Modeling and Forecasting the Information Sciences

Brian R. Gaines
Knowledge Science Institute
University of Calgary
Alberta, Canada T2N 1N4

Abstract

A model of the development of the information sciences is described and used to account for past events and predict future trends, particularly fifth and sixth generation priorities. The information sciences came into prominence as electronic device technology enabled the social need to cope with an increasingly complex world to be satisfied. Underlying all developments in computing is a tiered succession of learning curves which make up the infrastructure of the computing industry. The paper provides a framework for the information sciences based on this logical progression of developments. It links this empirically to key events in the development of computing. It links it theoretically to a model of economic, social, scientific and individual development as related learning processes with a simple phenomenological model. The fifth generation development program with its emphasis on human-computer interaction and artificial intelligence, and the sixth generation research program with its emphasis on knowledge science are natural developments in the foci of attention indicated by the model.

1 Introduction

Forecasting advances in technology and their impact has a track record of making fools of the forecasters (Schnaars, 1989). However, the game of life is one of anticipating the future. We model the past that we may learn the lessons of experience, and we extrapolate our models into the future. That the future reserves the right not be anticipated is a meta-lesson that Hume taught many years ago. Nevertheless, our individual behaviors and our civilizations are founded on the assumption that anticipation is possible and, at some level of modeling, history does repeat itself. Our technological civilizations go one step further and change the universe to reify our anticipations. The information sciences, in particular, structure a wholly artificial reality composed from the ultimate abstractions of the human mind. Paradoxically, they should be readily modeled because they are artifacts of our own mentation, but they may also be beyond modeling because to do so fully may involve ultimate understanding of ourselves.

This paper presents an integrative model of the information sciences that shows them as a tightly coupled, mutually supportive system. It first presents the underlying electronic device technology which provides both the basic support and physical constraints on information technology. It then presents a model of the learning curves of scientific and technological knowledge acquisition that underlies the development of the information sciences. The historic opportunities triggering successive advances in the information sciences are then analyzed and their learning curves superimposed to provide a model of the infrastructure of the information sciences. This model is extrapolated to provide forecasts of future directions, and fitted to various scientific and technological developments. The interactive synergies between levels are analyzed to show the basis of the positive feedback phenomena which continue to support the exponential growth of the information sciences and technologies.

2 Electronic Device Technology

There is a simple systemic model of the development of the information sciences. The precursors necessary to the birth of a new industry are a social need and a technology capable of satisfying it. The need allows a market to be generated. The technology allows products to be created for that market. For computers the enabling technology was electronics. The key operational concepts of a digital computer date back at least to Babbage, Lovelace and Boole in the nineteenth century (Goldstine, 1972). However, the mechanical technology of their time could not support a digital computer with adequate operations, storage capacity and speed.

The social need for computation in Babbage's time was navigation and its support through astronomical calculations themselves supported by the calculation of tables of logarithms. The enabling technology of electronic devices dates to the early 1900s with De Forest's development of the triode vacuum tube (Shurkin, 1984). However, it was not until the outbreak of World War 2 in 1940 generated the need for rapid calculation of ballistic trajectories that the social requirement for computation became sufficiently urgent. Mechanical device technology in the ball and disk differential analyzer had been taken to its limit and was inadequate. The pressures created by the urgent need to find an alternative technology supporting computation led to the use of vacuum tubes in the design of ENIAC by Mauchly and Eckert (Stern, 1981).

Thus the initial breakthrough for computing was in electronic device technology and it is interesting to see how well advances in electronic technologies accounts for generation changes in computing:

- The zeroth generation started in 1940 with the relay-based Bell Complex Number Computer followed in 1941 by the Zuse Z3.
- The first generation started in 1949 with the vacuum tube-based BINAC in the USA and EDSAC in the UK.
- The second generation started in 1956 with the transistor-based Bell Leprachaun followed in 1959 by the RCA 501.
- The third generation started in 1964 with the IBM 360 family using some integrated circuits.
- The fourth generation started in 1972 with the use of large-scale integration in the main memory of the IBM 370/158 and in the Intel 8008 8-bit microprocessor
- The fifth generation started in 1980 with the use of very large-scale integration by IBM to put the 370 processor on a chip followed by the HP-9000 in 1981 with 450,000 transistors on a chip.
- The sixth generation started in 1988 with ultra large-scale integration and some 16 million transistors on a chip.
- The seventh generation will start in 1996 with grand-scale integration and 1,000 million on a chip.

This definition of generations in terms of hardware works well for the zeroth through second generations because of the distinct qualitative changes involved. However, as Rosen (Rosen, 1983) notes in his analysis of computer generations 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 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 then the development of programming language compilers; the transistor made reliable operation possible and enabled routine electronic 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.

There is an important interplay in that these structural phenomena in that the information sciences have been, and continue to be, dependent on advances in electronics, but continuing developments in electronics are increasingly dependent on the information sciences. Modern microelectronics commenced in 1956 when silicon planar process was invented and enabled integrated circuit technology to be developed. 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). Figure 1 shows the exponential growth since that date:

- 20 in 1964 allowed the first flip-flop to be integrated.
- 5,000 in 1972 allowed the first 1 Kilobit ram (Intel 1103) and microcomputer (Intel 4004) to be integrated.
- 500,000 in 1980 allowed major central processing units such as the HP-9000 to be integrated.
- 16,000,000 in 1987 leading to research samples of 16 Megabit rams.

The current projected limit is some 1,000,000,000 million 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 shown in Figure 1.

Figure 1 shows an almost continuous progression on an exponential plot over 9 decades of performance increase. Exponential growth is common in many technologies, but never over more than two decades and then in periods of the order of 100 years rather than 10. 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. The distortion caused by the logarithmic plot is apparent in the linear plot of the same figures through the fifth generation as shown in in Figure 2. Now nothing appears to happen until 1980 when the technology suddenly shoots off. However, this is what has happened in every generation of computers as shown in Figure 3 where the exponential plot in each generation is rescaled to appear on the same vertical axis. There has been a revolutionary change in the base technology in every generation of computers.

