"A Veritable Bucket of Facts":
Origins of the Data Base Management System, 1960–1980

Thomas Haigh

Abstract

The data base management system (DBMS) provides a vital underpinning for most of today's information systems. Yet commercial DBMSs have been geared to the highly structured information needs of corporate administration rather than the full-text searching and indexing needs of many scientific information systems. Why did DBMS technology take this path, given that many pioneering on-line data base systems were designed for textual information retrieval? The data base concept derives from early military on-line systems. While the idea of an integrated data base, or "bucket of facts," spread into corporate data processing and "systems" circles during the early 1960s, it was seldom realized in practice. The DBMS, however, was an update to an earlier technology: the generalized file-processing system. 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. This reflected and reinforced the gulf between scientific information problems in indexing and information retrieval and standard commercial tools for data management.

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 Web site, literature search, or Internet shopping trip. Yet very little research addresses the history of this vital technology or of the ideas behind it. We know little about its technical evolution and still less about its use.[1]

A DBMS is a very complex piece of system software. A single DBMS usually holds multiple data bases, each one consisting of many different tables full of data. A DBMS includes mechanisms for application programs to store, retrieve, and modify the data and allows people to query it interactively to answer specific questions. Specialists, known as data base administrators, control the operation of the DBMS and are responsible for creating new data bases and defining the table structures used to store data. One of the most important features of a 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 a DBMS, a data base can be restructured or moved to a different computer without disrupting the programs written to use it. A DBMS polices access to the stored data, giving access only to the 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 such DBMSs as Oracle are now often simply called databases.) Historically, though, the two ideas were distinct. The data base concept originated about 1960, around ten years before the idea of a DBMS

[1] While many data base textbooks include a few pages on the development of data base theory along with their introductory definitions—for example, R. Elmasri and S. B. Navathe (1989) do 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 (Fry & Sibley, 1976, pp. 19–29). A short history, focusing on the role of public funding in the emergence of the relational model, is found in a text by the National Research Council (1999, ch. 6). On the technical side, detailed comparisons of early systems are given in C. J. Byrnes and D. B. Steig (1969), CODASYL Systems Committee (1971b), D. B. Steig (1972), and L. Welke (1972).

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 operation. 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 humbler 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 DBMS 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 it disappointed as a managerial panacea. Most early DBMSs 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 qualities and distinctive strengths and weaknesses are promoted instead as universal solutions. The same pattern has been seen many times: with 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 broad human or managerial concepts of information and the far more constrained capabilities of specific automated systems.

The Data Base and the Management Information System

During the 1970s the DBMS was promoted 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 hopes attached to the DBMS, 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 (Stone, 1960). Milton D. Stone was, as many of his fellow speakers were, enthusiastic about the incredible potential of the MIS, then a new and exciting concept (Haigh, 2001b). 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. It was 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 in a timely fashion to every manager in a corporation every piece of information necessary for the performance of their duties. 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 (1960, p. 17).

Data processing was already well entrenched as the dominant term for administrative computing (Haigh, 2001a), 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 accountants (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 supervisors or "data processing technicians," dismissed by Stone as drones who would follow whatever instructions were placed in front of them (Stone, 1960, p. 21).

The early 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 (Shannon & Weaver, 1949) 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" (Wellisch, 1972), while researchers attempting to automate record searching began to call this work "information retrieval" (Bowles, 1999; Miller, 1961). Glowing reports in *Fortune* magazine informed businessmen of the power of information theory (Bello, 1953) and of information retrieval (Bello, 1960). In 1958 the combination of computers, operations research methods, and simulation was first dubbed "information technology" (Leavitt & Whisler, 1958).

Men such as Stone 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" (Stone, 1960, p. 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" (Weindling, 1961, p. 13). 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" (Haslett, 1962, p. 17; Keller, 1962).

These buckets, pools, and hubs seem quaint and rather unhelpful metaphors today, and 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 term *data base*, unknown in data processing circles at the time 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 be stored, refined, or piped as necessary. This conception 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. The 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 (Semi Autonomous Ground Environment) anti-aircraft command and control network. SAGE (Edwards, 1996; Hughes, 1998) 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, consistent representation of the various bombers, fighters, and bases to all its users. The System Development Corporation, or SDC (Baum, 1981), 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.

