"A Veritable Bucket of Facts" Origins of the Data Base Management System

Thomas Haigh

University of Wisconsin--Milwaukee
Bolton Hall, Room 582, P.O. Box 413, Milwaukee, WI 53201-0413
thaigh@computer.org

ABSTRACT

The data base concept derives from early military on-line systems, and was not originally associated with the specific technologies of modern data base management systems. While the idea of an integrated data base, or "bucket of facts," spread into corporate data processing and management circles during the early 1960s, it was seldom realized in practice. File-processing packages were among the very first distributed as supported products, but only in the late 1960s were they first called "data base management systems," in large part through the actions of the Data Base Task Group of the Committee on Data Systems Languages (CODASYL). As the DBMS concept spread, the data base itself was effectively redefined as the informational content of a packaged DBMS. Throughout the process, managerial descriptions of the data base as a flexible and integrated repository for all corporate data stood in sharp contrast with the useful but limited nature of actual systems.[1]

1. INTRODUCTION

The Data Base Management System (DBMS) is the foundation of almost every modern business information system. Virtually every administrative process in business, science or government relies on a data base. The rise of the Internet has only accelerated this trend – today a flurry of database transactions powers each content update of a major website, literature search, or internet shopping trip. Yet very little research addresses the history of this vital technology, or that of the ideas behind it. We know little about its technical evolution, and still less about how its usage has changed over time. [2]

A data base management system is a very complex piece of system software. A single DBMS can manage multiple data bases, each one usually consisting of many different tables full of data. The DBMS includes mechanisms for application programs to store, retrieve and modify this data and also allows people to query it interactively to answer specific questions. Specialists, known as Data Base Administrators (DBAs) control the operation of the DBMS and are responsible for the creation of new data bases and the definition of the table structures used to store data. One of the most important features of the DBMS is its ability to shield the people and programs using the data from the details of its physical storage. Because all access to stored data is mediated through the DBMS, a data base can be restructured or moved to a different computer without disrupting the programs written to use it. The DBMS polices access to the stored data, giving access only to tables and records for which a given user has been authorized.

Today, corporate computer staff would usually conceive of a data base as the content of a data base management system. (In fact, the two concepts are so closely associated that DBMSs such as Oracle are often simply called data bases, even by IT specialists). Historically, though, the two ideas were distinct. The data base concept originated around 1960, approximately ten years before the idea of a DBMS gained general currency. The data base concept originated among the well-funded cold war technologists of the military command and control, and so was associated with the enormously complex and expensive technologies of on-line, real-time, interactive computer applications. By the mid-1960s it had entered managerial discourse, and was used to describe the huge pools of shared data needed to construct a "totally integrated management information system" (MIS) to integrate every aspect of the management of a large corporation.

On a technical level, however, the DBMS evolved from a more humble class of programs known as "file management systems", created within the unglamorous world of corporate data processing to simplify the creation of programs for routine administration. The data base management system conflated the managerial concept of the data base with the specific technology of the file management system. As this paper shows, in practice the DBMS worked well as a technical system to aid application programmers, but disappointed as a managerial panacea. Most early DBMS systems were used primarily for routine applications, were not queried directly by managers, and did not support the integration of all corporate data. In addition, while the corporate data base had originally been conceived as a repository of all important managerial information, actual DBMS technology supported only the kind of highly structured regular records with which earlier file management systems had been adept.

The story of the DBMS therefore provides an interesting example of the process by which particular technologies with very specific

---

[1] This is a revised version of an article originally published in W. Boyd Rayward and Mary Ellen Bowden, eds., The History and Heritage of Scientific and Technological Information Systems: Proceedings of the 2002 Conference (Medford, New Jersey: Published for the American Society for Information Science and Technology and the Chemical Heritage Foundation by Information Today, Inc., 2004.) http://www.chemheritage.org/events/asist2002/proceedings.html

[2] While many data base textbooks include a few pages on the development of data base theory along with their introductory definitions – for example [41] does this well – this can mean little when stripped of its historical context. The closest thing to a detailed history is a quarter-century old technical primer [44, pages 19-29]. A short history, focusing on the role of public funding in the emergence of the relational model, is found in [77, ch. 6]. On the technical side, detailed comparisons of early systems are given in [19, 31, 101, 111].

qualities and distinctive strengths and weaknesses are promoted instead as universal solutions. The same pattern has been seen many times: in early discussion of information retrieval as a problem that could be solved for the general case, with the christening of computers as information technology, and with more recent attempts to sell systems for “data warehousing”, “data mining” or “knowledge management” as universally applicable technical solutions to organizational needs. In all these cases, acceptance of the idea of information as a generalized quantity that can be stored in and processed by machines serves to elide the difference between very broad human or managerial concepts of information and the far more constrained capabilities of specific automated systems.

2. THE DATA BASE AND THE MANAGEMENT INFORMATION SYSTEM

During the 1970s, when data base management systems were first promoted to corporate managers, they were sold as the technological means by which all of a company’s computerized information could be assimilated into a single integrated pool of data. This idea was not, however, a new one. Indeed, its widespread discussion among experts on the managerial applications of computers dates back to the late 1950s, several years before the term “data base” was used in this context. To understand the initial concept of the data base, and its appeal, we must therefore begin by examining the concept of the Management Information System (MIS).

In March 1960, a senior representative of Arthur D. Little, then the largest and longest established management consulting firm, addressed his colleagues at a conference organized by the American Management Association to discuss new applications of computer technology to the problems of corporate administration [102]. Milton D. Stone was, like many of his fellow speakers, enthusing about the incredible potential of the Management Information System, then a very new and very exciting concept [48]. MIS, a concept unveiled to the managerial public for the first time only a year later, was already well on the way to becoming the single most widely discussed concept in the corporate computing world of the 1960s – promoted relentlessly by consultants, “systems men” (corporate staff specialists in administrative management), computer experts and computer manufacturers. Its advocates suggested that the best use of the computer, the only one to truly exploit its potential, was to build an enormous automated system capable of providing to each and every manager in an entire corporation every last piece of information necessary for the performance of their duties, in a timely fashion. It would reach, as Stone put it, “from board chairman to straw boss”, and include sophisticated modeling and forecasting capabilities as well as simple factual reporting. [102, page 17].

Data processing was already well entrenched as the dominant name for administrative computing [47], but MIS enthusiasts suggested that this conservative and evolutionary approach wasted the power of the computer on mere clerical automation. MIS was intended to remove these expensive and unfamiliar machines from the too-pedantic hands of the accountant (who held “prejudices born of a lifetime of education and practice in the world of fine-ruled yellow analysis pads”) and from former punched-card supervisor or “data processing technician”, dismissed by Stone as a drone who would follow whatever instructions were placed in front of him.

These early, rather vague, concepts of data pools embedded the assumption that all relevant information, whether internal or external, past or future, economic or human, could be accommodated within a single structure. The 1950s had seen a sudden proliferation of discussion about information within a number of different fields. Shannon’s mathematical theory of digital communication [98] was picked up as a powerful metaphor within the nascent meta-discipline of cybernetics. Librarians specializing in scientific and technical fields began to speak of themselves as information scientists [112], while researchers attempting to automate record searching started to call this work information retrieval [17, 76]. Glowing reports in *Fortune* magazine informed businessmen of the power of information theory [12] and of information retrieval [11]. In 1958, the combination of computers, operations research methods and simulation was first dubbed “information technology” [62]. Information was in the air, as a kind of universal solution to the various ills of business, science and government.

It was men such as Stone who first introduced managers to the idea of information as a generalized, abstract entity, separate from the forms, reports, files and memos in which it had previously been embodied. Stone recognized that a flexible and complete MIS could only be constructed if a firm’s entire mass of paperwork could be computerized and integrated “to produce an interrelated body of useful data, or information.” He suggested that “this body of data, a veritable ‘bucket of facts,’ [was] the source into which information seeking ladles of various sizes and shapes are thrust in different locations” [102, page 17]. Others, working with similar ideas, came up with other phrases over the next few years. Another consultant suggested that the office of the future would revolve around a “data hub”, defined as “a central source of information that can serve as an instant inquiry station for executives who need data for decisions.” [110] Representatives of Shell Oil spoke of the need for an “electronic data bank, or pool of information, from which reports of many types can be drawn.” [51, 60].[3]

These buckets, pools and hubs seem quaint and rather unhelpful metaphors today, and indeed those trying to construct them using the technology of the 1960s were doomed to disappointment. Rather than flowing smoothly and easily into an ocean of knowledge, information instead coagulated messily around the small memories, tape drives, and inflexible file structures of early mainframes. Yet, if we can step back for a moment from the familiarity of the phase “data base”, unknown in data processing circles as Stone spoke, is not a base of data even stranger, even more metaphorical, than a pool, bucket, hub or bank? These metaphors all serve to construct a particular version of information, in which the richness of social meaning that structures and supports information in its more specific manifestations (a parts list, a sales forecast, a letter of complaint) has been stripped away, leaving behind an inert substance that can

[3] It is worth pointing out in this context that Edgar F. Codd, creator of the relational data base model, informed the world of his invention in a paper entitled “A Relational Model for Large Shared *Databanks*” [33]. Even in 1970 the term was far from dead.

be stored, refined or piped as necessary. It implied that a single kind of technology or expertise, and therefore a single group of skilled professionals, could process information of any kind.

By the late 1960s, however, “data base” was a common expression in corporate computing circles, largely replacing the hubs, buckets and pools in which data had previously been rhetorically housed. This term was imported from the world of military command and control systems. It originated in or before 1960, probably as part of the famous SAGE anti-aircraft command and control network. SAGE [40] [56] was far more complex than any other computer project of the 1950s, and was the first major system to run in “real-time” – responding immediately to requests from its users and to reports from its sensors. As a result, SAGE had to present an up-to-date and consistent representation of the various bombers, fighters and bases to all its users. The System Development Corporation [10], a RAND Corporation group spun-off to develop the software for SAGE, had adopted the term “data base” to describe the shared collection of data on which all these views were based.

