enhances the basic form by drawing boxes repositioning fields and adding background text. The conceptual schema is therefore visualized not in terms of relations but by forms, an object easily recognizable in office environments. The CCA [25] system, currently under development by the Computer Corporation of America, is a management information system supported by INGRES and MicroINGRES. It provides the user with a combination of sight and sound by incorporating text, pictures, facsimile and digitized voice within a graphical data space. New developments, such as these is the topic of the next section.

5.0 PROTOTYPES

The current popularity of personal computers and office information systems has activated the research in DBMS for microcomputers.

The Evolutionary Database Management System (EDBMS) is a prototype [9] that can run effectively on a 16 bit microcomputer with 1M Byte of memory and a single Winchester drive. The main conceptual feature of the system is the Control Data Base module; an extended data dictionary that allows for an evolutionary system. Additions and alterations to the data definitions and schema can be thus be made at any time. Multiple physical definitions are given different version numbers and a provision for incremental conformation of existing occurrences to the most recent version is provided. EDBMS implements an extended network data model and a full relational model of data. Both views are directly embedded in the run-time system for efficiency. The integrity constraint processing features include inter-record constraints and iner-record constraints such as set membership constraints. The system is implemented in a precompiler language called IRAFOR. IRAFOR currently translates into standaard ANSI 77 FORTRAN. The interfaces to the operating system, as with Oracle V3, have been reduced to only two small modules or approximately 2% of the total software.

Research with direct manipulation [14] microcomputer based DBMSs such as Query-By-Example (QBE) resulted in the office system prototype known as Office-By-Example (OBE). The user manipulates data in the database through templates and forms [15]. The OBE functionality is built on top of a database system, thus illustrating once more the increased dependance of office information systems on microprocessor technology and DBMSs.

Some universities have begun experimenting with a data model for office information systems which retains the structure of forms within the intervals of the database system. The main objective of these systems is to increase user friendliness by replacing the traditional concept of record or relation by the easily recognizable form. For instance a group of researchers [16] at the University of Toronto have been experimenting with a distributed office information system built upon a DBMS using forms as an object which contains both data and a set of operations allowed on those data. Thus, forms became analogous to abstract data types in that each type of form consists of a set of attributes and a specific set of operations. The system is built upon the MRS relational data manager, specifically tailored for microcomputers [12], with the original prototype implemented on LSI 11/23's running UNIX.

IBM has recently announced work on an experimental distributed database system tailored to the office environment [18]. Each workstation consists of personal databases and public databases. Issues such as concurrency control and recovery are the major road blocks in the development of the prototype system.

The Cedar DBMS [19] is a prototype based on an entity relationship data manager which provides explicit transaction management. The primary conceptual feature of the system is the database server [20], where all system and user information resides. Conceptually, it is acting as a file cabinet for the workstations. Furthermore the database server maintains status information on the execution of each task. A task being defined as an individual transaction which may pass through multiple activities during the course of its execution.

In addition, a few expert system prototypes have recently, been reported in the literature [16] [17]. An expert system consists of a database that contains the collection of information that experts in the field use to make decisions and an inference scheme, the set of rules that experts apply to the information. The office semantics project at MIT [21] is an expert systems that helps the user achieve a goal given a database containing information about the organization and the goals of an activity. The Odyssey system [22] uses knowledge about trip planning to assist a user in filling out a collection of electronic forms in preparation for a trip.

6.0 LIMITATIONS OF MICROCOMPUTER DBMS AND RELATED TRENDS IN OFFICE INFORMATION SYSTEMS

Most commercially available database management systems lack several key components of a "true" DBMS. Chief among these are:

Integrity constraints

Security

Recovery

This paper has only addressed several database management system products at the upper end (in terms of capabilities) of those available. Yet, neither Condor III nor dBase II provide any security and little in the way of integrity constraints. These are not serious flaws, however. Both these systems are single user systems. Thus, in reality, the only security required may be physical security of the microcomputer. This is probably sufficient for office information systems in well secured buildings. Also, making integrity the responsibility of the user is not necessarily unreasonable. Further, very few large DBMS systems provide a full range of integrity constraints; it is not suprising that a microcomputer DBMS package would provide very little integrity checking.

Lack of recovery mechanisms in Condor III and dBase II is more serious. It puts the burden of making frequent copies of data on the user in order to prevent loss of his data. Any failure will result in the loss of all updates since the last time the user made a copy.

Several trends for the future of microcomputer database management systems have already begun. The increase in the number of

16-bit microcomputers and the announcement of availability of several 32-bit microprocessors indicate that larger and faster systems will be available. These systems will support larger amounts of memory than current microcomputers and thus be responsive to office information system. Although Lochovsky [10] says that: "The ubiquitous microprocessor is rapidly invading the office workplace", it will take time before the corresponding software issues are totally resolved.

Secondary storage is also becoming larger, with faster access times. Hard disks are becoming cheaper. In 1980 eighty percent of the 5 1/4 inch Winchester disks had less than 10 megabyte capacity and by 1988, fifty percent will have approximately a 25 megabyte capacity [4]. These larger systems will tend to support more complex operating systems, as evidenced by the availability of UNIX for a 6800-based (and Z8000-based) microcomputer. These systems will be multiple user systems. Advancements in hardware lead inevitably to advancements in software.

At Virginia Polytechnic Institute and State University we are presently engaged in a long term study of microcomputer based DBMSs for office automation, gathering data on practice and experience, measuring/assessing their impact and reporting on trends in the technology.

References

[1] Abbott, Jack L. Condor Series 20 DBMS, Byte 7:12, (December, 1982).

[2] Barley K. and Driscoll, J. A Survey of Database Management Systems for Microcomputers, Byte 6:11, (November 1981).

[3] Edelson, Roger H. dBase II: A Relational DBMS for CP/M Interface Age 6:8, (August 1981).

[4] Glazer, Sarah High-end, 16-bit Microprocessors Gravitate toward MC68000 Chips, Mini-Micro Systems 16:2, (February, 1983).

[5] McGlynn, Daniel R. Modern Microprocessor System Design: Sixteen-Bit and Bit Slice Architecture. New York: John Wiley and sons, (1980).

[6] Machrone, Bill MDBS: A Database Management System, Microsystems 3:3 (May/June 1982).

[7] Weiss, Harvey M. Dawn-scaling DBMS to the Microworld, Mini-Micro Systems 14:4, (April 1981).

[8] Halpern, Brent T. and Korth, Henry F. An Experimental Distributed Database System. ACM Proceedings: Database Week; San Jose, California (May 23-26, 1983).

[9] EDBMS: Evolutionary Database Management System; Product Description (1983) - Information Research Associates; Austin, Texas.

[10] Lochovsky, Fred H. Improving Office Productivity: A Technology Perspective. Proceedings of the IEEE, Vol. 71, No. 4, (April 1983).

[11] UNIFY: Product Description

[12] King K. J. and Maryanski, F. Information Management Trends in Office Automation. Proceedings of the IEEE, Vol. 71, No. 4 (April 1983) pp. 519-528.

[13] Query-By-Forms, Relational Technology Inc., Berkeley, Ca.; (1982).

[14] Schneiderman, B. The future of interactive systems and the emergence of direct manipulation. Proceedings NYU Symp. on User Interfaces (May 1982).

[15] Zloof, M. M. QBE/OBE: A language for office and business automation, IEEE Comput., vol. 14, No. 5 (May 1981) pp. 13-22.

[16] Tsichritzis, D. OFS: An integrated Form Management System, Proceedings Conference on Very Large Data Bases, ACM (1980) pp. 161-166.

[17] Tsichritzis, D. and Christodoulakis, S. Message Files. ACM Transactions on Office Information Systems, Vol. 1, No. 1 (January 1983) pp. 88-98.

[18] Hailpern, B. and Korth, H. An Experimental Distributed Database System Proceeding of ACM's Database Week (May 1983); Special Session on Databases for Business and Office Applications.

[19] Brown, M. R., Cattell, R. G., and Suzuki, N. The Cedar DBMS: A preliminary report. Proceedings of ACM SIGMOD Conf. (May 1981) pp. 205-211.

[20] Maryanski, F. Data-server design issues AFIPS Conf. Proc. (1982); (June 1982) pp. 249-438.

[21] Barber, G. J. Supporting organizational problem solving with a workstation. ACM Transactions on Office Information System vol. 1, (1983).

[22] Fikes, R. E. Odyssey: A knowledge-based assistant. Artificial Intelligence, vol. 16 (1981) pp. 331-362.

[23] Ferris, D. Micro DBMS. Computer Decisions (May 1983).

[24] Badal, D. Z. and Reive, R. Database Management in the Microworld -- DBASE II and MDBS III. ACM's Database Week; Special Session on Databases for Business and Office Applications (May 1983).