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 employed by far the world's largest concentration of programmers with experience in large-scale, real-time systems (Rowan, 1958). The company 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 (Anonymous, 1964a) 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 (System Development Corporation, 1965).

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. Robert V. Head, reporting on the event in *Datamation*, the leading trade magazine of business computing, 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 downtrodden." 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" (Head, 1965, p. 41).

It was around this time that the term *data base* made its first appearances in discussion of management information systems. In 1965 John Dearden, a professor of accounting at Harvard, 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 (1965). 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" (Simon & Sisson, 1966, p. 4).

The idea of the data base as a physical pool of data underlying an MIS was given early, clear, and highly influential support by Head, who defined the data base as the bottom level of a pyramidal structure (Head, 1967). 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 (Haigh, 2001b, pp. 45–50). The metaphor fits nicely with the idea of a data *base* supporting the rest of the information system. This conceit 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 (Head, 1970; Kircher, 1969; Wendler, 1966, p. 30).

SDC's attempt to push the data base concept into civilian discourse worked well. The term *data base* carried some specific associations with it, based on the particular characteristics of such firms as SDC and of military command and control projects. One of these was 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 between 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, funded with military money, all of which ran on the special, hugely expensive computers developed for SAGE (System Development Corporation, 1965). SDC's most ambitious attempt to commercialize data base technology came with a system called the commercial data management system, or CDMS, a derivative of an earlier system called TDMS (Time-shared Data Management System) developed under contract with the Advanced Research Projects Agency and given trial use at military installations. These systems allowed nonprogrammers to create data base definitions, 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 the commercial data management system through terminals connected to centralized computers were equally unsuccessful (Baum, 1981, pp. 116–121; Steig, 1972; Vorhaus, 1967).

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 items of data from all the routine operational systems (e.g., payroll, accounting, inventory, and billing) and somehow integrating them and making them available inside the data base.

File Management Systems and Data Processing

One of the two primary intellectual ingredients of the DBMS was the idea of a data base. The other was the "file management system" and its close relation, the "report generator." File management systems were intended to reduce the cost of producing routine administrative programs and 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 on "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 had quickly become apparent. Pioneering computer users had soon discovered that apparently simple clerical data processing activities, of the kind that were looked down on by enthusiasts of 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" of information from one file to another—all ubiquitous in computer systems today—all have their origins in punched-card systems.

The first generation of American data processing installations spent much more and took far longer than expected to get their machines up and running. Beginning with General Electric's famous 1954 use of a Univac computer to automate payroll processing (Osborn, 1954), data processing managers were shocked by the complexity of programming work and the rigid requirements computer technology imposed on such areas as data entry and the handling of special cases. As with 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 (Haigh, 2001a).

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 (McCracken, Weiss, & Lee, 1959, pp. 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 manufacturers began to supply their customers with standard functions to optimize these tasks (Bashe, Johnson, Palmer, & Pugh, 1986, pp. 181–185). This code 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 or she 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, to which a programmer could feed a description of the output desired and of the organization of the data inside the relevant "master file" and be rewarded with the desired report. The work of General Electric's team at the Hanford Nuclear Reservation (McGee & Tellier, 1960; McGee, 1959) was particularly important in establishing these techniques.

File management systems had their origin in the use of similar techniques to create and update data files and to 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 with 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 (Akera, 2001).

