High-Tech Hayekians: Some Possible Research Topics in the Economics of Computation

by Don Lavoie, Howard Baetjer and William Tulloh

Lavoie, Don, Baetjer, Howard, & Tulloh, William 1990. “High Tech Hayekians: Some Possible Research Topics in the Economics of Computation.” *Market Process*, 8.

In a thoroughly intriguing set of papers recently published in a book edited by B. A. Huberman entitled *The Ecology of Computation*, two computer scientists, Mark S. Miller and K. Eric Drexler, have made major advances in what might be called the *Economics of Computation*, a new, largely unexplored subdiscipline on the interface of economics and computer science. What makes this research especially interesting to several of us at the Center for the Study of Market Processes is that the kind of economics Miller and Drexler are using to overcome their own problems in computer science is "our kind." This was no accident. It turns out that Miller and Drexler were specifically alerted to the economic writings of Hayek by Phil Salin. Reading Miller and Drexler's papers, we were struck by the depth of understanding these computer scientists have of our literature, especially the work on property rights economics and the spontaneous order theory of F. A. Hayek. The success of these pioneering papers suggests that there may be some interesting research possibilities in computer science for the application of market process ideas.

In August of 1989, the three of us made a trip to Palo Alto to visit Miller and Drexler at the Xanadu Operating Company and the American Information Exchange Corporation, or "AMIX, " the two software design companies where Miller and Salin work, and for whom Drexler is a consultant. We were extremely impressed. Our time in Palo Alto was so replete with ideas, discussion and new insights that we can hardly begin to summarize them all. We came hoping to find one or two possible research topics. We returned with too many possible topics for the three of us to handle. Throughout the visit the conversations always involved Hayekian ideas, but at the same time they almost always involved computer science. Miller, Drexler, Salin, *et al.* consider market process ideas to be of enormous practical usefulness in the design of commercial software. In this note we would like to introduce the reader to the most important ideas we encountered, and indicate why we think market process economists should look into this fascinating work.

In all, we spoke to some twenty to thirty young computer scientists who are energetically applying market process insights to some very ambitious software ventures. They are not primarily academics, but they have academic interests. These academic interests, however, are focused on practical, real-world applications. How shall we describe this group? Each of them seems to be a rich brew of scientist, entrepreneur, programmer and philosopher. Each has a firm understanding of market principles and a deep appreciation for the general social principle of non-intervention, for reasons to be described shortly. The organizations they are involved in are start-up companies that are developing software to help people manage information in ways that enable knowledge to evolve more rapidly. We visited them in their homes and in their offices. The animation of our discussions rarely slackened, even in the cars on the way to restaurants.

The three people most intensely involved with market process ideas are as follows:

Mark S. Miller--software engineer formerly with Xerox Palo Alto Research Center, flowing fountain of ideas and enthusiasm, co-author of four of the essays in the Huberman book, and now chief architect for Xanadu. Mark has derived many insights about how to design computer programs from studying Hayek.

K. Eric Drexler--formerly with the artificial intelligence laboratory at MIT, now with the Department of Computer Science at Stanford University, and author of *Engines of Creation, the Coming Era of Nanotechnology,*an investigation of the nature and implications of startling advances in technology toward which we are moving. Nanotechnology refers to the next step beyond microtechnology, the ability to achieve technological control over matter at the molecular level. With serious interests ranging from molecular engineering and nanomachinery to Austrian economics, Drexler defies categorization in customary terms. He might best be described as a 1990's version of a Renaissance thinker. He assists with the Xanadu design work, and with his wife Chris Peterson runs the Foresight Institute, an organization devoted to preparing the world for nanotechnology.

Phil Salin--an entrepreneur, formerly in the telecommunications and space transportation industries and now on Xanadu's board of directors. He came to see the importance of Hayek's thought for understanding the business world. He was instrumental in bringing the Xanadu group to Palo Alto.

We also got the chance to visit with Ted Kaehler, a software engineer with Apple Computer Corporation and an expert on the field of Artificial Intelligence known as "machine learning," especially the approach called "neural networks." Ted is very interested in using market principles in the development of AI systems.

We also spoke with Marc Stiegler, vice president of Software Engineering at Xanadu and a professional writer of science fiction. He brings to Xanadu a constant reminder of the practical steps that need to be taken on a daily basis to reach the company's ambitious goals. We spoke to many other programmers on the Xanadu and AMIX teams. All shared an expectant outlook on the future, for computer science and for society. And yet these individuals are software engineers focused on practical concerns. They are not just speculating, but busy with the details of building systems that are immensely practical, however astounding they may be.

These individuals are coauthors with us, in a sense, of this paper, since the whole is a summary of our discussions with them. The development of the research ideas outlined below was very much a team effort. As usual with such efforts, the total output exceeds the sum of the inputs. We are grateful to have been on the team.

It would be useful to distinguish five different kinds of research areas that our discussions hit upon, within each of which we found several specific possible projects. Any given project may easily overlap these research areas, but it will help to keep them distinct, separating them by what each takes as its primary focus.

1. Process-Oriented Case-Studies of the Computer Industry

As with most media from which things are built, whether the thing is a cathedral, a bacterium, a sonnet, a fugue or a word processor, architecture dominates material. To understand clay is not to understand the pot. What a pot is all about can be appreciated better by understanding the creators and users of the pot and their need both to inform the material with meaning and to extract meaning from the form.

There is a qualitative difference between the computer as a medium of expression and clay or paper. Like the genetic apparatus of a living cell, the computer can read, write and follow its own markings to levels of self-interpretation whose intellectual limits are still not understood. Hence the task for someone who wants to understand software is not simply to see the pot instead of the clay. It is to see in pots thrown by beginners (for all are beginners in the fledgling profession of computer science) the possibility of the Chinese porcelain and Limoges to come.

--Alan Kay (1984, p. 53)

One type of research project which frequently came up was simply to use the tools of market process economics to examine the specifics of the computer industry. Such studies would be in keeping with the overall thrust of the research combining theoretical and empirical study that has been going forward at the Market Processes Center over the last several years. Not the least important reason economics might study computation is that a growing and increasingly important part of the economy consists of the computer industry.

One example of the kind of topic we have in mind here is considering software as capital. It is a cliche that we are entering the information age, and much of the capital "equipment" that counts today is in the form of instructions to computers. Software is a special kind of capital good with some economically interesting characteristics. Market process economics, since it concentrates on the way knowledge is used in society, may in fact find this industry especially intriguing. Increasingly computers play a central role in the process of knowledge transmission.

Studying software as capital may help us illuminate the market process approach to capital theory. Few kinds of tools today are more important than software, for software increasingly directs our "hard" tools. But software does not fit the assumptions made about capital goods in mainstream theory. It is reproducible at negligible cost; its use by one agent does not preclude its use by another; the major costs associated with it are information costs (often neglected in neoclassical theorizing); it does not wear out when used.

Market process economics emphasizes that capital is a complex structure, involving time and uncertainty, and it views markets as disequilibrium processes. The capital goods we call software are especially heterogeneous; not only do different programs accomplish entirely different tasks, but they are written in many different, incompatible languages. The patterns of complementarity of software-capital are exceptionally complex: most programs can run only on certain kinds of machines, performing tasks useful only in certain kinds of endeavors. A given CAD-CAM (computer aided design-computer assisted manufacture) system, for example, may require a particular computer with particular specifications, and a particular manufacturing device. In the software industry the detailed timing of events is particularly important. Change is so rapid that a product not released in time is doomed, and capital destruction, through the advance of knowledge, occurs constantly. Old programs and machines created at great cost and bought at great price become obsolete in months.

Conceiving of software as capital goods calls into serious question the common treatment of capital in economic models as accumulating solely out of saving.

In many growth models, for instance, new capital in a given period is defined as the unconsumed product of the previous period. This model implies that the most important thing new capital requires is previous physical output. It ignores what the Austrian view emphasizes: the role of knowledge and creativity. Beer barrels and blast furnaces are more than the saved wood and steel from a previous period. They are additionally the embodiment of careful plans and much accumulated knowledge. The new capital is a combination of the physical wood and steel with this essential knowledge.

With software, the knowledge aspect of capital reaches its logical extreme. Software has virtually no physical being. It is essentially a pattern of on and off switches in the computer. As such, it is pure knowledge. No physical output from a previous period is required to produce it, except perhaps for the Coca-Cola (a.k.a. "programming fluid") and Twinkies the software engineer consumes as he generates the code. The knowledge and creativity of the programmer are crucial.

