OSMAS: A MULTI-AGENT TESTBED FOR EXPERIMENTING WITH ORGANIZATIONAL STRUCTURES

Trevor T. Moores
Department of Management Information Systems
University of Nevada Las Vegas
Las Vegas, NV 89154-6034, USA
E-mail: trevor.moores@unlv.edu

KEYWORDS
Organization Structure, Organization Design, Multi-Agent System.

ABSTRACT
A multi-agent testbed for experimenting with organizational structures is presented. The testbed is used to experiment with five organizational structures used in Tom Malone's (1987) study of the efficiency and flexibility of organizational structures. The experiments replace the ordinal distinctions made by Malone of the performance of each structure with a ratio measurement of the differences.

INTRODUCTION

An organization is defined here as an association of individuals brought together in order to perform a set of tasks beyond the physical or mental capabilities of a single individual (Cyert & March, 1963; March & Simon, 1958). One approach to studying the costs of organizational structures is the development of computer models. In particular, computational organization theory (Carley & Prietula, 1994; Prietula et al., 1998) studies organizations from the viewpoint that they are information processing entities, where multiple distributed "agents" have resources, are assigned tasks, and act collectively to accomplish those tasks.

Given the social nature of these agent interactions, computational organization models are often developed using Artificial Intelligence (AI) techniques (Cohen, 1986; Masuch & LaPotin, 1989). For instance, Lin & Carley (1997) use the agent-based DYCORP simulation engine to model time pressure and training on organizational performance, while a number of researchers have provided expert system-like support for organizational design by applying contingency theory (Baligh et al., 1996), or by defining micro-level definitions of work processes (Malone et al., 1999).

This paper will attempt to advance the field by proposing a multi-agent testbed for experimenting with organizational structures. The system is here called OSMAS, which stands for Organizational Structures Multi-Agent System The multi-agent system being proposed here is designed on the basis that the only difference between agents within the same organizational structure is the range of actions they can perform, while the only difference between structures is with whom the agents are in contact and can assign tasks to. These differences affect the range of micro-level behaviors that each agent is capable of performing, and consequently, give rise to the macro-level behavior that is each type of organizational structure.

To demonstrate the capabilities of this design, the results of two experiments using OSMAS will be presented. The experiments are based on the formulations put forward by Malone (1987) to analyze the efficiency and flexibility of four organizational structures, namely, the product hierarchy, functional hierarchy, centralized market, and decentralized market. The aim is to replace the ordinal distinctions made by Malone's mathematical formulation with a ratio measurement of the differences produced by OSMAS. The next section will outline the analysis put forward by Malone (1987) before discussing the logical and physical implementation of OSMAS agents. The experimental results are then presented before outlining areas of further research.

ORGANIZATIONAL PERFORMANCE

Malone (1987) provides an analysis of human organizations and markets by defining an organization as a series of (human or machine) processors, where each processor performs a task that takes a certain amount of time to complete. The coordination structure used to assign tasks to processors is therefore defined as the fundamental component that determines organizational structure. A coordination structure is defined as:

"... a pattern of decision-making and communication among a set of actors who perform tasks in order to achieve goals (p1319)."

Goals can also be defined as products. Malone focuses on two types of coordination structure, namely, hierarchies and markets. Hierarchies are a means of coordinating tasks when all the production is carried out within the organization. Markets, on the other hand, have some aspect of production carried out by suppliers. Malone proposes to analyze four examples of hierarchies and markets (see Figure 1) in terms of the

