Studies in Computer Science
and Artificial Intelligence 2

The Ecology of Computation

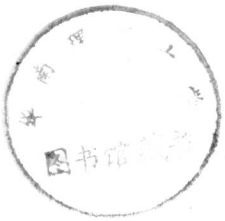
B.A. Huberman
editor

North-Holland

THE ECOLOGY OF COMPUTATION

edited by

B. A. HUBERMAN
Xerox Palo Alto Research Center
Palo Alto, CA
U.S.A.



1988

NORTH-HOLLAND – AMSTERDAM • NEW YORK • OXFORD • TOKYO

ISBN: 0 444 70375 6

Publishers:
ELSEVIER SCIENCE PUBLISHERS B.V.
P.O. Box 1991
1000 BZ AMSTERDAM
THE NETHERLANDS

Sole distributors for the U.S.A. and Canada:
ELSEVIER SCIENCE PUBLISHING COMPANY, INC.
52 VANDERBILT AVENUE
NEW YORK, N.Y. 10017
U.S.A.

Library of Congress Cataloging-in-Publication Data

The Ecology of computation.

(Studies in computer science and artificial intelligence ; 2)
1. Electronic data processing. 2. Artificial intelligence. I. Huberman, B. A. (Bernardo A.), 1943- . II. Series.
QA76.E26 1988 004 87-36405
ISBN 0-444-70375-6 (U.S.)

PRINTED IN THE NETHERLANDS

STUDIES IN COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

2

Editors:

D. G. Bobrow
Xerox Corporation
Palo Alto Research Centre
Palo Alto, California

H. Kobayashi
IBM Japan Ltd.
Tokyo

J. Nievergelt
ETH, Institut für Informatik
Zürich

M. Nivat
Université Paris VII
Paris

NORTH-HOLLAND – AMSTERDAM • NEW YORK • OXFORD • TOKYO

CONTENTS

The Ecology of Computation
B.A. Huberman (editor)

The Ecology of Computation

B. A. Huberman
Xerox Palo Alto Research Center
Palo Alto, CA 94304

A new form of computation is emerging. Propelled by advances in software design and increasing connectivity, distributed computational systems are acquiring characteristics reminiscent of social and biological organizations. These open systems, self–regulating entities which in their overall behavior are quite different from conventional computers, engage in asynchronous computation of very complex tasks, while their agents spawn processes in other machines whose total specification is unknown to them. These agents also make local decisions based both on imperfect knowledge about the system and on information which at times is inconsistent and delayed. They thus become a community of concurrent processes which, in their interactions, strategies, and competition for resources, behave like whole ecologies.

The appearance of such complex systems on the computational scene creates a number of interesting problems. At the operational level, the lack of global perspectives for determining resource allocation gives rise to a whole different approach to system level programming and the creation of suitable languages. Just to implement procedures whereby processes can cross trust barriers and manage to compute in a highly heterogeneous medium is a challenging task with no optimally known solution. Although human organizations often deal successfully with the problem of asynchronous operation and imperfect knowledge, the implementation of a computational analog is far from obvious. Nevertheless, pieces of such systems are already in place, and a

serious effort at designing open computational networks is under way in a number of laboratories.

On a different vein, the existence of computational ecologies leads to a number of fascinating questions concerning their function, dynamics, and efficiency. Stated succinctly, one would like to understand the overall system behavior from knowledge of what the individual processes can do. Since the rules whereby computational agents choose among possible strategies can be arbitrarily set by the designer, whole scenarios can be artificially created and tested. Moreover, the intrinsic nonlinearity of such systems leads to a rich repertoire of behaviors which can be studied at both the theoretical and experimental level. This leads in turn to a consideration of concepts such as evolutionarily stable strategies. Also, since computational ecologies have much in common with biological organizations, one expects that insights gained from one will help the understanding of the other. Just as insect colonies can reveal the workings of a natural computational system with simple components, the dynamics of an artificial open system can provide a quantitative testing ground for models of social organizations.

This book is a collection of articles which deal with the nature, design and implementation of open computational systems. Although varied in their approach and methodology, they are related by the goal of understanding and building computational ecologies. They are grouped in three major sections. The first one deals with general issues underlying open systems, studies of computational ecologies, and their similarities with social organizations. The second part deals with actual implementations of distributed computation, and the third one discusses the overriding problem of designing suitable languages for open systems. The book ends with a vision of a future knowledge medium by Stefik.