Figure 3 explains why the continuous improvement in the technology shown in Figure 1 feels like a succession of revolutionary changes. During the eight year span of each computing generation, revolutionary changes have taken place that correspond in magnitude to those taking place over some seventy years or more in the aircraft industry. However, to analyze the detailed structure of these revolutions in computer architecture, problem oriented languages, and so on, a more detailed model of the infrastructure of the information sciences is required.

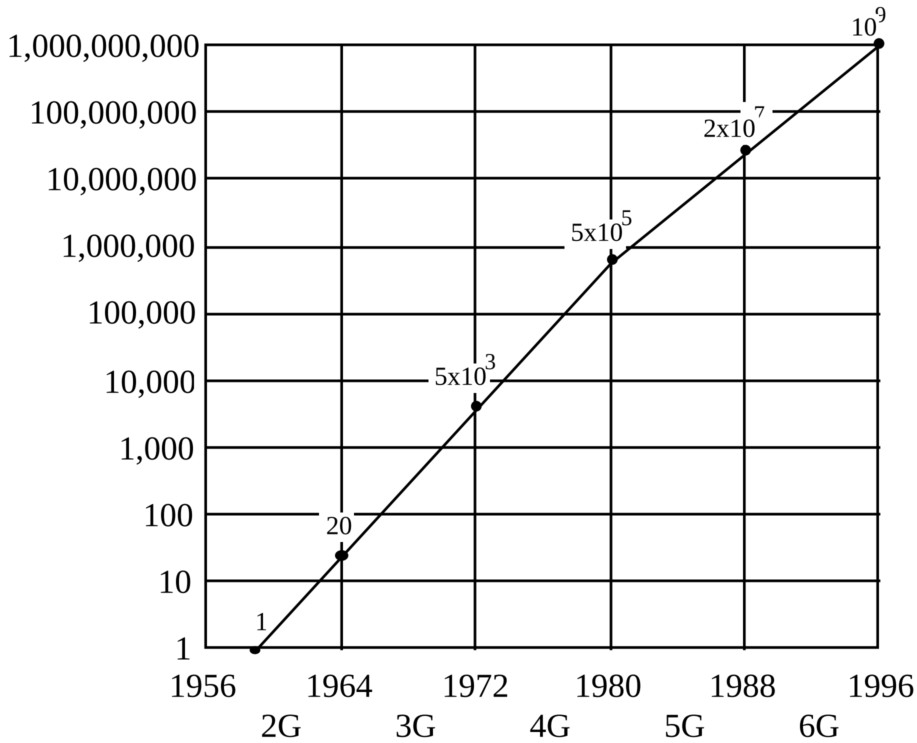

Figure 1 The number of devices on a chip

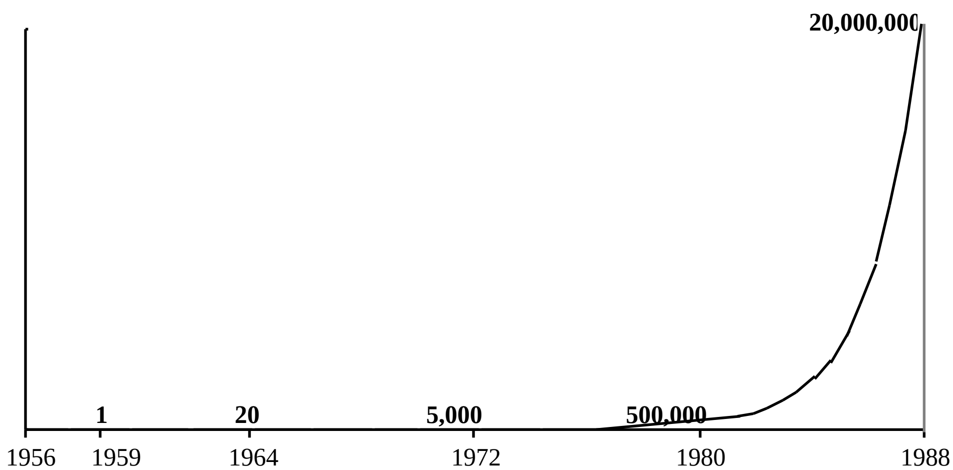

Figure 2 The number of devices on a chip on a linear scale

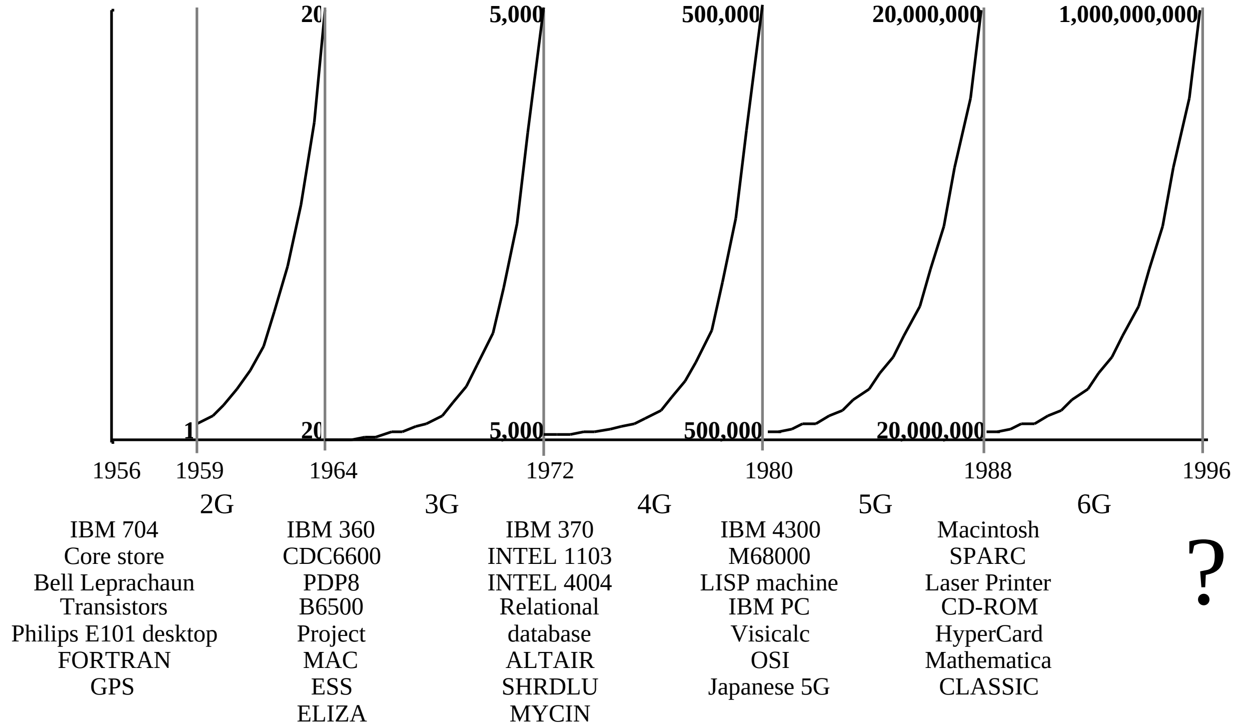

Figure 3 The number of devices on a chip during six generations of computers

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 (Marchetti, 1980). The logistic curve 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 (Dujin, 1983; Stoneman, 1983). One problem with using them predictively is that the asymptotic final level can only be estimated in retrospect, and attempting to determine the form of a logistic curve from data on the early parts is notoriously sensitive to error (Ascher, 1978). However, fitting logistic curves to historic data gives a very precise account of the development of major technologies such as the successive substitutions of one form of energy production for another (Marchetti and Nakicenovic, 1979).

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 and 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 but continuing a search based on intuition and faith. Sooner or later some inventor makes a *breakthrough* and 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. One automaton has been put in place effort can focus on cost reduction and quality improvements in what has become a *mature* technology.

Figure 4 shows this BRETAM sequence plotted along the underlying logistic learning curve. As noted previously, in most industries the learning curve takes some tens of years and the major effects are substitution ones. Substitution occurs an old technology has reached maturity and a new, more effective technology, reaches a point on its learning curve where it economically replaces the old one. There is also a secondary phenomenon that when a technology reaches a point on the learning curve where it is cost-effective and reliable new technologies develop dependent on the first one. For example, the electric lighting and appliance industries developed as the power generation industry came to offer cost-effective and reliable electricity supply.

The dependent technologies themselves develop along their own learning curves and may come to support their own dependents. Figure 5 shows a tiered succession of learning curves for dependent technologies in which a breakthrough in one technology is triggered by a supporting technology as it moves from its research to its empirical stage. Also shown are trajectories shown the eras of invention, research, product innovation, long-live product lines, and low-cost products. One phenomenon not shown on this diagram is that the new industries can sometimes themselves be supportive of further development in the industries on which they depend. Thus, in the later stages of the development of an industrial sector there will be a tiered structure of interdependent industries at different stages along their learning curves.

