PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT. © ICOT, 1992

Modeling the Generational Infrastructure of Information Technology

B R Gaines

Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca

Abstract

A socio-economic model of generational infrastructure of information technology is presented as a tiered progression of 'learning curves' in mutually supportive technologies. This model is used to analyze trends in research and product development, and the transition from 'computer science' to 'knowledge science' that characterizes the fifth generation. The achievements of fifth generation research are evaluated, and the expected directions of future generations research are projected.

1 Introduction

The Japanese Fifth Generation research program has had an important socio-economic impact internationally in raising government awareness of fundamental changes in the nature of information technology and its strategic role. It has been directly responsible for structural change in national computing policy such as the formation of the MCC within the US anti-trust ethos, and the ESPRIT program in Europe cutting across strongly entrenched national boundaries. One side-effect of this has been to bring into prominence what was previously seen only as a marketing/technical description of the evolution of information technology in terms of its generational infrastructure. As the fifth generation program comes to an end, this raises policy questions as to the nature and significance of the next generation, and as to the utility of conceptualizing computing research in terms of generational advances.

This paper analyzes the questions in terms of a general model of 'learning curves' whose time scale is largely determined by the medium term business cycle of capital replacement. It highlights an important difference between computing and other industrial technologies in that the pace of change in the base, vlsi, technology is so rapid that conventional 'substitution' effects are swamped by a tiered infrastructure of learning curves involving major qualitative differences in technologies. A detailed account of the underlying model and its fit to the past development of information technology has been given elsewhere (Gaines & Shaw, 1986; Gaines, 1990, 1991), and this paper focuses on fifth generation and later issues.

2 Electronic Device Technology

The initial breakthrough for computing was in electronic device technology, and a definition of computer generations in terms of hardware works well for the early machines. However, as Rosen (1983) notes it blurs thereafter and "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 generations, respectively, while the most trivial of hobbyist computers would be a fifth-generation system." The reason for this anomaly is that the substitution effects of one form of technology for another are gradual and do not correspond to major structural changes. The enabling effects of changing technologies are far more dramatic: the change from mechanical to electronic devices made it possible to store programs as data and enabled the use of computers as a general-purpose tool and the development of language compilers; the transistor made reliable operation possible and enabled routine data processing and then interactive timesharing; integrated circuits reduced costs to the level where computers became commonplace and made possible the personal computer dedicated to a single user.

Modern microelectronics commenced in 1956 when silicon planar process was invented and enabled integrated circuit technology to be developed. As Figure 1 shows, the number of devices on a chip follows Moore's law in doubling each year through the 1960s, and has continued to double every eighteen months through the 1970s and 1980s (Robinson, 1984). The current projected limit is some 1,000,000,000 devices on a chip in the late 1990s when quantum mechanical effects will become a barrier to further packing density on silicon planar chips. However, three-dimensional packing, semiconducting peptides, optical devices, or, most likely, new materials not yet considered, are expected to extend the growth.

Microelectronics shows over 9 decades of performance increase in 40 years. Such exponential growth is common in many technologies, but never over more than 2 decades and then in periods of the order of 100 years. Computer technology is unique in being based on underlying devices whose performance has increased at a rate and over a range achieved by no other technology. Logarithmic plots, such as that of Figure 1, do not adequately project the impact such a long-term sustained growth, but this is apparent in the linear plot of the devices on a chip by computer generation as shown in Figure 2. During each generation, changes have taken place that correspond in magnitude to those of some hundred years in other industries. These quantitative changes have led to major qualitative effects that are analyzed in the following sections.

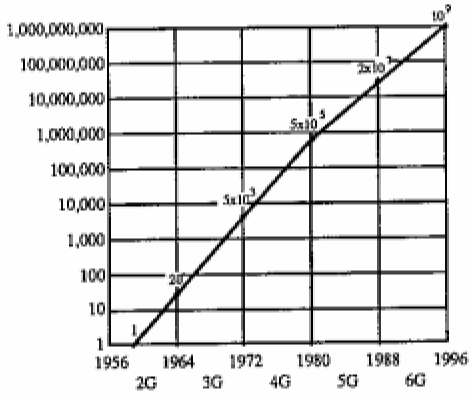


Figure 1 Growth of devices on a chip

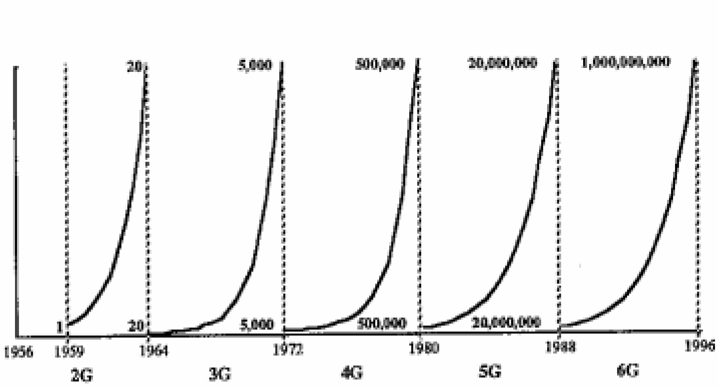


Figure 2 Devices on a chip during six generations of computers

3 Learning Curves in Scientific and Technological Development

There is a simple phenomenological model of developments in science technology as a logistic "learning curve" of knowledge acquisition (Ayres, 1968; Marchetti, 1981). It has been found to be a useful model of the introduction of new knowledge, technology or product in which growth takes off slowly, begins to climb rapidly and then slows down as whatever was introduced has been assimilated. Such curves arise in many different disciplines such as education, ecology, economics, marketing and technological forecasting (Van Dujin, 1983; Stoneman, 1983).