These techniques were very useful with tape storage. When firms began to store their data on disk drives, the extra complexity made the use of file management routines almost essential. The disk drive was first offered as a standard option for most major computer systems in 1962 (Anonymous, 1964b; Statland & Hillegass, 1963; Webster & Statland, 1962). 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 (Pugh, Johnson, & Palmer, 1991). A large disk system was the physical bucket into which facts could be placed, to be checked, added, 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 videotapes, 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 such on-line business systems as the celebrated SABRE airline reservation system (Copeland, Mason, & McKenney, 1995; Parker, 1965). 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 (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 (McCracken et al., 1959). 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 changed (e.g., to shift a growing file to its own disk and expand the storage areas for the remaining files) 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 disks (Canning, 1966, 1967). 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 retrieve records selectively 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. One of the most innovative and influential systems was General Electric's Integrated Data Store (IDS). This system began life about 1963 as an internal application used to track inventory levels. It pioneered new capabilities to link records in different files (e.g., a customer record with records for all the orders placed by that customer). This concept, known later as the "network data model," was a major influence on early DBMSs. A few years later, in 1965, what eventually became IBM's Information Management System (IMS) was produced in collaboration with North American Rockwell for use at NASA to handle the proliferation of parts involved in the Apollo program (Blackman, 1998). 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 (Forman, 1984; Postley, 1998; Postley & Jackabson, 1966).

The Data Base Management System and the Data Base Task Group

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 nonspecialists, 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 under way to add on-line access (Bryant & Semple, 1966).

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 the group 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.

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 end point 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 (Waites, 1971). 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 (Olle, 1972). On its formation in October 1965 the DBTG had originally been called the List Processing Task Force (its name was changed only in 1967).[2]

The work of the DBTG provided both a broad conceptual outline for what it now called a data base management system and detailed draft specifications for two specific parts of the overall system (one language for defining the data base structure and another 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.[3] This conceptual framework for the DBMS ultimately proved more

---

[2] 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) (Bryant & Semple, 1966).

[3] In 1969 the DBTG released its first major report on what it now called "data base management systems." Despite lobbying by such firms as General Electric to get their own systems adopted as the basis for a new standard, the group decided that no single existing system came close to proving the range of features required. Instead, the group surveyed the strengths and weaknesses 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 (CODASYL Systems Committee, 1971a) was issued and officially endorsed by CODASYL. The extensions to COBOL were reworked by a new standing committee into "Journal of Development" form as an official update to the COBOL standard (CODASYL Data Description Language Committee, 1974). Work on these standards continued into the 1980s, first through a new committee set up within CODASYL, and later at ANSI (American National Standards Institute). The specific proposals were controversial at the time, and several CODASYL members opposed them, including mainframe

influential than the DBTG's detailed proposals. Although 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, the task group insisted that future systems must provide all of them. A DBMS was expected to provide the efficient, batch-based access for programmers and hierarchical record-linking features that existing advanced file management systems such as GE's Integrated Data Store specialized in. However, it was also expected to allow nonprogrammers to use a simple, specially tailored interface to query and update the data base directly—the province of such systems as Mark IV. Similarly, the DBMS was expected to support interactive on-line use and batch operation with equal felicity (CODASYL Systems Committee, 1971b).

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.[4] Its final contribution was to insist that a standard DBMS allow more complex links 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 themselves ensured that all application programs and user requests would have access to them. The DBTG also decided that while the hierarchical approach used by such systems 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.

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 trend was accompanied by 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, multitasking 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 (Haigh, 2002). A 1973 article in *Infosystems,* the leading managerially oriented data processing publication, assured its readers that data base systems were akin to 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 to be stored, indexed and sorted with live data," they were now poised to rise majestically into the air (Romberg, 1973, p. 56).

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 MIS. 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

suppliers IBM, RCA, and Burroughs (Hare, 1971). The IBM user groups SHARE and GUIDE went so far as to produce a rival report of their own (Canning, 1972). 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. Continuing work to standardize a data definition language foundered on the reluctance of IBM to commit to the network concepts inherent in the CODASYL model, while its own flagship IMS product retained a hierarchical approach (Performance Development Corporation, 1980). 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 R. L. Flynn (1974) and J. P. Fry and E. H. Sibley (1976).

4 The DBTG standardized such terms as *record, 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 data base, 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 used to add, delete, update, and retrieve particular records from the Data Definition Language used to define the logical structure of the data base itself. While the Data Definition Language was to be a new and universally applicable language, the Data Manipulation Language took the form of a set of additions seamlessly integrated into an existing programming language.

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. DBMS functioned almost as a synonym for MIS. Richard Nolan—a professor at the Harvard Business School, a 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" (Nolan, 1973, pp. 101, 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" (Nolan, 1974, p. 27). Many had simply seized on "data base" as a new and more palatable name for this "total" MIS.