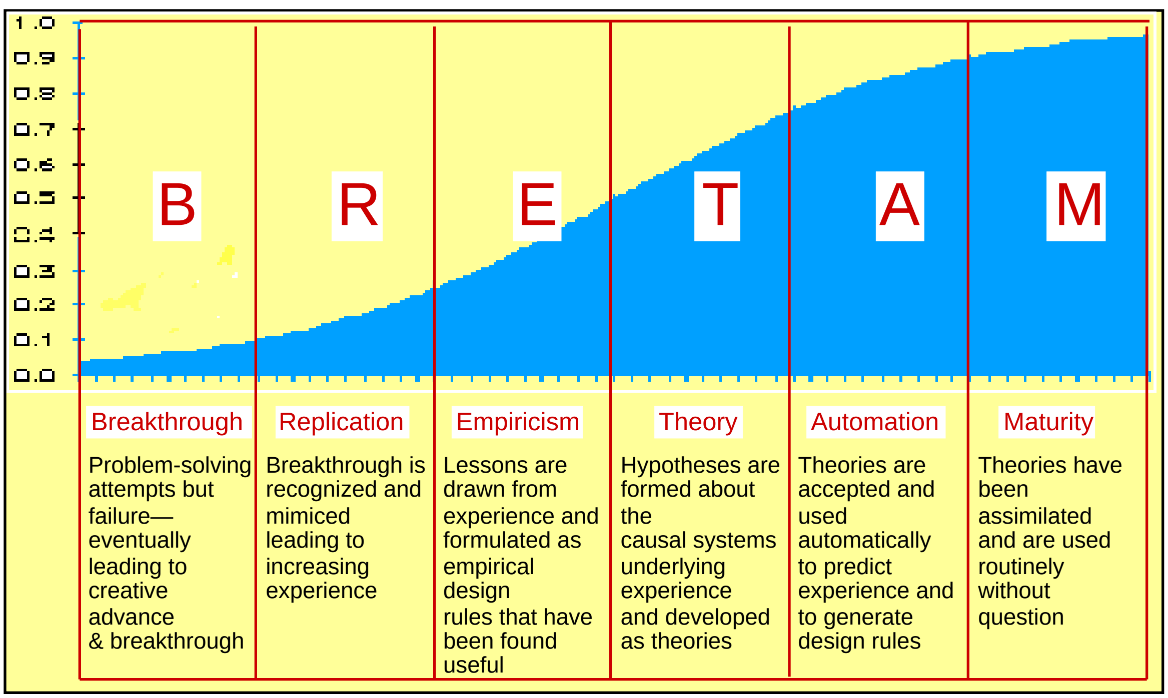

Figure 4 Qualitative changes along a technological learning curve

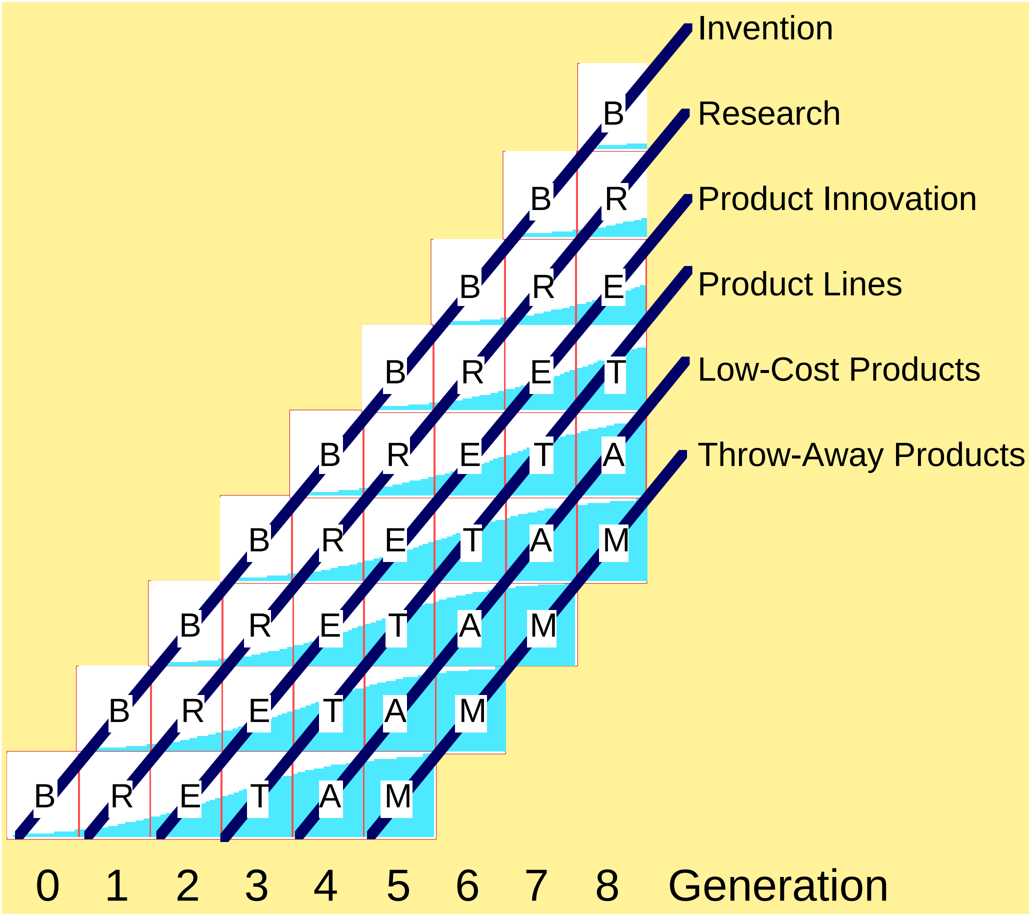

Figure 5 The tiered infrastructure of learning curves in the growth of a technology

4 The Infrastructure of the Information Sciences

Few industry sectors develop the elaborate infrastructure shown in Figure 5. However, 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 extremely well-defined and fundamental to their nature. This is shown in Figure 6.