It has also been noted in many disciplines that the qualitative phenomena during the growth of the logistic curve vary from stage to stage (Crane, 1972; De Mey, 1982; Gaines & Shaw, 1986). The era before the learning curve takes off, when too little is known for planned progress, is that of the inventor having very little chance of success. When an inventor makes a *breakthrough*, very rapidly his or her work is *replicated* at research institutions world-wide. The experience gained in this way leads to *empirical* design rules with very little foundation except previous successes and failures. However, as enough empirical experience is gained it becomes possible to inductively model the basis of success and failure and develop *theories*. This transition from empiricism to theory corresponds to the maximum slope of the logistic learning curve. The theoretical models make it possible to *automate* the scientific data gathering and analysis and associated manufacturing processes. Once automation has been put in place effort can focus on cost reduction and quality improvements in what has become a *mature* technology.

4 The Infrastructure of the Information Sciences

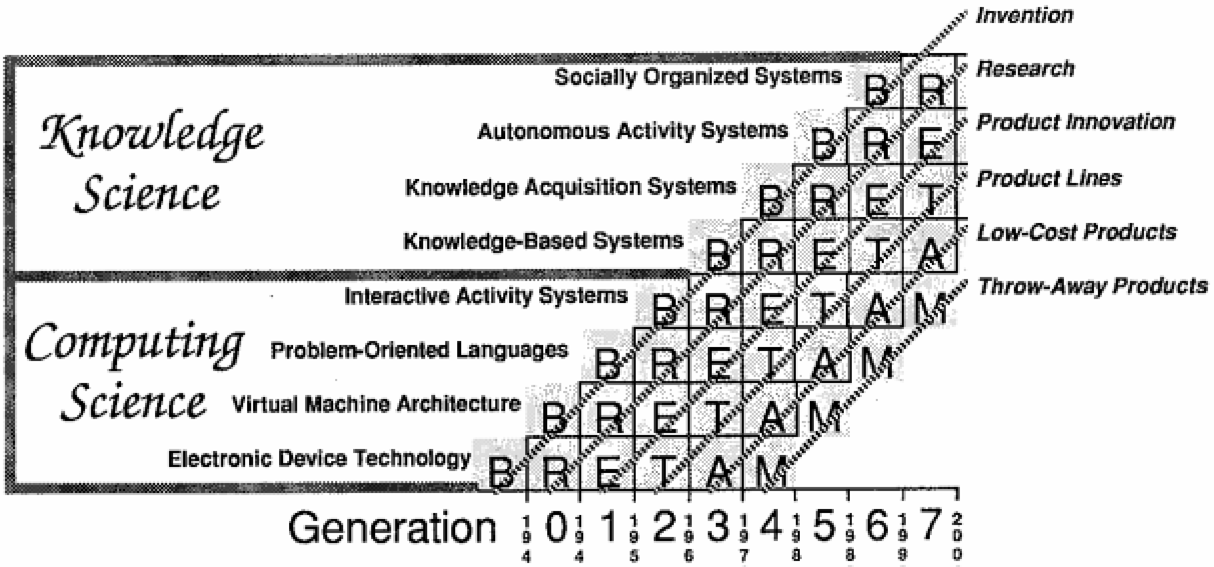
The fast, sustained, learning curve for electronic devices, and the scope for positive feedback in the information sciences, together result in a tiered infrastructure for the information sciences and technologies which is fundamental to their nature. It involves a succession of learning curves as rapid advances in one level of technology trigger off invention in others as shown in Figure 3.

The breakthrough in *electronic device technology* leading to the zeroth generation is placed at 1940 about the time of the Atanasoff and Berry experiments with tube-based digital calculations. Automation by 1980 had reached the extreme level where silicon compilers allow a designer to implement ideas directly in devices with little further human intervention (Fields, 1983).

The first breakthrough generating a computing infrastructure was the introduction of the stored program *virtual machine architecture*. Mauchly (1973) recognized the significance of stored programs, noting that subroutines create "a new set of operations which might be said to form a calculus of instructions." This was the key conceptual breakthrough in computer architecture, that the limited functionality provided directly by the hardware could be increased by stored programs called as subroutines or procedures, and that the hardware and these routines together may be regarded as a new virtual machine. This is the foundation of the development of a variety of forms of virtual machine architectures (Weegenaar, 1978) that separates out computing science as a distinct discipline from other areas of electronic applications.

The next level of breakthrough was in software to bridge the gap between machine and task through the development of *problem-oriented languages*. Work on the design of FORTRAN in 1954 and its issue in 1957 marks the beginning of the second generation era with languages targeted to specific problem areas of business data processing, text processing, database access, machine tool control, and so on. A 1968 paper on the coming fourth generation notes that "programming today has no theoretical basis" and calls for a scientific basis in the next generation (Walter, Bohl & Walter, 1968). Sure enough the theory linking languages to the underlying virtual machines developed during the fourth generation era, for example, that of abstract data types and initial algebras (Goguen, Thatcher & Wagner, 1978). In the fifth generation era the application of experience, design rules and theory to the automation of software production became the top priority (Balzer, Cheatham & Green, 1983).

