Computers in Context
Bo Dahlbom, Lars Mathiassen

Chapter 1 Computers

Listen to Howard Aiken physicist at Harvard and designer of some of the very first computers, speaking in 1956: "…if it should ever turn out that the basic logics of a machine designed for the numerical solution of differential equations coincide with the logics of a machine intended to make bills for a department store, I would regard this as the most amazing coincidence that I have ever encountered."

Aiken was terribly mistaken, but certainly not for lack of computer technology expertise. Compare his prediction to that of John McCarthy, mathematician and inventor of Lisp, writing in that same year 1956 to the Rockefeller Foundation, asking for money to finance the first Artificial Intelligence conference at Dartmouth College: "The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

Aiken was very much influenced by the usage that had motivated him, and others, when constructing the first computing machines. McCarthy based his view of what computers could do on an understanding of the principles of a general computing machine, rather than on the actual capacity and use of the machines of his time. Together they cover the field, if we are looking for answers to what computers can do.

Aiken is cautious and unimaginative. McCarthy is optimistic, at least when he goes from speaking in principle to predicting practice. But is he right in principle? What are the limits to the possible use of computers? And how are those limits set? By our imagination? By our needs? Or, by the principles of computing?

Using Computers

Think of the first time you suc-ceeded in making a computer do what you intended it to do. Think of the feeling it gave you of being in control of a powerful machine. Before getting too excited, don't forget the times you had to struggle to make the computer do what you wanted. You could not get the right result, or, worse, there was no reaction at all. Remember the direct way in which mistakes or incon-sistencies in your first programs were revealed and the difficulties you sometimes had when you were trying to correct the programs. You knew that something was wrong, but you had no idea what it was.

Computers are fascinating be-cause they are fast, powerful and extremely versatile machines, and because they are programmable. They will obey your most whimsical command, provided they can interpret it. It is really magic: they do exactly what you tell them to do. But of course you have to think and express yourself clearly, you have to be careful with your words.

We can use computers to play with texts and immediately see the conse-quences. It is easy to reuse text and to experiment with different formulations and the sequence in which we present our argument. There is no guarantee, of course, that texts produced on computers are better in quality, clearer and richer. But computers are extremely effective tools for producing, modifying and combining texts, and it is fascinating to play and experiment, trying to be convincing, clear or even poetic.

We also use computers to explore the world without having to suffer the real world consequences. Kids fight monsters without ever getting hurt. Pilots are trained in flight simu-lators without severe risks and for less money. Investments are evaluated without running the risk of losing fortunes. And bridges and highways are designed and tested without suffering collapses or traffic jams.

In general, we use computers to process, communicate, store and keep track of information. And they provide us with new and useful opportunities and liberate us from many laborious and boring tasks. It is not surprising that the development of computer systems raises difficult questions. We use computers to automate administra-tive tasks and to mechanize and automate production processes. But how can we make sure that the good qualities of the traditional manual way of doing things are not lost? And what about unemployment?

We use computers to provide us with information as a basis for decision making. To what extent can we rely on data from the computer? Are data up-to-date and correct? What kinds of interpreta-tions were made when data were originally registered? What about the uncertainties introduced by our own interpretations?

Computers are used to monitor and control complex technical systems to minimize errors and avoid break-downs and catastrophes. But computer systems are themselves complex artifacts that introduce new sources of error and uncertainty. And what about political issues? Do we want to use computers to keep track of people's every move and opinion? Do we want to use them to develop advanced military systems in outer space?

All these questions concern quality. People are concerned with the quality of work as computers replace old work habits and introduce new ones. We have to worry about the quality of data and information when computers are used, as they so often are, to provide decision support. As citizens we should be concerned with the quality of life in a society pervaded by information technology. And as systems developers we are concerned about the quality of a particular computer system in relation to the wants and needs of the customer or user.

When, in this book, we discuss the development and use of computer systems, we have all these aspects of quality, and more, in mind. In fact, our philosophy is that the development of computer systems is a constant struggle with quality. This struggle with quality constitutes the fundamental challenge and fascination of being a systems developer. Before expanding further on this theme, we must, however, go deeper into the very idea of computing.

The Mechanistic Heritage

Electronic computers have been around since World War II. Before then, computers were people working in big insurance companies or ballistic research laboratories performing long and tedious calcula-tions. These human computers used desk calculators to perform simple subtasks of addition and multiplication, combining these subtasks into the computation of more complex functions.