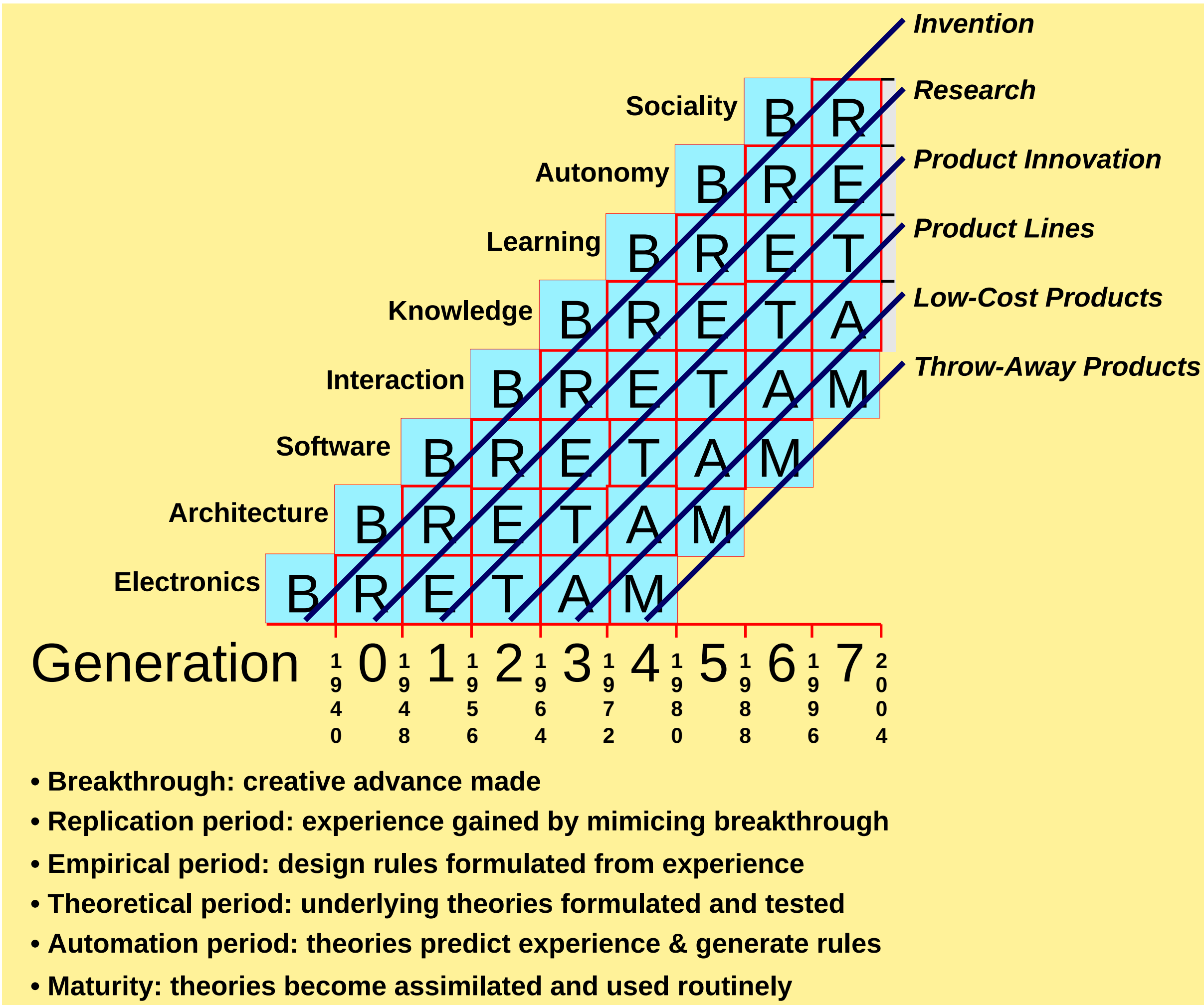

Figure 6 The infrastructure of the information sciences

The breakthrough in electronics leading to the zeroth generation is placed at 1940 about the time of the Atanasoff and Berry experiments with tube-based digital calculations. Experience was gained with COLOSSUS and ENIAC during the next eight years leading to a spate of empirical designs in the 1956-63 period, theoretical foundations for logic minimization and automata theory, automation with computer-aided design tools, and culminating into maturity with the 4004/8008 microprocessor chips in 1971/1972. Automation has 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 architecture which led to the transition from the ENIAC to the EDVAC designs. The key concepts were discussed by Mauchly in his paper of 1947 and the first implementations were the BINAC and EDSAC machines in 1949. Mauchly (Mauchly, 1947) recognized the significance of stored programs in enabling the machine instruction set to be extended, 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 use of subroutines to give access to arithmetic and operating system capabilities was followed by the development of machine architectures dependent on traps to procedures emulating missing hardware and led to theories such as those of semaphores, Petrinets and the logic of databases underlying diverse architectural concepts.

The next level of breakthrough was in software to bridge the gap between machine and task through the development of problem-orientated languages. Their foundations in the first generation were subroutine libraries providing virtual machines closer to the problem requirements, notably floating point arithmetic and mathematical functions. 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 and 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 and Green, 1983).

The next level of breakthrough was in 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 (Gaines and Shaw, 1983). The study of such systems led to design rules for HCI in the 1970s (Hansen, 1971) and theoretical foundations have started to emerge in the 1980s (Gaines, 1984). The improvement of human-computer interaction is 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 ISO open systems interconnection (OSI) layered protocol is an example of a theoretical model developed to unify and automate digital communications (Day and Zimmerman, 1983).

The next level of breakthrough was one of 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 and 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 massive VLSI production capability (Galinski, 1983) for the support of knowledge-based systems through PROLOG machines (Clark and Tarnlund, 1982) 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 its acquisition, breakthroughs in machine learning and inductive systems. Two breakthroughs in this area have been Lenat's AM learning mathematics by discovery (Davis and Lenat, 1982) and Michalski's inductive inference of expert rules for plant disease diagnosis (Michalski and Chilausky, 1980). In the fifth generation era machine learning became a highly active research area in its replication phase (Michalski and Carbonell, 1983; Michalski, Carbonell and Mitchell, 1986).

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 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, 1985)—in these developments the sixth generation breakthrough will come to be recognized, possibly in the notion of "situated action" (Suchman, 1987).

