DATA MANAGEMENT SYSTEMS FOR MICRO-COMPUTERS: A SURVEY

by

Alan J. Hughes

Eileen M. Lynch

Edward A. Stohr

October, 1981

Presented at the Conference on Surall Business Systems, Rutgers University, November 6-7, 1980.

Working Paper Series
STERN IS-81-69

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

I. INTRODUCTION

The rapid growth of micro-computer technology in recent years has suddenly made cheap computing available to small business users and hobbyists. The performance capability of micro-computers is now approaching that of the mini-computers of a few years ago. As was the case for mini-computers, we can expect that micro-computer manufacturers and software vendors will offer increasingly sophisticated user support and software packages. Operating systems are now becoming generally available together with a variety of packages for common business applications. Among these packages are many File Management Systems (FMS'S) and several Data Base Management Systems (DBMS's). An FMS, provides facilities for file creation, maintenance, retrieval and report generation. Usually a number of independent files are created each supporting a single application. Some FMS's can simultaneously access several files either for retrieval or for update (eg. master and transaction files). A DBMS on the other hand creates and manages a non-redundant data structure in which various data relationships are maintained and which can support a number of different applications.

This paper surveys the emerging field of data management software for micro-computer systems. Our objectives are to provide a guide to the available software and to survey actual and potential users and applications. For reference, excellent surveys of data base management systems for maxi- and mini- computers are contained in [1] and [5] respectively.

The market for general purpose data management software is discussed briefly in Section 2. The potential is large but as yet almost completely unsatisfied. Some of the problems involved in developing software for small business users are discussed in Section 3. This section also describes a desirable set of software characteristics for the micro-computer market.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

The results of a survey of micro-computer manufacturers and software vendors are described in Section 4. A list of the systems and vendors is given in Appendix I and a summary of the vendor survey questionnaire in Appendix II. All of the systems are available for machines with less than or equal to 64K memory and 8-bit microprocessor chips. Some are available for a wide variety of machines and operating systems; some vendors are planning version for the new 16-bit micorprocessors. The software features offered vary widely including simple single file retrieval system, multi-file retrieval systems, multi-file FMS's designed to assist in traditional business functions such as payroll and inventory control and full-fledged DBM's. Surprisingly, in terms of the features provided by this software we find some systems which are approaching the sophistication of their maxi-computer counterparts.

The packages in the sample were chosen to be typical examples covering a range of different system types. The list is not exhaustive. Nor does inclusion of a system in the list imply that it is to be preferred to another not on the list. Our objective is to provide an overview of the kinds of software available, not to recommend any particular package. Indeed it would be foolish to do so. First, because the correct choice of a package depends heavily on the intended application and secondly because new systems and upgrades of existing systems are being continually introduced.

In addition to the 'supply-side' it is important to consider the existing and potential uses of data management technology. The results of a small survey of DBMS users are reported in Section 5. The user survey questionnaire is also summarized in Appendix II.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

2. MARKETS FOR MICROCOMPUTER DATA MANAGEMENT SOFTWARE

The major markets for data management software are:

- o Small business, professional and non-profit organizations
- o Medium-large businesses
- o Small business software developers
- o Personal and home use

The first category listed above is likely to be the most important in the short to intermediate term. If we consider a small business as one with less than $4 million sales there is a potential market of perhaps one million companies. However it will be some time before the utility of computers for many companies (especially those with sales less than $2 million) becomes self-evident. Two related applications - word-processing and mailing have already established themselves as practical and useful for many small businesses with micro-computers at the smaller end of the range. More general business applications will be next. These range from simple file retrieval applications (eg. maintaining a list of customers) to traditional business applications such as inventory control. The special problems involved in developing software for small business will be discussed in section 3.

A number of users in our survey were developing decentralized special purpose systems within large organizations. These users were more sophisticated and were prepared to spend time learning the data management package and developing their applications. They purchased the more powerful file management and data base management systems.

A number of software developers have also purchased the more sophisticated file management and data base management systems and are using them as building blocks for developing business application packages for resale to end users.

Finally there is a market for file management systems for personal use. At this stage it is not clear how quickly this market will develop beyond a

relatively few hobbyist users. The volume of retrieval transactions for typical home applications such as recipes and lists of personal friends does not merit the extra overhead of computerized methods when compared with manual systems. The situation may change as information utilities linked to home television sets become widespread. In this case consumers may get the ability to tap-off limited files of information for special processing in their own 'distributed' data bases.

3. DESIRABLE PRODUCT CHARACTERISTICS

Development of micro-computer software for small business users presents a special set of problems and challenges never before faced by the software industry. In the first place machines in a suitable price range (say $5,000 to $15,000) have severe contraints in terms of memory capacity, speed and operating system capability. Secondly, the systems must be suitable for operation by naive users with neither the time nor the inclination to invest in supporting a computerized system. Larger companies always have been able to specialize their work forces for such tasks as data entry and validation, system operation and file maintenance. Small companies can not afford this. A few employees must perform all business functions. Time spent in supporting functions that are not directly revenue producing must be reduced to a minimum.

Figure 1 summarizes these and other characteristics of small business systems. The arrows indicate the major software implications. Our survey of available data management systems shows that designers have been sensitive to some of these needs. However, no current system comes close to satisfying all of them.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

SMALL BUSINESS CHARACTERISTICS	IMPLICATIONS FOR SOFTWARE
o PERSONAL	o GOOD 'HUMAN ENGINEERING'
o INDIVIDUALS PERFORM MANY FUNCTIONS	o RESPONSIVE SYSTEMS
o UNPREDICTABLE WORK ENVIRONMENT	o ON-LINE UPDATE (DEAL WITH ONE TRANSACTION AT A TIME)
o LITTLE UNDERSTANDING OF COMPUTERS	o FILE MAINTENANCE FUNCTIONS TRANSPARENT
o CAN NOT AFFORD TO PAY FOR PROGRAMMING	o 'TURNKEY' SYSTEMS o PORTABLE SYSTEMS
o UNDERGO GROWTH AND CHANGE	o FLEXIBLE SYSTEMS -CAN RESTRUCTURE
o LITTLE UNDERSTANDING OF BUSINESS SYSTEMS	o DESIGNERS OF APPLICATIONS MUST UNDERSTAND THE BUSINESS

FIGURE 1

DESIRABLE SOFTWARE ATTRIBUTES

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

Good Human Engineering

We have already mentioned the need for user-friendly systems. Section 4 will expand on some of the features provided by current software to meet this requirement.

