Oxford Surveys in Information Technology, Vol. 1, 1–53

Perspectives on fifth-generation computing

BRIAN R. GAINES

Department of Computer Science, York University; Department of Industrial Engineering, University of Toronto; Department of Computer Science, University of Calgary

Contents

Introduction: the shock of the fifth generation	2
Fifth-generation objectives	4
VLSI: the underlying technology push into consumer markets	5
The first four generations	8
Precursors in AI developments	10
Precursors in HCI developments	14
Past, present and future generations	17
The ICOT programme	19
Basic application systems	19
Basic software systems	24
New advanced architecture	28
Distributed function architecture	31
VLSI technology	34
Systematization technology	34
Development-supporting technology	37
The realism of the FGCS programme	37
Social impact of FGCS: an information age	37
Social determinants—a hierarchy of needs	40
Computing as a new medium	42
Summary and conclusions	44
Acknowledgements	47
References	47

Abstract

In 1981 the Japanese announced a programme of research on a fifth generation of computing systems (FGCS) that will integrate advances in very large scale integration, database systems, artificial intelligence, and the human/computer interface into a new range of computers that are closer to people in their communication and knowledge processing capabilities. The proposal was a shock at first, but Western research quickly reoriented to match the Japanese programme. This paper considers fifth-generation computing from a wide range of perspectives in order to understand the logic behind the programme, its chances of success, and its technical and social impact. The need for a consumer market for mass-produced powerful integrated circuits is shown to underlie the Japanese objectives. The project is

Address for correspondence: Department of Computer Science, York University, Downsview, Ontario, Canada.

placed in a historical perspective of work in computer science and related to the preceding generations of computers. The main projects in the Japanese programme are summarized and discussed in relation to similar research elsewhere. The social implications of fifth-generation developments are discussed and it is suggested that they grow out of society's needs. The role of fifth-generation computers in providing a new medium for communication is analysed. Finally, the basis for a Western response to the Japanese programme is summarized.

Introduction: the shock of the fifth generation

The Japanese initiative in 1981 of scheduling a development programme for a *fifth generation* of computers (Moto-oka 1982; Simons 1983; Fuchi 1984; Hirose and Fuchi 1984) shocked a drowsy West into realizing that computer technology had reached a new maturity. Fifth-generation computing systems (FGCS) would integrate advances in very large scale integration (VLSI), database management systems (DBMS), artificial intelligence (AI), and the human/computer interface (HCI) into a new range of computers that were closer to people in their communication and knowledge processing capabilities. It may be difficult to recapture the shock of this announcement: it was unforeseen, from an unexpected source, gave a status to AI research that was yet unrecognized in the West, and proposed an integration of technologies that were still seen as distinct.

FGCS involved a surprising confluence of features even to an industry habituated to innovation in technologies and concepts. Computer professionals are part of information technology, familiar with the computer and not naturally sympathetic to either those who fear it or those who see it as saviour. However, one attraction of the technology is surely the element of surprise, that each new journal and magazine shows something existing today that seemed to be some years away. No matter how we adapt to the pace of change, its acceleration always causes it to be a step ahead of us. The Japanese announcement is the most recent example of such *future shock* (Toffler 1970) in the computer industry.

The recognition of the significance of computing technologies such as AI, and the prestige and funding that accompany it, are clearly welcome to many in the computing industry. Around the world resources have been allocated to responsive competing activities. Western research has quickly reoriented to match the Japanese programme (HMSO 1982; Steier 1983). Wry remarks may be made to the effect that prophets have no honour in their own country (Feigenbaum and McCorduck 1983). However, it is accepted that a nation with the track record of competition in the auto and semiconductor industries achieved by the Japanese (Davidson 1984) is credible in its high technology planning. This credibility is now accepted as legitimating much related research in the West. We may rejoice in the funding and status accorded to the frontiers of

computing science, but we should still remember our own surprise at the announcement and continue to ask the questions that the fifth-generation concepts raise.

The pioneers of computing technology raised questions about its long-term impact on the structure of our society (Wiener 1950). Does fifth-generation computing presage massive social change, the *post-industrial society* (Bell 1973), the *third wave* of civilization (Toffler 1980) and the *information age* (Dizard 1982)? The Japanese programme has been justified in terms of economics, market requirements and leadership in technology. Are there other rationales which explain it? In the computer industry there has always been an interplay between technology-led and market-led developments. What are the technology and market forces on fifth-generation developments? How can we avoid the surprises in future; is there a better model of the technological, economic and social dynamics underlying the development of information technology? AI was oversold in the 1960s and there was a backlash in the 1970s; is the fifth generation being oversold now and will there be a backlash? The Japanese programme looks like system integration of every frontier technology in computer science. What will we have at the end and why should we want it? What are the implications of achieving the fifth-generation goals?

The wave of activity triggered off by the Japanese initiative is not a good environment in which to examine the current status of computing technology and our long range objectives. The objectives have been set. The paradigm has become accepted and a new value system has been widely adopted. Such shifts are common in science and technology (Kuhn 1962) and the very fact that one has occurred will soon be forgotten. However, there are many reasons why it is appropriate to look now at some of the deeper factors underlying the fifth-generation programme. The questions raised above are of broad general interest to many people, industries and nations throughout the world. They are also of major relevance to the planning of the fifth generation and responses to it. In establishing such activities we are engaged in both predictive and normative technological forecasting. Past forecasts for the computer industry have proved notoriously wrong. Some promising technologies have not matured. Some have advanced much more rapidly than expected. In addition, new and unexpected technologies have emerged. In general, overall trends have occurred well before predicted dates. Can we learn any lessons from this history in planning current programmes? If we cannot predict, how do we manage the change, unexpected failures, and unexpected opportunities? Will spin-offs outside the programme be more important than the goals achieved within it?

This paper provides a framework for discussing these questions by examining fifth-generation computing from a number of different perspectives—historical, social, economic and technical. It suggests that we are in a time of fundamental change in the impact of

computing on society, and vice versa, and that a variety of perspectives from inside and outside the industry can help us to focus on the key issues for the computing industry and for society.

Fifth-generation objectives

It is important to avoid confusion between two quite separate Japanese computer development programmes, both funded by MITI, the Ministry of Trade and Industry (Johnson 1983). The *National Superspeed Computer Project* is an eight-year programme started in January 1982 by the Scientific Computer Research Association, a consortium of the national Electro-Technical Laboratory and six Japanese computer companies. The objective is a machine with 10 billion floating point operations a second (10 BFLOPS) and 1 billion bytes memory with a speed of 1.5 billion bytes per second. The most likely technology is the high-electron mobility transistor (Eden and Welch 1983), with Josephson junction devices as an alternative (Zappe 1983). Thus, the outcome of this project could be machines highly competitive with successors to the Cray-1 and Cyber 205, and concern has been expressed about the erosion of the US position in supercomputers (Buzbee 1982; Barney 1983; Wilson 1983, 1984).

The *Fifth Generation Computer Project* was started in April 1982 and has a far broader base, a longer term and less specific objectives. ICOT, the Institute for New Generation Computer Technology, has 40 research staff seconded from industry and close links with a number of university projects on AI, its technology and applications, both in Japan and abroad. The objective is 'knowledge processing systems' using 'latest research results in VLSI technology, as well as technology of distributed processing, software engineering, knowledge engineering, artificial intelligence, and pattern information processing' and contributing 'to the benefit of all humankind' (ICOT 1983, p. 1). ICOT policy is determined by a main committee chaired by Moto-oka of Tokyo University, a social needs sub-committee chaired by Karatsu of Matsushita, an architecture sub-committee chaired by Aiso of Keio University, and a theory sub-committee chaired by Fuchi, the Director of ICOT. The short-term objective of current work at ICOT is tools production, notably a sequential inference machine (SIM) working at 30 KLIPS (logical inferences a second) operating in December 1983, and a parallel inference machine (PIM) using dataflow techniques scheduled for late 1984 and ultimately capable of working at 1 GLIPS (Kawanobe 1984).

Thus, while fifth-generation machines may be justly termed 'supercomputers', they are not targeted against current machines but are rather the necessary support to a new generation of knowledge processing systems. The only comparable machines are the various LISP machines of LMI, Symbolics and Xerox (Myers 1982), which currently occupy only a small and specialist market niche. There has

been concern expressed about the fifth-generation project which recognizes its knowledge engineering content (Feigenbaum and McCorduck 1983), but in much of the press coverage the classical supercomputer and the knowledge supercomputer have been lumped together as the fifth generation, and the essential differences between them have been lost.

It is easy to understand why the Japanese would wish to compete with the USA in the well established world market for supercomputers. However, it was a shock to the Western computer science world when Japan overnight legitimated AI and HCI research by making them the central themes of their fifth generation project. If we lose the distinction between the two types of supercomputer, we also lose this sense of shock and the curiosity to understand the significance of the logic behind their decision.

In the next section we develop a chain of reasoning stemming from developments in circuit technology and their impact on the marketplace which may provide a basic rationale for the surprising foci of attention of the Japanese programme.

VLSI: The underlying technology push into consumer markets

The dominant objective of the fifth-generation proposals is not the classical supercomputer but a more suitable one at the HCI, the *ease of use of computers by people*. As Moto-oka notes, 'In these systems, intelligence will be greatly improved to match that of a human being, and, when compared with conventional systems, man-machine interface will become closer to the human system' (Mota-oka, 1982, p. 7), and Karatsu reiterates the HCI theme: 'Until today, man had to walk toward the machine with effort to fit precisely. On the contrary, tomorrow, the machine itself will come to follow the order issued by man' (Karatsu 1982, p. 106). However, ease of use is not a primary objective in its own right. One can see it as satisfying a market need to make computers more readily used and hence more attractive and more widely used. Behind this market need, however, is a deeper 'need for a market' that is generated by the inexorable logic of VLSI chip manufacturing. We are moving towards a surplus of chips that can be absorbed only by the world-wide consumer market.

Some people attempting to buy certain processor and memory chips today will find the concept of a surplus ludicrous. However, scarcity in the latest devices is brought about by the current extreme price competition in small computer systems; simplistically, all micros are much the same in terms of hardware and are differentiated primarily by cost. Two hundred and fifty-six kilobyte RAMs and processors with in-built DMA and communications make a significant difference to board size and manufacturing cost. The newest devices will always be in great demand and short supply. However, history suggests that for the previous generation of devices there will be a price war between suppliers because of over supply.

This over supply problem will worsen with increasing power of the

coming generation of chips. There are three major factors underlying this power: increasing numbers of gates per chip; increasing speed of operation; increasing ease of design. In the 1970s Gordon Moore of Intel noted that the number of gates per chip had been doubling each year since the advent of the planar process in 1959. From 1973 to 1984 the rate of growth has dropped to a doubling every 18 months and will probably decline further. However, 1 Mbit chips have already been announced and there is confidence that one billion device chips will be achieved by the year 2000 (Robinson 1984).

A second factor involved in increasing power is that the increasing number of gates in modern VLSI is achieved through decreases in element size that also increase speed. Increases in complexity and increases in speed are positively correlated rather than subject to trade-off. Changes in the materials used in semiconductor manufacture, such as that from silicon to gallium arsenide (Johnsen 1984) will also serve to increase speed until we begin to approach the ultimate limits of quantum-mechanical effects.

The third factor increasing *available* power is the development of computer-aided design techniques for developing integrated circuits. *Silicon compilers* (Rupp 1981) are similar in concept to language compilers in that they take a high-level specification of design objectives, constraints and techniques, and convert it to a specification of a solution satisfying the objectives, within the constraints using the techniques. However, instead of producing object code for a target computer that will run a program, the silicon compiler produces a layout for a chip in a target circuit family that will result in an integrated circuit. Such computer-aided design systems have already proved their worth in the development of mainstream computer chips such as Ethernet (Hindin 1982) and Vax (Fields 1983) chips, and will increasingly enable the power of integrated circuits to be harnessed responsively to the changing needs of more specialist markets.

The potential power of modern integrated circuits has become extremely high. The problem is to know what to do with this power. Processor, memory and support chips account for the bulk of high-volume demand, with communications and graphics forming a second tier (*Electronics* 1984). The $12 billion turnover of the semiconductor industry in 1984 corresponded to some 25 chips a year for every person in the world; about one million transistor equivalents a person a year. We are manufacturing more and more of a product whose power is already great and increasing. The professional markets absorb much of this capacity now but only the consumer markets can support it in the long term. However, computer technology as yet plays only a minor role in these markets; one limited by the technical skills required of computer users. The key thrust of the fifth-generation programme is to overcome this limitation and make computers accessible to all by moving from information processing to knowledge processing, and from a

machine-like interface to the user to a human-like interface through speech, writing and vision.

Figure 1 shows the economic logic behind the fifth generation and the way this leads to the specified technical programme (Gaines and Shaw 1984*a*). The surplus capacity for chips is resolved by targeting consumer markets and this requires that anyone can use a computer and has the motivation to do so in a domestic context. This requires improvements to the human/computer interface, realistic simulation of real and fantasy worlds and the capability to encode expertise for use by others. These requirements then lead to projects for speech recognition, high-resolution graphics, languages for knowledge processing and so on. The customizability of computers is an important feature that must not be lost and yet the volumes involved are too large for professional programming. Hence automatic programming in some form or another is also necessary. The logic of the market place leads to the projects which form the fifth-generation programme.

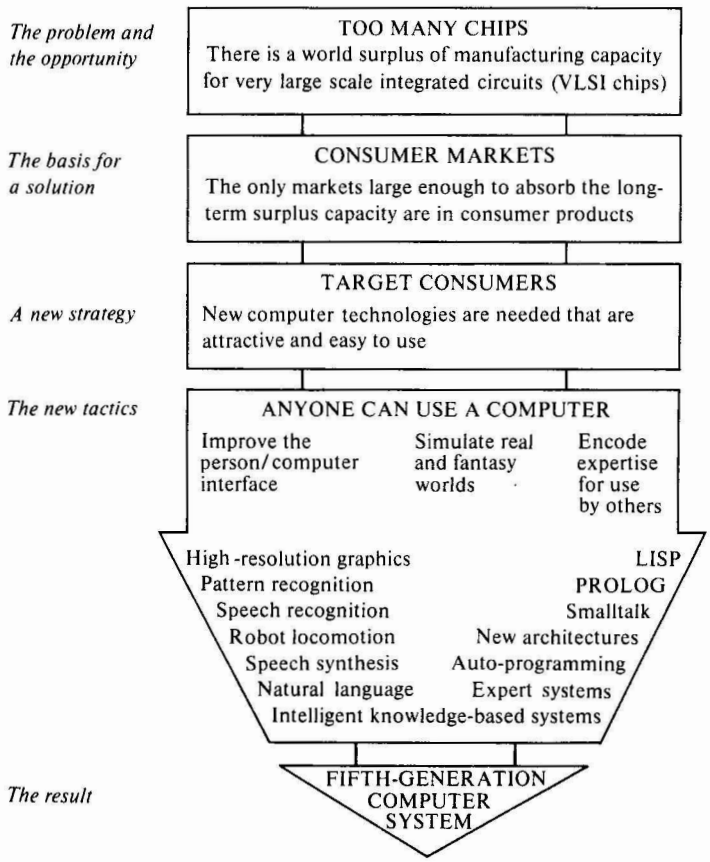

Fig. 1. The technology push from VLSI through consumer markets to the fifth-generation objectives.

Thus, the dramatic swing of computer technology into the consumer market through the advent of the personal computer will go very much further as the fifth-generation objectives are achieved. There will clearly be spin-offs of technology into the military, scientific, industrial and commercial markets, but these will be side effects of the main thrust rather than the primary target. The social impact of fifth-generation computing will be a universality of computer technology operating at the level of the mind of man. The key to commercial success in this technology will lie in understanding this impact and the new markets that underly it.

The following sections are concerned with perspectives on fifth-generation computing that gives a basis for this understanding.

The first four generations

The surprise at the fifth generation announcement was due to failure in technological forecasting (TF). We did not foresee AI and HCI technologies being so central on the time-scales scheduled by the Japanese. Such failures are common in all areas of TF (Godet 1983) and have been prevalent in the computer industry from the early days when a few machines were seen as adequate for all needs. It is worth examining analyses of the early generations and forecasts about the fourth and fifth generations to look for the basis of success and failure in prediction. Withington's (1974) analysis of 10 years ago is particularly interesting because he is describing three generations and predicting the fourth, which we now know about, and the fifth, which we are also predicting 10 years later. Table 1 presents a modified form of his analysis.

