Cybernetics and Systems: An International Journal, 37: 725–753

ISSN: 0196-9722 print/1087-6553 online
DOI: 10.1080/01969720600887012

SIMULATING STRUCTURAL CHANGE IN ADAPTIVE ORGANIZATIONS

OSWALDO TERÁN

Centro de Simulación y Modelos (CESIMO),
Universidad de Los Andes, Mérida, Venezuela

JUAN DE LARA
MANUEL ALFONSECA

Escuela Politécnica Superior Ingeniería Informática,
Universidad Autónoma de Madrid, Madrid, Spain

JOSÉ LISANDRO AGUILAR

CEMISID, Universidad de Los Andes, Mérida, Venezuela

ANA MUÑOZ

Instituto Universitario Tecnológico de Ejido,
Ejido, Mérida, Venezuela

This paper presents an agent-based model of an organization. The model is made of a social network—composed of the different organization workers—and a knowledge network. Workers are assigned tasks, for which they have to use information in the knowledge network. We have modeled the quality of the information by assigning each information item a probability of being wrong. Agents can interact with other agents, who can recommend to them new information items in the knowledge network for the task to be performed. Workers are assigned different information-seeking behavior (passive, active, and learning), governing the way in which they interact with each other. Moreover, indirect interaction is also

Address correspondence to José Lisandro Aguilar, Escuela de Ingenería de Sistemas, Universidad de Los Andes, Mérida, 5201, Venezuela. E-mail: aguilar@ula.ve

possible, as a publicly accessible knowledge base contains each agent's preferred information items.

The model was implemented in SDML, and its simulation shows that agents quickly learn to discern the better information items for the given task. However, group formation (agents' collaborating by exchanging information) takes longer to stabilize. Additionally, when the quality of items is changed, agents can quickly select the better new knowledge items, and organization performance improves again to a maximum that is only randomly disturbed.

INTRODUCTION

In computer modeling and simulation, systems are described using a certain language for the purpose of virtual experimentation (Zeigler et al. 2000). Many disciplines benefit from these techniques, as they are allowed to experiment in cases where using the real system would be difficult, expensive, non-ethical, or impossible. Some of the areas that traditionally have used simulation as a research or decision-making tool include physics, mathematics, biology, and economics. Lately, simulation is becoming increasingly popular in the social sciences with the emergence of agent-based simulation techniques (Gilbert and Troizsch 1999). In this paradigm, the fundamental element is the agent (Jennings et al. 1998), which can be defined as a "computer system, situated in an environment, which is able to perform flexible and autonomous actions to achieve its design objectives."

In this way, in agent-based modeling and simulation *macroscopic* phenomena *emerge* by the actions and interaction of elements at the *microscopic* level (Alfonseca and de Lara, 2002). Thus, the classical macro-level approach of representing systems as differential equations is no longer used. Instead, the variation of the quantities in the model is obtained from the explicit modeling of the individual elements. For example, an agent-based model of an ecosystem does not use the Volterra (1931) equations (which give the variation in the population of the species in the ecosystem by means of a system of differential equations), but explicitly models each individual in each species and their interaction in a given environment. Moreover, after parameter calibration both models should produce comparable results.

Social and organizational modeling originated from seminal work such as Simon (1948, 1984), Cyert et al. (1949, 1963), March et al.

(1958), Newell et al. (1976), Cohen (1985), and Newell (1990). The approach centers its attention on the decision-making process and behavior of the individuals and the overall system (that is, an organization). Several assumptions were made, such as bounded rationality of individuals, complexity of both the individual's surrounding and the organization's environment (e.g., technology is unclear), and ill-definedness of the individuals' decision-making situations (there is ambiguity of choice and problematic preferences). The individual's decision-making process is considered a key aspect for understanding social and organizational systems. Cognitive theories of individuals (Simon 1984; Cohen 1985; Newell 1990) have been suggested and implemented. Classical models in this area are the models proposed in Cohen et al. (1972) and Masuch et al. (1989). Both models follow Cyert et al.'s *Behavioural Theory of the Firm*. The two models share the objective of explaining how organizations can survive despite pervasive apparent disorder. More recently, other directions of research have emerged. One example is Carley's models for understanding structural change and learning in organizations. One of her models (Carley 2001) was the starting point of the model we show in this paper.

In the present article, we use the agent-based modeling and simulation paradigm to model inter-individual and individual-knowledge interactions (*who-who* network, and *who-what* network) in an organization. The evolution of these two webs is driven not only by individual interest in obtaining new knowledge (*active interaction*) or by individual affinity (shared knowledge, which we call *passive interaction*) as in Carley's model (Carley and Hill, 2001) and (Carley, 2001), but also by individual's learning in accordance to organizational performance. Learning-driven interaction occurs as agents classify both other agents and knowledge items. Learning or evolution of the individual's mental models is allowed by using the endorsement mechanism explained by Cohen (1985) and by Moss (1995) in order to classify other agents and knowledge items according to past performance.

During the simulation, the organizational network (*who-who*) undergoes a "*soft*" structural change, where the workers learn to discern the better knowledge items to perform the given task. After a certain period of time, the organization achieves certain stability and the "*entropy*" decreases. After such stability is achieved, a period of "*strong*" structural change is induced, as we change the quality of the knowledge items for the given tasks. As a consequence, the organization's learning, up to that point, becomes in part obsolete. A reorganization becomes necessary. This phenomenon occurs in a period of "*strong*" structural change in the organization, as the inter-agents and the agents-knowledge interrelations have to be changed in order to adapt to the new situation. After such variation of the reliability of the items, the organizational entropy highly increases. This entropy is lower during the period of soft structural change. Later on after the probability of the items is varied, certain stability is achieved again, and entropy decreases.

The aim of this paper is to analyze both soft and strong structural change. For this purpose, measures of behavior, such as performance, knowledge diffusion, group consensus, and dynamic group formation, are examined. The latter (dynamic group formation) is examined, as in (Carley and Hill 2001), in its simplest form: the triad (a group of three, collaborating individuals). Results show that agent's and organization's learning occurs in a relatively short time after modifying the quality of the knowledge items. Such learning is much faster than the initial learning at the beginning of the simulation when agents start obtaining knowledge from an almost empty "mind." (At the beginning of the simulation, the agents' database is almost empty, and individuals are given only two knowledge items.) As expected, changes at the inter-individual and individual-knowledge webs get slower as the simulation goes on in a period of low structural change, as agents' knowledge stabilizes and performance of both the agents and the organization increase.

The rest of the paper is organized as follows: Section 2 presents the basic organization model. Section 3 briefly comments the implementation in the SDML language. Section 4 discusses the experiments performed and obtained results. Finally, section 5 ends with the conclusions and future work.

THE ORGANIZATION MODEL

In our agent-based model of organizations, workers are represented as cognitive agents. The organization goal is classifying a given problem in one of two possible classes. This is a classical task in organizational modeling and is described in Carley and Hill (2001) and Carley (2001). Our model is in fact an extension of Carley's model in several ways, as it is discussed throughout the paper. The problem to be solved by the agents is modeled as a string of binary digits. The problem is classified as "1" if the number of "1" items exceeds the number of

"0" items and vice versa. The organization decision is based upon its workers' decision. In this way, the problem is partitioned and solved by the organization workers. Each one of them can analyze a limited number of items and produces an individual result according to this partial knowledge. The organization decision is made by taking the majority of classification results from all the agents. A schema of our model is shown in Figure 1.

Each agent has a seeking information behavior that is chosen probabilistically among *passive*, *active*, or *learning*. Passive agents interact with others similar to them, in the sense that they use similar information to solve the given task. In each interaction the agent may take a new knowledge item, up to the agent memory limit. Active agents tend to interact with agents that use information that is different from theirs. Finally, learning agents interact with the agents that gave them good knowledge items in the past. For this purpose, they classify other agents and knowledge items depending on the individual performance. Thus, the individual result is compared with the expected result, and the interaction preferences are updated accordingly.