Imperfect knowledge, asynchronous computation and inconsistent data are not exclusive of open computational systems. Human societies face the same constraints, often successfully, when trying to engage in collective problem solving, be it a scientific community or a design group. Such an analysis, in the pervasive context of office work, is presented by Hewitt from an open systems perspective.

The consideration of a computational system as an ecology brings to mind biological mechanisms such as mutations, which introduce variations into species. These alterations in the code of life, while leading to increased diversity,

allow for adaptation to a changing environment. Such mutation strategies have been often proposed as ways of improving the performance of artificial intelligence systems. A highly instructive example of such an approach is provided by *AM* and *Eurisko*, which were designed in order to explore and discover mathematical concepts by syntactically mutating small Lisp programs. Equally interesting is the process of resource allocation to the resulting programs, and which are distributed by an external agent on the basis of perceived degree of progress. Lenat and Brown perform an incisive analysis of the advantages and shortcomings of such systems, and speculate that the paradigm underlying them may be that of collections or societies of evolving, self–organizing, symbolic knowledge structures.

Within this context, Miller and Drexler discuss several evolutionary models such as biological ecosystems, human markets, and *Eurisko*, and outline their analogies and differences with computational ecologies. In this and a related paper on *Markets and Computation*, they elaborate a vision of direct computational markets which they term *agoric* open systems.

Since incomplete knowledge and delayed information are intrinsic features of computational ecologies, serious consideration has to be given to their dynamical behavior when operating with such constraints. Huberman and Hogg derive and analyze the appropriate equations governing game dynamics, and show that when processes can choose among many possible strategies while collaborating in the solution of computational tasks, the asymptotic dynamics can lead to nonlinear oscillations and chaos. These results imply that evolutionarily stable strategies may not exist in computational ecologies. They also discuss the possible existence of a universal law regulating the way in which the benefit of cooperation is manifested in the system, and compare it with findings in biological ecologies and human organizations.

This dynamical approach to distributed computation is to be contrasted with the static static one of Rosenchein and Genesereth, who apply classical game theory to resolve potential conflicts between computational agents having disparate goals.

The second part of the book discusses actual implementations of distributed computational systems. Two different papers describe operational systems designed from an open systems perspective. *Enterprise*, a market–like scheduler, described by Malone and collaborators, consists of independent processes or agents being allocated at run time among remote idle processors

through a bidding mechanism. The system provides substantial performance improvements over processing tasks on the machines at which they originate even in the face of large delays and inaccurate estimates of processing times.

An alternative system is *Mach*, described by Rashid in an article outlining the evolution of a class of network operating systems. A multiprocessor operating system kernel currently running on VAX architecture machines, *Mach* provides a number of attractive features from an open systems point of view. These include support for transparent remote file access between autonomous systems, internal symbolic debuggers, and most notably, network interprocesses which can be protected across system boundaries.

This section on distributed computation ends with an article by Drexler and Miller on *Incentive Engineering*. Within the framework of market–like mechanisms for resource allocation, it proposes a set of algorithms which allow for both processor scheduling as an auction process, and for distributed garbage collection through which unreferenced loops that cross trust boundaries can be collected.

This book would not be complete if it did not dealt with the problem of languages for computational ecologies. A suitable programming language for open systems should allow for programs in which modules reside and execute at geographically remote, but communicating, locations. It should also allow for the writing of robust programs which can survive hardware failures without loss of distributed information, while allowing concurrent access to that information while preserving its consistency. An example of such a programming language is provided by Liskov and Scheifler in their article on *Guardians* and *Actions*. In a more general vein, Kahn and Miller analyze the suitability of actor languages and concurrent logic programming for writing programs which both provide services and take advantage of them in a manner that scales from basic computational steps to very large distributed systems.

Lastly, there is a visionary description by Stefik of a future in which AI systems, distributed across society, will be able to communicate and share knowledge with each other. This *knowledge medium* will stand in sharp contrast to current expert systems which are built from scratch every time and essentially function as stand alone entities. In his description of such a medium, Stefik draws heavily both on the history of cultural changes and most importantly, on the notion of a knowledge ecology.