Withington's retrospective of the first three generations shows that they may be distinguished by hardware, software or pattern of

Table 1. Five generations of computer as seen in 1974

Generation	Years	Description	Hardware	Software	Effects
1	1953–8	Gee whiz!	Vacuum tubes and delay lines	None	Technicians and fears of automation
2	1958–66	Paper pushers	Transistors and magnetic cores	Compilers and I/O control	Proliferation of EDP groups and rigidity
3	1966–74	Communicators	Large-scale integrated circuits and interactive terminals	Operating systems and communications	Centralization of EDP as new function
4	1974–82	Information custodians	Very large file stores and satellite computers	Virtual machines	Redistribution of management functions
5	1982–	Action aids	Magnetic bubble, and laser holographic and distributed	Interactive languages and convenient simulation	Semi-automatic operating decisions

application, since all three are correlated. His description of the fourth generation then coming into use emphasizes large file stores and satellite computers which came about as database and distributed systems technology. Virtual machine software also came about but did not have as much impact as expected. Other articles of the mid 1970s emphasize *universal emulators* and the need for each generation of hardware to be able to run the software of all previous generations. Many manufacturers experimented with such emulation of competitor's machines but it has not become widely adopted, possibly because of software copyright and licensing problems. This was also the era of hardware/software unbundling.

Withington's projections of the fifth generation, like all at that time, fall far short of the target. Magnetic bubble technology lost out to advances in semiconductors. Interactive languages and convenient simulation were a major part of the fourth-generation thrust. Part of the reason for his failure in TF is that the three generations up to the fourth had been defined by IBM in introducing new families of machines. The third-generation 360, however, was upgraded to the 370 with little change in basic architecture and this has continued with the 3030, 4300 and 3080 series where greater power and cost-effectiveness has been the rule rather than architectural innovation (Wiener 1984). Some have doubted there has been a fourth generation:

> If the concept of computer generation is tied directly to advances in technology, we are faced with an anomalous situation in which the most powerful computers of 1979–1980, the CDC Cyber 176 and the CRAY 1, would have to be assigned to the second and third generation, respectively, while the most trivial of hobbyist computers would be a fifth-generation system (Rosen 1983).

This remark captures the essence of the problem of TF, for the business cycles underlying generations are related to market forces selectively emphasizing innovations in technology, not vice versa (Jantsch 1967). The market forces are themselves generated from social dynamics, and the technology is only a correlational, not a causal, variable. An early paper on the fourth generation in 1968 makes a plaintive remark which is another clue to the nature of computer generations: 'Today, programming has no theoretical basis and no theories. Why? Programming need not remain a handicraft industry. The designers of fourth generation computers must regard programming as a science rather than an art' (Walter, Bohl and Walter 1968).

After the second generation advances in the science of programming become more significant than refinement of computer architectures. The multi-tasking, time-sharing operation systems and database management systems of the third generation were the significant waves of new technology; they led to penetration of markets for integrated corporate information systems and computer utilities. The personal computer, word-processor, spread-sheet and games of

the fourth generation led to penetration of markets for low-cost personal resources for work and play.

Withington's titles for each generation are very evocative. The 1950s were an era of *gee whiz—computers can do anything*. This was necessary to support the act of faith by their users that one day they might become cost-effective. It took at least one more generation for this to happen. His name for the fifth generation is surprisingly apt; expert systems are certainly *action aids*, and they will be used for *semi-automatic operating decisions*. It is again an act of faith today that they, and other developments coming out of AI, will become cost-effective in a wide range of applications. Withington was able to project the function of the fifth generation better than its technology because it is the continuous evolution of social needs that determine the function whereas the technology is searched out from any available and may change discontinuously.

The following three sections analyse the underlying trends in the key fifth-generation technologies during the preceding generations. Later sections examine the way in which the generations have been driven by social needs. Withington's dates for the generations were based primarily on IBM generations and do not take into account other families of machines. More recent historical analysis suggests that it is better to define the generations in eight-year segments as follows: zeroth, 1940–7; first, 1948–55; second, 1956—63; third, 1964–71; fourth, 1972–9; fifth, 1980–7; sixth, 1988–95. These definitions will be used in the next two sections and discussed in the third.

Precursors in AI developments

Perhaps the greatest element of surprise about the Japanese programme was its dependence on advances in AI research. This has always been an important part of frontier research in computer science but not of major commercial significance. However, the argument of the VLSI push to consumer markets leads to a basic requirement for computing technologies that are part of AI, notably knowledge processing, natural language communication and speech recognition. The commercial requirement for AI to be routinely available marks the beginning of a new era for this field of research. The cycle of AI research shown in Table 2, from over-optimism in the 1960s through disenchantment in the 1970s to commercial success in the 1980s, is a significant historic framework for fifth-generation developments.

The concepts underlying AI of mind as mechanism were already present in the zeroth generation era of the 1940s with McCulloch and Pitt's (1943) work on the logic of neural networks and Rosenblueth, Wiener and Bigelow's (1943) analysis of behaviour, purpose and teleology. This led in the first-generation era to Wiener's (1948) development of *cybernetics* and von Bertalanffy's (1950) development of *general systems theory*, and the cybernetic machines

Table 2. The changing objectives of artificial intelligence research through the generations

Time period	Major events
Generation 1: Cybernetics (man as machine)	
1948–55	Cybernetics and general systems theory; Ashby's homeostat and Walter's tortoise; the Turing test and a checkers player
Generation 2: Generality and simplicity (the oversell)	
1956–63	General Problem Solver, Pandemonium, Perceptron, Adaline and neural nets
Generation 3: Performance by any means (the reaction).	
1964–71	Non-brain-like machines; emulation of human performance on general-purpose machines.
Generation 4:	
1972–9	(a) *Disenchantment (the over-reaction)* The work of Dreyfus, Lighthill and Weizenbaum
	(b) *Encoded expertise (achievement)* Development of effective expert systems; a more open forum—simulation of person, acquisition of knowledge, new architectures for AI
Generation 5: The fifth generation (commercialization)	
1980–7	A national priority—commercial and military; conversational interaction through speech and writing with an intelligent knowledge-based system

emulating feedback in animal behaviour such as Ashby's (1952) homeostat and Walter's (1953) tortoise. In this era also Turing (1950) raised the question of how we might test for the emulation of human intelligence in computers.

However, the generally accepted starting period for research on AI is in the second generation era of the late-1950s with work by Newell, Shaw and Simon (1958) on the *General Problem Solver*; by Samuels (1959) on a *checkers player*; by McCarthy (1959) on *programs with common sense*; by Selfridge (1959) on *Pandemonium*; by Rosenblatt (1958) on the *Perceptron*; by Widrow (1959) on *Adalines*; by Solomonoff (1957) on *mechanized induction*; and by Farley and Clark (1954) on *neural nets*. Minsky's (1961) survey gives a fairly comprehensive feeling for this era world-wide. The logic behind much of the work was that new forms of computer organized as aggregates of simple, self-organizing elements, could be induced to perform a task by mechanisms of learning through mimicking, reward and punishment similar to those of animal learning (Gaines and Andreae 1966).

These objectives and this type of work characterized the goals and approaches being taken world-wide at that time. They should be placed in the context of the early development of digital computers

where slow, expensive, unreliable machines with mercury-delay line memories were still in use programmed in various forms of assembler or autocode. The JOSS, Culler Fried, PLATO and Project MAC experiments on timesharing were part of the same era (Orr 1968). The atmosphere was one of immense excitement at the potential for computer systems to lead to the *augmentation of human reasoning* and *man–machine symbiosis* to use the terms of books and papers of that era (Sass and Wilkinson 1965).

The excitement and the funding died down through the 1960s as a result of two main factors. First, the work did not fulfil the promises made on its behalf: neural nets did not self-organize into brains; learning machines did not learn; perceptron-like elements did not recognize patterns. Secondly, the conventional digital computer became more reliable, smaller, faster, and cheaper, and the advantages of its sheer generality became widely realized. In the third-generation era of the mid-1960s effort switched out of designing brain-like machines to emulating human-like activities on general-purpose computers (Goldstein and Papert 1977). The focus of attention also became human performance, not human learning or human simulation. The initial rationale for this shift was as follows: if we could not program a computer to perform a task, then it was unlikely that we could program it to learn to perform that task; and if we could not program it somehow, then we certainly could not in a way that simulated a person. This became the definition of legitimate artificial intelligence research in the late 1960s and early 1970s: the performance of tasks that required intelligence when performed by people.

While the AI community regrouped around the revised goals in the third-generation era, the time-scale between research and widespread publication is such that even by the start of the fourth-generation era of the early 1970s many outside that community felt the necessity to express disenchantment with the whole endeavour. Dreyfus (1972), in his book entitled *What computers can't do*, detailed many of the over-optimistic claims for AI research and the ensuing under-achievement. He points to weaknesses in the philosophical and methodological foundations of work in AI. His 1979 revised edition at the end of the fourth-generation era reports some of the intervening debate resulting from his book and the changing priorities in AI research. His original book was a well balanced report that can be criticized primarily because it was out of date by the time it was published. Those in the AI community had already learnt the lessons he expressed and changed their strategy.

Lighthill's (1973) negative report on behalf of the Science Research Council on the status of AI research and the appropriate level of its funding in the UK reflected the disenchantment at the end of second-generation work with its over-sell and lack of achievement. However, there was another motivation behind the commissioning of that report and that was the fear of some of those responsible for

developing computer science departments in the UK that AI research would be funded most strongly outside those departments. At a mundane level the misgivings came from the realization that the expensive and powerful computers then necessary for AI research might not be used to improve the research facilities of recently formed computer science departments. More fundamentally it was sensed that developments in AI might prove to be a major part of the future foundations of computer science. Whatever the motivation, the outcome of the report was very negative for AI research in the UK during the fourth-generation era (Fleck 1982).

Weizenbaum's (1976) book, *Computer power and human reason*, was a far more personal statement than that of Dreyfus by someone who had been responsible for one of the early achievements of AI research. His ELIZA program (Weizenbaum 1966) was widely acclaimed in the late 1960s as the first successful attempt at passing the Turing test for AI. It could carry out *cocktail party conversation* with a person at a terminal that was remarkably human-like. However, the acclaim became embarrassment as it was realized that the simple mechanisms of ELIZA illustrated the weakness of the Turing test rather than a major advance in AI. People were all too ready to discern intelligence in machines and men, and common-sense human judgement in the matter was not an adequate criterion. The AI community was forced to re-consider what was meant by intelligence.

The shift of attention of AI work in the third generation era de-emphasized requirements for power and generality, consideration of computer architecture, and the simulation of human operation. It instead emphasized requirements to encode human expert knowledge and performance, by whatever means, for emulation by the computer. These targets resulted in practical achievements in the development of systems that could perform in diagnostic inference tasks as well as human experts, and the fourth-generation era became that of *expert systems* research (Michie 1979).

The strength of this paradigm shift cannot be over-emphasized. It defined the boundaries of an AI community that established parameters for funding and publication. The focusing of effort on performance led to achievements recognized outside this community and hence established the legitimacy of AI research. In recent years the community has become strong enough to begin to re-absorb some of the earlier objectives and concepts. Human simulation rather than just emulation is an objective of cognitive science (Lindsay and Norman 1977; Johnson-Laird and Wason 1977). Certain aspects of knowledge acquisition processes subsumed under learning are proving significant. The first steps are being taken towards new computer architectures targeted on AI requirements.

It would be foolish to pretend that the academic debates and funding struggles of the 1970s culminated in a clear realization of the significance of the eras described. The over-sell era remains in many

people's minds. The reaction era of regrouping under siege and bitter struggle remained in others. It took a nation of the East to draw to the attention of the West what its AI research had achieved in the achievement era and the significance of this for the world economy. We are now in the commercialization era where the understanding and development of FGCS are being treated as a national priority by many nations. The defence and commercial implications of achieving even part of the objectives set by the Japanese are a more compelling argument than any the AI community had been able to muster. The socio-cultural implications might also be overwhelming. The close AI community of earlier eras no longer exists. The financial pull of industry has fragmented effort (*Business Week* 1984) and it will be some time before new patterns of activity are clarified.

Precursors in HCI developments

The emphasis on the human/computer interface in the Japanese fifth-generation programme is less surprising than that on AI, and this mirrors the less dramatic history of HCI research and development (Gaines 1984). In many ways it parallels that of AI but has not given rise to the similar strong emotions. This may be due to the perceived thrust of the differing objectives: AI to emulate and possibly excel or replace us; HCI to support our interaction with the computer and enhance our own capabilities. It is probably more due to the differing resource requirements: AI research has needed a high proportion of the time of large expensive machines; HCI research has been possible ethologically by observing users of existing systems and through behavioural experiments on small laboratory machines (Gaines 1979). In recent years there has been a strong convergence between the two areas as AI techniques have come to be applied in HCI development. AI research has come to concentrate on the combination of person and computer in joint problem solving, and AI techniques have become usable in low-cost systems.

Again it is possible to see the beginnings of concern with HCI even in the zeroth-generation era of the 1940s with such remarks as Mauchly's in 1947 stressing the importance of ease of use of subroutine facilities: 'Any machine coding system should be judged quite largely from the point of view of how easy it is for the operator to obtain results' (Mauchly 1973). Control about social impact was expressed in the first generation era by Wiener (1950) in his book entitled *The human use of human beings*, and the second generation era saw Shackel's (1959) work on the ergonomics of a computer console. This era also saw the publication of what appeared an over-sell for its time, that of Licklider's (1960) *man–machine symbiosis*. However, the 'symbiosis' was not over-sold but stated only in low-key terms of computer support for our creative activities by off-loading data processing and information retrieval. Petri's (1962) theory of

asynchronous systems (Petri nets) was developed to model data processing systems including human users.

It was in the third-generation era that HCI began to flourish with the development from 1963 onwards of the first time-shared interactive systems such as MIT MAC (Fano 1965), RAND JOSS (Shaw 1968) and Dartmouth BASIC (Danver and Nevison 1969). Such systems came into widespread use, however, well before the human factors principles underlying their design were understood. Even towards the end of the third-generation era in surveying work on man–computer interaction, Nickerson (1969) remarked on its paucity and quoted Turoff to the effect that psychology should be able to contribute greatly to the design of interactive systems: 'when one looks at some of the current systems of this nature, it becomes quite evident that the evolution of these systems has not been influenced by this field.' Hansen (1971) seems to have made the first attempt to tabulate some user engineering principles for the design of interactive systems. Sackman's (1970) *Man–computer problem solving* and Weinberg's (1971) *Psychology of computer programming* did much to stimulate interest in the possible applications of human factors principles in computing science.

The fourth-generation era that saw achievement in AI also did so in HCI. Sime, Green and Guest (1973) aroused experimental psychological interest in HCI with the publication of their paper 'Psychological evaluation of two conditional constructions used in computer languages'. Martin's (1973) *Design of man–computer dialogues* began to formalize the guidelines for HCI design and the problems of the naive user were highlighted in Wasserman's (1973) paper at the National Computer Conference on 'The design of "idiot-proof" interactive systems'. The mid-1970s was the beginning of the era of the personal computer and the availability of low-cost computers with graphic displays led to their increasing use in psychological studies, and a boom in the associated literature. The decline in computer costs and the decreasing differences in computer facilities led to increasing commercial interest in good human factors as a marketing feature. *Ease-of-use* and *user-friendliness* began to be seen as saleable aspects of computer systems, and human engineers as product generators. Labour organizations intensified commercial interest as they promoted legislation relating to human factors of computer systems in the workplace, particularly the ergonomics of displays (Grandjean and Vigliani 1980). Research on HCI developed in industry and the papers from commercial sources further expanded an already swelling literature.

There has long been a close, but ambivalent relationship between studies of AI and those of HCI. Their key joint roles in the fifth-generation programme makes it important to examine this relation and those to other areas such as cognitive science and software engineering. Abstract definitions are probably of less value than a simple example that exaggerates the differences in approach.

Consider a problem of overcoming the barriers to database access by casual users:

(*a*) The *AI approach* to this problem might be to build a better database system with understanding of the user, his requirements, and natural language communication, i.e. to put all the load on the computer and make it intelligent enough to deal with the casual user.

(*b*) The *applied psychology* approach might be to develop a training programme for casual users that gave them as rapidly and effectively as possible the skills to use the database, i.e. to put all the load on the person and make him skilled enough to cope with the computer.

(*c*) The *HCI approach* might be to determine where the problems lie in the interface between user and computer and design a communication package that helps him formulate his requests in a way natural to him but which can be translated easily into a form natural to the database, i.e. to remove as much of the load as possible from both systems and share what is left between them.

In practice, a pragmatic system designer will take what he can get from all three approaches and put together a working system. However, the example shows, in very simplistic terms, both the close relationships between the disciplines and their differing orientations. Figure 2 incorporates related disciplines into an influence diagram indicating the rich structure now underpinning fifth-generation computer systems.