As with 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. As with 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 the 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 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" (Martin, 1977, p. 22). 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" (Gorry & Morton, 1974).

These quibbles did not stop hopeful accounts of the data base as a technological marvel that 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 data base administrator 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 (Lyon, 1971, p. 12). Nolan was still bolder: he believed that the data base administrator 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" (1974, p. 39). A consultant wrote that the data base administrator should be "something of a superstar" (Luke, 1975, p. 9).

Discussion of the data base administrator makes the rift between managerially oriented utopians and programmer-oriented pragmatists particularly apparent. Richard Schubert, who at the chemical firm B.F. Goodrich had overseen a remarkably ambitious in-house DBMS development project, noted simply 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" (1972, p. 47). 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 data base administrator 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 approach 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" (Curtice, 1974, p. 102). But, as with the data base administrator, the data dictionary slipped from the managerial into the technical: after the term was applied to scores of software products in the late 1970s (Canning, 1974), it came simply to describe that portion of the DBMS where Data Manipulation Language (see note 4) definitions were kept.

One IBM advocate of the data-dictionary approach likened data to money: "Once 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" (Cahill, 1970, p. 23). Nolan made a similar pitch in his book *Managing the Data Resource Function* (1974), the title of which suggested that information, like people and money, was a vital resource of business and therefore deserved similar managerial attention. Indeed, the claims made by Nolan that the data base administrator 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 officer, or CIO, in the 1980s (Synnott & Gruber, 1981).

Early Data Base Management Systems in Use

The DBMS enjoyed considerable practical success during the 1970s. By the end of the decade most large computer installations had installed one. DBMS and file management packages accounted for many of the most financially successful products of the independent software industry. Adoption of data base management software proved 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 and 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 uniformly structured, hierarchical data of the kind found on administrative forms.[5]

Yet the DBMS never quite lived up to the expectations of such people as Nolan, who saw it as a managerial panacea. 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 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 (Cuozzo & Kurtz, 1973). Most of the true costs were hidden, particularly the staff requirements. As they put it: "Some DBMS's 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" (p. 74). Later reports suggest that these problems continued for several years and that many firms installed DBMSs because of a "bandwagon" effect rather than a careful and informed evaluation (Schussel, 1975). 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 in which the data base was used directly by managers, or even by analysts (Nolan, 1973, p. 113).

Companies keen to get their hands on a DBMS had to go to considerable lengths. Schubert, at B.F. Goodrich, had been part of the DBTG and led his company into implementing its own system, the IDMS (Integrated Data Management System), based on a stripped-down version of the CODASYL proposals. It was used to support such batch-mode applications as billing and accounting as well as on-line access to order entry and its inventory of finished goods (Huhn, 1974). In 1973 Goodrich sold the rights to IDMS to John Cullinane, a marketing-savvy entrepreneur who by the early 1980s had built one of the era's largest and fastest-growing software companies around it (McClellan, 1984, pp. 242–246). Few companies were prepared to go this far to get a DBMS, and

[5] As with file management systems before them, the new systems still demanded that each record in a file (or "record type") include the same fields, each populated with data in the same format. In addition, because relationships (or "sets") were specified in the Data Definition Language and so built into the data base, the data base designer was forced to specify a complete and coherent set of links 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.

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 acquiring the most advanced DBMS packages, on-line use was limited and managerial applications rare. Two examples of firms using commercially supplied DBMSs in the early 1970s are McDonnell Douglas and Science Dynamics Corporation. McDonnell Douglas, using IBM's IMS system, claimed to have created a centralized data base containing all the information previously stored in 264 files that covered such things as spare parts, production scheduling, bills of materials handling, and inventory management. This data base made it much easier to change the ninety-five existing programs that relied on these files, to set up automatic cross-references between different records, and, the company hoped, to move toward on-line operation in the future (Hollenbach, 1973). Science Dynamics, a small firm devoted to accounts receivable processing for doctors, was using a DBTG-influenced DBMS on its Xerox computer to lower its daily processing times for updates and to design its new program more rapidly. The company was happy, despite the DBMS consuming a large part of the computer's memory and using 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 when the system was off-line in the belief that this "greatly reduces the possibility of a catastrophic loss of data" (Blanchard, 1974, p. 63).

