Artificial Life and Reactive Systems

Multi-Agent Simulation as a Tool for Modeling Societies: Application to Social Differentiation in Ant Colonies

Alexis Drogoul, Jacques Ferber

LAFORIA, Boîte 169, Université Paris VI, 75252 PARIS CEDEX 05 FRANCE
drogoul@laforia.ibp.fr, ferber@laforia.ibp.fr

Abstract. This paper presents the notion of multi-agent simulation that is based on the definition of computational agents that represent individual organisms (or groups of organisms) in a one to one correspondence. We discuss the properties of multi-agent simulation. We then present a multi-agent simulation system based on the definition of reactive agents whose behavior is governed by the selection of simple competing tasks due to stimulus's perception. An example of a simulation of an ant colony follows as an illustration of the multiple domains in which multi-agent simulation may be used.

1. Introduction

Understanding the process of emergence is important in the study of ecological and sociological systems. Our interest is the simulation of the evolution of complex systems where interactions performed between several individuals at the "micro" level are responsible of measurable general situations observed at the "macro" level. When the situation is too complex to study it is essential to recreate an artificial universe in which experiments can be conducted in a simulated laboratory where all parameters are controlled precisely. In this paper we describe a general model of simulation of complex societies based on the simulation of the behavior of its individuals. We propose then to illustrate this model of simulation with the modelling of an ant nest. One aim of this work is to understand the mechanisms of sociogenesis through simple ontogenetic factors.

2. Simulation

Simulation usually consists in artificially reproducing natural phenomena. Traditional techniques (described on Figure 1) are based on differential equations that relate global parameters to others and describe the system's dynamics. These equations have been intensely used for simulating societies but they present severe limitations:

1. **Micro to macro relationship.** One must define input and output parameters at the same level. It is then not possible to relate a global parameter such as the population size to a local parameter like the decision process of an individual.
2. **Complexity and realism of parameters**. Complexity in models leads to the definition of new parameters whose relation to reality is not obvious. Detailed models usually require complex differential equations including awkward parameters.
3. **Taking actions into account.** Numerical methods do not represent actions, i.e., activities which result in a modification of the world. They only see them by their measurable achievement or in terms of their probability to happen.
4. **Multitask behaviors and conditional task switching**. Actions cannot be considered as proceeding from decisions whose outcome depends on some conditions of the world. One can describe a hunting process by an equation that relates the number of preys to the probability for a predator to find one but this equation will not show the numerous strategies used by the predator, though these strategies have a strong influence on its efficiency.
5. **Qualitative information**. Numerical simulations cannot cope with qualitative data such as the relation between a stimulus and the behavior of an individual. These relations, though central to ethological models, are far beyond their scope.

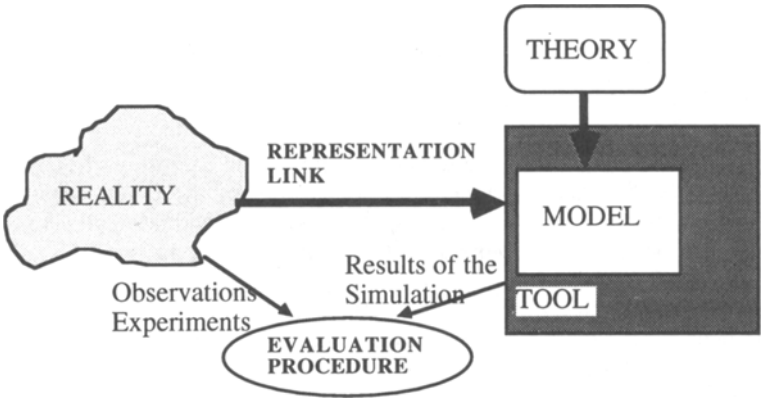

Fig. 1- the simulation process

3. Multi-agent simulation

The multi-agent simulation model is based on the idea that programs do exhibit behaviors entirely described by their internal mechanisms, namely the program instructions. By relating an individual to a program, it is possible to simulate an artificial world inhabited by interacting processes. We can then achieve simulation by transposing the population of a real biosystem to its artificial counterpart in which particular hypotheses can be explored by repeating experiments the same way than in

a real laboratory, but more easily. Each organism of the population is then represented as an agent whose behavior is programmed with all the required details (Doran & al. 1992; Hogeweg & Hesper 1985; Collins & Jefferson 1991a). Multi-agent simulations primarily help to model situations in which individuals have complex and different behaviors, and can take into account both quantitative (numerical parameters) and qualitative (individual behaviors) properties of a system in this model.

3.1. Goals of multi-agent simulation

Multi-agent simulation can be used for different purposes.
a) *Test hypotheses* about the emergence of social structures from the behaviors and interactions of each individual. This is done by testing the minimal conditions given at the micro-level that are necessary to observe these structures at the macro-level.
b) *Build theories* that contribute to the development of a general understanding of ethological, sociological and psycho-sociological systems, by relating behaviors to structural and organizational properties.
c) *Integrate different partial theories* coming from various disciplines (i.e., sociology, ethology, ethnology, cognitive psychology) into a general framework, by providing tools that allow the integration of different studies (e.g. Bousquet & al. 1992).

3.2. Multi-agent simulation and statistical analysis

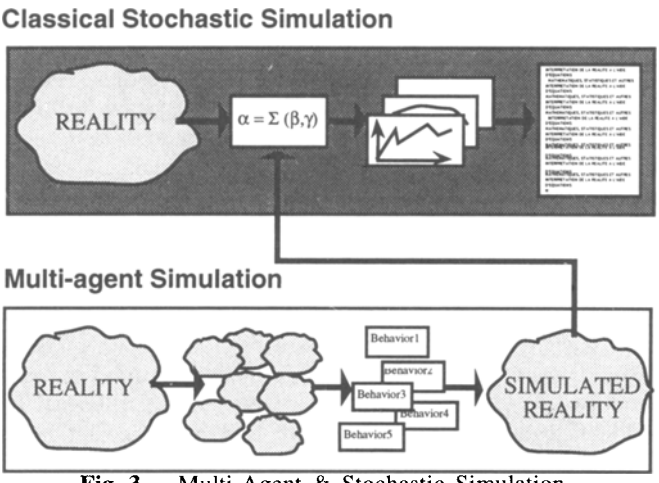

Fig. 3 - Multi-Agent & Stochastic Simulation

Multi-agents and numerical analysis are not contradictory, but they are intended to be used at different levels. Multi-agent models are used at a local level as analogical mappings of a real system. From this description, one can derive global parameters that can be studied and be incorporated into a mathematical model, as suggested in figure 3, which illustrates the differences between the two approaches.
In multi-agent simulations, numerical data and statistics are not eliminated, but they are used as evaluation procedures to compare the results coming from the simulation tool to the observation data coming from the "real" world. Thus mathematical models

are used at the macro-level whereas multi-agent simulation models are used to cross the micro-macro bridge by letting global configuration emerge from the local agent interactions.

4. A Reactive Multi-Agent Simulation System

We propose to illustrate the theory developed in the previous sections with an example of a reactive multi-agent system used to simulate an ant nest. We will first define what reactive agents are. Then, an attempt will be made to understand the difficulties encountered during the conception of reactive systems. Our aim is to show that such simulations of living systems can help both ethologists (in understanding sociogenesis through ontogenesis, for example) and computer scientists (by providing them new self-organizational paradigms).

4.1. Reactive Agents: Toward a definition...

The field of Distributed Artificial Intelligence distinguishes between cognitive and reactive multi-agent systems (Werner & Demazeau 1992). Cognitive agents have a symbolic and explicit representation of their environment on which they can reason and from which they can predict future events. Cognitive agents are driven by intentions, i.e., by explicit goals that conduct their behavior and make them able to choose between possible actions. Examples of this approach are given by J. Doran (Doran & al. 1992) which uses cognitive agents to model social changes in Paleolithic societies, and by Castelfranchi and Conte (Castelfranchi & Conte 1992) which build a theory of cognitive emergence by virtue of cognitive dependence, using the formal apparatus of Cohen and Levesque (Cohen & Levesque 1990).
Depending on one's point of view, reactive agents can then be characterized by opposition to cognitive agents, specifying the differences between them, or by reference to some human science traditions in order to explain a couple of conceptual choices. The following definition, although not exhaustive, will allow us to introduce our approach to simulation.
The behavior of a reactive agent can be first defined using a notion that directly comes from the most radical field of behaviorist psychology (Watson 1925), namely a strict "S-R" (Stimulus-Reaction) scheme. This scheme excludes *a priori* any "reasoning" between S and R, where S is considered as a particular state of the environment containing the entity, and R as a sequence of basic actions. Examples of some possible sequences can be found in (MacFarland 1981), in animal behavior, and (Anderson & Donath 1990), in artificial creatures. In this approach, the perception of the agent is simply viewed as a local sensing of a couple of stimuli (which can be visual, chemical, tactile, etc.).
If we go on using this definition, one can see that a reactive agent does not necessarily own a representation of itself, its world and other agents. Thus, unlike cognitive agents, it will not be able to act upon and control its own behavior. It will not need any memory of its experience and, considering the fact that it does not *know* the other agents, will not be able to use complex communication mechanisms. Moreover, in most cases, perception and communication will be strongly coupled. One of the key ideas from reactivity is also to get rid of anthropomorphism (as it is used in "standard AI" and, for a large part, in cognitive DAI) and explore new ways for obtaining intelligent machines (Brooks 1991; Meyer & Wilson 1991; Connah 1991)

Pros & Cons of Reactivity. In previous works like the new implementation of Pengi (Drogoul & al. 1991), we have shown that reactive agents, when correctly programmed, could exhibit really interesting behaviors emerging from simple interactions with a complex world. Their *local* reasoning allows them to deal easily with contingencies generated by modifications of the environment because they do not have any prior conceptions on the way it should be. We have also demonstrated that the use of reactive agents in problem solving could dramatically reduce the inherent complexity of such computations, from exponential to linear (Drogoul & Dubreuil 1991). Many other works, including those of R. Brooks (Brooks 1990, 1991), focus on the importance of closely coupling an entity to its environment regardless to its "intelligence", as an attempt to make it more adaptive in uncertain environments.
However, the major drawback of such agents lies in their hard-wired behavior. Their lack of flexibility in both perception and reasoning often prevents them from doing the best choice at the best time. We cannot expect a robot that does not sense cars to avoid them when crossing a road...

The More ... The Merrier. Looking more closely to applications made up of reactive agents clearly shows two things: (1) The functionalities are always performed by a group of agents, be they organized or not; (2) The lack of flexibility of each agent (as shown above) is compensated by either an important redundancy in the population or a good complementarity between the agents. (Steels 1989) provides a typical example of such systems and shows the robustness of this approach within uncertain environments. Another good example can be found in (Brooks & Flynn 1989). In fact, the adaptive behavior of their robots emerges from the interactions between tiny "reactive agents", namely the layers, that do not exhibit any sort of flexibility by themselves. The flexibility is to be found at the population level, i.e., in the robot behavior. This level is very organized, through the subsumption architecture, and each agent is complementary to the others. At their turn, the robots themselves can be considered as reactive agents interacting with each other and the environment to perform the whole required functionality. At this level, redundancy becomes the key answer to the complexity of the world.
It is quite a simple task to build up a reactive agent's architecture. It becomes more difficult to provide them with "good" behaviors and "good" capacities of interaction that will enable the population to be efficient. As pointed out by (Brooks 1991; Steels 1991), obtaining emergent specific tasks from a population of agents requires to understand the link between the behavior of a single agent and the global observed behavior of the society. This difficulty ("crossing the micro-macro bridge") is shared by numerous sciences, including but not limited to biology, ethology, economics, physics, philosophy, ethnology, ecology. Though each field has developed its own theories and concepts, a couple of common ideas can be identified: self-organizing processes (Atlan 1972; Varela 1983), swarm intelligence (Deneubourg & Goss 1989), and, in computer science, emergent functionalities (Steels 1991) or emergent behaviors (Wavish 1991).

Don't we miss a Methodology ? Within the reactive agents' field, two ways of "crossing this bridge" can be distinguished. The first consists in a sort of "trial and error" methodology, merely based on empirical studies and (successful) experience. The idea is to provide the agents with simple behaviors, to put them in an environment, let them interact and observe the global behavior of the system. When

this behavior does not fit with what was expected, the agents' behaviors have to be changed. The intuitive part of this methodology prevents it to be widely applicable in different domains.
The second, mainly present in the field of Artificial Life, consists in reproducing and simulating distributed living systems that seem to fit with the desired specifications. From this point of view, simulation is considered as a bias to understanding self-organized phenomena that can be found in our natural world. Studying the way these systems solve complex problems (exploiting uncertain environments, catching preys, building nests, cooperating, etc.) using simple "agents" may be a good indication on how to build artificial agents for solving similar problems (such as ore collecting under the sea or on another planet, for example). The advantages of this methodology are: (1) It can rely on almost a century of ethological researches; (2) Ethologists, biologists and ecologists are in search for new simulation paradigms (see part **3**) and then ready to collaborate more closely with computer scientists.
A couple of reports has been published in the last five years on such works (Theraulaz & al. 1991; Hogeweg & Hesper 1985), but this research is clearly on its early stages and we still miss a general theory. An important work is then to be done on empirical studies. This is the reason why we have developed a computer-based system called EthoModeling Framework (EMF), which provides items and tools for simulating the 'life' of populations of agents, be they simulations of animals or artificial entities.

4.2. EthoModeling Framework

The design of this system has been realized by taking two concurrent needs into account: on one hand, we needed a system allowing the testing of various hypotheses on the design of reactive agents, including behavior selection, learning, reinforcement, communication and so on, in order to study the impact of these choices on the global behavior of a society. So we decided to use an object-oriented methodology for the implementation of EMF, in order to allow heterogeneous agents to cohabit inside the same simulation, even if their internal mechanisms are different. On the other hand, for the reasons described above, we primitively conceived EMF as a modeling system, simple enough to be used by non computer scientists, in order to implement multi-agent simulations of living entities and compare the modeled societies to the real ones. In this paper, we will focus on the modeling and simulation part of the system rather than on the purely DAI researches that have been conducted with it (which will be presented in (Drogoul & Dubreuil 1992)).

Implementation. EMF is implemented under Actalk (Briot 1988), a language of actors under Smalltalk-80. Each agent is viewed as an actor-object (i.e., an object embodied within an actor that allows it to work in an asynchronous way). As pointed out by (Agha 1986), actors are well suited for the design of dynamic models because they implement the inherent parallelism of such simulations. EMF provides the programmer or user with a domain-independent kernel that rules the default internal functioning of the agents and the interactions between them and their environment. Its main part is a class named EthoBehavior. The simulation entities are then defined as instances of classes that inherit from EthoBehavior. Each class represents a particular species of agents (with its own features), each instance an individual in this species. A model (or a simulation) is then defined by a set of classes that respectively determine the environment and some populations of agents.

Basic Environment. The environment is defined as a large set of entities that are called places. The places are squares of the same size (it defines the granularity of the environment that can be, of course, modified). Places know at every time which agents are lying on them. They know their absolute position <x, y> (in a discretized spatial representation depending on the *granularity*) and the places belonging to their neighborhood, which are given by the Von Neumann formula:

$$neighbors(\text{place}\langle x,y\rangle) = \{p \mid p = (\text{place at } \langle x+a, y+b\rangle), -1 \le a \le 1, -1 \le b \le 1, p \ne \text{place}\} \quad (1)$$

Places are divided into two categories: free places and obstacles (see figure 4). The main difference between them is that obstacles cannot accept agents and do not propagate stimuli. But it is possible to define particular places that accept a limited number of agents, places already provided with some stimuli or some agents. As for the agents, new places are simply defined by inheritance from existing ones.

Basic Communication. EthoBehavior does not define any intentional communication mechanism between the agents. In fact, they do not own any representation of each other and cannot send messages. But agents can communicate in a very indirect manner by propagating their signature(s) in the environment. This communication mechanism is called non-intentional because it can be interpreted in many ways by other agents. For instance, the signature of an ant will activate a friendly behavior from another ant and an aggressive one from a predator. We state that each agent owns *personal stimuli*, i.e., a set of "pheromone-like" signals identifying it (pheromones are external hormones mainly used by social insects for their communication). When its state changes, the place on which the agent lies collects its *personal stimuli* and propagates them to the adjacent places. A stimulus is a doublet <*name*, *strength*> where *name* is the identifier of the stimulus and *strength* the value that will be propagated by the place (This *strength* is computed by the agent and defined for each class because it is highly model-dependent). Places store these stimuli in a dictionary whose keys are the stimuli names.

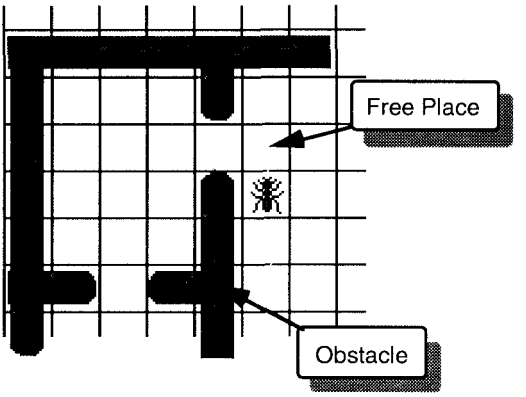

Fig. 4 - Basic Environment

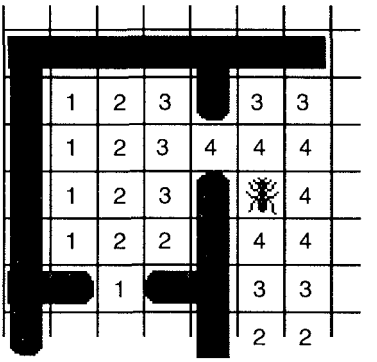

Fig. 5 - Example of Propagation

The diffusion of the *stimuli* depends on the places. Each place defines a *propagation function* called $f_p(v)$ where v is the strength of the stimulus to be propagated. When asked to propagate a stimulus s, a place calculates $v'=f_p(strength(s))$, stores v' in its

stimuli (with the name of *s* as key) and asks its neighbors to propagate a new stimulus *s'* with the same name as *s* and *v'* as strength. An example of propagation is shown on figure 5, with a stimulus of initial strength 5.
A place can store only one stimulus of a given name. When two stimuli from different sources arrive at the same time on a place, only the greater is taken into account (the propagation of the other is stopped). The default algorithm of the places for propagating a stimulus can be written very simply. Let *sn* be the stimulus to be propagated and *ss* its value. Let *stimuli(p)* be a function answering the names of the stimuli contained in *p*, *strength(p,sn)* a function answering the strength of the stimulus named *sn* in *p* and *add(p,sn,ss)* a function adding in *p* the stimulus named *sn* with *ss* as strength.

```
propagate(p,sn,ss) = if sn ∈ stimuli(p) then
                        (if f_p(ss) > strength(p,sn)) and (f_p(ss) > 0) then add(p,sn,f_p(ss));
    for each p' ∈ neighbors(p) do propagate(p',sn,f_p(ss)).
```

The *propagation function* of default free places is $f_p(v) = v - 1$. For obstacles, $f_p(v) = 0$ (they do not propagate stimuli). While this propagation is performed locally, it is possible to define places that propagate stimuli in a different way. Some can add noise ($f_p(v) = \mu(v - 1)$, $0 < \mu \leq 1$), some can slow them down ($f_p(v) = v - k$, $k > 1$), some can amplify them ($f_p(v) = v + k$, $k \geq 1$).
From a global point of view, this propagation results in a gradient field emanating from the position of the agent (Steels 1989). If *s(k,p)* represents the value of a stimulus named *k* on a place *p*, the diffusion process can then be formulated by the following equation:

$$s(k,p) = f_p(\max_{p' \in \text{neighbors}(p)} (s(k,p'))) \qquad (2)$$

The key features of this indirect communication between the agents are as follows:
• Because gradient fields depend on the structure of the environment, they induce its implicit topology (Drogoul & al. 1991). If an agent follows a gradient, it automatically bypasses obstacles that do not transmit stimuli (if we assume they also prevent the agent from moving on them).
• It provides the agents with a complete digest of what could interest them in their environment. In that way, they just need a simple domain-independent sensor system able to collect a local list of stimuli.
• It allows an agent to be reactive and opportunist. For instance, take the case of an *ant* following a gradient named *#food*, propagated by a *deadFly*. If the *deadFly* disappears, the *ant* will interrupt its behavior because the stimulus will have vanished. On the other hand, if another *deadFly* propagates a stronger stimulus while the *ant* is following the first one, the *ant* will spontaneously change its direction to head to the latter.
• Finally, it does not prevent agents from choosing other forms of communication. As a matter of fact, it is well stated that many animals communicate through the propagation of chemical signals (Deneubourg & Goss 1989), although they can do this more easily using intentional ways (for instance, tactile cues).

Agents Structure. The context in which we place the structure of the agents is that of the behavior-based artificial creatures (Maes 1991). An agent is seen as consisting of a set of behaviors that we call *tasks* among which one can be active at a

time. The selection and the suspension of a task are entirely *stimuli-oriented.* As in (Tyrrell & Mayhew 1991; Schnepf 1991) the term *task* refers to a set of behavioral sequences as opposed to the low-level actions (moving, and so on) that we call *primitives*. Moreover, some of these low-level actions are of no earthly use to us (for instance, obstacle avoidance) because the *stimuli-oriented* communication induces a topology of the environment. The agents are also basically provided with a mechanism of behavior reinforcement. The subclasses of EthoBehavior are divided into two sets: *abstract* classes and *concrete* classes. Abstract classes cannot be instantiated. They are used to define the primitives and knowledge associated with them. Concrete classes, whose instances are the agents of the simulation, inherit these primitives and use them to define the tasks (behaviors) of their agents. An example of such a hierarchy is provided on figure 10, within the MANTA project. Primitives represent low-level behaviors mainly related to "physiological" possibilities. We assume that they cannot be decomposed into behaviors of lower level. Agents of the same class share the same primitives.

What Does an Agent Know ? Although a couple of special classes may define new knowledge for their agents, the basic knowledge of an agent is reduced to its *personal environment*. We call personal environment a set of places among which it can collect stimuli. We will see that an agent does not need to know more about its world because its behavior is seen as a stereotyped response to a *stimulus*. Moreover, the selection and duration of the response are entirely governed by the intensity of the *stimulus*. The basic environment of an agent is constituted by its place. Other places can be of course added to increase the perception of the agent.

The Behavior of the Agents. As said above, the behavior of an agent is defined by a set of tasks, each of them being related to a particular stimulus name. Tasks usually encapsulate a sequence of primitives that is built up by the programmer. At the concrete class level, only the primitives inherited from the abstract superclasses are available for building tasks. From an ethological point of view, tasks are close to *fixed-action patterns* (although some can be viewed as *reflexes* or *taxes* (Beer 1990)).

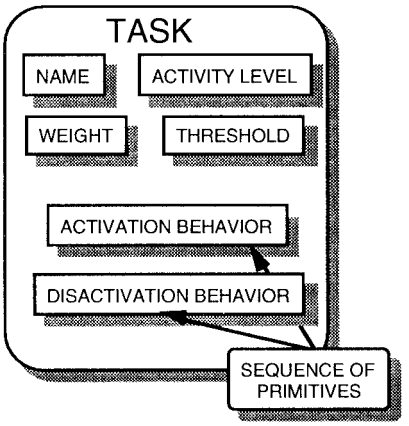

Fig. 6 - Task Definition

Each task must be provided with:
- A *name*, usually the name of the stimulus that triggers it.
- A *weight*, which specifies the relative importance of the task inside the agent. This number can be modified by the reinforcement process (see part 4.4).
- A *threshold*, under which the task will not be triggered by a stimulus (its strength multiplied by the weight of the task must surpass it).
- An *activity level* computed when the task becomes active.
- Two *sequences of primitives*, executed when the task becomes active and inactive.

A tool, called the Task Browser, can be used to specify the sequences of primitives encapsulated in the task. As a matter of example, consider the task named *cocoon*, taken from the class AntAgent, on figure 7. The Browser allows to work on all the concrete classes and indicates, for each of them, the tasks that have been defined and the primitives that are available by inheritance. A task is then simply built up by selecting the primitives and placing them on the BEHAVIOR window. Tasks can also be copied from one class to another, removed, added without any difficulty.

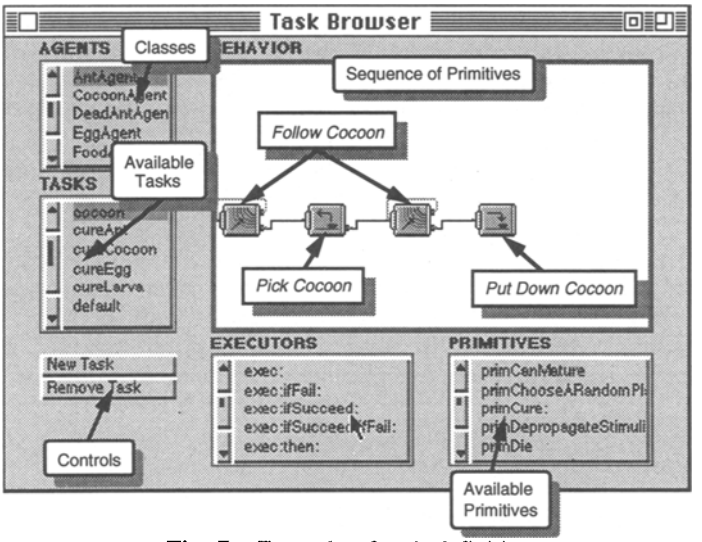

Fig. 7 - Example of task definition

The Task Selection Process. An agent knows the *current task* in which it is involved. When this task executes one of its *primitives*, the agent performs the *task selection process*, to determine if a task is more appropriated to its environment than the current one. This process is made up of three steps:

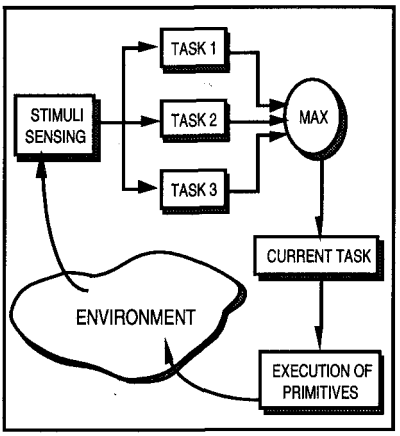

Fig. 8 - Task Selection Process

(1) *Sensing*: the agent collects the *stimuli* and eliminates those that do not match with a task name.
(2) *Selection*: the agent computes the *activation level* of each task by multiplying the *strength* of the stimulus and the *weight* of the related task. Tasks whose *activation level* surpasses their *threshold* and the *activity level* of the *current task* are selected.
(3) *Activation*: If some tasks can be activated, the agent chooses the one whose *activation level* is the greatest. Then it deactivates the *current task* and activates the selected one. When no tasks have been selected, the *current task* simply goes on.

When a task becomes active, it is placed in the current task of the agent and its activity level is initialized to the value of its activation level. Then, the agent performs its *activation behavior*. When a task is deactivated, the agent performs its *deactivation behavior* and zeroes its activity level.
Agents must be provided with a special task called *default*. This task is always viewed as activated (it does not depend on any stimulus), with an activation level equal to 1. It is then chosen when no tasks have been selected and when the activity level of the current task becomes nil. This task specifies the default behavior of the agent when its environment is not particularly attractive.

The Behavior Reinforcement Process. "Real" creatures, although they are often provided with preprogrammed behaviors, exhibit flexible mechanisms of behavior selection. They can take former experiences of interactions with their environment into account when choosing their future behavior. The activation of a behavior also integrates non-environmental conditions such as motivations. Our aim is not to reproduce the numerous types of plasticities (Beer & al. 90) that have been studied in animals, but to implement in EthoBehavior a simple mechanism of *behavior reinforcement*.
Behavior reinforcement has been observed in many animal species and particularly well studied in social insects as a mechanism of social organization (Theraulaz & al. 1991). In our perspective, it is simply defined by the sentence: *"The more an agent performs a task, the more it will be able to perform it again"*. The *reinforcement process* takes place just after the current task has been deselected and increases its weight by:

$$iw = weightIncrement * (1 - (activity\ level / activation\ level)) \qquad (3)$$

If the current task has stopped naturally, its activity level will be nil and its weight will be increased by the total amount of weightIncrement. Otherwise, the increase will be relative to the duration of the task.

Summary. Before presenting the MANTA project, we would like to replace EMF in the context of artificial life and DAI researches. The structure of our agents happens to be close to that proposed in (Schnepf 1991), although we did not know these works when beginning to code our framework. The principles of behavior selection differ from the approach of (Maes 1990, 1991) or (Brooks 1990) in that our *tasks* are not linked by predecessor/successor nor activator/inhibitor links. Like (Steels 1989) we use a non-intentional environment-driven type of communication in which autocatalytic or amplification mechanisms such as those described in (Prigogine & Stengers 1984; Deneubourg & al. 1986) can be reproduced. The behavior reinforcement idea proceeds from the same principles than (Deneubourg & al. 1987; Theraulaz & al. 1991) but we do not use stochastic nor mathematical functions to encode it. We do agree with (Agre & Chapman 1987) in that an agent has to make rapid responses to environmental events, but we do not agree with their definition of situation, based on *aspects*, which appears to be too static for our needs. In our agents, a situation, i.e. a trigger for a behavior, is dynamically computed from the conjunction of both the environment's state, namely the stimuli, and the agent's state, namely the weights of its tasks, and we do not have to write a *routine* for each possible situation. Although we are interested in making cooperation evolve among

our agents, we see it in a more emergent way than (Hickman & Shiels 1991). Thus, we did not implement any *mutual intelligibility* in them. Our aim is to make cooperation emerge from both the distributed division of labor due to the behavior's reinforcement and the competition between the agents for performing a task.

5. The MANTA (Modelling an ANTnest Activity) Project

MANTA is the first project implemented under EMF. Its aims are to model the behaviors of simple *Ectatomma ruidum* ants and to show that this model is able to generate a division of labor close to those observed in *Ectatomma ruidum* societies.
Ants' colonies are a fascinating model for people interested in the concepts of emergence and self-organization. Consequently, many studies regarding their social organization have been published in the fields of ethology, sociobiology or ecology. More recently, people involved in other research areas have begun to investigate the domain in order to obtain new models for understanding the emergence of "intelligent" behaviors: "*an Ant viewed as a behaving system is quite simple, the apparent complexity of its behavior in time is largely a reflection of the complexity of the environment in which it finds itself (...)*" (Simon 1969). In (Hofstadter 1979) and other works, the activity of a nest is compared to the activity of a brain. (Prigogine & Stengers 1984) explore autocatalytic mechanisms in ants in the field of dissipative structures. (Steels 1989; Deneubourg & al. 1991) use ant-like robots to study *emergent functionalities*. Ant-like agents are employed in the AntFarm simulation of (Collins & Jefferson 1991b) to study the evolution of their colonies. Many works also focus on the modeling of social insects' societies: (Hogeweg & Hesper 1983, 1985, 1991) use the MIRROR framework to model bumble bees societies evolution, (Theraulaz & al. 1991) models the division of labor in *Polistes* wasp colonies, (Deneubourg & al. 1987) proposes a model of learning aiming to reproduce the *Neoponera Apicalis* ant foraging.
Compared to these models, MANTA appears to be more ambitious. As we will see below, the agents have been provided with all the ants' behaviors described in (Corbara 1991). The idea was to translate directly the ethological knowledge into our entities. All the creatures that can be found inside an ant nest have been modelled (ants, of course, but also eggs, larvae, cocoons) as well as some environmental factors (humidity, light). Time has been taken into account and the whole life cycle of an ant, from birth to death, can be observed in the system.

5.1 Ectatomma Ruidum

The species modeled, *Ectatomma ruidum*, has a geographical distribution extending from southern Mexico to northern Brazil. The colonies contain a relatively small number of ants (less than 300). This species is usually monogynous (i.e., one queen) and a clear dimorphism distinguishes the queen from the workers. But no physiological distinctions can be found between the workers. The social organization of this species has been fully studied in (Corbara & al. 1986, 1989; Lachaud & Fresneau 1987) from the foundation of a society to its maturity, through an individual analysis of the behavior of each ant, the establishment of an inventory of behavioral acts, combined into behavioral categories and the determination of "functional groups"

by comparing and aggregating the behavioral profiles of the ants. From the point of view of our study, Ectatomma ruidum has two major properties:

• Like *Polistes* wasps (Theraulaz & al. 1991), the ants are able to perform a wide range of tasks - see the readjustments of individual behaviors following an experimental sociotomy in (Lachaud & Fresneau 1987) - but show a *differential reactivity* to *stimuli* depending on their behavioral profile. We hypothesize that it is directly related to a notion close to *behavior reinforcement.*

• The stability of distribution into "functional groups" - though a variability within these groups has been shown in (Corbara & al. 1989) - among numerous colonies allows the comparison between the social organization obtained in the model and that observed in the reality. Furthermore, this comparison is facilitated by the possibility of using the same tools in both cases.

5.2 MANTA's Environment

The environment reproduces a nest similar to those employed in laboratories. This nest can be modified during the simulation in order to test the influence of topology on a population. Figure 9 shows the main simulation window, which allows to modify parameters during the progress of a simulation. Some labels indicate the different agents involved in it.

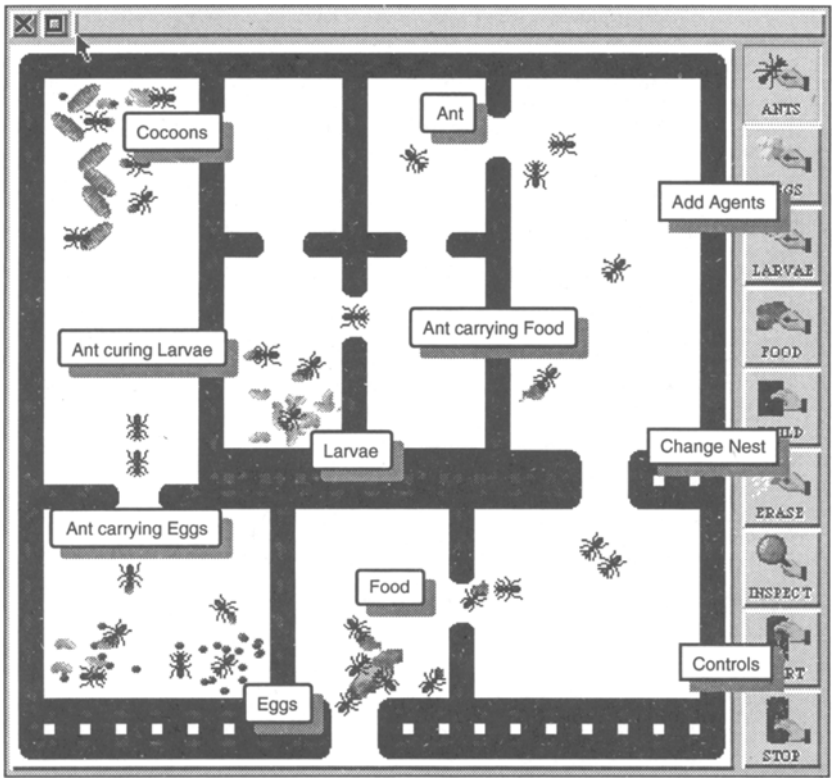

Fig. 9 - The Main Simulation Window and its Control Panel

5.3 The Agents

As said above, the classes that will represent the agents are subclasses of abstract classes (themselves subclasses of EthoBehavior). Each of these subclasses implements a set of primitives and stimuli, related to a particular capacity of the agent. Figure 8 shows the MANTA hierarchy of classes. We will briefly describe each of the abstract classes and then introduce the concrete classes of MANTA.
InterfaceBehavior implements the protocols of the agents' user-interface capacities (graphical trace, inspection, etc.). LocatedBehavior provides all its sub-instances with the ability to be in an environment and to act on it (propagating stimuli, for example). CuringBehavior implements the primitives needed by agents that will have to cure other agents or receive cares from them. FeedingBehavior implements primitives needed by agents that will have to feed themselves, feed another agent or be fed by one. MaturingBehavior provides its sub-instances with the capacity of growing old (and consequently, die!). The notion of time is implemented in this abstract subclass, as an internal stimulus. All its subclasses must define some domain-dependent pieces of knowledge, such as the average expectation of life of their instances. MovingBehavior gives them the possibility to move in their environment. SensingBehavior implements two primitives for following or fleeing a gradient field. CarryingBehavior implements the primitives needed for carrying other agents.
Depending on one's needs, the simulation concrete classes will then take place as subclasses of one or another abstract class.

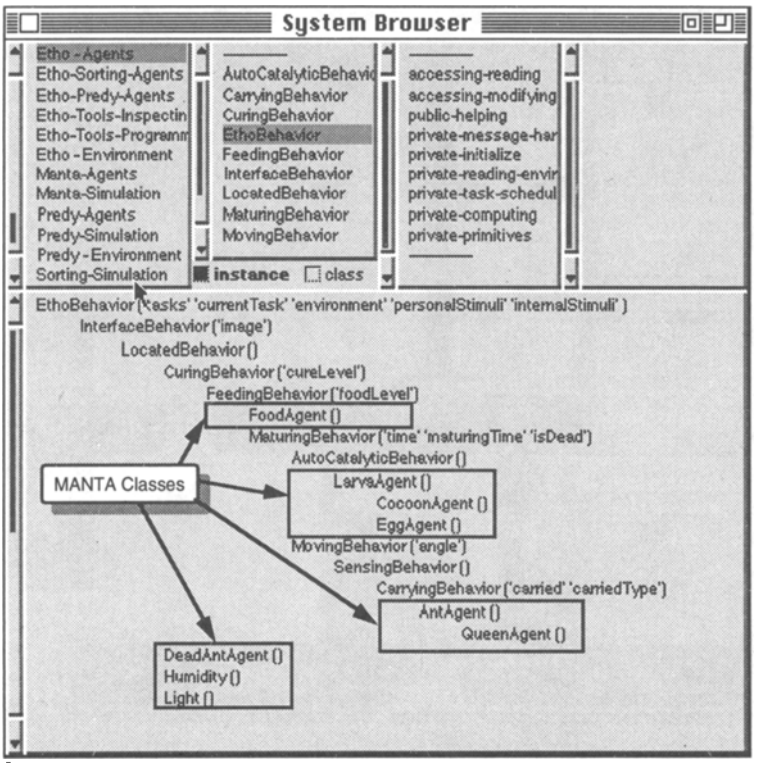

Fig. 10 - MANTA's Hierarchy of Classes

Environmental Agents. Each environmental agent represents an environmental factor, such as light or humidity, that can be placed anywhere and propagate its particular stimuli. For example, light agents are used to create the difference between the outside and the inside of a nest. Light agents are also intended for simulating the alternance between day and night (by changing the strengths of their stimuli). Because they cannot move nor sense, their classes (LightAgent, HumidityAgent) are direct subclasses of LocatedBehavior. These agents propagate stimuli respectively named #light and #humidity.

Spy Agents. Spy agents are intended to borrow information from the simulation and translate them into statistical, numerical or textual data. These agents can be added or removed at any time during the progress of a simulation. They can manage special windows displaying text or graphs, save their data on files or perform complex data analysis. As a matter of example, each of the spy agents used in the simulations reproducing a laboratory nest manages one of the tools used by ethologists doing real experiments (recording of the behaviors, multivariate analysis techniques, etc.). Spy agents do not propagate any stimuli.

Brood Agents. The agents that compose the brood are the eggs, the larvae and the cocoons. They represent the three steps of growth needed for an egg to become an ant. If we refer to ethological and biological data (Corbara 1991), these agents will just have to be fed, cured and carried by other agents to be satisfied. In the model, they are instances of EggAgent, LarvaAgent and CocoonAgent, subclasses of MaturingBehavior. The propagated stimuli are respectively named #egg, #larva and #cocoon. Due to their inheritance of classes that implement implicit stimuli, they also propagate stimuli named #cureEgg, #cureLarva, #cureCocoon, #maturingLarva and #hungryLarva. The strengths of these stimuli are directly related to some state of the agent (its foodLevel defined in FeedingBehavior, for example).

Ants and Queens. These are certainly the biggest agents in the simulation. As a matter of fact, we have provided them with all the behaviors described in (Corbara 1991) and depicted on figure 9. The stimuli that could be present in the environment (be they propagated by other agents or itself) are shown on the left side, while the tasks directly related to them are on the right side. These links between stimuli and tasks have been implemented after long discussions with ethologists, because it is difficult to verify their existence in nature. Some of them are obvious (like the link between the emission of a particular pheromone by the larvae and the "feed larvae" behavior), whereas others (like the link between maturingLarva, propagated in the model by a larva that is about to mutate, and the "care of larvae" behavior) are just hypotheses.
Due to the place of the two classes AntAgent and QueenAgent at the bottom of the hierarchy of classes, their agents will be provided with a lot of primitives (almost twenty), including all the moving, feeding, curing and sensing primitives. Although the two classes share the same properties (QueenAgent is a subclass of AntAgent), it has been decided to distinguish the queen from the ants, with respect to ethological observations. In nature, queens live much longer than workers (many years instead of a couple of months) and seem to be more sensitive to the brood than them. In the model, it simply results in: (1) providing queens with a longer expectation of life (which is a domain-dependent knowledge needed by MaturingBehavior); (2) arbitrarily

increasing the weight of the brood-dependent tasks (see figure 9) during the process of instanciation. Of course, the main difference between queens and workers is that queens are able to lay eggs, whereas workers are usually sterile. So QueenAgent has also been provided with a different default task, which simply consists in creating and putting down some EggAgents (the default task of AntAgent is to move randomly).

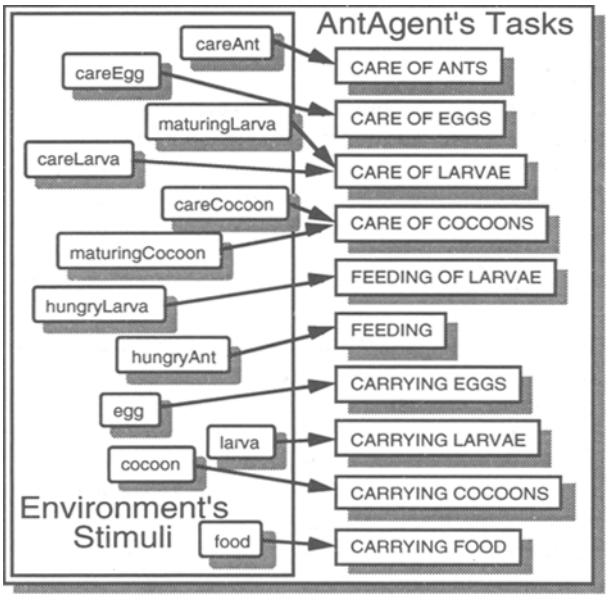

Fig. 11 - The tasks of AntAgent and QueenAgent

The creation of a task for a class of agents is a very simple process. A couple of specialized tools (already seen in part 4.2) are available and allow the visual programming and modification of any task. Let us take an example. Imagine that we have decided to make ants react to the "hungryCocoon" stimulus, which is present in the model (due to the inheritance of CocoonAgent from FeedingBehavior) but does not have any ethological sense. Creating this task simply consists in selecting AntAgent in the browser and choose the 'New Task' option. A dialog will appear, asking for some parameters (the name of the stimulus that triggers it, its weight, threshold, etc.). Then, the visual programming of the task may begin. A task is always written (and read) from left to right. In the example, the sequence of primitives could be translated by: "follow the gradient hungryCocoon then, if you have food, feed the agent that propagates it, otherwise follow the gradient named food (if there is any), pick some food, follow the gradient named hungryCocoon and feed the agent that propagates it". As soon as a primitive cannot be executed (a gradient has vanished, for example), the task is stopped.
In reality, the ants' tasks are usually simpler than this one (see figure 8 for an example of a 'real' task) and use less than four different primitives.

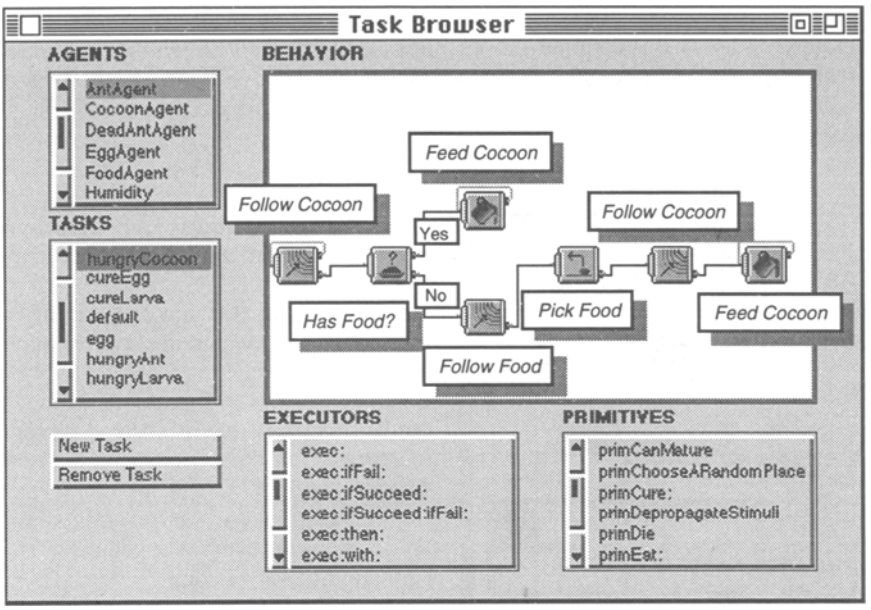

Fig. 12 - An imaginary task for feeding cocoons

5.4 Experiments

In earlier versions of MANTA (Drogoul & al. 1992a, 1992b), we have conducted a lot of experiments using simpler ant agents, only provided with three behaviors (care of eggs, larvae, and foraging). The case study was composed of 30 identical ants, 50 larvae, 50 eggs and 50 pieces of food disseminated in the nest and outside. As our purpose was to study the emergence of a division of labor, we did not add a queen. The time spent on each task by each ant has been then cumulated throughout the simulation (see figure below). The simulation ended when the eggs, larvae and pieces of food were totally sorted into three separate clusters.

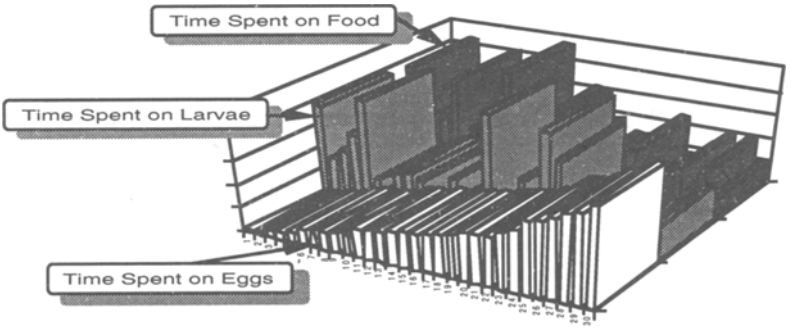

Fig. 13 - The differentiation of individuals

Although this example did not intend to simulate a real nest, a division of labor characterized by five functional groups appeared within all the simulated nests, as shown on figure below (this kind of figure is called a *socioethogram*:, or simply an *ethogram*. It results from the mutivariate analysis and clustering of the data concerning the time spent by each ant on each task):

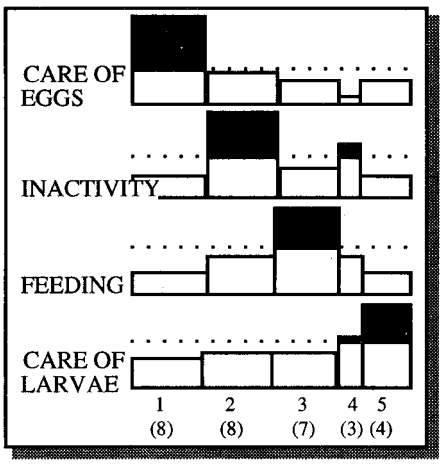

Fig. 14 - Functional Groups

Eggs nurses (Group 1, 8 ants): high level of care of eggs and low level of inactivity.
Unspecialized (Group 2, 8 ants): high level of inactivity, and mean level in other activities.
Feeders (Group 3, 7 ants): high level of feeding activities, important level of inactivity.
Larvae-Inactive (Group 4, 3 ants): high level of care of larvae and inactivity and low level of care of the eggs.
Larvae nurses (Group 5, 4 ants): high level of care of larvae and low level in other activities.

Of course, these divisions of labor appeared to be simpler than those observed in the reality. The reason is that we provided the ants with a relatively small set of behavioral capacities. However, this kind of structuration remained very stable throughout the many simulations we have conducted.
With the new version of MANTA, more interesting experiments can be made. Now that we have implemented the whole behavioral repertoire of ants and modeled all the agents that can inhabit a nest, the comparison between the model and reality is simply a matter of time. Several experiments are being conducted in our laboratory and their results should be available soon. These experiments include:
• A sociogenesis, which consists in beginning the simulation with only one queen and then letting the whole society emerge. Real ant nests are funded this way and it will be interesting to compare the different phases of evolution of the model to those described in (Corbara 1991).
• Sociotomies, which consist in cutting an adult society into two smaller societies in order to observe the readjustments of individual behaviors. Usually, an experimental sociotomy separates a group of highly specialized individuals from the rest of the colony, providing each sub-society with the same number of eggs, larvae and cocoons, as well as the same amount of food. An example can be found in (Lachaud & Fresneau 1987).

6. Conclusion

In this paper, we have presented a general model of multi-agent simulation and compared its paradigms to those of classical simulation. We have also expressed the

difficulties encountered by the researchers who want to make and use reactive DAI systems. This theoretical section has been followed by an example of such systems, called EMF. It is a reactive multi-agent system that provides tools for modelling societies of simple agents. We have pointed out the need in both ethology and DAI for such simulations. The presentation of EMF has been followed by the description of the MANTA project, in which we aim at modelling an entire ant society. The implementation of the agents and some early experimental results have been presented.

References

G. Agha (1986) "Actors - A model of Concurrent Computation for Distributed Systems", MIT Press.

P. E. Agre & D. Chapman (1987) "Pengi: an implementation of a theory of activity"*in* "Proceedings of the sixth National Conference on AI", 1987, pp. 268-272.

H. Atlan (1972) "L'organisation biologique et la théorie de l'information", Hermann, Paris 1972 (in french).

R.D. Beer (1990), *Intelligence as Adaptive Behavior*, Academic Press.

F. Bousquet, C. Cambier, C. Mullon & J. Quensiere (1992) "Simulating Fishermen Society", Simulating Societies Symposium, University of Surrey, April 1992.

J.P. Briot (1988) "From Objects to Actors, study of a limited symbiosis in Smalltalk-80" , LITP Report 88-58RXF.

R. Brooks (1990) "Elephants Don't Play Chess" *in* "Journal of Robotics and Autonomous Systems", Volume 6, p. 3-15.

R. Brooks (1991) "Intelligence without Reason" *in* "Proceedings of IJCAI'91", Volume 1, pp. 569-595.

C. Castelfranchi & R. Conte (1992) "Mind is not enough: Precognitive bases of social interaction", Simulating Societies Symposium, April 1992.

P.R. Cohen & H.J. Levesque (1990) "Intention is Choice with Commitment", in Artificial Intelligence, 42.

D. Connah (1991) "Why we need a New Approach to the Design of Agents", AISBQ n°76, Spring 1991.

R.J. Collins, D.R. Jefferson (1991a) "Ant Farm: Towards Simulated Evolution", *in* "Artificial Life II", C. Langton Ed, Addison-Wesley, 1991.

R.J. Collins, D.R. Jefferson (1991b) "Representation for Artificial Organisms" *in* "From Animals to Animats", MIT Press, page 382.

B. Corbara, D. Fresneau, J.P. Lachaud, Y. Leclerc, G. Goodall (1986) "An automated photographic technique for behavioural investigations of social insects" *in* "Behavioural Processes" 13, p. 237-249.

B. Corbara, J.P. Lachaud, D. Fresneau (1989) "Individual Variability, Social Structure and Division of Labour in the Ponerine Ant *Ectatomma Ruidum* Roger (Hymenoptera, Formicidae)" *in* Ethology 82, p. 89-100.

B. Corbara (1991), "L'organisation sociale et sa genése chez la fourmi Ectatomma Ruidum Roger", PhD Thesis, Université Paris XIII, 1991 (in french).

J.L. Deneubourg, S. Aron, S. Goss, J.M. Pasteels, G. Duerinck (1986) "Random Behaviour, Amplification Processes and Number of Participants:

How they Contribute to the Foraging Properties of Ants" *in* Physica 22D, North-Holland, Amsterdam, p. 176-186.

J.L. Deneubourg, S. Goss, J. Pasteels, D. Fresneau, J.P. Lachaud (1987) "Self-organization mechanisms in ant societies (II): learning in foraging and division of labor" *in* "From Individual to Collective Behaviour in Social Insects", Experientia Supplementum vol. 54, Birkhäuser Verlag, Basel, p. 177-196.

J.L. Deneubourg, S. Goss (1989) "Collective patterns and decision-making" *in* "Ethology Ecology & Evolution" 1: p. 295-311.

J.L.Deneubourg, S.Goss, N.Franks, A.Sendova-Franks, C.Detrain, L.Chretien (1991) "The dynamics of collective sorting Robot-like Ants and Ant-like Robots" *in* "From Animals to Animats", MIT Press, p. 356.

J. Doran, H. Carvajal , Y.J. Choo and Y. Li (1991), "The MCS Multi-Agent Testbed, Developments and Experiments", *in* "Cooperating Knowledge Based Systems", S.M. Deen Ed., Springer Verlag, 1990.

J. Doran, M. Palmer, N. Gilbert (1992) "The EOS Project: Modelling Upper Palaeolithic Social Change", Simulating Societies Symposium, University of Surrey, April 1992.

A. Drogoul, J. Ferber, E. Jacopin (1991) "Viewing Cognitive Modeling as Eco-Problem-Solving: the Pengi Experience", LAFORIA Technical Report, n°2/91

A.Drogoul & C. Dubreuil (1991) "EcoProblem Solving Model: Results of the N-Puzzle" *in* (Werner & Demazeau 1992)

A. Drogoul & C. Dubreuil (1992) "Reactive Agents and Self-Organization", in prep.

A. Drogoul, B. Corbara, D. Fresneau (1992a) "Applying EthoModeling to Social Organization in Ants" *in* "Biology and Evolution of Social Insects", J. Billen eds, Leuven University Press.

A. Drogoul, J. Ferber, B. Corbara, D. Fresneau (1992b), "A Behavioral Simulation Model for the Study of Emergent Social Structures" *in* "Proceedings of ECAL'91", MIT Press, to be published.

S. Hickman, M. Shiels (1991), "Situated Action as a Basis for Cooperation", *in* "Decentralized AI 2", North-Holland, 1991.

D. Hofstadter (1979) "Gödel, Escher, Bach: an Eternal Golden Braid", Basic Books, Inc., Publishers, New York

P. Hogeweg, B. Hesper (1983) "The Ontogeny of the Interaction Structure in Bumble Bee Colonies: a MIRROR model" *in* "Behavioral Ecology & Sociobiology" 12, Springer-Verlag, p. 271-283

P. Hogeweg, B. Hesper (1985) "SocioInformatic Processes: MIRROR Modeling Methodology" *in* J. theor. Biol. 113, Academic Press Inc., London, p. 311-330

P. Hogeweg, B. Hesper (1991) "Evolution as pattern processing: TODO as substrate for evolution" *in* "From Animals to Animats", MIT Press, p. 492

J.P. Lachaud, D. Fresneau (1987) "Social Regulation in Ponerine Ants"*in* "From Individual to Collective Behaviour in Social Insects", Experientia Supplementum vol. 54, Birkhäuser Verlag, Basel, p. 197-217.

Long-Ji Lin (1991) "Self-improving Reactive Agents: Case Studies of Reinforcement Learning Frameworks" *in* "From Animals to Animats", MIT Press, p. 297.

D. MacFarland (1981) "The Oxford Companion to Animal Behaviour", Oxford 1981.

P. Maes (1990) "Situated Agents can have Goals" *in* "Journal of Robotics and Autonomous Systems", Vol. 6, p. 49-70.
P. Maes (1991) "A Bottom-Up Mechanism For Behavior Selection In An Artificial Creature" *in* "From Animals to Animats", MIT Press, page 239
J. A. Meyer & S.W Wilson (1991), "From Animals to Animats", MIT Press.
I. Prigogine & I. Stengers (1984), "Order out of Chaos", New York: Bantam Books.
U. Schnepf (1991) "Robot Ethology: a Proposal for the Research into Intelligent Systems" *in* "From Animals to Animats", MIT Press, page 465
H. Simon (1969) "The Sciences of the Artificial", MIT Press.
L. Steels (1989) "Cooperation between distributed agents Through self-organisation" *in* "Journal on robotics and autonomous systems", North Holland, Amsterdam.
L. Steels (1991) "Towards a Theory of Emergent Functionality" *in* "From Animals to Animats", MIT Press, p. 451.
G. Theraulaz, S. Goss, J. Gervet, J.L. Deneubourg (1991) "Task differentiation in Polistes wasp colonies: a model for self-organizing groups of robots" *in* "From Animals to Animats", MIT Press, p. 346.
T. Tyrrell and J.E.W. Mayhew (1991) "Computer Simulation of an Animal Environment" *in* "From Animals to Animats", MIT Press, p. 263.
F. Varela (1983), "L'auto-organisation: de l'apparence au mécanisme", *in* "Colloque de Cerisy sur l'auto-organisation", Le Seuil, 1983 (in french).
P. Wavish (1992), "Exploiting Emergent Behaviour in Multi-Agent Systems" *in* (Werner & Demazeau 1992).
J.B. Watson (1925), "Behaviorism", New York 1925.
E. Werner & Y.Demazeau (1992) "Decentralized AI 3", North-Holland, June 1992.