The next level of breakthrough was in *interactive activity systems* when continuous interaction becoming a significant possibility as the mean time between failures of computers began to be hours rather than minutes in the early 1960s. The move from batch-processing to direct human-computer interaction was made in 1963/1964 with the implementation of MIT MAC, RAND JOSS and Dartmouth BASIC systems. The study of such systems led to design rules for HCI in the 1970s (Hansen, 1971) and theoretical foundations started to emerge in the 1980s



- **Breakthrough: creative advance made**
- **Replication period: experience gained by mimicing breakthrough**
- **Empirical period: design rules formulated from experience**
- **Theoretical period: underlying theories formulated and tested**
- **Automation period: theories predict experience & generate rules**
- **Maturity: theories become assimilated and used routinely**

Figure 3 The infrastructure of the information sciences

(Alexander, 1987). The improvement of human-computer interaction was a major stated priority in the Japanese fifth generation development program (Karatsu, 1982). Other forms of interaction also became feasible as a result of improved reliability such as direct digital control, and various forms of digital communications systems.

The next level of breakthrough was one of *knowledge-based systems* supporting knowledge-processing, the human capability to store information through its inter-relations and make inferences about its consequences. The breakthrough in knowledge-based systems dates from the development of DENDRAL (Buchanan, Duffield & Robertson, 1971) for inferring chemical structures from mass-spectrometry data and MYCIN (Shortliffe, 1976) for the diagnosis of microbial infections in the early 1970s. It led to a spate of expert system development in the fourth generation era of the 1970s (Gevarter, 1983), and pragmatic design rules for knowledge engineering in the current fifth generation era (Hayes-Roth, 1984). The utilization of their vlsi production capability (Gaines, 1984; Galinski, 1983) for the support of knowledge-based systems through PROLOG machines (Kitsuregawa & Tanaka, 1988) has been the other major priority in the Japanese fifth generation development program (Moto-oka, 1982).

Defining the upper levels of the infrastructure becomes more and more speculative as we move into the immediate past of our own era and look for evidence of learning curves that are at their early stages. It is reasonable to suppose that the level above the representation and processing of knowledge in the computer is that of *knowledge acquisition systems*, breakthroughs in machine learning and expertise modeling. Two breakthroughs in this area have been Lenat's AM learning mathematics by discovery (Davis & Lenat, 1982) and Michalski's inductive inference of expert rules for plant disease diagnosis (Michalski & Chilausky, 1980). In the fifth generation era machine learning became a highly active research area in its replication phase (Michalski & Carbonell, 1983). The general field of knowledge acquisition has also seen a massive growth in research (Boose, 1989).

One may speculate that the growth of robotics will provide the next breakthroughs in which goal-directed, mobile computational systems will act autonomously to achieve their objectives. The breakthrough into the sixth generation era commencing in 1988 will probably be seen as one of *autonomous activity systems*. It is possible to see the nascent concepts for this breakthrough in the adoption of the goal-directed programming paradigms of logic programming languages such as PROLOG. When, in a robot, a goal specification is expanded by such a programming system into a sequence of actions upon the world dependent on conditions being satisfied in that world, then the behavior of such a system will deviate sufficiently from its top-level specification, yet be so clearly goal-directed, as to appear autonomous. However, to achieve significant results with such systems we need to add perceptual acts to the planning structures of a language such as SIPE (Wilkins, 1984) and develop logic programming languages that cope with the resulting temporal logic (Allen, 1984)—in these developments the sixth generation breakthrough will come to be recognized, possibly in the notion of "situated action" (Suchman, 1987) and its application in subsumption architectures for autonomous robots (Brooks, 1990; Connell, 1990).

One may speculate further that interaction between these systems will become increasingly important in enabling them to cooperate to achieve goals and that the seventh generation era commencing in 1996 will be one of *socially organized systems*. The Japanese "Sixth Generation" research program proposals emphasize emulation of creativity and intuition and the development of interdisciplinary *knowledge sciences* (STA, 1985; Gaines, 1986a). This recognizes the distinction between "computer science" and "knowledge science" as shown in Figure 3, and that cutting edge innovation in the information sciences involves human and social considerations intrinsic to the nature of knowledge.

It is also possible that building an adequate forecasting model based on the premises of this paper may undermine the very processes that we model. If we come to understand the dynamics of our progress into the future then we may be able to modify the underlying process—to make the next steps more rapidly when the territory is better mapped.

5 Using the BRETAM Model

The tiered infrastructure model of Figure 3 also shows the superimposed trajectories of invention, research, and so on. The intersection of these with the horizontal lines of the different information sciences may be used to model and predict the primary focus of different types of activity in each generation of computers:

- *Invention* is focused at the **BR** interface where new breakthrough attempts are being made based on experience with the replicated breakthroughs of the technology below.
- *Research* is focused at the **R E** interface where new recognized breakthroughs are being investigated using the empirical design rules of the technologies below.
- *Product Innovation* is focused at the **ET** interface where new products are being developed based on the empirical design rules of one technology and the theoretical foundations of those below.
- *Product Lines* are focused at the **T A** interface where product lines can rest on the solid theoretical foundations of one technology and the automation of the technologies below.
- *Low-cost Products* are focused at the **AM** interface where cost reduction can be based on the automated mass production of one technology and the mature technologies below.
- *Throw-away Products* are at the **MM** interface where cost reduction has become such that maintenance and repair costs exceed replacement costs.