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. However, it is also reasonable to suppose in the light of past forecasting failures in computing technology that these speculations will be greatly in error. The projected breakthroughs may not occur or may occur much earlier. The recognized breakthroughs may be in completely different areas. The Japanese "Sixth Generation" research program proposals emphasize emulation of creativity and intuition and the development of inter-disciplinary *knowledge sciences* (STA, 1985; Gaines, 1986c). It is even 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 6 also shows the trajectories of invention, research, and so on, as superimposed on Figure 5. 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 RE 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 TA 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 the automated mass production of one technology and the mature technologies below.

For example, in the fourth generation (1972-79):

- BR: recognition of the knowledge acquisition possibilities of knowledge-based systems led to the breakthrough to inductive-inference systems.
- RE: 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.

In the fifth generation (1980-87):

- BR: recognition of the goal-seeking possibilities of inductive inference systems led to breakthroughs in autonomous-activity systems.
- RE: research focused on learning in knowledge-based systems.
- ET: the advantages of the non-procedural representation of knowledge for human-computer interaction led to the innovative designs of the Visicalc spread-sheet business product and the Lisp-machine scientific product.
- TA: 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.

By the end of the sixth generation (1988-95):

- 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.
- ET: the advances in inductive systems will lead to new products for extracting knowledge from large datasets.
- TA: 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.

6 An Example Application—Simulation

The tiered infrastructure model of Figure 6 is also useful tool for the analysis of the impact of the information sciences on particular industries that depend on information science and technology. The line of *product innovation* marks the practical availability of the various stages of new technology, and it lags the line of *invention* by sixteen years, and in its turn is lagged by the line of *low-cost products* by sixteen years. Thus there is a sixteen year gap between invention and significant application, and a thirty two year gap between invention and mass-production. Figure 7 shows the model applied to the evolving focus of attention in simulation:

In the first generation product innovation for simulation was concerned with the potential of electronic devices to simulate the behavior of differential equations. This led to the development of the operational amplifier in which a precision passive feedback network around an imprecise high-gain invertor is used to simulate the linear differential equations corresponding to the network (Smith and Wood, 1959). It also led to the development of the digital differential analyzer in which the differential equations are solved by incrementing digital registers (Lebedev, 1960; Mayorov and Chu, 1964).

In the second generation architectural considerations led to the combination of analog switches and operational amplifiers to give programmable analog computers. These could be combined with digital computers to give hybrid computers in which the analog sub-system is used for the rapid solution of differential equations and the digital sub-system is used to control decision processes, such as optimization, associated with the application of these equations (Korn and Korn, 1964). Digital differential analyzers were similarly combined with switching logic (Gaines and Joyce, 1967; Gaines, 1968) and general-purpose digital computers (Sizer, 1968).