The Ecology of Computation
B.A. Huberman (editor)
Elsevier Science Publishers B.V. (North-Holland), 1988

Offices Are Open Systems

Carl Hewitt

MIT Artificial Intelligence Laboratory

This paper is intended as a contribution to analysis of the implications of viewing offices as open systems. It takes a prescriptive stance on how to establish the information-processing foundations for taking action and making decisions in office work from an open systems perspective. We propose *due process* as a central activity in organizational information processing. Computer systems are beginning to play important roles in mediating the ongoing activities of organizations. We expect that these roles will gradually increase in importance as computer systems take on more of the authority and responsibility for ongoing activities. At the same time we expect computer systems to acquire more of the characteristics and structure of human organizations.

1. INTRODUCTION

In this paper we discuss the nature of office work from an open systems perspective. Coping with the conflicting, inconsistent, and partial information is one of the major challenges in office information systems. Due process is the organizational activity of human and computer systems for generating sound, relevant, and reliable information as a basis of action taking. Within due process logical reasoning takes place within relatively small coherent modules called microtheories. In general the microtheories will be inconsistent with one another. Due process makes use of debate and negotiation to deal with conflicts and inconsistencies between microtheories.

2. OFFICE WORK

We define an *office* as a place where *office work* is done, thus shifting the emphasis of our investigation from the nature of the locale to the nature of the activity performed. Office work can take place in an automobile with a mobile telephone, in the anteroom of a lecture hall, or at a networked personal computer. Of course, the situation including place, time, and participants can materially affect the work. All office work takes place within a particular concrete situation. The point that we want to make here is that there is no *special* place where office work has to take place.

Reprinted from *ACM Transactions on Office Information Systems*, Vol. 4, No. 3, July 1986, pp. 271-287.

Later we discuss how office work is situated in *particular concrete* space and time and how the situation provides an important part of the context in which the work is done.

We take *office work* to be information processing that is done to coordinate all the work that an organization does with the exception of direct manipulation of physical objects. The organizations in which office work takes place are "going concerns" in the sense of Everett Hughes [11]. For example, they include the processing of beliefs, goals, and mutual commitments as well as the development and management of responsibilities, policies, tasks, transactions, projects, and procedures. Office work is specialized by excluding *robotics*. Robotics involves information processing directly involved in the physical production, transformation, transportation, servicing, or consumption of physical objects.

Office work is situated social action in the sense that it is the action produced by participants at particular times and places. However, we need to extend the usual notion of situated social actions to encompass the social actions of computer systems in their interactions with other computer systems as well as the interactions of computer systems with human participants.

3. OPEN SYSTEMS

Offices are inherently open systems because of the requirement of communication with operational divisions as well as the external world in the task of coordinating the work of the organization. In all nontrivial cases the communication necessary for coordination takes place asynchronously. Unplanned dynamic adaptation and accommodation are required in organizational information systems to meet the unplanned changing needs of coordination since the execution of any plan requires articulation, change, and adjustment.

Open systems deal with large quantities of diverse information and exploit massive concurrency. They can be characterized by the following fundamental characteristics [9]:

(1) *Concurrency.* Open systems are composed of numerous components such as workstations, databases, and networks. To handle the simultaneous influx of information from many outside sources, these components must process information concurrently.

(2) *Asynchrony.* There are two sources of asynchrony in open systems. First, since the behavior of the environment is not necessarily predictable by the system itself, new information may enter the system at any time, requiring it to operate asynchronously with the outside world. Second, the components are physically separated distances prohibiting them from acting synchronously. Any attempt to clock all the components synchronously would result in an enormous performance degradation because the clocks would have to be slowed down by orders of magnitude in order to maintain synchronization.

(3) *Decentralized control.* In an open system, a centralized decision maker would become a serious bottleneck. Furthermore, because of communications asynchrony and unreliability, a controlling agent could never have complete, up-to-date information on the state of the system. Therefore control must be

distributed throughout the system so that local decisions can be made close to where they are needed.

(4) *Inconsistent information.* Information from outside the system or even from different parts of the same system may turn out to be inconsistent. Therefore decisions must be made by the components of an open system by considering whatever evidence is currently available.