During the war, new artillery weapons were developed at such a pace that the human computers were falling far behind in computing the necessary firing tables. As a result, the American government was interested in supporting attempts to construct "an automatic calculator." Early such machines, like Aiken's Mark 1, used electromagnetic relay technology. The decisive step to a full-blown computing machine was taken when the machines were made electronic with possibilities to store programs in their memories. The electronic representation of data made it possible to change the contents of registers much faster than in the mechanical machines. The idea of looking at computational procedures as data, of storing programs in memory, made it possible and easy to change the function to be computed.

The electronic computers were built to replace the human computers. They were designed as technical devices to be fed with numbers and computational procedures. They could compute according to prescribed procedures and deliver the computed results as output.

Classical computer science has linked human computing and machine computing in the so-called Church-Turing thesis: everything that a hu-man being can compute can be computed by a machine. According to this thesis, our intuitive under-standing of computation refers to the set of computations that can be formally prescribed. Closely related to this idea is the concept of an algorithm.

An algorithm is an instruction (plan or procedure) for how to manipulate some given input to produce some desired output. An instruction is an algorithm if it is finite, definite and effective. That it is finite means that it can be mathematically proved to terminate, that is, finish after a finite number of steps. That it is definite means that each manipulation is well-defined and precise. That it is effective means that each manipulation is so elementary that it can be performed by a human being using just pen and paper, or by a machine.

The algorithm, with its strict definition, is a bridge between our intuitive notion of (human) computation and the idea of a computing machine. Any algorithm can be implemented on a machine so if human computation is algorithmic, then the Church-Turing thesis is true. But notice that this thesis cannot be mathematically proved, since it relies on an intuition about what it is for a human being to compute.

The construction of the electronic computer was the crowning achievement of a long tradition in our culture identifying thinking with computation. This theory of thinking was formulated clearly already by the French philosopher and scientist René Descartes (1596-1650). Inspired by his work in mathematics, Descartes developed a theory of thinking as the rational manipulation of symbols by means of rules. When people spoke of the early computers as electronic brains, as intelligent, thinking machines, they simply applied the Cartesian idea of thinking as computation.

But something is obviously wrong here. Thinking is the essence of being human, according to Descartes, the human mind is a thinking thing, a res cogitans. But the computers we see around us are anything but human. They don’t reason, argue, plan, fantasize, imagine, memorize, hope, or foresee? They compute but they don’t think. Does this mean that Descartes was wrong? And not only Descartes, but our whole modern conception of thinking underly-ing the construction of the computer and the development of computer science? Some would say yes, there is something wrong here. Others would argue that it is only a matter of pro-gramming: If the computer on your desk does not seem to be a thinking thing, this is only because there has not been enough programming done.

Like so many other questions raised by the ongoing comput-erization of work and society, your answer to this question will express your fundamental view of the world. If your answer is “yes, with more programming, computers will become thinking machines,” it is likely that you will agree with many of the ideas of the Mechanistic world view as developed by the great 17th century system builders, like Descartes and the German philosopher and mathematician Gottfried Wilhelm von Leibniz (1646-1716). In this world view we find a general and powerful idea about computation, related to ideas of representation, formalization, program, order and control.

If your answer is “no, a machine can never be made to think,” then you are probably more of a Romantic, more attuned to the Romantic world view of the early 19th cen-tury. This is a very different conception of the world based on ideas about interpretation, uniqueness, chaos and change. In the next chapter we shall discuss this Romantic alternative.

We shall begin, in this chapter, by taking a more careful look at our Mechanistic heritage. This heritage exerts a powerful influence on all of us, whether we believe in the project of turning computers into thinking machines or not.

But before we do that, let us just say a few words about world views. We don’t often find reason to think deeply about how we view the world, and we seldom have to formulate or pledge allegiance to a coherent world view. Our world views tend to consist of more or less loosely related, sometimes conflicting elements, some explicit and some tacit, gathered from very different sources.

Most of us have world views that combine both Mechanistic and Romantic ideas, even if we would not normally identify them by these terms. Just in order to be interested in computers, you have to be Mechanistically oriented. And in order to be the kind of person who would read a book like this, you probably have to be touched by Romantic ideas.

A discussion of the Mechanistic world view is not only an examination of ideas that have dominated the development and use of the electronic computer. It is also a way of reflecting on our own ideas about computers and their use, reflecting on the roots, nature and legitimacy of those ideas.