In the third generation problem-orientated languages for simulation began to promote the standard digital computer as a simulation tool. The development of such languages as GPSS, SIMULA and SIMSCRIPT made available techniques for simulating systems combining linear and nonlinear elements in an environment where further statistical processing of data could also be simply programmed (Wexelblat, 1981). In this third generation era the flexibility and increasingly widespread availability of the general-purpose digital computer began to win out against the speed and cost-effectiveness of special-purpose analog computers and digital differential analyzers. It is also of interest to note that the blocks specifying asynchronous, interacting objects to be simulated in these languages were early precursors of the object-orientated programming languages of today.

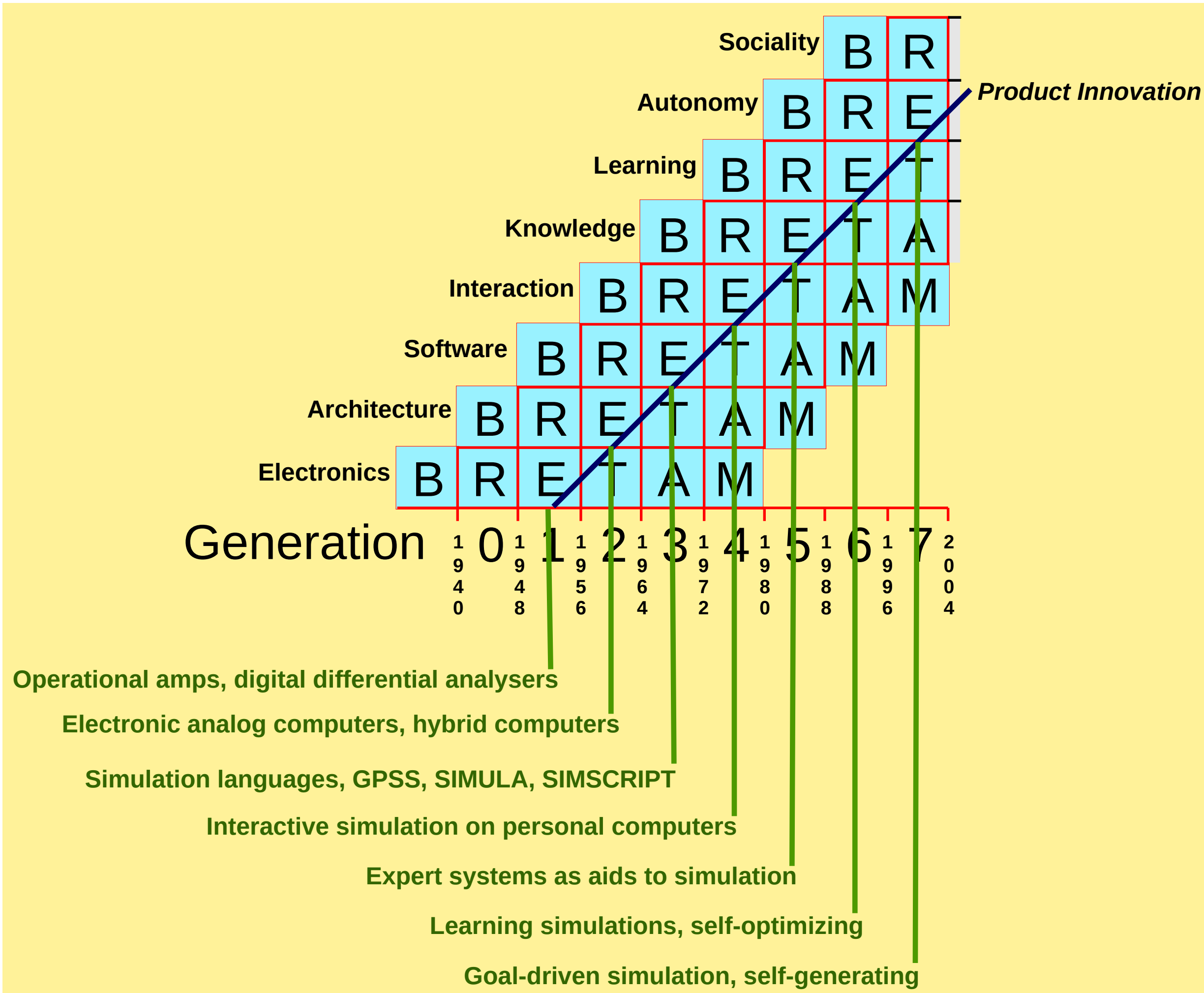

Figure 7 The evolving focus of attention in simulation

In the fourth generation the increased speed and reliability of computers shifted emphasis to interactive simulation and on-line decision-making. This again undermined the advantages of analog computers as interactive machines and these went out of manufacture during the fourth generation era. The human-computer interface to simulation languages, and ongoing simulations, became of major importance. In particular, the availability of low-cost color graphic interfaces made it possible to present both simulated structures and behaviors in pictorial and graphic forms more readily assimilated by people. The declining costs of personal computers with interactive graphics also opened up a wide range of applications of simulation in training, education and business management. A major example of this is the Xerox Star based on the simulation of the office desk-top and working environment (Smith, Irby, Kimball, Verplank and Harslem, 1983).

In the fifth generation the focus of attention at the frontier of simulation technology switched to expert systems in response to development of knowledge-based systems technologies in the computer industry (Gaines and Shaw, 1985; Shannon, Mayer and Adelsberger, 1985). The linguistic simulation of imprecise systems was shown to be feasible (Wenstop, 1981). The production rule languages used in expert systems were applied directly in symbolic and numeric simulation (Bruno, Elia and Laface, 1985). Simulation and expert system techniques were combined in integrated advisory systems (Gaines, 1986b; Gaines, 1986a; Wichmann, 1986).

In the sixth generation era we may expect product innovation based on machine learning to affect simulation, such as the use of inductive inference to allow real-time learning simulations to be developed as major new modeling tools. Seventh generation projections are more speculative but it is reasonable to suppose that such learning systems will form the foundation for autonomous robot technologies, in which knowledge acquisition and simulation will be combined to support goal-driven systems.