Other studies that fall within this research category include studies of the evolution of standards and of the development of interoperability. What becomes an industry standard is of immense importance in determining the path an industry takes. As, say, some particular operating system appears to be outstripping others in popularity, as MS-DOS did in the microcomputer industry, for example, the first versions of new programs tend to be written for that operating system, to improve their chances of acceptance. This is often the case whether or not that operating system is really the best available for the program's purpose. In this way, as software developers bet on an uncertain future, industry standards develop. "Interoperability" means making it possible for software that is built to one standard to be used with another standard via "software adapters." With interoperability, now available in limited degree, software developers have less at stake in using some less-preferred programming language or operating system that better suits their needs. Decisions to choose on the merits rather than by projections of others' preferences will tend to influence the evolution of industry standards. What are the economic causes and consequences of the evolution of a particular industry standard? What would be the economic consequences of more widespread interoperability? These are interesting questions worthy of research.

In sum, studying the computer industry seems likely to inform our understanding of economic processes from a new, illuminating perspective.

2. Information Technology and the Evolution of Knowledge and Discourse

Knowledge evolves, and media are important to the evolution of knowledge. Hypertext publishing promises faster and less expensive means for expressing new ideas, transmitting them to other people, and evaluating them in a social context. Links, in particular, will enable critics to attach their remarks to their targets, making criticism more effective by letting readers see it. Hypertext publishing should bring emergent benefits in forming intellectual communities, building consensus, and extending the range and efficiency of intellectual effort.

--Drexler (1987, p. 16)

Both the Xanadu and AMIX groups pay much attention to evolutionary processes, which is one reason Hayek's work is so attractive to them. They are particularly interested in the evolution of knowledge. One of the crucial questions which they focus on is the evolution of ideas in society. In this regard, the hypertext publishing system under development at Xanadu is of great importance. It will, they hope, provide a market for information of a scientific and scholarly kind that will more closely approach the efficiency standards of our current markets for goods and services.

The market for scholarly ideas is now badly compartmentalized, due to the nature of our institutions for dispersing information. One important aspect of the limitations on information dispersal is the one-way nature of references in scholarly literature. Suppose Professor Mistaken writes a persuasive but deeply flawed article. Suppose few see the flaws, while so many are persuaded that a large supportive literature results. Anyone encountering a part of this literature will see references to Mistaken's original article. References thus go upstream towards original articles. But it may be that Mistaken's article also provokes a devastating refutation by Professor Clearsighted. This refutation may be of great interest to those who read Mistaken's original article, but with our present technology of publishing ideas on paper, there is no way for Mistaken's readers to be alerted to the debunking provided by Clearsighted. The supportive literature following Mistaken will cite Mistaken but either ignore Professor Clearsighted or minimize her refutations.

In a hypertext system such as that being developed at Xanadu, original work may be linked downstream to subsequent articles and comments. In our example, for instance, Professor Clearsighted can link her comments directly to Mistaken's original article, so that readers of Mistaken's article may learn of the existence of the refutation, and be able, at the touch of a button, to see it or an abstract of it. The refutation by Clearsighted may similarly and easily be linked to Mistaken's rejoinder, and indeed to the whole literature consequent on his original article. Scholars investigating this area of thought in a hypertext system would in the first place know that a controversy exists, and in the second place be able to see both (or more) sides of it with ease. The improved cross-referencing of, and access to, all sides of an issue should foster an improved evolution of knowledge.

A potential problem with this system of multidirectional linking is that the user may get buried underneath worthless "refutations" by crackpots. The Xanadu system will include provisions for filtering systems whereby users may choose their own criteria for the kinds of cross-references to be brought to their attention. These devices would seem to overcome the possible problem of having charlatans clutter the system with nonsense. In the first place, one would have to pay a fee for each item published on the system. In the second place, most users would choose to filter out comments that others had adjudged valueless and comments by individuals with poor reputations.[1] In other words, though anyone could publish at will on a hypertext system, if one develops a bad reputation, very few will ever see his work.

Another difficulty of today's paper publishing environment is that turn-around times are extensive. Mistaken's persuasive but flawed article can be the last word on a subject for a year or so, before Clearsighted can get her responses accepted in a journal and then published. Even then, many of those who have read Mistaken may not have ready access to the journal in which Clearsighted is published. In a hypertext publishing environment, by contrast, Clearsighted's responses can be available literally within hours of the publication of Mistaken's original article. Thus a hypertext system seems able to inhibit the spread of bad ideas at their very roots. Refutations of bad ideas could immediately become known, and unorthodox new ideas, if sensible, could more rapidly gain the support of fair-minded thinkers.

The research program for economists that is suggested by hypertext publishing falls within the field of the philosophy of science. It involves interpreting and explaining the effects of technology on the shape of economic research in the past, and projecting how advances such as hypertext might reshape it in the future. In the early days of economics as a profession, book printing was the leading edge of technology, and accordingly most economics consisted of verbal arguments. In more recent years the advent of number-crunching computers has enabled the development of more complex mathematical modeling and econometric techniques. But today's microcomputers, through word processing, have also begun to dramatically enhance economists' ability to present verbal arguments. (This very sentence, for instance, was all but effortlessly inserted in the fourth draft of this article, with no necessity for retyping or cutting and pasting.) The advent of hypertext might revitalize the rich literary tradition of economics in at least three ways. First, it would facilitate research by drastically reducing time spent among library shelves. Second, it would enable easier reference to, and more rapid dissemination of, verbal arguments. Third, it would provide a medium through which any number of economists might easily carry on open-ended written discussions with one another, with no inconvenience of time and distance.

3. Complexity, Coordination and the Evolution of Programming Practices

Two extreme forms of organization are the command economy and the market economy. The former attempts to make economic tradeoffs in a rational, centrally-formulated plan, and to implement that plan through detailed central direction of productive activity. The latter allows economic tradeoffs to be made by local decision makers, guided by price signals and constrained by general rules.

Should one expect markets to be applicable to processor time, memory space, and computational services inside computers? Steel mills, farms, insurance companies, software firms--even vending machines--all provide their goods and services in a market context; a mechanism that spans so wide a range may well be stretched further.

--Miller and Drexler (1988b, p. 137)

The type of research that would be most directly a follow-up of the paper "Markets and Computation: Agoric Open Systems" by Miller and Drexler, would involve examining the history, and the possible future, of computer programming practices as illustrative of economic principles. Inside computers things are going on which have some amazing similarities--and of course some significant differences--to what goes on in human economies. Many of the problems that arise in human economic systems have their analogs in well-known problems in computational systems.

Economics has been conceived by many of its practitioners (e.g., Ludwig Mises, Lionel Robbins, Gary Becker) as applicable in general to the study of choice, the making of decisions, the application of scarce means to valued ends. Programmers are faced with difficult choices of how to make the best use of scarce computational resources. F .A.Hayek has recast the economic problem in terms of how societies may make effective use of the knowledge that is so widely dispersed among all the people they comprise. Hayek has argued that coordination in complex systems such as human economies exceeds the capabilities of central planning and direction. Coordination is achievable in complex economies only through decentralized decision-making processes: through specialization and the division of labor, through property rights, and through the price system.

Programmers are now facing similar problems of complexity. As programs and distributed computation systems grow larger, they are outrunning the capacity of rational central planning. Coping with complexity seems to depend on decentralization and on giving computational "objects" property rights in their data and algorithms. Perhaps it will even come to depend on the use of price information about resource need and availability that can emerge from competitive bidding among those objects.

As the following paragraphs indicate, there is much that could be done in the direction of elaborating on these analogies between economics and programming practices, and using them to develop a better understanding of economics. It doesn't matter where one stands on the question of how similar computational and market processes are for one to see the possible value of this research program. Even if the differences are enormous, explaining exactly what they are could be exceedingly valuable to both fields.

The Division of Labor and Modularity

An appreciation of the advantages of the division of labor is embodied in the programming principle of modularity. The earliest programs were linear, undivided sequences of instructions, but with the evolution of programming, practical considerations forced a dividing up of the problem into discrete modules. The extensive use of subroutines and structured programming enhanced the ability of programmers to solve their problems. They broke down the whole into manageable chunks, as it were, whose activities were known and clearly bounded.

For the various subroutines to operate effectively, they need a certain amount of autonomy--if other parts of the program interfere with them in unexpected ways, the result is a crashed program or nonsense. Or, we might say, the subroutines' "rights" to what is "theirs" need to be respected. The analogy to property rights is very close.