(5) *Arms-length relationships.* The components of an open system are at an *arms-length relationship*: The internal operation, organization, and state of one computational agent may be unknown and unavailable to another agent for reasons of privacy or outage of communications. Information should be passed by explicit communication between agents to conserve energy and maintain security. This ensures that each component can be kept simple since it only needs to keep track of its own state and its interfaces to other agents.

(6) *Continuous operation.* Open systems must be reliable. They must be designed so that failures of individual components can be accommodated by operating components while the failed components are repaired or replaced.

4. CONCURRENCY

The underlying concurrent basis of operation enables due process to react dynamically to asynchronous input and in many cases makes the results indeterminate.

4.1 Asynchronous Input

Concurrent systems differ from Turing machines in that they allow asynchronous communication from the external environment to affect ongoing operations. Sequential systems deal with this problem as a kind of "interrupt" in which they "switch tasks." Organizational information systems rarely have all the material at hand needed to make an important decision. Information that is known in advance to be required arrives asynchronously as the decision making proceeds and is often incomplete. Unanticipated information can arrive at any time in the process and affect the outcome even though it arrives quite late. For instance, an unanticipated story in the *Wall Street Journal* on the morning of a corporate board meeting to give final approval to a merger has been known to kill or delay a deal.

4.2 Indeterminancy

Concurrent systems are inherently indeterminate. The indeterminancy of concurrent systems does not stem from invoking a random element such as flipping a coin. Instead it results from the indeterminate arrival order of inputs to system components. In general, complete knowledge of the state and structure of a concurrent system together with exact knowledge of the times and values of inputs does not determine the system's output. Concurrent systems are indeterminate for the same reason that other quantum devices are indeterminate.

The indeterminancy of concurrent computation is different from the usual nondeterministic computation studied in automata theory in which coin flipping

is allowed as an elementary computational step. In general, it is not possible to know ahead of time that a concurrent system will make a decision by a certain lime. Flipping a coin can be used as a method of forcing decisions to occur by making an arbitrary choice. Often as a matter of principle, however, due process refuses to invoke arbitrary random measures such as coin flipping to make a decision. For example, a jury might not return a verdict, and the judge might have to declare a mistrial. (Agha [1] provides an excellent exposition of the nature of a mathematical model of concurrent computation and its differences with classical nondeterministic Turing-machine-based theories.)

5. CONFLICTING INFORMATION AND CONTRADICTORY BELIEFS

Conflicting sources of information and inconsistent beliefs are a staple of life in organizational information systems. This partly results from dealing with differing external organizations that retain their own autonomy and belief structures.

Inconsistencies inevitably result from the measurements and observations made on complicated physical systems. Higher level abstractions are used to attempt to construct a consistent description of parts of the environment in which the organization operates. For example, a firm's earnings might be labeled "provisional" and then "subject to audit." But, even after being published in the annual report, they might later have to be "restated." In this case "provisional," "subject to audit," and "restated" are attempts to construct a consistent description from conflicting information about earnings.

Whatever consistency exists among the beliefs within an organization is *constructed* and *negotiated* by the participants. In the case of reported earnings, the chief executive officer, finance department, board of directors, and regulatory authorities play important roles in constructing and negotiating the financial reports.

Any belief concerning an organization or its environment is subject to internal and external challenges. Organizations must efficiently take action and make decisions in the face of conflicting information and contradictory beliefs. How they do so is a fundamental consideration in the foundations of organizational information systems.

Conflicting information and contradictory beliefs are engendered by the enormous interconnectivity and interdependence of knowledge that come from multiple sources and viewpoints. The interconnectivity makes it impossible to separate knowledge of the organization's affairs into independent modules. The knowledge of any physical aspect has extensive *spatiotemporal*, *causal*, *terminological*, *evidential*, and *communicative* connections with other aspects of the organization's affairs. The interconnectivity generates an enormous network of knowledge that is inherently inconsistent because of the multiple sources of actors making contributions at different times and places.

For example, suppose that in the middle of 1986 an organization undertakes to consider its knowledge of sales currently in progress for that year for the New England region. In such a situation, there is an enormous amount of

information about other pieces of information. The following considerations show a small part of the enormous interconnectivity of knowledge:

Spatiotemporal interconnectivity. The organization has a great deal of knowledge about the history of sales in the New England region in the first few months of 1986, including how the sales were generated and recorded. In addition, it has sales projections of what will happen in the remainder of the year.

