The Major Tasks of Data Processing

W. N. Holmes*

(as published in *The Australian Computer Journal*
Vol.9, No.1, March 1977, pp.32–38)

This paper reviews the historical development of data processing, discerning three approximate decades of distinct evolutionary cycles. Each cycle is seen as forking, two contrasting styles of data processing forming and separating during the cycle. On this basis, a classification of the major general areas of data processing is suggested.

For each decade, characteristic aspects are discussed and both the lines of development are described. Finally, some observations regarding the present decade and its requirements are given, and some predictions relating to the next decade and its prerequisites are made.

DESCRIPTORS: Classification of data processing. Evolution of data processing. Philosophical implications. Computing milieu. Terminology.

CR CATEGORIES: 1.2, 1.3, 2.1

INTRODUCTION

Electronic digital computers have now been in use for some thirty odd years. During this time, many machines and programs have been developed. This development is usually seen as divided into stages according to the nature of the basic electronic components used in the manufacture of the computers—relays, vacuum tubes, transistors, and, lately, integrated circuits. Such an analysis is very superficial—it is the development of data processing which should be studied, not the development of manufacturing techniques for electronic digital computers.

Data processing is sufficiently developed and ramified to allow analysis in terms of what it does, rather than what it uses. This paper presents such an analysis, describing five phases—three past, one present, and one future. The phases are described as distinct and regular, so that they may be easily appreciated. Of course, they are in fact indistinct and irregular. In some countries the first phase has not yet arrived. Even in developed countries, many large computer users are operating still in early phases. Certain developments depicted here as sudden were in fact gradual. Certain developments predicted here have been slowly, almost imperceptibly, building up for some time.

Nevertheless, the analysis is broadly true to history, at least for the phases past. The significance of such an analysis comes from the possibility of appreciating better where data processing stands now, and whither it might be going.

THE PATTERN OF DEVELOPMENT

All development phases except the first can be taken as approximating decades, and will be referred to as such. During each decade, the data processing tasks of previous decades, and particularly the dominant tasks of the immediately preceding decade, are still undertaken, and are undertaken more effectively under the new circumstances. In particular, and because the new procedures and machinery

* *IBM Australia Limited 80 Northbourne Avenue Braddon ACT 2601.* *Manuscript received 19th November 1976, revised version, 17th January 1977.*

were designed to this end, the salient new developing area of the preceding decade is sanctified—given a huge and modish boost, crystallised and exploited.

Additionally, and quite inconspicuously, during each decade new ideas and approaches, one of which will form the main push of the succeeding decade, begin to emerge.

In this paper, consideration of each decade will include a review of the characteristics of the data, the programs, the machinery, and the applications dominant in that decade. This review will be followed by a description of contrasting data processing tasks seen as developing during the decade.

THE ARCHAIC, PUNCHED CARD, PERIOD

The archaic period was characterised by manual computing procedures, based on mainly numeric data, procedures which were arithmetically, and sometimes logically, assisted by mechanical or electromechanical devices. These devices were either of quite fixed purpose with minor adjustment effected by switch or lever, or were quite adaptable within a class of functions through exchange of program effected usually by swapping plugged panels.

Scientific calculations were assisted by mechanical calculators whose operands were set by fingers, and whose results were generated by hand-driven crank and laboriously written down for transfer to other computation.

Commercial calculations were in their first stages effected by the recording of data on punch cards, which recording required flocks of clerical personnel and heaps of filled-out forms. Punched cards in their millions were fed into machine after machine for prolonged cycles of sorting, collating, and tabulating. The work to be done was programmed in terms of steps in a manual procedure, including the setting up of machines often by changing their wiring through exchanging plugged panels mounted inside the machines.

This period lasted many decades and the development of the machinery was slow by present day standards. The techniques of this decade may seem drear and clumsy in retrospect, but were in fact cheap and effective compared to other then-possible techniques.

DECADE I - THE STORED PROGRAM

The first decade, approximately the 1950's, began when the stored program computer found use outside the development laboratory, within commercial punched card and paper tape procedures. The stored program computer was intended as an arithmetic instrument for scientific computation, and its eventual commercial exploitation was unanticipated.

The significant aspect of the stored program machinery was that it allowed the plugged panel to be replaced by a deck of cards or a roll of paper tape.

The plugged panel was cumbersome, unreliable, and difficult to copy and record. The stored program allowed procedures to be easily exchanged or transferred, and to be deliberately and extensively developed.