Property Rights and Object-Oriented Programming

The practical advantages that property rights give the economy can be provided for computer programs by what is called "object-oriented programming" (the common acronym in the computer language literature is OOPS, for object-oriented programming systems). In object-oriented programming, the different kinds of tasks that the program must carry out are assigned to "objects," essentially autonomous sections of code whose workings cannot be interfered with by other parts of the program, because the boundaries between objects are clear and respected. One subroutine's data cannot, "intentionally" or by accident, interfere with data "belonging to" another subroutine. One advantage to the programmer is that he need not hold in his mind all at once the myriad possible options and combinations of calculations and data. The programmer need not know how an object works, only what it does. Another advantage is that if an object is directed to do something it cannot do, it simply returns a message that it "does not understand," instead of acting on the bad instructions in a senseless or destructive way. The program's "labor" is thus not only divided up among many parts, but the "rights" of these parts are respected.

The integrity of the various data structures and algorithms in OOPS provides especially welcome clarity in very large programs written by teams of programmers. In this setting the likelihood of mutual interference would be very great--analogous to the tragedy of the commons--without the "property rights" structure of OOPS.

The Use of Knowledge in Computation

At one point Roger Gregory, a founder of Xanadu, made a comment which helped us understand why his group is so interested in market process economics. The reason is that programmers, in their day-to-day experience, cannot help but learn the severe difficulties in getting large, centrally planned systems to work properly. The bigger and more complex their own programs, the more difficulty they have with bugs. Virtually never does a large program work as intended the first time, even though the programmer has absolute control over every aspect of the system. Programmers spend many frustrated hours debugging their own work. Accordingly, they tend to be very dubious about the ability of government planners to develop successful systems in human society, where the complexity is far greater and the ability to control is far less.

What this suggests of course, is Hayek's line of thought that the more complex a system is, the more necessary it becomes that the orderliness of the system grow out of the interaction of relatively autonomous parts. The nature of complexity is arguably the central issue in computer science today, as it is in Hayekian economics. As Mises and Hayek have emphasized, the reason that central planning of economies does not and cannot work is that human society is too complex. Programmers are beginning to realize that "central planning" of computational systems is fraught with the same difficulties.

So far we have been describing economic insights that are already widely appreciated by computer scientists, albeit not in the same terms economists use. The most innovative aspect ofMiller and Drexler's papers is their introduction of "agoric systems." Derived from the Greek word for marketplace, agoric systems aim at solving the problem of maintaining coordination in complex computational systems by the same means as in complex economies: by a price system.

As Miller and Drexler (1988b, p. 163) put it:

Experience in human society and abstract analysis in economics both indicate that market mechanisms and price systems can be surprisingly effective in coordinating actions in complex systems. They integrate knowledge from diverse sources; they are robust in the face of experimentation; they encourage cooperative relationships; and they are inherently parallel in operation. All these properties are of value not just in society, but in computational systems: markets are an abstraction that need not be limited to societies of talking primates.

Miller and Drexler are concerned about the efficiency of computer resource use, both within particular programs running on single computers and across extended computational networks over large geographical areas. In both cases they envision allocation of scarce computational resources such as disk or memory space and processor time--being determined by a market process among computational objects. As in the economy, it is not enough that agents have property rights; it is necessary also that they be able to communicate their special knowledge of time and place.

Traditional programming practices have been based on a central planning approach, deliberately deciding on tradeoffs, such as that between the speed at which the program completes its task with the core space it takes to do so. Computer time has traditionally been allocated on a time-share or first-come-first-served basis, or by some other fixed prioritizing system.

Miller and Drexler wish to persuade the computer community to drop this central planning model for allocating computational resources. Instead, programs should be designed so that their different parts would "bid competitively" for, say, the "rental" of memory space, which would be more expensive per millisecond than disk space, just as downtown property rents at a higher rate than rural land. Likewise, in large, distributed systems, the various firms, individuals, research centers and so on would bid for the computational goods they need. Presumably this bidding and asking would be carried out by the computers themselves, according to pre-programmed instructions. We might find certain computational resources in extremely high demand, or in volatile markets, changing their prices several times a day--or several times a second.

Imagining Computation Markets of the Future

Miller and Drexler envision the evolution of what they call agoric open systems--extensive networks of computer resources interacting according to market signals. Within vast computational networks, the complexity of resource allocation problems would grow without limit. Not only would a price system be indispensible to the efficient allocation of resources within such networks, but it would also facilitate the discovery of new knowledge and the development of new resources. Such open systems, free of the encumbrances of central planners, would most likely evolve swiftly and in unexpected ways. Given secure property rights and price information to indicate profit opportunities, entrepreneurs could be expected to develop and market new software and information services quite rapidly.

Secure property rights are essential. Owners of computational resources, such as agents containing algorithms, need to be able to sell the services of their agents without having the algorithm itself be copyable. The challenge here is to develop secure operating systems. Suppose, for example, that a researcher at George Mason University wanted to purchase the use of a proprietary data set from Alpha Data Corporation and massage that data with proprietary algorithms marketed by Beta Statistical Services, on a superfast computer owned by Gamma Processing Services. The operating system needs to assure that Alpha cannot steal Beta's algorithms, that Beta cannot steal Alpha's data set, and that neither Gamma or the George Mason researcher can steal either. These firms would thus under-produce their services if they feared that their products could be easily copied by any who used them.

In their articles, Miller and Drexler propose a number of ways in which this problem might be overcome. In independent work, part of the problem apparently has already been overcome. Norm Hardy, senior scientist of Key Logic Corporation, whom we met at Xanadu, has developed an operating system caned KeyKOS which accomplishes what many suspected to be impossible: it assures by some technical means (itself an important patented invention) the integrity of computational resources in an open, interconnected system. To return to the above example, the system in effect would create a virtual black box in Gamma's computer, in which Alpha's data and Beta's algorithms are combined. The box is inaccessible to anyone, and it self-destructs once the desired results have been forwarded to the George Mason researcher.

In the sort of agoric open systems envisioned by Miller and Drexler, there would be a vigorous market for computational resources, which could be sold on a per-use basis, given a secure operating system. Royalties would be paid to the owners of given objects, which might be used in a variety of applications.

Programmers would be able to develop software by adding their own algorithms onto existing algorithms. They would not need to understand all the workings of what they use, only the results. Among other advantages, this would save the tremendous amount of time now used by programmers in the trivial redevelopment of capabilities that have already been well worked out. Most important, however, is the increased rapidity with which new products could be developed.

4. Mind as a Spontaneous Order: What is (Artificial) Intelligence?

If multilayered networks succeed in fulfilling their promise, researchers will have to give up the conviction of Descartes, Husserl, and early Wittgenstein that the only way to produce intelligent behavior is to mirror the world with a formal theory in the mind. Worse, one may have to give up the more basic intuition at the source of philosophy that there must be a theory of every aspect of reality--that is, there must be elements and principles in terms of which one can account for the intellegibility of any domain. Neural networks may show that Heidegger, later Wittgenstein, and Rosenblatt were right in thinking that we behave intelligently in the world without having a theory of that world.

--Dreyfus and Dreyfus (1989, p. 35)

The field of computer science that market process economists would be apt to find the most fascinating is Artificial Intelligence (AI). The traditional approaches to AI, still dominant in the specialization area known as "Expert Systems," takes intelligence to be an algorithmic, mechanical process. Although there are many commercially successful applications of these traditional AI systems, they have been extremely disappointing in terms of their ability to exhibit anything that deserves the name "intelligence." Indeed, precisely the aspects of intelligence that market process economists consider the most important, such as learning, creativity, and imagination, have proven to be the most difficult to produce artificially.

Over the past decade, however, a revolution has been occurring in AI researchers' thinking about thinking. The newer approaches, sometimes called Emergent AI, conceive of mental processes as complex, spontaneous ordering processes. Emergent AI traces its origins to early contributions to neural networks such as those of Donald 0. Hebb (1949), whom Hayek cites favorably in The Sensory Order, and Frank Rosenblatt (1958; 1962). These efforts had at one time been discredited by the more rationalistic approaches, but they are today making a dramatic comeback. As Sherry Turkle (1989, pp. 247-8) put it in an article contrasting "the two AIs":

Emergent Al has not been inspired by the orderly terrain of logic. The ideas about machine intelligence that it puts forward are not so much about teaching the computer as about allowing the machine to learn. This AI does not suggest that the computer be given rules to follow but tries to set up a system of independent elements within a computer from whose interactions intelligence is expected to emerge.