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

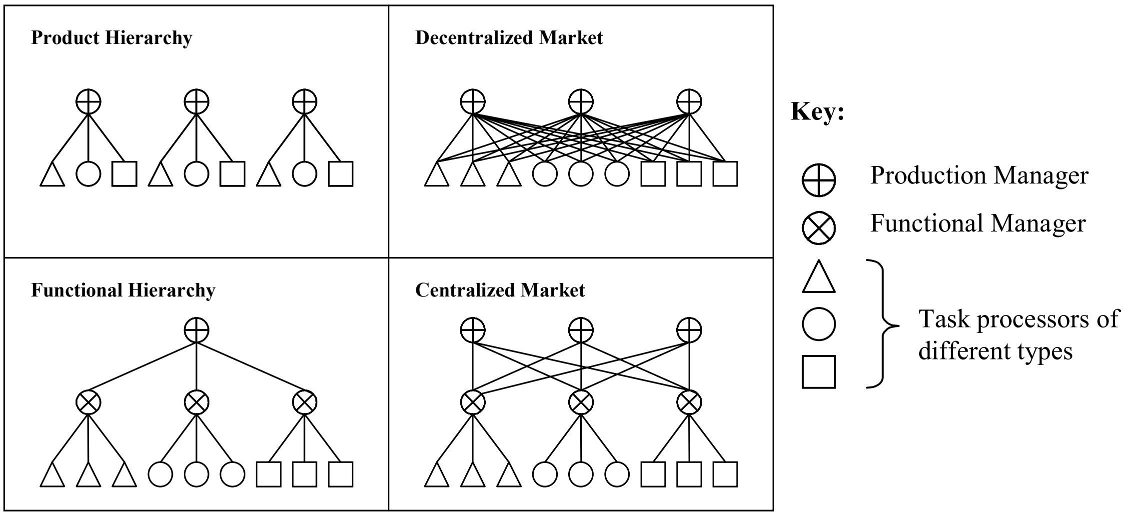


Figure 1: Organizational Structures as Markets or Hierarchies

relationship between processors and the number of messages required to assign a task.

The product hierarchy (PH), also known as the multi-divisional (Chandler, 1962) or M-form (Burton & Obel, 1984; Williamson, 1975), has product lines that are separated and have their own supply, manufacturing, sales and marketing departments. The product manager has control of all the different functions. Two messages are required to complete a task. One message is required to assign a task to a processor, and one to notify the production manager that the task is complete.

The functional hierarchy (FH), also known as the unitary or U-form (Burton & Obel, 1984; Williamson, 1975), has processors of the same type grouped together under one functional manager. An "executive office" assigns the task to a functional manager, who assigns the task intelligently to a processor. The functional manager must keep track of loads and processing capacity of the processors. Four messages are required to complete a task. One message is required for the production manager to assign the task to a functional manager, and one message is required for the functional manager to assign the task to a processor, with two messages required to notify that the task is complete (processor to functional manager, and functional manager to production manager).

The decentralized market (DM) has buyers in contact with all possible suppliers. It is assumed that suppliers are not randomly selected and the buyers have some way of selecting the "best" supplier by requesting bids from each supplier. Eight messages are required to complete a task. Six messages are involved in a "request for bids" phase, where a request is sent to each of the three processors of the same type, with three "bids" being returned. Two further messages are required to complete the task, one message from the production manager to assign the task to the chosen processor, and one message from the processor to notify the production manager that the task is complete.

The centralized market (CM) coordinates communication between buyers and suppliers using an intermediary layer of "brokers," thus substantially reducing the number of connections and messages required to complete a task compared to the decentralized market. Similar to the communication pattern in a functional hierarchy, four messages are required to complete a task. One message is required for the production manager to assign a task to a functional manager, and one message is required for the functional manager to assign the task to a processor, with two messages required to notify the respective managers that the task is complete (processor to functional manager, and functional manager to production manager).

Malone analyzes these structures in terms of their relative efficiency and flexibility, where efficiency is a function of production and coordination costs, while flexibility is a function of production and vulnerability costs (see Table 1). Production costs are proportional to the production capacity of the organization and the average delay in processing tasks. This is determined by the extent to which tasks processors are shared amongst products. Coordination costs are proportional to the number of communication links between agents and the number of messages required to assign a task. The number of links and the number of messages to assign a task is a function of the type of structure. Vulnerability costs are proportional to the costs incurred due to the failure of task processors or messages to complete their respective tasks. It is assumed that managers and processors will fail with a probability of greater than zero. The effect of such failures is determined by whether a single product line or all product lines are disrupted by the failure of one agent.

Table 1. Tradeoffs among Alternative Coordination Structures
(cf. Malone, 1987, pp1323)

Coordination Structure	**Evaluation Criteria**		
	Efficiency		Flexibility
	Production Costs	*Coordination Costs*	*Vulnerability Costs*
Product Hierarchy	H	L	H'
Functional Hierarchy	L	M-	H+
Centralized Market	L	M+	H-
Decentralized Market	L	H	L

NOTE: L=Low costs ("good"), M=Medium costs, H=High costs ("bad")

Comparing Figure 1 and Table 1, it can be seen that production and coordination costs are static properties associated with the *a priori* definition of each structure. The production costs of the product hierarchy are high because each production manager has only one of each type of processor, while production managers in the other three structures have, directly or indirectly, access to three processors of each type. Similarly, the number of agents in a structure and the number of messages required to assign a task is defined at the outset, and so, the coordination cost for the product hierarchy is low, while for the decentralized market it is high. Although the functional hierarchy and centralized market use 4 messages per task, the centralized market has more agents than the functional hierarchy (15 and 13, respectively), and so, the coordination cost of the centralized market must be higher than for the functional hierarchy.

Vulnerability costs are much more difficult to determine at the outset. The product hierarchy, decentralized and centralized markets will have one product line disrupted if a production manager fails, while the functional hierarchy will have all product lines disrupted with the failure of the "executive office." At the production manager level, therefore, the functional hierarchy is the most vulnerable structure. The functional hierarchy and centralized market are the only two structures that are vulnerable at the functional manager level, with all product lines disrupted with the failure of a functional manager. At the processor level, the product hierarchy is the most vulnerable, since it is the only structure where the failure of a processor results in a product line being disrupted. This is related to production cost.

In short, the decentralized market is the least vulnerable by not having a functional manager layer and having maximal contact between buyers and suppliers, the functional hierarchy and centralized markets are highly vulnerable, with the single point of failure at the production manager level suggesting the functional hierarchy is even more vulnerable than the centralized market.

It is virtually impossible to tell, however, whether the product hierarchy is more or less vulnerable than the functional hierarchy or centralized market. On the one hand, the product hierarchy is the only structure where a product line will be disrupted with the failure of a processor. On the other hand, the independence of the product lines means the product hierarchy does not have the central failure point of an "executive office," or the added failure point introduced by the functional manager layer. As Malone (1987: p1325) points out, the vulnerability of the product hierarchy will be a function of the relative cost of task failure compared to task transfer, and the random distribution of failures at the time of task assignment. In other words, investigation of the vulnerability costs of these structures is a dynamic problem and ideally suited to investigation using a multi-agent system.

THE OSMAS ARCHITECTURE

The general architecture of OSMAS is that agents communicate by placing a message in a central mailbox. Only agents whose name appears on the message can read the message. There are three types of agent: Production manager (PM), functional manager (FM), and task processor (PR). The logical and physical implementations of the OSMAS agents and the messages used to communicate between them are given in Figures 2a-d.

The task assignment message (see Figure 2a) has the same structure regardless of which agent is sending the message. A message is logically MAIL with a DATE field, a FROM field that denotes who sent the message, a TO field to identify the recipient, a MESSAGE field that provides the actual content of the message, and a PRODUCT field that denotes the product being referred to. For instance, a production manager may send a message to a buy processor saying "Buy Product 1," and the task processor would reply with "Bought Product 1" if the task had been completed. The physical implementation shows such a buy message, with a DATE of 1, FROM pm1 TO bp1, with the MESSAGE buy and PRODUCT1 being the subject of the message. Dates correspond to the round in which the message was sent.

The production manager agent (see Figure 2b) is logically a PM agent with a NAME, a PRODUCT LIST of products for which it is responsible, an AGENT LIST of agents it can assign tasks to, and a STATUS switch which is either ON for an agent that is active and can

Logical	Physical
a) Message MAIL(Date, From, To, Message, Product)	MAIL(1, pm1, pr1, buy, product1)
b) Production Manager PM(Name, Product List, Agent List, Status)	PM(pm1, [product1], [bp1, mp1, sp1], on)
c) Functional Manager FM(Name, Type, Agent List, Status)	FM(fm1, buy, [bp1, bp2, bp3], on)
d) Task Processor PR(Name, Type, Current Load, Status)	PR(bp1, buy, 0, on)

Figure 2. Logical and Physical Design of OSMAS Messages and Agents

perform tasks, or OFF for an agent that has failed and thus will be unable to complete a task. The status switch operates the same for each type of agent. The physical implementation, in PROLOG, is shown next to the logical design. In this case, we have a PM agent called pm1 that is responsible for product1, and can assign tasks to three other agents called bp1, mp1, and sp1. It can also be seen that pm1 is currently active. This agent is a production manager from a product hierarchy, since it is responsible for only one product and knows three processor agents of different types, namely, the buy processor bp1, the make processor mp1, and the sell processor sp1.

The functional manager agent (see Figure 2c) is logically an FM agent with a NAME, a TYPE that denotes the type of task processors it is in charge of, an AGENT LIST of agents it can assign tasks to, and a STATUS switch. Functional managers can handle orders for different types of product, and so does not need to keep track of product types, unlike the production managers. The physical implementation shows that this particular functional manager is called fm1, it is in charge of buy task processors, and thus knows the buy processors bp1, bp2, and bp3, and is currently active. Since functional managers in both the functional hierarchy and centralized market operate in exactly the same fashion, this agent could come from either structure.

The task processor agent (see Figure 2d) is logically a PR agent with a NAME, a TYPE that denotes the type of task it is able to perform, a CURRENT LOAD switch, and a STATUS switch. Processors, as the end point of a task, do not need to know the name of other processors, and simply reverse the order of the TO and FROM fields on a task assignment messages to notify the assigning manager that a task has been complete, so they also do not need to keep track of the name of the manager that assigned the task. The physical implementation shows that this particular task processor is called bp1, it is a buy processor, it is currently free, and is active. Buy processors all have names beginning with “b,” and Type=buy, Make processors have names beginning with “m” and Type=make, while sell processors have names beginning with “s” and Type=sell. Managers will check the type and current load of a processor before assigning a task, again, except for production managers in a decentralized market, who know the type, but assign tasks through bids.

Each round consists of the following events in order: An order arrives at one of the production manager agents. This “order” represents a task to be completed and can be viewed as a goal or as the manufacture of a product. The order is completed once all the underlying agents have completed their respective tasks. The production manager first attempts to assign the task to a functional manager. If no functional managers are known, the production manager will then attempt to assign the task to a processor of the right type. If the production manager knows more than one processor of the same type, a "request for bids" will be sent to each processor. Processors will reply to this request with a "bid." The production manager selects from one of the bids and assigns the task to a processor.

If the task has been assigned to a functional manager, this agent will assign the task to the first "free" task processor that it finds. A task processor is free if it is not currently executing a task. This check implements the intelligent task assignment duty required of it by Malone's description in the previous section. Once a processor receives a task, it executes the task and notifies the assigning manager (functional or production) that the task is complete. If the message is sent to a functional manager, then the functional manager will pass the notification on to the assigning production manager.

There are a number of areas where agents know something about other agents, and other areas where they have no information. These information issues are related to the property of bounded rationality normally exhibited by agents within an organization.

Each agent knows the name of agents they can assign tasks to. Managers in direct contact with processors will know what type of processor they are (buy, make, or sell) and whether they have the capacity to perform the task, except for the decentralized market, where product managers will only know the processor type, not the capacity. Agents will not know whether other agents have or are about to fail. Managers must recognize that a task has not been completed and set about reassigning the task if alternative processors are available. When a production manager fails the order is ignored. When a functional manager fails the task is not assigned to a processor. When a processor fails the task is not completed, although task processors in the centralized market will respond to a request for bids if they are free.

THE EXPERIMENTS

OSMAS represents the four organizational structures in Figure 1 by adjusting the product list and agent list of the PM agents. For instance, to change a decentralized market into a centralized market the agent list of the PM agents is altered to include the names of the FM agents (fm1, fm2, and fm3), rather than the nine task processors. With the link now established, the PM agents now assign tasks to the FM agents rather than the PR agents. Changes to the other structures involve the same type of manipulation of the agent list. For the functional hierarchy, only one PM agent knows the FM agents, and the product list is altered to include the name of all three products. The PR agents and simulation engine that execute the models remain the same. Two experiments are proposed here.

Experiment 1: Implement all four structures, with a count of completed tasks based on inter-dependent task processors. Each completed product requires a buy, make, and sell task to be completed.

Experiment 2: Implement a version of the matrix organizational form, with a coordination structure similar to a decentralized market with a layer of functional managers added, and with inter-dependent task processors.

In experiment 1, a task is deemed completed when the organizational structure successfully assigns a task to a task processor and receives notification that the task is complete. Given Malone's description of the task processors as aligned departments, specifically, as supply, manufacturing, sales and marketing departments, it seems reasonable to suggest that the final output (product or goal) is dependent on a series of inter-independent tasks. Afterall, an organization does not simply complete a task, it completes a series of tasks all of which are meant to support the production of a final output or product. Therefore, the flexibility of each organizational structure is determined by the number of products that are successfully completed, with three tasks (buy, make, sell) required to complete a product.

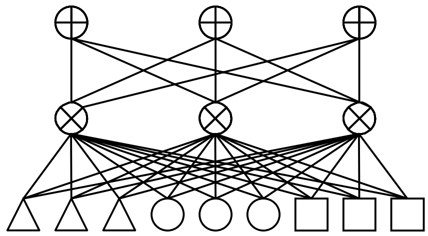


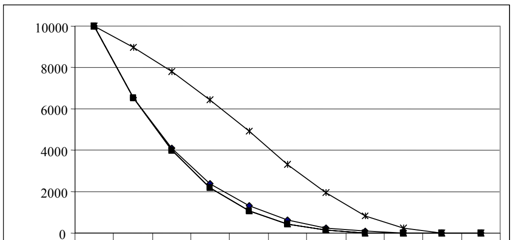
Figure 3: Representation of the Matrix Form of Organization

The second experiment seeks to investigate the performance of the most complex organizational structure using this base set of agents that is related to the matrix form of organizational design (Mintzberg, 1979). In this case (see Figure 3), there will be 15 agents, with each production manager in contact with every functional manager, and each functional manager in contact with every task processor. Since task processors are shared, the production cost of this structure will be the same as for the functional hierarchy, decentralized or centralized markets.

Similar to the functional hierarchy and centralized market, it is assumed that functional managers intelligently assign a task to a processor rather than enter into a bidding phase with processors. Thus, four messages are required to complete a task, one from the production manager to assign the task to a functional manager, one from the functional manager to assign the task to a processor, and two messages for the notification that the task has been completed (from task processor to functional manager, and from functional manager to production manager). Coordination costs are defined as being slightly higher than for the centralized market, with the same number of agents and messages involved, but with more connections between functional managers and task processors. As a result, the matrix form will have a lower vulnerability cost than the product hierarchy, functional hierarchy, or centralized market, but the extra layer of management at the functional manager level suggests the structure could be more vulnerable than the decentralized market.

RESULTS

Each structure is run for 100 cycles of 10,000 rounds each. This number of rounds ensures differences are due to differences in the organizational structures, rather than due to random effects. Each round involves the random assignment of one and only one product order to the appropriate production manager. Tasks are completed within one round. Therefore, with a zero failure rate, each structure can complete 10,000 tasks in each cycle. As the failure rate (FR) increases the differences in vulnerability of each structure will be made apparent by the decline in the number of completed tasks. For all structures, when FR=0 no agents fail, and therefore 10,000 tasks will be completed with no deviation. When FR=1 no agents are active, and therefore 0 tasks will be completed with no deviation. The interesting part is what happens between these two extremes.

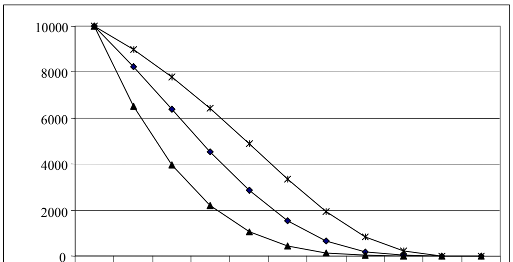


	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PH	10000	6552	4085	2389	1295	626	260	82	16	1	0
FH	10000	6525	3978	2209	1052	418	124	23	2	0	0
CM	10000	6529	3986	2199	1060	421	122	23	2	0	0
DM	10000	8970	7798	6435	4912	3334	1929	843	232	20	0

Figure 4: Comparison of Average Number of Products Completed in Experiment 1

The result of experiment 1, where a count of completed tasks is based on inter-dependent task processors, is presented in Figure 4. As can be seen, the decentralized market is the most successful (i.e., least vulnerable) structure, being able to maintain a much higher product completion rate than the other three structures. The decentralized market can complete more than half the assigned orders until close to 40% failure rate.

The effect of the extra functional manager layer can clearly be seen by the results for the other three structures. Although the functional hierarchy and centralized market have lower production costs by sharing task processors, the extra failure point at the functional manager level increases vulnerability costs to the point that any advantage of sharing task processors is cancelled out. In fact, the product hierarchy is marginally less vulnerable throughout, and is able to produce twice as many products when the failure rate increases beyond 50%, although the total number of products produced is small. Furthermore, while the functional hierarchy has a single catastrophic failure point with its one and only one production manager, because of the distribution of failure throughout the structure this problem does not make it any more vulnerable than the centralized market.



	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
MF	10000	8260	6403	4544	2867	1530	647	191	29	1	0
CM	10000	6529	3986	2199	1060	421	122	23	2	0	0
DM	10000	8970	7798	6435	4912	3334	1929	843	232	20	0

Figure 5: Comparison of Average Number of Products Completed in Experiment 2

The result of experiment 2, where the matrix form of organizational structure is implemented, is shown in Figure 5. The results for the centralized market and decentralized market are also shown to show the impact of increasing the number of connections (compared to the centralized market), and increasing the layers of management (compared to the decentralized market). As can be seen, the extra flexibility given to the structure by allowing each functional manager to be in contact with each task processor results in significantly less vulnerability when compared to the centralized market, but the extra failure point introduced by the functional manager layer means the decentralized market is still the least vulnerable structure.

This would seem to suggest that increasing the connections between agents can significantly reduce vulnerability, but adding a management layer to cope with this extra coordination offsets the benefits, with the decentralized market remaining significantly less vulnerable with fewer agents and fewer connections between agents. The decentralized market achieves this flexibility, however, through a much higher coordination cost.

DISCUSSION

The OSMAS simulation tool has been developed to investigate the cost of coordinating an organization from the viewpoint of vulnerability costs and the flexibility of organizational structure. Combining elements of computational organization theory and coordination theory, it was possible for OSMAS to simulate the expected behavior of four types of organizational structure by designing the micro-level behavior of agents in terms of who they can assign tasks to, and therefore, the range of task-assignment behavior they are able the perform.