AI and HCI are now major influences on computer applications and both have influenced software engineering: AI by the development of symbolic languages for knowledge processing and HCI by developments for dialogue engineering. AI has had a direct influence on HCI by the introduction of intelligent interfaces and natural language processing. It has also had an indirect influence through its

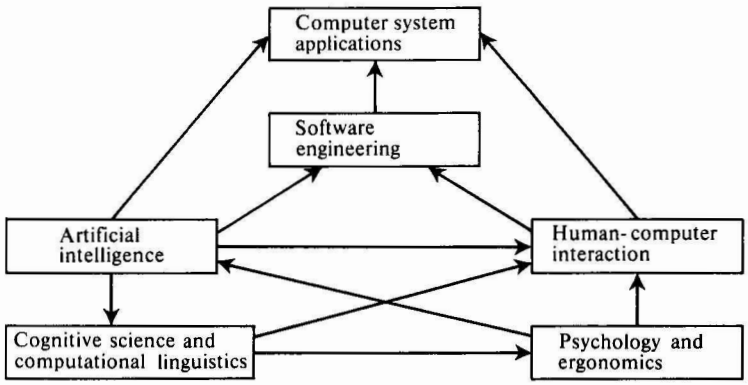

Fig. 2. Influences between the disciplines underlying fifth-generation systems.

promotion of interest in cognitive science and computational linguistics which provide tools for HCI. These areas have become influential in psychology and ergonomics, providing foundation for HCI. Completing the loop, psychology in its turn has influenced thinking in AI where the goal has been to emulate human intelligence rather than create truly artificial intelligence. The diagram is over-simplified and cases can be made for counter-influences to those shown, but it serves to show the key roles and close relationship of AI and HCI in supporting software engineering for the next generation of computer systems.

Past, present and future generations

It is useful to bring together the discussion of the past three sections and show the historic development of the various generations of computer systems. Table 3 is a retrospective view of the generations based on an extensive list of significant events in computing. Generations zero through four show an eight-year cyclic pattern that corresponds to the medium-term Juglar business cycle (Rau 1974; Bernstein 1982), and this has been projected to generations five and six. At the beginning of each generation a new systems technology comes into existence and at the end it is mature. The technology depends upon correlated advances in a number of sub-systems technologies with variation in which one is innovative at each stage. Withington's names have been retained for all but the fourth generation, which has been retitled *personal resources* to subsume both database access and the personal computer. The zeroth generation was certainly *up and down*, mainly down, and the sixth generation is seen as approaching the partnership of Licklider's *man–machine symbiosis*.

AI research has been shown separately because of its importance to the fifth-generation programme. It existed in each generation but its tactics changed markedly from one to another. The cybernetic approach of generality and simplicity dominated generations zero through three, but by the end of the third generation it was recognized as an oversell and there was a paradigm shift away from neural nets and learning machines to knowledge representation and performance. AI researchers reacted to the failure of systems based on brain-like mechanisms and switched effort to creating human-equal performance by any means. There was also an over-reaction, suggesting that the research was a waste of resources.

However, the performance paradigm produced breakthroughs such as SHRDLU (Winograd 1972), MYCIN (Shortliffe 1976) and ROBOT (Harris 1977), and the fourth generation ended with widespread acceptance of the commercial possibilities of AI: that natural language interaction with computers was possible in commercial systems; and that expert systems encoding human expertise could be used to enhance human decision making in medicine and oil prospecting. The fifth-generation proposals are based on these results

Table 3. Seven generations of computer as seen in 1984

Generation	Year	General features	State of AI research
0	1940–8	*Up and down* Relays replaced by vacuum tubes; COLOSSUS, ENIAC	*Mind as mechanism* Logic of neural networks; behaviour, purpose and teleology
1	1948–56	*Gee whiz!* Tubes, delay lines and drums; BINAC, EDSAC, ACE, WHIRLWIND, UNIVAC, EDVAC; IBM 701, 702, 650; numeric control, airborne navigational aids; *Human use of human beings*	*Cybernetics* Turing test; *Design for a brain*; Homeostat; Gray Walter's Tortoise; checkers player
2	1956–63	*Paper pushers* Transistors and core stores; I/O control programs; IBM 704, 7090, 1401, NCR 315; UNIVAC 1103, PDP 1, 3, 4, 5; FORTRAN, ALGOL, COBOL; batch, execs, supervisors; CTSS, MAC, JOSS, Petri nets; *Communications of the Association of Computer Manufacturers*	*Generality and simplicity—the oversell* Perceptron, GPS, EPAM; learning machines; *Self-organizing systems*; Dartmouth AI conference; Mechanization of Thought Processes symposium; IPL V, LISP 1.5
3	1964–71	*Communicators* Large-scale ICs; interactive terminals; IBM 360, 370; CDC 6600, 7600; PDP 6, 7, 8, 9, 10; TSS 360; DBMS, relational model; Intel 1103, 4004	*Performance by any means* Semantic sets, PLANNER; ELIZA, fuzzy sets, ATNs; dendral, scene analysis; resolution principle; Machine Intelligence 1; first IJACI; *International Journal of Man–Machine Studies*; *Artificial Intelligence*
4	1972–9	*Personal resources* Personal computers; supercomputers, VLSI; very large file stores; databanks, videotex; IBM 370/168—MOS memory and virtual memory; Intel 8080, NS PACE 16-bit; Altair and Apple personal computers; Visicalc, *Byte* magazine	*Encoded expertise and over-reaction* Prolog, Smalltalk; frames, scripts; systemic grammars; LUNAR, SHRDLU, MARGIE, LIFER, ROBOT, MYCIN, TEIRESIAS; Dreyfus, Lighthill and Weizenbaum's attacks on AI; *Cognitive Science*
5	1980–87	*Action aids* Personal computers with power and storage of mainframes plus graphics and speech processing; networks, utilities; OSI, NAPLPS standards; IBM 370 chip, Intel iAPX 32-bit; HP-900 chip with 450 000 transistors; Apple Macintosh	*Commercialization* Lisp and Prolog machines; expert system shells, knowledge-bases; EMYCIN, AL/X, APES; Japanese fifth-generation project; ICOT personal sequential inference machine and parallel inference machine; *Handbook of artificial intelligence*
6	1988–95	*Partners* Optical logic and storage; organic processing elements; AI techniques in routine use	*Modelling—emotion and awareness* Massive parallel knowledge systems; audio and visual sensors of ear and eye; multimodal modelling

and propose to take them to their logical conclusion under the pressure of the forces towards the consumer market already noted. This generation commenced in 1980 with the commercial availability of LISP machines and will be completed by 1988 as this technology becomes widely available at low cost and high speed, and is exploited in the consumer marketplace.

The projection of the sixth generation from this table is clearly speculative and prone to error. It will probably involve new subsystem technologies for high-density information storage and processing which are under investigation now, such as optical logic and organic processing elements (Tucker 1984). The AI and HCI activities will have completely converged and the research focus will move down from the cortical knowledge-processing level of the brain to the limbic attention-directing level. This is only now beginning to be studied in cognitive science (Norman 1980) and manifests itself in human emotions and awareness (Gaines and Shaw 1984*c*).

This is the historic framework in which FGCS developments are taking place. However, they are also defined by the content of the Japanese FGCS programme and the next section analyses this in detail.

THE ICOT programme

One of the most interesting features of the Japanese proposal is the presentation of a detailed technical programme involving some 27 projects. Each project is specified in terms of its objectives, particular research and development emphasis, and most projects have quantified targets and schedules. The projects are highly inter-related and the overall programme design is highly pragmatic with a mixture of top-down and bottom-up objectives (Aiso 1982).

The presentation of the technical programme is top-down, with the highest level being a *basic application system*, implemented in *basic software systems*, running on *new advanced architecture*, connected as *distributed function architecture*, and built with *VLSI technology*. Running across levels are *systematization technology projects* for system integration, and the programme is pump-primed through a *development supporting technology* project.

The following sections outline the individual project objectives and schedules, and relate them to other work.

Basic application systems

The projects grouped as application systems are concerned with some of the major commercial uses seen for FGCS and are probably best regarded as test cases that are perceived as driving the lower level technologies to their limits.

Machine translation system

The objective stated (Moto-oka 1982, p. 38) is a general-purpose, multi-lingual translation system handling a vocabulary of 100 000 words with 90 per cent accuracy leaving the remaining 10 per cent to human translators. The overall cost should be 30 per cent or less than human translation.

This application is of great significance to the Japanese since their technological society requires interaction with Western scientific literature that is particularly difficult given the different script and language structure of Japanese. Machine translation (MT) was an area where there were great aspirations even in the first generation era (Weaver 1955). Bar-Hillel was a leading worker in this area in the 1950s but by 1962 was expressing disenchantment not only with progress to date but also with the possibility of future progress:

> It seems now quite certain to some of us, a small but apparently growing minority, that with all the progress made in hardware, programming techniques and linguistic insight, the quality of fully autonomous mechanical translation, even when restricted to scientific or technological material, will never approach that of qualified human translators and that therefore MT will only under very exceptional circumstances be able to compete with human translation (Bar-Hillel 1964, p. 182).

He goes on to give reasons concerned with the human use of their special capabilities:

> A complete automation of the activity is wholly utopian, since the fact that books and papers are usually written for readers with certain background knowledge and an ability for logical deduction and plausible reasoning cannot be over-ridden by even the cleverest utilization of all formal features of a discourse (Bar-Hillel 1964, p. 183).

The use of *background knowledge*, *logical deduction* and *plausible reasoning* are all objective of fifth-generation systems and makes machine translation a particularly interest test of the extent to which they are being achieved. We have made much progress in natural language processing in the 22 years since Bar-Hillel made these remarks (Schank and Colby 1973) but they are valid today, no longer describing an impassable barrier, but certainly a major hurdle. The Japanese objectives are not too demanding at the current state of the art, certainly for technical material. Tucker (1984), for example, describes a practical system, PAHO, for Spanish–English translation and suggests how the use of Wilks' (1979) *preference semantics* and Schank's (1975) *conceptual dependencies* (Carbonell, Cullingford and Gershman 1981) may be used to give truly high-quality translation.

Question answering system

The objective stated (Moto-oka 1982, p. 39) is a general-purpose question answering system for use as an intelligent information service and as part of specialist systems for such applications as computer-aided design and robotics. The five-year target is a specialist system with a vocabulary of 2000 Japanese words and 1000

inference rules and requiring the user to eliminate ambiguity. This will be evaluated and extended to vocabularies of 5000 words involving 10 000 inference rules.

Question answering systems have been a key goal in AI since Turing (1950) proposed the human capability to discriminate question answering by a person from that by a computer program as an operational criterion for the achievement of AI. A major thread in the development of AI has been the increasing power of question answering: Weizenbaum's (1966) ELIZA answers questions with no understanding by generating the bland replies of a supportive psychiatrist; Winograd's (1972) SHRDLU answers linguistically complex questions based on its complete understanding of a very small deterministic domain totally under its control; Harris's (1977) INTELLECT (originally ROBOT) answers linguistically complex questions using information in a database to which it is a front end; Shortliffe's (1976) MYCIN answers questions about a useful domain of medicine with the competence of top experts in that domain; Davis' TEIRESIAS answers questions about MYCIN's inference process and makes suggestions for the improvement of MYCIN's knowledge base using the results (Davis and Lenat 1982).

Question answering systems have progressed through increasing linguistic capabilities based on a knowledge base, through detachment of the language processing shell from the knowledge base, through inferencing based on data in the knowledge base, to inferencing about the inferencing process itself. In these transitions the capability of the system to *explain* its own answers by accepting 'why' and 'how' questions at any stage in the dialogue has become accepted as highly significant. Expert systems that can answer questions only are far less valuable in practice than those that can explain the basis on which they arrived at those answers.

The number of inference rules used in expert systems varies widely: from PUFF, for pulmonary function diagnosis, using 55; through MYCIN, a system for microbial disease diagnosis, using some 400; to R1 (now XCON), a system for configuring computers, using some 2500. Thus the Japanese objectives for question answering systems are not very demanding at the current state of the art. Michie's (1979) *Expert systems in the micro electronic age* is a useful introduction to a wide range of systems, and Hayes-Roth, Waterman and Lenat's (1983) *Building expert systems* covers their construction in some depth. Buchanan and Shortliffe's (1984) *Rule-based expert systems* gives a comprehensive account of the development of MYCIN and related systems. Lehnert's (1978) *The process of question answering* methodically explores the linguistic problems of answering questions using a knowledge base and inferring the questioner's meaning and intentions.

Applied speech understanding system

The objective stated (Moto-oka 1982, p. 40) is a general-purpose

speech synthesis, speech recognition and speaker identification system for use in such applications as a phonetic typewriter, machine translation, and telephone enquiry services. The phonetic typewriter and speech response systems will build up from a vocabulary of a few hundred words to 10 000 words with error correction based on an analysis of the meaning of the text. The speaker identification system has an initial target of about 50 speakers building up to several hundred.

Automatic speech recognition (ASR) is particularly important to the Japanese because of the lack of suitable input devices for the Kanji character set. Japanese, like Chinese, is an ideographic language requiring some 3500 Kanji ideograms. Keyboards have been developed for the printing industry that have groups of 2000 characters on plastic overlays covering a tablet that is touched with a stylus to select a character. Japanese, unlike Chinese, also has phonetic alphabets, the Katakana and Hiragana, that are used to inflect ideograms and represent foreign words. The 50 Katakana characters have been made available on personal computer keyboards and a variety of techniques have been used to use the keyboards also for Kanji, for example using menu selection based on sound or shape. None of these has proved adequate for routine productive use and the FGCS design avoids the use of keyboards altogether. The more natural modes of spoken and handwritten input are seen as essential to the system design.

ASR, like machine translation, was another early aspiration of work on AI and HCI. Bar-Hillel's attack on MT in 1962 was paralleled for ASR by Pierce's (1969) attack on ASR in the *Journal of the Acoustical Society of America*. Fant notes that 'Pierce's challenge seriously affected the respectability of speech recognition at Bell Laboratories and brought about some restrictions in funding elsewhere but there followed a period of upswing through the ARPA projects in this country and a similar endeavor in Japan' (Fant 1975, p. x). The objectives of the ARPA projects were established in 1971 with a goal of achieving real-time connected speech understanding in a constrained domain with a vocabulary of 1000 words and less than 10 per cent error (Newell *et al.* 1973). Five major projects were funded over the period to 1976 and the objectives were comfortably achieved (Wood *et al.* 1976). Some of the techniques developed in these projects concerned with using understanding to aid recognition, such as the *blackboard* for inter-specialist communication in planning of HEARSAY II (Erman, Hayes-Roth, Lesser and Reddy 1980), have become major architectural concepts in the development of knowledge-based systems independent of ASR.

IBM has reported connected speech recognition systems with recognition rates from 67 to 99 per cent for a single speaker using 1000 word vocabularies (Bahl *et al.* 1978). In Japan Sakai and Nakagawa (1977) have reported an average 93 per cent recognition rate for 10 speakers using a 1983 word Japanese vocabulary. Thus the

Japanese objectives for ASR are not very demanding at the current state of the art. The problem of speaker identification for security purposes included in this project is conceptually a different one but can use similar signal processing techniques and the objectives stated are again reasonable (Furui 1981). Lea's (1980) *Trends in speech recognition* and Simon's (1980) *Spoken language generation and understanding* cover the state of ASR research internationally.

Applied picture and image understanding system

The objective stated (Moto-oka 1982, p. 41) is a general-purpose system for the structural storage of picture and image data, its retrieval and intelligent processing. An interim system will handle about 10 000 items and the final system should have a capacity of some 100 000 items, be able to analyse and process an image for storage in a few seconds, and retrieve an item within 0.1 s.

Computer vision was another important objective of early work on AI that was severely limited until recent years by the magnitude of basic signal processing involved. Parallel processing of image data using VLSI components is now making computer vision usable in robotics for parts handling (Edson 1984). The FGCS objectives are narrower than this, emphasizing the kind of image data storage and retrieval that is important for office documents. The main limitations here are as follows: first, *storage*, that an office document digitized at 100 lines per inch requires about 100 kbytes of storage and a photograph about 1 Mbyte; second, *speed of retrieval*, that stored documents have to be fetched to the screen in a second or so, giving a data rate of some 100–1000 kbytes/s; third, *contextual retrieval*, that searching an image database by analysing the content of the images requires retrieval and image processing speeds several orders of magnitude beyond that of retrieval through a directory.