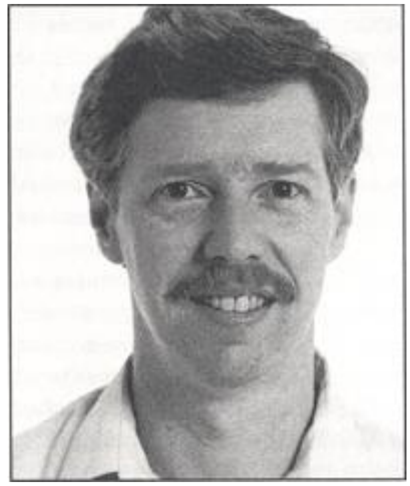
The critiques in the new AI literature of the failings of the rationalist approach to AI sound remarkably similar to Hayekian economists' critiques of the rational choice model. Even the very same philosophical traditions market process economists have used--post-Kuhnian philosophy of science, the later Wittgenstein, and contemporary phenomenology and hermeneutics--have been used by the newer AI researchers as the basis of their critique of Cartesian rationalism.

A great deal of work has been done in the new Emergent AI literature that simulates complex ordering processes. For example, there are the "genetic algorithms" and classifier systems approaches developed by John Holland, and the connectionist or neural networks approaches. A significant faction of the AI community thus finds itself arriving at essentially the same conclusions about the nature of the human mind as Austrian economists. The process of mind is a decentralized, competitive process. No CPU exists in the brain. Marvin Minsky's conception of the way the mind works in The Society of Mind is remarkably similar to Hayek's in The Sensory Order. Hayek draws significant methodological conclusions from his view of the nature of the mind, for example in his essays "Rules, Perception and Intelligibility" and "The Primacy of the Abstract."

Some of the interesting work that has been done within Al directly introduces market principles into the design of models of learning. Some initial research along these lines has been done by Ted Kaehler and Hadon Nash at Apple, in a system they named Derby. This approach tries to introduce market bidding processes and monetary cost calculations into a neural network model in order to generate artificial learning. In his work on classifier systems, John Holland makes explicit use of market processes for reinforcing a program's successful rules of action. "We use competition as the vehicle for credit assignment," he says. "To do this we treat each rule as a 'middleman' in a complex economy" (1986, p. 380). He speaks of suppliers, consumers, capital, bids, and payments.

This type of research involves using economic principles to advance artificial intelligence research, but our interest in it is not primarily altruistic; the point is not so much to help Al achieve its aspirations, but to see what it can teach us economists about the nature of human intelligence.

It is too early to tell whether these approaches to Al will be more successful in replicating learning, creativity, inductive thinking, and so forth than the traditional approaches, but it is already clear that the Emergent Al approach is able to do some things the older approaches couldn't. Even if one believes many current approaches to Al are utterly unworkable, their very failures might be worthy of closer study. The fact that mainstream Al research has not yet been able to reproduce certain key aspects of human intelligence may be highly significant. Books critical of mainstream Al, such as Hubert Dreyfus's *What Computers Can't Do,* and Terry Winograd and Fernando Flores's *Understanding Computers and Cognition*, present a powerful critique of the rational choice model, one from which we could borrow in our own efforts to challenge mainstream economics.



Understanding how the human mind works is not only of interest in that the mind is conceived as a specific instance of spontaneous order processes, and as such may display some interesting analogies to market orders. There is also the point that, for a subjectivist approach to economics, what the market order is, is an interplay of purposeful human minds. We need to know as much as we can about how human beings think and communicate for our substantive economics. Al, in its failures and its achievements, is wonderfully informative on these issues.

Even the skeptic about Al research and computer modeling in general could see that these simulation methods raise some provocative research questions for market process economics. To what extent have genuine learning and creativity been simulated in this research? How much does it matter that natural language simulation has not yet gotten very far? Just how different are these different levels of spontaneous order, as we advance from biological to animal and human cognitive processes, and to market processes, and what properties do they share?

5. Computation as a Discovery Procedure: Possibilities for Agoric Mental Experiments

Underlying our approach to this subject is our conviction that "computer science" is not a science and that its significance has little to do with computers. The computer revolution is a revolution in the way we think and in the way we express what we think. The essence of this change is the emergence of what might best be called procedural epistemology--the study of the structure of knowledge from an imperative point of view, as opposed to the more declarative point of view taken by classical mathematical subjects. Mathematics provides a framework for dealing precisely with *notions of "what is." Computation provides a framework for dealing precisely with notions of "how to."*

--Abelson and Sussman (1985, p. xvi)

Perhaps the research direction we talked about which will prove to be most controversial involves not so much using economics to study computers as the other way around, the direct use of computer modeling techniques to develop economic theory. This would be a matter of expanding on ideas sketched in another of Miller and Drexler's papers, "Comparative Ecology: A Computational Perspective." Up to now we have been talking about applying more or less standard market process economics to some computer-oriented topics. Those first topics could be done in words, as it were, but this one could also involve actually doing some computer programming of our own.[2] The idea here is that we could try to improve our understanding of the market order by developing spontaneous order simulations on a computer. We might be able at least to illuminate existing ideas in market process economics, and we might conceivably develop substantive new ones, by doing mental experiments within artificial minds.

How, it might be asked, could a school which stresses the non-mechanical nature of human action find itself trying to simulate action on electronic machines? After all, market process economists think an alternative approach to the subject matter is necessary precisely because neoclassical economics has tried to subsume action into a mechanistic model.

But what we mean by "mechanical" has been shaped by the age when machines were extremely crude and rigid things. As computers advance, they increasingly challenge our ideas about what "machines" are capable of. In principle, there is no reason why computers themselves could not become sufficiently "non-mechanistic" to be of interest to market process economics. As the previous section pointed out, research in Artificial Intelligence aspires to reproducing on electronic machines exactly the sorts of phenomena, such as creativity and learning from experience, in which market process economics is interested.

The market process approach has never been against models as such, but has only objected to modeling that loses sight of certain non-mechanical aspects of human choice. If the aspects of human action that the school considers the most important cannot be handled with the modeling tools of the mainstream, it may be necessary to devise better tools. We have to admit at this point that we are not at all sure ourselves whether market process economics can usefully deploy computer simulation methods. But the only way to tell if computer simulations can serve as such tools would be to try to build some, and see what can be done with them.

Market process oriented economists have often pointed out that the mathematics of differential calculus that has played such a central role in mainstream economics is not the appropriate mathematics for studying the economy. James Buchanan, for example, has suggested that game theory, since it can deal with the interplay of strategic choices through time, would constitute a more appropriate mathematics. Kenneth Boulding has pointed to topology as an alternative mathematics, because it can deal with shapes rather than quantities. Without disagreeing with the reasons game theory and topology might be useful formalizations, we suspect that the appropriate formalization for economics might not be a mathematics at all. Computer programming may constitute the kind of formalization most conducive to the development of market process theory.[2] It is a formal medium for articulating the "how to" of dynamic processes, rather the "what is" of timeless end-states with which mathematics is concerned. Mathematical modeling has often distracted economics from paying attention to the processes that market process economics emphasizes. Computer "simulations" of spontaneous order processes might prove to be the kind of modeling approach that is process-oriented enough to help rather than obstruct economic theorizing. Thus it could constitute a useful new complement to the traditional procedures of theorizing that market process economists now employ.

It is important to be clear about just what is meant by "simulation" here. It certainly has nothing to do with efforts to build direct simulations of specific real world markets, or of the overall economy. The "worlds" in the computer would be radically simplified, and the "agents" would be only "artificially" intelligent, which is to say, at this point in AI research, they would be rather stupid. But these agents may still be more like humans than the optimizing agent of mainstream neoclassical theorizing: they may be equipped with the capacity to learn from experience. But there would be no pretensions of capturing the complexities of reality within a model, or of being able to derive predictions about reality directly from the simulation exercises. The notion here is rather of using computers as an aid for conducting complicated mental experiments. It would not be empirical but theoretical research.

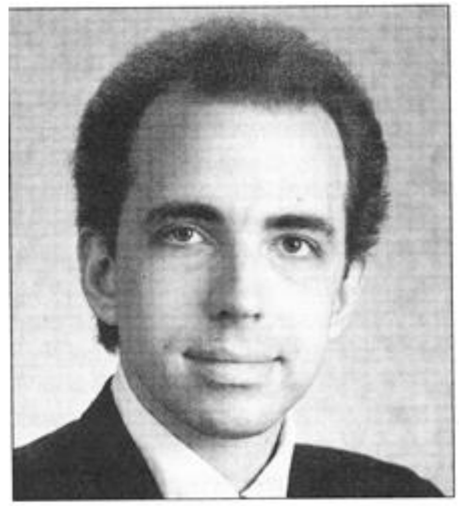
On the other hand, it would be more experimental, in a sense, than most theoretical research is today. This kind of simulation would differ from most contemporary theorizing, in that the purpose of the modeling exercise would not be to devise a whole deterministic mechanism, such as is the goal of IS/LM and Rational Expectations models in macroeconomics, or general equilibrium theory in microeconomics. Rather, the aim would be to set up constraining conditions, specifying institutional environments or decision rules for agents, and then to run the simulation in order to see what happens. The idea is not to create a mathematical model that already implies its conclusions in its premises. Rather, it is to run the simulations as mental experiments, where what is of interest is not what the end results are so much as how the process works. And we, the programmers, would not know how the process was going to come out until we ran the mental experiments. The order would emerge not by the programmer's design, but by the spontaneous interplay of its component parts.