Causal interconnectivity. The marketing department believes that increased advertising is causing sales to go up. On the other hand, the sales department believes that the increased sales commissions are the real reason for the increase in sales.

Terminological interconnectivity. Some of the sales are really barter agreements with uncertain cash value. Do the barter agreements qualify as sales?

Evidential interconnectivity. The accounting department fears that sales might really not be increasing because many of the products could be returned because of a new 30-day free trial offer. It does not believe that the evidence presented shows that sales are increasing.

Communicative interconnectivity. The organization consists of a community of actors operating concurrently, asynchronously, and nondeterministically. The asynchronous communications engender interconnectivity, which defies any complete description of the global state of the organization at any particular point in time.

Conflicting information and contradictory beliefs are an inherent part of office work that must be explicitly addressed in any foundation for organizational information systems.

6. DUE PROCESS

Due process is the organizational activity of humans and computers for generating sound, relevant, and reliable information as a basis for decision and action within the constraints of allowable resources [4]. It provides an arena in which beliefs and proposals can be gathered, analyzed, and debated. Part of due process is to provide a record of the decision-making process that can later be referenced.

Due process is inherently reflective in that beliefs, goals, plans, requests, commitments, etc., exist as objects that can be explicitly mentioned and manipulated in the ongoing process.

Due process does not make decisions or take actions per se. Instead it is the process that informs the decision-making process. Each instance of due process begins with *preconceptions* handed down through traditions and culture that constitute the initial process but are open to future testing and evolution. Decision-making criteria such as preferences in predicted outcomes are included in this knowledge base. For example, increased profitability is preferable to decreased profitability. Also, increased market share is preferable to decreased

market share. Conflicts between these preferences can be negotiated [18]. In addition preferences can arise as a result of conflict. Negotiating conflict can bring the negotiating process itself into question as part of the evaluative criteria of how to proceed, which can itself change the quality of conflict among the participants [7, 7a].

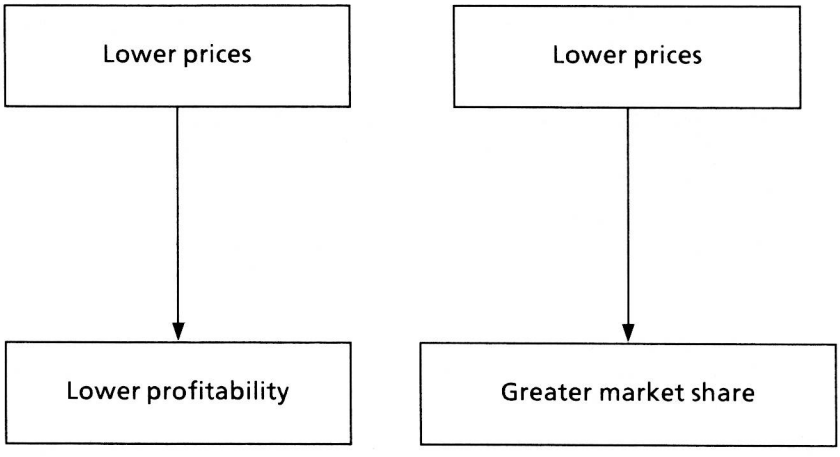


Figure 1

Changing the price of a product can affect both its profitability and market share in conflicting ways, as shown in Figure 1. Market research and internal cost analysis can help model the effects of lower prices on profitability and market share. The sales and financial divisions can have very different views on the subject. They need to organize their respective positions including counterarguments to opposing views. The cost-effectiveness of generating new information by market research and new product development can be considered by using due process.

All this activity takes place within a context that sets the time frame for the decision-making process. Sometimes the time frames can be very short, and, at the same time, the decision could be very important to the organization. Consider the sudden appearance of a new product that is drastically undercutting prices and demands a quick decision as to whether or not to cut prices. It is extremely common for a "case" to occur in due process that has to be settled promptly but has implications for more general issues. A company may develop a general vacation policy because a request by a particular employee for certain vacation privileges has to be granted or refused [13]. Due process takes place within action-taking and decision-making situations. It occurs at a particular place and time within a community of actors (both human and computer) that communicate with one another in a historical context involving information gathering, discussion, and debate.

The communications involved in due process can be analyzed along the following dimensions: