1

Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective

Fabien Michel
CReSTIC - Université de Reims

Jacques Ferber
LIRMM - Université Montpellier II

Alexis Drogoul
IRD - Paris

1.1 Introduction

This chapter discusses the intersection between two research fields: (1) *Muti-Agent Systems* (MAS) and (2) *computer simulation.*

On the one hand, MAS refer to a computer research domain that addresses systems which are composed of micro level entities -agents-, which have an autonomous and proactive behavior and interact through an environment, thus producing the overall system behavior which is observed at the macro level. As such, MAS could be used in numerous research and application domains. Indeed, MAS are today considered as an interesting and convenient

0-8493-8597-0/01/$0.00+$1.50
© 2001 by CRC Press, LLC

way of understanding, modeling, designing and implementing different kind of (distributed) systems. Firstly, MAS could be used as a programing paradigm to develop operational software systems. MAS are particularly suited to deploy distributed software systems that run in computational contexts wherein a global control is hard or not possible to achieve, as broadly discussed in [ZP02]. At the same time, MAS also represent a very interesting modeling alternative, compared to equation based modeling, for representing and simulating real-world or virtual systems which could be decomposed in interacting individuals [PSR98, KOPD02].

On the other hand, computer simulation is a unique way of designing, testing and studying both (1) theories and (2) real (computer) systems, for various purposes. For instance, according to Shannon, simulation is defined as [Sha75]:

> "*The process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system and/or of evaluating various strategies (within the limits imposed by a criterion or a set of criteria) for the operation of the system.*"

With respect to this definition, simulation could be thus considered as a *computational tool* used to achieve two major motivations which are not mutually exclusive:

- The understanding of a real system;
- The development of an operational real system.

So, the opportunities of using both MAS and simulation are numerous, precisely because they can be applied and/or coupled in a wide range of application domains, and for very different purposes. In fact, the number of research works and software applications that belong to the intersection between MAS and simulation is simply huge. To have an idea of how close MAS and simulation are today, one can consider that there are about 800 instances of the word *simulation*, distributed among more than 35% of the 273 papers published the 2007 agent community's most known conference: AAMAS'07 [DYHS07]. Moreover, considering this already very high percentage, one has to take also into account that, in this conference, there was no session directly related to simulation at all.

So, numerous works belong to the intersection between MAS and simulation. In this book, this intersection is considered according to two main perspectives:

1. Modeling and Simulation (M&S) for MAS;
2. MAS for M&S.

Roughly, the first case refers to projects wherein computer simulation is used as a **means** for designing, experimenting, studying, and/or running a MAS architecture, whatever the objectives. Especially, simulation could be used to ease the development of MAS-based software, by following a *software-in-the-loop* approach (e.g. [RR03]): Simulation allows to design, study and experiment with a MAS in a controlled and cost-efficient way, using simulated running contexts in place of the real running context (e.g. the Internet). Examples of related application domains are Supply Chain Management (SCM), Collective Robotics, self-organized systems, and Distributed Artificial Intelligence (DAI) to cite just a few of them.

The second case is related to simulation experiments that use MAS as modeling paradigm to build *artificial laboratories*. Well known examples are the simulation of virtual insect colonies (e.g. [DF92]), artificial societies (e.g. [EA96]), social systems (e.g. urban phenomena [BDP06]), etc. Today, using artificial laboratories represents a unique way of testing theories and models for many application domains.

The first part of this chapter provides the reader with an overview of some historical motivations belonging to each perspective. As it would be endless to enumerate them all, the chapter only focuses respectively on (1) DAI aspects for illustrating the M&S for MAS perspective, and on (2) some relevant works done in the scope of Artificial Life (AL) for illustrating the MAS for M&S perspective.

Beyond the previous distinction, there is of course only a thin line between concerns belonging to each category as they both rely on simulating MAS. So, the second part presents and studies some basic concepts of MAS in the scope of simulation. However, although this study highlights some general concerns related to MAS simulations, in the third part we will argue on the idea that it does not represent the full picture of the intersection between MAS and simulation. In fact, understanding this intersection requires to shift our point of view from the MAS field to the simulation field. Indeed, simulation is not only a computational tool, but a real scientific discipline. As such, M&S already defines some general issues, whatever the application domain. That is why one has also to consider these issues to apprehend some of the challenges that the MAS community is facing regarding simulation.

So, the next parts of the chapter studies MAS simulation works according to a pure M&S perspective, derived from the Zeigler's *framework for M&S* proposed in the early seventies [Zei72]. Doing so, our goal is twofold: (1) put in the light the relevance of studying MAS simulation according to this perspective, and (2) highlight some major challenges and issues for future research.

1.2 M&S for MAS: The DAI Case

In the literature, MAS are often related to the history of DAI [Fer99]. Not surprisingly, most of the first MAS researches wherein simulated systems are involved belong to the DAI field. To understand why there is a need for simulation to engineer MAS, let us here consider three historical researches which are often cited as forerunner examples of MAS: (1) the Contract Net Protocol [Smi80], (2) the Distributed Monitoring Vehicle Testbed (DVMT)[LC83], and the MACE platform [GBH87].

The CNET Simulator

Proposed by Smith, the Contract Net Protocol [Smi80] specifies problem-solving communication and control for task allocation (or *task sharing*) over a set of distributed nodes. To experiment with this negotiation protocol, Smith needed some instances of relevant distributed problems. To this end, Smith developed a simulated system called CNET, which purpose was to simulate such instances. For instance, CNET was used to simulate a distributed sensing system (DSS): A network of sensor and processor nodes distributed in a virtual geographic area. In this experiment, the agents was in charge of constructing and maintaining a dynamic map of vehicle traffic in the area.

With these experiments, Smith was able to test and refine his protocol. Obviously, it would have been very hard to achieve such experiments without the use of a simulated environment. The reason is twofold: (1) The deployment of the system in a real running context would have been costly and (2) real-world experiments cannot be entirely controlled (e.g. hardware reliability) so that they do not ease the development process, as they include irrelevant noise in it. So, CNET simulations were used to provide the suitable **testbed** which was required by Smith to experiment with the Contract Net Protocol.

The point here is that the CNET simulator has played a fundamental role in this research: CNET was **the** means by which the Contract Net Protocol has been evaluated and validated.

Therefore the CNET settings were a critical parameter doing this research and this was already noticed by Smith in the original paper. This historical research is a forerunner example of the importance that simulation has always taken in DAI research, and also MAS engineering. This is even more clear with the second example.

The DVMT Project

Initiated in 1981 by Lesser and Corkill and continued through 1991, the Distributed Monitoring Vehicle Testbed (DVMT) project [LC83] has been one of the most influential research for the MAS field. The purpose of the DVMT project was clear: Provide a generic software architecture, a so-called **testbed**, for experimenting with cooperative distributed problem solving networks composed of *semi-autonomous* processing nodes, working together to solve a problem (an abstract version of a vehicle monitoring task in this project). In this perspective, the DVMT was clearly presented as a simulation tool, which purpose was to enable researchers to explore design issues in distributed problem solving systems.

The interesting thing is that this research emphasized more on the need of a real DAI tool than on a particular DAI problem to solve. Indeed, considering the DAI research experiments which were previously conducted till that time, such as the well known Hearsay II project [EHRLR80], it was clear that further investigations was required to understand and explore all the issues raised by the use of these *Functionally Accurate, Cooperative Distributed Systems* (FA/C) [LC81], especially regarding the size of the network and the communication topology of the nodes. Questions like, how to select an appropriate network configuration with respect to the selected task characteristics, was at the heart of such researches, and this did require extensive experimentation. Lesser and Corkill pointed out the difficulties of doing such experimentations because of the inflexibilities in the design of the existing systems: The design of these systems were focused on the problem to solve rather than on the corresponding testbed. Thus, experimenting with the existing systems was time consuming or even not feasible.

Therefore, designing the DVMT, the approach of Lesser and Corkill was to [LC83]:

- Abstract a realistic distributed solving task to make it more generic and parametrizable.
- Develop the related distributed problem solving system using a software architecture as flexible as possible.
- Build a **simulation system** able to run this system under different environmental scenarios, node and communication topologies, and task data.

Motivating this approach, Lesser and Corkill said [LC83]:

> "*We feel this approach is the only viable way to gain extensive empirical experience with the important issues in the design of distributed problem solving systems. In short, distributed problem solving networks are highly complex. They are difficult to analyze formally and can be expensive to construct, to run, and to modify for empirical evaluation... Thus, it is difficult and expensive to gain these experiences by developing a "real" distributed problem solving application in all its detail.*"

Obviously, Lesser and Corkill were right as they were generalizing approaches like Smith's. But, more than a pragmatic method of doing this kind of research at that time, the approach followed by Lesser and Corkill clearly highlights (1) that modeling and simulation is a key for most of the DAI researches and thus (2) why the simulation aspects on which rely these researches cannot be bypassed or underestimated.