One of the first projects that needs to be undertaken is to see just how much there is in existing computer modeling, inside or outside of economics, which we might critically examine as illuminating the properties of spontaneous order processes. The various evolutionary process modeling strategies mentioned in the previous section, that are being used in Artificial Intelligence, in theoretical evolutionary biology, and in theoretical ecology, could be reinterpreted as referring to economic institutions or economies instead of brains or species. Existing program designs could be examined from a Hayekian perspective and modified in order to illuminate selected properties of spontaneous order processes. Or completely new programs could be developed with markets more directly in mind.

Imagine, for example, trying to contrive Mengerian simulations in which a medium of exchange spontaneously evolves, or free banking simulations that evolve clearinghouses and stable monies. In industrial organization, it might be possible to investigate how firm sizes vary with different industry characteristics, and how industries evolve as markets and technology change. Business cycle effects might be studied: could we, for example, observe changes in a simulated capital structure as a result of an injection of credit? We might probe constitutional economics by running a series of parallel simulations differing only in certain fundamental rules such as property rights and contract sanctity. What different emerging properties of the economic order would we observe?

Of course we need to be careful about the unrealistic nature of these mental experiments, and not let the modeling become an end in itself. The crippling vice of most economic theory today is its "model-fetishism." Economists get preoccupied with the game of modeling itself, and forget the need to interpret the mental experiment. Although most game theory in economics suffers as much from formalism as general equilibrium theory and macroeconomics do, the iterative game theory work of Robert Axelrod is very much the kind of research we are thinking of here. There, the computer tournament was couched in a substantive interpretive effort. The mental experiment was not just a game played for its own sake, but a heuristic vehicle, a mental experiment to help us to think about the evolution of human cooperation.

Other than Axelrod's work, probably the closest thing in the existing economics literature to this kind of simulation would be what is called experimental economics. Whereas our simulations would not use human subjects, as does most economic experimentation, the experimental economists' focus on the design and functioning of market institutions is very much in the spirit of what we have in mind. Moreover, the use of non-human agents would in many instances allow for greater flexibility in the design of spontaneous order simulations. lnstead of using rats, we could use artificial agents. As Ted Kaehler pointed out, this is a step down in many dimensions of intelligence, but there are other advantages of computer experiments which suggest that some interesting theory development might be possible along these lines. The more Hayekian contributors to the experimental literature, such as Vernon Smith and Ron Heiner, will undoubtedly have many useful ideas for us along these lines, but otherwise we believe this kind of theoretical spontaneous order simulation is unexplored territory for market process economics.



Miller, Drexler, and Salin deserve our thanks for introducing Hayekian ideas to the computer science community, and we certainly encourage computer scientists to follow up directly on their work. Conceivably, we economists might be able to introduce theoretical principles to computer scientists that could help them address their problems. But a more likely benefit of our taking up these questions is that, by applying our existing economics to the study of computational processes, we might help improve our economics, and that may help us think more clearly about human economies, which is, after all, what we are really interested in.

End Notes

[1]Plans for Xanadu also include provisions for reader feedback as to the value of different articles or notes, and for royalty payments per use to all authors, even of short comments.

[2]This is not necessarily as radical a change from writing "in words" as we are implying here. Computer programs are written in order to be read by people, and not merely to be run on machines. As Abelson and Sussman (1985, p. xv) put it in their classic textbook on programing, a computer program is not just a way of getting a computer to do something, it is "a novel formal medium for expressing ideas about methodology. Thus, programs must be written for people to read, and only incidentally for machines to execute."

Appendix: Annotated Bibliography

The interface of computer science and market process economics provides a wealth of exciting research opportunities. By studying both the application of market process ideas to computer science and the ways in which developments in computer science enrich our understanding of market processes, we can expand the interplay between theoretical insights and empirical praxis. In this bibliographical note, we hope to alert interested readers to some of these sources. While this list is far from exhaustive, we hope that it will provide a useful starting point for those wishing to pursue these research opportunities.

Process-Oriented Case-Studies of the Computer Industry

The computer industry, with its fast growth and rapid technological change, provides fertile soil for studying such traditional market process concerns as the role of entrepreneurship (Kirzner, 1973), the production and use of knowledge and information (Hayek, 1948), and the peculiarities of software as capital (Lachmann,1978). The computer industry, however, is not only an increasingly important sector in its own right, but also is one that is ushering in new forms of market interaction in a broad variety of industries. The emergence of electronic markets (Malone, et at., 1989) in such areas as computerized airline registration systems, program trading on increasingly interlinked electronic exchanges, as well as computer-aided buying and selling through home-shopping systems, has become an important research and policy topic.

More broadly, the rapid evolution of information and communication technologies and the peculiarities of information as a good underscores the important relationship between the legal environment and technical change (Liggio and Palmer, 1988). The interplay of technological, contractual, common-law, and legislative solutions to such problems as the assignment of property rights to intangible goods, for example the broadcast spectrum and intellectual property (Palmer, 1988), is an exciting and relatively unexplored research opportunity.

In addition, the blurring of the distinctions between the various communications media (print, broadcasting and common carriers), as evidenced by the recent breakup of AT&T, highlights the relationship between innovation, public policy, and market competition (Pool, 1982, Huber, 1987, Mink and Ellig, 1989).

A further example of the important dynamic interactions between technical and policy responses to change in the computer industry can be found in the emergence of industry standards, whether as the outcome of the market process, or imposed by legislation, or through agreement forged in standard committees (Katz and Shapiro, 1985, David, 1987). The problems of compatibility and interoperability between various software and hardware components highlights the role of market dialogue in the shaping of expectations and the formation of consensus in reaching an industry standard. In addition, the process of standard formation is constantly being threatened by the entrepreneurial discovery process (Hayek, 1978a) leading to the search for technical adapters and innovations which would make such standards obsolete. Moreover, this process may exhibit a high degree of historical path dependency, as the dominance of the technologically inferior QWERTY keyboard demonstrates (David, 1986, Could, 1987).

An additional area of interest to market process economists is the role that computers have played in the socialist calculation debate. For many theorists, the computer was seen as the answer to the challenge of Mises and Hayek to the workability of central planning (Lange, 1967). In light of recent trends in computer science towards decentralized means of coordination, the bold claims of future computopians takes on an ironic twist (Lavoie, 1990). A recent attempt to implement a version of a computopian plan in Chile (Beer, 1975) could be usefully examined by market process economists.

Information Technology and the Evolution of Knowledge and Discourse

While it is commonplace to hear how computers will transform our society, the reality seems to move much slower than the promised dramatic effects. Nevertheless, as computers become more widespread and interconnected, the use of computer-aided dialogue and communication could have important repercussions on the evolution of knowledge and scientific discourse (Bush, 1945). These new knowledge media (Stefik, 1988) could have an effect not seen since Gutenberg.

As Hypertext systems emerge (Nelson, 1973, Drexler, 1986, 1987), they could enhance the evolution of scientific knowledge through more rapid dissemination of knowledge, and more efficient means of criticism and debate. These developments could have important implications for the spread of economic ideas, and the pattern of discourse within economics (Tullock, 1965; McCloskey, 1985; Colander and Coats, 1989).

Complexity, Coordination and the Evolution of Programming Practices

As software applications become more advanced, and computer systems more powerful, a central problem that emerges is how to cope with the rapidly expanding complexity of software systems. In computer systems the knowledge embodied in the software is more important than the physical hardware upon which it runs. As Alan Kay expresses it, "architecture dominates material" (Kay, 1984). Computer programming is properly seen as a medium of expression of ideas, a product of the human mind (Ableson and Sussman, 1985). The techniques used to grapple with the intellectual complexity of large software systems (Brooks, 1975) could be usefully studied in comparison to techniques in economics. While the complexity of an economic system is obviously much greater than even the most complex software system (Hayek, 1967b), the methods used to maintain intellectual coherence, and to improve the use of knowledge may have analogs in each system. Such issues and techniques as modularity, specialization, embodied knowledge, and use of local knowledge have their counterparts in each system.