The characteristics of the two branches of data processing in this decade are sketched in the following table.

Number Processing	**File Processing**
Data	
Few input data	At least one large uniform file of data
Tabular or graphic output data	Listed, columnar output with (sub(sub . . .)) totals
Output quite different from input	Output similar to input
Programs	
Phased - input, calculation, output	Cyclic - record by record
	Much checking, validation

Standard subroutines for math. functions	Standard subroutines for input/output control
Operations	
Usually by the users or researchers themselves	Hosts of data preparation and control staff
Most data kept are programs in files	Millions of data cards kept
Machinery	
Binary arithmetic	Decimal arithmetic
Example Applications	
Spectrography, Astronomical prediction	Payroll, Invoicing

Characteristics

The computer installation of the first decade contained a variety of equipment, and, except for the computer itself, this equipment was inherited, in spirit if not in body, from the archaic era. Mostly the peripheral equipment was used to prepare input data for feeding to the computer, and for listing or otherwise cleaning up the results of the computation for consumption outside the installation.

The computer, often little more than an electronic calculator, was mainly used to punch cards or paper tape. In commercial work, the result of the computation might merely have been half a dozen new holes in each card read in. The programmer himself often operated the computer, and in any case the computer was devoted at any time entirely to the execution of one program, and program executions were long, usually for at least half an hour.

Programs for the computer were partly or entirely set up as wires in a plug board which was physically inserted in the computer to set it up for that particular program. When some of the program was kept in the computer's main store, that program was written in a programming language very closely related to the actual form of the program as it was stored in the computer. Programs were debugged by the programmer using the physical controls of the machine itself to delve into the details of his program's functioning. Only when his scheduled time had elapsed could the programmer be persuaded to leave his machine, and much debugging was done over weekends.

Developments

The salient development of the first decade was the introduction of magnetic storage devices, principally the magnetic tape drive. The large files of punched cards were slowly replaced by reels of magnetic tape, at least where the punched card was not used as a document of record.

In commercial installations the file update program began to reign supreme, and record designs, at first constructed in the image of their punched card predecessors, began to be expanded and embellished.

The two lines of development during the Stored Program Decade were continued from the Archaic Period—*scientific use* featuring heavy computation ("number crunching") was encouraged by computing machinery designed for that purpose, while commercial use was developed from punched card procedures.

DECADE II - THE OPERATING SYSTEM

The second decade, roughly the 1960s, was founded on a desire, natural enough, to get many more enterprises to use computers. The decade was fashioned by the need to have these computers used more effectively.

Although number processing continued to be important, the salient developments of the second decade sprang from improvements in file processing.

The main features of the two major tasks of the second data processing decade, the consolidating task of job processing and the developing tasks of transaction processing, are outlined in the following table.

Job Processing	Transaction Processing
Data	
Serial transcription	Loading and updating
Linear and uniform	Indexed and uniform
Batched and sorted input	Unsequenced input
Programs	
Large and monolithic	Small and systemic
Periodic and cyclic	Sporadic and episodic
Operations	
Central team	Distributed terminals
Carefully scheduled	Second shift housekeeping
Regular peaks	Reliability emphasised
Machinery	
Magnetic tape based	Single spindle fixed disc
Example Applications	
Accounts payable	Inventory control

Characteristics

The computing systems of the second decade typically had substantial magnetic file processing capability, and this came to be used to provide operating systems. These operating systems provided access to a library of programs, and allowed often small decks of cards, each representing a request to have a program run, to be stacked—run one after the other with a minimum of operator attention and computer idle time, particularly in the event of program malfunction. This style of operation is called *job processing*, and the queues of program run requests are called *job stacks*.

Programs were much more numerous and came to be written in ”high level” languages. Compilers for these languages and the operating systems themselves provided some small assistance for programmers in the way of program traces and data dumps, so that programmers could be denied physical access to the computer and would, it was hoped, get their programs going more efficiently, and allow more production work to be pushed through.

The computing machinery was often adaptable to both scientific and commercial work, but programs for the two types of work were written in different programming languages.

Early direct access storage was typically a fixed magnetic disc file of a million characters or so capacity, and this file was used to store the principal business file so that sporadic enquiries could be promptly handled during the day. Some direct access systems automatically preempted ordinary data processing while the enquiry was being handled, but typically these systems had one single-purpose enquiry terminal, and the enquiry file was updated each evening. In many cases the enquiry file merely contained an identification and a status—for example a part number and the number of parts in stock—while the complete data were maintained on magnetic tape.