Here, considering the scope of this chapter, one question is: Is it possible to generalize the importance of modeling and simulation aspects to MAS engineering in general? The answer is a strong yes. Indeed, in the preceding citation, it is possible to replace the words *distributed problem solving systems* by *MAS* to have a statement that still holds today in most cases when engineering multiagent-based software.

Still, DVMT was only a first step toward a more general way of engineering agent-based systems as it was focused on the simulation of a particular domain, considering a specific paradigm (FA/C), and motivated by pure DAI purposes. The next step relied on proposing a more generic testbed which would not have been built with a dedicated domain in mind. This step was achieved few years later by another historical pioneer work that has also deeply influenced MAS research: the MACE platform [GBH87].

MACE: Toward Modern Generic MAS Platform

The *Multi-Agent Computing Environment*, MACE, is the result of a work initiated by Gasser and his colleagues in the mid eighties [GBH87]. At that time, thanks to the previous works done on DAI testbeds, there was a general agreement in the DAI community that having a framework embedding simulation tools was essential for exploring DAI issues (see [Dec96] for a detailed discussion on historical DAI testbeds and their essential features). However, MACE represented a historical shift for several reasons:

- MACE was not related to a particular domain nor based on a particular DAI architecture;
- MACE was completely focused on providing DAI system development facilities able to model various kinds of DAI systems, and considering different levels of granularity for the agents;
- MACE proposed both (1) modeling and simulation tools to experiment DAI systems and (2) a physically distributed software environment able to deploy the corresponding systems over a real network;
- the MACE software was itself designed as a *community of system agents* that defined the concrete means for building agent-based software systems with MACE: MACE was agentified.

So, even though MACE was tagged as a DAI tool, it could be considered as the ancestor of many generic MAS platforms that were designed in the years that followed (e.g. MadKit [GFM01], James [HRU03], Spades [RR03], and the new version of MACE, MACE3J [GK02]). Indeed, most of the existing MAS testbeds and platforms (DAI related or not) provide simulation tools, particularly to enable a software-in-the-loop approach for designing MAS architectures.

1.3 MAS for M&S: Building Artificial Laboratories

1.3.1 The Need for Individual-Based Modeling

The interest of using MAS in the scope of M&S mainly appears when it comes to the simulation of complex systems (i.e. systems which are composed of many interacting entities). The modeling of complex systems has always been a motivation for scientific researchers. One historical example is the continuous deterministic model which has been proposed by Volterra to represent the population dynamic of two animal species (preys and predators) [Vol26].

By mathematically relating the population dynamics of two species using a differential equation system (DES), the purpose of this model was to explore the mechanism of interaction between predator and prey (sharks and sardines in the original Volterra's study). This model was found appropriate because the solution of this DES shows an intuitively sound result which is that both populations should oscillate: When the prey density is high, the predator multiply until there are not enough preys, which leads in turn to the regeneration of the preys, thanks to the lethal competition which occurs between the predators.

However, despite the apparent appropriateness of this model, some researches gave results which did not exhibit oscillatory behavior with experimental predator/prey populations [Gau34]. Therefore, many variations of the original model were done in order to reconcile the discrepancy between observed experimental results and the model: Adding more complex dynamics (e.g. variation of the predator's voracity over time), new constraints (e.g. a maximum number of animals in the environment), and/or new parameters (e.g. gestation time of prey and predator) to the system [Ros71].

There were also early attempts to formulate a stochastic version of the system, but it was only in the sixties that, thanks to the advent of computer simulations, it has been possible to experiment with such versions of the model (e.g. [Bar60]). At that time, these researches highlighted that one major drawback of deterministic models precisely relies on the fact that they are unable to take into account the apparent not deterministic nature of real life complex systems such as a predator/prey ecosystem. Indeed, real life interaction situations seem to always involve randomness because of the actual complexity of real world processes.

However, the counterpart of the stochastic versions of the original equations was that they exhibit more complex behaviors and great variability, even when used with the same parameters [BS76]. This raises the question of the relevance of the parameters with respect to the targeted system: Are they really capturing something form reality? Such models finally seem to be more related to a mathematical exercice than a modeling that helps to understand the targeted system. So, although this model has some interesting behavior with respect to real population dynamics, it is criticized for the lack of insights it gives about the dynamics of the true components of the systems: The preys and predators.

So, several problems remain with these approaches considering the modeling of complex systems which involve individual entities [Fer99]:

- Only a global perspective is possible
- Equation parameters hardly take into account the complexity of micro-level interactions
- The modeling of individual actions is not possible
- Integrating qualitative aspects is hard

As we have seen in the previous section, MAS platforms, relying on the modeling and simulation of autonomous proactive entities, did appear practically at the time the agent paradigm was defined, in the early eighties. However, the notion of agent itself, as a fundamental unit of modeling could be found in models and experiments which took place much more earlier, at least in the late fifties.

Indeed, the first attempts that tried to solve the previously cited problems, by integrating several levels of analysis in the modeling, were proposed by the microsimulation approach.

1.3.2 The Microsimulation Approach: The Individual-Based Modeling Forerunner

In the scope of social sciences, the notion of agent has always been essential. So, one of the first approaches which have been proposed as an alternative to mathematical models comes from social science and is named microsimulation [Orc57].

In [Orc57], Orcutt pointed out that the macroeconomic models which were proposed till that time failed to provide relevant information about the influence of governmental policies on the evolution of micro level entities (households and firms). Therefore, Orcutt's point was to emphasize the need of integrating micro level entities in the development of simulation models, making their results more relevant and useful for the study of social systems dynamics. To this end, the basic principle of microsimulation is to concretely integrate the micro level by means of rules, either deterministic or stochastic, that apply on the attributes of individual units, leading to the modeling of changes in their state and behavior.

Mainly used for population, traffic, and firm based models, microsimulation (or microanalytic simulation) is a modeling approach which is still intensively used today, as the activity of the International Microsimulation Association (IMA) shows*. The IMA website provides many references to actual works in the microsimulation field. Other introductions to this field could be found in books such as [CHT97, Har96].

So, although microsimulation has its roots within the scope of social sciences, it can be considered as the original starting point of Agent-Based Modeling (ABM) approaches (sometimes called Individual-Based Modeling, e.g. in the modeling of ecological systems [GR05]).

1.3.3 The Agent-Based Modeling Approach

Considering the integration of the micro level within the modeling, an ABM approach goes a bit further than microsimulation. ABM suggests that the model not only integrates the individuals and their behaviors, but also focus on concretely modeling the actions and interactions that take place between the entities, through the environment. Similarities and differences between ABM and previous approaches are discussed in more details in this book in [Tro08].

So, with respect to Equation-Based Modeling (EBM), using the MAS paradigm to model a system provides a completely different perspective which represents an attractive alternative regarding the problems raised previously: Contrary to EBM, wherein the system global dynamics are defined a priori using mathematical relations between global system properties (e.g. the total number of preys), ABM relies on the explicit modeling of micro level entities and dynamics (e.g. individual characteristics and behaviors, actions and interactions between the entities and the environment, etc.). The observed global behavior of the system being thus considered as the result of these micro level dynamics. For instance, considering a predator/prey system, each individual has to be modeled (cf. figure 1.1).

More generally, quoting [PSR98], the two main differences between EBM and ABM rely on (1) the way they model relations between entities and (2) the level at which they focus their attention. EBM model relations between system observables (i.e. measurable characteristics of interest) while ABM represent individuals that evolve by interacting with one another and the environment.

The interest of MAS relies on four main concepts:

*http://www.microsimulation.org. Accessed June 2008.

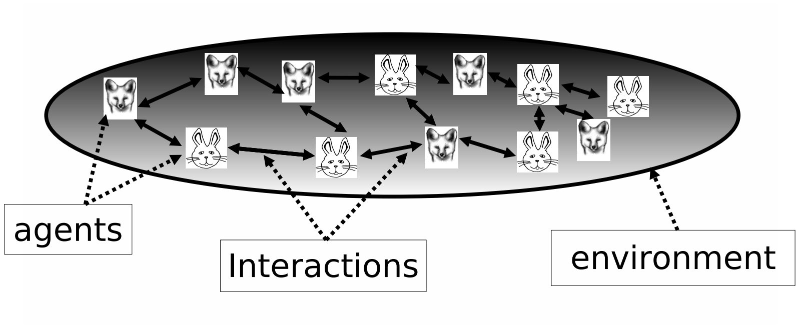

FIGURE 1.1 An agent-based model: The micro level entities, their actions and interactions, and the environment

1. Autonomous activity of an agent, i.e. its ability to carry out an action on its own initiative, (pro-activity), by controlling its behavior in order to increase its satisfaction and by deciding to help or to prevent others from satisfying their goals. As it could be noticed, the previous definition of agent stresses the autonomy of decision, which results from the independence with which an agent tries to satisfy its objectives (in the broad sense of the term), by using its competences and its own resources, or by asking for help from others.
2. The sociability of the agents, i.e. their ability to act with other agents from a social point of view. An agent in a MAS is not an isolated entity, but an element of a society. A society emerges from the interactions which take place between the agents, and conversely, the organization of a society constrains the behavior of the agents by ascribing them roles which will restrain their action potentialities.
3. Interaction is what connects the two preceding concepts. This interleaving of actions, where each action is decided inside an agent's mind, produces organized patterns of activities, emerging social forms, which, in return, force and constrain the behavior of agents. So it is through these interactions that forms of interaction emerge, such as cooperation, conflict or competition. This, in return, produces more or less stable organizational patterns which structure the individual action of each agent.
4. The situatedness of the agents, i.e. the fact that the agents are placed into an environment which defines the conditions in which the agents exist, act, and interact. The environment is the glue that connects the agents together, enabling interaction between the agents so that they are able to achieve their goals.

All the power of MAS comes from this cyclic process: Agents act in an autonomous way in a space constrained by the structure of the society in which they act, this structure resulting itself from the behaviors of these agents. There is a dependency loop between agents and societies, between the micro and the macro level, between individual and collective, which is finally at the core of complex system issues.

It is thus not by chance if MAS seem to be a major tool to model complex systems, as societies of agents. They propose much more than a simple technique of modeling. They are not simple abstract tools making it possible to numerically characterize the evolution of a system from its parameters. Being societies by themselves, and being built on the same basis as any complex systems, MAS prove to be "artificial micro-worlds", of which it is possible to control all characteristics and reproduce series of experiments as in a laboratory. Compared to animal and human societies, and to complex systems in general, MAS may be seen as "microcosms", small-scale models of real systems, like reduced size models of boats and buildings, while having the same dynamics and following the same principles as the real social systems.

Moreover, one of the main qualities of multi-agent modeling stands in their capacity of integration and in their flexibility. It is possible to put together quantitative variables, differential equations and behaviors based on symbolic rules systems, all in the same model. It is also very easy to incorporate modifications in the behavior of the individuals, by adding behavioral rules which act at the individual level. It is also possible to add new agents with their own behavioral model, which interact with the already defined agents. For example, in a forest management model, it is possible to introduce new animals, new plant species, new farmers, and to analyze their interactions with already modeled agents.

Because of emergent processes due to interactions, MAS makes it possible to represent complex situations where global structures result from interactions between individuals, i.e. to spring up structures of the macro level from behaviors defined at the micro level, thus breaking the barrier between levels.

Establishing an exhaustive list of all the simulations which rely on an ABM approach would be endless. Therefore, we will only focus on some representative and forerunner examples in the following sections.

1.3.4 Agent-Based Social Simulation: Simulating Human-Inspired Behaviors

Naturally, inspired by the forerunner works which have been done in the scope of microsimulation, social science was one of the first research field wherein an ABM approach has been applied. In this perspective, the seminal work of Schelling on residential segregation [Sch71] has deeply inspired the field. Schelling's model was one of the first to show clearly how global properties (segregation among the agents in this case) may emerge from local interactions.

Following this trend of research, Agent-Based Social Simulation (ABSS) becomes only widely used in the mid-nineties (a number of examples can be found in [CHT97]). These works model artificial (human-inspired) societies, considering various levels of abstraction as discussed in [Gil05].

To depict the approach underlying these models, Epstein and Axtell coined the notion of *generative social science* in the book *Growing artificial societies* [EA96]. As discussed in more details by Epstein in [Eps07], this notion enables to highlight the differences between ABM and both inductive and deductive social science. More on the different motivations of using agent-based models for social science could be found in papers such as [CGS98, Axt00, Gol02] and in the second edition of the book of Gilbert and Troitzsch which gives a comprehensive view of the field [GT05].

ABSS concrete examples are the modeling of urban phenomena (e.g. [VTD00, BDP06]), works done in game theory on the iterated prisoner dilemma (e.g. [BDM98]), and the modeling of opinion dynamics [DAWF02]. One can find many other examples in the electronic journal JASSS*, the *Journal of Artificial Societies and Social Simulation*.

Some generic ABSS related platforms have also been proposed (e.g. ASCAPE [Par01],

*http://jasss.soc.surrey.ac.uk

Moduleco [Pha04]). Moreover, a modeling language relying on rule-based agents has been proposed in [MGWE98], namely SDML. Today, the RePast toolkit [NCV06] is a representative example of platform which is used for ABSS.

1.3.5 Flocks and Ants: Simulating Artificial Animats

The Reynolds's Boids

Considering the use of the agent paradigm for M&S purposes, a forerunner work has been done by Reynolds on flocks [Rey87]. In the scope of computer graphic animation, the goal of Reynolds was to achieve a believable animation of a flock of artificial birds, namely *boids*. Reynolds remarked that it was not possible to used a scripted flock motion to achieve a realistic animation of group motion. So, Reynolds was inspired by two stream of research: (1) particle systems [Ree83] and (2) the Actor paradigm [Agh86].

Particle systems were already used for the animation of complex phenomena, such as clouds or fire, which were modeled as collections of individual particles having their own behavior and state. However, particles did not interact as their behavior only relied on their own internal state (position, velocity, lifetime, etc.) and potentially on some global parameters that could represent phenomena such as gravity.

The idea of Reynolds was that boids have to be influenced by the others to flock in a coherent manner: "*Boid behavior is dependant not only on internal state but also on external state*". So, Reynolds used the actor abstraction to define the boids behavior so that they do interact to flock. Boids change their directions according to others' to stay at the center of the local flock they perceive.

The results obtained were very compelling and the impact of the Reynolds's boids on the community of reactive agents can still be perceived today as it was one of the first works to be *bio-inspired*.

Indeed, beyond graphic animation, boids-inspired behaviors can be found in domains such as mobile robotics (e.g. flocks of unmanned aerial vehicles (UAV) [NPL07]), and human crowd simulation, e.g. like in [MT00] or [PAB07]. In this last reference, the research is also inspired by the pioneer work of Helbing and Péter Molnár on human crowd which uses the concept of *social forces* to model how pedestrian are *motivated to act* with respect to external environmental conditions (e.g. others pedestrians or borders) [HM95]. A famous example of application derived from these different technologies is the use of the Weta Digital's $MASSIVE^{TM}$ software for special visual effects in movies such as the *Lord of the rings*, in which there are huge battle scenes.

Although the boids paradigm is well suited for modeling collective moves, it is limited by the fact that it only relies on the direct perception of the others: The environment does not play any role in the interaction between the agents. On the contrary, the environment play an essential role for the modeling of ant colonies.

Ant Colonies

Regarding bio-inspired MAS related works, ant colonies probably represent the natural systems which has inspired the most. Indeed, firstly ant colonies exhibit many features which are expected from MAS such as self-organization, self-adaptation, robustness, emergent properties, and so on. Secondly, the ability of ants in solving complex problems in an efficient and constant way is fascinating considering their limited individual characteristics. The most well known example of that being how ants use pheromones to find the shortest path from the nest to food sources [BDG92].

So, as ant colonies was quickly recognized as a fascinating model of MAS, many MAS simulations done in the early nineties rely on simulating and studying characteristics of ant-based systems (e.g. [CJ92]). Among these forerunner works on ants, one of the most ambitious has been done in the scope of the MANTA (Modeling an ANTnest Activity) project [DF92]. Firstly, MANTA relied on an innovative modeling and simulation framework called EMF (EthoModeling Framework) which was sufficiently simple to be used by non computer scientists to implement multiagent simulations. Secondly, the purpose of MANTA was not only to simulate some characteristics of ants but to translate all the parameters of a real biological study into a virtual ant farm. So, MANTA simulations were composed of all the creatures that can be found inside an ant nest (ants, larvae, cocoons, eggs) and also took into account time and some environmental factors such as light and humidity.