Miller and Drexler have usefully applied the analogy between economic institutions and programming practices through their insight that the latest development in programming languages (Object Oriented Programming) can be viewed as being the reinvention of the concept of property rights in the domain of computer science (Miller and Drexler, 1988b). Insights from the economics of property rights could perhaps be usefully applied to the development of software systems (Demsetz, 1967, Anderson and Hill, 1975, Hayek, 1989).

Object oriented programming techniques such as encapsulation that allows for the separation of internal state from external behavior, as well as the coupling of data and procedures, promise to expand the range of a programmer's conceptual control. Object oriented programs perform computations by passing messages between various objects, which can be viewed in terms of their real-world analog (Thomas, 1989; Cox, 1986; Shriver and Wegner, 1988).

While current software systems can be very complex, their complexity is sure to increase as computation becomes more distributed across networks of heterogeneous computers with different users pursuing their own particular goals. As Miller and Drexler point out, central planning techniques are no longer adequate for coordination of these open systems, and a more decentralized coordinating mechanism is needed (Hewett, 1985; Bond and Glassner, 1988; Kahn and Miller, 1988; Huberman, 1988). An important aspect of these open, decentralized systems will be the need to maintain the security of the proprietary data and software of the different agents (Miller, et al., 1987; Hardy, 1988). An additional aspect of the combination of large distributed systems and object oriented programming is the promise it holds for the more rapid evolution of software applications. (Miller and Drexler, 1988b; Drexler, 1988, 1989). Agoric open system can take advantage of object oriented programming's ability to provide opportunities for easy reuse and recombination of components, and incremental improvements.

Mind as a Spontaneous Order: What is (Artificial) Intelligence?

A further area of interest for market process economists refers to the insights regarding the nature of rationality that have been achieved through the attempt to create artificially intelligent computers. While these attempts have yielded many interesting applications, they have had little success in creating anything resembling intelligence. However, much has been learned by the attempt. Economists have much to gain from the successes and failures of artificial intelligence. The work of Herbert Simon, of course, has been influential in both economics and computer science (Newell and Simon, 1972; Simon, 1983, 1985), and has been instrumental in bringing to the attention of economists the nature of the computational demands placed upon their perfectly optimizing agents. The inability of humans to fulfill the demands of the optimizing agents has become increasingly recognized (Kahneman, et al., 1982) as well as the implications that these less than perfect agents have for economic theory (Heiner, 1983).

The limitations of attempting to design intelligence as a mechanistic, decision-making process has led to a shift towards a view of intelligence as being an emergent property of a complex learning process (Graubard, 1989; Drexler, 1989). The mind is seen as a spontaneous order process in which the resulting intelligence is greater than is possible by design (Hayek, 1952). The mind is viewed as being subject to competitive and cooperative pressures like other complex, evolving systems. A variety of metaphors have been explored in attempting to create an emergent approach to artificial intelligence. Perhaps the best known are the connectionist or neural network approaches, which attempt to mimic the neural process of the brain itself. A wide variety of connectionist approaches are currently being attempted (Hillis, 1985; McClelland and Rumelhart, 1986; Edelman, 1987; Cowan and Sharp, 1989; Schwartz, 1989; Reeke and Edelman, 1989), including an attempt that applies some of the agoric insights to the problem of attributing success to various "neurons" (Kaehler, et al., 1988).

In addition, to the neural metaphor, computer scientists have attempted to apply social metaphors, recognizing the greater social intelligence (Lavoie, 1985) that emerges out of the interaction of less intelligent parts (Kornfield and Hewett, 1981; Minsky, 1986; Campbell, 1989).

Genetic and evolutionary analogies from biology have also been influential (Langton,1987). These approaches include the Eurisko project of Doug Lenat (Lenat, 1983; Lenat and Brown, 1988), and the genetic algorithm approach pioneered by John Holland (Holland, 1975; De Jong, 1988; Goldberg, 1989; Booker et al. , 1989). In addition the classifier system, also pioneered by John Holland, has attempted to build a parallel rule based learning system that combines the rule discovery properties of genetic algorithms, and an economic model for the problem of credit assignment (Holland, 1986; Holland et al. , 1986; Booker, et al., 1989).

These developments in computer science may improve our understanding of both a wide variety of spontaneous order processes, as well as the nature of intelligence. The failure of traditional Cartesian rationalist approaches to artificial intelligence has prompted a critique similar to much of the market process economists' critique of neoclassical rationality (Dreyfus, 1972; Dreyfus and Dreyfus, 1985, 1986, 1989; Winograd and Flores, 1986). These critiques have emphasized the important role that language and social interaction play in intelligence, as well as the limitations of the knowledge as representation approach (Dascal, 1989), in capturing the tacit and context-dependent nature of knowledge (Polanyi, 1962).

Computation as a Discovery Procedure: Possibilities of Agoric Mental Experiments

The advances in artificial intelligence and computer programming suggest that these techniques could be usefully applied to experimental modeling the complex processes of interaction that occur in economic systems. The goal of this type of modeling is not a predictive model that tries to simulate reality, but rather mental experiments to help us better understand spontaneous order processes.

One of the closest examples to the approach being suggested here is the work of Robert Axelrod on the evolution of cooperation. Axelrod's mixture of theoretical insights, a computer tournament, and empirical case studies has proved to be both influential and illuminating (Axelrod, 1984). His work has inspired a wide range of further empirical work and theoretical insights (Axelrod and Dion, 1988), including the use of genetic algorithms to generate strategies that in certain situations improved on the performance of "tit for tat" (Axelrod, 1987 , J. Miller, 1989).

Another area that highlights the importance of the dialogic interplay between theory and empirical observation is the fast-growing field of experimental economics (Plott, 1982, Smith, 1982). While the majority of experiments to date have focused primarily on the relatively straightforward auction-type institutions, the experimental results have demonstrated the importance played by the exchange institutions--the system of property rights in communication and exchange. As Vernon Smith notes, "it is not possible to design a resource allocation experiment without designing an institution in all of its detail" (1982, p. 923). This detailed focus on the institutional constraints is perhaps the most valuable aspect of the experimental approach. The focus to date in this young field has been on relatively simple market institutions, and on static outcomes, not on the dynamic adjustment and learning processes (Heiner, 1985). The complexity of keeping track of these adjustment processes suggest a fruitful role for computers. Computer-aided experimental markets, such as the computerized double auction mechanism (PLATO), have already helped to illuminate these dynamic processes (Smith, et at. , 1988). Furthermore, a group at the Sante Fe Institute has already combined Axelrod's computer tournament model with the experimental focus on market institutions, by running a computerized double auction tournament (Rust, et at., 1989).

Motivating much of this interest in computer modeling of spontaneous order processes is a dissatisfaction with traditional equilibrium approaches to capturing the evolutionary and self-ordering aspects of the market process. The development of order analysis, as an alternative to equilibrium-bound theorizing can be enhanced by our better understanding the working of spontaneous order processes (Hayek, 1973, 1989; Buchanan, 1982; Boettke, et at., 1986; Horwitz, 1989), and the nature of the rule systems and institutional order that help guide the ordering processes (Brennan and Buchanan, 1985; Buchanan, 1986; Langlois, 1986). This increasing interest in evolutionary approaches to economics (Nelson and Winter, 1982; Anderson, et at., 1988; Day, 1987; Allen, 1988; Silverberg, 1988; Holland, 1988) has been fueled in part by the developments in the new views of science (Lavoie, 1989; Prigogine and Stengers, 1984).

Further work in developing "agoric mental experiments" can begin by examining the current work in computer science that uses market principles. Bernardo Huberman and his coworkers at Xerox PARC, building on the work of Drexler and Miller (1988), have developed a computerized market allocation mechanism called Spawn (Waldspurger, et at., 1989), and have explored the dynamic properties of distributed computer systems (Huberman, 1988, 1989a,b; Huberman and Hogg, 1988; Huberman and Lumer, 1989; Kephart, et at., 1989a,b; Cecatto and Huberman, 1989). Market-based models for computation have also been explored by Tom Malone at MIT (Malone, 1988; Malone, et at., 1988), and Ferguson (1989). Nascent attempts to apply these computer techniques to economics have been attempted by members of the Sante Fe Institute (Marimon, et at., 1989), and by the Agoric research group at George Mason University (De Garis and Lavoie, 1989).

Bibliography

Abelson, Harold and Gerald J. Sussman. 1985. Structure and Interpretation of Computer Programs (Cambridge, MA: The MlT Press).

Allen, Peter. 1988. "Evolution, Innovation, and Economics," in G. Dosi et al., eds., Economic Theory and Technical Change (New York: Printers Publishers).

Anderson, Philip W ., Kenneth J. Arrow and David Pines, eds. 1988. The Economy as an Evolving Complex System (Redwood City, CA: Addison-Wesley).