The results of experiment 1 showed that sharing task processors and increasing the links between agents significantly increases the flexibility of an organizational structure, defined here in terms of vulnerability costs, or more specifically, the ability of an organization to successfully complete a three task product. By extending the analysis to experiment with a matrix organizational form, it was also found that adding extra layers of management to deal with this added coordination cost negates the advantage. In this case, the decentralized market was the least vulnerable structure, while the product hierarchy performed marginally better than the functional hierarchy and centralized markets. This is due to the lower production costs of the functional hierarchy and centralized markets being offset by the failure point introduced by the intermediary functional manager layer.

A wide variety of further research topics present themselves, including an investigation of alternative organizational structures, the nature of the link between agents, and the dynamic properties of the agents themselves. The coordination structure defined for each set of OSMAS agent is essentially static, being defined at the outset for the duration of the experiment. However, a more realistic model would include relationships that become stronger (or weaker) over time, producing a network of loosely coupled structures (Orton & Weick, 1990; Weick, 1976).

This raises the question of what factors might lead to a strengthening in the degree of coupling within an organizational structure (Beekun & Glick, 2001). Within coordination theory, the impact of technology to reduce communication costs has been put forward (Malone et al., 1987; Malone & Crowston, 1994). Such dynamic organizational modeling further raises the question of under what conditions an organization would migrate or adapt from one form to another (Ethiraj and Levinthal, 2004), and the incentives that might be needed to smooth the transition (Raghu et al., 2005).

Underlying these issues of organizational design there is the socio-cognitive effects of the agents themselves. OSMAS agents are essentially mechanistic, in that, they follow specific rules of how to respond to particular messages. A richer picture of organizational behavior can be developed by extending the cognitive and affective responses of each agent. Such agents can be used to apply game theory to the selfish motivations of OSMAS agents, or imbue them with altruistic tendencies that can sometimes be the core of a successful team, or simulate the roots of mistrust that can be equally effective in destroying working relationships.

Questions can then be raised over exactly how an agent chooses to assign a task to another agent. In OSMAS, this is represented simply in terms of identifying the right type of agent (e.g., a "buy" processor in order to assign a buy task, etc.). In reality, OSMAS agents would develop more complex models of the agents they know, an extension that can be used to investigate the effect of limited coordinating activities on optimal organizational design (Harris & Raviv, 2002).

CONCLUSION

The complexity and interplay of factors that influence the form and function of post-industrial organizations demands an array of tools and techniques to help in their analysis. The multi-agent system proposed here is by no means the only way of looking at current and future organizational forms, but it is one that allows for experimenting with intelligent, dynamic agents most closely resembling the protagonists in real organizations. The design of agents in OSMAS is deliberately simple. The only assumptions we make is that agents perform a particular task, and communicate with a finite number of other agents. The idea here is that the simulation tool implements only sufficient detail to perform the basic task of implementing a coordination structure, but has sufficient power to lay the foundation for further studies in the future direction of organizational form.

REFERENCES

Baligh, H.H., R.M. Burton, and B. Obel. 1996. "Organizational consultant: Creating a useable theory for organizational design." *Management Science* 42, No. 12 (Dec), 1648-1662.

Beekun, R.I., and W.H. Glick. 2001. "Organization structure from a loose coupling perspective: A multidimensional approach." *Decision Sciences* 32, No. 2 (Apr), 227-250.

Burton, R.M., and B. Obel. 1984. *Designing Efficient Organizations: Modeling and Experimentation*. Elsevier Science, New York, NY.

Carley, K.M., and M.J. Prietula. 1994. *Computational Organization Theory*. Lawrence Erlbaum & Associates, Hillsdale, NJ.