The availability of optical disc storage is making image databases feasible in cartographic, satellite photography and medical applications (Chang and Fu 1980) and database indexing techniques are being developed for such pictorial information systems (Blaser 1980). The Japanese objectives seem reasonable in this context and in such applications as the electronic delivery of documents and graphics (Costigan 1978). The image understanding aspect of the project title is not emphasized in the stated objectives and it seems unlikely that general computer vision will be part of FGCS, although specific tasks such as optical character recognition of printed and handwritten material are already feasible. Hanson and Riseman's (1978) *Computer vision systems* covers research in this area and the techniques involved are presented in Ballard and Brown's (1982) *Computer vision*. Kidode (1983) surveys a range of projects on image processing machines in Japan.

Applied problem solving system

The objective stated (Moto-oka 1982, p. 42) is an extension of game

playing systems that gives an answer to a stated problem. One development will be a Go-playing program with an interim performance of amateur subgrade 10 and eventually amateur grade 1. Another development will be a formula manipulating system that extends the capabilities of MACSYMA.

Problem solving involving search, planning, reasoning and theorem proving has been a major theme in AI (Pearl 1983). It is, however, such a general theme that there is no coherent set of theories or applications that define problem solving. Certainly question answering systems require a wide range or problem-solving techniques but problem solving itself encompasses other areas such as game playing. This fragmentation shows up in the Japanese objectives where a Go-playing program development is coupled with an extension of the MACSYMA (Mathlab Group 1977) system for assisting users to solve mathematical problems including algebraic manipulation. This project does not have the generality and coherence of the others and is probably intended to gather up aspects of FGCS applications not encompassed by them.

Basic software systems

The projects grouped as software systems are concerned with some of the essential software foundations seen for FGCS.

Knowledge-base management system

The objective stated (Moto-oka 1982, p. 43) is a general-purpose knowledge representation, storage, processing and retrieval system.

Database systems have been one of the most important developments in computer science in providing theoretically unified, practical models for information storage and retrieval (Date 1981). A major thrust of the FGCS project is to extend the models to encompass *knowledge bases* and the contrast between data and knowledge is extremely significant, although never explicitly defined. At a technical level the nature of the extension is readily explained. The classical database model assumes that all relevant information to a query is explicitly present in the database as a set of values of fields in tuples defining relations. The only information derivable from this extensional definition is the answer to logical queries about the values in these tuples and their relations. In particular, no inference rules other than those of this relational calculus are defined. The Japanese knowledge-base model extends this definition to include inference rules such as those of MYCIN and other expert systems based on *pattern-directed inference systems* (Waterman and Hayes-Roth 1978).

The use of the term *knowledge* in this way is reasonable given the history in AI of studies of *knowledge representation* (Woods 1983), but it is important to realize that a knowledge-base is still an ill-defined concept. People reason with non-monotonic systems where

the addition of new information can cause a previously valid deduction to become invalid (McDermott and Doyle 1980). They cope with vague or fuzzy information and are able to draw inferences from it (Zadeh 1983), and they are also able to cope with missing or contradictory information (Belnap 1976). Some features of human knowledge processing can be added by extending the logic used beyond standard predicate calculus to encompass modalities of knowledge, belief and time (Snyder 1971) but the formal and computational mechanisms for doing this are still problematic and extensions to vagueness and contradiction are even more difficult (Gaines 1981).

Thus, while knowledge bases are an essential feature of FGCS, the use of the term should not give us a false sense of security about the transition from data to knowledge. Fuchi recognizes this problem in his reply to the following interview question: 'Are you saying that the design of the fifth-generation may be modeled by learning more about the human thinking process?' He answers as follows:

> Yes, we should have more research on human thinking processes, but we already have some basic structures. For more than 2000 years man has tried to find the basic operation of thinking and has established logic. The result is not necessarily sufficient; it's just the one that mankind found. At present we have only one solution—a system like predicate calculus. It is rather similar to the way man thinks. But we need more research. What, really, is going on in our brain? It's a rather difficult problem (Fuchi, Sato and Miller 1984, pp. 14–15).

If the Japanese objectives for knowledge bases are seen as restricted to those aspects of human knowledge processing that can be adequately modelled through the standard predicate calculus, then they are not conceptually demanding at the current state of the art. The open question then remains as to whether the use of the term *knowledge base* is not premature and that something between data and knowledge would be more appropriate, perhaps *data and rule base*. However, as so often happens, a human psychological term has been taken and used in an engineering context prematurely and we will have to live with the consequences. Fuchi (1982) gives a feel for the role of knowledge information processing in FGCS. Bobrow and Collins' (1975) *Representation and understanding* covers research on knowledge representation and Fahlman's (1979) *NETL* illustrates in detail the use of semantics nets. Japanese work on knowledge bases using logic programming is reported in Suwa *et al.* (1982), Miyachi *et al.* (1984) and Nakashima (1984). Gaines (1983) and Quinlan (1983) discuss non-standard logics for knowledge processing.

Problem solving and inference system

The objective is stated (Moto-oka 1982, p. 44) as being to 'constitute the core of the processing functions in the fifth generation computers', and consists of developing 'problem solving systems by establishing a processing model of the problem solving and inference

systems' and explaining 'its processing ability theoretically'. The developments will be concerned with problem-solving and inference algorithms leading to a coding language for problem solving and an inference machine with a capacity of 100–1000 MLIPS,

The emphasis on problem solving and inference at the software level contrasts with the weakness of its specification at the applications level (see page 23), and shows the techniques required to be general-purpose tools underlying all FGCS applications. A coding language for problem solving is the key software objective and Furukawa, Nakajima, Yonezawa, Goto and Royama (1982) describe such a language 'based on predicate logic with extended features of structuring facilities, meta structures and relational data base interfaces'. The Japanese FGCS project has taken standard predicate logic as its basis for problem solving and inference, primarily because it appears to offer the possibility of analysing its 'processing ability theoretically' as required in the objectives. The problems of standard predicate calculus for knowledge representation have been discussed in the previous section.

The project has also assumed that the PROLOG logic programming language is at least an adequate starting point for applications and hardware implementation. The main problems of PROLOG as an adequate logic programming language are as follows: current language implementations use depth-first search and hence are critically dependent on the order in which assertions are processed; and PROLOG does not distinguish between negation as failure and logical negation, between not knowing X and X not being true. The first problem is inadequately treated by the *cut* facility in PROLOG, which allows the programmer to control the search. It is being overcome by the development of breadth-first implementations, termed *concurrent* (Shapiro 1983) because assertions are processed non-deterministically. The second problem requires a logic programming implementation based on more general theorem-proving techniques than resolution such as natural deduction (Hansson, Haridi and Tarnlund 1982), although some semantics for negative statements can be incorporated in PROLOG (Sakai and Miyachi 1983; Aida, Tanaka and Moto-oka 1983).

The development of problem-solving and inference systems based on logic programming in standard predicate calculus is a reasonable one at the present state of the art. There are many technical and implementation problems to be overcome and it seems likely that development of more powerful theorem-proving techniques will be necessary, as well as the non-determinism of depth-first search. More powerful logics will probably be developed, initially through interpreters of their definitions expressed in standard predicate calculus and then embedded in the language once their validity and utility have been determined. For example, Stickel (1984) shows how PROLOG may be extended to be a logically complete theorem prover. Kowalski's (1979) *Logic for problem solving* introduces logic

programming and Clark and Tarnlund's (1982) *Logic programming* covers many aspects of implementation and application.

Intelligent interface system

The objectives stated (Moto-oka 1982, pp. 45–46) are general-purpose human/computer interface systems capable of flexible conversation in natural language, pictures and images, and eliminating the 'language gap between the user and his computer'. One development is of a natural language system covering 'computer and one branch of scientific and technological terminology' which adapts itself to speakers and communicates with unspecified speakers in Japanese and English on a real-time basis. Another development is to handle about 70 per cent of the picture and image data required in applications involving machine drawing or medical photographs in such a way as 'to enable smooth user interaction with the computer through image and picture media'.

The intelligent interface system objectives are to provide the software support for such applications as those of question answering, speech understanding, and image understanding. The detailed description of this project (Tanaka *et al*, 1982*b*) emphasizes language processing and graphics editing. The objectives seem reasonable in the light of work on natural language information processing that encompass the structure of English in great detail in computational form (Sager 1981; Harris 1982) and implementations of comprehensive natural language interfaces on comparatively small machines such as ASK (Thompson, Thompson and Ho 1983). The human factors of graphic interaction also have become better understood through the result of experience with the icon, mouse and window systems of the Xerox Star (Smith, Irby, Kimball, Verplank and Harslem, 1983).

The notion of improved HCI through intelligent interfaces has already gone beyond that expressed in the Japanese objectives to encompass intelligent help systems (Rissland 1984) based on inference of the user's intentions (Norman 1984). There has been an evolution of HCI techniques for dialogue engineering computer conversations that spans formal dialogues, forms, menus, graphic interaction, natural language and the *why* and *how* accountability of expert systems (Gaines and Shaw 1984*a*). Systemic, psychological and computational principles are being developed that underlie effective HCI in fifth-generation computer systems and can be implemented as dialogue shells interfacing to a variety of facilities (Gaines and Shaw 1984*b*). The main weakness in system design currently is in our knowledge of multi-modal interaction where voice and gesture inputs are used in combination with audio and multi-screen visual outputs. Such multi-modal interaction will become available at low cost in FGCSs and create both opportunities and problems in effective HCI. Coombs and Alty's (1981) *Computing skills and the user interface* and Sime and Coombs' (1983) *Designing*

for human–computer interaction, cover developments in HCI techniques.

New advanced architecture

The projects grouped as advanced architecture are concerned with some of the essential hardware foundations seen for FGCSs.

Logic programming system

The objective stated (Moto-oka 1982, p. 47) is the development of the architectures necessary to support inferences using a computational model based on predicate logic with a power of expression approximating natural languages. Three machines are targeted: a firmware base machine with a speed of 0.1 MLIPS; a personal logic programming machine at 0.1–1.0 MLIPS; and a parallel logic programming machine at 50–1000 MLIPS. Two language developments are targeted: extended PROLOG; and a new logic programming language.

The implementation of logic programming machines is at the heart of that part of the FGCS programme that is being developed at ICOT in Tokyo (Uchida *et al.* 1982; Yokoi *et al.* 1982). This emphasizes the significance of the bottom-up approach in the actual development programme despite its ostensible top-down, market driven presentation. The availability of VLSI to implement parallel processing logic programming computers is a major driving force for the actual activities being undertaken. The sequential inference machine (SIM) which became operational in December 1983 is built around a microprogrammed sequential processor in an IEEE 796 bus supporting up to 80 Mbytes of main memory and up to 63 processes. It has a 1200×912 pixel bit-map display, two 38 Mbyte fixed head discs and two 1 Mbyte floppy discs. There is a 40-bit data path with 8 bits used as tags and 8K words of cache memory. The control memory consists of 16K words 64 bits wide and contains an interpreter executing Kernel Language 0 (Chikayama 1983). The instruction set includes operations on arrays and character strings as well as inferencing functions for unification and backtracking. The machine has a 200 ns cycle time and executes PROLOG at about 30 KLIPS; it was built by Mitsubishi.

The next machine to be built is a parallel inference machine (PIM) executing Kernel Language 1 and using dataflow techniques to achieve an eventual speed of 1 GLIPS (Kawanobe 1984). Parallelism is natural in logic programming for goals expressed as A OR B which can be sought out independently. It is more problematic for A AND B where both goals must be satisfied simultaneously by a consistent assignment of values (unification). Unification parallelism is also possible in which multiple sets of value assignments are tested at the same time. Moto-oka and Stone (1984) have reviewed these possibilities and problems and they are key topics in logic programming research (Nakagawa 1984).

Functional machine

The objective stated (Moto-oka 1982, p. 48) is the development of architectures to support a functional model and programming language suitable for symbol manipulation. The programme calls for work on LISP and possible alternative functional languages, and the design of suitable machines using parallel and associative processing. Three machines are targeted: a personal LISP machine at two to three times general-purpose computer speed (4 MIPS); a parallel reduction machine at 10 times general-purpose computer speed; and a data flow machine at several hundred times general-purpose computer speed.

While the FGCS project has a commitment to logic programming and PROLOG machines, this has not excluded objectives concerned with what may be seen as the competing technologies of functional programming and LISP machines. These are already well established, with several manufacturers already making LISP machines based on the original developments at MIT (Bawden *et al.* 1979; Weinreb and Moon 1981; Manuel 1983). It is possible to implement PROLOG within LISP (Robinson and Sibert 1982) and attractive to gain access to many features of the LISP programming environment in this way, so the PROLOG/LISP competition may be far more apparent than real. The architectural requirements of LISP and PROLOG machines are also similar so that it seems likely that this project will combine with the previous one in terms of hardware.

Relational algebra machine

The objective stated (Moto-oka 1982, p. 49) is the development of architectures with particular attention to the memory hierarchy to handle set operations using relational algebra as a basis for database systems. The interim target is a machine with up to 100 processors and the final target one with 500–600 processors. Three storage capacities are targeted: small for high speed operations, 10–100 Mbit/s; medium for medium and high speed operations, 100 Mbit/s to 10 Gbit/s; large for low and medium speed operations, 10–1000 Gbits/s.

Relational database machines are seen as underlying the knowledge bases of FGCS. Their hardware implementation by dataflow techniques has already been extensively studied (Boral and DeWitt 1982; Glinz 1983; Kubera and Malms 1983) and the Japanese objectives seem reasonable. The machine *Delta* being developed at ICOT is intended to connect through a local area network with a number of SIMSs (see p. 28). Its functional design has been described in some detail (Tanaka *et al.* 1982*a*; Kakatu, Miyazaki, Shibayama, Yokota and Murakami 1983) and a related development is taking place at the University of Tokyo (Kitsuregawa, Tanaka and Moto-oka 1983) on high-speed computation of relational joins. Leilich and Missikoff's (1983) *Database machines* covers hardware implementations.

Abstract data type support machine

The objective stated (Moto-oka 1982, p. 50) is the development of memory structures and processor functions to support modularity in very large and complex systems. The interim target is to develop a machine equivalent to 100 parallel von Neumann abstract data type support machines, and the final target is 1000.

The use of VLSI to allow the design of computer architectures giving increased ease of programming through enforced modularity has been a topic of increasing interest (Myers 1978). Fabry (1974) suggested a *capability-based* addressing scheme which enforced protection and modularity in a simple, flexible way and this has been used in a number of machines including the Intel 432 microprocessor (Tyner 1981). The Japanese objectives are reasonable and it will be interesting to see if this project links in with the other FGCS developments.

Data flow machine

The objective stated (Moto-oka 1982, p. 51) is the development of a parallel processing architecture based on the data flow model with particular attention to the requirements for logic programming and functional machines. The targets are as follows: initial, 16 processors with 8 Mbit memory; interim, 100 processors with 100 Mbit memory and achieving 50 MIPS;* final, 1000–10 000 processors with 1–10 Gbit and 1–10 BIPS.

The use of data flow techniques for database machines has been discussed above. In Japan Mushanino Laboratory is working on a LISP machine, Eddy, using data flow techniques (*Electronics* 1983*b*). In the USA Denning commenced work on a data flow machine in 1967 and a wide range of developments are now under way (*Electronics* 1983*a*).

Innovative von Neumann machine

The objective stated (Moto-oka 1982, p. 52) is the development of a innovative von Neumann machine taking advantage of VLSI, such as an object-orientated architecture. The targets are as follows: interim, a processor with 1 million transistors on a chip; final, 10 million transistors.

Apart from storage, the major use of VLSI has been to make available increasingly powerful computers on a chip from the 2300 device Intel 4004 in 1971 to the 450 000 device Hewlett Packard 9000 in 1981. The most innovative von Neumann architecture to date has been the Intel iAPX432 in 1981 which, as discussed earlier, gives a capability-based processor able to combine modularity and protection. It has not found widespread use because of its low speed and lack of supporting software. However, the concepts behind the 432

* 1 MIPS = 1 million instructions per second; 1 BIPS = 1 billion instructions per second.

are attractive and the underlying software concepts have been developed extensively by Goldberg and his collaborators, giving rise to the *object-orientated* architecture of *Smalltalk* (Goldberg and Robson 1983; Krasner 1984; Goldberg 1984). The object-architecture requirements are now sufficiently well defined for a hardware implementation to be straightforward. The 1 million transistor objective is feasible now and a Moore's law (see p. 6) calculation indicates that 10 million should be achieved by 1989.

Distributed function architecture

The projects grouped as distributed function architecture are concerned with some of the essential organizational foundations seen for FGCSs, in particular the use of large numbers of coupled powerful VLSI processing units.

Distributed function architecture

The objective stated (Moto-oka 1982, p. 53) is the development of a distributed function architecture giving high efficiency, high reliability, simple construction, ease of use, and adaptability to future technologies and different system levels.

Modularity is attractive in both hardware and software because it decouples sub-system developments and allows system modification and upgrade without total re-design. FGCSs are coming at a time when the pace of change of technology has accelerated to a level where a design is obsolete by the time it is complete, let alone prototyped and into manufacture. We have to design FGCSs on the assumption that the underlying technologies, sub-systems, processing concepts, and so on, will all change during this generation. The characteristic feature of FGCSs is at the very high level of moving from information processing to knowledge processing. How this move is implemented at lower levels is not crucial to FGCS concepts. Thus, a modular distributed function architecture is a key design philosophy for FGCSs. The availability of low-cost powerful microprocessors has already made it attractive to make system components stand alone in their capabilities. Multiple microprocessor systems have been developed for a wide range of applications in many different configurations (Fathi and Krieger 1983). Weitzman's (1980) *Distributed micro/minicomputer systems* covers the main concepts and developments.

Network architecture

The objective stated (Moto-oka 1982, p. 54) is the development of loosely coupled computer systems using high-speed local and global networks. This will involve standardization of network architectures and protocols and the use of VLSI, optical fibre, satellite communications and local area network technologies.

The development of high-speed digital communications networks

allowing computers to be linked locally or remotely is now a core component of computer system development and has broken down the final barriers between the computer and communications industries. In the USA in particular the deregulation of the communications industry, itself a logical consequence of this convergence, has allowed AT&T and IBM to come in direct competition with one another. The main limitation on the development of systems around digital networks has been lack of standards, and governments and industry have made immense efforts to create internationally agreed specifications for networking. The ultimate evolution is seen to be towards the *integrated services digital network* (ISDN) carrying telephone, television, information utilities and computer communications (Kostas 1984) and standards for this are evolving (Bhusri 1984). For *local area networks* the hierarchical *open systems interconnection* ISO standard provides the rules for any digital equipment to 'join the club' on a local network with mutual compatibility (Day and Zimmerman 1983). FGCS developments do not specifically impact the network architecture but have to be completely compatible with it if they are to serve as interfaces from users into the information and communication utilities of the 1990s.

Database machine

The objective stated (Moto-oka 1982, p. 55) is the development of a special-purpose machine for high-speed access to large capacity databases. The targets are as follows: experimental, relational database machine with a capacity of 100 Gbytes and 1000 transactions a second; practical, 1000 Gbytes and 10 000 transactions a second.

The implementation relational database machine using data flow techniques is covered on pp. 29–30. There have also been developments of more conventional high-speed database machines using a special-purpose processor for each head on a disc. Boral and DeWitt (1983) give a critical survey of such machines and point out that current mass storage devices do not provide sufficient input/output bandwidth to justify highly parallel database machines. In terms of their discussion and simulation results the transaction rates targeted by the Japanese appear reasonable and do not require special-purpose hardware. The capacities projected are large but a reasonable projection for the 1990s, although the costs of this amount of high speed storage will probably still be very high at that time unless major breakthroughs occur with new storage techniques.

High-speed numerical computation machine

The objective stated (Moto-oka 1982, p. 56) is the development of a special-purpose machine for high-speed scientific and technical computation. Processor elements with 40–100 MFLOPS will be developed based on new high-speed devices, together with a parallel processing system based on a thousand 4 MLOPS processors. These will be coupled to a head per track disc of 50–60 Gbytes.

The relation of the number-crunching supercomputer to the knowledge-crunching FGCS is a very interesting one. The underlying technologies are very similar and, for example, LISP machines offer very powerful arithmetic capabilities such as numeric array operations (Weinreb and Moon 1981). At the level of applications the machines seem very different and the majority of expert system developments involve little numeric data-processing. This is because the knowledge encoded is that elicited from an expert and is primarily 'what' to do rather than 'why' to do it. In medicine, for example, the potential user is not seeking the deep knowledge of the underlying physical processes of particular illnesses but the superficial knowledge of how to treat patients according to their symptoms. This model-free approach has been successful in many applications where the underlying processes are not well understood but people have developed skills in dealing with them. The current generation of expert systems reverse the normal scientific paradigm and model the person doing the task, not the task itself. A combination of both approaches is more powerful than either alone and expert systems will increasingly combine simulation and knowledge processing. Thus numerical supercomputers will form part of FGCS as necessary resources even though their development is not central to the basic FGCS concept.

High-level man–machine communication system

The objective stated (Moto-oka 1982, p. 57) is the development of a system to input and output characters, speech, pictures and images and interact intelligently with the user. The character input/output targets are as follows: interim, 3000–4000 Chinese characters in four to five typefaces; final, speech input of characters, and translation between Kana and Kanji characters. The picture input/output targets are: interim, input tablet 5000 × 5000 to 10 000 × 10 000 resolution elements; final, intelligent processing of graphic input. The speech input/output targets are: interim, identify 500–1000 words; final, intelligent processing of speech input. It is also intended to integrate these facilities into multi-model personal computer terminals.

This project encompasses all the peripheral and signal processing technologies necessary to support the speech understanding system (see p. 21), the image understanding system (see p. 23) and the intelligent interface system (see p. 27). As noted in these sections, the speech recognition objectives are reasonable in terms of the state of the art, and systems have been developed for recognizing Japanese (Mizoguchi and Kakusho 1984). Techniques for graphic interaction also have developed in recent years and handwritten input in particular can now be processed intelligently (Gaines *et al.* 1984), for example in cleaning up drawings (Tudhope and Oldfield 1983) and character recognition (Suen, Berthod and Mori 1980), including Japanese (Yamamoto, Yamada and Oka 1984).

VLSI technology

The projects grouped as VLSI technology are concerned with the underlying circuit developments seen as essential to FGCSs.

VLSI architecture

The objective stated (Moto-oka 1982, p. 58) is the development of architectures to make full use of VLSI with 10 million transistors per chip by 1990.

The 10 million transistors a chip requirement is one that is reasonable by 1989 and assumed by several of the other projects. The Japanese have become world leaders in advanced VLSI technology (Galinski 1983) and, as was argued earlier in this paper, it is the technology push of VLSI that underlies the overall FGCS programme.

Intelligent VLSI–CAD system

The objective stated (Moto-oka 1982, p. 59) is the development of a VLSI computer-aided design system storing and using design knowledge such that a custom chip with 1 million devices can be designed in one month and made available in three.

The availability of VLSI and the capability to make use of it are two distinct issues. In recent years the masking costs of integrated circuits have come down and we have passed the crossover point where the cost of chip development is now dominated by the labour cost for its design. As chips hold more devices the problem becomes worse and the development of CAD tools for VLSI has become increasingly important (Mead and Conway 1980). The *silicon compiler* (see p. 6) has become as important as the software compiler and has proved its worth in such developments as DEC's microVAX chip (Fields 1983). VLSI–CAD systems are fundamental to FGCSs but not specific to the programme, and are being developed for general VLSI use (Sakamura, Sekino, Kodaka, Vehara and Aiso 1982). Rabbat's (1983) *Hardware and software concepts in VLSI* covers key activities in VLSI–CAD world-wide. Ullman's (1984) *Computational aspects of VLSI* and Uhr's (1984) *Algorithm-structured computer arrays and networks* show the interaction of VLSI–CAD and developments in computer architecutre.

Systematization technology

The projects grouped as 'systematization' technology are concerned with the system integration of all the software, hardware and circuit developments seen as essential to FGCSs.

Intelligent programming system

The objective stated (Moto-oka 1982, p. 60) is the development of a complete systems development system that uses inference to design a

program that satisfies stated specifications, synthesizes it by fetching programs from an algorithm bank, and verifies the result. The targets are: a system for the program base, synthesis and verification; a system to maintain, improve and manage programs; a consultant system for program design.

The development of high productivity programming environments is currently one of the major goals of the computer industry (Boehm and Standish 1983). It is assumed to be an area where AI techniques will contribute by providing *participating programming assistants* (Balzer, Cheatham and Green 1983), and it is natural to assume that FGCSs will use knowledge-based techniques to support their own programming. This has already happened in expert system development, where Davis' debugging system for MYCIN was encoded in the same rule-based framework as MYCIN itself but operating on *meta-knowledge* (Davis and Lenat 1982). In FGCSs users will become increasingly unaware of programming as a separate activity. Conversational interaction with the system will result in changes to its behaviour that may be viewed either as resulting from changes in its knowledge or its programming (Gaines 1978). Wegner's (1979) *Research directions in software technology* covers developments in high productivity programming environments.

Knowledge-base design system

The objective stated (Moto-oka 1982, p. 61) is the development of an organically contained knowledge-base which stores the technical data and knowledge required to design, develop and operate a knowledge information processing system.

One of the most difficult problems with knowledge-based systems is what Feigenbaum (1980) terms *knowledge engineering*, the reduction of a large body of knowledge to a precise set of facts and rules. As Hayes-Roth *et al.* (1983) note:

> Knowledge acquisition is a bottleneck in the construction of expert systems. The knowledge engineer's job is to act as a go-between to help an expert build a system. Since the knowledge engineer has far less knowledge of the domain than the expert, however, communication problems impede the process of transferring expertise into a program. The vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem-solving; thus the knowledge engineer and expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalize the domain concepts.

Expert system studies have resulted in a number of reasonably domain-independent software support systems for the encoding and application of knowledge (Michie 1979). However, we need to understand more about the nature of expertise in itself (Hawkins 1983) and to be able to apply this knowledge to the elicitation of expertise in specific domains. The problem of knowledge elicitation from a skilled person is well known in the literature of psychology.

Bainbridge (1979) has reviewed the difficulties of verbal de-debriefing and notes that there is no necessary correlation between verbal reports and mental behaviour, and that many psychologists feel strongly that verbal data are useless. However, this remark must be taken in the context of experimental psychologists working within a positivist, behavioural paradigm. Other schools of psychology have developed techniques for making use of verbal interaction, for example through interviewing techniques that attempt to by-pass cognitive defences, including those resulting from automization of skilled behaviour. Welbank (1983) has reviewed many of these techniques in the context of knowledge engineering.

Knowledge engineering will be as important in the fifth-generation era as software engineering became in the fourth. If there is one major area of weakness in the FGCS development then this is it. We are relying on our capability to create tools for harnessing knowledge-processing technology in the same way that we generate software tools for harnessing information-processing technology. What automata theory provided in the 1970s cognitive psychology must provide in the 1980s. We have moved out of the domain of mathematics into the domain of the mind.

Systematization technology for computer architecture

The objective stated (Moto-oka 1982, p. 62) is the development of techniques to build virtual and real systems on a large scale with high reliability and to optimize them for system configuration and load balance.

Automatic system design (Breuer 1975) is another fourth-generation objective where progress will be greatly accelerated by fifth-generation methodologies. Knowledge-based systems have already proved their value in integrated circuit design (Lenat, Sutherland and Gibbons 1982) and the configuration of computers (McDermott 1981). Coupled with simulation facilities and graphics they will make it possible to design, evaluate and interact with a wide range of systems from any design discipline. Computer architecture will be just one of the many applications of intelligent CAD systems.

Databases and distributed database systems

The objective stated (Moto-oka 1982, p. 63) is the development of a database system for fifth-generation computers, integrating one or more databases and knowledge-bases.

The integration of the databases of fourth-generation computing with the knowledge-bases of fifth-generation computing (Amamiya, Hakozaki, Yokoi, Fusaoka and Tanaka 1982) will be a major task of great commercial significance. We can see some aspects of this already in such systems as Harris' (1984) INTELLECT for natural language access to databases. If the database is the dominant entity then the knowledge-base is primarily concerned with interpreting the query language and the problem is soluble now. As the knowledge-

base itself becomes more significant then we move over to expert systems technology and the techniques of Davis' TEIRESIAS (Davis and Lenat 1982).

Development-supporting technology

The FGCS development programme is seen as bootstrapping itself through a number of phases of system development, each of which provides tools for the next phase. The initial supporting activities are grouped as a separate project.

Development support system

The objective stated (Moto-oka 1982, p. 62) is the construction at an early stage of the project of VLSI–CAD, personal computers, computer networks, and systems to support the development of software and knowledge-bases.

Computer systems are essential tools in designing and developing computer systems. The ICOT project team has purchased a number of LISP machines and made the rapid development of PROLOG and database machines a first priority in the programme. The concept of *software tools* has become a familiar one in the Unix operating environment, and one outcome of FGCS developments will be a family of tools for the new knowledge-based system environments.

The realism of the FGCS programme

It has been common in articles on the Japanese FGCS programme to take it for granted that is a 'high-risk' and 'speculative' research programme. When a detailed analysis is made of the actual objectives, however, all of them appear eminently reasonable and totally consistent with the known state of the art. FGCSs were already upon us at the beginning of the 1980s and the Japanese programme is more descriptive of the inevitable direction of development than it is prescriptive of a radical new direction. What is missing, and perhaps speculative and high-risk, is what are the applications of this new technology. It has been argued earlier in this article that the consumer marketplace is necessary to exploit VLSI capability and that this dictates extremely user-friendly technology. However, the question of what role the friendly FGCS would play was left open. This is still an open question that, in the rush and excitment of the technology, we are still neglecting to ask. The following sections attempt to develop a framework for answering this question.

Social impact of FGCS: an information age

The emphasis of the Japanese FGCS project on social consequences 'to the benefit of all humankind' has already been noted. Hajime Karatsu, Chairman of the Social Needs Sub-Committee of ICOT,

lists a range of priorities for FGCS applications: (*a*) expectation for improvement in the fields of low productivity, such as the office, education, public service and government; (*b*) internationalization of Japan, the transfer of Japanese experience in high technology to other countries while retaining a competitive edge in some key areas; (*c*) shortage of energy and natural resources, optimization of energy consumption and search for new resources; (*d*) high age and high education, coping with an aged and educated society, a problem for all developed nations but particularly poignant for Japan with its extreme respect for the aged; (*e*) information society and human beings, the mis-match between much of modern technology and the needs of people (Karatsu 1982). While FGCSs are presented as contributing social benefits primarily because they are more naturally accessible to people, it is an open question what their impact will be.

For many years we have been told that we are on the brink of an *information age* (Dizard 1982), a *post-industrial society* (Bell 1973), in which we shall establish different relationships to one another, and to our environment, based on computer and communications technology. Toffler advances the concept of a *third wave* following previous agricultural and industrial waves: 'Without clearly recognizing it, we are engaged in building a remarkable new civilization from the ground up .... It is, at one and the same time, highly technological and anti-industrial' (Toffler 1980, p. 10). He rationalizes much of the turmoil around the world as 'agonies of transition' between two civilizations and, in relation to the individual, notes:

> Caught in the crack-up of the old, with the new system not yet in place, millions find the higher level of diversity bewildering rather than helpful. Instead of being liberated, they suffer from overchoice and are wounded, embittered, plunged into sorrow and loneliness intensified by the very multiplicity of their options (Toffler 1980, p. 224).

This emphasis on choice presents a different view of the problems of the information age from those we have come to expect, that computers will be used to restrict our freedom or replace us in the work-place. In *The conquest of will*, Mowshowitz fears that computer-based information systems will lead to '... the alienation of individual responsibility which results from excessive bureaucratization of decision-making, (Mowshowitz 1976, p. x), and Laudon, in *Computers and bureaucratic reform*, while noting the limited impact of the computer, concludes: 'the information systems we describe ... illustrate the use of a new technology to further the political aims and interests of established and limited groups in society' (Laudon 1974, p. 311).

Fears about job-displacement go back to the early years of computing. Wiener's prophetic statements of 1950 on *The human use of human beings* are uncomfortably close to reality today:

> the automatic machine ... is the precise economic equivalent of slave labor. Any labor which competes with slave labor must accept the economic

conditions of slave labor. It is perfectly clear that this will now produce an unemployment situation, in comparison with which the present recession and even the depression of the thirties will seem a pleasant joke (Wiener 1950, p. 189).

He is also concerned about political abuse and quotes a Dominican friar, Dubarle, reviewing his earlier book, *Cybernetics* in *le Monde*, 28 December 1948:

conceive a State apparatus covering all systems of political decisions . . . . In comparison with this, Hobbes' *Leviathan* was nothing but a pleasant joke. We are running the risk nowadays of a great World State, where deliberate and conscious primitive injustice may be the only possible condition for the statistical happiness of the masses: a world worse than hell for every clear mind (Wiener 1950, p. 209).