Agents are able to discard knowledge items in their memory and substitute them with others recommended by other agents. Moreover,

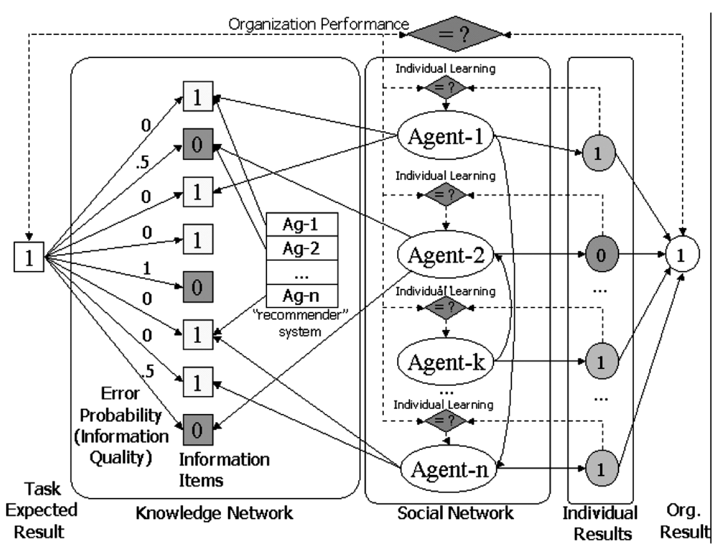

Figure 1. A scheme of the organization model.

similar to cooperative systems, we have modeled a "*recommender system*" containing the best items found by each agent. In this way, an agent can probabilistically choose an item from the recommender system.

From the local interactions of the agents, we expect the organization to develop self-organization mechanisms and reach a certain degree of stability. According to (Bonabeau et al., 1999), self-organization relies on four basic elements: (i) multiple interactions between the agents in the system, (ii) positive feedback (amplification of certain behaviors), (iii) negative feedback (counteracting positive feedback by deadening mechanisms), and (iv) amplification of random fluctuations, to facilitate the discovery of new solutions and prevent convergence to suboptimal solutions. In our case we need a mechanism to help in the discovery of new solutions; that is, finding better items for the given task. Thus, we assign a small probability for an agent to choose a random item in the problem space. This is especially useful if the organization environment changes. If the fitness of the knowledge items for the task changes, the ability to access unused items allows a re-organization of the *who-who* and *who-what* networks. This phenomenon is explained in detail in Section 4.

Figure 2 depicts a conceptual model of the organization using a UML class diagram (Booch et al. 1999). This is a kind of diagram commonly used in computer science. Each box represents a concept (a *class* in computer science jargon), that may contain a compartment for the concept attributes. Concepts are related to other concepts by means of relationships. A hollow triangular arrow means an "is-a" relationship. For example, a *problemItem* is an *Information*. Moreover, child classes inherit all the properties of parent classes. In this way, *problemItem* inherits the *Value* attribute from *Information*. A black diamond in a relationship extreme means composition ("made-of"). For example, a *problem* is made of *problemItem*s. Relationships are optionally labeled with their name and multiplicity. For example a *problem* is made of zero or more *problemItem*s. Moreover, relationships can be qualified with attributes, which are shown in class-like notation. For example, relationship *itemError* between *problemItem* and *correctAnswer* has attribute *reliability*, modeling the error probability (if zero, the item is correct).

For the implementation we do not strictly follow this conceptual diagram, but rearrange the information for better performance and easier coding.

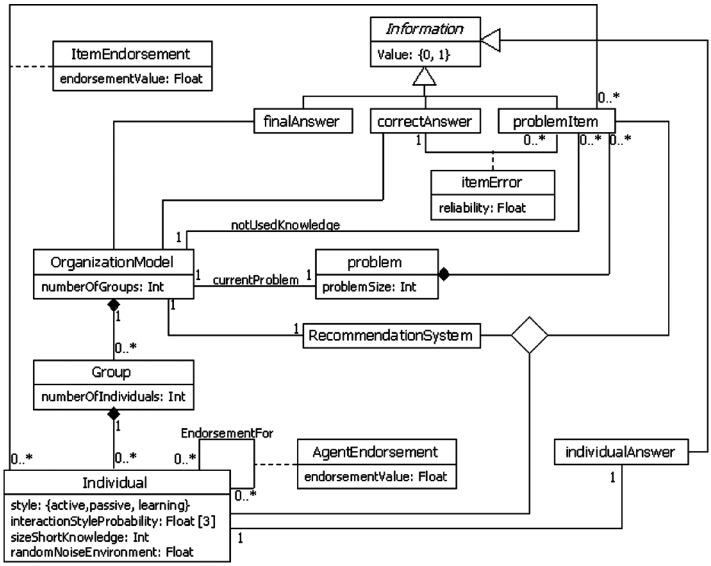

Figure 2. Conceptual model of the organization (in UML notation).

Agents classify other agents (*AgentEndorsement* in Figure 2) and knowledge items (*ItemEndorsement* in Figure 2) depending on the obtained performance. This classification is a numerical value (a *weight*) that is used to know how good the interaction with the other agent or the use of the item was in the past. For this classification, we use the concept of *endorsement* (Cohen 1985, Moss 1995). This concept defines the following formula to calculate the weight:

$$E(\mathbf{b}, \bar{\mathbf{a}}) = \sum_{\mathrm{val}(\bar{\mathbf{a}}_i) \geq 0} \mathbf{b}^{\mathrm{val}(\bar{\mathbf{a}}_i)} - \sum_{\mathrm{val}(\bar{\mathbf{a}}_i) < 0} \mathbf{b}^{|\mathrm{val}(\bar{\mathbf{a}}_i)|}$$

where $\mathbf{b}$ is the chosen basis, $\bar{\mathbf{a}}$ is a vector of attributes of the object to be endorsed and val is a function giving the value of that attribute.

In this model, individuals endorse information items and other individuals. When information interchange occurs, individuals keep track of both the item and the individual suggesting such item. After each *problemCycle*, an individual compares its suggestion with the real input item, and, according to the goodness of its answer, it endorses the item and the individual that suggested it. The above *E* function calculates the weight of an item or an agent according to these subsequent evaluations. The base **b** is set to 1, and the two possible evaluations are either 1, in the case that the agent suggestion is good, or –1, in the other case. Thus, the function *E* reduces to

$$\mathrm{E}(\mathbf{b}, \bar{\mathbf{a}}) = \sum_{\mathrm{val}(\bar{\mathbf{a}}_\mathrm{i}) \geq 0} 1 - \sum_{\mathrm{val}(\bar{\mathbf{a}}_\mathrm{i}) < 0} 1$$

where val($\bar{\mathbf{a}}_i$) might take the value 1 or –1. An individual remembers only those endorsements set in the last 4 days. This value may seem small; however, since there are 40 cycles each day, quick learning occurs.

When an individual interacts in the learning mode, it chooses another individual in accordance to its weight. This selection is random; each individual has a probability of being chosen proportional to its weight. Similarly, when an agent gives a hypothesis about the input, it uses a set of items chosen in accordance to a uniform distribution probability, where each item has a chance proportional to its relative weight. If an individual has a memory of *size S* (in this experiment *S* is usually set to seven, a number suggested by Herbert Simon), then *S* items have to be randomly chosen at each problem cycle. In order to suggest the problem result, individuals always use the item endorsements. On the contrary, only agents whose behavior is learning endorse other agents.