Anderson, Terry L. and P. J. Hill. 1975. "The Evolution of Property Rights: A Study of the American West," Journal of Law and Economics, April.

Axelrod, Robert. 1984. The Evolution of Cooperation (New York: Basic Books).

.1987. "The Evolution of Strategies in the Iterated Prisoner's Dilemma," in

L. D. Davis ed., Genetic Algorithms and Simulated Annealing (Los Altos, CA: Morgan Kaufmann).

Axelrod,Robert and D. Dion. 1988. "The Further Evolution ofCooperation," Science, 242, December 9.

Beer, Stafford. 1975. Platform for Change (New York: Wile).

Boettke, Peter, Steven Horwitz and David L. Prychitko. 1986. "Beyond Equilibrium Economics: Reflections on the Uniqueness of the Austrian Tradition," Market Process, vol. 4, no.2, Fall.

Bond, Alan, H. and Les Glasner, eds. 1988. Readings in DistributedArtificial Intelligence (San Mateo, CA: Morgan Kaufmann).

Booker, L. B., D. E. Goldberg and J. H. Holland. 1989. "Classifier Systems and Genetic Algorithms," Artificial Intelligence, 40.

Brennan, G. and J. M. Buchanan. 1985. The Reason of Rules: Constitutional Political Economy (Cambridge: Cambridge University Press).

Brooks, Frederick P., Jr. 1975. The Mythical Man-Month (Reading, MA: Addison-Wesley)-

Buchanan, J ames M. 1982. "Order Defined in the Process of Its Emergence," Literature ofLiberty, vol. 5, no.4, winter, reprinted inJ.M. Buchanan,Liberty, Market and State (New York: New York University Press, 1986).

.1986. Liberty, Market and State: Political Economy in the 80's(New York: New York University Press).

Bush, Vannever. 1945. "As We May Think," Atlantic Monthly, July.

Campbell, Jeremy. 1989. The Improbable Machine: What the Upheavals in Artificial Intelligence Research Reveal about How the Mind Really Works (New York: Simon and Schuster).

Ceccatto, H. A. and B. A. Huberman. 1989. "Persistence of Nonoptimal Strategies," Proceedings of the National Academy of Science, vol. 86, May.

Colander, David and A. W. Coats, eds. 1989. The Spread of Economic Ideas (Cambridge: Cambridge University Press).

Cowan, Jack D. and David H. Sharp. 1989. "Neural Nets and Artificial Intelligence" in S. R. Graubard, ed., The Artificial Intelligence Debate: False Starts, Real Foundations (Cambridge, MA: The MIT Press).

Cox, Brad J. 1986. Object Oriented Programming: An Evolutionary Approach (Reading, MA: Addison-Wesley).

Dascal, Marcelo 1989. "Artificial Intelligence and Philosophy: The Knowledge of Representation," Systems Research, vol. 6, no.1.

David, Paul. 1986. "Understanding the Economics of QWERTY: the Necessity of History," in W. Parker, ed., Economic History and the Modern Economist (Oxford: Basil Blackwell).

.1987. "Some New Standards for the Economic of Standardization in the Information Age," in P. Dasgupta and Stoneman, eds., Economic Policy and Technological Performance (Cambridge: Cambridge University Press).

Day, Richard. 1987. "The Evolving Economy ," European Journal of Operations Research, 30, June.

De Garis, Hugo and Don Lavoie. 1989. "Evolution of the Money Concept: A computational Model," unpublished Manuscript, George Mason University.

De Jong, Ken. 1988. "1AJarning with Genetic Algorithms," Machine Learning, 3.
Demsetz, Harold. 1967. "Towards a Theory of Property Rights," American Economic

Review, May.

Drexler, K. Eric. 1986. Engines of Creation (New York: Doubleday).

.1987. "Hypertext Publishing and the Evolution of Knowledge," Foresight Institute, Palo Alto, CA.

.1988. "Biological and Nanomechanical Systems: Contrasts in Evolutionary Capacity," in C. Langton, ed., Artificial Life (Redwood City, CA: Addison-Wesley).

.1989. "AI Directions," Foresight Update, 5, March-

, and M. S. Miller. 1988. "Incentive Engineering for Computational Resource Management" in B. A. Huberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

Dreyfus, Hubert L. 1972. What Computers Can't Do: A Critique of Artificial Reason (New York: Harper and Row).

, and Stuart E. Dreyfus. 1985. Mind over Machine (New York: The Free Press). , and Stuart E. Dreyfus. 1986. "Why Expert Systems Do Not Exhibit Expertise," Expert.

, and Stuart E. Dreyfus. 1989. "Making a Mind versus Modelling the Brain:Artificial Intelligence Back at a Branchpoint," in S. Graubard, ed., The Artificial Intelligence Debate: False Starts, Real Foundations (Cambridge, MA: MIT Press).

Edelman, G. M. 1987. Neural Darwinism: The Theory of Neuronal Group Selection (New York: Basic Books).

Ferguson, Donald D. 1989. "The Application ofMicroeconomics to the Design of Resource Allocation and Control AJgorithms," Ph.D. Thesis, Columbia University.

Goldberg, David E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning (Reading, MA: Addison-Wesley).

Gould, StephenJ. 1987. "The Panda's Thumb of Technology," Natural History, January. Graubard, Stephen R., ed. 1989. The Artificial Intelligence Debate: False Starts, Real

Foundations (Cambridge, MA: MIT Press).

Hardy, Norman. 1988. "Key KOS Architecture," manuscript, Santa Clara, CA: Key Logic.

Hayek, F. A. 1948. "The Use of Knowledge in Society," in Individualism and Economic Order (Chicago: University of Chicago Press).

.1952. The Sensory Order: An Inquiry into the Foundations of Theoretical Psychology (Chicago: University of Chicago Press).

.1967a. "Rules, Perception and Intelligibility," in Studies in Politics, Philosophy and Economics (Chicago: University of Chicago Press).

.1967b. "The Theory of Complex Phenomena," in Studies in Politics, Philosophy and Economics (Chicago: University of Chicago Press).

.1973. "Cosmos and Taxis," Chapter 2 in Law, Legislation and Liberty: Rules and Order, val. 1, (Chicago: University of Chicago Press).

.1978a. "Competition as a Discovery Procedure," in New Studies in Politics, Philosophy, Economics and the History of Ideas (Chicago: University ofChicago Press).

.1978b. "The Primacy of the Abstract," in New Studies in Politics, Philosophy, Economics and the History of Ideas (Chicago: University of Chicago Press). .1989. The Fatal Conceit: The Errors of Socialism (Chicago: Universityof Chicago Press).

Hebb, Donald 0. 1949. The Organization of Behavior (New York: John Wiley). Heiner, Ronald A. 1983. "The Origin of Predictable Behavior," American Economic

Review, September.

.1985. "Experimental Economics: Comment," American Economic Review, March.

Hewitt, Carl. 1985. "The Challenge of Open Systems," Byte, April

Hillis, Daniel W. 1985. The Connection Machine (Cambridge, MA; MIT Press).

Holland, John. 1975. Adaptation in Natural and Artificial Systems (Ann Arbor: University of Michigan Press).

.1986. "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-BasedSystems," in R. S. Michalski et al. eds., Machine Learning II (ws Altos, CA: Morgan-Kaufmann).

.1988. "The Global Economy as an Adaptive Process," in P. Anderson, K. Arrow, P. Pines, eds., The Economy as an Evolving Complex System (New York: Addison-Wesley).

Holland, John,K.J.Holyoak,R.E. Nisbet, andP.R. Thagard. 1986. Induction: Processes of Inference, Learning, and Discovery (Cambridge, MA: MIT Press).

Horwitz, Steve. 1989. "The Private Basis ofMonetary Order: An Evolutionary Approach to Money and the Market Process," Ph.D. dissertation, George Mason University.

Huber, Peter. 1987. "The Geodesic Network: 1987 Report of Competition in the Telephone Industry," (Washington, D.C.: Government Printing Office).

Huberman, B. A., ed. 1988. The Ecology of Computation (Amsterdam: North-Holland). .1989a. "An EcologyofMachines: How Chaos Arises in Computer Networks,"

The Sciences, July/August.

.1989b. "The Performance of Cooperative Processes," unpublished manuscript, Xerox PARC.

Huberman, B. A. and T. Hogg. 1988. "The Behavior of Computational Ecologies," in Ruberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

.and E. Lumer. 1989. "Dynamics of Adaptive Controls," unpublished manuscript, Xerox PARC.