It is worth noting that, at the same time, Dorigo also took inspiration from ants to define a new kind of optimization algorithms, namely the *Ant system* heuristic [Dor92]. Especially, he derived from this approach a distributed algorithm called *Ant Colony System* (ACS) which was applied to the traveling salesman problem [DG97]. Today, *Ant Colony Optimization* (ACO) [DS04a] represents a whole trend of research in the domain of swarm intelligence.

The common point between all these researches is that they all identified that one major characteristics of ants is their ability to use pheronomes (evaporative scent markers) to indirectly communicate and coordinate through the environment. Indeed, pheromones enable the ants to achieve complex tasks thanks to a simple environmental mechanism: Pheromones evaporate, so that the system can forget obsolete information which is the fundamental feature of such systems [Par97]. For instance, paths leading to depleted food sources progressively disappear with time. This powerful mechanism has been successfully translated into computer programs and simulations, thus defining the concept of digital pheromones which has been used in numerous works. Additionally, using a pheromone-based approach of course does not entail to only consider ant-like agents. One example can be found in this book: *Polyagents* [PB08].

1.4 Simulating MAS: Basic Principles

MAS have been developed around a set of principles and concepts. These concepts are: Agents, Environment, Interaction and Organizations, as presented in the Vowels approach [Dem97]. We will give a brief overview of the first three aspects in this chapter. Then, we will discuss issues related to the modeling of time in MAS simulations.

1.4.1 Agent

Many definitions of agency have been proposed in the field of MAS, each one being more adapted to a specific flow of research (see [Woo02]). The following one is adapted from [Fer99]: An *agent* is a software or hardware entity (a process) situated in a virtual or a real environment:

1. Which is capable of acting in an environment,
2. Which is driven by a set of tendencies (individual objectives, goals, drives, satisfaction/survival function),
3. Which possesses resources of its own,
4. Which has only a partial representation of this environment,
5. Which can directly or indirectly communicate with other agents
6. Which may be able to reproduce itself

7. Whose autonomous behavior is the consequence of its perceptions, representations and interactions with the world and other agents (cf. figure 1.2).

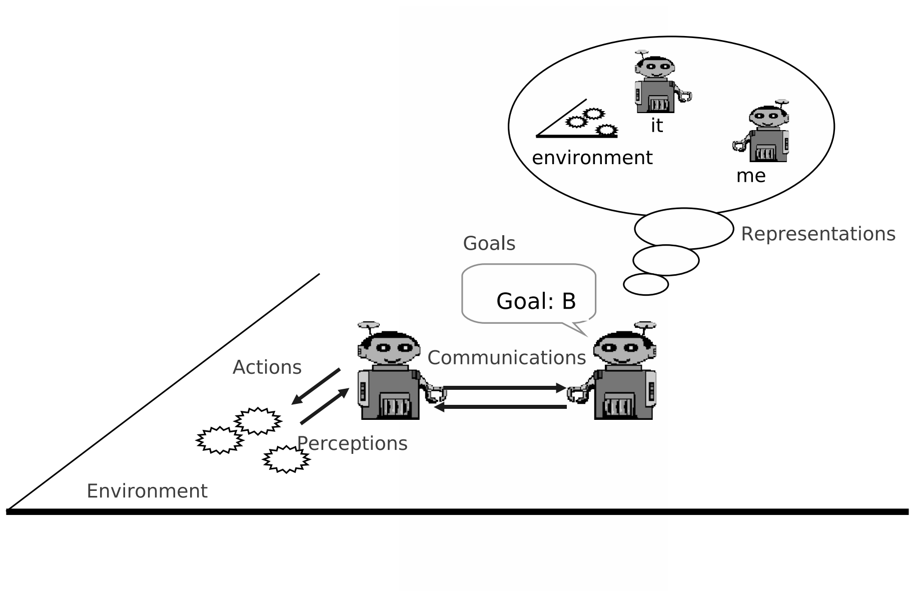

FIGURE 1.2 A multiagent world

Agent Architectures

The term generally used to describe the internal organization of an agent is that of *architecture*, by analogy with the structure of computers.

It is generally considered that there are two main approaches for analyzing agent architectures: (1) The reactive approach in which we only consider perception-action (or stimuli-response) architectures and (2) the cognitivist approach which relies on mental issues, such as the explicit representation of the environment (and other agents) by agents. A third approach, called hybrid, consists in trying to get together the first two.

Reactive Architectures. A reactive agent does not have an explicit representation of its environment nor of other agents. Its behavior is entirely described in terms of stimuli-response loops which represent simple connections between what they perceive and the set of available operations that may be performed. The most well-known architectures in this domain are the subsumption architecture in which tasks in competition are arbitrated along predefined priorities [BC86], the competitive task architecture, where concurrent tasks have their weight modified through a reinforcement learning process [DF92], and connectionist architectures which are based on neural nets. Some approaches combine these different structures within an integrated architecture, like the one of Tyrell for instance, which combines control by priorities and neurons, within a hierarchical structure [Tyr93]. There are also architectures that combine behaviors, each behavior being represented as a vector of actions. The *Satisfaction-Altruism* function allows to combine behaviors centered on the desires of agents with cooperative behaviors centered on the needs of others [SF00].

Cognitive Architectures. Cognitive architectures are founded on the computational metaphor which considers that agents reason from knowledge described with a symbolic formalism. This knowledge explicitly represents their environment (states, properties, dynamics of objects in the environment) and the other agents. The most well-known architecture of this type is the BDI (Belief-Desire-Intention) which postulates that an agent is characterized by its beliefs, its goals (desires) and intentions [RG92]. It is assumed that cognitive agents are intentional, i.e. that they intend to perform their action, and that these actions will allow them to satisfy their goals. In other words, a BDI agent acts rationally from its beliefs about world states, its knowledge (and those of others), its intentions (and the one of others) to achieve its goals. A BDI agent is assumed to possess a library of plans, each plan being designed like a recipe making it possible to achieve a particular goal. An intention is set when an agent makes a commitment about achieving a specific goal by using a particular plan. The management and update of beliefs, goals and intentions are carried out by the BDI engine which selects the plans and the actions to be undertaken.

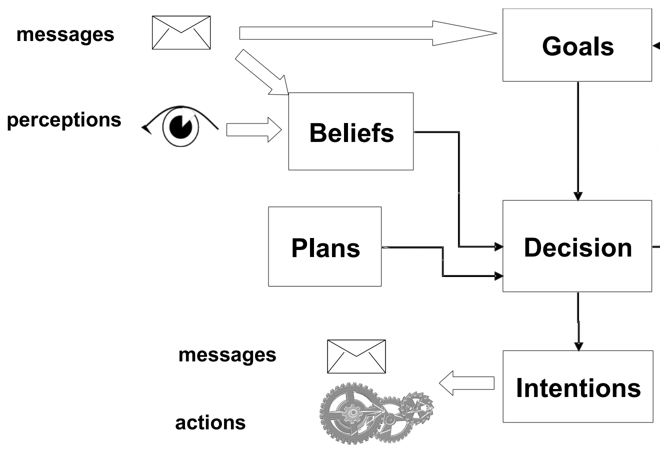

FIGURE 1.3 A BDI architecture with its different modules

Figure 1.3 illustrates the functional representation of a BDI architecture. External perceptions and messages are used to form beliefs. It is supposed that an agent may initially have several goals. These initial goals, added with the goals resulting from requests of other agents, are analyzed and selected in the decision component to trigger the set of plans compatible with the agent's beliefs. If there are no plans that can be found, a problem solver (not shown here) has the responsibility to decompose the initial problem into sub-problems, by producing sub-goals that the agent will have to satisfy. When an agent chooses to execute a plan, the actions of this plan are transformed into intentions which will produce environmental actions and communications with other agents. The main quality of BDI architectures is to create a behavior which mimics that of a rational human being. As an example, The Jason platform, which is described in this book [Bor08], enables to design Agent-Based Simulation Using BDI Programming.

Hybrid Architectures. The two main types of agents, cognitive and reactive, propose solutions apparently diametrically opposite, but in fact, they may be viewed as complementary. In order to build best suited architecture to solve a problem (in terms of response time, precision or efficiency), it is possible to create hybrid architectures which combine the two types of approaches, and then build more flexible agent architectures.

In such architectures, agents are composed of modules which deals independently with the reflex (reactive) and reflexive (cognitive) aspect of the agent behavior. The main problem is then to find the ideal control mechanism ensuring a good balance and a good coordination between these modules. Let us cite the Touring Machine [ABB$^+$93] and InteRRap [MP93], the most well know examples of hybrid architectures.

