

# Applied Information Technology: from Data Processing to Strategic Information Systems

Elisabeth K. Somogyi, *PA Computers and Telecommunications, London*

Robert D. Galliers, *Western Australian Institute of Technology, Bentley, W.A.*

---

**Abstract:** This paper provides a brief historical overview of developments in the application of information technology. It highlights the key turning points and examines how the role of information systems has changed during the brief history of this new technology.

---

## Introduction

Computers have been used commercially for over three decades now, in business administration and for providing information. The original intentions, the focus of attention in (what was originally called) data processing and the nature of the data processing effort itself have changed considerably over this period. The very expression describing the activity has changed from the original 'data processing', through 'management information' to the more appropriate 'information processing'.

A great deal of effort has gone into the development of computer-based information systems since computers were first put to work automating clerical functions in commercial organizations. Although it is well known now that supporting businesses with formalized systems is not a task to be taken lightly, the realization of how best to achieve this aim was gradual. The change in views and approaches and the shift in the focus of attention have been caused partly by the rapid advancement in the relevant technology. But the changed attitudes that we experience today have also been caused by the good and bad experiences associated with using the technology of the day. In recent years two other factors have contributed to the general change in attitudes. As more coherent information was made available through the use of computers, the general level of awareness of information needs grew. At the same time the general economic trends, especially the rise in labour cost, combined with the favourable price trends of computer-related technology, appeared to have offered definite advantages in using computers and automated systems. Nevertheless this assumed potential of the technology has not always been realized.

This paper attempts to put into perspective the

various developments (how the technology itself changed, how we have gone about developing information systems, how we have organized information systems support services, how the role of systems has changed, etc.), and to identify trends and key turning points in the brief history of computing. Most importantly, it aims to clarify what has really happened, so that one is in a better position to understand this seemingly complex world of information technology and the developments in its application, and to see how it relates to our working lives. One word of warning, though. In trying to interpret events, it is possible that we might give the misleading impression that things developed smoothly. They most often did not. The trends we now perceive were most probably imperceptible to those involved at the time. To them the various developments might have appeared mostly as unconnected events which merely added to the complexity of information systems.

## The early days of data processing

Little if any commercial applications of computers existed in the early 1950s when computers first became available. The computer was hailed as a mammoth calculating machine, relevant to scientists and code-breakers. It was not until the second and third generation of computers appeared on the market that commercial computing and data processing emerged on a large scale. Early commercial computers were used mainly to automate the routine clerical work of large administrative departments. It was the economies of large-scale administrative processing that first attracted the attention of the system developers. The cost of early computers, and later the high cost of systems development, made any other type of application economically impossible, or very difficult to justify.

These first systems were batch systems using fairly limited input and output media, such as punched cards, paper-tape and printers. Using computers in this way was in itself a major achievement. The transfer of processing from unit record equipment such as cards allowed continuous batch-production runs on these expensive machines. This was sufficient economic justification and made the proposition of having a computer in the first place very viable indeed. Typical of the systems developed in this era were payroll and general ledger systems, which were essentially integrated versions of well-defined clerical processes.

Selecting applications on such economical principles had side effects on the systems and the resulting application portfolio. Systems were developed with little regard to other, possibly related, systems and the systems portfolio of most companies became fragmented. There was usually a fair amount of duplication present in the various systems, mainly caused by the duplication of interrelated data. Conventional methods that evolved on the basis of practical experience with developing computing systems did not ease this situation. These early methods concentrated on making the computer work, rather than on rationalizing the processes they automated.

A parallel but separate development was the increasing use of operational research (OR) and management science (MS) techniques in industry and commerce. Although the theoretical work on techniques such as linear and non-linear programming, queueing theory, statistical inventory control, PERT-CPM, statistical decision theory, and so on, was well established prior to 1960, surveys indicated a burgeoning of OR and MS activity in industry in the United States and Europe during the 1960s. The surge in industrial and academic work in OR and MS was not unrelated to the presence and availability of ever more powerful and reliable computers.

In general terms, the OR and MS academics and practitioners of the 1960s were technically competent, enthusiastic and confident that their discipline would transform management from an art to a science. Another general remark that can fairly be made about this group, with the wisdom of hindsight, is that they were naive with respect to the behavioural and organizational aspects of their work. This fact unfortunately saw many enthusiastic and well-intentioned endeavours fail quite spectacularly, setting OR and MS into unfortunate disrepute which in many cases prohibited necessary reflection and reform of the discipline (Galliers and Marshall, 1985).

Data processing people, at the same time, started

developing their own theoretical base for the work they were doing, showing signs that a new profession was in the making. The different activities that made up the process of system development gained recognition and, as a result, systems analysis emerged as a key activity, different from O&M and separate from programming. Up to this point, data processing people possessed essentially two kinds of specialist knowledge, that of computer hardware and programming. From this point onwards, a separate professional — the systems analyst — appeared, bringing together some of the OR, MS and O&M activities hitherto performed in isolation from system development.