Representation and Formalization

A powerful process of change, called modernization, began in 16th century Europe. The modernization process affected all aspects of human life. It changed a society dominated by farming and craft, religion and miracles, authority and tradition into a world of technology and science, democracy and liberty, progress and revolutions. Religious upheavals, industrial expansion, the central perspective in art, the exploitation of America and the invention of the machine were all parts of this process of modernization.

To come to grips with their changing world, the modern Europeans developed natural science. The new physics taught that our naive perception of the world is loaded with errors, that the world is not what it seems to be. The sun seems to be moving across the sky, but isn't. Objects seem to be colored, but aren't. With the recognition of systematic perceptual errors, the world and our representation of it become clearly distinguished. Our perception of the world does not coincide with the world itself. The world itself is out there, our experience of it is in here, in the mind.

This dualism of external world and inner life that we now take for granted is really a modern, recent idea. The explicit distinction, between the world and our representations of it, was a necessary condition for the project of modern science: to replace our naive perceptions with true, scientifically based representations of reality.

No one did more than Descartes to promote the idea of knowledge as the representation in the mind of a world out there. The mind is a mirror of the world, Descartes could say in a time when glass-making had advanced enough to make mirrors high fashion. But an imperfect, cloudy mirror distorts its object, and our minds are not to be trusted, so we have to make sure that our ideas are clear and exact before we depend on them to give us an accurate picture of the world.

Unclear ideas confuse us about the world and get confused with one another. Images, the pictures of the world we see in our mind, contain an abundance of material—subjective properties like color, taste, and smell—that is not to be found in the world. The Mechanistic world view challenges us to strip our ideas of such subjective material in order to make them true to the world. Like Galileo (1564-1642) before him, Descartes came to argue that we have to use mathematics as a means of representa-tion in order to map the world in a clear and exact way. The world is like an open book for us to read, and it is written in the language of mathematics.

Galileo worked with geometry, using lines, triangles and squares to represent properties like velocity and acceleration. Transforming geometry into algebra in his analytic geometry, Descartes could do mechanics with numbers and algebraic functions and argue that algebra provides us with the exact symbols we need to represent the world truthfully.

Today, it is easy for us to recognize the powerful idea of formal-ization in the Mechanistic struggle to arrive at clear and exact ideas. But the Mechanists did not only want to formalize the ideas or symbols we use to represent the world, they also wanted to formalize the process of thinking itself. With the use of mathematical symbolism as means of representation follows the idea that thinking is the manipulation of these symbols, that thinking is mathematical reasoning, calculation and proof. What dis-tinguishes mathematical reasoning from ev-eryday pondering is its ex-plicit reliance on rules, on logic. Just as the use of mathematics in representing the world is moti-vated by a desire for exact, explicit knowledge, so the idea of thinking as computation is developed to give us a conception of thinking as exact and explicit rule following.

We have to realize, of course, that the idea of thinking as mathematical computation, as formulated by Descartes and later developed by Leibniz, was more of an idea than a full-blown theory. Both symbolism and math-ematical reasoning were in those days still fairly informal and needed much work before we could truly speak of them as exact and explicit.

Leibniz worked hard on developing a mathematical notation, a universal calculus, that could serve as means for represent-ing and reasoning about all our knowledge of the world. As part of this project he built several small calculators to test his ideas of exact reasoning. In these efforts he was motivated by the modern belief in knowledge as the universal problem solver and means to progress. He hoped that such a language, used both locally and in international diplomacy, would put an end to conflicts of all kinds, based as they were on misunderstanding due to the use of inexact, informal language. Indeed, a powerful dream in a 17th century Europe ravaged by wars and religious conflicts.

By introducing explicit rules for how to express ourselves, we can develop more exact representations and avoid misunderstanding. The computer, as we have come to know it, is based on the application of explicit rules and the idea of for representation.

For computers to be of any use, we have to agree on how to apply a certain concept or how to interpret a specific physical state as a symbol. The computer is of no use without formalizations. And without the computer there would be one reason less to formalize observa-tions about patients in hospitals, or information about employees and cus-tomers in the organizations where we work.

The Mechanistic ideas of representation and formalization are at the very heart of computing. Data are representations of facts, and the computer is a technology for storing and manipulating data. Without the idea of representation we would have no computers. Without computers we would not have the same potential for automatically manipulating our representations of the world.