According to a 1975 survey of large industrial firms, about one-third were using some kind of advanced file management system (Powers, 1975). Of that third, around half were using systems intended for direct ad-hoc querying by nonprogrammers, such as Mark IV, and half were using systems designed to integrate with such conventional programming languages 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, although 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 (Cheney & Lyons, 1980). 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" (p. 28).

Even products designed explicitly for use by nonspecialists found their main markets to be among data processing specialists. Because file management systems cost less and could run on more modest hardware, they remained more widely used than fully fledged DBMSs. The 1975 survey by Powers found that 41 percent 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 percent of firms) with files stored on tape rather than on disk. These systems still worked with individual files rather than vast integrated data bases: 55 percent of their users had not even begun to integrate files to remove redundant information.

During the 1970s the Mark IV file management system marketed by Informatics Inc. 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. When compared with 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 nonprogrammers. 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 (Forman, 1984, pp. 9–26). The proceedings of this group suggest that nonspecialists found advanced work harder than had been expected. According to the Eastern Airlines representative, although most of its two hundred users were "a complete new breed of coders . . . non-programmers, [with] 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" (Mark IV User Group, 1971, Appendix F).

While there was substantial demand for products that would let nonspecialists produce computerized reports without the assistance of programmers, the leading DBMSs did not do a good job of meeting it. 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.

By the end of the 1970s 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 DBMSs were now common in large corporations, few were being used to support new kinds of managerial application. Even the most sophisticated DBMSs 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.

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 called 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 such better established areas as numerical analysis, compiler theory, and the theory of algorithms. The event is better remembered, however, for Bachman's speech (1973). "The Programmer as Navigator" 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 effect of generalized DBMS would be much greater for programmers than for managers.

The acceptance of the DBTG concept of a DBMS 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 DBMS. As such, the commercial success of DBMS packages supported the growing prestige of corporate computing staff, against attempts by information scientists and documentationalists (Aspray, 1999) 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 such elaborate military systems of the 1960s as SDC's TDMS—on-line access, flexibly structured data, and 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. Attempts by Informatics to sell RECON IV—an information-retrieval product developed under contract for NASA—as a commercial package that would let firms build their own "data bases of massive amounts of information in natural language form" yielded no more than three or four sales (Forman, 1984, chap. 5, pp. 16–21, and chap. 11, pp. 19–21). Although the industrial research budgets of leading corporations might have paid for subscriptions to the newly available on-line scientific data bases of the 1970s (Hahn, 1996), the managers and computing departments of the same companies had little interest in using these technologies to manage their own information.

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 (Chamberlin, 1976; Codd, 1970), gradually gained acceptance among data base 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-purpose data bases shared between different departments and computer systems. The relational model has also been praised for its nonprocedural nature, which further separated the user from the physical storage mechanisms involved (Michaels, Mittman, & Carson, 1976). This simplified programming and insulated application code from changes in the data base structure. Winning his own Turing award in 1981, E. F. Codd, the originator of the relational model, suggested that the CODASYL network model had forced the programmer to become too much of a navigator, at too low a logical level (Codd, 1981).

Use of early DBMSs 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 (Pansophic Systems Incorporated, 1981). 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 such earlier mainframe products 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 DBMSs 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 aims to use the next major revision of Windows to replace the conventional file system and the e-mail repositories found on today's computer systems with a multitalented DBMS. In some ways the DBMS has indeed become a universal container for computer data.

Conclusions

The data base management system provides an interesting example of the tensions hidden behind such phrases 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 such words as *SELECT* would make it easy for managers to write their own queries (McJones, 1997), 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 such things as payroll administration, because each record included the same pieces of data (e.g., years of service, Social Security number, hourly rate, and overtime status). It made it very simple and efficient to update information and so is well suited to administrative systems where records are constantly updated. However, it was entirely useless for representing and searching less rigidly formatted data, such as full-text records, correspondence, or even scientific abstracts. 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. Such systems as AltaVista and, more recently, Google have proved remarkably adept at returning relevant results from a sea of unstructured data.

