Simulating Organizations

Computational Models of Institutions and Groups

Foreword by Michael D. Cohen

AAAI PRESS / THE MIT PRESS
Menlo Park, California, Cambridge, Massachusetts, London, England

Simulating Organizations

Computational Models of Institutions and Groups

Foreword by Michael D. Cohen

Edited by Michael J. Prietula, Kathleen M. Carley, and Les Gasser

AAAI PRESS / THE MIT PRESS
Menlo Park, California, Cambridge, Massachusetts, London, England

445 Burgess Drive
Menlo Park, CA 94025

Copublished and distributed by The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts and London, England.

Library of Congress Cataloging-in-Publication Data
Simulating organizations : computational models of institutions and groups / edited by Michael Prietula, Kathleen Carley, and Les Gasser ; foreword by Michael D. Cohen.
p. cm.
Includes bibliographical references and index.
ISBN 0-262-66108-x (paperback : alk. paper)
1. Organization. 2. Organizational sociology—Mathematical models. 3. Artificial intelligence. I. Prietula, Michael J. II. Carley, Kathleen M. III. Gasser, Leslie George, 1949– .
HD31.S565 1998
302.3'5'011—dc21 97-51313
CIP

Printed on acid-free paper in the United States of America.

To those who have intellectually
inspired this new field
through their research and their spirit of
interdisciplinary conversation:

Richard Cyert, James March,
Allen Newell, and Herb Simon.

Contents

Foreword

Michael D. Cohen

This volume adds forceful new evidence that computational modeling of organization is experiencing a sudden burst of new vitality and maturation. There has been an immense gain in the sophistication of the computational tools employed, and—more importantly—in the integration of those tools with fundamental concepts derived from social science studies of human organizations.

The sporadic growth history of this area is actually rather puzzling. Organizational issues were among the first to be attacked when computational modeling developed in the 1950s around the work of Herbert Simon and his colleagues at what was then Carnegie Tech. By 1963 we had "a behavioral theory of the firm," with half a dozen interesting organizational models created by Cyert and March and their numerous colleagues and students. From that point on, however, the rate of advance was rather slow. While there have been, of course, some notable additions to the repertoire of interesting organizational models, nothing has occurred that is comparable to the explosion of computational modeling of cognition.

There are a number of candidate explanations of the slow growth of organizational modeling relative to cognitive modeling. Among them are (1) high costs of acquiring organizational data to which models can be compared (relative to laboratory cognitive data); (2) lower formalization of established theories in the organizations field; (3) low exposure to computing among practicing organization theorists; (4) poor intuitive match of process control structures in traditional computing languages to naturally occurring organizational processes.

All of these factors may be improving now. For example, increasing computerization of the workplace is making data on organizational activities far more available to would-be simulators than at earlier times. But it is recent changes with respect to the fourth item on the list that seem to me to deserve special notice.

One major development on this front has been the wide diffusion of object-

~~orientation in the software community~~. Instead of controlling processes in organizational models using iterative loops and conditional branching, it is now natural to represent processes as activation of object methods modulated by message passing. Objects provide a highly natural way of implementing model agents who have specialized capabilities and subtle, implicit, networks of interaction. This is a far more congenial framework within which to express intuitions about organizational processes.

The "organization-friendly" character of object-orientation is hardly an accident, of course. Alan Kay's early writing about Smalltalk, for example, made explicit use of analogies to organizational phenomena characterizing the design of the new language (1977).

A second development is the growing interest in organizational questions shown by researchers in the field of distributed artificial intelligence, who have recognized that distributed computers an programs must deal with many issues that are profoundly similar to those facing human organizations. (Here one should acknowledge early work by Carl Hewitt [1977], Victor Lesser [Durfee, Lesser and Corkill 1987], and the Hayes-Roths [1978].) Again the result has been a series of new ways of thinking about processes that are highly congenial to expressing intuitions about human organizations.

Both of these developments trace out an intriguing, round-about, path in which ideas about human organizations serve as analogies that inspire developments in computer science which in turn become tools useful for building computational models of organizations.

The influence of these developments can be seen clearly in the contents of this volume. Object-oriented methodology now underlies many of the systems being developed, and ideas from the field of distributed artificial intelligence are directly in evidence in many of the chapters.

The results are at last beginning to look highly promising. The increased interest in formal tools for organization theorists I clearly signaled by the emergence of a new journal on the topic: *Computational and Mathematical Organization Theory.* While there is still a great deal of hard work to be done, we can begin to imagine a day when theorists of organizations will routinely state theories and derive their implications using computational tools that are easily comprehended and widely shared - though sharing the tools will only happen if the field commits itself to the special efforts it requires.

The three editors of the volume have played a central role as catalysts of these developments. All of us interested in making computation a viable intellectual tool of organization theory are in their debt.

Ann Arbor, Michigan

Acknowledgements

The direct seeds of this book reside unequivocally in the AAAI Workshop on Artificial Intelligence and Theories of Groups and Organizations, put together by Kathleen Carley, Les Gasser, David King, and Mike Prietula. The presentations, discussions, e-mails, and collaborations emerging from that workshop and from the annual Computational and Mathematical Organization Theory Workshop associated with the INFORMS meetings have helped to energize a direction of theory and research reflected in this volume. The ultimate value of this book stems from the contributions of the authors. Significantly, the expertise reflected in these chapters is spread across a broad spectrum of disciplines: computer sciences, organization theory, sociology, operations research, management science, psychology, information systems, and beyond. It is not easy to get a group as diverse as this to speak to, and not by, each other. We believe such a conversation is well under way. Out of such activity emerges fascinating discussions. Out of such discussions science progresses. We thank them.

We also thank the American Association of Artificial Intelligence for taking a risk on this workshop, and the AAAI Press for taking a risk on this volume. Specifically, we thank Kenneth Ford of NASA and the University of West Florida (our AAAI Press editor) and Mike Hamilton of AAAI Press.

We also thank our colleagues for their insightful comments and our families for their support. Finally, we thank Kathy Murphy at the Fisher School of Accounting of the University of Florida for helping shepherd the process of accumulating chapters and other desiderata from us all. This one is done....

A Computational Approach to Organizations and Organizing

Michael J. Prietula, Kathleen M. Carley, and Lès Gasser

Organizations and groups permeate our lives, and their influence is growing. Indeed, each of us is so familiar with being in or being affected by organizations that we may tend to think we know how they work—as Giddens has pointed out, people must be in some sense fairly good social theorists just to get along in daily life (Giddens 1984). However, naive theories about organizations are often wrong, and it can be dangerous to rely upon them. Organizations are large complex highly volatile systems whose behavior affects and is affected by the environment in which they operate. Individuals can, but do not always, affect how organizations operate. The character of information technology that is available can affect organizational processes, and so on. For effective guidance in organization design, management, and operation, stronger principles, and the methods for generating and deriving them are needed.

Clearly, organizational behavior is affected by a large number of interacting factors. Organizational theory can be characterized as the study of how this multiplicity of factors combine to influence the behavior of organizations and the people and technologies comprising them. Some would even argue that across such factors there are general principles of organizing that are true for all groups and organizations regardless of whether the actors within the organizations are human or artificial. As such, organizational theory can also be characterized as the search for these general principles.

Computational organization theory (COT) is the study of organizations as computational entities. COT researchers view organizations as inherently computational because they are complex information processing systems. An organization as a computational system is composed of multiple distributed "agents" that exhibit organizational properties (such as the need to act collectively and struggles for power), are assigned tasks, technology, and resources, and across which knowledge, skills, and communicative capabilities are distributed. Computational organization theory focuses on understanding the general factors and nonlinear dynamics that affect individual and organizational behavior with a special attention on decision making, learning, and adaptation (Carley 1995). In computational organization models, information, personnel, decision responsibility, tasks, resources, and opportunity are distributed geographically, temporally, or structurally within, and sometimes between, organizations.

Computational organization theories are meso-level theories of organizations. An idealized characterization of organizational studies partitions research into two types. A macro perspective, (conventional organization theory) treats the complexities of individual behavior as largely irrelevant, or simplifies variety across individuals into an ideal individual type. A micro (i.e., organizational behavior) perspective focuses on the individual and often minimizes the constraints on action afforded by tasks and social-situations, and institutions. Computational organization theories are typically meso-level in the sense that they seek to explain and predict macro-level behavior, such as overall organizational performance, from micro-level actions, such as the interaction among agents, each of which are "cognitively" limited.

Computational analysis can help us to grasp some fundamentals of human information processing behavior (Simon 1973). Thus, computational modeling can be valuable for the study of organizations as collections of intelligent agents. Further, compared with experiments using human subjects, computational models are generally less noisy, easier to control, more flexible, more objective, and can be used to examine a larger variety of factors within less time. Computational analysis also makes it possible to determine whether or not important nonlinearities in behavior emerge as scope conditions are extended. For example, computational models may be larger (e.g., more agents) or may cover a longer period (more tasks) than can be covered in a human laboratory experiment. Ostrom (1988) argues that computer simulation offers a third symbol system in studying social science (with natural language and mathematics being the first two symbol systems) and notes that "computer simulation offers a substantial advantage to social psychologists attempting to develop theories of complex and interdependent social phenomena." This same advantage is true for organizational theorists, perhaps even more so given the nature of organizations.

Organizations are complex, dynamic, nonlinear adaptive and evolving systems. Organizational behavior results from interactions among a variety of adaptive agents (both human and artificial), emergent structuration in response to non-linear processes, and detailed interactions among a large number of other factors. As such, organizations, or at least many aspects of them, are poor candidates for analytical models. Thus, computational analysis becomes an invaluable tool for theory building as it enables the researcher to generate a set of theoretical propositions from basic principles even when there are complex in-

teractions among the relevant factors. The computational model can be thought of as a hypothesis generator that generates a set of propositions that can be more easily assured precise and internally consistent. In addition, computational models allow researchers to demonstrate proof of concept; i.e., to demonstrate whether or not a set of factors that are completely modelable are capable by themselves of generating certain phenomena. Used in this way, computational models can be used to show the potential legitimacy of various theoretical claims in organization science. Finally, using computational models it is often possible to determine the logical consistency of a set of propositions and the extent to which theoretical conclusions actually follow from the underlying assumptions. In some areas, formal logic plays this role. However, currently, multi-agent logics are not sufficiently developed to play this role.

Computational techniques for theorizing about organizations and organizing are invaluable tools for both the researcher and the manager. Most of the work in this book focuses on the scientific rather than the administrative side of this enterprise. Nevertheless, some hints (Chapters 1, 7, 8 and 10) are given as to how these tools might ultimately be of use to managers. Essentially, these computational models, once appropriately validated, can be used as decision aids to help the manager think through the impacts of new information technologies, organizational redesigns, or the reengineering of tasks. Clearly the models are not quite there yet, but that is one of the ultimate objectives.

Given such a computational model, three general evaluation criteria can be brought to bear: sufficiency testing, process testing, and component analysis. Sufficiency testing is the weakest form of validation and focuses entirely on the outcome of the behavior. In essence, it states that a computational model should at least be able to produce the behavior it purports to explain and is similar to the achievement criterion in cognitive modeling (Simon and Baylor 1966). Chapters 1, 2, 3, 4, 5, 6, 9 and 10 meet this criteria. Process testing makes a stronger statement, and is similar to the process criterion in cognitive modeling (Simon and Baylor 1966). This test goes beyond showing that an unspecified set of mechanisms produces a certain result by demonstrating that particular mechanisms (or knowledge) can produce the behavior. In this test, comparisons are made at some level of abstraction between the model and a referent (a proposed gold standard). Chapters 1, 2, 3, 4 and 9 are movements in this direction. Component analysis examines specific contributions of the mechanisms or knowledge represented in the reasoning events. The relative impact of different components are contrasted. Chapters 1, 3, 4 and 9 have this characteristic. Computational theorists can build theories of organizations and organizing by hypothesizing about the behavior of their models, testing these hypotheses through a series of virtual experiments, generating a new consistent set of hypotheses derived from these computational results, and then testing these hypotheses with "real" data. The chapters that move farthest in this direction are Chapters 4 and 9.

Hypothesizing about the behavior of computational models may seem trivial and obvious, as the program components are well defined; however, the complexities of today's simulation systems inhibit accurate a priori specifications of their behavior. Even with simple models, such as Team-Soar (Chapter 3), the model of cooperation (Chapter 5), and the original garbage can model (Cohen, March, and Olsen 1972), new findings emerged that were not hypothesized given the program components and new insights were gained into organizational performance. Though predictions may be made concerning the behavior of such models, tests of actual program performance must be made to verify them. To this end, researchers often use these models to run virtual experiments, collecting data that is then analyzed graphically and statistically.

Within the field of computational organization theory, computational analysis is used to develop a better understanding of the fundamental principles of organizing multiple information processing agents and the nature of organizations as computational entities. Research in this area has two main foci. The first foci is has to do with building new concepts, theories, and knowledge about organizing and organization. Most of the chapters have this foci. The second foci has to do with developing tools and procedures for the validation. Chapters 7, 8 and 11 have this foci.

This book is divided into four sections. Each of these sections corresponds to a major area in which there is on-going research. Section One is titled "Organizations as Multi-Agent Systems." Human (or human-derivative) organizations, like games, are "artificial" in the sense that they are crafted by humans (Simon 1981). Organizational behavior emerges both from the artificial construct that constrains individual interactions and the natural limits on human (or agent) behavior. This type of emergent behavior is seen in the results presented in Section One. All three chapters in this section draw heavily on work in artificial intelligence. Unlike most games, however, organizations are highly volatile with no specifiable (or perhaps predictable) equilibria. Indeed, within organizations it is the norm that the rules change, the players change, and the situations change. This volatility is due in large part to the agents which comprise them and the way in which the agents interact. Hence, within organizations, the form of the rules and procedures depends on the agents and their personal history as they respond to the changing environment. The first chapter (Chapter 1, "WebBots, Trust, and Organizational Science"), by Carley and Prietula, describes an experiment in which a strong model of artificial intelligence called Soar (Laird, Newell, and Rosenbloom 1987) is used to explore the significant issue of trust between intelligent agents. Not only is this a unique direction of inquiry, but the study produces quantifiable data on deliberation and communication derived directly from the theoretical stance articulated in code. In the second chapter of this section (Chapter 2, "Team-Soar: A Model for Team Decision Making"), Kang, Waisel, and Wallace also incorporate a Soar approach. In this chapter, the authors model a naval command and control team tasked with making critical decisions regarding the hostility of an incoming aircraft. Events such as those occurring with the U.S.S. Stark and the U.S.S. Vincennes have been traced to dysfunctional team behaviors. Their approach is to simulate and analyze possible sources of team dysfunctionality to improve team decision making behaviors. In the third chapter in Section One (Chapter 3, "Designing Organizations for Computational Agents"), So and Durfee describe a framework for understanding organizational design design for computational agents, use that framework for analyzing the expected performance of a class of organizations, and describe how the analyses can be applied to predict performance for a distributed information gathering task. An interesting component of this chapter is the concept of organizations re-designing themselves, addressing an emerging critical problem in network administration.

Section Two ("Organizations and External Conditions") explores the relationship between organizational action, agent behavior, and environmental volatility. Part of the volatility within organizations comes from the advent of new technologies. Further, organizations often try to employ technologies to curb the impact of other forms of organizational volatility. Lin (Chapter 4, "The Choice between Accuracy and Errors: A Contingency Analysis of External Conditions and Organizational Decision Making Performance") uses a version of the radar task described in Chapter 2, but focuses on exploring the relationship between organizational performance, organizational designs, and environmental properties. It appears that the reliability of an organization resides in the fit between the design and the task the choice of design becomes a strategic decision between what type of errors the organization is willing to accept or minimize. Huberman and Glance (Chapter 5, "Fluctuating Efforts and Sustainable Cooperation") show that when individuals confronted with a social dilemma contribute to the common good with an effort that fluctuates in time, they can generate an average utility to the group that decreases in time. This paradoxical behavior takes place in spite of the fact that typically individuals are found to be contributing at any one time. This phenomenon is the result of an intermittency effect, whereby unlikely bursts of defection determine the average behavior of the group. Thus, typical behavior of individuals comprising a group, can be inconsistent with a groups average properties. In the final chapter in this section (Chapter 6, "Task Environment Centered Simulation"), Decker describes the TÆMS framework (Task Analysis, Environment Modeling, and Simulation) to model and simulate complex, computationally intensive task environments as multiple levels of abstraction, and from multiple viewpoints. TÆMS is a tool for building and testing computation theories of coordination. This framework permits researchers to mathematically analyze (when possible) and quantitative simulate (when necessary) the behavior of multi-agent systems with respect to interesting characteristics of their task environment. As such, it is a testbed for exploring centralized, parallel, or distributed control algorithms, negotiation strategies, and organizational designs. To illustrate TÆMS, Decker investigates a simple question: Is there a difference between performance due to either the choice of organizational structure or the decomposibility of the technology?

The chapters in Section Three ("Organizations and Information Technology") address issues of technology, but within the realm of information technology and information systems. Most computational models of organizations do not consider the role of information technology, Thus, the chapters in Section Three represent initial forays into the how to think formally about the role of information technology in organizations. Fox, Barbuceanu and Lin (Chapter 7, "An Organizational Ontology for Enterprise Modeling") begin to address the next generation of Enterprise Model. Specifically, they propose that the next generation be a common sense enterprise model, which possess the capability to deduce answers to queries requiring relatively shallow knowledge of the domain. Thus, a key component of future information systems is an enterprise model that goes well-beyond the capabilities of current database or enterprise systems. Key to the articulation of such a model are the fundamental ontologies upon which the model is defined. The authors present a discussion of their approach to defining ontologies and ontological competence in their pursuit of the next generation enterprise model. In Chapter 8 ("Modeling, Simulating, and Enacting Complex Organizational Processes: A Life Cycle Approach"), Scacchi describes the approach and mechanisms to support the engineering of organizational processes throughout their life cycle. Organizations are, in part, defined by their processes. As events change (e.g., technology, tasks, environment) an organization may have to review and redefine its processes and process streams. Scacchi describes a knowledge-based computing environment, the articulator, that supports the defining and simulation of complex organizational processes. Kaplan and Carley (Chapter 9, "An Approach to Modeling Communication and Information Technology in Organizations") describe the communicating and information technology (COMIT) computational framework used to investigate information processing impacts of changing either the information technology or the communication structure on organizational performance. COMIT generates aggregate and detailed statistics on the number and duration of actions (e.g., communication, information lookup) and task completion quality. To illustrate, the authors describe a study in which levels of technology (high, low) are crossed with levels of experience and work structure (solo, collaborative). Their results suggest that technology, training, and organizational design can interact in complex ways to influence performance, and that computational approaches as COMIT can help reveal those complexities and their effects. In the final chapter of this section (Chapter 10, "Organizational Mnemonics: Exploring the Role of Information Technology in Collective Remembering and Forgetting"), Sandoe presents a conceptual model of organizational remembering and forgetting, and describes a simulation derived from the conceptual model. Sandoe argues that organizational remembering (and forgetting) occurs in three ways: an organization can remember (1) structurally, through the establishment

of rules, roles, policies; (2) mutually, through advisory relationships among its members, and (3) technologically, through the creation of physical or symbolic artifacts. Sandoe then conducts a study where three organizational forms (hierarchy, network, hub) are simulated and tested with respect to environmental turbulence, turnover, and cost.

In the concluding essay ("Validating and Docking: An Overview, Summary and Challenge"), Burton addresses the chapter contributions in three contexts. First, Burton discusses the chapters with respect to the important issue of validity in the context of a framework which summarizes a computational model along three dimensions: its purpose, its process, and the analysis of its results (Burton and Obel 1996). Second, Burton categorizes the chapters according to a scheme developed by Carley (1995) which situates a computational simulation with respect to its explanatory role, of which four are proposed: organizational design, organizational learning, organizations and information technology, and organizational evolution and change. Finally, Burton speculates on the contribution of the collective in the context of "docking" or aligning simulation models for comparative purposes (Axtell, Axelrod, Epstein and Cohen 1996).

Computational organizational theorists are trying to use computational techniques to develop a firm scientific base for the study of organizations. As noted, organizations are often complex, nonlinear, adaptive systems. The natural complexity of organizations is reflected in the fact that many of the existing models and theories of organization are vague, intuitive, and under-specified. The more explicit and well-defined these theories, the greater our ability to make scientific progress. Computational theorizing about organizations helps to achieve this. The chapters in this book contribute to this endeavor. These chapters are the outgrowth of the tremendous outpouring of work in this area in the second half of the twentieth century since Cyert and March's *A Behavioral Theory of the Firm* in 1963. Recent work in this area combines traditional organizational concerns with performance, design, and adaptation with technique and approaches informed by work in the area of distributed artificial intelligence (Bond and Gasser 1988, Gasser and Huhns 1989). However, the computational organization theories of today, unlike much of the early work in distributed artificial intelligence work, are often grounded in existing cognitive, knowledge-based, information-processing theories of individual behavior and information processing, institutional, population ecology, or other models of organizations. Computational organization theorists extend the work on individual behavior to the organizational level (e.g., Simon 1947). This combination and extension gives precision to the notion of bounded rationality by specifying the nature of the boundaries and the role of social and historical information in defining organizational action (Carley and Newell 1994, Carley and Prietula 1994). This book contributes to our understanding of both organizations and organizing and provides illustrations for how to conduct research in this area.

Simulating Organizations

SECTION ONE

Organizations as Multi-Agent Systems

I

WebBots, Trust, and Organizational Science

Kathleen Carley and Michael Prietula

WebBots are artificial creatures. Now, by "artificial" we do not mean that they do not exist, for they do. In fact, we built some. Yet, WebBots are neither biological nor mechanical creatures. WebBots are computer programs, but they are computer programs of a very special type. WebBots are programs that help their human counterpart(s) to achieve goals and solve problems. What is unique about WebBots is that they do much of their work on their own over webs of interconnected networks.

One of the major applications we see for WebBots is to be "intelligent explorers" on networks (including the Internet) for their human (i.e., corporate) counterparts. Thus an organization might have dozens, hundreds, or even thousands of corporate WebBots actively searching, communicating, traveling, and even reproducing over networks around the world for a wide variety of purposes. Simpler types of such creatures are being researched or even employed by firms such as AT&T, IBM, Apple Computer, Xerox, Microsoft, Hertz, Ford, and even the White House (Houlder 1994; Keller 1994). Although the specifics of any vibrant and emerging technology is extremely difficult to predict with certainty, the current trends in information technologies all point to a single, inescapable prediction: *the WebBots are coming!*

WebBots (or whatever you wish to call them) can take on a wide variety of forms. In this chapter, we will briefly mention some of these, but we are going to describe a different kind of WebBot. The WebBots we describe have very unique properties. To get these unique properties, we propose a very unique architecture for WebBots. The interesting elements of the proposed architecture are that it provides a fundamental framework for general WebBot intelligence and permits a unique set of mechanisms for defining, measuring, and sharing corporate learning, memory, and knowledge.

In this chapter, we first offer a brief look at WebBotlike programs. This is not a new concept; rather, we are building our approach on a long stream of incremental research from several different perspectives. We then present an architecture that can realize a specific type of intelligent WebBot agent, an agent that can reason and communicate with other WebBots. Since a central point of this chapter is exploring the social aspects of WebBots, we next describe a computational experiment in which we simulate an organization of WebBots. In this experiment, we assign tasks to a small group of WebBots, adjusting and experimenting with a particularly important aspect of WebBot interaction—trust and forgiveness in information exchange. We conclude with a speculative discussion on the implications of applying an organizational science perspective to an organization of WebBots. Should we begin to define an organizational science of WebBots? Is it possible? We argue that this is not only possible (though certainly not easy), but essential, in order to successfully assimilate such technology (or technologies) into the corporate environment. We propose that the foundations for studying WebBot organizational science have already been formed.

About WebBots

In one sense, this chapter is quite speculative. The WebBot creatures of the type we are addressing are not quite ready for prime time—but close. They are in the digital Catskills of information technology: corporate and university laboratories. The WebBots in our world are related to digital creatures that go by many names, depending on their particular capabilities, or even on the particular laboratory or organization where they are being created. There is no commonly accepted definition for the term WebBot; however, the concept of a WebBot has emerged at various times over the past decades in both formal and informal settings.

We have witnessed lively discussions at our research conferences over who invented or used what term first, who actually constructed the first (fill in your term here), and what stream of research was actually most responsible for the current perspective(s). It is perhaps easiest to think of WebBots as belonging to a large family of computational architectures that differ on various dimensions of form or function but possess a general family resemblance. Recall the wide variety of droids depicted in the *Star Wars* trilogy? Similarly, we can imagine a wide variety of WebBotlike "digital analogs."

The research lineage of approaches such as ours can be traced to several sources, with perhaps the general theme mostly related to Negroponte (1970), though the concept of an "intelligent agent" has been around in thought, if not in form, since at least the 1960s. Several fascinating perspectives abound in the field, such as knowbots (Kahn and Cerf 1988), softbots (Etzioni, Lesh, and Segal 1994), varieties of software agents (Genesereth and Ketchpel 1994; Greif 1994, Guha and Lenat 1994; Keller 1994), apprentices (Dent et al. 1992), intelligent

agents (King 1995; Roesler and Hawkins 1994), distributed intelligent agents (Hayes—Roth 1990; Rosenschein 1992), and a host of similar creatures in the distributed artificial intelligence literature (e.g., Gasser and Huhns 1989; Sycara et al. in press). Furthermore, there is an important convergence of several aspects of these themes that are opportunistically being applied to the Internet (e.g., Cheong 1996).

Additionally, this work draws from the work in computational organization theory (Carley and Prietula 1994; Carley 1995) and addresses the issue of organizational design. Like the work on intelligent agents, rigorous research using computational methods to explore issues of organizational design dates back to the 1960s (Cyert and March 1963). This work uses computational techniques to examine how organizations of intelligent, and often adaptive, agents should be coordinated (Masuch and LaPotin 1989; Levitt et al. 1994; Prietula and Carley 1994; Lin and Carley in press).

The options for organizations currently range from purchasing available application—specific software (e.g., generally for information retrieval, data mining, web mining, or news filtering) to building their own agents within a particular technology using a form of scripting language, such as General Magiclike Telescripts (White 1994), crafting their own proprietary systems for specific purposes, or hiring a firm to build or apply agent technology (e.g., Comshare or Andersen Consulting's Enterprise Intelligent Systems group). Additionally, research projects are underway to provide general agent design languages and open architectures (e.g., Cohen et al. 1994; Shoham 1993). However, one must be careful to understand the "granularity and form" of the architectures and languages. For example, there are large differences between building agents from enhanced components of a programming facility, like a predefined object package within C++, and building agents from a much higher architectural level, such as those often afforded by the distributed artificial intelligence approaches (Bond and Gasser 1988). One goal of this chapter is to add an approach to this last list of efforts that brings a quite different perspective on agent design.

Our collective role in this chapter is not one of historian; consequently, we are permitted to exploit the available degrees of freedom afforded by this claim to offer our own interpretation and work from there. Our first interpretation is as follows:

> A *WebBot* is a computer program that operates autonomously to accomplish a task or set of tasks as an intellectual advisor and assistant to a human counterpart.

WebBots of the sort we are describing, then, are presented with goals (explicit or implicit) and turned loose within a system or a network (or within many networks) to accomplish some electronic type tasks. We might tell a *WebBot* (let us ignore issues of natural language communication) to perform the following tasks:

- Monitor intranet work events across a set of terminals and provide a report on Monday of the list of people who ... (an intranet monitoring WebBot).
- Engage a search every day for new additions to telecommunications home pages that ... (a watcher WebBot] and then add them to a resource list ... [a fetch WebBot).
- Keep an up-to-date list on the references of recent hearings on the cable industry where ... (a fetch WebBot) and then go get the text, graphics, or audio-feeds if they are available on-line ... (a fetch WebBot).
- Watch the corporate knowledge bases for any new additions regarding audit and technology issues in the health care industry ... (a corporate watcher WebBot) ... and send them e-mail requesting a copy of their knowledge report where ... (a corporate communicator WebBot).
- Go out and scan employees' disks checking for viruses and report back the results ... (a virus checking intranet WebBot).
- Go out and scan the Web to detect what sites are pointing to your site, then send and collect an information survey over the Web to those sites ... (a tracer WebBot).

In the first example, a network administrator might want to check logins, idle time, and workloads for the corporate intranet. For the second, a WebBot will access Internet search resources (e.g., search agents programming other search agents) every day to determine if new home pages (i.e., World Wide Web sites) have been added for a particular topic. The third example addresses what might have been called "library research," where the electronic card catalogs are periodically monitored and text resources are obtained when appropriate. The fourth example depicts a corporation that has set up a type of knowledge base containing experiences (e.g., problems, solutions, and explanations) of client engagements or internal projects. The next example illustrates a WebBot that has virus-monitoring responsibility for software on employees' disks on their company's intranet. In the final example, a WebBot explores to see what links are pointing to a corporate web resource and then engages an automated web-based survey sequence.

The set of WebBots available to their human counterparts are defined in terms of the business processes they are to accomplish at one end with the embedded procedural knowledge of how to accomplish them at the other. The embedded procedural knowledge, as one may surmise, must include aspects of the digital image of the relevant corporate resources that enable it to perform the tasks at hand. WebBots must know about (or have the capacity to figure out) the corporate environment within which they reside.

Four general observations can be made on the requirements of the WebBots we are describing that differentiate them from their fellow agents. First, if an organization (we will assume that organizations may be some of the major investors in such creatures) engages a set of WebBots as assistants for their employees, and employees in organizations often require interaction with other employees, then an obvious observation can be made:

Some WebBots' tasks will necessitate interacting with other WebBots.

Second, business processes themselves can be rather complex and, in fact, constructed (defined in terms) of simpler business processes. That is, business processes will be specified in terms of other business processes. Therefore, a second observation can be made:

Some of the WebBots' tasks will consist of sequences of, or even hierarchically defined levels of, business processes.

Third, the WebBots must be able to engage a sufficient amount of deliberation to reasonably deal with the vague, ambiguous, and uncertain environment encountered in the attempted execution of the business processes within a dynamic corporate setting. That is,

WebBots should have the capacity to reason about and learn from their actions.

Finally, there is a direct implication of two types of potential communication in this simple structure: human–WebBot, WebBot–WebBot. Human–WebBot communication is, essentially, a fundamental question of human-computer interface development. This is a critical element in systems design, since from the perspective of the user, the interface *is* the system. Good systems design teams understand and address this issue. Tell the WebBot what to do (e.g., by means of some scripting language, mouse clicks, voice commands), and have the WebBot report back to the user in the most appropriate (or desired) manner available.

What is interesting are the implications of the WebBot-to-WebBot communications. What might they say to each other? How should they say it? In part, these are also system design questions whose answers depend on the nature of the problems to be addressed and by the particular agent technology. One could imagine that this would involve communication for requesting and providing information for task-related purposes, including coordination. Yet we are envisioning communication also occurring on a fundamentally different level. Because of the particular architecture out of which our WebBots are constructed, they are quite capable of direct knowledge exchange (Zhu, Prietula, and Hsu in press).

The fundamental structures that compose their knowledge can be shared among WebBot agents. Agents do not need to be actually built solely of this particular architecture; rather, they simply have to include this architecture as a subcomponent of their architecture. Thus, the inclusion of a direct knowledge exchange (DKE) capability renders an agent that is DKE-enabled. DKE-enabled WebBots have a remarkable capability:

WebBots can directly share the knowledge they accumulate in the performance of their tasks.

We are describing WebBots that have a fundamental intelligence and that are able to reason about their task and their environment, which includes the behavior of other WebBots. Furthermore, each WebBot is capable of educating any other WebBot by directly communicating its knowledge. There could be, of course, much simpler and specialized forms of WebBots, but in this chapter we are addressing a more ambitious species—a species capable of rudimentary problem solving and learning (thus changing its behavior) in the service of a goal.

Thus we can offer a modestly improved definition:

A *WebBot* is a computer program that operates autonomously and intelligently to accomplish a task or set of tasks as an intellectual advisor and assistant to a human counterpart and other WebBots.

A Computational Study of WebBots

While we are also envisioning many WebBots interacting with each other, we are also describing an embedded electronic population of WebBots interacting within an explicit (or implicit) WebBot organizational structure. On one level, as defined, there is a distinct set of business processes that are defined in terms of the WebBots that instantiate them. On another level, there is a new and different organization of agents interacting in a world influenced by, but not populated with, humans. Our primary interest in WebBots is not in what they might be able to achieve, but rather in the implications of WebBot social interactions with each other in this electronic organizational subculture.

What can we begin to say about an organizational substructure of intelligent WebBot agents? We conducted a computational exploration of a simple organizational situation based on the type of WebBot we have described. We asked the following question: to what extent does WebBot honesty affect individual and collective organizational behavior?

Organizations are composed of individual agents whose collective activity defines the "behavior" of an organization. Similarly, the individual decisions of WebBots affect the behavior of the entire WebBot collective and, consequently, the organization's general behavior. From this perspective, certain types of WebBots' social behaviors (i.e., interacting with each other within the organization in the execution of their tasks) and their effects on organizational performance (individual and collective properties of their behavior) are explored. The WebBot's social behavior is defined by a set of behavioral predispositions they have reflecting specific "rules of social engagement" defining a rudimentary social cognition component of WebBot deliberation. The rules we are investigating are concerned with honesty and benevolence judgments within the context of a socially situated task.

The simulation described in this chapter is unique. It reflects a "theory on a theory" as the fundamental WebBot architecture is itself a theory of individual intelligence, called *Soar*. With this theory of individual WebBot architecture forming the basis for creating WebBots, an assemblage of WebBots are linked together, interacting in a social environment in their performance of a task. Two types of knowledge are encoded in each WebBot: task-specific knowledge, enabling the task to be accomplished, as well as social-interaction knowledge, reflecting the social cognition rules of social engagement for the properties investigated (honesty, benevolence). By situating these WebBots in an organizational task permitting social interaction, a small organizational unit is defined. How each WebBot behaves is based on the nature of the goals, the knowledge to work on those goals, and how the task unfolds in the context of other WebBots.

In cognitive science terms, each WebBot defines its own problem space reflecting critically perceived aspects of the task environment (Newell and Simon 1972). Each WebBot's problem space also contains models of other encountered WebBots and their behaviors, for these other WebBots are also components of the task environment. It is from models of each other's behaviors that decisions are made regarding interactions, and it is the nature of these social interactions (i.e., interaction decisions) that define collective organizational behavior. Yet, each WebBot constructs its task-specific social reality and performs its problem-solving behaviors in the same manner and with the same underlying architecture. It is entirely knowledge-based, with a single set of mechanisms operating under a unified approach to defining all aspects of deliberate problem solving—Soar.

The Soar Architecture

Soar is a symbol-oriented computational architecture for general intelligence (Laird, Newell, and Rosenbloom 1987; Newell 1990). In the Soar architecture, tasks are represented as search, through the application of operators to monitor and manipulate symbol structures in working memory, within problem spaces to achieve goals (existence of a particular symbol structure). Knowledge in Soar is represented as *if* <antecedent> *then* <consequent> productions of the following general form:

if <a structure is found in working memory>
then <propose a change in working memory structures>

Changes in working memory are proposed until sufficient evidence has accumulated to support a specific change in memory structure. This is accomplished through a preference system embedded in the architecture itself. Details can be found in Laird et al. (1990).