Rules and Rationality

The modern world view is a rational view of the world. Rational think-ing is the conscious, competent administration of ideas, aided by a method. To rationalize is to set on rules, to develop methods, write up pro-grams. To be really rational you need not only follow rules, you have to know and be able to state and defend the rules you are following in your thinking. Before you undertake an action, you formulate the rules; before you develop a system, you formulate your method. The real work lies in choosing, formulating and motivating the rules, the method. The rest is routine. A machine can do it.

Modern science with its method is only one example of a social institution rationalized by explicit rules. Descartes' scientific method—his "rules for the direction of the mind"—has its coun-terpart in the explicit and unequivocal rules controlling a bureau-cracy, in the written instruction manual for op-erating and repairing a modern machine, in the explicitly formulated consti-tution of a modern democratic society, in the official curricu-lum for a modern educational institution, and so on.

For the Mechanists it is the business of science to map the world, to give a systematic, preferably axiomatized, definite, and true account of the world. This is possible because the world it-self is an ordered, fundamentally unchanging system.

Newton's mechanics, with its three basic laws, is the outstanding example of an ordered system. Leibniz gave us a particularly powerful conception of the world as such a system, a world that is deterministic and governed by two fundamental principles: the principle of sufficient reason and the principle of pre-established harmony. Nothing can exist without a reason, and everything that is has to be in harmony with everything else that is.

The 17th century intellectuals were fascinated by machines, and this interest influenced their image of the world. Descartes and Leibniz both played with machines as toys and as powerful ideas. They saw not only the world as a machine, but also the body (Descartes) and society (Leibniz). They laid the foundation, as we have seen, for thinking of thinking itself as a me-chanical process.

When you see something as a machine, be it the world, the body or society, you want to take it apart and make explicit the rules governing its behavior. When you know the rules, or perhaps better the laws, governing the functioning of a machine, then you can control it. When you realize that the heart is a pump you know how to deal with it. A society governed by rules can be controlled by those formulating the rules. To follow rules in your thinking is to control your thinking.

The Mechanistic emphasis on ra-tionalism, on methods and programs, has had a strong impact on the development of computer technology and on the ways we think of programming and systems development.

To use computers we need programs, and to program we need methods. The history of computing can be seen as a continuing attempt to develop programming languages and methods for programming.

But the real challenge in computing lies in exploring and applying the Mechanistic world view, while at the same time understanding and appreciat-ing the limits of its application. When we try to control the world with computer programs or methods for systems development, we should not forget that the Mechanistic world view is based on the assumption that the world we are trying to understand and control is itself an ordered, fundamentally unchanging system.

Computers and Bureaucracies

The Mechanistic world view has influenced organizations and society long before the invention of the electronic computer, and we can learn a lot about computers and formalization by looking more generally at the ways we organize human activities.

Every organized human activity — from producing milk to developing computer systems — gives rise to two fundamental and opposite requirements: the division of labor into various tasks, and the coordination of these tasks to accomplish the activity as a whole.

If more than one person participates in a systems development project, we must divide the task, for in-stance by designing the system as a set of related modules. Such modules are not only parts of the final system. They also define separate work tasks to be performed as part of the devel-opment effort. One of the important functions of a design specification is to serve as a basis for the division of labor during the project.

But a good design document, dividing the labor well, does not automatically lead to a satisfactory system. During implementation, we must evaluate and test the indi-vidual modules, and, even more cru-cial, perform integration tests and evaluate the operation of the total system. In addition to design documents defining the division of labor, we need other techniques to support the coordination of individual tasks to accomplish a satisfactory result as a whole.

Organizations use different strategies for dividing labor and achieving coordination between individuals and groups. Ship yards, textile factories, hospitals, car repair shops and systems development projects each have different approaches to effective organization.

To understand the differences between organizations and to facilitate the design of effective organizations we identify a number of abstract or ideal types of organization with different structural characteristics. The most well-known is the bureaucracy.

An organization is bureaucratic to the extent that the behavior of its actors is predetermined or predictable. The bureaucratic approach to organization is to rely on rules in prescrib-ing behavior and achieving coordination. Bureaucracies are programmed. The assumption is that we know in advance what to do: the task uncer-tainty of the organization is low.

Bureaucracies are designed to be efficient by minimizing direct interac-tion between individuals and groups. Coordination is achieved by having each group or individual follow prescribed rules. When the rules do not apply, decisions are made by a hierarchy of managers, the hierarchy being the most efficient way to organize communication.