For example, by the end of the fourth generation (1972-80):

- **BR**: recognition of the knowledge acquisition possibilities of knowledge-based systems led to the breakthrough to inductive-inference systems.
- **R E**: research focused on the natural representation of knowledge through the development of human-computer interaction, e.g. the Xerox Star direct manipulation of objects.
- **ET**: experience with the human-computer interaction using the problem-oriented language BASIC led to the innovative product of the Apple II personal computer.
- **TA**: the simplicity of the problem-oriented language RPG II led to the design of the IBM System/3 product line of small business computers.
- **AM**: special-purpose chips allowed the mass-production of low-cost, high-quality calculators.

By the end of the fifth generation (1980-88):

- **B R**: recognition of the goal-seeking possibilities of inductive inference systems led to breakthroughs in autonomous-activity systems.
- **R E**: research focused on knowledge acquisition for knowledge-based systems.
- **ET**: the advantages of the non-procedural representation of knowledge for human-computer interaction led to the innovative designs of Lisp and Prolog machines.
- **T A**: the ease of human-computer interaction through a direct manipulation problem-oriented language led to the Apple Macintosh product line of personal computers.
- **AM**: the design of highly-integrated language systems has allowed the mass-production of low-cost, high-quality software such as Turbo Pascal.
- **MM**: calculators have become so low in cost that replacement is preferable to repair.

By the end of the sixth generation (1988-96):

- **BR**: recognition of the cooperative possibilities of autonomous intelligent systems will lead to a breakthrough in socially organized systems.
- **RE**: research will be focused on autonomous intelligent behavior in systems such as neural networks and subsumption robots.
- **ET**: the advances in inductive systems will lead to new products for extracting knowledge from large datasets.
- **T A**: non-procedural problem-oriented languages will become routinely available on main-frame computers.
- **AM**: highly interactive personal workstations will drop in cost to a level where they become standard office equipment.
- **MM**: workstation replacement will have become more effective than maintenance and repair.

6 Significant Developments and Interactions

The BRETAM model can be used to highlight the significant developments in information technology for purposes of planning research, development and applications. Figure 4 left shows the cross section of Figure 3 that is relevant to the state of the art in information technologies during the previous, fifth generation of computers. The top three levels on the right of invention, research and innovation show why the fifth generation is generally recognized for its innovations in artificial intelligence. It was during this period that knowledge-based system products such as expert system shells first became available. However, in terms of reliance upon proven technology, it is the lower levels of product lines and below that are significant. The fifth generation was that in which human-computer interaction was dramatically improved through *graphic users interfaces, object oriented languages* brought control of complex system development in software engineering, and *networking* became ubiquitous. All these innovations took for granted advances in the underlying device technology that offered very fast powerful and reliable processors and large high-speed memories at low-cost.

Figure 4 right shows the equivalent picture of what is happening now as we progress through the sixth generation of computers. Hardware and networking have become almost negligible in cost and almost indefinitely powerful. Large-scale distributed systems are becoming readily available in terms of equipment and hardware architectures. Object oriented languages, and their associated application programming support environments (APSEs) and class libraries, are becoming routinely available at very low-cost. Graphic user interfaces (GUIs) are become standardized and portable across platforms as a routinely available technology. By the end of this generation the lowest level of knowledge-based system technology will have become available as well-supported product lines. These will support large-scale conceptual modeling at the enterprise level, the integration of heterogeneous information and

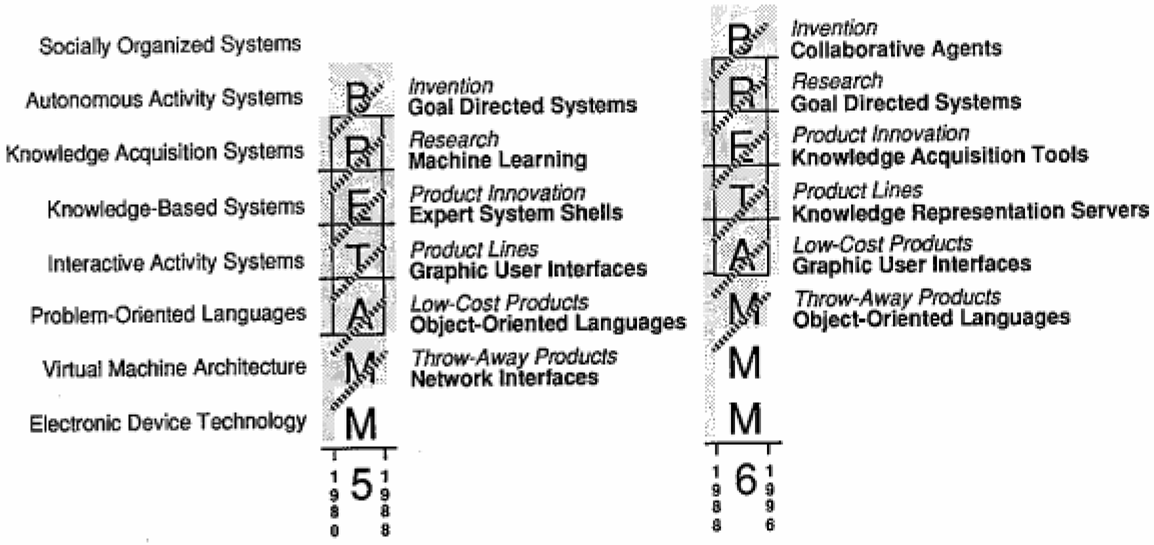


Figure 4 Significant technologies in the fifth and sixth generations

processes at lower levels, and the emulation of many aspects of human skilled behavior. Whether they are called "object oriented deductive databases," "second generation expert system shells," or "knowledge representation servers," or something completely different, is a matter for fashion, chaos theory, linguistics and marketing, to determine—we can already define their functionality and exhibit their application and that is enough.