Responsive Systems

Small business systems are characterized by an upredictable work environment and are driven by randomly occuring events. Because it is not possible to specialize employee work functions there is no possibility of buffering the system from its environment. This implies a requirement that systems be 'responsive' in the sense that the operator can quickly change the mode of operation without extensive file management operations and without compromising system integrity. The ability to move quickly from the processing of one transaction type to another or to a different file type is a desirable attribute.

On-Line Update

One aspect of a responsive system is to allow on-line update of files. Most of the file management systems allow this. Some systems allow the on-line user to batch update requests for greater efficiency; some allow a given update to be performed on a selected group of records. However, the primary access to the system is usually one record at-a-time. This contrasts with the master file/transaction file mode of processing which predominates in many larger businesses. The one transaction at-a-time mode fits with other characteristics of a small business and is suitable for a small volume of heterogeneous transactions. As volume grows however, specialization of data entry functions can yield great economies. Some of the file management systems already on the market anticipate this requirement.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

File Maintenance Fuctions Transparent

Since business users usually have little understanding of computers, file maintenance operations such as backup and recovery from system failure are lilely to be troublesome areas. As far as possible these functions should be handled automatically. This is an area of weakness for many existing systems.

Unless the user can afford a hard disk system the physical handling of diskettes in and out of different drives is unavoidable. However the user must also create and manage back-up files in an organized fashion. Only a few of the systems surveyed have a built-in function to back-up a file. As a consequence the user must understand the operating system in sufficient detail to perform this function.

Turnkey Systems

Since small businesses usually can not afford to invest in programming it is obvious that they need to purchase 'off-the shelf' computerized solutions to their business problems.

Where the applications are clear and simple as in mailing labels and maintaining simple lists of information this can be done easily enough. Several file management systems include quite elegant form design facilities to add an extra dimension of generality. Some are integrated with word-processing and mailing packages offered by the same vendor. However, more complex business applications such as payroll, inventory control and general ledger are harder to provide as 'turnkey' systems because of differences in business requirements.

Portability

The ability to transport programs from one computer system to another is important on both the supply and demand sides of the software market. For suppliers it gives a broader market and makes greater investment in development worthwhile. For the small businessman it reduces the risk of software

investment. There can be little doubt that the widespread use of Z80 and 8080 chip technology and the CP/M operating system has contributed greatly to the growth of the micro-computer software industry. Many software vendors are increasing their markets by releasing versions compatible with more than one computer system.

Flexible Systems

The need for systems which can endure change and growth is at least as great as in large systems. With data management systems this translates into an ability to change file definitions and data relationships. Very few of the file management systems allow new fields to be added to an existing record definition. Some of the systems require that indexes be rebuilt after the addition of a record to a file. At the other extreme an add-on package of 'restructuring' routines can be purchased for one of the data base management systems. This allows a complete redefinition of both records and record interrelationships under program control.

Good Business Systems

Many small companies do not have a good understanding of either manual or computerized business systems. This places an additional burden on software designers - they must thoroughly understand the business requirements. Exception conditions must be anticipated and an audit trail provided. The user must be protected from errors of omission and commission. When errors are discovered they must be easy to correct. Ideally, as the industry matures application systems will be designed by teams consisting of both business and systems specialists. Well designed efficient software in a narrow sense is a necesary but by no means sufficient prequisite for success. The software should perform the correct mix of business functions.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

4. SURVEY OF AVAILABLE DATA MANAGEMENT SOFTWARE

4.1 Classification Scheme

The data management systems examined in this study are listed in Table 1. They can be classified into a number of distinct types:

1. File Management Systems
   - 1.1 File Access Methods (FAM)
   - 1.2 Self-contained File Management Systems (FMS)
   - 1.3 Application Developments Systems (ADS)
2. Data Base Management Systems (DBMS)

As illustrated in Figure 2 the access paths through which users interact with their data form the major basis for the above classification. These access paths contain the logic of the user application with regard to some or all of the following:

File Description (FD) - Data item and record definitions

Schema Description (SD) - FD plus complex data relationships

Loading Data (LD) - adding data to create a file

Update (UP) - changing data values; additions; deletions

Information Retrieval (IR) - answering queries from a terminal

Report Generation (RG) - formatted reports; special forms

Job Processing (JP) - processing and computation involving multiple files and a number of job steps

Roughly speaking the classification indicates a trade-off between the ease of use and the functional generality and flexibility of the software package. Users are relieved of complex and onerous tasks when the logical functions listed above are carried-out by the software package rather than by user-written programs in a high-level language such as BASIC. However, unless the software package is particularly powerful the ability of users to manipulate and perform calculations on their data may be restric[illegible]

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

TABLE 1

CHARACTERISTICS OF DATA MANAGEMENT SYSTEMS

SYSTEM	HARDWARE	REQUIRED CPU SIZE	OPERATING SYSTEM	SOURCE LANGUAGE	SYSTEM TYPE	NO. OF USERS	PRICE	DATE RELEASED
ANALYST	Z80,8080 Micro	52K	CP/M MP/M	CBASIC 2	FMS	750	$250	Aug. 1979
CBS	Z80,8080	48K	CP/M	FORTRAN + ASSEMBLY	ADS	400	$395	Mar. 1980
CCA DATA MANAGER	APPLE II TRS-80 II	24K	TRSDOS	BASIC	FMS	1000	$100	1979
CONDOR SERIES 20/DBMS-I	Z80 Micro	48K	CP/M CDOS	Z80 ASSEMBLY	DBMS	60	$695	1980
CONDOR DBM-I	Z80 Micro	56K	CP/M CDOS	Z80 ASSEMBLY	DBMS	15	$5000	Jan. 1978
CROMEMCO DBMS	CROMEMCO SYSTEM II OR III	64K	CDOS	BASIC	FMS	2000	$298	1979
DBMASTER	APPLE II	48K	DOS 3.3	BASIC+ ASSEMBLY	FMS	600	$189	Nov. 1980
FMS-80	Z80,8080, 8085 Micro	32K	CP/M, CDOS, MP/M	8080 ASSEMBLY	ADS	200	$750	Oct. 1979
INFO MANAGER	NORTH STAR HORIZON	56K	(INTEGRAL) ASP 2.1.1	'C'	FMS	500	$499	Jun. 1980
MAXI MICRO MANAGER	TRS80 163	48K	TRSDOS, NEWDOS, VTOS	BASIC + ASSEMBLY	FMS	100	$99.95	Oct. 1980
MDBS	Z80, 8080, 6502 Micro	64K	CP/M NORTH STAR TRSDOS	ASSEMBLY	DBMS	300	$900	Nov. 1979
MICRO B+	Z80, 8080 Micro	N/A	CP/M	ASSEMBLY	FAM	100	$260	Sep. 1979
MICRO-SEED	Z80 Micro CP/M	64K	CP/M	FORTRAN + ASSEMBLY	DBMS	N/A	$900	1979
OS-DMS 'Nucleus'	OSI Challenger II or III	56K	OS-65U	BASIC	FMS	1000	$300	Jun. 1978
OZZ	Commodore 8032	32K	COMMODORE DOS	ASSEMBLY	FMS	150	$395	Nov. 1980
PRISM/ ADS	Z80, 8080, 8085 Micro	48K	CP/M	CBASIC	ADS	N/A	$795	Nov. 1980
PROFILE II	TRS-80 II	32K	TRSDOS	N/A	FMS	N/A	$179	1980
SELECTOR III	Z80,8080, 8085 Micro	46K	CP/M	CBASIC	FMS	1200	$345	Apr. 1979
SELECTOR IV	Z80, 8080, 8085 Micro	52K	CP/M	CBASIC	ADS	150	$550	Sep. 1980
SUPERKRAM	APPLE II, COMMODORE PET	16K	APPLEDOS CBMDOS	ASSEMBLY	FAM	500	$175	Nov. 1979

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

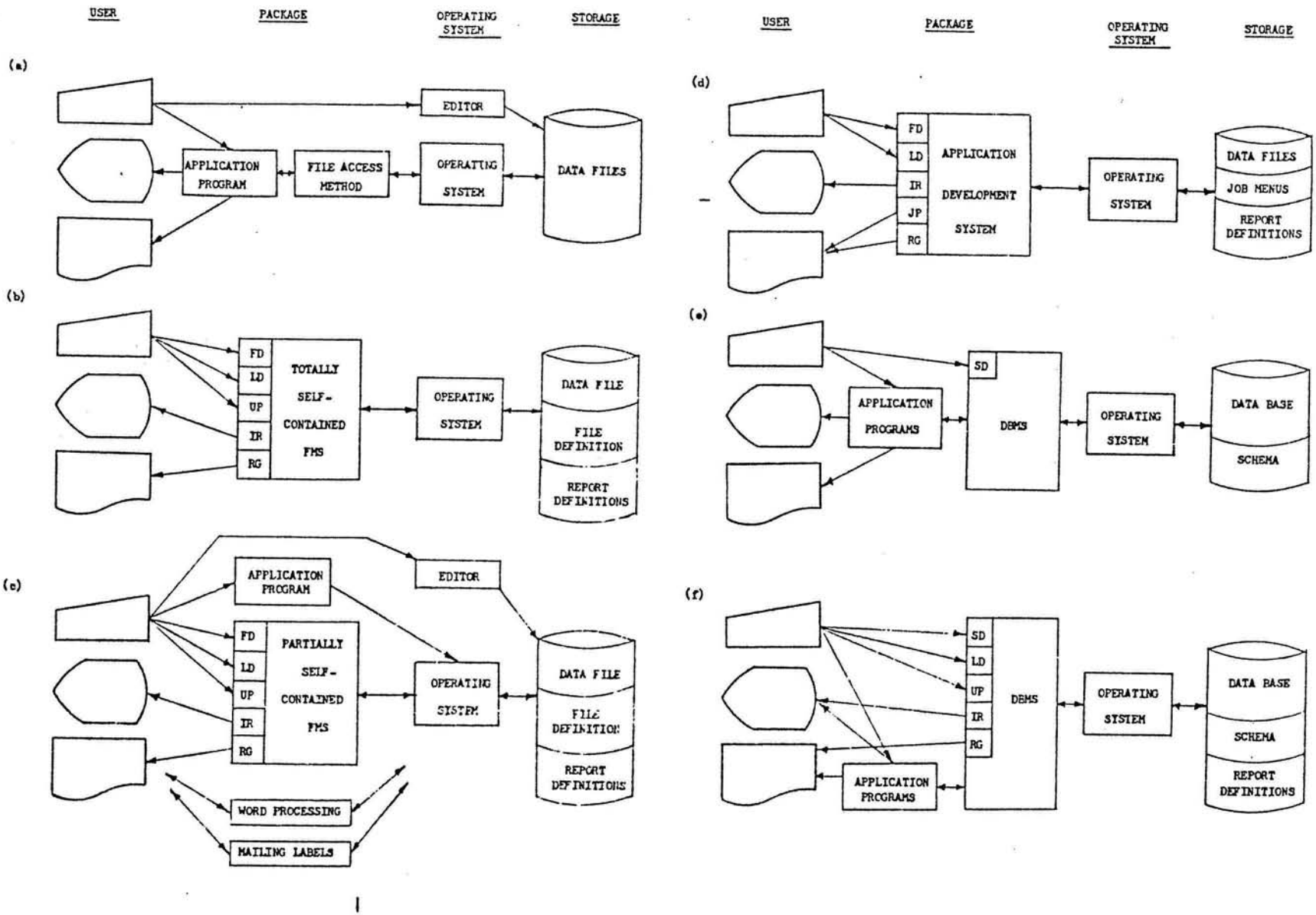


Figure 2

ACCESS PATHS FOR CLASSES OF DATA MANAGEMENT SYSTEM

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

File access methods (Figure 2a) provide efficient multi-key access from a host language program to the data files. However, the logic for all of the above functions must be coded within the calling programs.

The Self Contained FMS's are further categorized in Figure 2 according to to whether they are totally or partially self-contained. A software package is self-contained if it provides the complete interface between a user application and its data base. Thus all of the functions that are necessary for the application are performed using the language, prompting and menu selection features of the package itself. The user does not have to program in a "host" high level language (such as BASIC) or invoke utilities such as editors and sort routines that are not part of the package.

The totally self-contained FMS's (Figure 2b) provide a single interface tailored for naive users. This allows them to perform functions FD, LD, UP, IR and RG by specifying their requirements non-procedurally. i.e. users specify what has to be done but the details of how the processing is to be carried-out are left to the package. This works well for simple low volume applications. However, existing non-procedural interfaces generally can not cope with the complex sequence of processing needed to update and maintain files for more sophisticated applications. Greater generality is achieved in the partially self-contained packages (Figure 2c) where the FMS files can also be generated and/or accessed by separate application programs or by word-processing and mailing-label packages supplied by the same vendor.

An Application Development System (Figure 2d) represents a relatively new generalization of a FMS to allow transaction/master file processing. In addition to the FD, LD, UP, IR and RG interfaces of an FMS, an ADS provides a JP (job processing) interface through which users can define a sequence of job

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

steps involving simple computations, matching and merging and transfer of data between transaction and master files.