In a bureaucracy, management is kept separate from actual production. Workers are not supposed to make decisions. They produce goods or services accord-ing to in-structions, only informing their managers about deviations and problems. Managers make decisions. They develop new plans and formulate instructions based on previous plans and status information.

A bureaucracy is like a computer, and like the computer, it is a powerful expression of Mechanistic ideals. A bureaucratic orga-nization is programmed, its work tasks are explicitly defined and formalized. It is a machine in which computing machines have their natural place, providing efficient processing and communication of information about products, activities and resources.

The computer is a perfect bu-reaucrat, and it invites us to think like bureaucrats. We cannot use it without formalizing. The formalization imperative is in most cases quite obvious. We cannot develop a com-puter based account-ing system without formalizing what is meant by an account and what is meant by various transac-tions on an account. In other cases, the formal-ization impera-tive is less obvious even if it is still there. In developing a com-puter based text processing system we do not have to formalize what we write about. But we do have to formalize the format in which we write it. Otherwise the computer will be of little help in manipulating, storing and communicating the text.

Traditional production control sys-tems provide a classic example of the bureaucratic use of computers. Coordination is viewed as a rational decision process where sta-tus information is produced on the shop floor and compiled through the computer system. Plans are cre-ated by production managers and foremen on different levels of detail and these plans are distributed through the computer system.

The role of the foreman is to make de-tailed plans expressing what each individual worker has to do and how the machines on the shop floor should be utilized. The foreman communicates these plans to the workers, and they report back to him when jobs are finished or breakdowns occur. He makes decisions on how to handle delays and break-downs, compiles reports to the production planner, and receives overall plans for the production in his department.

The computer is used to communicate, process and store the information needed to manage and coordi-nate the activities on the shop floor into one integrated effort. Workers, foremen and production planners use the computer to generate, receive and process information. The assumption is that the information provided by the computer system represents the actual state of production and commitments. The computer is used in ac-cordance with the Mechanistic view of the world to support rational decision making in optimizing management of resources.

In our everyday activities we all rely on bureaucratic approaches. And we have begun to use computers to support us in doing so. But we tend to forget the basic weakness of the bureaucratic approach. When the environment changes and the task uncertainty increases, we are ill prepared. Since our behavior and thinking have been shaped by bureaucratic procedures, we are unwilling to engage in change, and we don't have the necessary resources to do so. Our computers are not making it any easier.

Coping with Change

An organization operates in an environment. This environment comprises everything outside the (full control of the) organization that is of importance to its performance. This includes the nature of its products, customers, and competitors, the economic and political climate in which it must operate, and the kind of technolo-gies on which it is depends.

The environment of a specific organization may range from stable to dynamic, from that of the stone mason whose customers demand the same product decade after decade, to that of a medical research team trying to develop a new and effective medicine. A stable environment may, of course, change over time. But the variations are predictable. In contrast, the environment of the research team is dynamic in the sense of being highly unpredictable. The team has one or more theories on how to cure a specific disease, but they have no certain knowledge about what technologies to use, nor can they effectively predict whether and when they will succeed in their efforts. They simply have to try.

The effectiveness of an organizational structure is strongly dependent on the environment of the organization. The economic use of information is the strength, but also the weakness, of the bureaucracy. The more the environment of a bureaucracy changes, the more often its rules will not apply, and the more the management hierarchy will have to intervene. The Achilles heel of a bureaucracy is its inability to respond effectively to change.

When a bureaucratic organization fails, more organic structures will emerge. An organic structure can be defined by the absence of formalization, by the absence of explicit rules prescribing or determining the behavior of the actors. Bureaucratic structures are hierarchical. Organic structures are more like networks.

The organic approach to or-ganization is to rely on informal and direct interaction to achieve coordination between individuals and groups. The assumption is that the task uncertainty of the organization is high. New information related to organizing the activity will therefore become available as the activity is performed. To make this information useful, the involved actors must have the opportunity and obligation to communicate and interact. All actors are supposed to engage actively in decision making and planning as the activities are performed. Management and production are integrated.

Bureaucratic organizations are like machines, organic organizations like living organisms. Bureaucracies are based on a belief in, and a striving for, stability, order and control. Organic structures are designed to cope effectively with dynamic environments. They go beyond the basic assumptions of the Mechanistic world view, offering a constructive response to the limitations of bureaucracies.