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 were 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. The corporate data pools imagined forty years ago have inched ever closer to reality. Whether this will ever lead to a fundamental change in the way management works, or make businesses treat information with the same respect as money, remains to be seen.

References

Akera, A. (2001). Voluntarism and the fruits of collaboration. *Technology and Culture, 42,* 710–736.

Anonymous. (1964a). A panel discussion on time-sharing. *Datamation, 10*(11), 38–44.

Anonymous (Ed.). (1964b). *Disc file applications: Reports presented at the nation's first disc file symposium*. Detroit: American Data Processing.

Aspray, W. (1999). 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), 4–20.

Bachman, C. W. (1973). The programmer as navigator. *Communications of the ACM, 16*, 653–658.

Bashe, C. J., Johnson, L. R., Palmer, J. H., & Pugh, E. W. (1986). *IBM's early computers*. Cambridge, MA: MIT Press.

Baum, C. (1981). *The system builders: The story of SDC*. Santa Monica, CA: System Development Corporation.

Bello, F. (1953, December). The information theory. *Fortune, 48(6),* 136–141, 149–150, 152, 154, 156, 158.

Bello, F. (1960). How to cope with information. *Fortune, 62*(3), 162–167, 180–182, 187–189, 192.

Blackman, K. R. (1998). IMS celebrates thirty years as an IBM product. *IBM Technical Journal, 37,* 596–603.

Blanchard, J. S. (1974). We bet our company on data base management. *Datamation, 20*(9), 61–65.

Bowles, M. D. (1999). The information wars: The two cultures and the conflict in information retrieval, 1945–1999. In M. E. Bowden, T. B. Hahn, & R. V. Williams (Eds.), *Proceedings of the 1998 Conference on the History and Heritage of Scientific Information Systems* (pp. 156–166). Medford, NJ: Information Today.

Bryant, J. H., & Semple, P. (1966). GIS and file management. In Association for Computing Machinery (Ed.), *Proceedings of the 21st National Conference* (pp. 97–107). New York: Association for Computing Machinery.

Byrnes, C. J., & Steig, D. B. (1969). File management systems: A current summary. *Datamation, 15*(11), 138–142.

Cahill, J. J. (1970). A dictionary/directory method for building a common MIS data base. *Journal of Systems Management, 21*(11), 23–29.

Canning, R. G. (1966). New views on mass storage. *EDP Analyzer, 4*(2).

Canning, R. G. (1967). Data management: File organization. *EDP Analyzer, 5*(12).

Canning, R. G. (1972). The debate on data base management. *EDP Analyzer, 10*(3).

Canning, R. G. (1974). The data dictionary/directory function. *EDP Analyzer, 12*(10).

Chamberlin, D. M. (1976). Relational data-base management systems. *ACM Computing Surveys, 8*(1), 43–66.

Cheney, P. H., & Lyons, N. R. (1980). MIS update. *Data Management, 18*(10), 26–32.

CODASYL Data Description Language Committee. (1974). *CODASYL data description language: Journal of development, June 1973.* Washington, DC: U.S. Government Printing Office.

CODASYL Systems Committee. (1971a). *CODASYL Data Base Task Group: April 1971 report*. New York: Association for Computing Machinery.

CODASYL Systems Committee. (1971b). *Feature analysis of generalized data base management systems*. New York: [n.p., Distributed by the Association for Computing Machinery].

Codd, E. F. (1970). A relational model for large shared databanks. *Communications of the ACM, 13,* 277–390.

Codd, E. F. (1981). Relational database: A practical foundation for productivity. *Communications of the ACM, 25,* 109–117.

Copeland, D. G., Mason, R. O., & McKenney, J. L. (1995).

SABRE: The development of information-based competence and execution of information-based competition. *IEEE Annals of the History of Computing, 17*(3), 30–57.

Cuozzo, D. E., & Kurtz, J. F. (1973). Building a base for data base: A management perspective. *Datamation, 19*(10), 71–76.

Curtice, R. M. (1974). Some tools for data base development. *Datamation, 20*(7), 102–106.