A DBMS (Figure 2e) is distinguished by its recognition of logical data relationships beyond those of the simple data item/record relationships of traditional file processing. It does this via an SD (schema definition) interface through which the user specifies the relationships that are to be maintained in the data base. The resulting logical description (schema) is stored separately from the physical data and consulted by the DBMS whenever storage or retrieval requests are carried-out. As illustrated in Figure 2f it is likely that future DBMS's will become partially self-contained by providing other non-procedural interfaces as has been done by the self-contained FMS's.

Although individual packages within each class differ greatly in the range of features offered, Table 2 summarizes the capabilities of each class of system in terms of the non-procedural interfaces that are usually provided. In the following sections we describe the main features of each type of system as revealed by the survey questionnaire described in Appendix II.

4.2 File Access Methods

These are designed to be accessed via host-language programs such as BASIC, and PL/1. Their purpose is to provide fast random and/or sequential access to files. Two of the surveyed systems fall in this category. Both allow retrieval from files based on multiple key values and allow for more than one file to be simultaneously open. One of the systems (SUPERKRAM) uses a VSAM-like file structure; the other (MICRO-B+) a B-tree approach. They are designed for use by programmers rather than end-users.

4.3 Self-Contained File Management Systems

These systems attempt to provide a single interface through which all file maintenance and retrieval functions can be performed. As indicated by Table 1

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

Table 2

Non-procedural Interfaces Provided by Various Systems

	1.1 File Access Method	1.2.1 Totally Self-contained FMS	1.2.2 Partially Self-contained FMS	1.3 Application Development System	2.1 DBMS	2.2 DBMS with Interfaces
File Description (FD) (item/record definitions)	No	Yes	Yes	Yes	Yes	Yes
Schema Description (SD) (Complex data relationships)	No	No	No	No	Yes	Yes
Loading Data (LD)	No	Yes	Yes	Yes	No	Yes
Update (UP)						
Single file updates	No	Yes	Yes	Yes	No	No
Computed fields	No	Usually	Usually	Yes	Sometimes	Sometimes
Information Retrieval (IR)						
Single file/multiple keys	No	Yes	Yes	Yes	No	Yes
Multiple files/multiple keys	No	Sometimes	Sometimes	Yes	No	Yes
Report Generation (RG)						
Columnar reports	No	Yes	Yes	Yes	No	Yes
Job Processing (JP)						
Master/transaction file Update	No	No	No	Yes	No	No

this category represents the largest number of available software packages. They provide facilities for record definition, data entry, update, retrieval and deletion combined with sort and report generation capabilities. Most of these systems are designed to allow access to only one file at a time and lead the user, via a series of formatted menu screens, through the various functions. They are geared for the naive user and are fairly successful in allowing a businessman with a limited knowledge of computers to run the system unassisted.

The systems falling in this category are: ANALYST, CCA, CROMEMCO DBMS, DBMASTER, INFO-MANAGER, MAXI-MICRO-MANAGER, OS-DMS, OZZ, PROFILE II and SELECTOR III. The following is an examination of the most common features of these packages.

General Characteristics - The file management systems surveyed are all less than 2 years old with a large increase in number this year. They are priced between $100 and $700. They have varying numbers of users from a few for the just released systems to more than 1000 for the more established systems. They are designed for list management, simple master file maintenance and accounting applications. The systems are evenly split between those written in a high level language (usually BASIC) and those written in assembler or machine code. They are also evenly split between those that can run only as a separate application and those that also allow invocation from user-provided application programs.

System Requirements - The systems surveyed were of two types - those designed to run on one manufacturer's hardware and operating system and those designed to run on a variety of machines usually with the CP/M operating system. The storage required varies from 24 k to 56 k with most systems requiring approximately 48K. Most systems run on 1 or 2 diskettes and can

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

accomodate hard disks. A language interface to BASIC is required in about half of the systems. All require some display device. Some also require a direct cursor address CRT and a printer.

User Interface - The following activities involve user interaction with a data management system: (1) File definition, (2) Entry of new data, (3) File update and maintenance, (4) Ad-hoc data base inquiry, (5) Generation of reports and special forms.

The interfaces provided by the surveyed systems fall in the following categories: (1) System prompts, (2) Menu selection, (3) Screen forms, (4) Command language.

Generally the systems provide a combination of methods and a uniform style of interaction for each of the five activities listed above. System prompts are invariably used - often in combination with (and in particular to control the use of) one or more of the other types of interface. Only a few of the systems provide a "Help" feature to generate a list of allowable user responses to each prompt. Selection from a menu is the most popular interface provided to guide the user through the various program options. File definition, data entry and file maintenance are usually handled by prompting for the required information on an item-by-item basis. Several systems employ screen forms and cursor-addressing to allow a "fill-in-the blanks" mode of data entry. Some of these systems allow users to define their own forms. A combined prompt and command language approach is used in some of the systems to allow users to compose retrieval requests involving several selection criteria.

Data Structures - All of the systems allow character and integer data types. Many systems have a date data type and some include others such as binary, logical, and real. The systems' limits as to field size, record size, and field label size vary widely. At the low end of the scale maximum record

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

size is limited to about 250 bytes for three of the systems which can be restrictive. For the most part file size is only limited by disk capacity. Almost all of the systems require the use of fixed length records without repeating fields or groups of fields. Most packages support computed fields (i.e. field content computed by package using a previously defined formula). Only one system allows multiple record types within the same file (e.g. header and trailer records).

Most systems use multiple access techniques with relative record number and primary key being the most frequently encountered combination. Five of the systems support secondary key indexes. Generally the indexes are only valid for a static file and have to be rebuilt in a separate step after records have been added. Only three of the systems allow more than one file to be open at a time.

File Definition - All of the packages provide an online screen editor to aid in file definition. Users are first presented with a menu of system functions among which is file definition. They are then led through a series of steps to define file names, record layouts, and field names and characteristics. The resulting file definition is stored and used to prompt the user upon data entry. Only two of the systems allow modifications to be made to this file definition after its initial creation without file reconstruction.

Data Entry and Update - All of the packages provide interactive data entry and update facilities. Like file definition, this is usually done via a series of menu screens and prompts. Most often the system prompts the user field-by-field which can be time consuming if a large amount of data is to be entered at one session especially at original file loading. Some of the packages require the file to be sorted or some utility to be run immediately after creation in order to create the index or pointer list for keyed access. A few packages

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

check the input for validity (e.g. no alphas allowed in numeric fields). Record updates for the most part have to be performed by manually accessing each record unless the user writes a separate batch program. The record is accessed by its relative record number or primary key, the new value placed in the appropriate field and the record stored. A few packages provide the facility to update all records in a class, or which meet some criteria. This greatly enhances system performance for across-the-board changes. Deletes are usually handled in the same way as updates. The packages are about evenly split as to whether the space occupied by a deleted record is immediately reusable. A few systems allow for restoration of deleted records if file compaction has not been performed.