Wiener discounts Dubarle's fears but notes the 'real danger . . . that such machines, though helpless by themselves, may be used by a human being or a block of human beings to increase their control over the rest of the human race' (Wiener 1950, p. 212), a fear which is prevalent today and is leading, or has led, to highly restrictive legislation governing the access to, and use of, data kept on computers.

A Luddite movement against computers is also conceivable and Weizenbaum's book on *Computer power and human reason* has much in it to excite such a response: 'Science promised man power. But, as so often happens when people are seduced by promises of power . . . the price actually paid is servitude and impotence' (Weizenbaum 1976). These words echo Butler's remarks of nearly a century ago about a previous era of high technology: 'The servant glides by imperceptible approaches into the master; and we have come to such a pass that, even now, man must suffer terribly on ceasing to benefit the machines' (Butler 1872). More recently, Brod in *Techno stress* has pointed to the human cost of the computer revolution in terms of personal life, noting that 'our devotion to the new machine prevents us from seeing the possible consequences of spending long hours—in work and at play—with a machine' (Brod 1984, p. 3).

However, is information technology a cause of social problems or part of society's response to solving them? We live in an increasingly over-crowded world where resources are stretched to their limits and all aspects of our existences have become interdependent. The *Club of Rome* reports have drawn attention to the perils involved (Peccei 1982), and there are technologies, such as bio-engineering, that pose far greater threats than does computing (Cherfas 1982). Information technology may be necessary to our survival and the problems that we attribute to it may be side-effects of coping with increasingly complex world dynamics. Thus our concerns about information technology must be set in the context of a model of society as a whole if we are to begin to discern cause and effect.

The German sociologist Luhmann in his work *Trust and power*

encapsulates the dynamics of society by postulating *complexity-reduction* as the fundamental motivation for all our social institutions:

> The world is overwhelmingly complex for every kind of real system . . . Its possibilities exceed those to which the system has the capacity to respond. A system locates itself in a selectively constituted 'environment' and will disintegrate in the case of disjunction between environment and 'world'. Human beings, however, and they alone, are conscious of the world's complexity and therefore of the possibility of selecting their environment—something which poses fundamental questions of self-preservation. Man has the capacity to comprehend the world, can see alternatives, possibilities, can realize his own ignorance, and can perceive himself as one who must make decisions (Luhmann 1979, p. 6).

Luhmann's model seems to underlie De Bono's optimism about the role of computers in *Future positive*:

> By great good fortune, and just in time, we have to hand a device that can rescue us from the mass of complexity. That device is the computer. The computer will be to the organization revolution what steam power was to the industrial revolution. The computer can extend our organising power in the same way as steam extended muscle power . . . . Of course we have to ensure that the result is more human rather than less human. Similarly we have to use the computer to reduce complexity rather than to increase complexity, by making it possible to cope with increased complexity (De Bono 1979, pp. 18–19).

It is into this debate between pessimism and optimism, between added complexity and controlled complexity, that the next stage of computer development has been introduced: the FGCS that will be ubiquitous, accessed and programmed by all. It is a development that cannot be understood outside this social context because it is both formed by it and a major force in forming it. The next section outlines a framework for analysing this social context and the relationship of computer developments to human needs.

Social determinants—a hierarchy of needs

All of the preceding arguments extrapolate trends, whether in computing or in media, to provide a framework for FGCSs. They do not provide an underlying causal model that might indicate how the technology will be used: in commercial terms where the markets will be; in social terms what the consequences will be. One possible perspective from which to view the causal dynamics is that of Maslow's (1971) hierarchy of personal needs which gives the dynamics of individual motivations and priorities. The logic of his hierarchy is that upper level needs are of low priority until lower level ones are satisfied. We need to satisfy basic biological needs before we are concerned with safety, and only when we are safe are we concerned with belonging and so on.

Maslow's theory is subject to debate (Lederer 1980) but it gives a logic for the social infrastructures that support individual needs. The left-hand column of Table 4 shows his hierarchy of needs, and the

Table 4. The hierarchy of personal needs, its social support and the role of the computer

Individuals	Socio-economic infrastructure	Computers
Self-actualization	Realizing personal potential; facing life as it is; aesthetics; peak experiences	Tools in hands of individuals to give new perspectives
Esteem	Role in family, work and community; other recognized achievements (artistic, horticultural)	Tools in hands of individuals to perform role better
Belonging	Family, work, religion, politics, entertainment	Networking, bulletin boards, games
Safety	Social norms, police, military, medicine, insurance	Command and control, crisis decision-making
Basic biological needs	Agriculture, energy, housing, ecology, finance, physical communications	Planning, management and control

centre column shows those social systems that have been set up to aid the individual in satisfying his needs. The right-hand column shown the roles that computers have played in supporting these social systems. The two lowest levels account for the impetus behind the first three generations of computers. They have been used commercially to support the industries necessary to basic biological needs and militarily to support safety. The levels above are all new markets that have opened up only with the fourth generation and personal computers. It is these levels that provide the new markets for FGCSs as they come to saturate those at the lower levels.

The hierarchy of needs model has some interesting consequences. It suggests a divergence between developed world and third world priorities in the use of computers. For countries where food and energy crises still exist the use of computers in agriculture, energy, communications and so on will be the main priority. For the developed nations the use at the lowest levels is taken for granted and the levels of belonging and esteem are where the fifth-generation markets lie. The logic at these levels is also significant in the analysis of job displacement. Esteem is enhanced if computers provide the capability to do a task more effectively but not if the perceived role is undermined, e.g. to that of servant to a machine. This is the conflict between those who see the computer as a tool to enhance life and those who see it as destroying the social fabric.

FGCSs will impact society but they are also a product of that

society and its needs. We cannot understand their social impact if we view them only as agents of change and not also as a response to change. The changes wrought by computer technology are superficial and reflect far deeper underlying changes in society itself. The emphasis on HCI in FGCSs is an indication of our need to control computers and use them to support our humanity rather than be controlled and de-humanized by an increasingly technological society.

Computing as a new medium

In viewing the key FGCS markets as being at the upper levels of the Maslow hierarchy we gain a new perspective on computers that sees them as a new *medium for communication* (Gaines and Shaw 1984*a*; Shaw and Gaines 1984). We can test this analogy by asking whether there is anything fundamentally new in expert systems. Has not the purpose of work on computer systems always been to encode human expertise for later reproduction? An accountancy program may be seen as the recording of the expertise of an accountant for use by a businessman. An auditor evaluating the program may expect to be able to ask it those questions he would ask the accountant: for example, from what original information did you calculate this figure? If the program does not have built into it audit trail facilities which enable it to answer such questions, then it is inadequate in exactly the same way that an accountant would be who could not answer that question—it is a simulation of an incompetent accountant. This is more illuminating than just describing it as a poor program.

In retrospect we may see expert systems as being a more accurate term for what we have so far called computer science; a name that defines the objectives rather than the tool used to achieve it. After all we do not have pencil science but rather literature. The change of name is very significant, however, because it draws attention to the nature of the computer as a medium to encode and reproduce expertise. Media developments have played major roles in the growth of society and culture. It also helps to explain some of the phenomena associated with personal computing: for example that programs are being marketed using the same techniques that have been effective for programme material in previous media such as books and gramophone records.

The computer differs from previous media in being two-way and interactive. Books and television are primarily one-way media. They are not able to support two-way conversations. They do not allow us to interact with the author of the book or with the events being portrayed. We can think of computer programs as being a way of extending *interaction* through time and space. When I listen to a gramophone record I can imagine I am at a concert hearing the artist perform. When I read a book I can imagine that I am hearing the words spoken to me by the original author, or that I am seeing and

hearing what he describes. Similarly, when I enter into a dialogue with a computer program I can imagine that I am interacting with a person or a world which the programmer has simulated. Unlike the performer or the author, the simulation is able to provide responses to my interventions, in some way to *interact* with me although the person simulated may be far distant or dead, or may never have existed.

Figure 3 shows how the computer differs from previous media in being two-way and interactive. Books and television are primarily one-way media; they are not able to support two-way conversations. They do not allow us to interact with the author of the book or with the events being portrayed. This emphasis on interactive simulation ties in with the notion of virtuality in computing. This is an important technical concept in third- and fourth-generation systems (Weegenaar 1978) annd Nelson (1980) has suggested that it underlies all applications of computing, that we simulate a virtual world. Smith

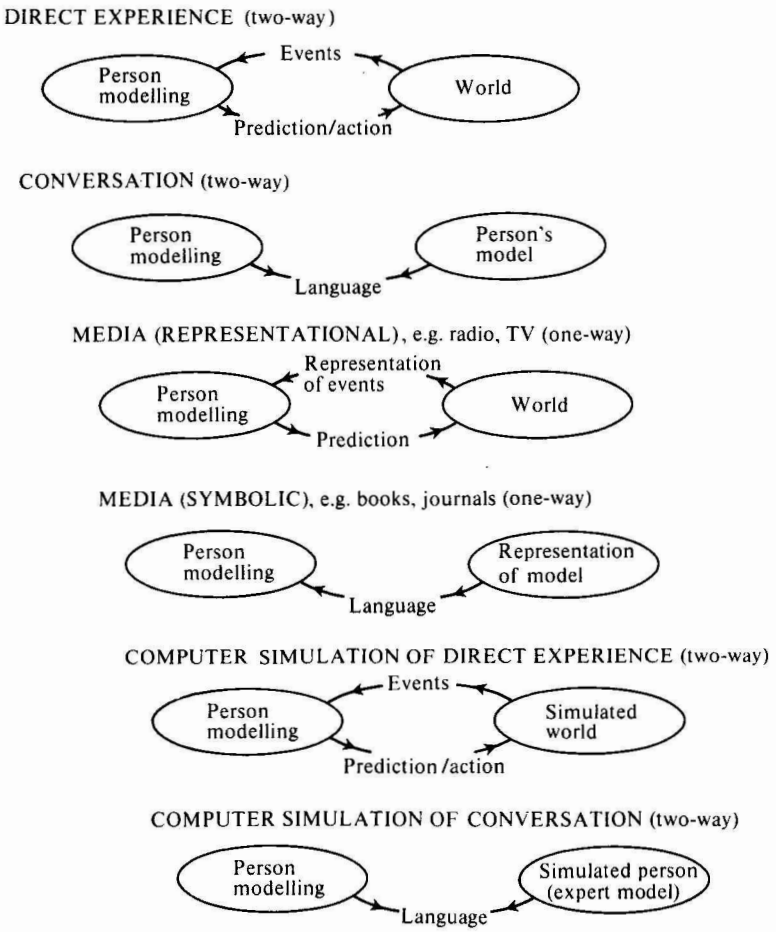

Fig. 3. Computing in the evolution of media: an interactive medium for encoding simulated worlds and people.

made it the key design concept of his *Pygmalion* system (Smith 1977) and later the Xerox Star (Smith, Irby, Kimball, Verplank and Harslem 1983). Media reify the world of knowledge and create an ecology of knowledge (Wojciechowski 1983) within the knowledge environment (Gaines and Shaw 1983*a*, *b*). Thus, media are vital to the knowledge economy (Machlup 1980) in enabling knowledge products to be instantiated in concrete form, and the computer outforms all other media by instantiating action as well as facts.

FGCS developments will complete the new computing medium and make it comparable in all its capabilities with other media such as television. All new media have had initial limitations and gone through phases of improvement and development. The wax cylinder gramophone could not sustain the record industry of today and neither could Baird's spinning disc support television as we now know it. The personal computers of today already provide an impressive new conversational medium for entertainment, education and business. However, they cannot yet compete with television in their audio and video facilities. We can only converse with them by typing at keyboards, not through speech. They can give access to vast stores of information but they are not able to process it as knowledge.

These limitations severely restrict the scope of the new medium as it is now and they are precisely what the FGCS development programme addresses. The FGCS, if the objectives of the programme are met, may be characterized as providing a two-way, interactive medium with the audio-visual facilities and knowledge processing to replicate all capabilities of the most advanced current one-way media.

Summary and conclusions

The sense of shock at the Japanese fifth generation computer development programme was a reliable indicator that we are moving into a new phase in the social impact of computing. Many economists are suggesting that the 1980s will be a time of major change as we move into the downwave part of a long-term cycle in which our society will be re-structured (Mensch 1979; Beckman 1983; van Dujin 1983). The computer is playing a major role in that re-structuring and we in the information industry need to understand this and our roles in it. Technology-based forecasts alone will tell us little about the future and we need to examine it from a number of other perspectives, each of which provides the basis for some partial understanding.

This paper has examined views of the social impact of computing and sought to give a framework from which the bases of opposing arguments can be derived. Computer generations follow the basic eight-year business cycle and we may extrapolate the patterns of change but gauge little of their logic. Examination of computer applications in terms of individual needs and the social structures

that supports them indicates the probable nature of new markets for fifth-generation systems and hence their applications and social consequences. The analogy of computing as a new medium of communication places it in on a long-term trend line outside the computer industry and rationalizes the technical objectives of the fifth generation.

Finally, what is the implication of the Japanese programme for the rest of the world? The USA, UK and EEC have already made plans for fifth-generation programmes in response to that of Japan. The USA sees a danger of being bypassed by Japan in the computer industry as it has in the car and video recorder industries (Kahn and Pepper 1980; Franko 1983; Wolf 1983; Davidson 1984). The UK is in a similar position to Japan as an island nation with limited resources that needs to export and sees information technology as vital to its future (HMSO 1980). Other countries such as Canada and those of the Third World and Latin America vary in their positions. Some are rich in basic resources and already trade these for much of their high technology requirements, whereas others do not have the means for trade. In general, countries outside the USA–EEC group do not have the resources to compete in high technology on a broad front. Should they attempt to do so at all and, if so, in what areas? There is no easy answer but the model of computer development presented in this paper leads to some conclusions:

Social needs

(1) It is important to understand the pattern of social change underlying developments in high technology.
(2) The needs of an increasingly complex, interdependent world drive high-technology developments. Society impacts information technology at least as much as information technology impacts society.
(3) The decreasing cost of information technology allows basic social needs to be satisfied in nations that previously could not afford the technology.
(4) The satisfaction of basic social needs moves the focus of attention to higher-level needs and creates new markets for information technology.

National Roles

(5) The efficient use of world resources leads to specialization: for example, Japan taking a lead in high technology under the pressure to export knowledge products since it has few natural resources.
(6) The most significant danger of allowing any one country to become the major source of high technology is over-dependence by other countries on a single source of supply under conditions where world scenarios may change.

(7) Another danger of extreme specialization is that of becoming too remote from some aspects of the technology. To understand the technology requires a depth of involvement with it, certainly in applications, and possibly in development, even if this has no direct commercial benefits.

(8) Countries with large natural resources should be particularly concerned with the use of knowledge technology to preserve these resources, exploit them efficiently and preserve their value. This applies to both developed and developing countries.

Co-operation

(9) The range of technologies to be integrated in fifth-generation systems makes it very difficult, if not impossible, for any one organization to encompass them all. Even nations may find this difficult.

(10) Rapid technological change may make a totally competitive economy inadequate for the development of fifth-generation systems. A more subtle system of partial co-operation and partial competition may be necessary to co-ordinate the necessary resources.

(11) Organizations and nations that are able to enter this new economic environment rapidly, and operate in it effectively, will tend to dominate industry and commerce in the future.

Development

(12) Where development is undertaken with the intention of direct commercial applications it must be 'rifle-shot' into well specified markets, preferably based on strong home markets.

(13) Each nation has its own pattern of local markets that could benefit from fifth-generation computer systems and there is scope for vertical market specialization based on these.

(14) Bureaucracy is increasingly expensive for all countries and forms a major home market for knowledge-processing systems.

Application

(15) Fifth-generation systems will impact all industries and those businesses that do not take full advantage of them will find it increasingly difficult to compete.

(16) It is important for every industry to become an educated consumer of high technology.

(17) Primary industries, such as oil, forestry and agriculture, are major potential users of knowledge products. They are also significant Third World industries where there is export potential for the knowledge technology, for good-will as well as currency.

New markets

(18) The new Western markets for leisure, job enhancement and personal development have a massive growth potential.

(19) The low cost, ease of use and multi-language communication capabilities of fifth-generation computer systems will also lead to more basic applications in third-world countries concerned with government and resource management.

(20) Computing systems provide a new medium for communication that is interactive and transcends time and space. Fifth-generation systems will make all aspects of this medium competitive with those of the best in other media.

(21) Language and cultural barriers will limit any one nation from providing the 'software' to be distributed through the new computing medium for markets in other nations. They provide a natural division between the technology and its applications.

(22) Media are vehicles for extending ourselves, our society and our culture. We are moving into a different world and the future is extremely uncertain. However, the new knowledge technologies are the best tools available for coping with prediction, decision and action under uncertainty.