If one focuses on a particular area of development or application, the BRETAM model may be used to examine the influences on it from technologies at different levels, and hence in differing states of maturity. For example, Figure 5 shows the influences on the development of Computer-Aided Software Engineering (CASE) tools during the fifth and sixth generations. CASE tools were developed as part of the *automation* phase for problem-oriented languages and, while commercial tools are designed to support many paradigms, the full impact is dependent on the development of formal *specification languages* subject to verification of correct implementation. The computer-support of CASE diagramming tools, group support, and theorem-proving verification methods has been dependent on the availability of low-cost high-power processors and displays from the electronic device technology level supporting workstation and networks at the virtual machine architecture level. The application of this low-level technology to CASE has been dependent on the development of graphic user interfaces at the interactive activity systems level. The complexity of reasoning required in operating a full CASE environment has made it a major target for applications of expert systems technology at the knowledge-based systems level, and we may expect an increasing application of machine learning techniques to support automatic programming as the knowledge acquisition systems level develops.

Thus, at any given level, there is dependency on the availability of the more mature technologies below, and support for further development from innovations in the less mature technologies above.

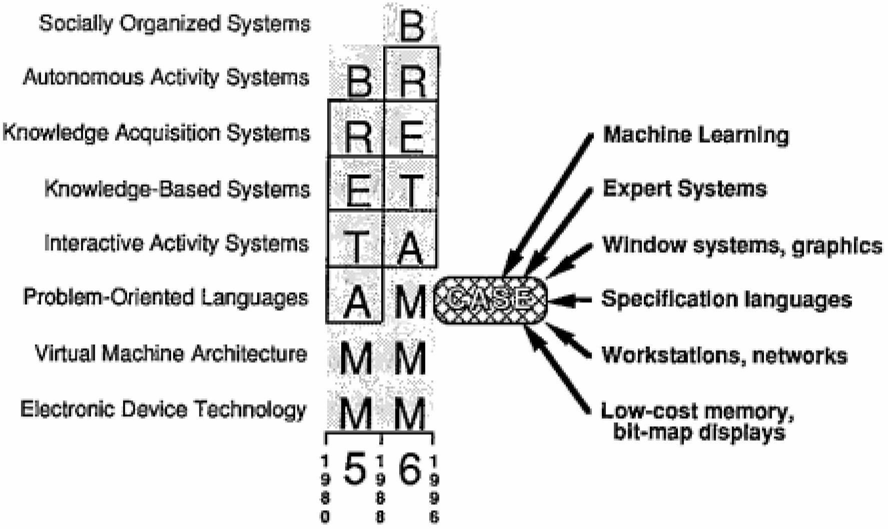


Figure 5 Influences between levels in development of Computer-Aided Software Engineering (CASE)

7 Current Issues at Each Level

The levels in the BRETAM model can be used as the basis for a check list of current issues in information technology.

In *electronic device technology* the packing densities are becoming so great that to sustain the line of growth in Figure 1 new approaches are needed. Electron beam lithography has been used experimentally to fabricate devices with gate lengths down to 65nm and current efforts are targeted on 20nm (Allee, Broers & Pease, 1991). There is also extensive exploration of alternative materials to silicon for the fabrication of computing devices. Nonlinear interactions between material and photon streams are capable of supporting computation and maintaining continuing research activities on optical computing technologies (Lebreton, 1991). Research also continues on organic semiconductors (Gunshor, 1988) where high fabrication densities are theoretically possible and genetic engineering techniques offer new manufacturing approaches. It is significant to note that the technology of DNA, its analysis and fabrication, is not only critically dependent on information technology but also is itself a parallel "information processing technology"—in organic material rather than silicon. The most innovative advances in materials during the fifth generation era have been those targeted on *nanotechnology*, of system fabrication at the molecular level. The molecular 'train sets' of today (Stoddart, 1991) are a fascinating curiosity that illustrate a breakthrough in both fabrication and instrumentation at the molecular level, and offer opportunities for new technologies which are currently at the limits of our imagination.

In *virtual machine architecture* the primary motivation has been to take advantage of the opportunities for parallel processing offered by modern vlsi technology. This has resulted in a very wide range of approaches from machines utilizing tens of thousands of conventional processor chips (Alder, 1988) to neural network technologies performing computations in radically different ways (Soucek, 1991). The essential tension between special-purpose computing and general-purpose computing that has existed since the earliest days of the digital computer underlies research activities and product innovations in this area. It is commonly stated as one of relation to the level above, of the provision of problem-oriented languages to interface computational requirements to computational resources, and it also a problem of integration, of combining special-purpose functionality with general-purpose support technology.

In *problem-oriented languages* the move towards a mature technology is apparent in many changing attitudes to the technology as well as in new developments. The application of a conceptual framework of manufacturing to software has become very fruitful and is a major thrust in Japan, Europe and the USA (Fernström, 1991; Humphrey, 1991; Basili, Caldiera & Cantone, 1992). However, the requirements for zero-defect, maintainable software manufactured from reusable parts has generated new requirements in the base technology. The development of Ada and C++ have been two practical extensions to previous technology addressing issues of reusability such as abstraction and encapsulation. However, there are also more fundamental trends towards theoretically well-founded languages such as PROLOG and ML that offer possibilities for combining verification with reasonable run-time performance. In the sixth generation era the combination of features and experience from logic programming, functional programming and object-oriented programming, functional programming and object-oriented programming to provide new generation languages will be a major area of research and development—Aït-Kaci's proposals for LIFE are an indication of what might be achieved (Aït-Kaci & Podelski, 1991).