Developments

The most significant development of the second decade was the introduction of direct access storage. To a large degree, general adoption of this type of storage was forced by the many inconvenient aspects of operating systems based on magnetic tape.

Increased speed of the computers relative to the input and output devices, and the elaboration of enquiries and updates to an increasing number of direct access files, both led to the development of multiprogramming operating systems.

Computers operating under these systems were able to run two or more, often many more, independent programs concurrently, a mode of operation called, unfortunately and clumsily, *multiprogramming*.

The most popular form of multiprogramming ran a production job stack concurrently with a program development job stack.

While the second decade was dominated by the adoption of operating systems, two distinct purposes were thereby served, and this gave rise to the two different lines of development of the decade.

On the one hand, operating systems gave sophisticated, diverse and effective support to the file processing of the preceding decade—reducing operator intervention, improving data transcription efficiency, allowing easy transition from program to program in a suite, and providing for compilation and listing of programs without disrupting production runs. This line of development, reaching its peak in the second decade, concentrated on the task of job processing.

On the other hand, operating systems provided for the gross management of numerous data files, including those kept on magnetic discs, and gave support to methods of access to the records of those files which were completely impractical with records kept on magnetic tape—methods of access which allowed operations on individual records in any sequence whatsoever or not at all. This line of development was the trail breaker of the second decade, and emphasised the task of transaction processing.

DECADE III - THE DATA MANAGEMENT SYSTEM

The third decade, roughly the 1970s, is founded on the objectives of providing a multiplicity of terminal users with certain standard services, and of reducing the need for terminal operators monotonously and repetitively to collect data.

Although job processing continues to be an important task, the emphasis and consolidation are shifting entirely to transaction processing based on formatted data—a task here called *record* processing, to contrast it with the new and promising major task of the third decade—transaction processing based on unformatted data—here called *text* processing.

In the following table, the characteristics of the task of record processing, and of its possible offshoot of text processing, are suggested.

Record Processing	**Text Processing**
Data	
Small unit records	Large amorphous records
Identifiers, values and links	Unstructured data
Unique identifiers	Approximate identifiers
Functional identifiers	Trivial identifiers
Change by content	Change by accumulation and obsolescence
Programs	
Data dictionaries	Inverted files
Interpreted	Utilities
Housekeeping and selection	Search and rearrangement
Operation	
Clerical	Direct use
Repetitive task	Extemporaneous task
Example Applications	
Management information system	”Information” retrieval
Airline reservations	Document processing

Characteristics

The third decade sees the evolution of the operating system from a set of program and peripheral device management services, to an integrated and complex system taking complete responsibility for all data available to, or passing through, the computing system.

This responsibility will, at least towards the end of the decade, extend to the content of the records—

the representation used for the data and the links between records. In the second decade the majority of the data processed were organised in files which were physically supplied to, and then removed from, the computing system. In the third decade, because of the increased capacity and decreased cost of direct access storage devices, the bulk of current data storage is permanently attached to the computing system, and is coming to be managed as a single entity.

The data management system of the third decade, apart from merely coordinating access to all data, also mediates access to the data to provide services such as checking authority, assuring integrity, and providing insensitivity to changes in format of the data.

The facilities of the data management system are both necessary to, and promoted by, the supply of appropriate services to individual users at terminals. To the automatic data management system, the coordination of services to concurrent users is tremendously complicated. To the human designers and implementors of application programs, this coordination by the data management system will allow complete processing of individual transactions by their application program without any need to consider the effect of other transactions which might be presented by individual users at the same time. On the other hand, application programs can ask for data by name, and the data management system will use a data definition dictionary to resolve which of its data are meant. In this way, programs will continue to function correctly despite alterations in actual record design.

Thus, the data coordination provides support which is very simple to use, support which provides completely up-to-the-minute data for the user at the terminal.

Another aspect of data coordination has become important during this decade as a corollary to the coordination of transaction messages as seen by the application programmer. The coordination of the physical and electronic activity of terminals can be a very onerous duty for the data management system. Although the terminals may have wildly different and even varying characteristics, the data management system must cope with them, ideally without any consequences for the application program, and must also cope with differing telecommunications network conventions and with transaction messages which drift in in bits and pieces, from far and near, some fast and some slow.

The role of the programmer in data processing is being sustained for the time being. As a kind of counterpoint to the provision of data management services, certain special services are provided for the programmer at the terminal. Files of program code and test data are manipulated by transcribing and editing utility programs. Programs under development are interpretively executed so that their operation may be investigated and modified by the programmer in terms of the language used to write the program in the first place.

Developments

The main development of the third decade is based on the processing of non-numeric data. In previous decades non-numeric data have been widely used, at least in commercial data processing, but this use has been either incestuous—in translation from one programming language to another, or nominal—as identifying data accompanying quantitive data.

The decreasing cost of direct access storage devices, the increasing speed of computation, and the widenlng use of terminals, is forcing the processing of textual descriptive data, for their own sake.

Now that the majority of numeric and identifying data in an enterprise is being stored in the electronic data processing system itself, attention is turning to the use of descriptive data, non-numeric and textual.

Text processing in the third decade mostly takes the quite passive form of ”information retrieval” whereby that somewhat arbitrary linguistic unit, the ”word”, is the basic unit handled. Typically, a large file of bibliographic data, including for instance abstracts of documents, is loaded, set up, to be retrieved by reference to words in the bibliographic data through use of a very large index file which points to each word in the data file. This usually large combination of data file and index file is called an ”inverted” file.

Another form of text processing finding favour nowadays might best be called ”document processing”. Through a terminal the rough text of a document such as a letter or a report can be entered or altered, and a formatted version of the document (with for example line justification and pagination automatically carried out) can be printed as and when required by the data processing system.

DECADE IV - THE TEXT MANIPULATION SYSTEM

The fourth decade, roughly the 1980s will strive to provide a range of useful services to *all* employees of enterprises depending on computers. The developments of the preceding decade will be consolidated by the greater use of increasingiy automatic machinery to collect data and to act on them, and by the increasing capacity of machinery to store and transmit them.

Although record processing will continue to be an important task, the emphasis and development will shift to text processing, and textual, graphic, and even facsimile data communication, will come into its own. In countries using the Latin alphabet, upper and lower case textual representation will become usual, and this will increase the problems of spelling and ambiguity.

In terms of the major data processing tasks to be expected, the passive task of enquiry processing will consolidate during the fourth decade, while the active and manifold task of request processing should be the main task for development. The following table will help to contrast the styles of these two major tasks.

Enquiry Processing	**Request Processing**
Data	
Mainly from on-line storage	Mainly from the terminal
Programs	
Insensitive to input data	Sensitive to input data
Concentrating on simplifying	Adaptive
the human interface	Richness of capability
Static and highly modular	Interpreted, multilingual
Operations	
Natural language or brief formal	Iterative and cumulative
question and answer	Problem solving
Machinery	
Special purpose terminals	Large capacity displays
Example Applications	
Management services	Decision making
Text file searches	Business simulation

Characteristics

The fourth, text manipulation system, decade, will see the extension and elaboration of data processing services throughout large enterprises and organlsations, and will see everyday exchange of services between data processing systems despite the many administrative and ethical hazards already being pointed out. The pervasiveness of data processing services will be encouraged by the cheapness of data storage and transmission, and discouraged by the cost of people to provide, supervise, and use the services.

The services will be based on the massive storage of all data collectable about the organisation and its interactions with other entities, including the customary numeric or financial data, and, later, voluminous informal data including, for instance, all internal and external correspondence, much of which may be electronically forwarded and may never be actually printed. All these data will be continually updated as fast as the changed data can be collected, and increasingly specialised machinery will be put to collecting data.

The big problem for data processing installations will be that of exploiting the huge quantities of data stored.

A first solution will lie in the provision of limited but versatile and easy-to-use services to employees and customers. These services will typically concentrate on extracting and refining current data—relatively simple and predominantly numeric services, or will concentrate on locating and displaying selected retrospective data—highly conditioned and mostly textual services. The obstacle to this solution is the difficulty of designing the services so they are easy to use for everybody who needs them.

A second solution will lie in the provision of a supporting system which allows a skilled user to put together his own service as he needs it. The formidable obstacle to this solution is the problem of designing the system so that the task of learning to use it is not too demanding, but so that the task of exploiting it for significant tasks is not too demanding either.

Developments

The restrictions and difficulties experienced with single-purpose terminal procedures are already beginning to force the development of procedures which present the data about the transactions being processed in common language typically as a ”page” of text on a display screen. Towards the end of the fourth decade of data processing, the user will be able to control the service he is using almost in any way he likes. This control will be made possible by, in the first place, linguistic analysis of his common language requests and responses, and in the second place by lexicographical inference from the data referred to.

The point is that possibilities for use of the available computing systems will not be stereotyped, and the scope for manipulation of data will be greatly extended, firstly by the increasing ability to transmit data between systems, and secondly by the potentiality of microcomputers in allowing a variety of devices for collection and use of data to be easily and cheaply connected to large computing systems.

The contrast which will evolve will be between the consolidating major task inherited from the preceding data processing decade and here called *enquiry processing,* and the blossoming new major task here called *request processing.* The emphasis of enquiry processing will be on the purposeful streamlining of routine and stereotyped services to increase their effectiveness in day-to-day operations of the enterprise. The emphasis of request processing will be on the impromptu extension of multifarious improvised services to increase their effectiveness in creative and spontaneous applications.

REQUIREMENTS FOR TRANSITION FROM DECADE III TO DECADE IV

The data management decade features traditional record processing enhanced by a data management system which allows all numeric and identifying data in the system to be handled in a complete and coordinated way.

In record processing, the development of standards, both for the physical characteristics of terminal devices and for the structuring and manipulation of large collections of data (popularly called ”data banks” or ”data bases”) will make it easier to have practical data management systems introduced. Coincidentally, the belated adoption of formal programming procedures (variously under the popular names of ”structured programming”, ”top-down design”, ”chief programmer teams” . . .) is already assisting this trend.

In text processing, the data management systems used at present are extemporaneous and limited. The typical approach uses an inverted file which can only be changed in a gross or infrequent manner, mainly because an inverted file system which allows general updating as part of its normal repertoire is relatively complex and slow.

Nevertheless, text processing systems must be developed which provide updating as easily and conveniently as passive enquiry.

The handling of textual enquiries must also be improved. At present, retrieval of textual data through enquiry is hobbled by its complete dependence on the spelling of the text.

Many developments are possible in this area. The possible developments are probably best considered under two headings—pragmatic and analytic.

Pragmatic development will exploit the ”experience” available as the text data file is being used. Users searching the file (particularly when they are ”browsing”) will go from item to item, document to document. This procession allows related items or documents to be linked in a quantitative way, so that subsequent searchers can be assisted by the experience of prior searchers.

Analytic development exploits and will exploit linguistic or preset relationships between words to provide links between related documents, either by reconciling spelling differences (for instance through error, affix, or inflection) or by reference to a thesaurus. Ultimately, such systems will become multilingual.

A second important area which must be developed as a prerequisite for the text manipulation system decade is the document processing system. On the one hand, present systems must be developed to provide easy conversion of data to various graphic forms, starting with tables and charts and figures, and on the other hand the developed systems must be integrated with the text processing system so that the documents are readily and cleanly accessible to enquiry. Both these aspects depend on the clear recognition of the distinction between textual data and control data in a document file, and on the adoption of simple principles for design of such control data. In this kind of system, the control data is effectively a program which mingles with its own input data, and this distinct kind of programming should perhaps be distinctively called *queued programming*.

SUMMARY

The analysis of this paper is condensed in the table below.

The table is clearly at odds with the ”vacuum tube to integrated circuit” school of computing history, and the contrast should be instructive, even if the analysis is disputed.

To concentrate on how machinery is used rather than on how it is manufactured is surely more useful, and may lead to perhaps surprising observations. For instance, although the minicomputer is distinctive in the size and cost and performance conferred by its method of manufacture, its significance lies not nearly so much in allowing old large enterprises to undertake new complex tasks, as in its allowing new small enterprises to undertake old simple tasks.

Phase	**Main Task**	**Secondary Task**	**Machine Development**	**Objectives**
– Archaic	Tabulation	Computation	Plugged panel	Simplification
I Stored program	Commercial	Scientific	Magnetic storage	Automatic computation
II Operating System	Job processing	Transaction processing	Direct access storage	Machine productivity
III Data Management	Record processing	Text processing	Terminals	Concurrent services
IV Text Management	Enquiry processing	Request processing	Networks	Universal services

BIBLIOGRAPHY

This bibliography [*omitted here, but see the original publication*] collects articles and books relating to the history of data processing. The majority of these references do not present the same point of view as that proposed by this paper, but are brought together as a pool of readily available background reading, in particular for use in schools.