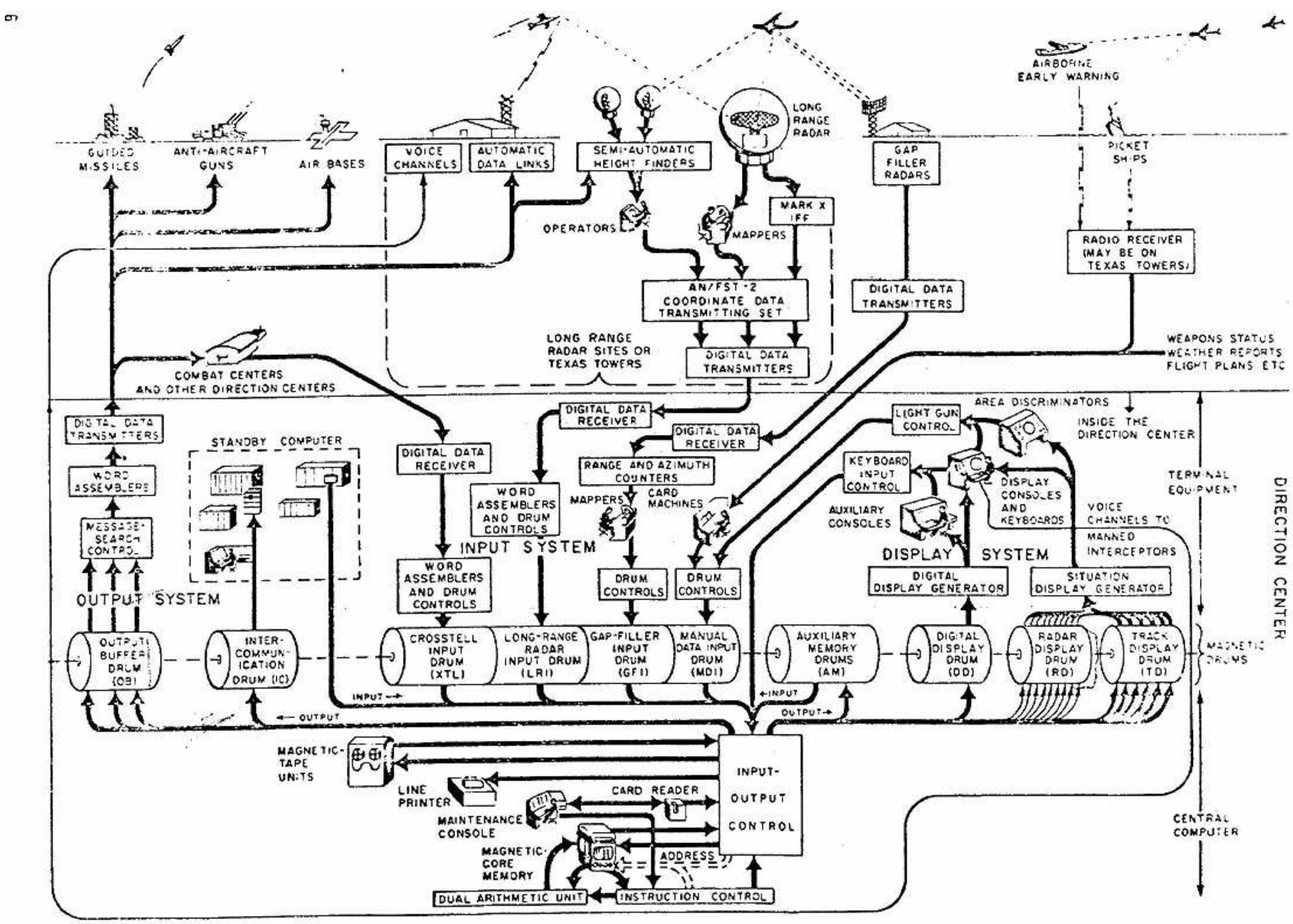

Figure 1: The SAGE system integrated data from many different sources (shown here as the “input system”) and provided selective views of the overall situation via video consoles (shown here as the “display system”). Information was consolidated in the “central computer” (not shown). Developers of SAGE software may have been the first to use the term “data base” to describe a centralized body of data shared between many different subsystems. The image is taken from [58].

SDC actively promoted the data base concept for military and business use. Its interest in general purpose data base systems was part of its attempt to find new markets for its unique expertise in the creation of large, interactive systems. During the late 1950s and early 1960s, SDC held by far the world’s largest concentration of programmers with experience in large-scale, real-time systems [94]. It paid particular attention to the fashionable area of “time-sharing” computer systems, in which one computer was used interactively by several people, each free to run whatever programs they required. Because computers were then large and expensive, time-sharing promised to make general-purpose, interactive computer use by non-specialists a commercial reality for the first time. SDC invested heavily in this area [4], and identified “computer-centered data base systems” as a key application of time-shared systems – hosting (in collaboration with military agencies) two symposia on the topic in 1964 and 1965. [104].

The SDC Data Base Symposia were crucial in spreading the data base concept beyond the world of real-time military contractors.

The approximately 185 participants at the second symposium included high-ranking military officials, business data processing celebrities, and corporate and academic researchers. Reporting on the event in *Datamation*, the leading trade magazine of business computing, Robert V. Head observed that data bases had already unleashed the "biggest single strike" of new jargon "since the great time-sharing goldrush of 1963," leaving potential users "sullen and down-trodden." He concluded by wondering whether it was "possible that users, led by the military, will surrender to these data base systems without a shot being fired in anger." [52, page 41]

It was around this time that the "data base" term made its first appearances in the ongoing discussion of management information systems. In 1965, Harvard accounting professor John Dearden was using the term "data base" to describe the truly important set of corporate facts and figures that had to be shared between different areas within a business [39]. Within the more technical literature it appeared as a means of pooling information from different files, so that each piece of data would be stored only once. Its great advantage would be "to permit categories of information to be added, deleted, expanded and otherwise revised, without completely redesigning the file or reprogramming the retrieval routines" [99, page 4].

The idea of the data base as a physical pool of data underlying an MIS was given an early, clear and highly influential statement by Head, who defined the data base as the bottom level of a pyramidal structure [53]. The data base pooled information from all the company's operational systems, and on top of it were erected reporting systems and models to inform higher level managers. [48, 45-50]. The metaphor fit very nicely with the idea of a data *base* supporting the rest of the information system. This obviated the need for systems experts to determine in advance exactly what information each manager would require. Instead managers could interrogate the data base and receive whatever information they needed. The data base was often called a "reservoir" of information [54, 61, 113, page 30].

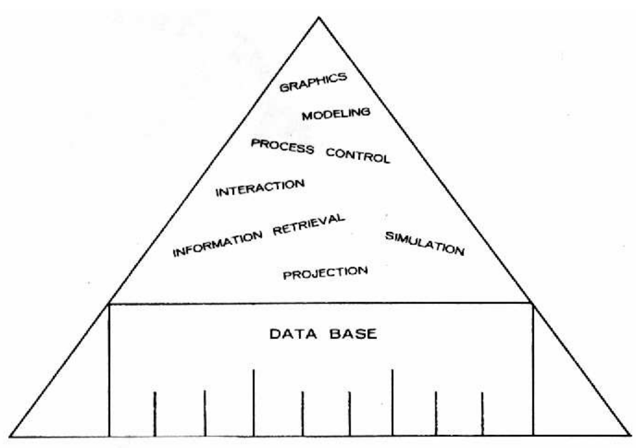

Figure 2: Head's concept of the data base as the support for other components of the management information system [53] was highly influential.

SDC's attempt to push the data base concept into civilian discourse worked well. The term data base carried some specific associations, based on the particular characteristics of firms like SDC and of military command and control projects. One of these associations was with the idea of real-time operation – the data base would be constantly and, if possible, automatically updated with current information gathered from a number of different sources. It was also assumed that, as in SAGE, a data base could be "interrogated" in real-time by its users, answering questions interactively within seconds. In addition, the data base would be shared among many different programs, each one using only a subset of the overall information contained within it.

In contrast, SDC's attempts to sell its own technology as a means of realizing this goal were not nearly as successful. SDC had used its data base symposia to showcase its own on-line systems [16], funded with military money, all of which ran on the special, and hugely expensive, computers developed for SAGE. [104]. SDC's most ambitious attempt to commercialize data base technology came with a system called CDMS (the Commercial Data Management System), a derivative of an earlier system called TDMS (Time-shared Data Management System) developed under contract from ARPA (Advanced Research Projects Agency) and given trial usage at military installations. These systems were intended to allow non-programmers to create data base structures, load data into them and then issue queries and retrieve their results on-line. Attempts to sell the TDMS computer program failed because it was expensive, needed a powerful computer all to itself, and could run only on SDC's own custom-developed operating system. Attempts to rent use of CDMS through terminals connected to centralized computers were equally unsuccessful. [10, pages 116-121, 101, 107].

In the late 1960s, the much discussed administrative data base remained a dream without any clear technological avenue of fulfillment. These early attempts to provide managers with interactive, on-line access to data stored in computer files suffered from a number of problems. These included the enormously expensive nature of the technology, a lack of interest on the part of most managers, and the largely unaddressed problems of taking data from all the routine, operational systems (payroll, accounting, inventory, billing and so on) and somehow integrating it and making it available inside the data base.

3. FILE MANAGEMENT SYSTEMS AND DATA PROCESSING

Besides the rather ill-defined concept of the "data base" the other main intellectual ingredient of the Data Base Management System, and the key technological foundation for the actual data base management systems of the 1970s, was the "file management system" (together with its close relation, the "report generator"). File management systems were intended to reduce the cost of producing routine administrative programs, and to make the finished programs easier to change and maintain. Report generation systems made it easier to produce printed reports based on particular criteria. These ideas, unlike the data base concept itself, were indigenous to the world of administrative data processing, where they had slowly evolved. Whereas the data base reflected a focus "blue sky" technology, on-line operation, scientific genius and enormous expense, these file management systems were initially oriented toward clerical tasks, were used and appreciated primarily by programmers and data processing supervisors, lacked features for interactive or on-line use, and did not cost much. Rather than glamorous managerial systems, they were humble but highly effective tools for computer technicians.

The need for such tools became apparent during the mid-1950s, as soon as computers were first applied to administrative tasks. Pioneering computer users had soon discovered that apparently simple clerical data processing activities, of the kind looked-down upon by enthusiasts for MIS, were far from trivial in practice. The pioneers of the late 1950s and early 1960s developed many new techniques and approaches as they struggled to contain programming and operations costs while maximizing flexibility. The techniques used to store data on tape were taken from existing punched card methods. Indeed the concepts of records, files, fields, special codes to mark the beginning and end of files, and the "merging" information from one file to another (all still ubiquitous in computer systems today) all have their origins in punched card systems.[4]

The first generation of American data processing installations spent much more and took far longer than expected to get their machines up and running. From General Electric's famous 1954 use of a Univac computer to automate payroll processing [81], data processing managers were shocked by the complexity of programming work and the rigid requirements computer technology imposed on areas such as data entry and the handling of special cases. Like punched card machines before them, early computers generally worked on one record at a time. The tiny internal memories of early computers, coupled with the inflexible, serial nature of tape storage, meant that a single major job such as payroll might require dozens of programs to be run one after another, each reading and writing information from several tapes [47].

File management systems evolved from the reuse of subroutines written to handle input and output tasks within application programs. Early computer programs included all the instructions necessary to specify the minute details of reading and writing information from tape or disk, and were forced to check regularly whether a particular record had yet been retrieved [69, 178-204]. Skilled programmers spent much of their time crafting routines to read records from tapes and print lines on paper, dealing each time with the many errors, synchronization problems, tape jams and so on that could frustrate their task. Programming groups soon hit on the idea of producing a single set of well written and reusable subroutines to handle these chores. Standard code was modified slightly to fit the particular situation and then inserted into each application program. Technological change also played a part. Application programs were closely tied to particular hardware configurations– even changing the tape drive used for temporary storage required considerable editing work, while adapting a program to make efficient use of more memory or additional tape drives involved a fundamental rewrite. The problem was compounded as companies attempted to reap the benefits of automation by using the output of one major application as the input to another, for example by linking their production scheduling system to their inventory control system, their accounts receivable system and their billing system. As computer manufacturers began to build more powerful capabilities into their data processing hardware, including buffers and auxiliary processing units to smooth the flow of data, the programming required to read and write records on tape became more complex. As a result, computer manufacturers began to supply their customers with standard functions to optimize these tasks [9, pages 181-185]. This made it easier to create new programs, but did little to help with other problems.

Another problem was the difficulty in extracting information from the computer – while daily, weekly or monthly runs of different parts of a payroll system might each produce voluminous printed reports, the only way to obtain a special report was to write another program. If a manager needed to tabulate data in a different way, or to include only a subset of the original records in the calculations, he could either wait for a programmer to become available or wade through the printout tallying records manually. By the late 1950s, the more innovative data processing teams had begun to address this through the creation of "report generation" programs, into which programmers fed a descriptions of the output desired and of the organization of the data inside the relevant "master files" and were rewarded with the desired reports. The work of General Electric's team at the Hanford Nuclear Reservation [72, 73] on its IBM 702 (IBM's first large computer designed primarily for administrative use) was particularly important in the establishment of these techniques.

File management systems had their origin in the use of similar techniques to create and update data files, as well as retrieve information from them. The most important initial areas were generalized routines to sort data into a particular order (a very important operation, and one that tape-based computers were very bad at doing compared to earlier punched card machines) and perform other routine maintenance operations on files. Because one major application might contain dozens of small programs, each reading and writing certain files, it might otherwise take Herculean efforts on the part of the programming staff to do something as simple as adding an extra digit to the employee number. By separating generalized file manipulation code from standardized descriptions of the record format used in each file, these approaches began to make it easier for programmers to modify record formats without completely rewriting programs. Such routines were written by the programming teams working inside computer using companies. In the early days of computing, it was common for system or utility programs of this kind to be shared freely, most notably through the SHARE user group established for users of large IBM computers [1]. During the late-

[4] By the 1940s, most punched cards included 80 columns of data, each one of which coded a single number or letter. Information within each card was grouped into fields, each occupying a fixed width within each record card. Consider a factory using punched cards to process its payroll [67]. Some fields needed only one column – for example sex (M or F). Other fields, such as last name, might be assigned a dozen columns. Each record would be punched onto one, or in some cases several, of the cards in the deck. The complete deck representing all the factory workers was known as a file, by analogy with conventional paper records. Each record card within the file had to follow exactly the same layout of fields, and to process a particular job the machine operators had to rewire the control panel of each machine (such as sorter, collator, or tabulator) to reflect this specific field layout. (Many jobs involved "merging" information from several files – for example combining wage information from the "master file" of personnel cards with the attendance information punched onto a weekly punched card by an IBM time clock). The concepts of file, record, and field were transferred directly to tape storage – though the records were now laid out sequentially along the strip of magnetic tape. Additional codes were introduced to mark the beginning and end of files and provide checks against corrupted data.

1950s SHARE coordinated efforts to develop General Electric's report generation system into more powerful systems for the IBM 709 called 9PAC, and a related project for the IBM 704 called SURGE.[5]

File management systems also proved an important niche for the nascent independent software package industry. Mark IV – the most successful product of the early independent software industry – was a file management system descended from report software produced for the Douglas Aircraft Company [43, 88, 89].

4. RANDOM ACCESS STORAGE

These file management techniques were very useful with tape storage, but when firms began to start storing their data on disk drives, the extra complexity of programming random access data storage and retrieval made their use almost essential. The disk drive was first offered as a standard option for most major computer systems in 1962 [2, 100, 109], though it had been available in a handful of IBM systems a little earlier. Whereas tape had previously been the only way of magnetically storing reasonably large files of information, it was suddenly possible to hold up to one billion characters of data on the disk drives connected to a single large IBM computer.

Disk technology progressed rapidly, and by the mid-1960s disks were standard options on many of the newly announced "third generation systems", along with operating systems, large memories, remote terminals, and other features marketed as the key to on-line application development [91]. Disk storage promised to bring the MIS dream of a fully integrated system much closer to reality. A large disk system appeared to be the physical bucket into which facts could be placed, and from which they could be checked, retrieved and updated by many different application programs.

In tape storage, records were generally sorted into a particular order and placed one after another along the tape. This was a fundamental limitation, because, as with today's video tapes, it might be necessary to wind through the entire tape to reach a desired spot. Users would still need to keep paper files, or leaf though big piles of routine printout, to get speedy access to a specific record. Disk drives, however, offered "random access" storage, giving almost instant access to any part of a disk. This promised to allow the speedy retrieval of specific data as needed, making it much easier to create special reports or to build on-line business systems such as the celebrated SABRE airline reservation system [34, 83]. Random access promised almost instant record retrieval, but although it was easy to order the computer to read a particular part of a disk (such as drive 4, platter 5, side 1, track 3, sector 15), there was no easy way to jump straight to a particular record (e.g. customer account 15274). One could, of course, keep the records sorted in order, but this would require an enormous amount of work rearranging the existing records every time a new one was added. Programmers experimented with a variety of strategies to arrange and index data on random access devices [69]. No single technique was suitable for all situations, and most of them were very complicated to program.

Another set of problems was caused by having several programs share a single disk, each using different program code to read and write records. Among these problems were the risk that an errant program might scramble an area of the disk holding information belonging to another, the overhead imposed by writing several different versions of the code required to handle complex indexing techniques, and the certainty that at some point the physical layout of the disk storage would be change (for example to shift a growing file to its own disk and expand the storage areas for the remaining ones) and all the programs would have to be modified at once.

The most obvious way to deal with this enormous increase in complexity was to rely on a new breed of generalized file management systems built to work with random access discs [22, 24]. These systems were intended to speed program development, reduce maintenance costs, shield application programs from the consequences of changes in the physical disk layout and make it easier to selectively retrieve records based on their contents.

By the end of the 1960s, every major computer manufacturer offered at least one piece of advanced file management software. These were usually based on the expansion of systems originally produced for use within a single organization. The most innovative, and influential, of these systems was General Electric's Integrated Data Store (IDS), created by Charles W. Bachman. IDS began life circa 1963, as part of an effort known internally as "Integrated Systems Project II." Its goal was the production of an integrated system for production control, flexible enough to be easily customizable by GE's many departments but powerful enough to give rapid results to queries on production scheduling and inventory levels while automatically placing orders and calculating the optimum order quantities. The resulting system, MIACS (sometimes, but not always, Manufacturing Information And Control System) relied on IDS to handle its data storage and retrieval needs. The project was very much in keeping with the early 1960s push to create integrated MIS systems, and Bachman recalls that top management were told that the project name stood for ***Management*** Information And Control System [6].

Manufacturing involves the assembling of multiple components into larger parts, which themselves usually serve as components in one of more kinds of larger assemblage. The need to solve this "parts explosion" problem made it particularly important for IDS to support the creation of linkages between different kinds of

[5] According to Charles W. Bachman, SHARE's SURGE project began as an attempt to modify the Hanford Report Generator for use with the IBM 704 but eventually took a different direction. The 9PAC system for the IBM 709, however, was produced by a different SHARE committee and appears to have been widely used. Although 9PAC was used exclusively with tape storage, it did permit the creation of hierarchical relationships between records. Child records were simply stored on tape immediate following their parent records. Data definitions for the file were stored in a header. [6] The project included a Report Generator and a Generalized File Maintenance system in its creators aimed to incorporate capabilities for calculated updates and modifications to the format of existing files. Specifications for the latter are in [70]. Their shared file structure is described in [71]. The series of SHARE Secretarial Distributions sent to all IBM 709 sites include much discussion of the project, including a series of drafts of documentation, requests for information, comments and suggestions. The 9PAC Subcommittee of the SHARE Data Processing Committee was formed in late May or early June of 1959.

record. While earlier systems had supported the idea of sub records, stored sequentially and hierarchically within master records, IDS was much more flexible. This generalized concept of linkages between record types, known later as the "network data model" was a major influence on early DBMS implementations.[6]

IDS was designed from the beginning for use with disk drives. It took over an entire GE 225 computer, providing basic operating system functions including an early implementation of paged virtual memory to squeeze out maximum performance from the 8K standard memory of the 225. The task scheduler for the MIACS application itself relied on IDS to store data. MIACS application programs (written in General Electric's own GECOM language) used simple instructions to navigate through the relationships between records and to STORE, GET, MODIFY or DELETE records one at a time. In the first implementation of IDS, a preprocessor replaced these special instructions with the appropriate strings of assembly instructions. However, efficiency concerns forced a switch to a different approach, where IDS performed this expansion interpretatively, combining the requested operation with metadata about the record type involved. This part of IDS remained resident in memory, waiting to deal with data requests from the application programs [6].