Developing MAS with a specific architecture in mind (cognitive, reactive and even hybrid approaches), may be a disadvantage when developing open MAS: some developers think in terms of reactive agents whereas others prefer to use a cognitive approach. In an open system, all these agents must live together, conform to a framework of execution and of behavior, and thus must be able to interact within the same interaction space.

Modeling the Behavior of Agents

In order to simplify the model of agents, the process which takes place between the perception of inputs and the production of outputs can be considered as the *deliberation function* of the agent. So, an agent is a cyclic three phases process: (1) perception, (2) deliberation, and then (3) action (cf. figure 1.4).

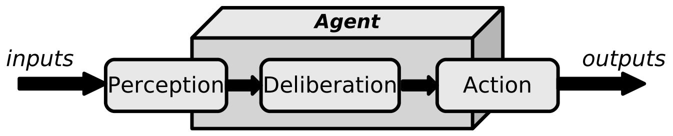

FIGURE 1.4 An agent as a three phases process

As the purpose of this section is not to explain the modeling details of existing agent architectures, we here just present a formalism which is inspired by the work of Genesereth and Nilsson [GN87] and focuses on representing the cyclic nature of the behavior of an agent. Let $\sigma \in \Sigma$ be the actual state of the world, the behavior cycle (perception / deliberation / action) of an agent a is represented using a function $Behavior_a : \Sigma \mapsto A_a$, A_a representing the set of possible actions that the agent a can take:

- $Perception_a : \Sigma \mapsto P_a$, that computes a percept $p_a \in P_a$ from the system's state Σ.
- $Deliberation_a : P_a \times S_a \mapsto S_a$, that computes the new internal state s_a of the agent.
- $Action_a : P_a \times S_a \mapsto A_a$, that produces the action of a.

So, firstly an agent obtains a percept p_a which is computed by the *Perception* function using the current state of the environment σ. Such a function may simply return some raw data (quantitative variables) used to defined the state of the environment (e.g. coordinates of objects, the current temperature in celsius degrees, etc.), but it may also represent more complex processes that transform raw data to high level percepts representing qualitative aspects of the environment (e.g. near or far from an object, the temperature is hot or cold, etc.), thus easing the deliberation of an agent as discussed in [CCC+05].

Secondly, the *Deliberation* function (*Memorization* in [GN87]) defines how the agent uses p_a to make its internals evolve according to p_a, updating its own representation of the world for instance. The deliberation process of an agent defines the core part of its behavior and characterizes its architecture (reactive or cognitive). As such, it is obviously the part which has been studied the most. Still, it is worth noting that this function is skipped for tropistic agents (without memory) as they directly match percepts to actions.

Finally, the *Action* function represents how an agent makes its decision, based on its new internal state and current percept, and thus chooses the action to take. Most of the time, this action is directly concretized by the modification of the environment which is supposed to succeed the action (e.g. $\sigma = \{door(closed)\} \mapsto \sigma = \{door(open)\}$). In other words, the direct modification of the environment is the means by which is the result of the action of an agent is computed.

1.4.2 Environment

An Essential Compound of MAS

In the beginning of MAS, in the old days of DAI, the concept of environment has not been given an important consideration. But the development of MABS has shown the importance of the environment because, in MABS models, agents are situated in a concrete environment: the simulated environment.

More recently, the environment has also been pointed out as an essential compound of MAS as it in fact broadly defines all the perceptions and actions that an agent may have or take: The environment defines the conditions in which the agents exist in the MAS [OPFB02]. Especially, the term environment could also refer to the infrastructure in which agents are deployed and thus be studied as a first order abstraction from an agent-oriented software engineering perspective, as thoroughly discussed by Weyns et al. in [WPM+05b].

To distinguish the different concerns which could be related with the concept of environment in MAS, Valckenaers et al. have proposed a structured view on environment-centric MAS applications, thus identifying three base configurations which one is simulation [VSSRA07]. So, in the scope of a simulation configuration, according to this proposition the environment simply refers to the part of the simulation that models the relevant portion of the real world (either it exists, needs to be realized or no longer exists) or a imaginary world. From now on, we will only consider this aspect of the environment, i.e. the modeling part that represents the world in which the simulated agents evolve.

Modeling the Environment

The inputs an agent receives comes from the environment is situated in (and of which the other agents are part). Similarly, the outputs an agent produces go in the environment.

From a M&S perspective, the environment is another dynamical system (ABM relies on a *multi-model* approach). However, contrary to an agent model, the dynamic of the environment does not represent any autonomous behavior. Regarding the characteristics and dynamics of the environment, Russell and Norvig propose to consider several properties [RN03]:

- *Accessible* vs. *inaccessible*. An environment is defined as accessible if its complete

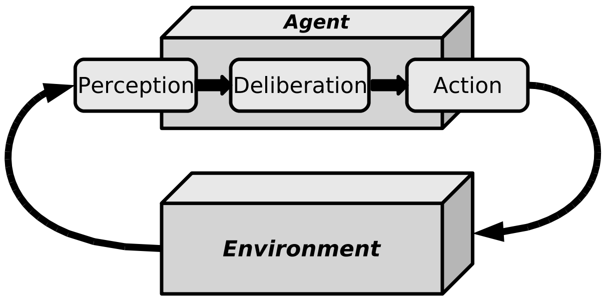

FIGURE 1.5 An agent receives inputs from the environment and produces outputs on it

state can be can perceived by an agent.

- *Deterministic* vs. *nondeterministic.* An environment is deterministic if its next state is entirely determined by its current state and the actions selected by the agents.
- *Episodic* vs. *nonepisodic.* In an episodic environment, an episode (an agent perceiving and then acting) does not depend on the actions selected in previous episodes.
- *static* vs. *dynamic.* In a static environment, changes only occur under the influences of the agents, contrary to a dynamic environment which posses an endogenous evolution.
- *Discrete* vs. *continuous.* In a discrete environment, the number of possible perceptions and actions is limited and clearly defined.

The modeling of the environment usually embeds the representation of some physical places wherein the agents evolve. As the environment defines the perceptions and actions of the agents, two main approaches can be distinguished with respect to the granularity of these perceptions and actions:

1. Discretized: The environment is discretized in bounded areas that define space units for the perception/action of the agents (environment-centered)
2. Continuous: The range of each perception/action depends on the acting agent and the nature of the perception/action (agent-centered)

The first approach consists in modeling the environment as a collection of connected areas, which thus defines the topology of the environment. Figure 1.6 shows three examples for such an approach: The rooms of a house thus defining non-uniform cells (a), a physical space divided in regular zones (uniform cells) (b) and the nodes of a network (c). With such a modeling, the idea is to use the environmental characteristics to define the range of the perceptions and actions (e.g. an entire room, a cell, or a network node).

The most usual examples of this kind of approach are those wherein the environment is discretized in a regular grid of cells (or patches) (cf. figure 1.6 (b)). Platforms allowing such a modeling of the environment are numerous (e.g. StarLogo [Res94], TurtleKit [MBF05]).

Widely used for their simplicity of implementation, grid-based environment models have

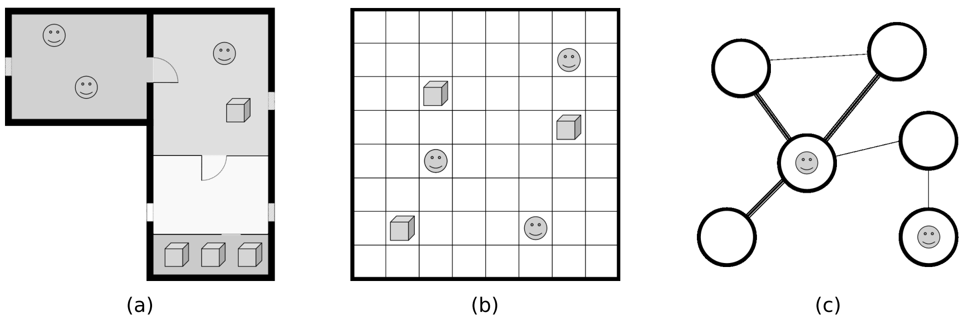

FIGURE 1.6 Examples of discretized environments

also the advantage of easing the modeling of environmental dynamics such as the diffusion and evaporation of digital pheromones. As major drawback, a grid-based model raises the problem of the granularity of the perceptions/actions which an agent can make. Indeed, whatever the perceptions/actions of an agent, which can be very heterogeneous, their range will always be the same (modulo a factor): The cell.

On the contrary, the continuous approach considers each agent as the reference point from which the range of perceptions/actions is computed (cf. figure 1.7. This kind of modeling is required when accurateness is needed. For instance, this modeling is usually used to simulate soccer robots (e.g. in RoboCup soccer simulators [KAK+97]): The granularity of the movements of both the agents and the ball has to be accurately modeled.

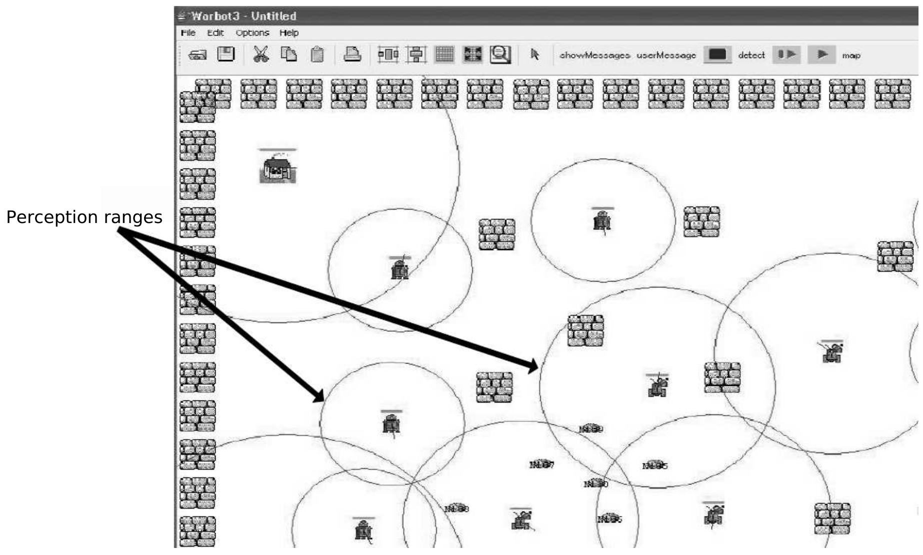

FIGURE 1.7 Continuous approach for the perception of the agents

Except from the fact that such a modeling could be required, the main advantage is that the continuous approach is far more flexible than the former considering the integration of heterogeneous perceptions/actions within the modeling of the agents. The counterpart is that it is of course more difficult to model and implement such an approach.

1.4.3 Interactions

Communications between agents are at the core of interactions and social organizations. Without communicating, an agent is totally isolated, deaf and dumb, closed on its perception-deliberation-action loop. Because they communicate, agents can cooperate and coordinate their actions, and become true social beings. Communication is expressed through language items (messages, signals) which, once interpreted, will produce an effect on agents. There are a great number of approaches of communications: social sciences, linguistics and philosophy of language developed a whole set of concepts, in particular in all what is known as speech acts. Biology and ethology have also produced theories on communication through signals.

Communication by Message Passing

The traditional model of communication between agents relies on message passing. A transmitter sends a message to a recipient knowing directly (or indirectly via a directory) its address. This simple model is used in most of the MAS. It has been extended by the speech act theory, which gives precise semantics to the interaction. Appeared with the work of Austin [Aus62] and Searle [Sea69], the concepts of speech act theory have been simplified with the development of the KQML [FFMM94] and ACL [Age] languages. Speech act theory considers that communications may be interpreted as mental acts. For example, if an agent A, with the goal of having the world in state S sends a request to an agent B to perform the action a, it supposes that A believes that the world is not in state S, that a has not been performed, and that the achievement of a would result in the state S, and that A believes that B is able to do a. It is the same if A informs B that a proposition p is true. Its meaning resides in mental issues: A believes that p is true, that B has no beliefs about p, and the expected result by A is that finally B eventually believes that p is true.

Research works have also been conducted to define interaction protocols, i.e. to define the sequence of messages which characterize an interaction situation. For example, if A requests B to perform a, it waits for an answer from B saying if B agrees to accomplish the job. If B does agree, it will inform B when the task a will be achieved. This means that it is possible to represent interactions between agents with communication diagrams. Proposed by Odell, Bauer and Müller, AUML [BMO01] is a MAS description language which extends UML sequence diagrams by adding specific extensions and by reducing some of the obviously too specifically "object-oriented" aspects.

Interactions Using Signals

Besides speech acts and communication standards, agents may communicate by using signals. This type of communication is based on biology and ethology, where animals tend to behave collectively by using signals. In the MAS domain, two main kinds of signals are used: marks and fields. Marks are traces that agents make while moving. Like little thumb, they may drop marks on their way (pheromones, tracks, pebbles, objects of any sort,..) which could then be interpreted by other agents. This mechanism produces the famous ant lines. The tracks, in the form of pheromones dropped by the ants while returning to their nest, are used as interaction medium to signify to other ants where food can be found (figure 1.8).

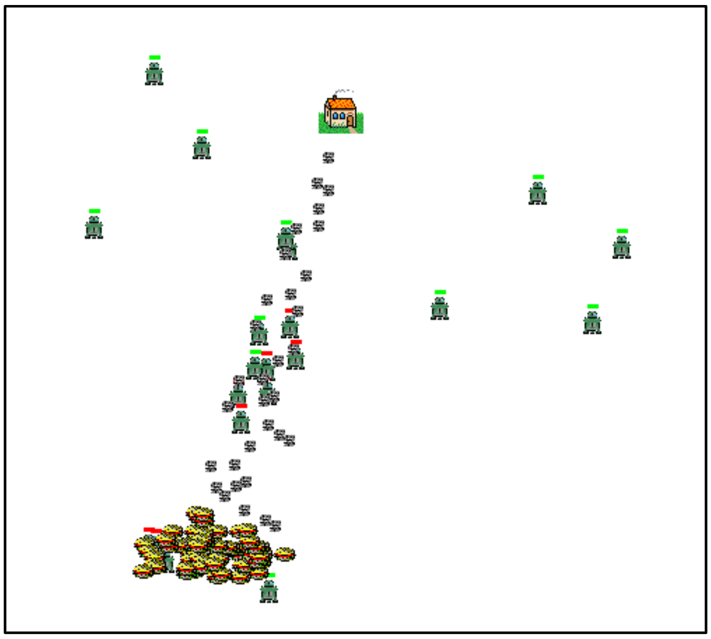

FIGURE 1.8 Ant agents, represented as robots, carry resources back to their base while dropping pheromones which are used as marks

Signals may be spread in the environment. And in this case, they may be used to indicate, remotely, the presence of obstacles, desirable objects, agents to help or to avoid. These signals are used to coordinate a set of agents acting in a common environment. Let us consider the intensity of signals relative to the distance between the source and a location in the environment. If these signals are propagated in a uniform and homogeneous way, they form potential fields. It is then possible to define attractive and repulsive forces from the gradient of these fields [Ark89]. The goals are then represented as attractive fields and obstacles as repulsive fields. The movement is obtained by a combination of attractive and repulsive fields, and an agent has only to follow the greatest slope direction (figure 1.9). When obstacles can move, i.e. when there are other agents, it is necessary to add avoidance behaviors [ZF93, SF00, CSF02].

One has to notice that these signals may either be real (and part of the environment) or virtual, i.e. rebuilt by agents as simple means to coordinate their movements in the environment, like it is done in [MGMB07] in the scope of image segmentation: Simulated agents create potential fields, without affecting the pixels, to attract others in areas of interest (object edges).

1.4.4 Modeling Time

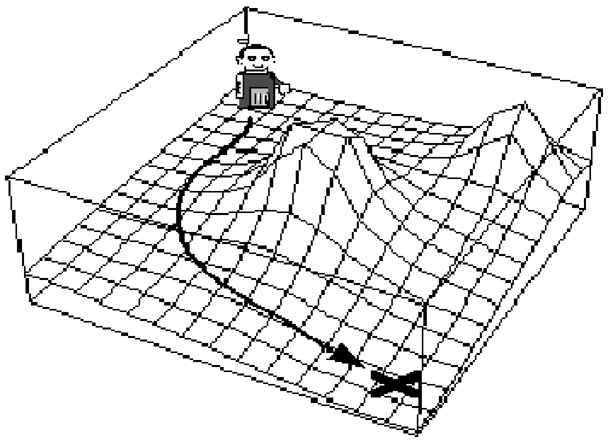

FIGURE 1.9 An agent moves in a field by following its gradient

Like for the simulation of other kinds of system, the evolution of time should be modeled when simulating MAS. Especially, all the agents are supposed to act and interact concurrently with respect to the principle of causality [FTPD98].

Time could be modeled using three main approaches: (1) continuous time (by means of time functions which can compute the system state for any time stamp, e.g. DES), (2) discrete time (time evolves discretely with respect to constant time intervals), and (3) discrete event-based (time evolves discretely from one event to the next considering a continuous time line, i.e. the time interval between two events could be any real number). In this respect, modeling time for a MAS supposes to consider at least three aspects:

1. The modeling of the behavior time of the agents
2. The modeling of the endogenous dynamics of the environment
3. The evolution of the environment with respect to the agent actions.

Modeling the Behavior Time of an Agent

The models which are used to represent how an agent reacts to external events and deliberates are inherently discrete. Indeed, although an agent could be put in an environment which dynamics are modeled using a continuous approach, the deliberative process of an agent is usually modeled as a process that makes its internal state variables change discretely, i.e. instantaneously. Moreover, in almost every agent-based model, the perception/deliberation/action cycle is associated to a single instant t for simplicity reasons.

However, in some application domains, the temporal thickness of the behavior of an agent has to be explicitly taken into account, and therefore modeled. For instance, in the scope of DAI, the evaluation of the deliberation efficiency of an agent could be the aim of the study. For example, this efficiency could be modeled as a function of the actual computation time which is used by a deliberating agent, like in the Phoenix simulator [CGHH89]. The goal is to be able to measure and compare the efficiency of different agent architectures.

For example, the Multi-Agent System Simulator (MASS), which has been designed to study the efficiency of agents' deliberation in multi-agent coordination/negotiation processes, explicitly integrates time models of the mechanisms involved in the deliberation processes (method invocations, access to resources, etc.) [VHL01]. A similar approach has been used in the Sensible Agents platform [BMM$^+$01] to experiment with agent-based systems in dynamic and uncertain environments. Other example, Uhrmacher and Schattenberg have proposed to model the overall behavior of an agent as discrete event system using the DEVS (Discret Event Systems Specification) [Uhr01]. Doing so, the authors are able to model not only the deliberation time of an agent, but also the reaction time of an agent to external events, thus explicitly distinguishing, within the modeling of time, between the reactive and the proactive processes of an agent. The James simulation platform relies on this approach [SU01]. In this book, the reader will find a chapter about the new version of this platform, namely James II [HR08].

Modeling the Temporal Evolution of the Environment

Depending on the application domain, modeling the temporal evolution of the environment could be crucial. For instance, simulating agents in a network, it could be interesting to model different lag times to evaluate the impact of network congestion, node failure, and so on. Moreover, the environment may not only react to the agents inputs but also evolves according to its own dynamic, namely its *endogenous* dynamic. For instance, in a robocup simulation, a rolling ball continues to move even when the agents do not perform any actions.

As the environment could represent very different kind of systems, continuous or discrete time modeling could be considered depending on the experiment requirements. However, due to the fact that the agents have their perception and produce their actions in a discrete manner, the environment temporal dynamics generally embeds some event based mechanisms to ease the coupling between the model of the environment and the agent models.

Coupling the Agents and the Environment: Scheduling the MAS

Once the agent behaviors and environment dynamics are defined, they have to be coupled to finally model the targeted MAS, so that a MAS could be considered as three-tuple: $MAS =< Agents, Environment, Coupling >$ [Par97]. Figure 1.10 gives the overall picture of this coupling.

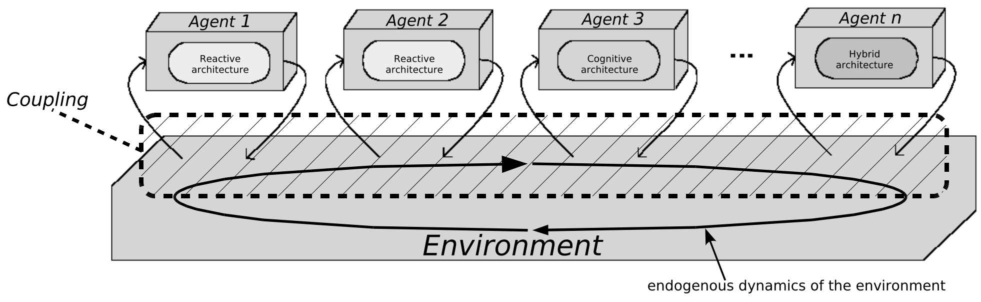

FIGURE 1.10 Coupling the agents and the environment

The main problem of the coupling is to make it coherent with respect to time [Par97, FTPD98]. In other words, achieving this coupling relies on defining a function *Evolution* such that the evolution of the MAS from one moment t to the next $t{+}dt$ results from the combination of the agent actions, $A_1(t), A_2(t)...A_n(t)$ with the dynamics produced by the natural evolution of the environment, $E_n(t)$, at t :

$$sigma(t + dt) = Evolution(\uplus(A_n(t), E_n(t)), \sigma(t)) \ . \qquad (1.1)$$

The symbol $\uplus$ is used here to denote the action composition operator. It defines how the actions produced at the instant t must be composed in order to calculate their consequences on the previous world state $\sigma(t)$. Intuitively, this coupling does not seem to be more than scheduling the activation of each model considering the modeling of time which is chosen (i.e. discrete time or event-based). However, several difficulties exist in the scope of ABM as we will now briefly see.

Due to the simplicity of its implementation, discrete time simulation (i.e. dt is constant) is the most used technique for simulating MAS. Indeed, in a discrete time simulation, all the models are sequentially activated for a time t, and then the global clock of the system is increased of one time unit. So, the implementation can be as simple as the following loop:

```
while(globalVirtualTime != endOfSimulation){
        for (SimulatedAgent agent : AllTheAgents)
                agent.act(); //perception, deliberation, action
        virtualEnvironment.evolve(); // for a dynamic environment
        globalVirtualTime++;
}
```

However, there is a major issue with such a scheduling technique: The order in which each model is activated may change the result obtained for the system state since the environment evolves for each action. This in turn may lead to very different system dynamics with the same behaviors [MGF04]. Figure 1.11 illustrates this issue on a prey/predator problem: the prey (the triangle) could be dead or alive depending on the activation list.

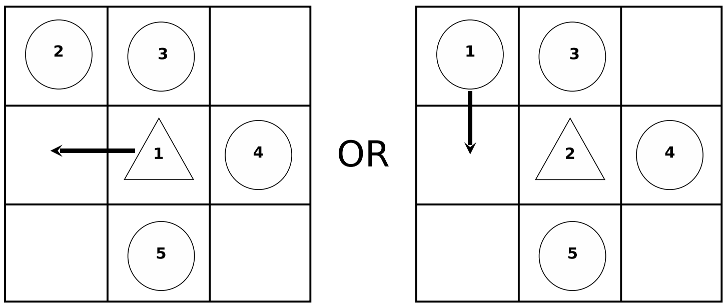

FIGURE 1.11 The prey could be dead or alive depending on its rank in the activation list

Several variations of the discrete time approach have been proposed to solve this problem. One solution is to systematically shuffle the activation list at each time step in order to smooth the problem (e.g. [EA96]).Another solution is to simulate concurrency by having all the agents operate on temporary variables, so that the perceived environment is the same for all the agents. Once done, the next system state is computed. So, the activation list has no effect on the results. However, the new problem is that one has to solve the conflicts which may exist between two or more actions that produce different states for system (e.g. the same ball at two different places). Solutions to such conflicts could be hard to find and lead to complex solving algorithms, as discussed in [DH01] for instance.

More generally, whatever the time management (discrete time or event-absed), modeling interaction (concurrent actions) in MAS is hard to achieve using action representations such as the one presented in the end of section 1.4.1 because they directly match the decision of an agent to a modification of the environment: One action gives one result. Quoting Parunak [Par97], *this leads to an (unrealistic) identity between an agent's actions and the resulting change in the environment, which in turn contributes to classical AI conundrums such as the Frame Problem*, as also further discussed in [FM96]. Therefore, the actions (equation 1.1) are not composed to produce a result but rather produce results one by one, which indeed leads to conflicting, sometimes unrealistic, situations. So, implementing simultaneity using such procedures may still be done but it takes complex codes which are more related to programming tricks than to an efficient and generic simultaneity modeling solution [Fer99]. In section 1.6.2, we will see that several works have addressed this particular problem by relying on the Influence / Reaction model [FM96].

Besides concurrency, another issue which is raised by a discrete time approach is the temporal granularity of actions. Indeed, in the same way the discretisation of the environment raises problems about the modeling of heterogeneous perceptions/actions, the discretisation of time implies to take care of the temporal granularity of actions. Figure 1.12 illustrates this problem: If an agent is able to move across more than one spatial unit in one time step, then several agents may have crossing paths without generating any collision, whatever the discrete time algorithm (i.e. there is no conflict here). This is an important issue in MABS involving mobile agents like traffic simulations (e.g. [DS04b]).