This type of analysis can be applied to any long-term technology that is heavily influenced by the information sciences, such as computers in education, military computing, computers in printing and so on. It enables investment and development strategies to be based on a reasonably detailed schedule of expected technologies and products arising out of the information sciences.

7 Positive Feedback in the Infrastructure

The tiered BRETAM model has so far been described as if the primary effects derived from the sciences and technologies in the lower layers supporting those above them. This is an important phenomenon but it is complemented by an equally important downwards influence in which the later sciences and technologies are used to continue developments in the earlier ones as they reach the later stages of their learning curves. Innovation becomes very difficult in the late stages of a learning curve because there is so much to be known in the sciences and so much to be controlled in the technologies. However, coping with knowledge and control is the primary function of the information sciences and, for example, developments in computers, software, interactive tools and expert systems are highly supportive of continuing developments in electronics and devices.

Figure 8 illustrates then phenomena for software engineering which has just entered its maturity phase. Computer-Aided Software Engineering (CASE) tools (McClure, 1989) which were the end product of the automation phase of software engineering are critically dependent on the availability of electronics supporting low-cost memory and bit-map displays. The require the architecture of modern networked workstations. They are based on the development of specification languages (Gehani and McGettrick, 1986) for software systems (that were themselves the outcome of the theory phase of software engineering).

These are all phenomena of support from below. The positive feedback effects come from support from above. CASE tools are highly interactive using window systems and graphics developed for human-computer interaction (Fisher, 1988). To deal with large scale system development they are beginning to incorporate knowledge bases using an expert systems approach (Loucopoulos and Harthoorn, 1988). Experiments in programming by example (Myers, 1988) are dependent on developments in machine learning.

It is possible to treat these feedback effects as an artifact of our having split the information sciences into components. An alternative view might be that there is only one learning curve and all other phenomena derive from the way we conceptualize it. This has an element of truth in that the information sciences do have a unity that derives from our needs to manage a complex social system (Beninger, 1986). Understanding the unity of the information sciences is important to our grasping the significant interactions between our compartmentalized specializations. The tiered

BRETAM model shows both the essential unity and the rich structure of the information sciences as we know them.

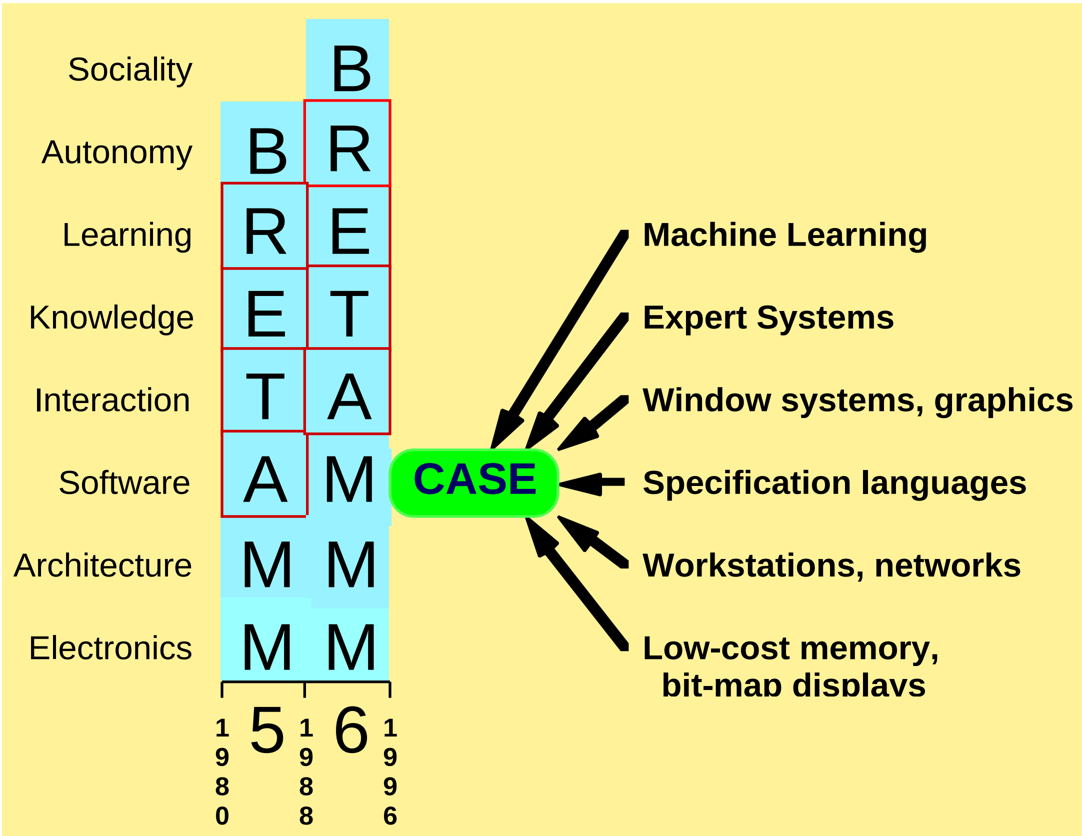

Figure 8 Upwards and downwards support of CASE in the infrastructure

8 Conclusions

A model of the development of the information sciences has been described and used to account for past events and predict future trends, particularly fifth and sixth generation priorities. The information sciences came into prominence as electronic device technology enabled the social need to cope with an increasingly complex world to be satisfied. Underlying all developments in computing is a tiered succession of learning curves which make up the infrastructure of the computing industry. The paper provides a framework for the information sciences based on this logical progression of developments. It has linked this empirically to key events in the development of computing. It has linked it theoretically to a model of economic, social, scientific and individual development as related learning processes with a simple phenomenological model. The fifth generation development program with its emphasis on human-computer interaction and artificial intelligence, and the sixth generation research program with its emphasis on knowledge science are natural developments in the foci of attention indicated by the model.