Dearden, J. (1965). How to organize information systems. *Harvard Business Review, 43*(2), 65–73.

Edwards, P. (1996). *The closed world: Computers and the politics of discourse in Cold War America*. Cambridge, MA: MIT Press.

Elmasri, R., & Navathe, S. B. (1989). *Fundamentals of database management systems*. New York: Benjamin/Cummings.

Flynn, R. L. (1974). A brief history of data base management. *Datamation, 20*(8), 71–77.

Forman, R. L. (1984). *Fulfilling the computer's promise: The history of informatics, 1962–1982*. Woodland Hills, CA: Informatics General Corporation.

Fry, J. P., & Sibley, E. H. (1976). Evolution of data-base management systems. *ACM Computing Surveys, 8*(1), 7–42.

Gorry, G. A., & Morton, M. S. S. (1974). A framework for management information systems. In R. L. Nolan (Ed.), *Managing the data resource function* (pp. 87–105). St. Paul, MN: West Publishing.

Hahn, T. B. (1996). Pioneers of the online age. *Information Processing and Management, 32*(1), 33–48.

Haigh, T. (2001a). The chromium-plated tabulator: Institutionalizing an electronic revolution, 1954–1958. *IEEE Annals of the History of Computing, 23*(4), 75–104.

Haigh, T. (2001b). Inventing information systems: The systems men and the computer, 1950–1968. *Business History Review, 75*(1), 15–61.

Haigh, T. (2002). Software in the 1960s as concept, service, and product. *IEEE Annals of the History of Computing, 24*(1), 5–13.

Hare, V. C., Jr. (1971). A special report on the SIGBDP forum: "The new Data Base Task Group report." *Data Base, 3*(3), 1.

Haslett, J. W. (1962). Total systems: A concept of procedural relationships in information processing. In A. D. Meacham & V. B. Thompson (Eds.), *Total systems* (pp. 16–19). Detroit, MI: American Data Processing.

Head, R. V. (1965). Data base symposium. *Datamation, 11*(11), 41.

Head, R. V. (1967). Management information systems: A critical appraisal. *Datamation, 13*(5), 22–27.

Head, R. V. (1970). MIS-II: Structuring the data base. *Journal of Systems Management, 21*(9), 37–38.

Hollenbach, R. (1973). An application of a data base system. *Data Management, 11*(9), 30–31.

Hughes, T. P. (1998). *Rescuing Prometheus*. New York: Pantheon Books.

Huhn, G. E. (1974). The data base in a critical on-line business environment. *Datamation, 20*(8), 52–56.

Keller, A. E. (1962). The man behind systems at Shell Oil. *Business Automation, 7*(2), 20–24.

Kircher, P. (1969). Breakthrough in management information systems. *Journal of Data Management, 7*(2), 28–31.

Leavitt, H. J., & Whisler, T. L. (1958). Management in the 1980s. *Harvard Business Review, 36*(6), 41–48.

Luke, J. W. (1975). Data base systems: Putting management back in the picture. *CSC Report, 9*(1), 8–12.

Lyon, J. K. (1971). The role of the data base administrator. *Data Base, 3*(4), 11–12.

Mark IV User Group. (1971). *Proceedings of the MARK IV User Group: Meeting IX, Jan. 25–27, 1971, San Francisco, CA*. In the Evan Linick Collection of Proceedings of the Mark IV User Group (CBI 130), Charles Babbage Institute, University of Minnesota, Minneapolis.

Martin, J. (1977). *Computer data-base organization*. Englewood Cliffs, NJ: Prentice-Hall.

McClellan, S. T. (1984). *The coming computer industry shakeout: Winners, losers, and survivors*. New York: John Wiley & Sons.

McCracken, D. D., Weiss, H., & Lee, T.-h. (1959). *Programming business computers*. New York: John Wiley & Sons.

McGee, R. C., & Tellier, H. (1960). A re-evaluation of generalization. *Datamation, 6*(4), 25–29.

McGee, W. C. (1959). Generalization: Key to successful electronic data processing. *Journal of the Association for Computing Machinery, 6*(1), 1–23.