If Soar cannot directly and unambiguously achieve a goal with its current knowledge base, the architecture automatically generates an impasse, causing a new goal, called a subgoal, to be created and addressed—the resolution of the impasse. This, in turn, may cause further impasses.

Essentially using a depth-first, look-ahead search, the subgoaling process proposes envisioned subproblem spaces that correspond to each of the available actions. Soar traces the decision trees that would unfold if each possible subproblem space was in fact chosen, and then evaluates the outcomes of each alternative action in terms of the current goal. Since both the production memory and the working memory are always accessible and the same problem solving mechanisms apply in any subproblem space, the full problem solving power of Soar is available to be brought to bear for each subproblem.

Once Soar resolves a subgoal, an analysis of the working memory elements (symbol structures) leading to the resolution of the subgoal is made, linking them to the eventual working memory elements of the resolution. From this, new long-term productions are created, called *chunks*, which represent accumulated knowledge to directly resolve the state causing the impasse if it is again encountered, thus avoiding subgoal deliberations. Soar has learned.

All decisions in Soar are made in a two-phase decision cycle. During the first phase, called the *elaboration* phase, any and all productions whose antecedent conditions have been satisfied fire, proposing their consequent contributions to working memory. As working memory structures change, different productions may become able to fire, while others may lose support. This process continues until no more productions can fire. Thus, all productions effectively fire in parallel. The decision cycle in Soar represents the fundamental metric for deliberation: the more decision cycles, the more cognitive effort.

Soar-based WebBots

With much of the fundamental effort of intelligent deliberation a component of the architecture, WebBots can be created from Soar by adding task-specific knowledge. The task for the study consisted of a basic simulation of Soar agents making judgments over a network regarding tasks. The network task is interpreted as follows: Soar agents are WebBots that search out electronic information resources over the network. A WebBot receives a net resource target to find (e.g., a Web site that contains some desirable information resources), then proceeds to search for the Web site containing the target. An option available to the WebBot is to send out an electronic message to other WebBots to see if they have encountered this requested resource and can tell where it might be located.

In modeling the access to the various resources, we imposed a sequence of processes required to access them. For example, one might imagine logging in to a mailbox to get an assignment, sending out the e-mail, getting a response, evaluating the response, logging in, and accessing/searching through a variety of interim Web pages or pointers to finally access the resource, which may or may not be at a particular Web site.

The WebBots themselves were provided with the following rudimentary characteristics:

- *Communication*—ability to ask other WebBots if they have seen a resource on the net, and answer other WebBots questions regarding the same issue.

- *Location memory*—ability to recall what resources it has seen when it has visited a net site
- *Social memory*—ability to recall its interactions with other WebBots regarding requests for net site information (i.e., was it correct or not)
- *Rules of social engagement*—when asked for information, consistent truth (trustworthy WebBots), or consistent misleading (untrustworthy WebBots)
- *Social judgment*—a scoring scheme for judging whether a WebBot was trustworthy based on social memory of past communications and engagements.

For this study, five different organizational sizes were examined (one through five WebBots), and each organization was homogeneous with respect to one of two conditions. In the first condition, the WebBots were all honest—they attempted to respond accurately to questions of possible Web resource locations from either their memory or their current perspective (i.e., directly observing it from their location). In the second condition, all WebBot organizational agents attempted to deceive other WebBots when they received requests for Web locations.

In both conditions, all WebBots engaged in social judgments. Consisting of ratings of trustworthiness based on the veracity (or lack thereof) of each of the other WebBots' past communications. WebBots incorporate three levels of trustworthiness of an information source: trustworthy (location of a Web site was correct), possibly untrustworthy (location of last requested Web site was incorrect), and untrustworthy (location of last two requested Web sites were incorrect). Thus, if at some time in the past WebBot *X* told WebBot *Z* that net resource *A* was at net site ξ, and in acting on this information, WebBot *Z* finds that net resource *A* is indeed at net site ξ, then WebBot *Z*'s "opinion" of WebBot *X* would support a social judgment of trustworthiness. If WebBot *Z*'s prior opinion of WebBot *X* was "possibly untrustworthy," then that opinion would be upgraded to trustworthy.

On the other hand, if WebBot *Z* fails to find net resource *A* at net site ξ, then WebBot *Z* would downgrade its opinion of WebBot *X*. If WebBot *X* was previously considered "trustworthy" (the initial judgment values of all WebBots), then it would be downgraded to "possibly untrustworthy." Two consecutive incorrect messages from a given WebBot results in a judgment of "untrustworthy." Once some WebBot *Z* deems another WebBot *X* untrustworthy, WebBot *Z* automatically presumes that all further communication from WebBot *X* will also be incorrect. These WebBots are not forgiving.

The characterizations of WebBots are highly stylized but represent the facets of a broad range of behaviors found in functional and dysfunctional human agents (or human-created agents, like computer viruses and cracker codes). As WebBots can be programmed to perform in any particular manner, and as organizational members may compete, in part, in terms of the behaviors of their