Whether the uncertainties of the information age are problems or opportunities is a matter off personal and national choice. It is to be hoped that we will welcome them even though they may be surprising and disturbing, the more so the more we believe ourselves familiar with this technology. The future will not be familiar for anyone. We need to prepare for it now.

Acknowledgements

Financial assistance for this work has been made available by the National Sciences and Engineering Research Council of Canada. I am grateful to my wife and colleague Mildred Shaw for her collaboration in this work and comments on the manuscript.

References

Aida, H., Tanaka, H., and Moto-oka, T. (1983). A Prolog extension for handling negative knowledge. *New Generation Computing*, **1**, 81–91.

Aiso, H. (1982). Fifth generation computer architecture. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 121–7. North Holland, Amsterdam.

Amamiya, M., Hakozaki, K., Yokoi, T., Fusaoka, A., and Tanaka, Y. (1982). New architecture for knowledge base mechanisms. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 179–88. North Holland, Amsterdam.

Ashby, W. R. (1952). *Design for a brain*. Chapman and Hall, London.

Bahl, L. R., Baker, J. K., Cohen, P. S., Cole, A. G., Jelinek, F., Lewis, B. L., and Mercer, R. L. (1978). Automatic recognition of continuously spoken sentences from a finite state grammar. In *Proceedings of the 1978 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 418–21. IEEE, New York.

Bainbridge, L. (1979). Verbal reports as evidence of the process operator's knowledge. *International Journal of Man–Machine Studies*, **11**, 411–36.

Ballard, D. H., and Brown, C. M. (1982). *Computer vision*. Prentice Hall, Englewood Cliffs, N.J.

Balzer, R., Cheatham, T. E., and Green, C. (1983).

Software technology in the 1990s: using a new paradigm. *Computer*, **16** (11), 39–45.

Barney, C. (1983). Who will pay for supercomputer R&D? *Electronics*, **56** (18), 87–8.

Bar-Hillel, Y. (1964). *Language and information: selected essays on their theory and application*. Addison-Wesley, Reading, Mass.

Bawden, A., Greenblatt, R., Holloway, J., Knight, T., Moon, D., and Weinreb, D. (1979). The Lisp machine. In *Artificial intelligence: an MIT perspective* (eds P. H. Winston and R. H. Brown), pp. 343–73. MIT Press, Cambridge, Mass.

Beckman, R. C. (1983). *The downwave: surviving the second great depression*. Pan Books, London.

Bell, D. (1973). *The coming of post-industrial society*. Basic Books, New York.

Belnap, N. D. (1976). How a computer should think. In *Contemporary aspects of philosophy* (ed. G. Ryle), pp. 30–56. Oriel Press, Stocksfield, UK.

Bernstein, J. (1982). *The handbook of commodity cycles: a window on time*. John Wiley, New York.

Bhusri, G. S. (1984). Considerations for ISDN planning and implementation. *IEEE Communications Magazine*, **22**, 18–32.

Blaser, A. (ed.) (1980). *Data base techniques for pictorial applications*. Springer-Verlag, Berlin.

Boehm, B. W., and Standish, T. A. (1983). Software technology in the 1990s: using an evolutionary paradigm. *Computer*, **16** (11), 30–7.

Bobrow, D. G., and Collins, A. (eds) (1975). *Representation and understanding: studies in cognitive science*. Academic Press, New York.

Boral, H., and DeWitt, D. J. (1982). Applying data flow techniques to data base machines. *Computer*, **15** (8), 57–63.

Boral, H., and DeWitt, D. J. (1983). Database machines: an idea whose time has passed? In *Database machines* (eds H.-O. Leilich and M. Missikopf), pp. 166–87. Springer-Verlag, Berlin.

Breuer, M. A. (ed.) (1975). *Digital system design automation*. Pitman, London.

Brod, C. (1984). *Techno stress: the human cost of the computer revolution*. Addison-Wesley, Reading, Mass.

Buchanan, B. G., and Shortliffe, E. H. (eds) (1984). *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, Mass.

Business Week (1984). Artificial intelligence is here. *Business Week* (2850), 54–62.

Butler, S. (1872). *Erewhon*.

Buzbee, B. L. (1982). Japanese supercomputer technology. *Science*, **218**, 1189–93.

Carbonell, J., Cullingford, R., and Gershman, A. (1981). Knowledge-based machine translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **3**, 376–392.

Chang, S. K., and Fu, K. S. (eds) (1980). *Pictorial information systems*. Springer-Verlag, Berlin.

Cherfas, J. (1982). *Man-made life: an overview of the science, technology and commerce of genetic engineering*. Pantheon, New York.

Chikayama, T. (1983). ESP—extended self-contained Prolog—as a preliminary kernel language of fifth generation computers. *New Generation Computing*, **1**, 11–24.

Clark, K. L., and Tarnlund, S.-A. (eds) (1982). *Logic programming*. Academic Press, London.

Coombs, M. J., and Alty, J. L. (eds) (1981). *Computing skills and the user interface*. Academic Press, London.

Costigan, D. M. (1978). *Electronc delivery of documents and graphics*. Van Nostrand Reinhold, New York.

Danver, J. H., and Nevison, J. M. (1969). Secondary school use of the time-shared computer at Dartmouth College. In *Proceedings of the Spring Joint Computer Conference, 34*, pp. 681–9. AFIPS Press, N.J.

Date, C. J. (1981). *An introduction to database systems*. Addison-Wesley, Reading, Mass.

Davidson, W. H. (1984). *The amazing race: winning the technorivalry with Japan*. John Wiley, New York.

Davis, R., and Lenat, D. B. (1982). *Knowledge-based systems in artificial intelligence*, McGraw-Hill, New York.

Day, J. D., and Zimmerman, H. (1983). The OSI reference model. *Proceedings of the IEEE*, **71**, 1334–40.

De Bono, E. (1979). *Future positive*. Maurice Temple Smith, London.

Dizard, W. P. (1982). *The coming information age: an overview of technology, economics and politics*. Longman, New York.

Dreyfus, H. L. (1972). *What computers can't do: the limits of artificial intelligence*. Harper, New York.

Eden, R. C., and Welch, B. M. (1983). Integrated circuits: the case for gallium arsenide. *IEEE Spectrum*, **20** (12), 30–7.

Edson, D. (1984). Bin-picking robots punch in. *High Technology*, **4** (6), 57–61.

Electronics (1983*a*). A plethora of projects in the US try data-flow and other architectures. *Electronics*, **56** (12), 107–10.

Electronics (1983*b*). Japan is busy trying to make manufacturable data-flow computers. *Electronics*, **56** (12), 114.

Electronics (1984). World markets survey and forecast. *Electronics* **57** (1), 123–54.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. (1980). The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, **12**, 213–53.

Fabry, R. S. (1974). Capability-based addressing. *Communications of the Association of Computer Manufacturers*, **17**, 403–12.

Fahlman, S. E. (1979). *NETL: a system for representing and using real-world knowledge*. MIT Press, Cambridge, Mass.

Fano, R. M. (1965). The MAC system: a progress report. In *Computer augmentation of human reasoning* (eds M. A. Sass and W. D. Wilkinson), pp. 131–50. Spartan Books, Washington, D.C.

Fant, G. (1975). Key-note address. In *Speech recognition* (ed. D. R. Reddy), pp. x–xiii.

Farley, B. G., and Clark, W. A. (1954). Simulation of self-organizing system by a digital computer. *IRE Transactions of Information Theory*, **PGIT-4**, 76–84.

Fathi, E., and Krieger, M. (1983). Multiple microprocessor systems: what, why and when. *Computer*, **16** (3), 23–32.

Feigenbaum, E. A. (1980). Knowledge engineering: the applied side of artificial intelligence. Report STAN-CS-80-812, Department of Computer Science, Stanford University.

Feigenbaum, E. A., and McCorduck, P. (1983). *The fifth generation: artificial intelligence and Japan's computer challenge to the world*. Addison-Wesley, Reading, Mass.

Fields, S. W. (1983). Silicon compiler cuts time to market for DEC's MicroVAX I. *Electronics*, **56** (21). 47–8.

Fleck, J. (1982). Development and establishment in artificial intelligence. In *Scientific establishments and hierarchies* (eds N. Elias, H. Martins and R. Whitley), pp. 169–217. D. Reidel, Dortrecht.

Franko, L. G. (1983). *The threat of the Japanese multinationals—how the West can respond*. John Wiley, Chichester.

Fuchi, K. (1982). Aiming for knowledge information processing systems. In *Fifth generation computer systems* (ed. T. Moko-ota), pp. 107–20. North Holland, Amsterdam.

—— (1984). Significance of fifth generation computer systems research and development. *ICOT Journal* (3), 8–14.

——, Sato, S., and Miller, E. (1984). Japanese approaches to high-technology R&D. *Computer*, **17** (3), 14–18.

Fuji (1983). *Japan Science and technology outlook*. Fuji Corporation, Tokyo.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-29**, 254–72.

Furukawa, K., Nakajima, R., Yonezawa, A., Goto, S., and Aoyama, A. (1982). Problem solving and inference mechanisms. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 131–8. North Holland, Amsterdam.

Gaines, B. R. (1978). Man–computer communication—what next? *International Journal of Man-Machine Studies*, **10**, 225–32.

—— (1979). The role of the behavioural sciences in programming. In *Structured software development*, vol. 2, pp. 57–68. Infotech International, Maidenhead.

—— (1981). Logical foundations for database systems. In *Fuzzy reasoning and its applications* (eds E. H. Mamdani and B. R. Gaines), pp. 289–308. Academic Press, London.

—— (1983). Precise past—fuzzy future. *International Journal of Man–Machine Studies*, **19**, 117–34.

—— (1984). From ergonomics to the fifth generation: 30 years of human–computer interaction studies. In *Proceedings of INTERACT 84: 1st IFIP Conference on Human–Computer Interaction*. North-Holland, Amsterdam.

—— and Andreae, J. H. (1966). A learning machine in the context of the general control problem. In *Proceedings of the 3rd Congress of the International Federation for Automatic Control*. Butterworth, London.

—— and Shaw, M. L. G. (1983*a*). Dialog engineering. In *Designing for human–computer interaction* (eds M. Sime and M. J. Coombs), pp. 23–53. Academic Press, London.

—— —— (1983*b*). Is there a knowledge environment? In *The relation between major world problems and systems learning* (ed. G. Lasker), pp. 27–34. Society for General Systems Research, New York.

—— —— (1984*a*). *The art of computer conversation: a new medium for communication*. Prentice Hall, Englewood Cliffs, N.J.

—— —— (1984*b*). Dialog shell design. In *Proceedings of INTERACT 84: 1st IFIP Conference on Human–Computer Interaction*. North Holland, Amsterdam.

—— —— (1984*c*). Generating emotions in computer systems. *Man–Environment Systems*, in press.

——, McKellar, I. D., Dinger, W. P., Fast, S. R., Fowles, B. J., Fraccaro, M. A., Jolivet, G. C., and Maludzinski, A. B. (1984). Some experience in the real-time processing of handwriting. In *Proceedings of the Seventh International Conference on Pattern Recognition*.

Galinski, C. (1983). VLSI in Japan: the big leap forward, 1980–1981. *Computer*, **16** (3), 14–21.

Glinz, M. (1983). A dataflow retrieval unit for a relational database machine. In *Database machines* (eds H.-O. Leilich and M. Missikoff), pp. 20–40. Springer-Verlag, Berlin.

Godet, M. (1983). Forecasting the blunders in forecasting. *Futures*, **1983**, 181–92.

Goldberg, A. (1984). *Smalltalk-80: the interactive programming environment*. Addison-Wesley, Reading, Mass.

—— and Robson, D. (1983). *Smalltalk-80: the language and its implementation*. Addison-Wesley, Reading, Mass.

Goldstein, I., and Papert, S. (1977). Artificial in-

telligence, language and the study of knowledge. *Cognitive Science*, **1**, 84–123.

Grandjean, E., and Vigliani, E. (eds) (1980). *Ergonomic aspects of visual display terminals*. Taylor and Francis, London.

Hansen, W. J. (1971). User engineering principles for interactive systems. In *Proceedings of the Fall Joint Computer Conference, 39*, pp. 523–32. AFIPS Press, Montvale, N.J.

Hanson, A. R., and Riseman, E. M. (eds) (1978). *Computer vision systems*. Academic Press, New York.

Hansson, A., Haridi, S., and Tarnlund, S.-A. (1982). Properties of a logic programming language. In *Logic programming* (ed. K. L. Clark and S.-A. Tarnlund), pp. 267–280. Academic Press, London.

Harris, L. R. (1977). User oriented data base query with the ROBOT natural language query system. *International Journal of Man–Machine Studies*, **9**, 697–713.

—— (1984). Experience with INTELLECT. *AI Magazine*, **5** (2), 43–50.

Harris, Z. (1982). *A grammar of English on mathematical principles*. John Wiley, New York.

Hawkins, D. (1983). An analysis of expert thinking. *International Journal of Man–Machine Studies*, **18**, 1–47.

Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. (eds). *Building expert systems*. Addison-Wesley, Reading, Mass.

Hindin, H. J. (1982). Dual-chip sets forge vital link for Ethernet local-network scheme. *Electronics*, **55** (20), 89–91.

Hirose, K., and Fuchi, K. (1984). *The culture of the fifth generation computer* [in Japanese]. Kaimeisha, Tokyo.

HMSO (1980). *Information technology*. HMSO, London.

—— (1982). *A programme for advanced information technology: the report of the Alvey Committee*. HMSO, London.

ICOT (1983). *Outline of research and development plans for fifth generation computer systems*. ICOT, Tokyo.

Jantsch, E. (1967). *Technological forecasting in perspective*, OECD, Paris.

Johnsen, G. (1984). Gallium arsenide chips emerge from the lab. *High Technology*, **4** (7), 44–52.

Johnson, C. (1983). *MITI and the Japanese miracle: the growth of industrial policy 1925–1975*. Stanford University Press, Stanford, Calif.

Johnson-Laird, P. N. (1977). *Thinking: readings in cognitive science*. Cambridge University Press, Cambridge.

—— and Wason, (1977).

Kahn, H. and Pepper, T. (1980). *The Japanese challenge: the success and failure of economic success*. William Morrow, New York.

Kakatu, T., Miyazaki, N., Shibayama, S., Yokota, H. and Murakami, K. (1983). A relational database machine 'Delta'. ICOT Technical Memorandum TM-0008, Institute for New Generation Computer Technology, Tokyo.

Karatsu, H. (1982). What is required of the fifth generation computer—social needs and impact. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 93–106. North Holland, Amsterdam.

Kawanobe, K. (1984). Present status of fifth generation computer systems project. *ICOT Journal* (3), 15–23.

Kidd, A. (1982). *Man–Machine dialogue design*. Martlesham Consultancy Services, Ipswich, UK.

Kidode, M. (1983). Image processing machines in Japan. *Computer*, **16** (1), 68–80.

Kitsuregawa, M., Tanaka, H. and Moto-oka, T. (1983). Application of hash to data base machine and its architecture. *New Generation Computing*, **1**(1), 63–74.

Kostas, D. J. (1984). Transition to ISDN—an overview. *IEEE Communications Magazine*, **22** (1), 11–17.

Kowalski, R. (1979). *Logic for problem solving*, North Holland, New York.

Krasner, G. (1984). *Smalltalk-80: bits of history, words of advice*. Addison-Wesley, Reading, Mass.

Kubera, R. and Malms, M. (1983). A concept for a dataflow database machine—an overview. In *Database machines* (eds H.-O. Leilich and M. Missikoff), pp. 41–5. Springer-Verlag, Berlin.

Kuhn, T. S. (1962). *The structure of scientific revolutions*. University of Chicago Press, Chicago.

Laudon, K. C. (1974). *Computers and bureaucratic reform*. John Wiley, New York.

Lea, W. A. (ed.) (1980). *Trends in speech recognition*. Prentice-Hall, Englewood Cliffs, N.J.

Lederer, K. (ed.) (1980). *Human needs: a contribution to the current debate*. Oelgeschlager, Gunn and Hain, Cambridge, Mass.

Lehnert, W. (1978). *The process of question answering*. Lawrence Erlbaum, Hillsdale, N.J.

Leilich, H.-O., and Missikoff, M. (eds) (1983). *Database machines*. Springer-Verlag, Berlin.

Lenat, D. B., Sutherland, W. R., and Gibbons, J. (1982). Heuristic search for new microcircuit structures: an application of artificial intelligence. *AI Magazine*, **3** (3), 17–33.