A few years later, around 1965, the first version of what eventually became IBM's Information Management System (IMS) was produced by IBM in collaboration with North American Rockwell to handle the proliferation of parts involved in the Apollo program [13]. The original version of this application, known as Generalized Update Access Method, ran on an IBM 7010 computer, and used a specialized hierarchical file management system to store its data on disk. IBM and NAA also developed a system called RATS (Remote Access Terminal System) so that interactive application programs could be accessed via terminals. In 1966 work began on a new version created to run as an application under OS/360 on the new System 360 machines, and it was this version that IBM distributed to other customers from 1968 onward. Like IDS, IMS was used by application programmers, using packaged procedures to embed data handling capabilities in their code. The OS/360 version allowed one memory resident copy of IMS to simultaneously service the data needs of multiple application tasks [26, 84].

General Electric offered an improved version of IDS to users of its computers, and IBM did the same with IMS. During the 1960s computer vendors "bundled" their software with hardware, using it as a free promotional tool to entice users into buying computers.

5. THE DATA BASE MANAGEMENT SYSTEM AND THE DBTG

The technological innovation represented by systems such as IDS was paralleled by conceptual developments. Until about 1968, the concepts of data bases and file management systems remained largely distinct. The data base was used interactively on-line, could be used by non-specialists and was closely associated with the MIS and the idea of a single huge reservoir of corporate information. File management systems were used primarily by programmers, to reduce development and maintenance costs for routine data processing applications. The most advanced file management systems were beginning to add features to make it easier to pool information from multiple files, and efforts were underway to add on-line access [18].

Combining the *data base* and the file *management system* created the Data Base Management System. The DBMS idea was shaped and promoted through the work of a body called the Data Base Task Group (DBTG), an ad-hoc committee of the computer industry group CODASYL (Committee On Data SYstems Languages). CODASYL's focus was the creation of data processing standards, and it is best known for its work designing and maintaining the COBOL programming language used for most business application programming from the late 1960s to the early 1990s. The DBTG was chaired by William Olle of RCA (then a manufacturer of mainframe computers) and its members were drawn from computer vendors, universities, consulting companies and a few large companies making heavy use of computers in their own business operations. Charles Bachman, the creator of IDS, was an early member of the committee and promoted the ideas he developed for IDS as the basis for its work.

As its name suggests, the DBMS was intended to be a new kind of product, extending the capabilities of existing file management systems to support the kind of advanced, on-line, interactive capabilities and huge integrated data stores associated with the data base concept. This was, in many ways, the endpoint of a natural evolution. The DBTG was dominated by the same manufacturers who were adding features to their file management systems and had begun to promote them as supporting, or even being, Management Information Systems [108]. The purpose of the DBTG was to define the capabilities of these new systems, and to develop new standards for them. Its creation was prompted by the realization within CODASYL that COBOL, while doing a great deal to standardize data storage on tape systems and to separate record definitions from program logic, was entirely inadequate when faced with the challenge of random access, disk based storage [80]. On its formation in October 1965 the DBTG had originally been called the List Processing Task Force (its name was changed only in 1967).[7]

In 1969 the DBTG released its first major report on what it now called "Data Base Management Systems". Despite lobbying by firms such as General Electric to get their own systems adopted as

[6] IDS is often, and with some justification, called the first data base management system. On a technical level, it was years ahead of its time and pioneered many important techniques. Powerful as it was, however, the initial version of IDS lacked some of the features associated with later systems and formalized in the CODASYL definition of a DBMS. Record definitions were punched directly onto cards in a special format rather than being defined and modified via a data definition language. It did not provide an interface for ad-hoc querying, or support for online access, since it was created purely to support the MIACS application. It did not provide different views or subsets of the overall database to different users. Neither did it support multiple databases simultaneously. Just as importantly, nobody at the time called it a data base management system. That concept itself did not exist at the time.

[7] The phrase "data base management system" was used at least once before the renaming of the DBTG, to describe IBM's forthcoming Generalized Information System (GIS) [18].

the basis for a new standard, the group decided that no single existing system came close to providing the range of features required. Instead, the group surveyed the strengths and weakness of existing systems, and began the attempt to standardize useful characteristics. Work continued, in part because the task group's parent committee was unsatisfied with the original results. In April 1971, a second and definitive major report [30] was issued and officially endorsed by CODASYL.[8]

The work of the DBTG provided both a broad conceptual outline for the data base management system, and detailed draft specifications for two specific parts of the over-all system (a data definition language for defining the data base structure, and a separate data manipulation language for accessing the data from within COBOL). It also outlined a way of giving individual programs access to selective or simplified versions of the full data base.

This conceptual framework for the DBMS ultimately proved more influential than the DBTG's detailed proposals. When the CODASYL work is mentioned at all today, it is usually for the propagation of the network data model. Since IDS, a commercial product, used this model prior to the DBTG's establishment this would seem a rather limited contribution to history. In fact, the DBTG appears to have created, or at the very least to have publicly defined for the first time, the very idea of the data base management system as we know it today.

The DBTG provided a new vocabulary with which to discuss these problems, including the separation of the "data definition language" used to define data base structures from the "data manipulation language" used by application programmers to work with the data itself.[9] Its final contribution was to insist that a standard DBMS allow more complex linkages to be established between different files (or, as they were now to be called, record types) within the same data base. The DBMS was intended to make these relationships (or, as the DBTG called them, "sets") as explicit and enforceable as previous file management systems had made the specification of fields within an individual file. Because most of the logic to maintain these relationships had previously been hidden within individual programs, placing relationships inside the DBMS along with the data itself ensured that all application programs and user requests would have access to them. The DBTG also decided that while the hierarchical approach used by systems such as IMS was good for some things, it proved unduly restrictive when applied to others. It instead specified a "network" model to represent these relationships, allowing the creation of more complex relationships between different groups of records.

Though most of the characteristics that the DBTG specified for a DBMS had already been demonstrated by at least one file management or data base system, it insisted that future systems must provide all of them. A DBMS was expected to provide the efficient, batch-based access for programmers and networked record-linking features that existing systems such as IDS specialized in. However, it was also expected to allow non-programmers to use a simple, specially tailored interface to query and update the data base directly – the province of systems such as Mark IV. Likewise, the DBMS was expected to support interactive on-line usage and batch operation with equal felicity.[31].

The term Data Base Management System, almost unknown before its adoption by the DBTG, spread rapidly from 1971 onward. It was applied retroactively to some existing systems, and used to describe virtually every new file management system, regardless of its fidelity to the specific ideas of the DBTG. This accompanies a great deal of publicity, as a flood of textbooks, technical articles and managerially-oriented pieces expounded on the potential of the data base. Following a traumatic transition to third generation equipment, many large corporations were now running powerful computers with large disk drives and flexible, multi-tasking operating systems and beginning to experiment with on-line terminals for data access. Meanwhile, the newly established market for independently produced packaged software was dominated by system software, particularly file management and data base management systems [49]. A 1973 article in *Infosystems*, the leading managerially oriented data processing publication, assured its readers that data base systems were like the aeronautical efforts of the Wright brothers: although carefully planned early efforts had "never developed much lift when applied to the practical realities of processing large files that had

---

[8] The specific proposals were controversial at the time, and several CODASYL members opposed them (including mainframe supplies IBM, RCA and Burroughs). [50] The IBM user groups SHARE and GUIDE went so far as to produce a rival report of its own. [23]. CODASYL's standards for the DBMS languages were not as successful as its work on COBOL, in the sense that no complete implementation of the specification was ever produced. The extensions to COBOL were reworked by a new standing committee into "Journal of Development" form as an official standard, published as [28]. Work on these standards continued into the 1980s, first through a new committee set up within CODASYL, and later at ANSI. This included a FORTRAN DML, to complement the COBOL DML in the earlier reports. [29] But these efforts to standardize a data definition language (DDL) failed to set a marketplace standard, in part because of the unwillingness IBM to commit to the network concepts inherent in the CODASYL model while its own flagship IMS product retained a hierarchical approach [85]. However, most of the advanced systems then under development were influenced to a more or less profound extent by ideas in the CODASYL reports – for a good summary of the most advanced commercial systems of the mid-1970s see [42, 44]

[9] The DBTG standardized terms such as "record" and "set" and "data base" and added some new ones, including "schema" (which remains ubiquitous today) to describe the logical format of data within the data base, and "sub-schema". A sub-schema (similar to what would be called a view in today's relational systems) allowed different users and applications to see only a portion of the overall database, allowing selective access to records and potentially shielding the application from changes in the underlying schema – a property referred to as "data independence." The DBTG also separated the Data Manipulation Language (DML) used to add, delete, update and retrieve particular records from the Data Definition Language (DDL) used to define the logical structure of the data base itself. While the DDL was to be a new and universally applicable language, the DML took the form of a set of additions seamlessly integrated into an existing programming language.

to be stored, indexed and sorted with live data" they were now poised to rise majestically into the air [93].

One immediate and dramatic result of the debut of the DBMS concept was a new surge of interest in the data base as the foundation of a company-wide management information system. During the late 1960s, a spate of bad publicity based on reports of delays and disasters among MIS pioneers had begun to raise doubts as to its practicality. The DBMS concept appeared as a technical savior for the idea of MIS as a single, all-encompassing system used directly by managers of all levels, and it featured prominently in many articles and textbooks of the early and mid-1970s. It functioned almost as a synonym for MIS. Richard Nolan, a Harvard Business School professor, consultant and one of the most prominent writers on computers and management during the 1970s, used a 1973 Harvard Business Review article to define the data base, rather boldly, as "a single pool or bank" where "all computer-readable data" is stored. He predicted that the long-awaited use of computers by senior executives was finally at hand, "from the union of the data-base concept and the corporation-model concept..." [78, pages 101 and 105]. As he observed the next year, "if the term Data Base or DB is used to replace the term MIS, the titles of recent articles are remarkably similar to the titles of MIS articles of several years ago" [79, page 27]. Many had simply seized on data base as a new and more palatable name for this "total" MIS.

Like the concepts of management information systems and of information retrieval, the idea of a data base was the intellectual product of a social movement trying to construct a new sense of information, as something that could be processed, retrieved and created using new bodies of scientific techniques. Like these other information concepts, the idea of a data base functioned in part to define a new area of professional authority. Considerable tension is apparent in the early 1970s, between those who, whether by virtue of temperament, practical experience or technical orientation, saw the DBMS as a practical tool to improve programmer efficiency and those who took this more utopian view of the data base and made few mentions of its technological underpinnings.