Kaehler, Ted, Radon N ash and Mark S. Miller. 1988. "Betting, Bribery, and Bankruptcy: A Simulated Economy Learns to Predict," unpublished manuscript.

Kahn, Ken and Mark S. Miller. 1988. "Language Design and Open Systems" in B. Ruberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

Kahneman, D., P. Slovic and A. Tversky. 1982. Judgement Under Uncertainty: Heuristics and Biases (New York: Cambridge University Press).

Katz, Michael L. and Carl Shapiro. 1985. "Network Externalities, Competition, Compatibility,"American Economic Review, May.

Kay, Alan. 1984. "Computer Software," Scientific American, September.

Kephart, J. 0., T. Hogg and B. A. Huberman. 1989a. "Dynamics of Computational Ecosystems: Implications for Distributed Artificial Intelligence," unpublished manuscript, Xerox PARC.

Kephart, Jeffrey 0. T. Hogg and Bernardo Huberman. 1989b. "Collective Behavior of Predictive Agents," manuscript, Xerox, PARC.

Kirzner, Israel M. 1973. Competition and Entrepreneurship (Chicago: University of Chicago Press).

Kornfeld, William A. and Carl Rewitt. 1981. "The Scientific Community Metaphor," in Transactions on Systems, Man, and Cybernetics.

Lachmann, Ludwig M. 1978. Capital and Its Structure (Kansas City: Sheed Andrews and McMeel).

Lange, Oskar. 1967. "The CfJmputer and the Market" in C. H. Feinstein, ed., Socialism, Capitalism and Economic Growth: Essays Presented to Maurice Dobb (Cambridge: Cambridge University Press).

Langlois, Richard N. 1986. Economics as a Process: Essays in the New Institutional Economics (New York: Cambridge University Press).

Langton, Christopher G. 1987. Artificial Life: The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Redwood City, CA: Addison-Wesley).

Lavoie, Don. 1985. National Economic Planning: What is Left? (Cambridge, MA: Ballinger Publishing Co.)

.1989. "Economic Chaos or Spontaneous Order? Implications for Political Economy of the New View of Science," Cato Journal, vol. 8, no.3, Winter.

.1990. "Computation, Incentives, and Discovery: The Cognitive Function of Markets in Market Socialism," The Annals, January.

Lenat, D. B. 1983. "The Role ofHeuristics in Learning by Discovery: Three Case Studies" in R. S. Michalski, et al., eds., Machine Learning: an Artificial Intelligence Approach I. (Palo Alto, Ca: Tioga Publishing).

,andJ.S.Brown. 1988. "Why AM andEuriskoAppear to Work"inB.Huberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

Liggio, Leonard P. and Tom G. Palmer. 1988. "Freedom and the Law: A Comment on Professor Aranson's Article," Harvard Journal of Law and Public Policy.

Malone, Thomas W. 1988. "Modelling Coordination in Organizations and Markets" in A. Bond and L. Glassner, eds., Readings in Distributed Artificial Intelligence (San Mateo, CA: Morgan Kaufmann).

, R. E. Fikes, K. R. Grant, M. T. Howard. 1988. "Enterprise: A Market-like Task Scheduler for Distributed Computing Environments," in B. A. Huberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

, J. Yaws and R, Benjamin. 1989. "The Logic of Electronic Markets," Harvard Business Review, May-June.

Marimon, Ramon, Ellen McGrattan, and Thomas J. Sargent. 1989. "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents," unpublished manuscript, Sante Fe Institute.

McClelland, J. L. and D. E. Rumelhart, et al. 1986. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vols. 1 and 2 (Cambridge, MA: MIT Press).

McCloskey, Donald. 1985. The Rhetoric of Economics (Madison: UniversityofWisconsin Press).

Miller, John H. 1989. "Coevolution of Automata in the Repeated Prisoner's Dilemma," unpublished manuscript, Sante Fe Institute.

Miller, Mark S., Daniel G. Bobrow, Eric Dean Tribble, and Levy Jacobs. 1987. "Logical Secrets," in E. Shapiro, ed., Concurrent Prolog: Collected Papers (Cambridge, MA: MIT Press).

, and K. Eric Drexler. 1988a. "Comparative Ecology: A Computational Perspective," in B. A. Huberman, ed., The Ecology ofComputation (Amsterdam: North-Holland).

, and K. Eric Drexler. 1988b. "Markets and Computation: Agoric Open Systems," in B. A. Huberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

Mink, Phillip and Jerome Ellig. 1989. "The Courts, the Congress, and the Future of Telecommunications," (Washington, D.C.: Citizens for a Sound Economy Foundation).

Minsky, Marvin. 1986. The Society of Mind (New York: Simon and Schuster).

Nelson, Richard R. and Sidney G. Winter. 1982. An Evolutionary Theory of Economic Change (Cambridge, MA: Harvard University Press.

Nelson, Ted. 1973. Computer Lib; self-published.

Newell, A. and Herbert Simon. 1972. Human Problem Solving, (Englewood Cliffs, New Jersey: Prentice Hall).

Palmer, Tom. 1988. "Intellectual Property," Market Process, Fall.

Plott, Charles. 1982. "Industrial Organization Theory and Experimental Economics," Journal of Economic Literature.

Polanyi, Michael. 1962. Personal Knowledge: Towards a Post-Critical Philosophy ,(Chicago: Chicago University Press).

Pool, Ithiel de Sola. 1983. Technologies of Freedom, (Cambridge, MA: Harvard University Press).

Prigogine, llya and Isabelle Stengers. 1984. Order Out of Chaos: Man's New Dialogue with Nature (New York: Bantam Books).

Reeke, George N., Jr. and Gerald M. Edelman. 1989. "Real Brains and Artificial Intelligence," in S. R. Grauman, ed., The Artificial Intelligence Debate: False Starts, Real Foundations (Cambridge, MA: MIT Press).

Rosenblatt, Frank. 1958. "The Perceptron, a Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review, 62, p. 386.

.1962. "Strategic Approaches to the Study of Brain Models," H. von

Foerster, ed., Principles of Self Organization (Elmsford, NY: Pergamon Press).

Rust, John, Richard Palmer and John Miller. 1989. "A Double Auction Market for Computerized Traders," unpublished manuscript, Sante Fe Institute, August.

Schwartz, Jacob T. 1989. "The New Connectionism: Developing Relationships Between Neuro-science and Artificial Intelligence," in S. R. Graubard, ed., The Artificial Intelligence Debate: False Starts, Real Foundations (Cambridge, MA: MIT Press).

Shriver, B. and P. Wegner, eds. 1988. Research Directions in Object Oriented Programming (Cambridge, MA: MIT Press).

Silverberg, Gerard. 1988. "Modelling Economic Dynamics and Technical Change: Mathematical Approaches to Self-Organization and Evolution," in G. Dosi et al., eds., Economic Theory and Technical Change (New York: Printers Publishers).

Simon, Herbert. 1983. Reason in Human Affairs (Stanford, CA: Stanford University Press).

.1985. Sciences of the Artificial (Cambridge, MA: MIT Press).

Smith, Vernon L. 1982. "Microeconomic Systems as an Experimental Science," American Economic Review.

, Gerry L. Suchanek and Arlington W. Williams. 1988. "Bubbles, Crashes, and Endogenous Expectations in Experimental Spot Asset Markets," Econometrica, September.

Sokolowski, Robert. 1989. "Natural and Artificial Intelligence," in S. R. Graubard, ed., The Artificial Intelligence Debate: False Starts, Real Foundations (Cambridge, MA: MIT Press).

Stefik, M. J. 1988. "The Next Knowledge Medium," in B. A. Huberman, ed., The Ecology of Computation (Amsterdam: North-Holland).

Thomas, Dave. 1989. "What's in an Object," Byte, March.

Tullock, Gordon. 1965. The Organization of Inquiry (Durham, N.C.: Duke University Press).

Turkle, Sherry. 1989. "Artificial Intelligence and Psychoanalysis: A New Alliance," in S. R. Graubard, ed., The Artificial Intelligence Debate: False Starts, Real Foundations (Cambridge, MA: MIT Press).

Waldspurger, Carl A., Tad Hogg, Bernardo A. Huberman, Jeffery 0. Kephart and

Scott Stornetta. 1989. "Spawn: A Distributed Computational Economy," unpublishedmanuscript, Xerox PARC.

Winograd, Terry and Fernando Flores. 1986. Understanding Computers and Cognition (New York: Addison-Wesley).

Don Lavoie is an associate professor of eonomics at George Mason University, Fairfax, Virginia, and editor of Market Process. Howard Baetjer and William Tulloh

are both graduate students at George Mason University and fellows of the Center for the Study of Market Processes.