Information Retrieval - All of the packages provide some record retrieval facility but vary widely as to their capabilities. Most allow retrieval via relative record number, primary, and secondary keys. A number also allow Boolean operators and range values for key retrieval. A few provide the facility to supply only a partial key, or a key with imbedded wild-card characters. Two of the systems provide "instring" search capabilities to display records matching a particular string in any field. Some systems display the first record satisfying the supplied criteria but most display all the records which meet the criteria. Some systems also keep track of the current position in the file and allow access to the next or previous record.

Report Generation - All of the systems provide report generation facilities in the form of columnar reports using field names as headings. Some allow the user to supply headings other than field names, to generate totals and subtotals based on a change in a field value (level breaks), and to perform some statistical analysis. None of the systems provide graphic displays but most have mailing label print capability. Most systems allow report formats to

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

be stored for repeated use. Most allow the report output to be directed to the terminal or the printer at the user's discretion.

Special Features - Many of the systems surveyed had various special features. A machine language sort seemed to be the most common extra. Some of the sorts were quite sophisticated with record include/exclude and multi-level sort criteria. Some also allow Boolean operators in the sort criteria. In addition the systems surveyed provided variously: date routines, data compaction routines, file reorganization utilities, and space allocation reporting. Only one of the systems surveyed provided any data dictionary support. A few provide password protection.

Performance - Access times to individual records and throughput in terms of number of record updates per minute for batch processing are dependent, among other things, on the type of auxilliary storage device used. Depending on the access method employed performance can degrade seriously with large file sizes and long search keys. Access times in the range from .5 to 2.5 seconds per record were reported for file sizes less than 1000 records.

Support - Generally these systems can be installed in less than a day and used successfully within two. A really complete understanding of all the facilities would probably be achieved with a month's use of the package. All of the systems are sold with some documentation. Most manuals lead the user through the functions in a step-by-step manner. There are a number of user groups for each of the hardware manufacturers which will provide help to new users. Usually the distributor of a package will provide users with some initial help in setting up files but few provide any on-site support. Most vendors provide upgrade notification for their customers.

Summary - The file management systems currently available provide the small businessman with little knowledge of computing techniques with a fast

easy method to store and retrieve data. They are aimed at the user who does one record lookup or update at a time with minimal amounts of calculation or statistical analysis. They provide such a user with simple reports with a minimum amount of work.

As a business grows its data processing needs change. New information must be carried in files, transaction volumes increase, and more complex data manipulation is required. The systems currently available do not provide adequately for this type of growth. Usually files can not be modified after implementation and complex applications must be user-written.

4.4 Application Development Systems

These systems are designed to help users with a small amount of computer experience and intimate knowledge of their processing requirements. Users define the sequence of file processing steps necessary for their application by answering prompts or using a special command language provided by the package. The result is a customized stand-alone system for performing tasks such as inventory control, accounts payable or receivable, sales analysis or general ledger.

Four of the systems in Table 1 have been designated as Application Development Systems because they provide facilities to assist in updating master files from transaction files. All four of the systems provide special interfaces (prompts, menus or screen formats) for file definition, data entry, update, screen formatting, information retrieval and report generation. Three of the systems allow the master file update function to be performed using user-defined procedures.

PRISM/ADS (an extension of the PRISM/IMS file management system) is essentially a single file information retrieval system when used in a stand-alone fashion. When called from a user-written BASIC host

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

program PRISM/ADS provides screen format, master file management and report generation facilities. Transaction files (generated external to PRISM) can be processed by the application program and matched with master file records using PRISM. FMS-80 and SELECTOR IV allow multi-file retrieval and processing. Master file updates can be performed by user-defined procedures using computational and conditional statements in languages provide by the packages.

The fourth system, CBS, allows users to define their processing requirements non-procedurally by specifying one or more menus. A menu item may specify that the steps in another menu be performed. Each job step in a multi-file application may be defined in terms of the input and output files used and the CBS sub-program that is to be performed. In this way it is easy to define simple update operations involving addition to or subtraction from a master file data item value or movement of data to or from the transaction file. Up to eight job steps and twenty files may be involved in a single application.

Systems like these make it easier to customize business applications and may be used increasingly in the future by both software developers and end users. Unlike the DBMS systems to be described in the next section they concentrate on processing logic rather than on the logical relationships among data items. Careful design and management of the transaction and master files is necessary to avoid proliferation of redundant data and difficult system maintenance problems.

4.5 DATA BASE MANAGEMENT SYSTEMS

DBMS systems can be classified according to the data model (logical view of data) on which they are based. There are three principle models; hierarchical, network and relational. Two of the systems surveyed (MDBS and Microseed) are network database systems generally conforming to the design

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

principles described in the CODASYL Data Base Task Group report, [2]. CONDOR Series 20/DBMS and CONDOR DBM-1 are based on a relational design, [3]. MDBS has a simpler version, HDBS, which implements an hierarchical data view.

A data base management system provides a means for many different application programs to share the same data files. This greatly reduces redundancy of data storage and the resulting update problems. Multiple applications can share common data because the DBMS allows each program to access any data it needs in any required order.

By way of contrast traditional data processing without a DBMS will relate a program to its data by maintaining multiple files and using sort and match-merge operations. Note that this can provide an efficient method of processing applications with large volumes of transactions. However, a program must be written and the requisite files created for each application. This usually means that multiple copies of the same data item must be maintained. It also leads to inflexible systems since new programs and files need to be set-up each time a new information request has to be satisfied.

Simple hierarchical relationships (eg. one employee to many dependents) can often be handled efficiently in traditional file processing by specifying records with a root segment (eg. employee) and repeating groups (eg. dependents). Hierarchical retrieval can also be achieved by a user-maintained coding scheme and the use of partial keys or wild-card characters in some of the file management systems discussed earlier. However, without a DBMS it is difficult to represent either (1) many-to-many relationships (eg. suppliers supply many different parts; a given part may be supplied by many suppliers) or (2) relationships where a given record has more than one owner (eg. a line item record needs to be related to a Purchase Order header record, a supplier record and an item record).

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

The Two Network Systems

Before a network DBMS can be used it is necessary to define a data base 'schema' describing the data items, record types and interrecord relationships or 'sets' that are to be maintained in the database. MDBS utilizes a special editor routine via which the user defines the data base in a non-procedural 'data definition language' (DDL). In MICROSEED this statement is written to a file using the normal system editor. A graphical representation of the information stored in a typical schema is shown in Figure 4 below.