As one of the more practically oriented textbooks on the subject [66, page 22] explained, "A much-publicized but impractical idea of a data base says that a corporation keeps all its processable items of data in a large reservoir in which a diversity of data users can go fishing." The same idea had been given an early statement by Michael Scott Morton, founder of a prominent MIT group researching the managerial applications of computer technology, who in 1971 suggested that "the 'integrated' or 'company-wide' data base [was] a misleading notion, and even if it could be achieved would be exorbitantly expensive" [45].

These quibbles did not stop hopeful accounts of the data base as a technological marvel which would finally centralize and control information of all kinds, turning it from an abstraction into a solid organizational power base. This dream was enshrined in a new figure, the Data Base Administrator. According to one of the earliest descriptions, the DBA must "at once be technically qualified, if not inventive… he must encourage the users to work with him willingly and yet he will be forced to rule against their pet projects; he must represent both management and the users simultaneously; he must be all things to all people at all times." The author admitted that this role did "not exist as a formally established function in today's business" but considered its emergence imminent [64, page 12]. Nolan was still bolder: he believed [79,39] that the DBA would be responsible for "data as a resource… much broader than just computer-readable data," once the "data resource function [had been] carved out of the general management function." A consultant [63, page 9] wrote that the DBA should be "something of a superstar."

Discussion of the DBA makes the rift between managerially-oriented utopians and programmer-oriented pragmatists particularly apparent. Schubert, who at B.F. Goodrich had overseen a remarkably ambitious in-house DBMS development project, noted simply [95, page 47] that "Data base administration is accomplished by one or more technical experts who are knowledgeable in data base design and creation, operation of the data base management system, and the use of one or more data manipulation languages. The data base administrator must also be capable of working well with systems analysts, programmers, and computer operations personnel." It seems likely that this reflected practice in those firms actually using the technology rather than just talking about it; certainly, by the time DBMS technology became ubiquitous in the 1980s the DBA was a technical specialist rather than an information executive.

The idea of the "data dictionary" was given considerable discussion in the early 1970s. This was a central registry of the information gathered and produced by different parts of the business. By standardizing different representations of the same information, and establishing clear rules about who was responsible for each piece, companies could eliminate duplication and lay the groundwork for greater integration. This was originally seen as a managerial, rather than a technical, tool: one Arthur D. Little consultant noted that "in its simplest form, a data dictionary is a well-organized, up-to-date notebook containing basic information about data elements" [36, page 102]. But, as with the DBA, the data dictionary slipped from the managerial into the technical – after the term was applied to scores of software products in the late 1970s [21] it came simply to describe that portion of the DBMS where DML definitions were kept.

One IBM advocate of the data dictionary approach [20,23] likened data to money: "[o]nce management realizes the relationship of reliable data to corporate well-being, they will treat their data with the same care used to handle their cash." Nolan made a similar pitch in his book *Managing the Data Resource Function,* the title of which suggested that information, like people and money, was a vital resource of business and therefore deserved similar managerial attention [79]. Indeed, the claims made by Nolan that the DBA would be charged with overall responsibility for all corporate information, using computer technology where appropriate but ultimately claiming managerial rather than technical authority, directly prefigure those made more generally for the new position of Chief Information Office or CIO in the 1980s [103].

6. EARLY DBMS SYSTEMS IN USE

The DBMS enjoyed considerable practical success during the 1970s. By the end of the decade, most large computer installations had installed a DBMS package of some kind. Many of the most financially successful products of the independent software industry were DBMS or file management packages. Adoption of data base management software proved to be a boon to application programmers. In administrative applications of the kind traditionally carried out by corporate data processing departments, an enormous amount of programmer time was taken up doing the things that DBMSs were supposed to automate. They made programs cheaper to develop, much easier to maintain, and facilitated the integration of different business tasks. Data base management technology as defined by the DBTG was very good at dealing with very uniformly structured, hierarchical data of the kind found on administrative forms.[10]

Yet the DBMS never quite lived up to the expectations of people like Nolan, who saw it as a managerial panacea. Indeed, the managerial hype that developed around DBMS technology may have made it hard for firms to make informed technical decisions. As early as 1973, a report [35] by two Booz, Allen & Hamilton consultants suggested that both software and the hardware needed remained immature, that little experience so far existed in its use and that the generalized features offered by the DBMS brought a hefty performance penalty and might well trigger the purchase of more memory or a new processor unit. Most of the true costs were hidden, particularly the staff requirements. As they put it, "Some DBMSs are as complex as the operating system which services them. Also, this group must continuously apply and test new program fixes and new features to keep the system 'alive and well.' It is not uncommon to see a small systems programming team double or even triple as the result of a DBMS" [35, page 74]. Later reports, in [97], suggest that these problems continued for several years, and that many firms installed DBMS packages because of a “bandwagon” rather than a careful and informed evaluation. Despite direct access by executives being a theoretical keystone of the data base as an MIS tool, no surveys of the early 1970s were able to find any firms where the data base was used directly by managers, or even by analysts [78, page 113].

Companies keen to get their hands on a DBMS had to go to considerable lengths. Richard F Schubert of chemical firm B.F. Goodrich had been part of the DBTG, and led his company into implementing its own system IDMS based on a stripped down version of the CODASYL proposals. It was used to support batch mode applications such as billing and accounting as well as on-line access to order entry and its inventory of finished goods [57]. In 1973, Goodrich sold the rights to IDMS to marketing savvy entrepreneur John Cullinane, who by the early 1980s had built an eponymous software firm, one of the era’s largest and fastest growing, around it [68, 242-246]. Few companies were prepared to go this far to get a DBMS, and indeed experts of the early 1970s agreed that the exceptionally complex and generalized nature of the technologies involved made the selection of a good package far more sensible than trying to develop a system in-house.

Even among firms acquitting the most advanced DBMS packages, on-line use was limited and managerial applications rare. Let us consider two examples of firms using commercially supplied DBMSs in the early 1970s. McDonnell Douglas, using IBM’s IMS system, claimed to have created a centralized data base containing all the information previously stored in 264 files covering things like spare parts, production scheduling, bill of materials handling, and inventory management. This made it much easier to change the 95 existing programs that relied on these files, and to set up automatic cross references between different records and, it hoped, to move toward on-line operation in the future [55]. The second firm, much smaller, was devoted to accounts receivable processing for doctors. It used an DBTG influenced DBMS on its Xerox computer to lower its daily processing times shorter for updates and design its new program more rapidly. The results pleased it, despite the fact that the DBMS consumed a large part of the computer’s memory and used ten times more processor capacity than the tape based version. It had moved cautiously into on-line operation – while records were retrieved using terminals, all updates were queued and applied at night while the system was off-line in the belief that this “greatly reduces the possibility of a catastrophic loss of data” [14, page 63].

According to a 1975 survey of large industrial firms [90], about one third were using some kind of advanced file or data base management system. Of that third, around half were using systems intended for direct ad-hoc querying by non-programmers, such as MARK/IV, and half were using systems designed to integrate with the conventional programming languages such as COBOL. Hybrid systems, of the type envisioned by CODASYL, had yet to make much impact. Only about a quarter of the systems were used primarily for on-line access, and only two firms claimed to have implemented a data base for the entire firm, though most reported using it for multiple areas of the business. This was very slow to change. Five years later a survey of management information systems in thirty two large corporations found that most of these companies had now installed powerful DBMS packages [27]. Yet when the researchers looked at the actual use made of these systems they found that, "The users surveyed were only beginning to develop DBMS applications.... This is possibly because of the difficulties involved in developing and controlling such activities” [27, page 28].

Even products designed explicitly for use by non-specialists found their main markets to be among data processing specialists. Because they cost less and could run on more modest hardware, file management systems remained more widely used than fully fledged DBMSs. The 1975 survey by Powers found that 41% of firms using these packages reported that information could only be retrieved with the aid of a programmer. Unlike the more powerful systems designed primarily for application programmers to use, these systems were still used primarily (in 77% of firms) with files stored on tape rather than on disk. These systems still worked with individual files rather than vast integrated data bases – indeed, 55% of their users had not even begun to integrate files to remove redundant information.

During the 1970s, the MARK IV file management system became the most successful single product in the admittedly short history of the industry: the first to reach the milestones of $1 million, $10 million, and $100 million in cumulative sales. Compared to the DBTG proposals, its capabilities were modest. Its initial appeal was straightforward: first it was highly efficient in batch operation, and second it had been designed for use by non-

[10] Like file management systems before them the new systems still demanded that each record in a file (or “record type”) include exactly the same fields, each populated with data in exactly the same format. In addition, because relationships (or “sets”) were specified in the Data Definition Language, and so built into the database, the data base designer was forced to specify a complete and coherent set of linkages between different files – something that proved essentially impossible to do for the kind of large-scale, complete and multifunctional data bases envisioned by MIS proponents.

programmers. Requests for data were entered onto one of four simple paper forms, and then keypunched into computer form for later processing. But even MARK IV found its main audience among programmers. As time went by, development of MARK IV focused more and more on the needs of full time programmers, who used it as a foundation for the construction of complex application programs. An official company history credited this process to the influx of data processing specialists into its "IV League" user group, which ensured that their opinions "overwhelmed the voices of the non-programming end users" in the company's planning [43, page 9.26]. The proceedings of this group suggest that non-specialists found advanced work harder than had been expected. According to an Eastern Airline representative, while most of its 200 users were "a complete new breed of coders… non-programmers, little or no data processing background," attempts to train them in information retrieval techniques without giving an understanding of what went on in "the mysterious black box" of the computer had failed. Contrary to their expectations, "The only users able to move into extended capabilities with any degree of success were those with some data processing background" [65, Appendix F].

While there was a substantial demand for products that would let non-specialists produce computerized reports without the assistance of programmers, the leading DBMS systems did not do a good job of meeting this. One of the most successful software products of the 1970s, Pansophic's Easytrieve, was an easy to use report generation system designed to extract information from files and data bases. Easytrieve thrived in competition with more complex DBMS and file management software, and many firms purchased the optional modules needed to use it in conjunction with the most powerful DBMSs [87].

By the end of the 1970s, then, it was clear that DBMS technology had failed to live up to the hopes vested in it by its more managerially focused promoters. While powerful DBMS systems were now common in large corporations, few were being used to support new kinds of managerial application. Even the most sophisticated DBMS systems were used mostly in batch mode rather than on-line, and by programmers rather than managers. The data base management system was more of an improved file management system. Massive, integrated data stores remained very hard to construct, while interactive computer models of the kind anticipated by advocates of MIS remained conspicuous by their absence.