In *interactive activity systems* the need to achieve interoperability between diverse equipment across high-speed local and wide-area networks has been a major developmental thrust in the fifth generation leading to widespread adoption of the ISO OSI standards (Day and Zimmerman, 1983), and to major standardization efforts at the upper levels concerned with data content and application integration (Modiri, 1991). There has been a parallel thrust in human-computer interaction where the need to develop products with platform-independent graphic user interfaces has led to the adoption of CMU's X-Windows and OSF's Motif as virtual standards under Unix, and the development of user interface management systems such as OIT and XVT that allow a single interface definition to be implemented automatically on virtually all personal computers and workstations. What is still missing in networking is the capability to integrate communities across wide area networks such that the system appears as a single entity rather than a local having one set of functions, loosely and heterogeneously coupled to a global network. What is missing in the user interface are the capabilities to recognize spoken language and to interact in natural language. These have been objectives since the first generation of computers and it is reasonable to expect that advances in all the lower level technologies supporting knowledge-based systems together with the development of more powerful knowledge bases make speech and language attractive targets for sixth generation development.

In *knowledge-based systems* the need to development secure theoretical foundations for knowledge representation may be seen as a major dynamic in all research. The acceptability of arbitrary heuristics in artificial intelligence systems declined in the fifth generation, and complexity analyses of basic knowledge representation systems began to identify the intrinsic problems of deductive knowledge representation systems (Brachman and Levesque, 1984; Nebel, 1988; Schmidt-Schauß, 1989). In the mid-1980s Aït-Kaci (1984, 1986) gave a lattice-theoretic model of knowledge base languages with operational semantics through term rewriting that resolved many of the issues of complexity and deduction algorithms for term subsumption knowledge representation systems. This ψ-calculus is particularly interesting because it provides foundations for complex object representation in deductive databases, for type computation in functional programming languages, and for knowledge representation in artificial intelligence. Developments based on sound theory are now targeted on the provision of knowledge representation services, and effort is becoming increasingly focused on the development of standardized modular systems with layers of well-defined and well-implemented functionality. This research is underway not only in the artificial intelligence community but also in the deductive database (Ohori, 1990), logic programming (Yardeni and Shapiro, 1991) and functional programming communities (Fuh and Mishra, 1990). This convergence of interests is to be expected as certain aspects of 'knowledge' and 'intelligence' are factored out to become realized by standard computational data structures and processes.

In *knowledge acquisition systems* the replication era of pragmatic copying of techniques and tools in the 1980s has given way to attempts to integrate methodologies and understand their underlying basis in the 1990s.

McDermott's (1988) *role limiting methods* are based on the abstraction of control knowledge from a family of related tasks, and the use of this to classify knowledge requirements and usage. As we identify role limiting methods we may come to rationalize them and develop alternative approaches that are more principled which is the basis of Chandrasekaran's (1988) analysis of *generic tasks*: The KADS research project (Akkermans, Harmelen, Shreiber and Wielinga, 1992) in the ESPRIT program has led to major advances in principled approaches to knowledge acquisition using a software engineering model. Apart from the development of more principled knowledge acquisition methodologies, we may also expect existing knowledge acquisition tools to be applied to the develop of large-scale knowledge bases in the sixth generation era. The Cyc project at MCC (Lenat & Guha, 1990) has been the outstanding example of such an attempt during the fifth generation, and the move towards knowledge interchange standards (Neches, Fikes, Finin, Gruber, Patil, Senator & Swartout, 1991) may be seen as supporting more widespread developments in the sixth generation.

In *autonomous activity systems* we are only at the beginning of an understanding of the nature of situated action and the interplay between the activities we interpret as 'planning' and 'representation' and the underlying neural processes which seem completely different in nature (Clancey, 1990). It is probable that for advances in this area we shall have to adopt a much wider perspective on knowledge-based systems in computing that analyzes their essential relationships to the cognitive and social knowledge processes of the human species (Gaines, 1991). Similar considerations apply to the expected breakthrough in *socially organized systems*. Current experiments in computer-supported cooperative work, intelligent agents surrogates, and distributed artificial intelligence are the necessary preliminaries to a major advance, and are reflected in the emphasis on system integration at lower levels. Socially organized will be a very active but relatively uncoordinated area of research throughout the sixth generation era.

8 Conclusions

It is interesting to examine the fifth generation program, and its world-wide emulation, in the light of the BRETAM model. In terms of timing, the Japanese announcement in 1982 came as knowledge-based system technology was moving into its empirical phase with expectations of significant product innovation by the late-1980s. The distinction between the lower four levels of 'computer science' and the qualitative change in the nature of information technology innovation in a new era of 'knowledge science' was also becoming apparent. The entire focus of the ICOT research and development program has been consistent with this rationale since it has concentrated on the enabling technologies for effective knowledge-based systems in terms of use of vlsi technology for new machine architectures supporting logic programming on interactive work stations. Thus the emphasis has been on the application of mature or maturing computer science technologies to provide a solid technological foundation for innovation in knowledge-based systems.