In addition, a given organization can be decomposed in static groups, or departments, each of them made of a certain number of individuals. Individuals belonging to different static groups do not interact. For the experiments in this paper, we set the number of static groups to one, as we do not want to restrict the interaction between individuals to study dynamic group formation.

IMPLEMENTATION IN SDML

For our implementation, we have used the SDML language (Strictly Declarative Simulation Language, Moss et al., 1998), which is based on the KD45 modal logic. With this system the user has access to a certain number of predefined types, or *agents*, with a basic functionality, which should be sub-classified to describe the actual problem concepts. Agent behavior is specified in a declarative way, by means of rules. Agents may be nested, and at each level of nesting one may define a number of *time levels*, the most external one being *eternity.* Initial and final rules can be associated with different time levels. For instance, in our organization model, we have defined the *day* time level, inside which is the *problem cycle*. Each *day* encloses 40 *problem cycles*, meaning that 40 problems should be solved every day. A complete simulation run lasts for a number of *days* (in the range of 500). Figure 3 shows the relationship between the time levels and the associated actions, using a UML activity diagram.

In our model, the most external agent is the *universe*, which is an instance of *universalAgent*, one of the built-in types of agent. Agent *universe* creates (and contains) another agent (called *Organization*), which belongs to the type we call *OrganizationModel* and contains the main simulation logic. This agent creates the groups inside the organization (instances of the agent type we call *Group*). Each of these groups, in turn, contains several instances of the agent type *Individual*. This hierarchy has been reflected in Figure 2, and is explicitly shown in Figure 4. The agent types *OrganizationModel* and *Group* are subtypes of the SDML type *parallelAgent* since their instances contain several agents running in parallel.

Agent *Organization* sets the simulation environment, creates the organization groups, and gives values to the simulation parameters. An overview of the database for this agent is shown in Figure 5. In our implementation, we have followed the conceptual model described in

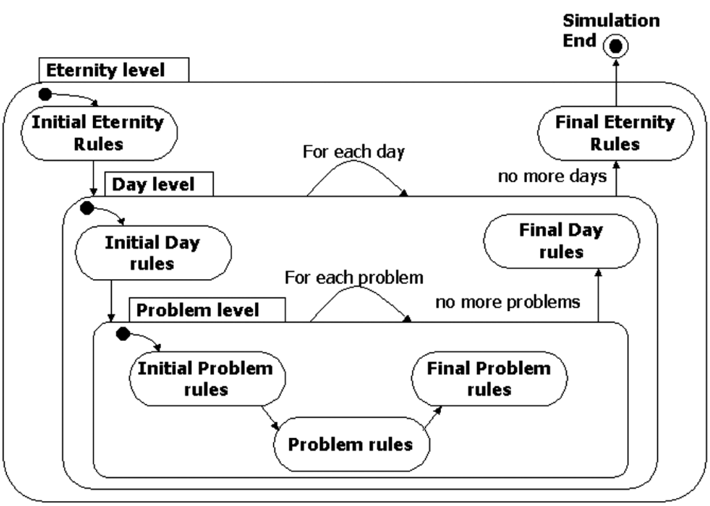

Figure 3. Time levels for the organization model.

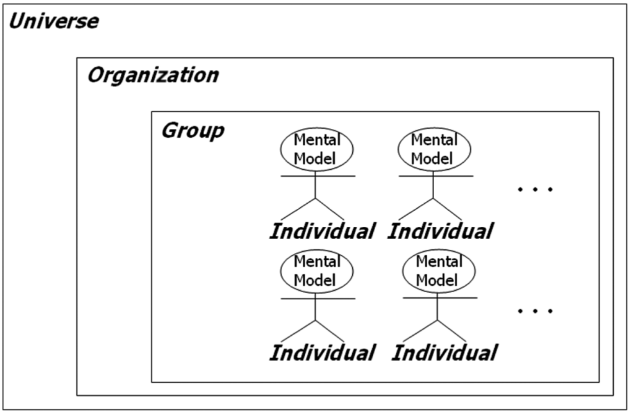

Figure 4. Agent hierarchy of the model.

Figure 2, with a few changes for the sake of efficiency. Among the parameters created by this agent, we may mention the following:

- *interactionStyle*, which describes the probabilities of an agent being active, passive, or learning. In our experiments, these probabilities

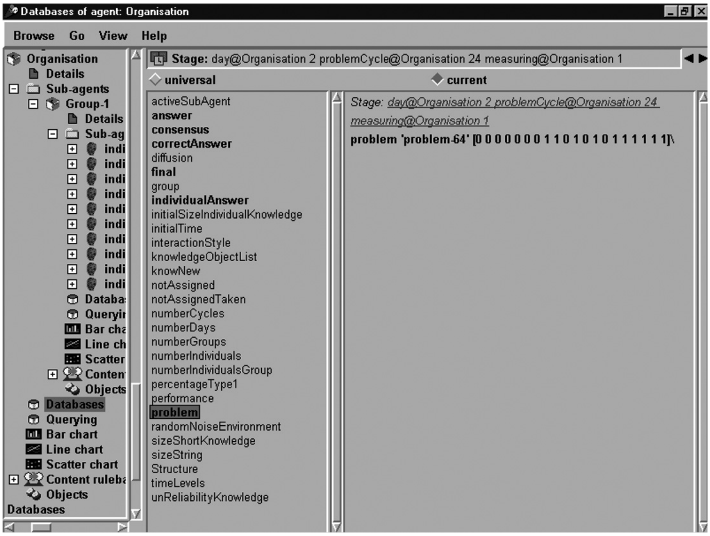

Figure 5. Database of agent *Organization*.

are the same for every individual; thus, this parameter has been moved to this level to save memory.

- *unReliabilityKnowledge* is a list indicating the probability that each information item is erroneous. In Figure 2, this is the *reliability* attribute in the *itemError* relation that joins items *problemItem* and *correctAnswer.*
- *sizeShortKnowledge* is the number of items an agent uses while classifying its input. As this value is the same for every individual, it has been moved to the organization level.
- *randomNoiseEnvironment* is the probability that an individual (interacting in the active mode) chooses an item among those not used by any agent, rather than using the recommendation system. The recommendation system is made of a list of contributed items, one per individual. Every day each individual contributes the item with the highest endorsement value in its database.
- *Problem* is a list of items that define the problem in each problem cycle. (Figure 5 shows the problem for cycle 24, identified as 'problem-64.')
- *correctAnswer* is a list of binary digits containing a description of the problem. The problem is generated from this list, changing each bit according to the probability associated to each item.
- *individualAnswer* is the answer of each individual, which is used to compute the global answer.
- *Answer* is the global answer provided by the organization.

Subagent *Group-1* of *Organization* is the only existing static group in the experiments described in this paper. It contains data such as the number of individuals in the group/organization (*numberOfIndividuals)*, plus additional information indicating who interacts with whom.

Individuals are responsible for most of the simulation dynamics. They interact with other agents by interchanging information items every day, make a classification of the problem (1 or 0), learn from their experience by using endorsements for characterizing and evaluating both information items and individuals, decide the mode of interaction every day, and so forth. The database for an individual is shown in Figure 6.

Next, we present the simulation mechanism by describing the actions to be performed by each group of rules described in Figure 3. In the initial *eternity rules*, the *organization*, *group*, and *individuals* set their parameters. In particular, the group creates the agents inside it.