7. THE DATA BASE MANAGEMENT SYSTEM SINCE 1980

In 1973, Charles W Bachman was awarded the Association for Computing Machinery's Turing Medal – the most prestigious award in computer science. The citation singled out his creation of the pioneering IDS system (which it retroactively termed a DBMS) and his work on the DBTG to incorporate these ideas into its specifications. This award was in itself an important event, representing a new level of acceptance among computer science researchers of data base problems as intellectually respectable subjects of inquiry alongside better established areas such as numerical analysis, compiler theory and the theory of algorithms. The event is better remembered, however, for Bachman's speech [7]. Entitled "The Programmer As Navigator," it developed the idea that the shift to DBMS technology represented something akin to the Copernican revolution – in that the work of programmers would now revolve around the data base rather than the hardware of the computer. Though this prophecy took several decades to come true, knowledge of data base systems has now become a fundamental requirement for virtually all administrative applications programming, systems analysis and advanced web design work. But, as its title also implied, the impact of generalized DBMS would be much greater for programmers than for managers.

The acceptance of the DBTG concept of a data base management system thus implied a new and more concrete vision of what a data base was – basically a body of electronic data that could be managed by a data base management system. As such, the commercial success of DBMS packages supported the growing prestige of corporate computing staff, against attempts by information scientists and documentationalists [5] to turn the library, rather than the computer room, into the heart of any corporate information system. Despite the MIS influenced hopes of the 1970s that a DBMS could be the heart of a system including all corporate information, it proved adept at handling only a small subset of this material. The data base, as realized through an extension of existing file processing tools, embodied the highly structured, administrative transaction-oriented view of information held by data processing staff and computer vendors.

The narrowing of the data base concept, and its close association with the DBMS, also represented a shift away from the idea, implicit in much earlier discussion of information retrieval, that all important information was scientific or at least was amenable to the same retrieval techniques as scientific information. The data base concepts pioneered by elaborate, military systems of the 1960s such as SDC's TDMS – on-line access, flexibly structured data, interactive definition of data formats by users, played little part in the leading commercial systems of the 1970s. Neither was there a significant commercial market for products based on these technologies. Though the industrial research budgets of leading corporations might have paid for subscriptions to the newly available on-line scientific data bases of the 1970s [46], the managers and computing departments of the same companies had little interest in using these technologies to manage their own information. The commercial emergence of on-line information services, discussed in great detail in [16], appears to have taken place in almost complete isolation from work on DBMS packages.

The DBMS concept proved far more important and longer lasting than the particular methods for its realization put forward by the CODASYL DBTG. During the 1970s a new approach, the relational model [25, 33], gradually gained acceptance among database researchers. The relational model was far more conceptually elegant and flexible than the network model endorsed by CODASYL, which proved both restrictive (because relationships must be specified when the data base is designed) and insufficiently abstracted from the physical storage of data (programmers were still forced to write code to navigate explicitly from one record to another when working with linked data). Because the relational model shifted the responsibility of specifying relationships between tables from the person designing them to the person querying them, it permitted tables to be joined in different ways for different purposes. This turned out to be necessary (if not sufficient) for the establishment of large, general purposes data bases shared between different departments and computer systems. The relational model has also been praised for its non-procedural nature – further separating the user from the physical storage mechanisms involved [75]. This simplified programming and insulated application code from changes in the database structure. Accepting his own Turing award in 1981 for his development of the relational data base model, Edgar F. Codd suggested that the CODASYL network model had forced the programmer to become too much of a navigator, at too low a logical level [32].

Use of early DBMS systems was highly concentrated. According to internal reports prepared by one software firm, as late as 1981 TOTAL, the market leader, had just 4,171 installations while IBM's IMS won second place with an estimated 1,500 [82]. The first widely used relational DBMS, Oracle, was launched in 1980 and found an early niche in the rapidly growing market for minicomputer systems. During the 1990s, relational systems gained the power and maturity to gradually edge out earlier mainframe products such as IMS, though even today the transition is far from complete. At the same time, the increasing power of personal computer systems opened new niches for DBMS technology on desktop computers and inexpensive departmental servers. Almost every custom business application produced during the past decade relies on a relational DBMS to store and retrieve data. Relational DBMS systems are widely used on personal computers. Indeed, Microsoft now bundles a version of its powerful SQL Server DBMS with the "professional" editions of its Office suite, and has even adapted it for use with its Pocket PC hand held computers. Microsoft has long aimed, though so far without success, to replace the conventional file system and the email repositories found on today's Windows operating systems with a multitalented DBMS. In some ways, the DBMS has indeed become a universal container for computer data.

8. CONCLUSIONS

The data base management system provides an interesting example of the tensions hidden behind phrases such as "information technology." The progression of the concepts of data base and data base management system over the 1960s and 1970s demonstrate an unmistakable tension between the rather limited and technically focused achievements of actual information systems and the universal, almost utopian claims that information problems can be defined, and therefore solved, for the general case if only the right tools or technologies can be deployed. The technologies of the file management system, however much improved, could never realize the grand dreams set forward for corporate data bases as universal sources of information. While the invention of the DBMS concept initially revived hopes for the creation of all-powerful data bases, in the longer term its effect was to redefine the very concept of a data base (or as we now say, database) as the contents of a DBMS.

Despite their remarkable ubiquity, DBMSs based on the relational model continued to incorporate the same assumptions about information as earlier file management systems. In particular, the complexity of relational query construction meant that to query and update the data base still required the involvement of a programmer, a specially written application program, or trained specialist. The designers of the now-standard SQL language had assumed that replacing algebraic characters with words such as "SELECT" would make it easy for managers to write their own queries [74], but the complexity and rigor could not be removed so easily. And although the relational model made it easier to join tables together in different ways, data base designers still had to specify the exact format of each column within the table, and include exactly the same fields in each row.

As a result, the DBMS was very well suited to the bureaucratic records for things such as payroll administration, because each record included the same pieces of data (years of service, SSN, hourly rate, overtime status and so on). It made it very simple and efficient to update information, and so is well suited to administrative systems where records are constantly updated. On the other hand, it was entirely useless for representing and searching less rigidly formatted data, such as full-text records, correspondence, or even scientific abstracts. There has certainly been no shortage of interest by database researchers in the design of alternative and more flexible models. Many of these have been promoted in commercial products, including object oriented database systems [114], multi-valued databases and other approaches. Indeed, "post relational" has become a marketing buzzword during the last decade, and just like "relational" in the 1980s, and indeed "data base management system" in the 1970s, it has been applied to such a broad range of products as to thoroughly blur its actual meaning [37].

The point remains, however, that today's dominant data base technology still includes rigid concepts of fields and data types inherited from punched card systems and remains far from living up to the vision of a universal repository for data of all kinds. Only with the rise of the World Wide Web in the mid-1990s did widespread attention turn back to the indexing and management of huge amounts of natural language information. Systems such as AltaVista and, more recently, Google have proved remarkably adept at returning relevant results from a sea of unstructured data. However, these technologies remain quite distinct from mainstream DBMS systems.

As DBMS use proliferated, firms found themselves unable to integrate all corporate data into a single pool in the manner promised by early data base advocates. When DBMS technology achieved almost universal use, large firms were left with hundreds or thousands of disconnected and duplicated data bases and no easy way to merge them. Data warehousing, one of the leading obsessions of corporate IT departments and consulting firms of the mid- and late-1990s, was an attempt to construct enormous read-only data bases for reporting purposes in which all data was linked and reformatted into a standard form. Firms intended to use these buckets of facts for "data mining" and the provision of "business intelligence" to best their competitors [38]. The corporate data pools imagined forty years ago have inched ever closer to reality, yet the messy reality of specialized, limited and inflexible data storage technologies continues to contrast with the pristine simplicity of the original vision.

9. APPENDIX: FILE AND DATA BASE MANAGEMENT PACKAGES OF THE 1960S AND EARLY 1970S

System	Produced By	Approximate release date	Notes
9PAC	SHARE	1959	Consisted of generalized File Maintenance and Report Generation systems. Developed cooperatively by a group of IBM 709 installations through the SHARE organization, and heavily influenced by the earlier generalized report generation system produced by General Electric's Hanford site. 9PAC used tape storage, but included record descriptions in the file header and allowed record hierarchies within a file.
GIRLS (Generalized Information Retrieval and Listing), MARK I-III	Douglas Aircraft	1962-196?	A series of programs developed by John A. Postley, beginning with generalized reporting and evolving into a batch based file management system intended for use by non-programmers. [43, 88, 89]. Commercialized as Mark IV (see below).
IDS (Integrated Data Store)	General Electric (later Honeywell Information Systems)	1963 onward for internal use. GE product from mid-1960s.	File management system, evolved from internal GE application used to track inventory levels. Batch based, used by programmers, often with COBOL. Pioneered the creation of links between records in different files. Through its creator, Charles W Bachman[11], was a major influence on CODASYL specifications. [8, 22, 92]. With Honeywell takeover of GE computer business, ran on Honeywell H600 and H6000 series machines.
TDMS/CDMS (Time-shared/ Commercial Data Management System)	SDC	1967, 1969	On-line timeshared system, allows non-programmers to create data base definitions, load data, and issue queries. Ran on IBM/360 series. TDMS by military. See [10, 116-121] and [107]. Renamed CDMS and offered as a commercial product for the IBM/360 series from 1969, but required SDC's own operating system. Then as a flagship service for SDC's nationwide network of SDC Data Centers. [10, 16, 19, 101].
IMS (Information Management System)	IBM	1968 as product	Evolved from application produced in collaboration with Rockwell for use at NASA to tack components for the Apollo program. [13]. Used hierarchical data model, versus the network model of IDS, but supported online applications. Still in use on some mainframes.
MARK IV	Informatics General	1968	File management system, developed from above. Intended for batch mode use by non-programmers. Simple interface for creation, updating and querying of files. Highly successful product, though use came primarily from programmers. On-line support late and limited. [19, 101, 111]
GIS (Generalized Information System)	IBM	1969	Intended to function in batch or on-line mode. Oriented toward ad-hoc querying by non-specialists. Ambitious but repeatedly delayed. [18] [19] [61].[86]
IDMS (Integrated Database Management System)	B F Goodrich initial work. Cincom as product.	Initial development 1969-1972.	Developed internally by B F Goodrich for internal use on IBM/370 mainframe. Powerful DBMS heavily influenced by CODASYL concepts, including the network data model, and by previous experience with IDS. Worked primarily in batch mode, with support for on-line applications. Marketed by Cullinane Corporation from 1973 onward. One of the most successful DBMS products of the late 1970s and early 1980s. [57, 59, 68, 95, 96].
TOTAL	Cincom	1970	Firm founded by former IBM salesman. TOTAL was an early CODASYL influenced DBMS, based on the network data model. [3].