Chandler, A.D. 1962. *Strategy and Structure: Chapters in the History of the American Industrial Enterprise*. MIT Press, Cambridge, MA.

Cohen, M.D. 1986. "Artificial Intelligence and the dynamic performance of organizational designs." In *Ambiguity and Command: Organizational Perspectives on Military Decision Making*, J.G. March and R. Weissinger-Baylon (Eds.), Pitman, Marshfield, MA, 53-71.

Cyert, R.M., and J.G. March. 1963. *A Behavioral Theory of the Firm*. Prentice-Hall, Englewood Cliffs, NJ.

Ethiraj, S.K., and D. Levinthal. 2004. "Bounded rationality and the search for organizational architecture: An evolutionary perspective on the design of organizations and their evolvability." *Administrative Science Quarterly* 49, No. 3 (Sep), 404-480.

Harris, M., and A. Raviv. 2002. "Organization design." *Management Science* 48, No. 7 (Jul), 852-865.

Lin, Z., and K.M. Carley. 1997. "Organizational response: The cost performance tradeoff." *Management Science* 43, No. 2 (Feb), 217-234.
Malone, T.W. 1987. "Modeling coordination in organizations and markets." *Management Science* 33, No. 10 (Nov), 1317-1332.
Malone, T.W., and K. Crowston. 1994. "The interdisciplinary study of coordination." *ACM Computing Surveys* 26, No. 1 (Mar), 87-119.
Malone, T.W., K.G. Crowston, J. Lee, B. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C.S. Osborn, A. Bernstein, G. Herman, M. Klein, and E. O'Donnell. 1999. "Tools for inventing organizations: Toward a handbook of organizational processes." *Management Science* 45, No. 3 (Mar), 425-433.
Malone, T.W., J. Yates, and R.I. Benjamin. 1987. "Electronic markets and electronic hierarchies." *Communications of the ACM* 30, No. 6 (Jun), 484-497.
March, J.G., and H.A. Simon. 1958. *Organizations*. Wiley & Sons, New York, NY.
Masuch, M., and P. LaPotin. 1989. "Beyond garbage cans: An AI model of organizational choice." *Administrative Science Quarterly* 34, No. 1 (Mar), 38-67.
Mintzberg, H.D. 1979. *The Structuring of Organizations*. Prentice-Hall, Englewood Cliffs, NJ.
Orton, J.D., and K.E. Weick. 1990. "Loosely coupled systems: A reconceptualization." *Academy of Management Review* 15, No. 2 (Apr), 203-223.
Prietula, M.J., K.M. Carley, and L. Gasser (Eds.). 1998. *Simulating Organizations: Computational Models of Institutions and Groups*. AAAI Press/MIT Press, Menlo Park, CA.
Raghu, T.S., B. Jayaraman, B., and H.R. Rao. 2004. "Toward an Integration of Agent- and Activity-Centric Approaches in Organizational Process Modeling: Incorporating Incentive Mechanisms." *Information Systems Research* 15, No. 4 (Dec), 316-325.
Weick, K.E. 1976. "Educational organizations as loosely coupled systems." *Administrative Science Quarterly* 21, No. 1 (Mar), 1-19.
Williamson, O.E. 1975. *Markets and Hierarchies: Analysis and Antitrust Implications*. Free Press, New York, NY.

TREVOR T. MOORES was born in Vancouver, Canada and studied at Sunderland Polytechnic, UK, where he received a BA (Hons) in Combined Studies (Arts) majoring in Philosophy and Psychology. He received an MSc in Intelligent Knowledge Based Systems from the University of Essex, UK, and a PhD in Management Information Systems from the University of Aston, UK, in 1993. He has taught at the City University of Hong Kong, PR China, as well as Emporia State University, USA. He is currently an Assistant Professor at the University of Nevada Las Vegas, USA. His email is trevor.moores@unlv.edu. His webpage can be found at http://www.unlv.edu/faculty/tmoores/new.