Strangely enough, the computer has proved to be a useful element in organic strategies. In electronic mail systems some degree of formalization is required. Communication forms and facilities for storing and retrieving information are formalized. Also, such systems have to provide formal informa-tion about the addresses in the network. The success of electronic mail systems, however, lies in the informal and direct way in which communication is possible.

Electronic mail systems are explicitly designed to support organic coordination. But traditional computer systems can also be reinterpreted from an organic point of view. Let us illustrate how we can learn more about the use of a traditional production control system by such a reinterpretation.

From an organic perspective, coordination can be viewed as dynamic negotiation and creation of commitments. The role of the foreman is to act as middle man between the production planners and the workers. As in the bureaucracy, workers, foremen and production planners communicate in a cooperative effort to coordinate activities. But at the same time they play opportunistic games for individual gain. Information is not necessarily to be trusted, because individual actors might withhold information or, even worse, misinform to gain a personal advantage.

The foreman is not primarily a decision maker. He has to listen, negotiate, and manipulate to manage resources in a satisfactory way. As in the bureaucratic case, the computer based production control system is used to communicate, process and store information. But to be effective, the com-puter system should also support the actors in negotiating and administrating commitments, in addition to traditional production planning.

The organization is no longer viewed as a machine but rather as a dynamic network or an organism. Uncertainty of information has become an impor-tant is-sue and the best one can hope for is a satisfactory, rather than optimal, utilization of resources on the shop floor.

We often rely on organic approaches to organization in our everyday activities. We discuss with colleagues, we participate in ad-hoc meetings, and we are members of autonomous groups that have been assigned more or less well-defined tasks. We have also become accustomed to using computers in doing so.

People cooperate across organizational and national bound-aries. Project groups are formed in which the members are organiza-tionally and geographically dispersed. These kinds of organic structures would not be effective, or even possible, without technologies like telephone and airmail that can link the actors to-gether in an informal and highly interactive fashion. But computer networks and hypermedia radically increase our ability to informally and directly coordinate activities between groups and individual actors across organizational and geographical boundaries.

But before the picture we have painted of organic forms of thinking and organizing becomes too glowing, we had better stop and consider its weaknesses. And then it is clear, of course, that with an organic approach we do not utilize the fundamental strength of the computer: its ability to process formalized information. With the use of a computer we do not have to write each letter from scratch. We can use a standardized format, and let the computer aid us in comparing, categorizing, and evalu-ating messages. Often it is conve-nient to have some kind of standardized format, or even standardized ways of providing the information by enumerating in advance the possible choices as in a traditional questionnaire.

There are both practical and economic reasons for formalizing information when using computers. In general, we know that the more stable and repetitive our work, the more we tend to apply bureaucratic approaches. Not only because we are told to do so, but because we find it effective and helpful.

In the way exemplified here, we can go back and forth between the way we think of computers and the way we think about social organizations, in the process learning more about both. Mechanistic ideals and the formalization imperative continue to have a strong influence on both computers and organizations. Modular thinking and hierarchical structures have dominated our conception of computer systems. To deal with the complexity of these systems we have struggled to separate concerns as we divide labor in the classical bureaucracy. We have also tried to economize with exchange of information by relying on hierarchical structures thereby minimizing the number of active information channels between modules. We have been very successful in turning the computer into a perfect bureaucrat, computer systems into well ordered bureaucracies.

But as a consequence of this, the technology is not particularly well suited to the use in more organic forms of organizations. And its bureaucratic nature has been a restraint on our imagination in finding new use for the technology. For a long time its use was mainly restricted to the bureaucratic functions of organizations and it acted as a support for the bureaucratic nature of those organizations. But that is changing now as our interest in organic forms of organization has grown enough to make us begin to rethink the very nature of computer technology.

Parallel architectures, neural nets, networks and hyperstructures are all examples of more organic ways of thinking of computers and computer systems. New ways of organizing have changed the way we think about computers and vice versa.

The computer artifact has long since out-grown the image of the human computer. As its use has diversified our understanding of it has diversified as well. If we work with electronic mail systems we naturally focus on those qualities of computers that make us think of them as media. If we work with statistical program packages we will attend to other qualities and see the artifact as a computing machine. Understanding and evaluating a text processing system it is best to treat the system as a tool for producing texts. But in many cases several competing perspectives are equally relevant in understanding and designing computer systems. If all you have is a hammer, the world tends to look like a nail. It isn't.