Acknowledgements

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

References

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

Ascher, W. (1978). **Forecasting: An Appraisal for Policy-Makers and Planners**. Baltimore, Maryland, John Hopkins University Press.

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

Beninger, J.R. (1986). **The Control Revolution: Technological and Economic Origins of the Information Society**. Cambridge, Massachusetts, Harvard University Press.

Bruno, G., Elia, A. and Laface, P. (1985). A rule-based system for production scheduling. **Proceedings COMPINT'85, IEEE 85CH2136-O**. pp.35-39.

Buchanan, B.G., Duffield, A.M. and 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.

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

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

Davis, R. and 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**. Dordrecht, Holland, Reidel.

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

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

Fisher, A.S. (1988). **CASE: Using Software Development Tools**. New York, John Wiley.

Gaines, B.R. (1968). Varieties of computer—their applications and inter-relationships. **Proceedings of IFAC Symposium on Pulse Rate and Pulse Number Signals in Automatic Control,**, Budapest, IFAC. 1-16.

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

Gaines, B.R. (1986a). The design of expert systems for planning flexible manufacturing. **Proceedings of Japan-USA Symposium on Flexible Automation**. pp.761-768. Osaka,

Gaines, B.R. (1986b). Expert systems and simulation in industrial applications. Luker, P.A. and Adelsberger, H.H., Ed. **Intelligent Simulation Environments**. La Jolla, California, Society for Computer Simulation.

Gaines, B.R. (1986c). Sixth generation computing: a conspectus of the Japanese proposals. **ACM SIGART Newsletter 95** 39-44.

Gaines, B.R. and Joyce, P.L. (1967). Phase Computers. **Proceedings of 5th Congress of International Association for Analog Computation**. pp.48-57.

Gaines, B.R. and Shaw, M.L.G. (1983). Dialog engineering. Sime, M. and Coombs, M.J., Ed. pp.23-53. London, Academic Press.

Gaines, B.R. and Shaw, M.L.G. (1985). From fuzzy sets to expert systems. **Fuzzy Sets and Systems 36**(1-2) 5-16.

Gaines, B.R. and 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.

Gehani, N. and McGettrick, A.D., Ed. (1986). **Software Specification Techniques**. Reading, Massachusetts, Addison-Wesley.

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

Goguen, J.A., Thatcher, J.W. and 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.

Goldstine, H.H. (1972). **The Computer from Pascal to Von Neumann**. Princeton, New Jersey, Princeton University Press.

Hansen, W.J. (1971). User engineering principles for interactive systems. **AFIPS Fall Joint Computer Conference**. pp.523-532. New Jersey, AFIPS Press.

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

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

Korn, G.A. and Korn, T.M. (1964). **Electronic Analog and Hybrid Computers**. New York, McGraw-Hill.

Lebedev, S.A., Ed. (1960). **Computer Engineering**. Oxford, Pergamon Press.

Loucopoulos, P. and Harthoorn, C. (1988). A knowledge-based requirements engineering support environment. **Proceedings Second International Workshop on Computer Aided Software Engineering, 1**. pp.13-10-13-14.

Marchetti, C. (1980). Society as a learning system: discovery, invention and innovation cycles revisited. **Technological Forecasting & Social Change 18** 267-282.

Marchetti, C. and Nakicenovic, N. (1979). The dynamics of energy systems and the logistic substitution model. International Institute for Applied Systems Analysis, Laxenburg, Austria. RR-79-13.

Mauchly, J.W. (1947). Preparation of problems for EDVAC-type machines. Randell, B., Ed. **The Origins of Digital Computers: Selected Papers**. pp.365-369. Springer, Berlin.

Mayorov, F.V. and Chu, Y. (1964). **Electronic Digital Integrating Computers—Digital Differential Analyzers**. London, Iliffe.

McClure, C. (1989). **CASE is Software Automation**. Englewood Cliffs, New Jersey, Prentice-Hall.

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

Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Ed. (1986). **Machine Learning: An Artificial Intelligence Approach Volume II**. Los Altos, California, Morgan Kaufmann.

Michalski, R.S. and 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.

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

Myers, B. (1988). **Creating User Interfaces by Demonstration**. Boston, Academic Press.

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

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

Schnaars, S.P. (1989). **Megamistakes: Forecasting and the Myth of rapid Technological Change**. New York, Free Press.

Shannon, R.E., Mayer, R. and Adelsberger, H.H. (1985). Expert systems and simulation. **Proceedings of Autofact'85**. pp.11.1-11.10. Detroit, Michigan, Society of manufacturing Engineers.

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

Shurkin, J. (1984). **Engines of the Mind: A History of the Computer**. New York, Norton.

Sizer, T.R.H., Ed. (1968). **The Digital Differential Analyser**. London, UK, Chapman & Hall.

Smith, D.C., Irby, C., Kimball, R., Verplank, B. and Harslem, E. (1983). Designing the Star user interface. Degano, P. and Sandewall, E., Ed. **Integrated Interactive Computing Systems**. pp.297-313. Amsterdam, North-Holland.

Smith, G.W. and Wood, R.C. (1959). **Principles of Analog Computation**. New York, McGraw-Hill.

STA (1985). **Promotion of R&D on Electronics and Information Systems That May Complement or Substitute for Human Intelligence**. Tokyo, Science and Technology Agency.

Stern, N. (1981). **From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers**. Bedford, Massachusetts, Digital Press.

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

Suchman, L. (1987). **Plans and Situated Actions: The Problem of Human Machine Communication**. Cambridge, UK, Cambridge University Press.

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

Wenstop, F. (1981). Deductive verbal models of organizations. Mamdani, E.H. and Gaines, B.R., Ed. **Fuzzy Reasoning and its Applications**. pp.149-167. London, Academic Press.

Wexelblat, R.L., Ed. (1981). **History of Programming Languages**. New York, Academic Press.

Wichmann, K.E. (1986). An intelligent simulation environment for the design and operation of FMS. Lenz, J.E., Ed. **Proceedings of 2nd International Conference on Simulation in Manufacturing**. pp.1-11. Bedford, UK, IFS Conferences.

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