Licklider, J. C. R. (1960). Man–computer symbiosis. *IRE Transactions on Human Factors in Electronics*, **HFE-1**, 4–11.

Lighthill, J. (1973). Artificial intelligence: a general survey. In *Artificial intelligence: a paper symposium*. Science Research Council, London.

Lindsay, P. H., and Norman, D. A. (1977). *Human information processing*. Academic Press, New York.

Luhmann, N. (1979). *Trust and power*. John Wiley, Chichester.

McCarthy, J. (1959). Programs with common sense. In *Proceedings of a symposium on the mechanization of thought processes* (eds D. V. Blake and A. M. Uttley), pp. 75–84. HMSO, London.

McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, 115–37.

McDermott, D., and Doyle, J. (1980). Non-monotonic logic I. *Artificial Intelligence*, **13**, 41–72.

McDermott, J. (1981). R1: the formative years. *AI Magazine*, **2** (2), 21–9.

Machlup, F. (1980). *Knowledge and knowledge production*. Princeton University Press, Princeton, N.J.

Manuel, T. (1983). Lisp and Prolog machines are proliferating. *Electronics*, **56** (22), 132–7.

Martin, J. (1973). *Design of man–computer dialogues*. Prentice-Hall, Englewood Cliffs, N.J.

Maslow, A. H. (1971). *Motivation and personality*. Harper and Row, New York.

Mathlab Group (1977). *MACSYMA reference manual*. Computer Science Laboratory, MIT, Cambridge, Mass.

Mauchly, J. W. (1973). Preparation of problems for EDVAC-type machines. In *The origins of digital computers: selected papers* (ed. B. Randell), pp. 365–9. Springer-Verlag, Berlin.

Mead, C., and Conway, L. (1980). *Introduction to VLSI systems*. Addison-Wesley, Reading, Mass.

Mensch. G. (1979). *Stalemate in technology: innovations overcome the depression*. Ballinger, Cambridge, Mass.

Michie, D. (ed.) (1979). *Expert systems in the micro electronic age*, Edinburgh University Press, Edinburgh.

Minsky, M. (1961). A selected descriptor–index bibliography to the literature on artificial intelligence. *IRE Transactions on Human Factors in Electronics*, **2**, 39–55.

Miyachi, T., Kunifuji, S., Kitakami, H., Furukawa, K., Takeuchi, A., and Yokota, H. (1984). A knowledge assimilation method for logic databases. In *Proceedings of the 1984 International Symposium on Logic Programming*, pp. 118–25.

Mizoguchi, R., and Kakusho, O. (1984). Continuous speech recognition based on knowledge engineering techniques. In *Proceedings of the Seventh International Conference on Pattern Recognition, Montreal* (to be published).

Moto-oka, T. (ed.) (1982). *Fifth generation computer systems*. North Holland, Amsterdam.

—— and Stone, H. S. (1984). Fifth generation computer systems: a Japanese project. *Computer*, **17** (3), 6–13.

Mowshowitz, A. (1976). *The conquest of will: information processing in human affairs*. Addison-Wesley, Reading, Mass.

Myers, G. J. (1978). *Advances in computer architecture*, John Wiley, New York.

Myers, W. (1982). LISP machines displayed at AI conference. *Computer*, **15** (11), 79–82.

Nakagawa, H. (1984). AND parallel Prolog with divided assertion set. In *Proceedings of the 1984 International Symposium on Logic Programming*, pp. 22–8.

Nakashima, H. (1984). Knowledge representation in PROLOG/KR. In *Proceedings of the 1984 International Symposium on Logic Programming*, pp. 126–30.

Nelson, T. (1980). Interactive systems and the design of virtuality. *Creative Computing*, **6** (11), 56–62.

Newell, A., Barnett, J., Forgie, J., Green, C., Klatt, D. H., Licklider, J. C. R., Munson, J., Reddy, D. R., and Woods, W. A. (1973). *Speech understanding systems: final report of a study group*. North Holland, Amsterdam.

Newell, A., Shaw, J. C., and Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review*, **65**, 151–66.

Nickerson, R. S. (1969). Man–computer interaction: a challenge for human factors research. *IEEE Transactions on Man–Machine Systems*, **MMS-10**, 164–80.

Norman, D. A. (1980). Twelve issues for cognitive science. *Cognitive Science*, **4**, 1–32.

—— (1984). Stages and levels in human–machine interaction. *International Journal of Man–Machine Studies* **21**, in press.

Orr, W. D. (ed.) (1968). *Conversational computers*. John Wiley, New York.

Pearl, J. (ed.) (1983). *Search and heuristics*. North Holland, Amsterdam.

Peccei, A. (1982). *One hundred pages for the future: reflections of the President of the Club of Rome*. Pergamon Press, Oxford.

Petri, C. A. (1962). Kommunikation mit Automaten. Dissertation an der Universitat Bonn.

Pierce, J. R. (1969). Whither speech recognition? *Journal of the Acoustical Society of America*, **46**, 1049–51.

Quinlan, J. R. (1983). Inferno: a cautious approach to uncertain inference. *Computer Journal*, **26**, 255–69.

Rabbat, G. (ed.) (1983). *Hardware and software concepts in VLSI*. Van Nostrand Reinhold, New York.

Rau, N. (1974). *Trade cycles: theory and evidence*. Macmillan, London.

Rissland, E. L. (1984). Ingredients of intelligent user interfaces. *International Journal of Man–Machine Studies*, **21**, in press.

Robinson, A. L. (1984). One billion transistors on a chip? *Science*, **223**, 267–8.

Robinson, J. A., and Sibert, E. E. (1982). LOGLISP: motivation, design and implementation. In *Logic programming* (eds K. L. Clark and S.-A. Tarnlund), pp. 299–313. Academic Press, London.

Rosen, S. (1983). Generations, computer. *Encyclopedia of computer science and engineering* (eds A. Ralston and E. D. Reilly), pp. 657–8.

Rosenblatt, F. (1958). The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–407.

Rosenblueth, A., Wiener, N. & Bigelow, J. (1943). Behavior, purpose and teleology. *Philosophy of Science*, **10**, 318–26.

Rupp, C. R. (1981). Components of a silicon compiler system. In *VLSI 81* (ed. J. P. Gray), pp. 227–36. Academic Press, London.

Sackman, H. (1970). *Man–computer problem solving*. Auerbach, Princeton, N.J.

Sager, N. (1981). *Natural language information processing: a computer grammar of English and its applications*. Addison-Wesley, Reading, Mass.

Sakai, K., and Miyachi, T. (1983). Incorporating naive negation into PROLOG. ICOT Technical Report TR-028, Institute for New Generation Computer Technology, Tokyo.

—— and Nakagawa, S. (1977). A speech understanding system of simple Japanese sentences in a task domain. *IECEJ Transactions*, **60E** (1), 13–20.

Sakamura, K., Sekino, A., Kodaka, T., Uehara, T., and Aiso, H. (1982). VLSI and system architectures—the new development of system 5G. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 189–208. North Holland, Amsterdam.

Samuels, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, **3**, 210–29.

Sass, M. A., and Wilkinson, W. D. (eds) (1965). *Computer augmentation of human reasoning*. Spartan Books, Washington, D.C. See pp. 131–50.

Schank, R. C. (ed.) (1975). *Conceptual information processing*. North Holland, Amsterdam.

—— and Colby, K. M. (eds) (1973). *Computer models of thought and language*. W. H. Freeman, San Francisco. See pp. 187–247.

Selfridge, O. (1959). Pandemonium: a paradigm for learning. In *Proceedings of a symposium on the mechanization of thought processes* (eds D. V. Blake and A. M. Uttley), pp. 511–29. HMSO, London.

Shackel, B. (1959). Ergonomics for a computer. *Design*, **120**, 36–9.

Shapiro, E. Y. (1983). A subset of concurrent Prolog and its interpreter. ICOT Technical Report TR-003, Institute for New Generation Computer Technology, Tokyo.

Shaw, J. C. (1968). JOSS: experience with an experimental computing service for users at remote consoles. In *Conversational computers* (ed. W. D. Orr), pp. 15–22. John Wiley, New York.

Shaw, M. L. G., and Gaines, B. R. (1984). Fifth generation computing as the next stage of a new medium. In *Proceedings of National Computer Conference, 53*. AFIPS Press, Arlington, Va.

Shortliffe, E. H. (1976). *Computer-based medical consultations: MYCIN*. Elsevier, New York.

Sime, M. E., and Coombs, M. J. (eds) (1983). *Designing for human–computer interaction*. Academic Press, London.

Sime, M. E., Green, T. R. G., and Guest, D. J. (1973). Psychological evaluation of two conditional constructions used in computer languages. *International Journal of Man–Machine Studies*, **5**, 105–13.

Simon, J. C. (ed.) (1980). *Spoken language generation and understanding*. D. Reidel, Dordrecht.

Simons, G. L. (1983). *Towards fifth-generation computers*. National Computing Centre, Manchester.

Smith, D. C. (1977). *Pygmalion*. Birkhauser, Basel.

——, Irby, C., Kimball, R., Verplank, B., and Harslem, E. (1983). Designing the Star user interface. In *Integrated interactive computing systems* (eds P. Degano and E. Sandewall), pp. 297–313. North Holland, Amsterdam.

Snyder, D. P. (1971). *Modal logic*. Van Nostrand Reinhold, New York.

Solomonoff, R. J. (1957). An inductive inference machine. *IRE National Convention Record*, **1957**, 56–62.

Steier, R. (1983). Cooperation is the key: an interview with B. R. Inman. *Communications of the Association of Computer Manufacturers*, **26**, 642–5.

Stickel, M. E. (1984). A Prolog technology theorem prover. In *Proceedings of the 1984 International Symposium on Logic Programming*, pp. 211–17.

Suen, C. Y., Berthod, M., and Mori, S. (1980). Automatic recognition of handprinted characters—the state of the art. *Proceedings of the IEEE*, **68**, 469–87.

Suwa, M., Furukawa, K., Makinouchi, A., Mizoguchi, T., Mizoguchi, F., and Yamasaki, H. (1982). Knowledge base mechanisms. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 139–45. North Holland, Amsterdam.

Tanaka, H., Amamiya, M., Tanaka, Y., Kadowaki, Y., Yamamoto, M., Shimada, T., Sohma, Y., Takizawa, M., Ito, N., Takeuchi, A., Kitsuregawa, M., and Goto, A. (1982*a*). The preliminary research of data flow machine and data base machine as the basic architecture of fifth generation computer systems. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 209–19. North Holland, Amsterdam.

Tanaka, H., Chiba, S., Kidode, M., Tamura, H., and Kodera, T. (1982*b*). Intelligent man–machine interface. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 147–57. North Holland, Amsterdam.

Thompson, B. H., Thompson, F. B., and Ho, T.-P. (1983). Knowledgeable contexts for user interaction. In *Proceedings of National Computer Conference, 52*, pp. 29–38. AFIPS Press, Arlington, Va.

Toffler, A. (1970). *Future shock*. Pan, London.

Toffler, A. (1980). *The third wave*. Bantam, New York.

Tucker, A. B. (1984). A perspective on machine translation. *Communications of the Association of Computer Manufacturers*, **27**, 322–41.

Tucker, J. B. (1984). Biochips: can molecules compute? *High Technology*, **4**(2), 36–47.

Tudhope, D. S., and Oldfield, J. V. (1983). A high-level recognizer for schematic diagrams. *IEEE Computer Graphics and Applications*, **3** (3), 33–40.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, **59**, 433–50.

Tyner, P. (1981). *iPAC 432 general data processor architecture reference manual.* Intel, Aloha, Oregon.

Uchida, S., Tanaka, H., Tokoro, M., Takei, K., Sugimoto, M., and Yasuhara, H. (1982). New architectures for inference mechanisms. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 167–78. North Holland, Amsterdam.

Uhr, L. (1984). *Algorithm-structured computer arrays and networks.* Academic Press, Orlando, Fla.

Ullman, J. D. (1984). *Computational aspects of VLSI.* Computer Science Press, Baltimore, Md.

van Dujin, J. J. (1983). *The long wave in economic life.* George Allen and Unwin, London.

von Bertalanffy, L. (1950). The theory of open systems in physics and biology. *Science*, **111**, 139–64.

Wallich, P. (1984). On the horizon: fast chips quickly. *IEEE Spectrum*, **21** (3), 28–34.

Walter, C. J., Bohl, M. J., and Walter, A. B. (1968). Fourth generation computer systems. *Proceedings of Spring Joint Computer Conference, 32*, pp. 423–41. Thompson, Washington, D.C.

Walter, G. (1953). *The living brain.* Duckworth, London.

Wasserman, T. (1973). The design of idiot-proof interactive systems. *Proceedings of the National Computer Conference, 42*, pp. M34–M38. AFIPS Press, Montvale, N.J.

Waterman, D. A., and Hayes-Roth, F. (eds) (1978). *Pattern-directed inference systems.* Academic Press, New York.

Weaver, W. (1955).

Weegenaar, H. J. (1978). Virtuality and other things like that. *Proceedings of IEEE COMCON*, 287–93.

Wegner, P. (ed.) (1979). *Research directions in software technology*, MIT Press, Cambridge, Mass.

Weinberg, G. M. (1971). *The psychology of computer programming.* Van Nostrand Reinhold, New York.

Weinreb, D., and Moon, D. (1981). *Lisp machine manual.* Massachusetts Institute of Technology, Cambridge, Mass.

Weitzman, C. (1980). *Distributed micro/minicomputer systems.* Prentice-Hall, Englewood Cliffs, N.J.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Journal of the Association of Computer Manufacturers*, **9**, 36–45.

—— (1976). *Computer power and human reason: from judgement to calculation.* W. H. Freeman, San Francisco.

Welbank, M. (1983). *A review of knowledge acquisition techniques for expert systems*, Martlesham Consultancy Services, Ipswich.

White, R. R. (1967). On-line software—the problems. In *The transition to on-line computing* (ed. F. Gruenberger), pp. 15–26. Thompson, Washington, D.C.

Widrow, B. (1959). Adaptive sampled-data systems—a statistical theory of adaptation. *WESCON Convention Record*, **1959**, 74–85.

Wiener, H. (1984). Mainframe maneuvers. *Datamation*, **30**, 159–66.

Wiener, N. (1948). *Cybernetics.* John Wiley, New York.

—— (1950). *The human use of human beings.* Riverside Press, Cambridge, Mass.

Wilks, Y. (1979). Machine translation and artificial intelligence. *Translating and the computer* (ed. B. M. Snell), pp. 27–43. Elsevier, New York.

Wilson, K. G. (1983). Universities and the future of high-performance computing technology. In *Proceedings of National Computer Conference, 52*, pp. 469–77. AFIPS Press, Arlington, Va.

—— (1984). Science, industry and the new Japanese challenge. *Proceedings of the IEEE*, **72**, 6–18.

Winograd, T. (1972). *Understanding natural language.* Edinburgh University Press, Edinburgh.

Withington, F. G. (1974). Five generations of computers. *Harvard Business Review*, **4**, 99–108.

Wojciechowski, J. (1983). The impact of knowledge on man: the ecology of knowledge. *Hommage à François Meyer*, pp. 161–75. Laffitte, Marseille.

Wolf, M. J. (1983). *The Japanese conspiracy: the plot to dominate industry world wide and how to deal with it.* Empire Books, New York.

Woods, W. (1983). What's important about knowledge representation? *Computer*, **16** (10), 22–7.

——, Bates, M., Brown, G., Bruce, B., Cook, C., Klovstad, J., Makhoul, J., Nash-Webber, B., Schwartz, R., Wolf, J., and Zue, V. (1976). Speech understanding systems. BBN Report Number 3438, Bolt, Beranek and Newman, Cambridge, Mass.

Yamamoto, K., Yamada, H., and Oka, R.-I. (1984). Recognition of handprinted Chinese characters and Japanese cursive syllabary characters. In *Proceedings of the Seventh International Conference on Pattern Recognition, Montreal*, to be published.

Yokoi, T., Goto, S., Hayashi, H., Kunifuji, S., Kurokawa, T., Motoyoshi, F., Nakashima, H., Nitta, K., Sato, T., Shiraishi, T., Ueda, K., Umemura, M., and Umeyama, S. (1982). Logic programming and a dedicated high-performance personal computer. In *Fifth generation computer systems* (ed. T. Moto-oka), pp. 159–64. North Holland, Amsterdam.

Zappe, H. H. (1983). Josephson computer technology—an overview. In *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers*, pp. 516–17. IEEE, New York.

Zadeh, L. A. (1983). Commonsense knowledge representation based on fuzzy logic. *Computer*, **16**, 61–5.