McJones, P. (1997, August 20). *The 1995 SQL Reunion: People, projects and politics*. Available: http://www.mcjones.org/System_R/SQL_Reunion_95/sqlr95-System.html (accessed 12 February 2000).

Michaels, A. S., Mittman, B., & Carson, C. R. (1976). A comparison of the relational and CODASYL approaches to data-base management. *ACM Computing Surveys, 8*(1), 125–151.

Miller, E. (1961). Information retrieval—1961. *Datamation, 7*(10), 19–21.

National Research Council. (1999). *Funding a revolution: Government support for computing research*. Washington, DC: National Academy Press.

Nolan, R. L. (1973). Computer data bases: The future is now. *Harvard Business Review, 51*(5), 98–114.

Nolan, R. L. (Ed.). (1974). *Managing the data resource function*. New York: West Publishing.

Olle, T. W. (1972). Recent CODASYL reports on data base management. In R. Rustin (Ed.), *Data base systems* (pp. 175–184). Englewood Cliffs, NJ: Prentice-Hall.

Osborn, R. F. (1954). GE and UNIVAC: Harnessing the high-speed computer. *Harvard Business Review, 32*(4), 99–107.

Pansophic Systems Incorporated. (1981). *Competitive products survey: Data base management systems software products*. Unpublished report, author's collection.

Parker, R. W. (1965). The SABRE system. *Datamation, 11*(9), 49–52.

Performance Development Corporation. (1980). *An interview with Charles W. Bachman (Part II), Data Base Newsletter, Volume 8, No. 5*. In the Charles W. Bachman Papers (CBI 125), Charles Babbage Institute, University of Minnesota, Minneapolis.

Postley, J. A. (1998). Mark IV: Evolution of the software product, a memoir. *IEEE Annals of the History of Computing, 20*(1), 43–50.

Postley, J. A., & Jackabson, H. (1966). The third generation computer language: Parameters do the programming job. In Data Processing Management Association (Ed.), *Data processing* (Vol. 11, pp. 408–415). Los Angeles: Data Processing Management Association.

Powers, V. (1975). Implementing generalized data base management systems. *Data Management, 13*(5), 14–16.

Pugh, E. W., Johnson, L. R., & Palmer, J. H. (1991). *IBM's 360 and early 370 systems*. Cambridge, MA: MIT Press.

Romberg, B. W. (1973). Data bases: There really is a better way to manage your files. *Infosystems, 20*(5), 56–57, 58.

Rowan, T. C. (1958). The recruiting and training of programmers. *Datamation, 4*(3), 16–18.

Schubert, R. F. (1972). Basic concepts in data base management. *Datamation, 18*(7), 42–47.

Schussel, G. (1975). When not to use a data base. *Datamation, 21*(11), 82, 91, 98.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana: University of Illinois Press.

Simon, L., & Sisson, R. (1966). Evolution of a total system. *Total Systems Letter, 1*(11), 1–4.

Statland, N., & Hillegass, J. R. (1963). Random access storage devices. *Datamation, 9*(12), 34–45.

Steig, D. B. (1972). File management systems revisited. *Datamation, 18*(10), 48–51.

Stone, M. D. (1960). 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* (pp. 14–22). New York: American Management Association, Finance Division.

Synnott, W. R., & Gruber, W. H. (1981). *Information resource management: Opportunities and strategies for the 1980s*. New York: John Wiley & Sons.

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

Vorhaus, A. H. (1967). TDMS: A new approach to data management. *Systems & Procedures Journal, 18*(4), 32–35.

Waites, W. G. (1971). MIS or IMS? *Journal of Systems Management, 22*(1), 32–34.

Webster, E., & Statland, N. (1962). Instant data processing. *Business Automation, 7*(6), 34–36, 38.

Weindling, R. E. (1961). Office will run every business activity. *Office Management and American Business, 22*(1), 12–15.

Welke, L. (1972). A review of file management systems. *Datamation, 18*(10), 52–54.

Wellisch, H. (1972). From information science to informatics: A terminological investigation. *Journal of Librarianship, 4*(3), 157–187.

Wendler, C. C. (1966). What are the earmarks of effective total systems? *Systems & Procedures Journal, 17*(4), 29–31.