However, the main focus of interest was making those operations which were closely associated with the computer as efficient as possible. Two important developments resulted. First, programming (i.e. communicating to the machine the instructions that it needed to perform) had to be made less cumbersome. A new generation of programming languages emerged, with outstanding examples such as COBOL and FORTRAN. Second, as jobs for the machine became plentiful, development of special operating software became necessary, which made it possible to utilize computing power better. Concepts such as multi-programming, time-sharing and time-slicing started to emerge and the idea of a complex large operating system, such as the IBM 360 OS, was born.

New facilities made the use of computers easier, attracting further applications which in turn required more and more processing power, and this vicious circle became visible for the first time. The pattern was documented, in a light-hearted manner, by Grosch's law (1953). In simple terms it states that the power of a computer installation is proportional to the square of its cost. While this was offered as a not-too-serious explanation for the rising cost of computerization, it was quickly accepted as a general rule, fairly representing the realities of the time.

### The first sign of maturity

Computers quickly became pervasive. As a result of improvements in system software and hardware, commercial systems became efficient and reliable, which in turn made them more widespread. By the late 1960s most large corporations had acquired big mainframe computers. The era was characterized by the idea that 'large was beautiful'. Most of these companies had large centralized installations operating remotely from their users and the business.

Three separate areas of concern emerged. First, businesses started examining seriously the merits of introducing computerized systems. Systems

developed in this period were effective, given the objectives of automating clerical labour. But the reduction in the number of moderately paid clerks was more than offset by the new, highly-paid class of data processing professionals and the high cost of the necessary hardware. In addition, a previously unexpected cost factor, that of maintenance, started eating away larger and larger portions of the data processing budget. The remote 'ivory tower' approach of the large data processing departments made it increasingly difficult for them to develop systems that appealed to the various users. User dissatisfaction increased to frustration point as a result of inflexible systems, overly formal arrangements, the very long time required for processing changes and new requests, and the apparent inability of the departments to satisfy user needs.

Second, some unexpected side-effects occurred when these computer systems took over from the previous manual operations: substantial organizational and job changes became necessary. It was becoming clear that data processing systems had the potential of changing organizations. Yet, the hit and miss methods of system development concentrated solely on making the computer work. This laborious process was performed on the basis of ill-defined specifications, often the result of a well-meaning technologist interpreting the unproven ideas of a remote user manager. No wonder that most systems were not the best! But even when the specification was reasonable, the resulting system was often technically too cumbersome, full of errors and difficult to work with.

Third, it became clear that the majority of systems, by now classed as 'transaction processing' systems, had major limitations. Partly, the centralized, remote, batch processing systems did not fit many real-life business situations. These systems processed and presented historical rather than actual information. Partly, data was fragmented across these systems, and appeared often in duplicated, yet incompatible format.

It was therefore necessary to re-think the fundamentals of providing computer support. New theoretical foundations were laid for system development. The early trial-and-error methods of developing systems were replaced by more formalized and analytical methodologies, which emphasized the need for engineering the technology to pre-defined requirements. 'Software engineering' emerged as a new discipline and the search for requirement specification methods began.

Technological development also helped a great deal in clarifying both the theoretical and practical way forward. From the mid-1960s a new class of

computer – the mini – was being developed and by the early 1970s it emerged as a rival to the mainframe. The mini was equipped for 'real' work, having arrived at the office from the process control environment of the shopfloor. These small versatile machines quickly gained acceptance, not least for their ability to provide an on-line service. By this time the commercial transaction processing systems became widespread, efficient and reliable. It was therefore a natural next step to make them more readily available to users, and often the mini was an effective way of achieving this aim. As well as flexibility, minis also represented much cheaper and more convenient computing power: machine costs were a magnitude under the mainframe's; the physical size was much less; the environmental requirements (air conditioning, dust control, etc.) were less stringent and operations required less professional staff. The mini opened up the possibility of using computing power in smaller companies. This, in turn, meant that the demand grew for more and better systems and, through these, for better methods and a more systematic approach to system development.

### Practical solutions to practical problems

A parallel but separate area of development was that of project management. Those who followed the philosophy that 'large is beautiful' did not only think in terms of large machines. They aspired to large systems, which meant large software and very large software projects. Retrospectively it seems that those who commissioned such projects had little understanding of the work involved. These large projects suffered from two problems, namely, false assumptions about development and inadequate organization of the manpower. Development was based on the idea that the initial technical specification, developed in isolation from the users, was infallible. In addition, 'large is beautiful' had an effect on the structure of early data processing departments. The highly functional approach of the centralized data processing departments meant that the various disciplines were compartmentalized. Armies of programmers existed in isolation from systems analysts and operators with, very often physical, brick walls dividing them from each other and their users. Managing the various steps of development in virtual isolation from each other, as one would manage a factory or production line (without of course the appropriate tools!) proved to be unsatisfactory. The initial idea of managing large computer projects using mass production principles missed the very point that no two systems are the same and no two analysts or programmers do exactly the same work. Production line management methods in

the systems field backfired and the large projects grew manyfold during development, eating up budgets and timescales at an alarming rate.

The idea that the control of system development could and should be based on principles different from those of mass production and of continuous process management dawned on the profession relatively late. By the late 1960s the problem of large computing projects reached epidemic proportions. Books, such as Brooks's *The Mythical Man Month* (1972), likening system development to the prehistoric fight of dinosaurs in the tar-pit, appeared on the book-shelves. Massive computer projects, costing several times the original budget and taking much longer than the original estimates indicated, hit the headlines in the popular press.

Salvation was seen in the introduction of management method that would allow reasoned control over system development activities in terms of controlling the intermediate and final products of the activity, rather than the activity itself. Methods of project management and principles of project control were transplanted to data processing from complex engineering environments and from the discipline developed by the US space programme.

Dealing with things that are large and complex produced some interesting and far-reaching side-effects. Solutions to the problems associated with the (then fashionable) large computer programs were discovered through finding the reasons for their apparent unmaintainability. Program maintenance was difficult because it was hard to understand what the code was supposed to do in the first place. This, in turn, was largely caused by three problems. First, most large programs had no apparent control structure; they were genuine monoliths. The code appeared to be carved from one piece. Second, the logic that was being executed by the program was often jumping in an unpredictable way across different parts of the monolithic code. This 'spaghetti logic' was the result of the liberal use of the 'GO TO' statement. Third, if documentation existed at all for the program, it was likely to be out of date, not accurately representing what the program was doing. So, it was difficult to know where to start with any modification, and any interference with the code created unforeseen side effects. All this presented a level of complexity that made program maintenance problematic.

As a result of realizing the causes of the maintenance problem, theoreticians started work on concepts and methods that would help to reduce program complexity. They argued that the human mind is very limited when dealing with highly complex things, be they computer systems or anything else. Humans can deal with complexity

only when it is broken down into 'manageable' chunks or modules, which in turn can be inter-related through some structure. The uncontrolled use of the 'GO TO' statement was also attacked, and the concept of 'GO TO-less' programming emerged. Later, specific languages were developed on the basis of this concept; PASCAL is the best known example of such a language.

From the 1970s onwards modularity and structure in programming became important and the process by which program modules and structures could be designed to simplify complexity attracted increased interest. The rules which govern the program design process, the structures, the parts and their documentation became a major pre-occupation of both practitioners and academics. The concept of structuring was born and structured methods emerged to take the place of traditional methods of development. Structuring and modularity have since remained a major intellectual drive in both the theoretical and practical work associated with computer systems.

It was also realized that the principles of structuring were applicable outside the field of programming. One effect of structuring was the realization that not only systems but projects and project teams can be structured to bring together — not divide — complex, distinct disciplines associated with the development of systems. From the early 1970s, IBM pioneered the idea of structured project teams with integrated administrative support using structured methods for programming (Baker, 1972), which proved to be one of the first successful ploys for developing large systems.

### From processes to data

Most early development methods concentrated on perfecting the processes that were performed by the machine, putting less emphasis on data and giving little, if any, thought to the users of the system. However, as more and more routine company operations became supported by computer systems, the need for a more coherent and flexible approach arose. Management need for cross-relating and cross-referencing data, which arises from basic operational processes, in order to produce coherent information and exercise better control, meant that the cumbersome, stand-alone and largely centralized systems operating in remote batch mode were no longer acceptable. By the end of the 1960s the focus of attention shifted from collecting and processing the 'raw material' of management information, to the raw material itself: data. It was discovered that interrelated operations cannot be effectively controlled without maintaining a clear set of basic data, preferably in a way that would allow data

to be independent of their applications. It was therefore important to de-couple data from the basic processes. The basic data could then be used for information and control purposes in new kinds of systems. The drive for data independence brought about major advances in thinking about systems and in the practical methods of describing, analysing and storing data. Independent data management systems became available by the late 1960s.

The need for accurate information also highlighted a new requirement. Accurate information needs to be precise, timely and available. During the 1970s most companies changed to on-line processing to provide better access to data. Many companies also distributed a large proportion of their central computer operations in order to collect, process and provide access to data at the most appropriate points and locations. As a result, the nature of both the systems and the systems effort changed considerably. By the end of the 1970s the relevance of data clearly emerged, being viewed as *the* fundamental resource of information, deserving treatment that is similar to any other major resource of a business.

There were some, by now seemingly natural side-effects of this new direction. Several approaches and methods were developed to deal with the specific and intrinsic characteristics of data. The first of these was the discovery that complex data can be understood better by discovering their apparent structure. It also became obvious that separate 'systems' were needed for organizing and storing data. As a result, databases and database management systems (DBMS) started to appear. The intellectual drive was associated with the problem of how best to represent data structures in a practically usable way. A hierarchical representation was the first practical solution. IBM's IMS was one of the first DBMSs adopting this approach. Suggestions for a network-type representation of data structures, using the idea of entity-attribute relationships, were also adopted, resulting in the CODASYL standard. At the same time, Codd started his theoretical work on representing complex data relationships and simplifying the resulting structure through a method called 'normalization'.

Codd's fundamental theory (1970) was quickly adopted by academics. Later it also became the basis of practical methods for simplifying data structures. Normalization became the norm (no pun intended) in better data processing departments and whole methodologies grew up advocating data as the main analytical starting point for developing computerized information systems. The drawbacks of hierarchical and network-type databases (such as the inevitable duplication of

data, complexity, rigidity, difficulty in modification, large overheads in operation, dependence on the application, etc.) were by then obvious. Codd's research finally opened up the possibility of separating the storage and retrieval of data from their use. This effort culminated in the development of a new kind of database: the relational database.

Design was also emerging as a new discipline. First, it was realized that programs, their modules and structure should be designed before being coded. Later, when data emerged as an important subject in its own right, it also became obvious that system and data design were activities separate from requirement analysis and program design. These new concepts had crystallized towards the end of the 1970s. Sophisticated, new types of software began to appear on the market, giving a helping hand with organizing the mass of complex data on which information systems were feeding. Databases, data dictionaries and database management systems became plentiful, all promising salvation to the overburdened systems professional. New specializations split the data processing discipline: the database designer, data analyst, data administrator joined the ranks of the systems analyst and systems designer. At the other end of the scale, the programming profession was split by language specialization as well as by the programmer's conceptual 'distance' from the machine. As operating software became increasingly complex, a new breed – the systems programmer – appeared, emphasizing the difference between dealing with the workings of the machine and writing code for 'applications'.

### Towards management information systems

The advent of databases and more sophisticated and powerful mainframe computers gave rise to the idea of developing corporate databases (containing all the pertinent data a company possessed), in order to supply management with information about the business. These database-related developments also required data processing professionals who specialized in organizing and managing data. The logical and almost clinical analysis these specialists performed, highlighted not only the structures of data but also the many inconsistencies which often exist in organizations. Data structures reflect the interpretation and association of data in a company, which in turn reflect interrelationships in the organization. Some data processing professionals engaged in data analysis work began to develop their own view of how organizations and their management could be transformed on the basis of the analysis. They also developed some visionary notions about themselves. They thought that they would decide



(or help to decide) what data an organization should have in order to function efficiently, and who would need access to which piece of data and in what form.

The idea of a corporate database that is accurate and up to date with all the pertinent data from the production systems, is attractive. All we need to do – so the argument goes – is aggregate the data, transform them in certain ways and offer them to management. In this way a powerful information resource is on tap for senior management. Well, what is wrong with this idea?

Several practical matters presented difficulties to the naive data processing visionary who believed in a totally integrated management information system (MIS) resting on a corporate database. One problem is the sheer technical difficulty of deciding what should be stored in the corporate database and then building it satisfactorily before an organizational change, brought about by internal politics or external market forces or both, makes the database design and the accompanying reports inappropriate. In large organizations it may take tens of man-years and several elapsed years to arrive at a partially integrated MIS. It is almost certain that the requirements of the management reports would change over that period. It is also very likely that changes would be necessary in some of the transaction processing systems and also in the database design. Furthermore, assuming an efficient and well integrated set of transaction processing systems, the only reports that these systems can generate without a significant quantum of effort are historical reports containing aggregated data, showing variances – ‘exception reports’ (e.g. purchase orders for items over a certain value outstanding for more than a pre-defined number of days) and the like. Reports that would assist management in non-routine decision making and control would, by their nature, require particular views of the data internal to the organization that could not be specified in advance. Management would also require market data, i.e. data external to the organization’s transaction processing systems. Thus, if we are to approach the notion that seems to lie behind the term MIS and supply managers with information that is useful in business control, problem solving and decision making, we need to think carefully about the nature of the information systems we provide.

It is worth noting that well organized and well managed businesses always had ‘systems’ (albeit wholly or partly manual) for business control. In this sense management information systems always existed, and the notion of having such systems in an automated form was quite natural, given the advances of computing technology

that were taking place at the time. However, the unrealistic expectations attached to the computer, fuelled by the overly enthusiastic approaches displayed by the data processing profession, made several, less competently run, companies believe that shortcomings in management, planning, organization and control could be overcome by the installation of a computerized MIS. Much of the later disappointment could have been prevented had these companies realized that technology can only solve technical and not management problems.

Nevertheless, the notion that information provision to management, with or without databases, was an important part of the computing activity, was reflected by the fact that deliberate attempts were made to develop MISs in greater and greater numbers. Indicative of this drive towards supporting management rather than clerical operations is the name change that occurred around this time: most data processing departments became Management Services departments. The notion was that they would provide, via corporate databases, not only automated clerical processing but also, by aggregating and transforming such data, the information that management needed to run the business.

That the data processing profession during the 1970s developed useful and powerful data analysis and data management techniques, and learned a great deal about data management, is without doubt. But the notion that, through their data management, data aggregation and reporting activities, they provided management with information to assist managerial decision making had not been thought through. As Keen and Scott Morton (1978) point out, the MIS activity was not really a focus on management information but on information management. We could go further: the MIS activity of the era was concerned with *data* management, with little real thought being given to meeting management information needs.

In the late 1970s Keen and Scott Morton were able to write without fear of severe criticism that

... management information system is a prime example of a ‘content-free’ expression. It means different things to different people, and there is no generally accepted definition by those working in the field. As a practical matter MIS implies computers, and the phrase ‘computer-based information systems’ has been used by some researchers as being more precise.

Sprague and Carlson (1982) attempted to give meaning to the term MIS by noting that when it is used in practice, one can assume that what is being referred to is a computer system with the following characteristics:

- an information focus, aimed at middle managers;
- structured information flows;
- integration of data processing jobs by business function (production MIS, personnel MIS, etc.); and
- an inquiry and report generation facility (usually with a database).

They go on to note that

. . . the MIS era contributed a new level of information to serve management needs, but was still very much oriented towards, and built upon, information flows and data files.

The idea of integrated MISs seems to have presented an unrealistic goal. The dynamic nature of organizations and the market environment in which they exist forces more realistic and modest goals on the data processing professional. Keeping the transaction processing systems maintained, sensibly integrated and in line with organizational realities, is a more worthwhile job than freezing the company's data in an overwhelming database.

The era also saw data processing professionals and the management science and business modelling fraternities move away from each other into their own specialities, to the detriment of a balanced progress in developing effective and useful systems.

### The emergence of information technology

Back in the 1950s Jack Kilby and Robert Noyce noticed the semi-conducting characteristics of silicon. This discovery, and developments in integrated circuitry, led to large-scale miniaturization in electronics. By 1971 microprocessors using 'silicon chips' were available on the market (Williams and Welch, 1985). In 1978 they hit the headlines – commentators predicting unprecedented changes to business and personal life as a result. A new, post-industrial revolution was promised to be in the making (Tofler, 1980).

The impact of the very small and very cheap, reliable computers – micros – which resulted from building computers with chips, quickly became visible. By the early 1980s computing power and facilities suddenly became available and possible in areas hitherto untouched by computers. The market was flooded with 'small business systems', 'personal computers', 'intelligent work stations' and the like, promising the naive and the uninitiated instant computer power and instant solution to problems.

As a result, three separate changes occurred. First, users, especially those who had suffered unworkable systems and waited for years to receive systems to their requirements, started bypassing data processing departments and buying

their own computers. They might not have achieved the best results but increased familiarity with the small machines started to change attitudes of both users and management.

Second, the economics of systems changed. The low cost of the small machines highlighted the enormous cost of human effort required to develop and maintain large computer systems. Reduction, at any cost, of the professional system development and maintenance effort was now a prime target in the profession, as (for the first time) hardware costs could be shown to be well below those of professional manpower.

Third, it became obvious that small dispersed machines were unlikely to be useful without interconnecting them – bringing telecommunications into the limelight. And many office activities, hitherto supported by 'office machinery' were seen for the first time as part of the process started by large computers – that is, automating the office. Office automation emerged, not least as a result of the realization by office machine manufacturers, who now entered the computing arena, that the 'chip' could be used in their machines. As a consequence, hitherto separate technologies – that of telephony, telecommunication, office equipment and computing – started to converge. This development pointed to the reality that voice, images and data are simply different representations of information and that the technologies that deal with these different representations are all part of a new complex technology: information technology.

The resulting developments became diverse and complex: systems developers had to give way to the pressure exercised by the now not so naive user for more involvement in the development of systems. *End user computing* emerged as a result, promoting the idea that systems are the property of users and not the technical department. In parallel, the realization occurred that useful systems can only be produced if those who will use them take an active part in their development. Integrating the user became a useful obsession, helping the development of new kinds of systems.

It also became clear that a substantial reduction in the specialist manual activity of system development is necessary if the new family of computers, and the newly-discovered information technology, are to be genuinely useful. Suddenly, there were several alternatives available. Ready-made application systems emerged in large numbers for small and large machines, and *packages* became a fashionable business to be in. *Tools for system development*, targeting directly the end user and supporting end-user computing,

were developed in the form of special, high-level facilities for interrogating databases and formatting reports. Ultra high-level languages emerged carrying the name 'fourth generation languages' (4GLs) to support both professional and amateur efforts at system development.

For the first time in the history of computing, serious effort was made to support with automation the manifold and often cumbersome activities of system development. Automated programming support environments, systems for building systems, analysis and programming workbenches appeared on the market, many backing the specialist methodologies which, by now, became well-formulated, each with its own cult following.

### New approaches to system development

In addition, new discoveries were made about the nature of systems and system development. From the late 1960s it was realized that the development of a system and its operations can be viewed as a cycle of defined stages. The 'life-cycle' view of systems emerged and this formed the basis of many methods and methodologies for system development. It became clear only later that, while the view of a life-cycle was the correct one, a *linear* view of the life-cycle was counter-productive. The linear view was developed at the time when demand for large-scale systems first erupted and most practitioners were engaged mainly in development. The first saturation point brought about the shock realization that these systems needed far more attention during their operational life than was originally envisaged. As the maintenance load on data processing departments increased from a modest 20 to 60, 70 and 80 per cent during the 1970s, many academics and practitioners started looking for the reasons behind this (for many, undesirable and unexplained) phenomenon.

It was discovered that perhaps three different causes can explain the large increase in maintenance. First, the linear view of the life-cycle can be misleading. Systems developed in a linear fashion were built on the premise that successive deductions would be made during the development process, each such deductive step supplying a more detailed specification to the next one. As no recursive action was allowed, the misconceptions, errors and omissions left in by an earlier step would result in an ever-increasing number of errors and faults being built into the final system. This, and the chronic lack of quality control over the development process, delivered final systems which were far from perfect. As a result, faults were being discovered which needed to be dealt with during the operational part of

the life-cycle, thereby increasing unnecessarily the maintenance load. It was discovered that early faults left in a system increase the number of successive faults in an exponential way, resulting in hundredfold increases in effort when dealing with these faults in the final system.

Second, there are problems associated with specifications. The linear life-cycle view also assumed that a system could be safely built for a long life, once a specification had been correctly developed, as adjustments were unlikely to be required provided the specification was followed attentively. This view had negated the possibility that systems might have a changing effect on their environment, which, in turn, would raise the requirement for re-tuning and readjusting them. The followers of this approach had also overlooked the fact that real business, which these systems were supposed to serve, never remains constant. It changes, thereby changing the original requirements. This, in turn, would require readjusting or even scrapping the system. Furthermore, the idea that users could specify precisely their requirements seems to have been largely a fallacy, negating the basis on which quite a few systems had been built.

Third, maintenance tends to increase as the number of systems grows. It is misleading to assume that a percentage increase in the maintenance load is in itself a sign of failure, mismanagement or bad practice. Progressing from the state of having no computer system to the point of saturation means that, even in a slowly changing environment and with precision development methods, there would be an ever-decreasing percentage of work on new development and a slow but steady increase in the activities dealing with systems already built.

Nevertheless, the documented backlog of system requests grew alarmingly, estimated by the beginning of the 1980s at two to five years' worth of work in major data processing departments. This backlog evolved to be a mixture of requests for genuine maintenance, i.e. fixing errors, adjustments and enhancements to existing systems, and requirements for new systems. It was also realized that behind this 'visible' backlog, there was an ever-increasing 'invisible', undocumented backlog of requirements estimated at several times the visible one. The invisible backlog consisted largely of genuine requests that disillusioned users were no longer interested in entering into the queue.

As a response to these problems, several new developments occurred. Quality assurance, quality control and quality management of system development emerged, advocating regular and specific tests and checks to be made on the system during



its development. Walk-throughs and inspections were inserted into analysis, design and programming activities to catch 'bugs' as early (and as cheaply) as possible.

The notion that systems should be made to appeal to their users in every stage of development and in their final form encouraged the development of 'user friendly' systems, in the hope that early usability would reduce the requests for subsequent maintenance. Serious attempts were made to encourage an iterative form of development with high user involvement in the early stages, so that specifications would become as precise as possible. The idea of building a prototype for a requirement before the final system is built and asking users to experiment with the prototype before finalizing specifications helped the system development process considerably.

By now, the wide-ranging organizational effects of computer systems became clearly visible. Methods for including organizational considerations in system design started to emerge. A group of far-sighted researchers, Land and Mumford in the UK, Agarin in the USA, Bjorn-Andersen in Denmark, Ciborra in Italy and others, put forward far-reaching ideas about letting systems evolve within the organizational environment, thereby challenging the hitherto 'engineering-type' view of system development. For the first time since the history of computing began, it was pointed out that computerized information systems were, so to speak, one side of a two-sided coin, the other side being the human organization where these systems perform. Unless the two are developed in unison, in conjunction with each other, the end result is likely to be disruptive and difficult to handle.