The parallel programs triggered off in the USA, UK and Europe, have not had such a specific focus but have tended to address the whole cross-section of fifth generation technology issues from chip encapsulation, through parallel architectures, specification languages, user interfaces, network protocols, knowledge representation systems, knowledge acquisition and planning. They have also addressed specific application domains significantly impacted by fifth generation developments, such as office automation and computer-integrated manufacturing. These other areas have also been addressed in Japan but not as part of the specific 'fifth generation' initiative.

As a final conclusion, it is reasonable to conjecture that the essence of sixth and seventh developments will that of treating 'knowledge' as the raw material to be processed rather than 'data.' This conclusion will not be surprising for anyone at this meeting, but it is one that may be given more quantitative substance and detail through the model presented in this paper. In particular, it is important to note that the shift towards a knowledge perspective in no way reduces our dependency on continuing advances in electronics, machine architectures, software engineering and improved human-computer interaction. These are the bedrock on which the foundations of knowledge-based systems are laid.

9 Acknowledgements

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada.

10 References

Aït-Kaci, H. (1984). A lattice-theoretic approach to computation based on a calculus of partially-ordered types. University of Pennsylvania, Philadelphia.

Aït-Kaci, H. (1986). An algebraic semantics approach to the effective resolution of type equations. **Theoretical Computer Science 45** 293-351.

Aït-Kaci, H. and Podelski, A. (1991). Towards a meaning of LIFE. Paris Research Laboratory, Digital Equipment Corporation. 1-42.

Akkermans, H., Harmelen, F. van, Shreiber, G. and Wielinga, B. (1992). A formalisation of knowledge-level models for knowledge acquisition. **International Journal of Intelligent Systems** to appear.

Alder, B.J., Ed. (1988). **Special Purpose Computers.** Boston: Academic Press.

Alexander, H. (1987). **Formally-Based Tools and Techniques for Human-Computer Dialogues.** Chichester, UK: Ellis-Horwood.

Allee, D.R., Broers, A.N & Pease, R.F.W. (1991). Limits of nano-gate fabrication. **Proceedings IEEE, 79**(8), 1093-1105.

Allen, J.F. (1985). Towards a general theory of action and time. **Artificial Intelligence, 23**(2), 123-154 (July).

Ayres, R.U. (1968). Envelope curve forecasting. **Technological Forecasting for Industry and Government: Methods and Applications.** pp. 77-94. Englewood-Cliffs, New Jersey: Prentice-Hall.

Balzer, R., Cheatham, T.E. & Green, C. (1983). Software technology in the 1990s: using a new paradigm. **Computer, 16**(11), 39-45 (November).

Basili, V.R., Caldiera, G. & Cantone, G. (1992). A reference architecture for the component factory. **ACM Transaction Software Engineering 1**(1), 53-80.

Boose, J H. (1989) A survey of knowledge acquisition techniques and tools. **Knowledge Acquisition 1** (1), 39-58 (March).

Brachman, R.J. and Levesque, H.J. (1984). The tractability of subsumption in frame-based description languages. **Proceedings of AAAI-84.** San Mateo, California, Morgan Kauffman. 34-37.

Brooks, R.A. (1990). Elephants don't play chess. Maess, P., Ed. **Designing Autonomous Agents.** pp. 3-15. Cambridge, Massachusetts: MIT Press.

Buchanan, B.G., Duffield, A.M. & Robertson, A.V. (1971). An application of artificial intelligence to the interpretation of mass spectra. Milne, G.W.A., Ed. **Mass Spectrometry Techniques and Applications**. New York: John Wiley.

Chandrasekaran, B. (1988). Generic tasks as building blocks for knowledge-based systems: the diagnosis and routine design examples. **The Knowledge Engineering Review 3**(3) 183-211.

Clancey, W.J. (1990) The frame of reference problem in the design of intelligent machines. In K. van Lehn & A. Newell, **Architectures for Intelligence: The Twenty-Second Carnegie Symposium on Cognition.** Hillsdale: LEA.

Connell, J.H. (1990). **Minimalist Mobile Robots**. Boston: Academic Press.

Crane, D. (1972). **Invisible Colleges: Diffusion of Knowledge in Scientific Communities**. Chicago: University of Chicago Press.

Davis, R. & Lenat, D.B. (1982). **Knowledge-Based Systems in Artificial Intelligence**. New York: McGraw-Hill.

Day, J.D. and Zimmerman, H. (1983). The OSI reference model. **Proceedings IEEE 71**(12) 1334-1340.

De Mey, M. (1982). **The Cognitive Paradigm**. Dordrect: Reidel.

Fernström, C. (1991). The Eureka software factory: concepts and accomplishments. **ESEC'91: 3rd Eurpean Software Engineering Conference**. Berlin, Springer. 23-36.

Fields, S.W. (1983). Silicon compiler cuts time to market for DEC's MicroVAX I. **Electronics, 56**(21), 47-48 (October).

Fuh, Y.-C. and Mishra, P. (1990). Type inference with subtypes. **Theoretical Computer Science 73**(2) 155-175.

Gaines, B.R. (1984). Perspectives on fifth generation computing. **Oxford Surveys in Information Technology, 1**, 1-53.

Gaines, B.R. (1986). Sixth generation computing: a conspectus of the Japanese proposals. **ACM SIGART Newsletter**, No.95, 39-44 (January).

Gaines, B.R. (1990). Intelligent systems as a stage in the evolution of information technology. Ras, Z.W. & Zemankova, M., Eds. **Intelligent Systems: State of the Art and Future Directions.** pp.431-455. Chichester, UK: Ellis Horwood.