[11] The Charles W. Bachman Papers (CBI 125), Charles Babbage Institute, University of Minnesota, Minneapolis include considerable material on IDMS, though most of this dates from after Bachman joined Cullinane in 1980.

ADABAS	Software AG	1970	Highly efficient DBMS using “inverted file” structures to index files, providing high speed batch mode capabilities for application developers. ADASCRIPT query language intended for use by non-specialists. Still sold.
DMS 1100	Univac	1971	DBMS based on CODASYL concepts, produced by Univac for users of its 1100 series machines. [15, 105]
EASYTREIVE	Pansophic (distributed and eventually acquired)	1973	An easy to use querying and report generation system, combined with a tape based file management system. Highly successful, and one of the top ten products of the independent packaged software industry during the 1970s. Its success illustrates the continuing demand for simple systems, and the popularity of optional modules to connect it to IMS and IDMS as an alternative front-end system for querying shows the limitations of early DBMS systems in handling ad-hoc queries and reports. [87]
System 2000	MRI	Early 1970s	Company founded by University of Texas researchers. Hierarchical system, but influenced by CODASYL model. Offered for IBM, CDC, Univac machines. Included report capabilities, good support for ad-hoc queries. [106]

10. ACKNOWLEDGMENTS

Thanks to Rick Snodgrass for introducing this paper to SIGMOD Record, to Mary Ellen Bowden for encouraging me to write it in the first place, to Boyd Rayward for his close attention and helpful comments through multiple drafts, to ACM SIGMOD for funding my oral history interview with Charles W. Bachman and to the Computer History Museum for supporting my oral history interview with Robert L. Patrick.

11. REFERENCES

[1] Akera, A. Voluntarism and the Fruits of Collaboration. *Technology and Culture*, *42*, 4 (October 2001), 710-736.

[2] Anonymous (ed.), *Disc File Applications: Reports Presented at the Nation's First Disc File Symposium*. American Data Processing, Inc., Detroit, 1964.

[3] Anonymous MIS, the Impossible Dream? *Infosystems*, *20*, 2 (February 1973), 70.

[4] Anonymous A panel Discussion on Time-Sharing. *Datamation*, *10*, 11 (November 1964), 38-44.

[5] Aspray, W. Command and Control, Documentation, and Library Science: The Origins of Information Science at the University of Pittsburgh. *IEEE Annals of the History of Computing*, *21*, 4 (October-December 1999), 4-20.

[6] Bachman, C.W. *Oral History Interview by Thomas Haigh, 25-26 September 2004, Tucson, AZ, 2004*, Personal collection of author, to be published in ACM Digital Library (details unknown).

[7] Bachman, C.W. The Programmer as Navigator. *Communications of the ACM*, *16*, 11 (November 1973), 653-658.

[8] Bachman, C.W. and Williams, S.B. *The Integrated Data Store, A General Purpose Programming System for Random Access Memories,* May 1 1964, Charles W. Bachman Papers (CBI 125), Charles Babbage Institute, University of Minnesota, Minneapolis.

[9] Bashe, C.J., Johnson, L.R., Palmer, J.H. and Pugh, E.W. *IBM's Early Computers*. MIT Press, Cambridge, MA, 1986.

[10] Baum, C. *The System Builders: The Story of SDC*. System Development Corporation, Santa Monica, 1981.

[11] Bello, F. How to Cope with Information. *Fortune*, *62*, 3 (September 1960), 162-167, 180-182, 187-189, 192.

[12] Bello, F. The Information Theory *Fortune*, 1953, 136-141, 149-150, 152, 154,156, 158.

[13] Blackman, K.R. IMS Celebrates Thirty Years as an IBM Product. *IBM Technical Journal*, *37*, 41998), 596-603.

[14] Blanchard, J.S. We Bet Our Company on Data Base Management. *Datamation*, *20*, 9 (September 1974), 61-65.

[15] Borgerson, B.R., Hanson, M.L. and Hartley, P.A. The Evolution of the Sperry Univac 1100 Series: A History, Analysis, and Projection. *Communications of the ACM*, *21*, 1 (January 1978), 25-43.

[16] Bourne, C.P. and Hahn, T.B. *A History of Online Information Services: 1963-1976*. MIT Press, Cambridge, MA, 2003.

[17] Bowles, M.D. The Information Wars: The Two Cultures and the Conflict in Information Retrieval, 1945-1999. In Bowden, M.E., Hahn, T.B. and Williams, R.V. eds. *Proceedings of the 1998 Conference on the History and Heritage of Scientific Information Systems*. Information Today, Inc., Medford, NJ, 1999, 156-166.

[18] Bryant, J.H. and Semple, P. GIS and File Management. In Association for Computing Machinery ed. *Proceedings of the 21st National Conference*. ACM, New York, 1966, 97-107.

[19] Byrnes, C.J. and Steig, D.B. File Management Systems: A Current Summary. *Datamation*, *15*, 11 (November 1969), 138-142.

[20] Cahill, J.J. A Dictionary/Directory Method for Building a Common MIS Data Base. *Journal of Systems Management*, *21*, 11 (November 1970), 23-29.

[21] Canning, R.G. The Data Dictionary/Directory Function. *EDP Analyzer*, *12*, 10 (November 1974).

[22] Canning, R.G. Data Management: File Organization. *EDP Analyzer*, *5*, 12 (December 1967).

[23] Canning, R.G. The Debate on Data Base Management. *EDP Analyzer*, *10*, 3 (March 1972), 1-16.

[24] Canning, R.G. New Views on Mass Storage. *EDP Analyzer*, *4*, 2 (February 1966).

[25] Chamberlin, D.M. Relational Data-Base Management Systems. *ACM Computing Surveys*, *8*, 1 (March 1976), 43-66.

[26] Charles Babbage Institute *Software History Dictionary Project: Information Management System (IMS)*, Charles Babbage Institute, February 17 2004 [cited April 15 2006] http://www.cbi.umn.edu/shp/entries/ims.html

[27] Cheney, P.H. and Lyons, N.R. MIS Update. *Data Management*, *18*, 10 (October 1980), 26-32.

[28] CODASYL Data Description Language Committee *CODASYL Data Description Language: Journal of Development, June 1973.* U.S. Govt. Print. Off., Washington, DC, 1974.

[29] CODASYL FORTRAN Data Base Manipulation Language Committee CODASYL FORTRAN Data Base Facility *Journal of Development*, 1977.

[30] CODASYL Systems Committee *CODASYL Data Base Task Group: April 1971 Report*. Association for Computing Machinery, New York, 1971.

[31] CODASYL Systems Committee *Feature Analysis of Generalized Data Base Management Systems*. [n.p., Distributed by the Association for Computing Machinery], New York, 1971.

[32] Codd, E.F. Relational Database: A Practical Foundation for Productivity. *Communications of the ACM*, *25*, 2 (February 1981), 109-107.

[33] Codd, E.F. A Relational Model for Large Shared Databanks. *Communications of the ACM*, *13*, 6 (June 1970), 277-390.

[34] Copeland, D.G., Mason, R.O. and McKenney, J.L. SABRE: The Development of Information-Based Competence and Execution of Information-Based Competition. *IEEE Annals of the History of Computing*, *17*, 3 (Fall 1995), 30-57.

[35] Cuozzo, D.E. and Kurtz, J.F. Building a Base for Data Base: A Management Perspective. *Datamation*, *19*, 10 (October 1973), 71-76.

[36] Curtice, R.M. Some Tools for Data Base Development. *Datamation*, *20*, 7 (July 1974), 102-106.

[37] Date, C.J. What do you mean, 'Post Relational'? 2000 [cited April 20 2004] http://www.pgro.uk7.net/cjd1a.htm

[38] Davenport, T.H. Competing on Analytics. *Harvard Business Review*, *84*, 1 (January 2006), 98-107.

[39] Dearden, J. How to Organize Information Systems. *Harvard Business Review*, *43*, 2 (March-April 1965), 65-73.

[40] Edwards, P. *The Closed World: Computers and the Politics of Discourse in Cold War America*. MIT Press, Cambridge, MA, 1996.

[41] Elmasri, R. and Navathe, S.B. *Fundamentals of Database Management Systems*. Benjamin/Cummings, New York, 1989.

[42] Flynn, R.L. A Brief History of Data Base Management. *Datamation*, *20*, 8 (August 1974), 71-77.

[43] Forman, R.L. *Fulfilling the Computer's Promise: The History of Informatics, 1962--1982*. Informatics General Corporation, Woodland-Hills, CA, 1984.

[44] Fry, J.P. and Sibley, E.H. Evolution of Data-Base Management Systems. *ACM Computing Surveys*, *8*, 1 (March 1976), 7-42.

[45] Gorry, G.A. and Morton, M.S.S. A Framework for Management Information Systems. In Nolan, R.L. ed. *Managing the Data Resource Function*. West Publishing Company, St. Paul, 1974, 87-105.

[46] Hahn, T.B. Pioneers of the Online Age. *Information Processing and Management*, *32*, 11996), 33-48.

[47] Haigh, T. The Chromium-Plated Tabulator: Institutionalizing an Electronic Revolution, 1954-1958. *IEEE Annals of the History of Computing*, *23*, 4 (October-December 2001), 75-104.

[48] Haigh, T. Inventing Information Systems: The Systems Men and the Computer, 1950-1968. *Business History Review*, *75*, 1 (Spring 2001), 15-61.

[49] Haigh, T. Software in the 1960s as Concept, Service, and Product. *IEEE Annals of the History of Computing*, *24*, 1 (January-March 2002), 5-13.

[50] Hare, V.C., Jr A Special Report on the SIGBDP Forum: 'The New Data Base Task Group Report". *Data Base*, *3*, 3 (Special Issue 1971), 1.