Despite these new discoveries, official circles throughout the world had, successively failed to support developments in anything but technology itself and the highly technical, engineering-type approaches (Land, 1983). It seems as though the major official projects were mounted to support successive problem areas one phase behind the time! For example, before micros became widespread, it was assumed that the only possible bottleneck in using computers would be the relatively low number of available professional programmers. Serious estimates were made that if the demand for new systems should increase at the rate shown towards the end of the 1970s, this could only be met by an ever-increasing army of professional programmers. As a result, studies were commissioned to find methods for increasing the programming population several-fold over a short

period of time.\* Wrong assumptions tend to lead to wrong conclusions, resulting in misguided action and investment, and this seems to be hitting computing at regular intervals. Far too much attention is paid in the major development programmes of the 1980s to technology and far too little attention is paid to the *application* of the technology.

### New types of systems

The 1980s have brought about yet another series of changes. It has become clear that sophisticated hardware and software together can be targeted in different ways towards different types of application areas. New generic types of systems emerged on the side of data processing systems and MISs. Partly, it was realized that the high intelligence content of certain systems can be usefully deployed. Ideas originally put forward by the artificial intelligence (AI) community, which first emerged in the late 1950s as a separate discipline, now became realizable. Systems housing complex rules have emerged as 'rule-based' systems. The expressions 'expert systems' and 'intelligent knowledge-based systems' (IKBS) became fashionable to denote systems which imitate the rules and procedures followed by some particular expertise. Partly, it was assumed that computers would have a major role in supporting decision-making processes at the highest levels of companies and the concept of decision support systems (DSS) evolved. When remembering the arguments about management information systems, many academics and professionals have posed the question whether 'decision support system' was a new buzz word with no content or whether it reflected a new breed of systems. Subsequent research showed that the computerized system is only a small part of the arrangement that needs to be put in place for supporting top-level decision makers.

Manufacturers got busy in the meantime providing advanced facilities that were made available by combining office systems, computers and networks,

\* This approach is reminiscent of the famous calculation in the 1920s predicting the maximum number of motor cars ever to be needed on earth. The number was put at around 4 million on the basis that not more than that number of people would be found to act as chauffeurs for those who could afford to purchase the vehicles. It had never occurred to the researchers in this case that the end user, the motor car owner, might be seated behind the wheel, thereby reducing the need for career chauffeurs. Or that technological progress and social and economic change might reduce the need for specialist knowledge and the price and might also change the economic justification, factors which all affect the demand for motor cars.

and by employing the facilities provided by keypads, television and telecommunications. Electronic mail systems appeared, teleconferencing and videotex facilities shifted long-distance contact from the telephone, and – besides the processing of data – voice, text and image processing moved to the forefront. The emphasis shifted from the provision of data to the provision of information and to speeding up information flows.

### Important new roles for information systems

The major task for many information systems (IS) departments in the early 1980s is making information available. The problems of interconnecting and exchanging information in many different forms and at many different places turned the general interest towards telecommunications. This interest is likely to intensify as more and more people gain access to, or are provided with, computer power and technologically pre-processed information.

As a result of recent technological improvements and changes in attitudes, the role of both data processing professionals and users changed rapidly. More systems were being developed by the users themselves or in close co-operation with the users. Data processing professionals started assuming the role of advisers, supporters and helpers. Systems were being more closely controlled by their users than was the practice previously. A new concept – the information centre – emerged, which aimed at supporting end-user computing and providing information and advice for users, at the same time also looking after the major databases and production systems in the background.

The most important result of using computer technology, however, was the growing realization that technology itself cannot solve problems and that the introduction of technology results in change. The impact of technological change depends on why and how technology is used. As management now had a definite choice in the use of technology, the technological choices could be evaluated within the context of business and organizational choices, using a planned approach. For this reason more and more companies started adopting a planned approach to their information systems. 'System strategy' and 'strategic system planning' became familiar expressions and major methods have been developed to help such activities.

It has been realized also that applying information technology outside its traditional domain of back-room effectiveness and efficiency, i.e. moving systems out of the back room and into the 'sharp end' of the business, would create, in many cases,

distinct competitive advantage to the enterprise. This should be so, because information technology can effect the competitive forces that shape an industry by

- building barriers against new entrants;
- changing the basis of competition;
- changing the balance of power in supplier relationships;
- tying in customers;
- switching costs; and
- creating new products and services.

By the mid-1980s this new strategic role of information systems emerged. From the USA came news of systems that helped companies to achieve unprecedented results in their markets. These systems were instrumental in changing the nature of the business, the competition and the company's competitive position. The role of information systems in business emerged as a strategic one and IS professionals were elevated in status accordingly. At the same time the large stock of old systems became an ever-increasing burden on companies wanting to move forward with the technology.

More and more researchers and practitioners were pointing towards the need for linking systems with the business, connecting business strategy with information system strategy. The demand grew for methods, approaches and methodologies that would provide an orderly process to strategic business and system planning. Ideas about analysing user and business needs and the competitive impact of systems and technologies are plentiful. Whether they can deliver in line with the expectations will be judged in the future.

### Summary

The role of computerized information systems and their importance in companies have undergone substantial transition since the 1950s. Over the same period both the technology and the way it was viewed, managed and employed changed considerably. The position and status of those responsible for applying the technology in various organizations have become more prominent, relevant and powerful, having moved from data processing, through management services, to information processing. At the same time, hitherto separate technologies converged into information technology.