Gaines, B.R. (1991). Modeling and forecasting the information sciences. **Information Sciences 57-58** 3-22 (Sep-Dec).

Gaines, B.R. (1991) Between neuron, culture and logic: explicating the cognitive nexus. **ICO: Intelligence Artificielle et Sciences Cognitives au Québec 3**(2) 47-61.

Gaines, B.R. & Shaw, M.L.G. (1986). A learning model for forecasting the future of information technology. **Future Computing Systems, 1**(1), 31-69.

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

Gevarter, W.B. (1983). Expert systems: limited but powerful. **IEEE Spectrum, 18**, 39-45.

Goguen, J.A., Thatcher, J.W. & Wagner, E. (1978). An initial algebra approach to the specification, correctness and implementation of abstract data types. Yeh, R., Ed. **Current Trends in Programming Methodology**. pp. 80-149. Englewood Cliffs, New Jersey: Prentice-Hall.

Gunshor, R.L, Ed. (1988). **Multifunctional materials**. Bellingham, Washington: Internnational Society for Optical Engineering.

Hansen, W.J. (1971). User engineering principles for interactive systems. **AFIPS Proceedings of the FJCC**. Vol. 39, pp. 523-532.

Hayes-Roth, F. (1984). The industrialization of knowledge engineering. Reitman, W., Ed. **Artificial Intelligence Applications for Business**. pp. 159-177. Norwood, New Jersey: Ablex.

Humphrey, W.S. (1991). Software and the factory paradigm. **Software Engineering Journal 6**(5), 370-376.

Karatsu, H. (1982). What is required of the fifth generation computer—social needs and impact. In Moto-oka (1982) pp. 93-106.

Kitsuregawa, M. & Tanaka, H., Eds. (1988). **Database Machines and Knowledge Base Machines**. Boston: Kluwer.

Lenat, D.B. & Guha, R.V. (1990). **Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project**. Reading, Massachusetts: Addison-Wesley.

Levreton, G.J., Ed. (1991). **Optics for Computers: Architectures and Technologies**. Bellingham, Washington: Internnational Society for Optical Engineering.

Marchetti, C. (1981). Society as a learning system: discovery, invention and innovation cycles revisited. **Report RR-81-29** (November). Laxenburg, Austria: International Institute for Applied Systems Analysis.

Mauchly, J.W. (1973). Preparation of problems for EDVAC-type machines. Randell, B., Ed. **The Origins of Digital Computers: Selected Papers**. pp. 365-369. Berlin: Springer-Verlag.

McDermott, J. (1988). Preliminary steps toward a taxonomy of problem solving methods. **Automating Knowledge Acquisition for Expert Systems**. Boston, Kluwer. 225-256.

Michalski, R.S. & Chilausky, R.L. (1980). Knowledge acquisition by encoding expert rules versus computer induction from examples—A case study involving soyabean pathology. **International Journal of Man-Machine Studies, 12**, 63-87.

Michalski, R.S. & Carbonell, J.G., Eds. (1983). **Machine Learning: An Artificial Intelligence Approach**. Palo Alto, California: Tioga.

Modiri, N. (1991). The ISO reference model entities. **IEEE Network Magazine 5**(4) 24-33.

Moto-oka, T., Ed. (1982). **Fifth Generation Computer Systems**. Amsterdam: North-Holland.

Nebel, B. (1990). **Reasoning and Revision in Hybrid Representation Systems**. Berlin, Springer.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. and Swartout, W.R. (1991). Enabling technology for knowledge sharing. **AI Magazine 12**(3) 36-56.

Ohori, A. (1990). Semantics of types for database objects. **Theoretical Computer Science 76**(1) 53-91.

Robinson, A.L. (1984). One billion transistors on a chip?. **Science, 223**, 267-268 (January).

Rosen, S. (1983). Generations, computer. Ralston, A. & Reilly, E.D., Eds. **Encyclopedia of Computer Science and Engineering**. pp. 657-658.

Schmidt-Schauß, M. (1989). Subsumption in KL-ONE is undecidable. **Proceedings of KR'89: First International Conference on Principles of Knowledge Representation and Reasoning**. San Mateo, California, Morgan Kauffman. 421-431.

Shortliffe, E.H. (1976). **Computer-Based Medical Consultations: MYCIN**. New York: Elsevier.

Soucek, B., Ed. (1991). **Neural and Intelligent Systems Integration**. New York: Wiley.

STA (1985). Promotion of Research and Development on Electronics and Information Systems That May Complement or Substitute for Human Intelligence. Tokyo: Science and Technology Agency.

Stoneman, P. (1983). **The Economic Analysis of Technological Change**. Oxford: University Press.

Stoddart, F. (1981). Making molecules to order. **Chemistry in Britain**, 714-718 (August).

Suchman, L.A. (1987). **Plans and Situated Actions**. Cambridge: University Press.

Van Dujin, J.J. (1983). **The Long Wave in Economic Life**. London: George Allen & Unwin.

Walter, C.J., Bohl, M.J. & Walter, A.B. (1968). Fourth generation computer systems. **AFIPS Proceedings of Spring Joint Computer Conference, 32**, 423-441.

Weegenaar, H.J. (1978). Virtuality and other things like that. **Proceedings of IEEE COMCON, CH1388-8/78**, 287-293.

Wilkins, D.E.. (1984). Domain-independent planning: representation and plan generation. **Artificial Intelligence**, 269-301.

Yardeni, E. and Shapiro, E. (1991). A type system for logic programs. **Journal Logic Programming 10**(2) 125-153.