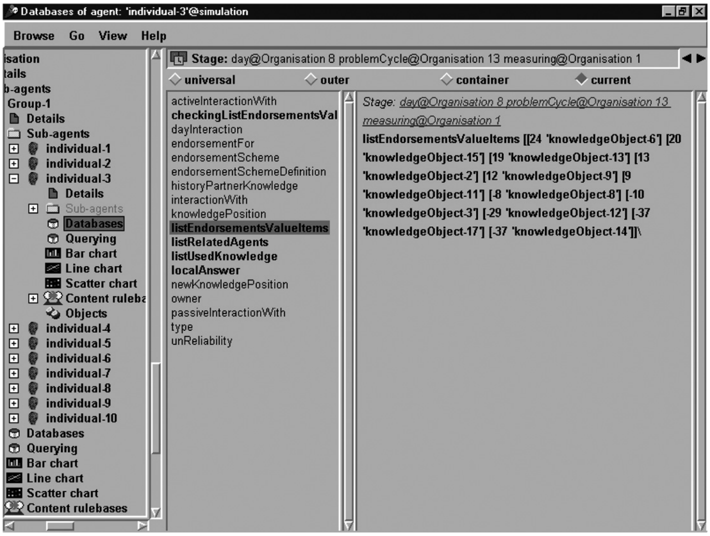

Figure 6. Database for an individual.

At the beginning of each day (*initial day rules*), the *organization* creates a list with the knowledge not used by any agent. The *group* determines who interacts with whom; that is, the inter-individual interaction (in case interaction is active or passive). Each *individual* updates the knowledge items (an item is kept only if it is among the first *l* information items with the highest endorsement value, where *l* is the size of the individual's short memory, or if the endorsement value for such item is positive), chooses its interaction style (passive, active, or learning) and who it will interact with (in the case that selected interaction style is learning). In case of being active, it chooses an item from the environment. It can be taken either from the recommendation system or from the list of not used items. Then, it gets an item from an individual (keeping track not only of the knowledge item but also of the agent) and updates the history of its partners (when a knowledge item is lost, the particular link with the agent the item came from is lost) and the endorsements (those not older than four days are kept). Finally, it makes a global suggestion (the item with the highest endorsement value) to the *organization*.

At the beginning of each cycle (*initial problem cycle rules*), the *organization* generates the problem. Then, in every problem cycle (problem cycle rules), each *individual* classifies the problem and endorses the used items and related individuals, after comparing their answers and the correct answer. At the end of each problem cycle (*final problem cycle rules*), the organization calculates its answer and the consensus. Finally, at the end of each day (*final day rules*), the *organization* calculates several statistics, such as knowledge diffusion, triads, and performance (explained in the next section).

SIMULATION EXPERIMENTS

In this section we enumerate a few experiments performed with the above-described model. In our first set of experiments we tested different combinations of interactions styles for the agents, by setting their probabilities to one of the following combinations: equally biased, always passive, always active, or always learning. Differences in the agent memory size (3, 7, or 10) were also tested.

The reliability of the input items was changed during the simulation runs, as this paper's objective was to examine the dynamics of the simulated organization when the quality of the items changes. The initial reliability of the input items, at the beginning of the simulation runs, is the following:

$$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 1\ 1\ 1\ 1\ 1\ 1]$$

Simulations were run for 500 days. The reliability values were randomly changed on simulation days 100, 200, 300, and 400. As explained above, the individuals solve 40 problems every day, one per problem cycle. Thus, in a full simulation run, 20,000 problems are solved. 40 problem cycles per day is appropriate because they provide the individuals with sufficient experience to make learning possible. Individuals learn as they endorse information items and other individuals, after each problem cycle, according to how good their answer was. On the other hand, running the simulation for one hundred days before modifying the reliability of the information items, guarantees that agent learning, group formation, and performance stabilizes. In a sense, the last part of each period of 100 days is independent from the previous period.

For the case of equally biased individuals, the simulation was run four times (i.e., four trajectories were studied). Thus, for this case, $4 * 5 = 20$ simulation periods of soft structural change and $4 * 4 = 16$ transition periods from soft structural change to strong structural change were analyzed. On average, each simulation took about two days to finish with a Pentium IV computer.

Results

Organizational Performance. Organizational performance measures (as a percentage) how often the organization gives the correct answer to the problems. Seven different types of experiment were performed. The first four gave rise to very similar behavior, while the last three were different. Figure 7 summarizes the results of the seven cases.

(a) *Base model*: The interaction probabilities are set to 0.3, 0.3, and 0.4 for the passive, active, and learning modes, respectively. The agents were assigned a memory of seven items. When simulation begins, the time needed to achieve maximal performance is about 20 days. After this *transitory period*, performance oscillates in the [0.95, 1] range, and some aspects (not all) of soft structural change decrease. For instance, group formation does not decrease (see the next section). However, after the reliability of the input items changes; i.e., after a strong structural change is induced, the length of the transitory period reduces to less than half (4–8 days). This can be understood by remembering that, at the beginning of the simulation, the agent's mental model is empty (it contains only two information items without endorsements), while, after a reliability change, the agent knows many items, although they are wrongly endorsed after the change, and the agent needs to update them. Just after a strong structural change, the performance decreases to around 0.5, i.e., the organizational performance becomes random. In this model, once the stable period is reached, the randomness of the environment input items has little impact on the performance.

(b) *Active model*: The agents are always *active*. Compared with the base case (a), a faster learning is observed at the beginning of the simulation.

(c) *Short memory model*: This model is identical to the base model (a), except that the agent's memory size is 3. In this model, the performance increases faster, and the range of oscillation is smaller (the performance is almost always equal to the maximum, 1). After changing the reliability of the information items, the performance of this model is affected more strongly than in the base model.

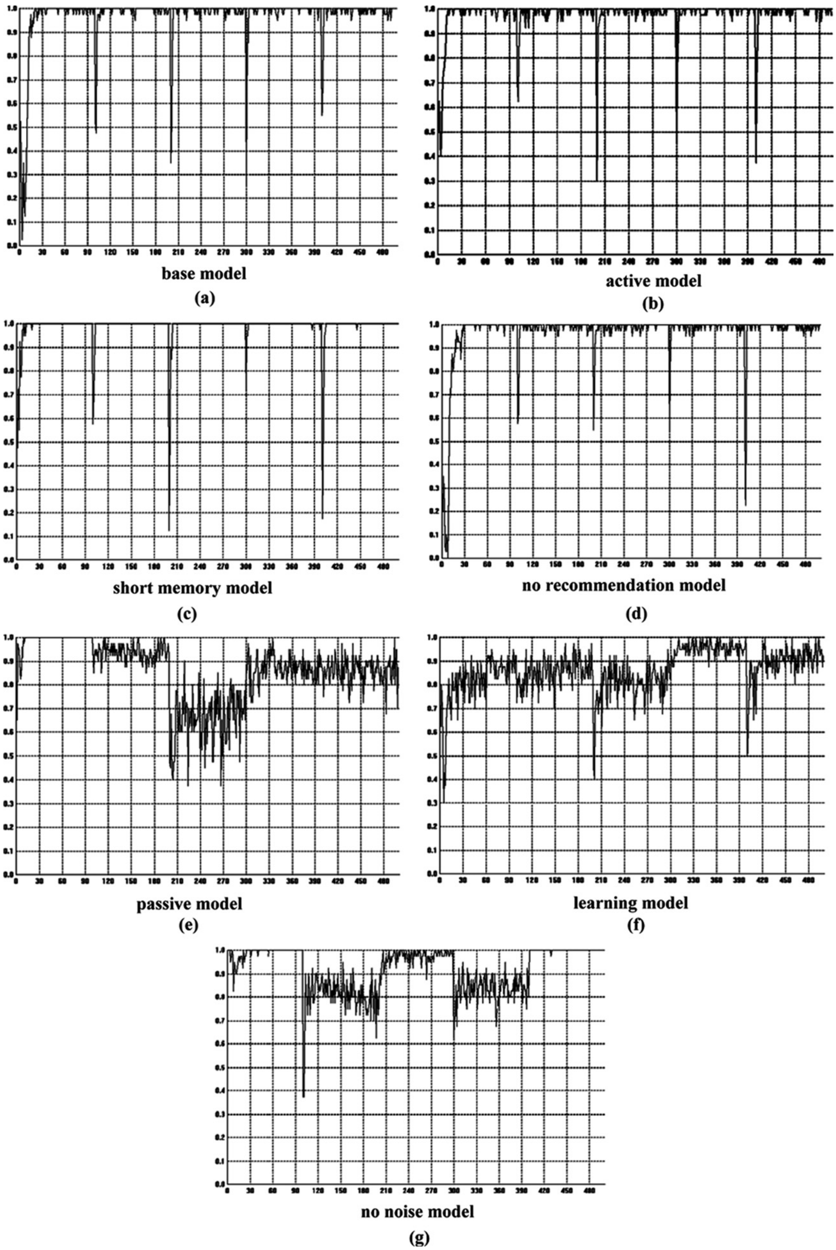

Figure 7. Performance for (a)–(g) models: strong structural change induced each 100 days.