As technology moved from its original fragmented and inflexible form to being integrated and interconnected, the management of its use in terms of both operations and system development changed in emphasis and nature. Computer operations moved from a highly regulated, centralized and remote mode to becoming more *ad hoc* and

available as and when required. The systems effort itself progressed from concentrating on the programming process, through discovering the life-cycle of systems and the relevance of data, to more planned and participative approaches. The focus of attention changed from the technicalities to social and business issues.

Systems originally replaced clerical activities on the basis of stand-alone applications. The data processing department's original role was to manage the delivery and operation of these predominantly back-room systems. When data became better integrated, and more management-orientated information was provided, the management services departments started concentrating on better management of their own house and on making links with other departments and functions of the business which needed systems. This trend, combined with the increased variety and availability of sophisticated and easier to use technology, has led to the users taking a more active role in developing their own systems.

Lately, since it is realized that information is an important resource which can be used in a novel way to enhance the competitive position of businesses, information technology and information systems are becoming strategically important for businesses. Information systems are moving out of the back-room, low-level support position, to emerge as the nerve centres of organizations and competitive weapons at the front end of businesses. The focus of attention moved from being tactical to becoming strategic, and changed the nature of systems and the system portfolio.

It is evident that activity in the information systems field will continue in many directions at once, driven by fashion and market forces, by organizational need and technical opportunity. However, it appears that the application of information technology is at the threshold of a new era, opening up new opportunities by using the technology strategically for the benefit of organizations and businesses. It is still to be seen how the technology and the developers will deliver against these new expectations.

## References

- Baker, F.T. (1972) Chief programmer team management of production programming. *IBM System Journal*, Spring.
- Brooks, F.R., Jr. (1972) *The Mythical Man-Month*. Addison-Wesley.
- Codd, E.F. (1970) A relational model of data for large shared data banks. *Communications of the ACM*, 13, 6.

Galliers, R.D. and Marshall, P.H. (1985) *Towards True End-User Computing: From EDP to MIS to DSS to ESE*. Western Australian Institute of Technology, Bentley, W.A.

Grosch, H.R.J. (1953) High-speed arithmetic: the digital computer as a research tool. *J. Opt. Soc. Am.*, April.

Keen, P.G.W. and Scott Morton, M.S. (1978) *Decision Support Systems: An Organisational Perspective*. Addison-Wesley.

Land, F.F. (1983) *Information Technology: The Alvey Report and Government Strategy*. An Inaugural Lecture, The London School of Economics.

Sprague, R.H. and Carlson, E.D. (1982) *Building Effective Decision Support Systems*. Prentice-Hall.

Tofler, A. (1980) *The Third Wave*. Bantam Books.

Williams, G. and Welch, M. (1985) A microcomputing timetable. *BYTE*, 10, 9, September.

---

## Biographical notes

Elisabeth Somogyi is head of PActel's IT Strategy Centre. She specializes in strategy formulation, strategic systems planning and the management of information systems and information technology. She undertakes consulting assignments and research, and also lectures on these subjects.

She graduated as a chemical engineer and mathematician in Hungary, where she subsequently worked in the chemical process industry. Since 1968 she worked in the UK, first in data processing and later in IT consultancy. She has held line management positions in leading consultancies (Langton Information Systems, F International, Butler Cox).

In recent years her consulting work was concerned with developing strategies, business and technical plans for large industrial, commercial and government organizations. She has also acted as adviser to the board and chief executive of large UK companies and was involved in IT education programmes for top management. She has carried out research in the strategic application of information systems, project management, quality management and new system development and design methods. She has also lectured on these subjects in the UK and in Europe.

She is a fellow of the British Computer Society, chairman of its Business Information Specialist Group, and a member of the Institute of Bankers and of the British Institute of Management. She is an invited lecturer at the London School of

Economics, where she teaches the 'Management of Information Systems', and is author of many research reports, papers and journal contributions.

Bob Galliers received his Bachelor's Degree in Economics from Harvard University in 1970, where he was a Harvard College Scholar. He returned to the UK at that time and pursued a career in Public Administration before entering Lancaster University where he obtained his Master's Degree in Systems with Distinction in 1978. Thereupon he joined the University's Systems Consultancy Company (ISCOL Ltd), specializing in consulting and applied research in Information Systems

Strategy Formulation. He joined the Western Australian Institute of Technology (WAIT) – now, Curtin University of Technology – in 1982 with the mission of establishing its master's course in Information Systems. He has been head of WAIT's (Curtin's) School of Computing and Quantitative Studies since 1983 and is currently undertaking research at the London School of Economics while on sabbatical.

**Address for correspondence:** PA Computers and Telecommunications, Rochester House, 33 Greycoat Street, London SW1P 2QF.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.