MICROSEED follows the CODASYL 'subschema' concept and requires that a separate file be generated for each application program describing the subset of the schema that will be utilized. The subschemas are defined using the same DDL as the schema. Subschemas can be useful in providing data independence and provide a form of security protection since an application can only access those parts of the schema that have been declared in its sub-schema. MDBS by-passes this step but incorporates a 'data-group' concept into the host language program interface with the DBMS. This allows the user to specify a subset of data items which can be accessed using a group name.

Having defined the structure of the data base in this manner the next step is to load the data into the database. To do this the user must write a program in a host-language such as FORTRAN, BASIC or COBOL. The host language program reads the user's data files and stores the data in the database file by issuing 'calls' to the DBMS. This 'extension to the host language' is called a Data Manipulation Language' (DML).

In general acess to the data for applications can be obtained either via a high-level 'query language' or by writing programs in the host language (which accesses the data via DML calls as described above). The applications to be described later have used the host-language approach.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

Initial entry to the network of records is obtained via a special 'System" record in MDBS; thereafter pointer chains are followed to move from one record type (eg. Organization) to another (eg. Employee). MICROSEED allows initial entry to an instance of a record in the network via a DBMS-supplied hashing function, thereafter pointer chains are followed in a record-by-record fashion as above.

Both MDBS and MICROSEED incorporate significant extensions to the original CODASYL specification. MICROSEED incorporates a query language as part of the basic package. MDBS allows a direct representation of many-to-many relationships. This can significantly decrease both the complexity of the data base and subsequent processing. MDBS also allows set relationships to be defined over a single record-type (rather than between two record-types). For example, this allows direct representation of bill-of materials relationships between instances of 'parts' records. MDBS supplies an interactive query language as a separate add-on feature. In addition MDBS also supplies a separate 'data restructuring' package which allows item, record and set relationships as defined in an existing schema to be changed under program control. This feature can be very significant in allowing users to adapt their systems to changing requirements.

The Relational System

The two Condor systems are based on a relational view of data. In this view all data items are stored in 'flat' files or 'relations' ie. there are no repeating groups or other hierarchical relationships involved. To give maximum flexibility and simplicity to the user view, logical data base relationships between records are not explicity defined by pointers or links as in the hierarchical or network models. Instead relationships inherent in the data are realized by referencing keys that are common to more than one relation.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

The data definition interfaces to DBM-1 and Series 20 utilize screen formats and are similar to those for many other systems in this study. Multiple 'views' of each relation are possible; each view may contain a different description of the data items.

Users can access the data base using a high level language based on an algebraic approach. This language permits users to form new relations for subsequent temporary or permanent storage or for use in reports or in answer to queries. The new relations are subsets of existing relations or are obtained either by appending one relation to another or by merging two relations on a common key field.

DBM-1 and Series 20 provide password protection and have report generators. They can be used in either interactive or batch mode and can also be called from host-language programs. In common with the ADS systems described earlier it is possible to define job streams to perform traditional transaction processing.

5. SURVEY OF DATA BASE MANAGEMENT SYSTEM USERS

We now discuss the results obtained from a survey of users of one of the two network DBMS systems (MDBS) and the relational systems (Condor DBM-1 and Series 20). Although the sample size is small we feel that the results are interesting and indicate the potential for this new area of micro computer software.

Table 3 summarizes some characteristics of the user sample. Most of the users were computer professionals. However, one non-professional user with only one year's programming experience was successfully using the system. All

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

of the users in the sample had received assistance from the vendor either in installing the system or in developing applications.

Sample Size = 10

Industry: Government (3), Professional (2), Software industry (4), Microcomputer manufacturer (1)

Average Data Processing experience of persons using DBMS = 4 years

Hardware: CPU capacity: 64K(9); 56K(1)
Floppy disk (6); hard disk (4)

Assistance from Vendor: All 10 users had received assistance from the vendor either during or subsequent to installation.

Applications: The 10 users cited 28 existing applications. All were planning new applications.

User Satisfaction: All systems were rated as "very satisfactory" by all users.

TABLE 3
Profile of DBMS Users

Table 4 contains some statistics describing the DBMS application programs and data bases. Since the DBMS occupies approximately 19k of memory (including page buffers) and space must be allowed for application program buffers and the operating system, the application programs must necessarily be rather small. However the functions performed by the DBMS itself greatly reduce the necessary length of programs when compared with a non-data base environment.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

Host Language Application Programs

Language	No. of Users	No. of Lines of Code	
		Minimum	Maximum
BASIC	5	250	500
COBOL	1	200	450

Data Base Sizes:

Minimum	Maximum
50K bytes, 400 records	3 M bytes, 10,000 records

Complexity of Data Base

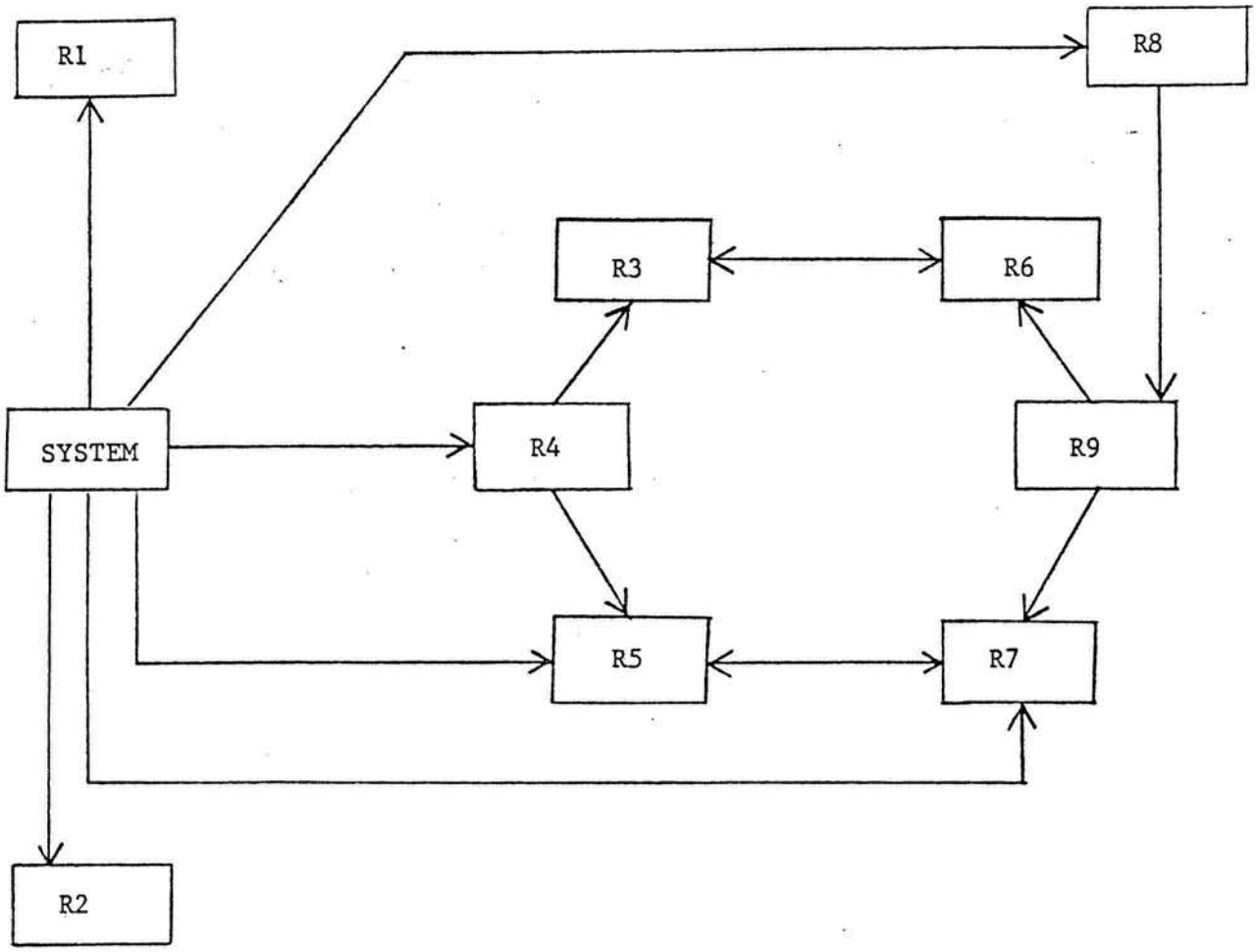
	Minimum	Maximum
Number of record types	2	13
Total number of fields	5	150
Number of 1: n sets	2	17
Number of m: n sets	0	4

TABLE 4

Characteristics of Existing DBMS Applications (MDBS)

The lower part of Table 4 gives an idea of the logical complexity of the data bases in terms of the intra-record and inter-record relationships maintained. Figure 3 gives a graphical representation of the schema for one of the user applications. In the figure record types are denoted by the rectangles R1, R2,... and set-types by the directed arcs. For example, the arrow from R9 to R10 represents a 1: many relationship between R9 and R10 (an instance of record type R9 may own zero, one or more instances of record type R10). The double-headed arrow between R7 and R10 represents a many-to-many relationship between these two record types.

The range of data base sizes for the relational systems was similar to that for MDBS. The number of relations in a data base varied from a minimum of 2 to a maximum of 20 while the number of attributes per relation varied from a minimum of 5 to a maximum of 77.



DATABASE SIZE		200,000 bytes
Number Of Record Types		10
Total Number Of Fields		96
Number of Sets:	1:n	11
	m:n	2

FIGURE 3

SCHEMA FOR ONE OF THE USER APPLICATIONS

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

From Table 4 and Figure 3 it can be seen that the users were taking advantage of the ability of the DBMS to maintain quite complex data relationships. In fact a number of the users indicated that they could not attempt such applications without a DBMS. The survey also indicated that some of the users were attempting to develop much more complex applications. One MDBS user planned a data base with 60 record types and 70 sets. Another was developing an integrated data base application with 48 sub-programs each requiring more than 40 K bytes of memory.

6. CONCLUSION

The data management systems surveyed cover a broad spectrum from self-contained FMS's to general purpose DBMS's. Some of the FMS's are particularly easy to use; however they have limited capabilities. At the other extreme the network DBMS's require programming skills but provide great generality of function. These relationships are plotted in Figure 4.

A trend already apparent in the industry is an attempt to move towards the ideal system shown in the upper right-hand corner of the figure. This is constrained by the capabilities of present micro-computer hardware, by our limited knowledge of what the end user really requires, and by our ability to produce reliable, efficient software.

This movement is occuring incrementally from both ends of the spectrum. File management systems are attaining a capability to process multiple files and to assist the user in developing applications. Data base management systems are being augmented by routines to help the user load data into the data base, by query languages and by report generators.

Competition is forcing improvement in the quality of micro-computer data management software. However the rule of caveat emptor still applies. Some systems will continue to have programming errors; documentation is not