(d) *No recommendation model*: This model is the same as (a), but without recommendation system. This model's behavior is very similar to the base model (a)'s.
(e) *Passive model*: The agents are always passive. The performance stabilization is achieved in about 20 days. However, as expected, after some time the passive interaction stops, as the individuals exhaust their possibilities for information exchange. This confirms Carley's results, but in our case learning and information diffusion seem to be much faster. After a strong structural change is induced, good learning is a question of chance, and behavior becomes good or bad in general, without improving beyond a certain upper bound (different for each period of soft structural change). In all the cases, the performance oscillates in a certain range.
(f) *Learning model*: The agents are always learning. Stabilization takes longer, and performance oscillates in a wider interval (which only occasionally reaches the upper bound of 1), as compared to the base model (a). In this case, the randomness in the environment continues affecting the organization's performance over time.
(g) *No-noise model*: Similar to (a), but without *noise*. The presence of *noise* means that the agents can choose items from the environment, currently unused by other agents, perhaps discarded by the agents. In this model, the performance improves until it oscillates in a certain range, which only occasionally reaches the value of 1. In general, randomness in the environment continues affecting the organizational behavior.

Summary: Active learning and noise from the environment seem to be the main source of the differences between the different configurations of the experiment. The last three models are markedly different from the first four, whose distinctive characteristic is the fact that agents are active and receive unused information (noise) from the environment. This is understandable, because active agents look for individuals with novel knowledge, apart from been able to perceive noise from the environment. In the three last cases, either the agents are not active, or they cannot perceive items from the environment. The ability to access unused knowledge seems essential to recovering the performance after a strong structural change is induced. Interestingly, individual learning does not seem to help the organization performance. As shown in Figures 7b and 7f, an organization whose individuals are active performs better than those whose individuals are learning. Moreover, the recommender system does not seem to improve the performance, as long as individuals are able to interact in an active way.

Dynamic Group Formation, Triads. In this subsection we analyze dynamic group formation in its simplest form: groups of three, or triads (Carley and Hill 2001). As in the previous subsection, we have analyzed seven models, in four of which active interaction between agents and noise from the environment have not been suppressed. These models present similar behavior. Figure 8 presents the results.

(a) For the *base model*, the stabilization of triads takes longer than performance stabilization, about 40 days for the first period of soft structural change (compared to 20 days in the previous set of experiments), and about 10 days after reliability changes (compared to 4–8 days). These results agree with Carley's model (Carley and Hill 2001). Before the organizational structure has stabilized, the performance reaches the best it can get (it is stabilized and oscillating). However, both stabilizations seem to happen in less time than in Carley's model. On the other hand, group disturbance is present and goes down sometimes to values as low as 64 (one half of the maximal 120, the upper bound of the oscillation range). This happens shortly after a strong structural change has been induced. The performance is less erratic than the triads—apparently, group disturbance does not affect performance. In this case, the maximal number of triads (120) is usually reached.
(b) Compared to the base model, in the *active model*, the maximum number of triads (120) is also reached, but the groups are much more unstable; the number of triads is much more variable. In fact, this number is usually in the [90, 120] range; although in some periods of soft structural change this range becomes smaller or wider (120 always being the upper bound). At the beginning of the simulation, the transitory period is about 24 days, again longer than the transitory period for performance, which is about 12. The oscillation ranges for the different intervals of soft structural change are different.
(c) In the *short memory* model, the triads are much more unstable than in the base model. They rarely reach the maximum of 120, sometimes going down to values such as 70, during the period of soft structural change, which never happens in the *base model*. It seems that

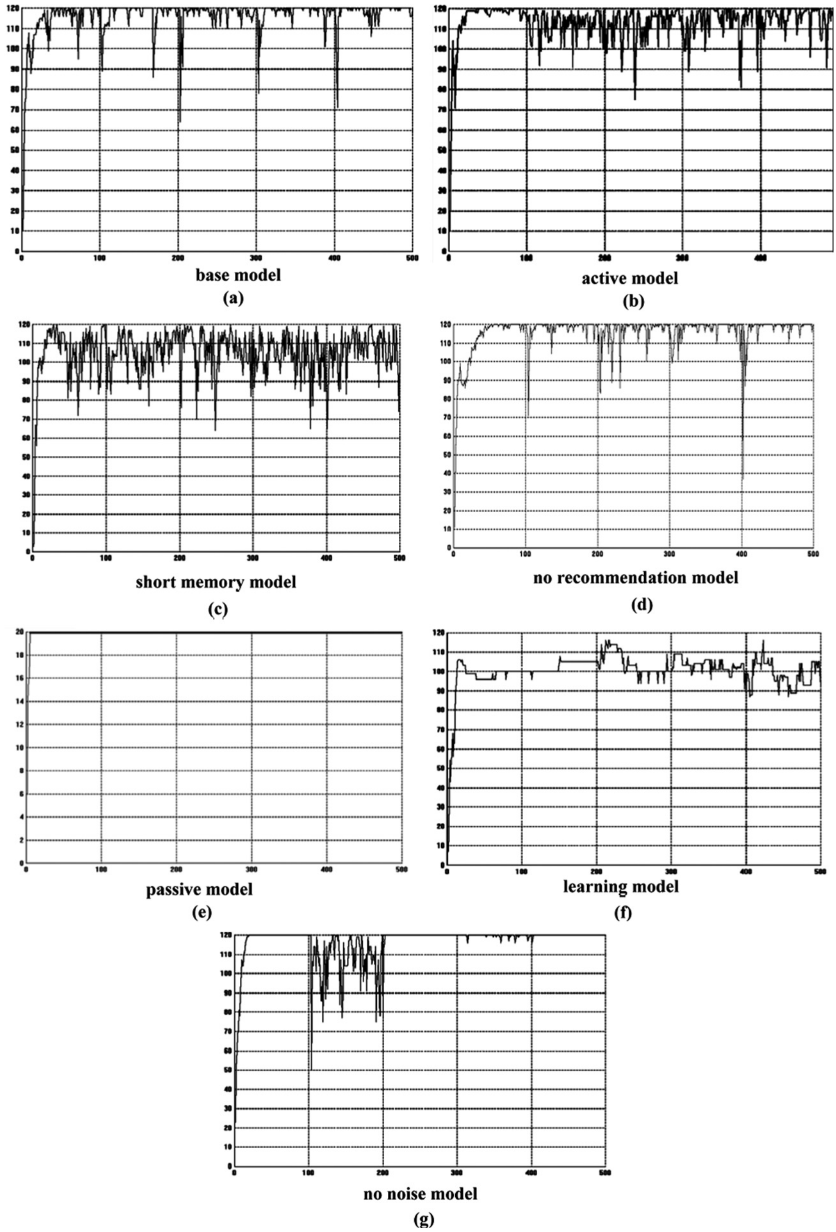

Figure 8. Triads.

stabilization takes less time, about 20 days, but it is not clear, because the number of triads is completely erratic. The stabilization observed in Figure 8 is not very convincing, but this trend is backed by the performance stabilization at about day 8 (see the previous section and Figure 8). It seems that a shorter memory (3, compared to 7) allows for faster learning, but group formation is unstable. This might happen because individuals choose more dynamically new individuals and forget the old ones, which favors fast learning. This is helped by the fact that each day consists of many problem cycles (40), which allows the information items to be evaluated many times, so a good selection can be reached.

(d) For the *no recommendation* model, at the beginning, the triad stabilization takes about 60 days (much longer than the performance stabilization, which takes about 30 days). The number of groups is usually near the maximal 120, although in this case it was quite erratic. The triads are not disturbed after reliability changes. This aspect of their behavior is similar to that observed in the base model, although more instable. This instability is lower than in the *active model.*

(e) For the *passive model*, once the interaction of agents stops, the number of triads achieves a fixed value, which is different for every simulation experiment. In our case, this value happened to be 20. After reliability changes, the number of triads keeps its previous value. For this configuration, at the beginning, the group formation stabilizes in about six days, and the performance in about 10. This is an atypical case, compared with the previous ones. It seems that, after interaction stops, the individuals continue learning and improving their answers. In this configuration, the individuals are not allowed to interact with their knowledge models, but they still use endorsements to classify the information items.

(f) For the *learning model*, very unlike the *base* and the *active models*, the maximal number of triads (120) is never reached, but groups are more stable. This may happen because the individuals keep their endorsements of other individuals for several days, thus slowing down the group change. At the beginning, the triad formation transitory period is about 16–20 days, again longer than the performance transitory period, which is about 10–11 days. Group formation is much less erratic than the performance, which continues erratic during the whole simulation, in the range [0.7, 0,95]. Only for two of the five cases of soft structural change we have simulated, the performance reaches the upper bound of 1. Examples of ranges for the number of triads in the different intervals of soft structural change are [95, 106], [93, 115], [90, 110]. In some of these intervals, the variability of the triads is high and erratic.

(g) Finally, for the *no noise model*, the performance stabilizes around 28, and the number of triads in around 20 days. This is another atypical case. Usually the number of triads reaches the 120 maximum with a small oscillation range, with some exceptions (as in the second period of soft structural change).

Summary: In most of the configurations, the organizational structure, defined as the number of triads, stabilizes later than the performance. This is not the case for two models: the passive, and the no noise model, where both the performance and the group formation stabilize in less time than in Carley's model. In addition, the performance is less erratic than the number of triads—apparently, group disturbance does not affect performance. In all the models, except in the learning and the passive, the triads reach their maximal value of 120, although in the active and the short memory models (especially in the latter) their number is erratic. Learning behavior in agents seems to produce more stable groups because of the selection mechanisms of the agents with which the interactions take place. As endorsements are kept for several days and change slowly, group changes take longer.

Agent's Consensus and Diffusion. Consensus is the proportion of agents that suggest the answer that the organization finally adopts. For example, if for one problem the organization's answer (or hypothesis) about the number of ones and zeroes in the input is 1, and seven agents (from a total of 10) suggested this value, then the consensus for that problem is 0.7.

Diffusion is the proportion of agents that effectively exchange information in a given day. An agent may want to exchange information, but the interaction will effectively occur only if both agents do not share the same knowledge items.

Agent consensus and diffusion is shown in Table 1. Again, models (a)–(d) are very similar, although in the active model the information diffusion is smaller. The short memory model shows a higher diffusion than the base model; having fewer items in memory raises the probability of knowledge exchange. For the other three models, the consensus is smaller. The passive model shows that diffusion stops, as expected. The

Table 1. Agent consensus and diffusion

	Consensus envelopes		Diffusion envelopes	
Measure model	General	Most common	General	Most common
Base	[0.5, 1]	[0.8, 1]	[0, 0.7]	[0.3, 0.5]
Passive	[0.5, 1]	[0.6, 0.9]	[0,0]	[0,0]
Active	[0.5, 1]	[0.8, 1]	[0, 0.6]	[0, 0.2]
Learning	[0.5, 1]	[0.6, 1]	[0.2, 0.6]	[0.2, 0.4]
No noise	[0.5, 1]	[0.6,1]	[0.1, 0.7]	[0.2, 0.6]
No recommendation	[0.6, 1]	[0.7, 1]	[0.1, 0.7]	[0.2, 0.7]
Short memory	[0.7, 1]	[0.8, 1]	[0.2, 0.8]	[0.3, 0.7]

reason is that, in this model, the agents tend to interact with other agents that have a similar knowledge, thus blocking knowledge diffusion. Figures 9 and 10 display the evolution of consensus and diffusion for the base model.

Structural Change: Organizational Behavior and Reorganization. The aim of this section is to examine *structural change* (SC), i.e., organizational behavior (soft SC) within a period with fixed reliability of the

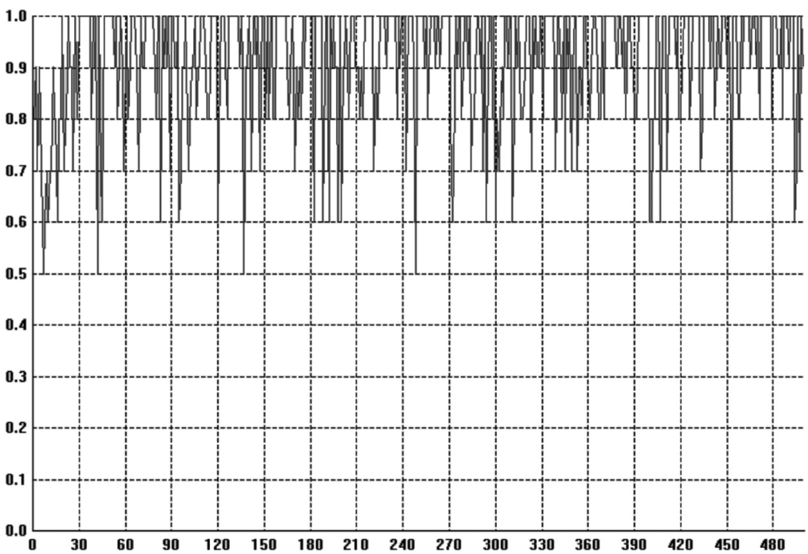

Figure 9. A consensus trajectory for the base model.

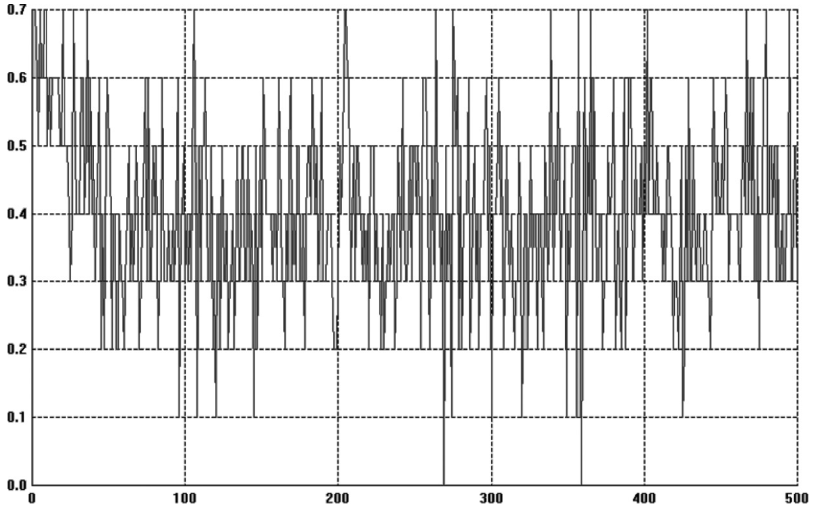

Figure 10. A diffusion trajectory for the base model.

information items, and re-organization (strong SC) after a variation of the reliability of these items. Structural change is analyzed by examining the total weight or endorsement value assigned to information items with different reliability. If the agents are learning it should happen that, while the reliability of the items is fixed, the agents will give higher endorsement value to those with reliability 1, less to those with reliability 0.5, and the lowest to those with reliability 0. Once the reliability of the items changes, reorganization must happen, as the agents should adapt their mental models to the new environment. Between a period of soft structural change and the next one, strong structural change happens, while the mental models are reorganized.

This behavior is studied by considering the differences between: (a) the total endorsement value for all the items with certain reliability (e.g., 1) and (b) the total endorsement value for all the items with another reliability (e.g., 0, or the conjunction of the other two: 0 or 0.5). In fact, three cases were considered:

I. Total endorsement value for all the items with reliability 1, vs. total endorsement value for all the items with reliability 0.
II. Total endorsement value for all the items with reliability 1, vs. total endorsement value for all the items with reliability 0.5.

III. Total endorsement value for all the items with reliability 1, vs. total endorsement value for all the items with reliability 0 or 0.5.

For the *base model*, the unreliable items are quickly discarded in favor of the reliable items (see Figure 11). However, the agents seem confused with items with reliability 0.5, as expected. In this case, the difficulty for differentiating items with reliability 0.5 is lower than when the agents are only learning, as explained below. A similar behavior is observed in the case of active agents and for the *short memory model*, although in this case learning is much faster; i.e., the selection of the best items occurs quickly. The simulation shows that the stabilization of the items used is slower when the memory has only 3 items (it takes about 16–20 days after inducing a structural change, while in the other configurations it takes around 20–40 days; and it happens about 60 days after the beginning of the simulation, compared to about 70 in the other cases). When the recommendation system is eliminated, structural change and learning is similar to that in the base model. This is understandable, as the items placed at the recommendation system are usually similar to those that the agents exchange among them.

As in the other cases, agents of learning type have difficulties to discern and classify items with reliability 0.5. Such agents also have some

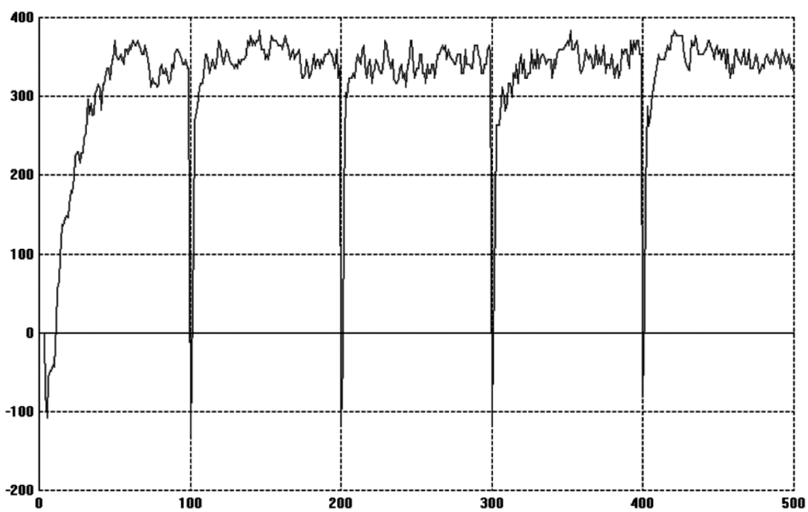

Figure 11. Relation between used items with reliability 1 and 0, base model.

upper bound for learning (sometimes this is quite poor). This is understandable, if we remember that the agents will take items from the environment and from the recommendation system only when they are active, not when they are of the learning type. In some sense, learning individuals are somewhat blind, like the passive individuals. Despite these limitations, it is shown that learning individuals behave much better than passive individuals. This suggests that, to model learning agents (the sort of agents researchers in social simulation usually simulate), it is important to consider indirect interactions, such as that provided by a recommendations system, and the interaction with the organizational environment, such as the noise that makes available to the individuals previously discarded or totally new information, which might become useful after a structural change in the environment. The simulated random influence from the environment helps the organization to avoid being trapped in an old and obsolete culture.

For the *passive model*, there is not structural change. Once the agents stop exchanging information, their learning is very limited, and the difference in endorsements depends only on the reliability assigned to the items. Once more, when noise is eliminated from the environment (as in the *no noise model*), there are important differences in structural change compared to the other configurations. Structural change is limited. Aggregating noise from the environment seems to impel structural change.

Item Weight Distribution. This section analyzes the distribution of item endorsements (or weights) in the agent population when the model stabilizes (at simulation days 100, 200, 300, 400, and 500). Weight distribution presents similarities in the base, passive, and no recommendation models (therefore, only the first one is discussed). As stated before, agents give their answer by using the endorsements scheme in all the configuration cases, but the models differ on how the agents interact. Figure 12a shows the distribution for the base model. It depicts low (close to 0) values and medium-high values (above the mean, but below the maximal value).

The distribution is different for the learning model, shown in Figure 12b. In this case, the first peak disappears. For the no-noise model (Figure 12c), the endorsements of the items considered increase as they are re-evaluated (the bad items are discarded). Usually, there is no agent whose mental model has endorsements near to zero; the distribution is

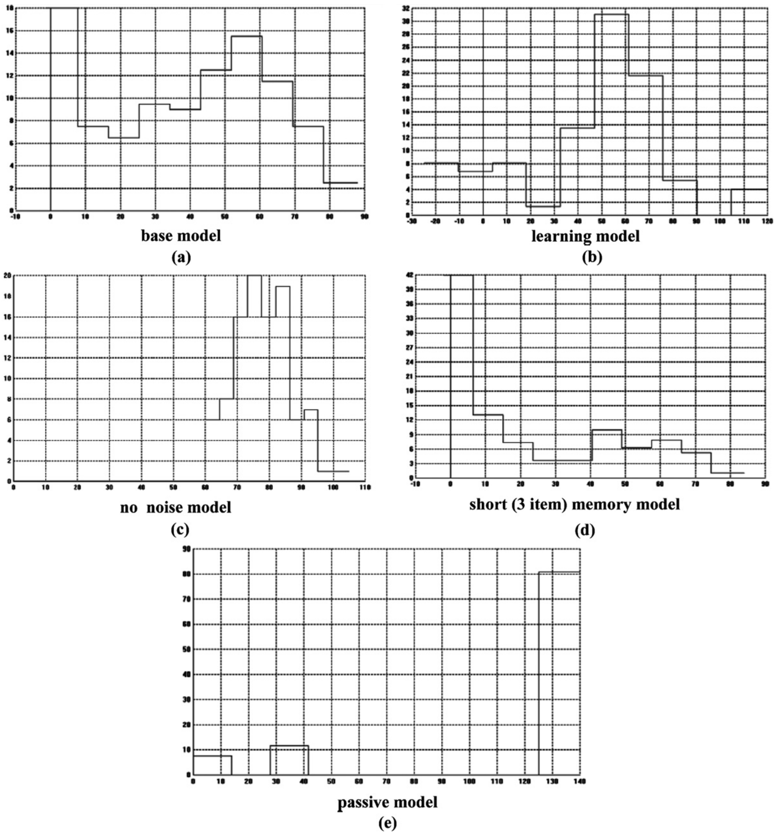

Figure 12. Endorsements distribution in the agent population.

displaced towards the right. For the short memory model, the selection of items is more rigorous, seeing that only three items are taken into account to reach a decision. As seen in Figure 12d, the peak at the right disappears. Items with low endorsement value accumulate in the first part of the graph, while the distribution is somewhat uniform for the upper three-quarters of the graph.

For the passive model (Figure 12e), the distribution shows two peaks: a few accumulate around zero, the others at the upper bound of the interval. This happens because the agents can perceive few items, and this does not change once the model stabilizes (passive agents soon stop interacting), even after a strong structural change is induced.

CONCLUSIONS AND FUTURE WORK

In this work, we have presented an agent-based model of an organization that improves Carley's model CONSTRUCT-O (Carley and Hill 2001). That model is an improvement to previous ones (Carley 1992; Carley and Svoboda 1996). In CONSTRUCT-O, the agent interaction style is either passive or active, where each agent can be assigned a mixed strategy. Agents are assigned a problem, as in our model (classifying a string by looking at a portion of it). Group formation (triads), performance, consensus, and diffusion were measured.

In the present paper, we have continued and improved Carley's model by adding a new interaction style (learning) to the agents, by using the concept of *endorsement*, and by inducing strong structural changes to the organization. For this purpose, we have added a probability of being wrong to each problem bit (thus modeling the quality of the information). After the stabilization of the organization (what we call a period of *soft* structural change) we induce a strong change by modifying these probabilities. Our results for the periods of soft change agree with Carley's model, although the stabilization in our models is much faster.

We have also experimented with other knowledge store and diffusion mechanisms, such as the "*recommender system*." Some interesting results were found. For example, individual learning (as described here) is not essential for organization performance, while active behavior is. Note that active behavior leads to "*structural learning*" within an organization, that is, changes in groups of agent collaborators. Working groups may change without damping performance, and they are more stable with learning agents. Being able to access unused knowledge (noise) becomes essential when drastic changes are made to the problem assumptions. Organization with passive agents cannot react to this kind of changes in the environment.

According to cognitive theories, learning agents, as in the case implemented for learning agents in this work, imitate the rudiments of human learning (Moss 1995; Moss et al. 1998). Active and passive interactions are also ways in which humans exchange information and learn, in accordance with the theories of cultural transmission (Carley 2001). Recommendation systems are novel technological resources to help people to deal with large amounts of information and increase organizational performance. On the other hand, noise from the environment occurs at any organization, all of which are open systems. Individual learning in our model is a combination of these different explanations of learning. Claiming larger validity for one of them in detriment of the others is not the purpose of this paper, and does not seem a defensible position. However, some of these approaches have a more solid theory behind them. A further goal, which goes beyond the reach of the present paper, is to investigate the combination of these different models that gives the best representation of human behavior and learning in organizations.

In the *learning* model, the agents develop links to the knowledge items and to other agents with which they interact. The weight of these links changes depending on the previous interactions. Finally, each agent produces an output, which depends on the knowledge items. This process resembles the working of an artificial neural network (Haykin 1999), where the agents are similar to neurons, and the endorsements are similar to the synapse strength. Thus, the metaphor of organization as a *global brain* seems appropriate. Note also that the process of training the network is similar to the simulation loop in which different problems are presented in every cycle. Our agents add and remove links to other agents and knowledge items. This is similar to a neuron generating or deleting links (axons) to other neurons. The process of modifying the network morphology to find an optimal one is called "pruning" or "brain damage" (Le Cun et al. 1990) in the artificial neural network area, and several methods have been proposed for this task.

In the future, we plan to extend the model by considering more complex knowledge structures, similar to the Internet. Ideas similar to Heylighen's (1999) could then be integrated in the model. Different organization structures and hierarchies can be modeled. Other mental models for the agents and more complex problems could also be considered. Finally, it would be interesting to analyze the algorithms proposed by the neural networks community and compare them with our handling of *endorsements*.

REFERENCES

Alfonseca, M. and J. de Lara. 2002. Two-level evolution of foraging agent communities. *BioSystems*, 66(1–2):21–30.

Bonabeau, E., M. Dorigo, and G. Theraulaz. 1999. *Swarm intelligence: from natural to artificial systems.* New York: Oxford University Press.

Booch, G., J. Rumbaugh, and I. Jacobson. 1999. *The unified modeling language user guide.* Boston: Addison-Wesley. See also the OMG's UML home page: http://www.omg.org/uml

Carley, K. 2001. Intra-organizational complexity and computation. In *Companion to organizations*, edited by Joel C. Baum. Oxford: Blackwell.

Carley, K. 1992. Organizational learning and personnel turnover. *Organization Science*, 3(1):20–46.

Carley, K. and V. Hill. 2001. Structural change and learning within organizations. In *Dynamics of organizations: Computational modeling and organizational theories*, edited by Alessandro Lomi and Erik R. Larsen. Cambridge, MA: MIT Press/AAAI Press/Live Oak.

Carley, K. and S. M. Svoboda. 1996. Modeling organizational adaptation as a simulated annealing process. *Sociological Methods and Research*, 25(1): 138–168.

Cohen, P. 1985. *Heuristic reasoning: An artificial intelligence approach*. Boston: Pitman Advanced Publishing Program.

Cohen, M., J. March, and J. Olsen. 1972. A garbage can model of organisational choice. *Administrative Science Quarterly*, 17(1):1–25.

Cyert, R. and J. March. 1959. A behavioural theory of organisational objectives. In *Modern organization theory*, edited by M. Haire. New York: Wiley.

Cyert, R. and J. March. 1963. *A behavioural theory of the firm*. Englewood Cliffs, NJ: Prentice Hall.

Gilbert, N. and K. Troitzsch. 1999. *Simulation for the social scientist*. Berkshire, UK: Open University Press.

Haykin, S. 1999. *Neural networks: A comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall.

Heylighen, Francis. 1999. Collective intelligence and its implementation on the web: Algorithms to develop a collective mental map. *Computational and Mathematical Theory of Organizations*, 5(3):253–280.

Jennings, N. R., K. Sycara, and M. Wooldridge. 1998. A roadmap of agent research and development. In *Autonomous agents and multi-agent systems*. Dordrecht, Netherlands: Kluwer.

Le Cun, Y., J. S. Denker, and S. A. Solla. 1990. Optimal brain damage. In *Advances in neural information processing systems*, vol. 2. San Francisco: Morgan Kaufmann Publishers.

March, J. and H. Simon. 1958. *Organisations*. New York: Wiley.

Masuch, M. and P. LaPotin. 1989. Beyond garbage cans: An AI model of organizational choice. *Administrative Sciences Quarterly*, 34:38–67.

Moss, S. 1995. Control metaphors in the modelling of decision-making behavior. *Computational Economics*, 8:283–301.

Moss, S., H. Gaylard, S. Wallis, and B. Edmonds. 1998. SDML: A multi-agent language for organizational modelling. *Computational Mathematical Organization Theory*, 4(1):43–69.

Newell, A. 1990. *Unified theory of cognition*. Cambridge, MA: Harvard University Press.

Newell, A. and H. Simon. 1976. Computer science as empirical enquiry: Symbol and search. *Communications of the ACM*, 19(3):113–126.

Simon, H. 1986. The failure of armchair economics [Interview]. *Challenge*, 29(5):18–25.

Simon, H. 1984. *The sciences of the artificial*. Cambridge, MA: MIT Press.

Simon, H. 1948. The proverbs of administration. *Public Administration Review*, 6(June):53–57.

Volterra, V. 1931. *Leçons sur la théorie mathématique de la lutte pour la vie*. Paris: Gauthier-Villars.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of modelling and simulation: Integrating discrete event and continuous complex dynamic systems*, 2nd ed. San Diego: Academic Press.