[51] Haslett, J.W. Total Systems - A Concept of Procedural Relationships in Information Processing. In Meacham, A.D. and Thompson, V.B. eds. *Total Systems*. American Data Processing, Inc., Detroit, MI, 1962, 16-19.

[52] Head, R.V. Data Base Symposium. *Datamation*, *11*, 11 (November 1965), 41.

[53] Head, R.V. Management Information Systems: A Critical Appraisal. *Datamation*, *13*, 5 (May 1967), 22-27.

[54] Head, R.V. MIS-II: Structuring the Data Base. *Journal of Systems Management*, *21*, 9 (September 1970), 37-38.

[55] Hollenbach, R. An Application of a Data Base System. *Data Management*, *11*, 9 (September 1973), 30-31.

[56] Hughes, T.P. *Rescuing Prometheus*. Pantheon Books, New York, 1998.

[57] Huhn, G.E. The Data Base in a Critical On-Line Business Environment. *Datamation*, *20*, 8 (September 1974), 52-56.

[58] International Business Machines Introduction to the AN/FSQ-7 Combat Direction Central and AN/FSQ-8 Combat Control Central, Ed Thalen, 1965 [cited April 22 2006] http://ed-thelen.org/SageIntro.html

[59] Karasz, P. and Hoelscher, C. History of IDMS, Australian IDMS Users Group, 1999 [cited July 5 2002] http://users.senet.com.au/~cherlet/idmshist.html

[60] Keller, A.E. The Man Behind Systems at Shell Oil. *Business Automation*, *7*, 2 (February 1962), 20-24.

[61] Kircher, P. Breakthrough in Management Information Systems. *Journal of Data Management*, *7*, 2 (February 1969), 28-31.

[62] Leavitt, H.J. and Whisler, T.L. Management in the 1980s. *Harvard Business Review*, *36*, 6 (November-December 1958), 41-48.

[63] Luke, J.W. Data Base Systems: Putting Management Back In The Picture. *CSC Report*, *9*, 11975), 8-12.

[64] Lyon, J.K. The Role of the Data Base Administrator. *Data Base*, *3*, 4 (Winter 1971), 11-12.

[65] Mark IV User Group *Proceedings of the MARC IV User Group: meeting IX*, Jan. 25-27, 1971, San Francisco, CA, 1971, Evan Linick Collection of Proceedings of the Mark IV User Group (CBI 130), Charles Babbage Institute, University of Minnesota, Minneapolis.

[66] Martin, J. *Computer Data-Base Organization*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.

[67] McCaffrey, J.J. *From Punched Cards to Personal Computers*, June 10 1989, John J. McCaffrey Memoirs (CBI 47), Charles Babbage Institute, University of Minnesota, Minneapolis.

[68] McClellan, S.T. *The Coming Computer Industry Shakeout: Winners, Losers, and Survivors*. John Wiley & Sons, New York, 1984.

[69] McCracken, D.D., Weiss, H. and Lee, T.-h. *Programming Business Computers*. John Wiley and Sons, New York, 1959.

[70] McGee, R.C. "Preliminary Manual for the Generalized File Maintenance System, SSD 046," December 23 1958, SHARE Records, 1955-86, NMAH 567, Archives Center, National Museum of American History, Behring Center, Smithsonian Institution, Washington, DC.

[71] McGee, R.C. The Structure of a Standard File, SSD 045, November 10 1958, SHARE Records, 1955-86, NMAH 567, Archives Center, National Museum of American History, Behring Center, Smithsonian Institution, Washington, DC.

[72] McGee, R.C. and Tellier, H. A Re-Evaluation of Generalization. *Datamation*, *6*, 4 (July-August 1960), 25-29.

[73] McGee, W.C. Generalization: Key To Successful Electronic Data Processing. *Journal of the Association for Computing Machinery*, *6*, 1 (January 1959), 1-23.

[74] McJones, P. The 1995 SQL Reunion: People, Projects and Politics, August 20 1997 [cited 12th Feb 2000] http://www.mcjones.org/System_R/SQL_Reunion_95/sqlr95-System.html

[75] Michaels, A.S., Mittman, B. and Carson, C.R. A Comparison of the Relational and CODASYL Approaches to Data-Base Management. *ACM Computing Surveys*, *8*, 1 (March 1976), 125-151.

[76] Miller, E. Information Retrieval--1961. *Datamation*, *7*, 10 (October 1961), 19-21.

[77] National Research Council *Funding A Revolution: Government Support for Computing Research*. National Academy Press, Washington, DC, 1999.

[78] Nolan, R.L. Computer Data Bases: The Future is Now. *Harvard Business Review*, *51*, 5 (September-October 1973), 98-114.

[79] Nolan, R.L. (ed.), *Managing the Data Resource Function*. West Publishing Co, New York, 1974.

[80] Olle, T.W. Recent CODASYL Reports on Data Base Management. In Rustin, R. ed. *Data Base Systems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972, 175-184.

[81] Osborn, R.F. GE and UNIVAC: Harnessing the High-Speed Computer. *Harvard Business Review*, *32*, 4 (July-August 1954), 99-107.

[82] Pansophic Systems Incorporated Competitive Products Survey: Data Base Management Systems Software Products, December 31 1981, Author's personal collection (donated by Joseph A Piscopo to Charles Babbage Institute, University of Minnesota).

[83] Parker, R.W. The SABRE System. *Datamation*, *11*, 9 (September 1965), 49-52.

[84] Patrick, R.L. Oral History Interview with Thomas Haigh, February 16 2006, Mountain View, CA, 2006, Forthcoming from Computer History Museum, Mountain View, CA.

[85] Performance Development Corporation An Interview With Charles W Bachman (Part II), Data Base Newsletter, Volume 8, No. 5, September 1980, Charles W. Bachman Papers (CBI 125), Charles Babbage Institute, University of Minnesota, Minneapolis.

[86] Pickard, A. The Status of GIS. In *SHARE XXXI Proceedings*, 1968, 2.27-22.43.

[87] Piscopo, J.A. "Oral History Interview by Thomas Haigh, 03 May, Washington DC," 2002, OH 342, Charles Babbage Institute, University of Minnesota, Minneapolis.

[88] Postley, J.A. Mark IV: Evolution of the Software Product, a Memoir. *IEEE Annals of the History of Computing*, *20*, 1 (January-March 1998), 43-50.

[89] Postley, J.A. and Jackobsohn, H. The Third Generation Computer Language: Parameters do the Programming Job. In Data Processing Management Association ed. *Data Processing, vol. 11*, Los Angeles, California, 1966, 408-415.

[90] Powers, V. Implementing Generalized Data Base Management Systems. *Data Management*, *13*, 5 (May 1975), 14-16.

[91] Pugh, E.W., Johnson, L.R. and Palmer, J.H. *IBM's 360 and Early 370 Systems*. MIT Press, Cambridge, MA, 1991.

[92] Reside, K.D. and Seiter, T.J. The Evolution of an Integrated Data Base. *Datamation*, *20*, 9 (September 1974), 57-60.

[93] Romberg, B.W. Data Bases: There Really Is a Better Way to Manage Your Files. *Infosystems*, *20*, 5 (May 1973), 56-57, 58.

[94] Rowan, T.C. The Recruiting and Training of Programmers. *Datamation*, *4*, 3 (May-June 1958), 16-18.

[95] Schubert, R.F. Basic Concepts in Data Base Management. *Datamation*, *18*, 7 (July 1972), 42-47.

[96] Schubert, R.F. Directions in Data Base Management Technology. *Datamation*, *20*, 9 (September 1974), 48-51.

[97] Schussel, G. When Not To Use a Data Base. *Datamation*, *21*, 11 (November 1975), 82, 91, 98.

[98] Shannon, C.E. and Weaver, W. *The mathematical theory of communication*. University of Illinois Press, Urbana, 1949.

[99] Simon, L. and Sisson, R. Evolution of a Total System. *Total Systems Letter*, *1*, 11 (January 1966), 1-4.

[100] Statland, N. and Hillegass, J.R. Random Access Storage Devices. *Datamation*, *9*, 12 (December 1963), 34-45.

[101] Steig, D.B. File Management Systems Revisited. *Datamation*, *18*, 10 (October 1972), 48-51.

[102] Stone, M.D. Data Processing and the Management Information System: A Realistic Evaluation of Data Processing's Role in the Modern Business Enterprise. In American Management Association ed. *Data Processing Today: A Progress Report -- New Concepts, Techniques and Applications -- AMA Management Report Number 46*. American Management Association, Finance Division, New York, 1960, 14-22.

[103] Synnott, W.R. and Gruber, W.H. *Information Resource Management: Opportunities and Strategies for the 1980s*. John Wiley & Sons, New York, 1981.

[104] System Development Corporation *Preprint for Second Symposium on Computer-Centered Data Base Systems*, Sponsored by SDC, ARPA, and ESD, September 1 1965, Burroughs Corporation Records (CBI 90), Charles Babbage Institute, University of Minnesota, Minneapolis.

[105] The Diebold Group, I. *Current Status of Data Bases and Data Base Management*, The Diebold Group, Cambridge, Mass, 1976.

[106] Various. Panel Discussion: SIGBDP 3 -- Performance Measurement and Data Base Design. In Association for Computing Machinery ed. *Proceedings of the 1975 Annual Conference*. Association for Computing Machinery, New York, 1975.

[107] Vorhaus, A.H. TDMS: A New Approach to Data Management. *Systems & Procedures Journal*, *18*, 4 (July-August 1967), 32-35.

[108] Waites, W.G. MIS or IMS? *Journal of Systems Management*, *22*, 1 (January 1971), 32-34.

[109] Webster, E. and Statland, N. Instant Data Processing. *Business Automation*, *7*, 6 (June 1962), 34-36, 38.

[110] Weindling, R.E. Office Will Run Every Business Activity. *Office Management and American Business, 22*, 1 (January 1961), 12-15.

[111] Welke, L. A Review of File Management Systems. *Datamation*, *18*, 10 (October 1972), 52-54.

[112] Wellisch, H. From Information Science to Informatics: A Terminological Investigation. *Journal of Librarianship*, *4*, 3 (July 1972), 157-187.

[113] Wendler, C.C. What are the Earmarks of Effective Total Systems? *Systems & Procedures Journal*, *17*, 4 (July-August 1966), 29-31.

[114] Zand, M., Collins, V. and Caviness, D. A Survey of Current Object-oriented Databases. *ACM SIGMIS Database*, *26*, 1 (February 1995), 14-29.