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

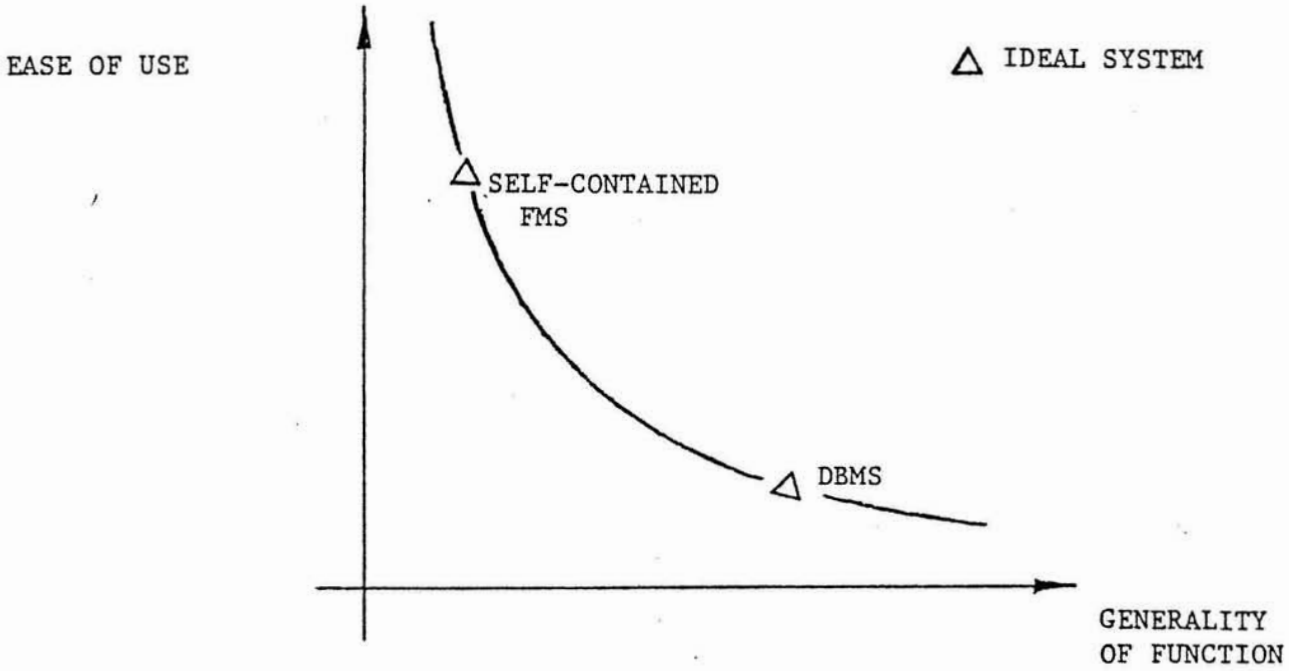


FIGURE 4

TRADE-OFF BETWEEN EASE OF USE AND GENERALITY OF FUNCTION

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

uniformly good; vendor support is necessarily less than that provided in the mini and maxi-computer markets; advertising claims and brochures sometimes use confusing terms. Usually the software will perform the intended business function if used properly. However supporting functions such as error reports, file maintenance and audit trails are often lacking. This forces users to learn details of the operating system and to employ data processing disciplines and controls for which they are unprepared.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

References

1. ACM, Special Issue: Data-Base Management Systems, Computing Surveys, Vol. 8, No. 1, March 1976.

2. CODASYL Data Base Task Group. April 1971 Report. ACM, New York.

3. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks", Comm. ACM 13, 6 June 1970, pp 337-387.

4. Postley, J.A., "The MARK IV System", Datamation, January 1968, pp 28-30.

5. Slonin, Jacob, Michael W. Farrell and Paul S. Fisher, "A Survey of Mini-Data Base Management Systems in 1977", Proceedings First SIGMINI Symposium on Small Systems, New York, 1978, pp 26-34.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

APPENDIX I

LIST OF SURVEYED SYSTEMS AND VENDORS

ANALYST
Structured Systems
Group, Inc.
5208 Claremont Ave.
Oakland, CA 94618

CBS
Dynamic Microprocessor
Associates
545 Fifth Avenue
New York, NY 10017

CCA DATA MANAGER
Personal Software Inc.
592 Weddell Drive
Sunnyvale, CA 94086

CONDOR SERIES 20/DBMS
and CONDOR DBM-I
Condor Computer Corp.
3989 Research Park Dr.
Ann Arbor, MI 48104

CROMEMCO DBMS
Cromemco Inc.
280 Bernardo Ave.
Mountain View, CA 94043

DBMASTER
Stoneware Microcomputer
Products
1930 4th Street
San Rafael, CA 94901

FMS-80
Systems Plus
1921 Rock Street
Mountain View, CA 94043

INFO MANAGER
North Star Computers, Inc.
1440 4th Street
Berkeley, CA 94710

MAXI MICRO MANAGER
Adventure International
Box 3435
Longwood, FL 32750

MDBS
Micro Data Base Systems Inc.
Box 248
Lafayette, IN 47902

MICRO B+
Fair Com
2606 Johnson Drive
Columbia, MO 65201

MICRO-SEED
International Data Base
Systems Inc.
2300 Walnut Street
Philadelphia, PA 19103

OS-DMS "NUCLEUS"
Ohio Scientific Inc.
1333 S. chillicothe Rd.
Aurora, OH 44202

OZZ
Commodore Business System Inc.
950 Rittenhouse Road
Norristown, PA 19406

PRISM
Micro Applications Group
7300 Caldus Avenue
Van Nuys, CA 91406

PROFILE II
Tandy Corporation
Fort Worth, TX 76102

SELECTOR III and SELECTOR IV
Micro-Ap Inc.
9807 Danovan Drive
San Ramon, CA 94583

SUPERKRAM
United Software of America
750 Third Avenue
New York, NY 10017

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

APPENDIX II

Survey Design

As mentioned previously an attempt was made to examine both (1) the available software and (2) the users and the nature of their applications. The contents of the survey questionnaires for these two groups are now briefly summarized.

Vendor Questionnaire

The questionnaire addressed to the vendors was technical in nature. However it stressed the features provided from the user's point-of-view rather than from the view point of the software system specialist. It contained questions suitable for both FMS and DBMS systems under the following general headings:

1. General Information

   Vendor; price; type of system; date of release; number of users.

2. System Requirements

   Micro processor type; CPU requirements and capacity; operating systems; language interfaces.

3. Physical Data Structures

   Data types supported; system limits on size of fields, number of records, etc.; access methods.

4. Logical Data Structures

   Record structure; inter record relationships.

5. Degree of Program and Data Independence

   Separation of logical and physical file descriptions.

6. User Definition of Records, Field Names and Formats

   Editor, prompting, screen format facilities.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

7. Data Entry and Update

Special data entry program; screen formats; single or group updates and deletes; space recovery.

8. Information Retrieval Facilities

Query language; prompts; screen formats; selection criteria possible.

9. Report Generation Facilities

Screen format generator; columnar reports; graphs; mailing labels; special forms.

10. Utility Routines Provided

Sort routines, master file/transaction file update; data base usage statistics; calendar and date routines; data compaction; available space reports.

11. Security and Integrity of Data Base

Pass-word protection; update validation; recovery and back-up features.

12. Ease of Use and Vendor Support

Required sophistication of user; documentation; consulation, training.

13. Performance Statistics

Time per retrieval/update; tuning.

In addition to obtaining results from the vendor surveys we also arranged live demonstrations of most of the packages shown in Table 1.

User Questionnaire

The objectives of this part of the study were to obtain information about the users, their reactions to the packages, the features they found most helpful and would most like to have, and information about the applications to which the packages were devoted. Slightly different questionnaires were sent to the users of data base management systems. An additional objective here was to ascertain the complexity of the data base structures used.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69

1. User Profile

   Industry; size of firm; data processing environment; personnel; intensity of use of package.

2. Hardware Configuration

   CPU; operating system; auxilliary storage; hours of use of micro-computer.

3. File Management or Data Base Management Package

   Name; vendor; version; hours of use of package; number of applications.

4. User Rating of Package

   Satisfaction with package; documentation; problems in installation; quality of assistance provide.

5. General Comments

   Major problems; best features; new features desired.

6. Applications for which Package Is Used

   Short descriptions of present and future applications.

7. Details of Individual Applications

   Retrieval only; update master file; computations; graphics; number of files used; size of files; intensity of use of application.

8. Complexity of Data Base

   Number of record types and relationships

9. Application Programs Using Data Management Systems

   Source language; size.

Because of space limitations we can only give an overview of the results of these two surveys. In fact, the results of the user survey are further restricted to a small subset of DBMS users.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-81-69