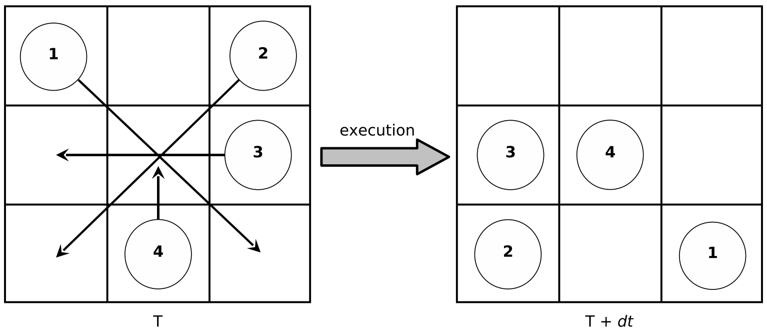

FIGURE 1.12 The problem of the temporal granularity of actions

So, depending on the application domain, an event-based approach could be required as it represents a far more flexible approach when it comes to the modeling of heterogeneous actions because it eases the integration of actions having different granularities. Moreover event-based simulations are often considered as more representative of the reality that we observe in the everyday life. Indeed, it is difficult to think about reality as a system in which all the entities would be "updated" simultaneously [HG93].

Like for the discrete approach, there are various way of modeling and implementing an event-based approach, depending on how the events are generated and handled (classical event-based implementations are presented in [Bal88]). The event list could be determined using a predefined order, like in the SWARM platform [MBLA96], or generated dynamically during the simulation, like in JAMES which uses DEVS [HRU03], or in DIMA which relies on the activity notion [Gue00].

1.4.5 Simulating MAS as Three Correlated Modeling Activities

To sum up this section, we propose here a high level view of agent-based simulation models. To this end, we consider that simulating a MAS relies on three correlated modeling activities: (1) the modeling the agents, (2) the modeling of the environment, and (3) the modeling of the coupling between the agents and environment models. Figure 1.13 illustrates this view:

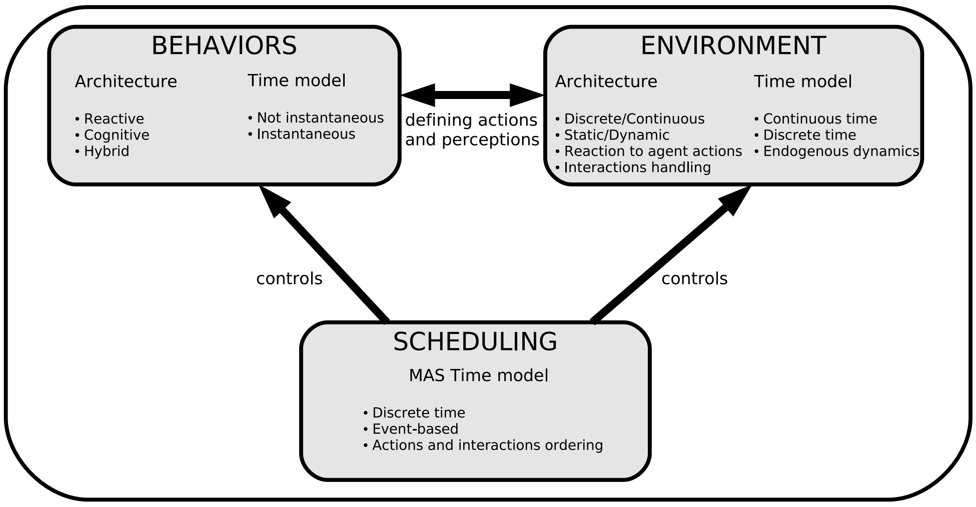

FIGURE 1.13 Simulating a MAS as three modeling modules

- The first module is related to the modeling of the agent behaviors, namely the *behavior module*. Considering this module especially involves to choose an agent architecture and a behavior time model.
- The second module, the *environment module*, is related to the definition of the virtual place wherein the agents evolve and interact. Modeling the environment notably supposes to decide if the environment should be (1) discrete or continuous, and (2) static or dynamic. If the environment has an endogenous dynamic,

then a time model of this dynamic should also be defined. Additionally, as agent interactions are mediated by the environment, how the agents interact is mainly defined through the modeling of the environment, by defining (1) how the actions of the agents affect the environment state and (2) how these actions are perceived by the agents (messages and/or signals).

- Finally, the *scheduling module* defines how all the models, environment and agents, are coupled and managed with respect to time. Besides choosing a particular time management, considering this module implies to define how the environment evolution, the agent perceptions, actions and interactions are done with respect to one another, thus defining the MAS causality model. It is worth noting that this fundamental part of the modeling is not always specified nor even considered during the elaboration of agent-based models as it is often regarded as something for which the simulation platform should be responsible, thus ignoring a major part of the model dynamics*. In section 1.6.3, we further discuss why defining this module should be considered as a primary modeling activity.

This tripartite perspective also defines how the modules are related to each other. The scheduling module controls how the agents and the environment are executed considering the representation of time (e.g. actions and interactions management and ordering). The relation between the agent module and the environment module concerns the definition of all the possible agent actions and perceptions with respect to the environment. This relation is bidirectional since adding new perceptions or actions requires to modify both modules.

1.4.6 A Still Incomplete Picture

This overall picture proposed in figure 1.13 highlights the different modeling activities which have to be considered when designing a MAS simulation model. Each of these activities raises very different issues according to the considered application domain. Most of them are discussed in detail throughout this book. However, it is difficult to identify what are the actual challenges for these different modules without going into the details of some application domains. However, in the scope of this chapter, we want to provide a more global perspective of the intersection between MAS and simulation that will help to identify some fundamental questions and general concerns for the next years.

Going toward a global perspective of the intersection between MAS and simulation, it is necessary to understand that the modules view is rather incomplete. Indeed, it is only a "MAS-side" view of the problem, in the sense that it focuses on MAS design issues. The other side of the coin is simulation itself. Indeed, simulating a system, whatever it is, also relies on considering some general issues about the design process of the experiment. To make this point clear, let us consider the Fischwick's definition for simulation [Fis97]:

> "*Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output.*"

Fischwick thus distinguishes between three fundamental activities:

*Other researchers usually have a different perspective and consider scheduling as an issue at the level of the simulation framework, rather than at the level of modeling.

1. Model design: The simulation is expressed in a model that specifies the characteristics of the simulated system.
2. Model execution: The model specifications are implemented in concrete computational structures that constitute the *simulator* of the experiment.
3. Execution analysis: The outputs of the simulation are checked according to some validation rules and then interpreted.

By avoiding to state what is the goal of simulation, the idea of Fishwick is to emphasize that simulation is a real discipline, not only a tool. Like any other, this discipline has its own characteristics and requirements (terminology, methodology, etc.) which are independent from the way it could be apply.

Therefore, understanding the different issues which exist considering both MAS and simulation, it is also necessary to have some insights about the general issues which are related to M&S. To this end, the next section briefly presents the *Framework for M&S* which has been originally proposed by Zeigler in the beginning of the seventies [Zei72, ZKP00]. In the scope of this chapter, the interest of the Framework for M&S is twofold: (1) Put in the light the relevance of studying MAS simulation according to this framework, and (2) highlight some major challenges and issues for future research in the MAS community.

1.5 The Zeigler's Framework for Modeling and Simulation

In [ZKP00]*, Zeigler proposed the *Framework for M&S* which relies on defining the *entities* of the framework and their fundamental *relationships*. This section briefly presents these entities, namely the *source system*, the *experimental frame*, the *model*, and the *simulator*, and these relationships, namely the *modeling relation* and the *simulation relation*.

1.5.1 Source System

As shown by Fishwich's definition, the terms *real system* used in Shannon's definition do not necessarily refer to a real-world system, but more generally to the system which is the object of the current study, i.e. the phenomenon to simulate. This is what is called the *source system* in the Framework for M&S: The real or virtual system which is targeted by the simulationist. Thus, a source system may define a theoretical phenomenon which first exists only in the mind of somebody, and which does not exist before it is simulated. So, this terminology enables to distinguish the real or virtual targeted system from the system which is finally obtained by executing the implementation of the model on a simulator.

If the source system does exist in the real world, it should be considered as the source of observable data that define what is called the *behavior database* of the system (i.e. a relation between inputs and outputs). For instance, if the purpose of the simulation is to study a real ant farm, it is possible to collect data about the ants (e.g. lifetime, population dynamics, etc.).

1.5.2 Experimental Frame

*The first edition of this book, *Theory of Modeling and Simulation*, was published in 1976.

Since a source system could be modeled considering very different purposes and objectives, the definition of a source system should be always done in the scope of an associated *experimental frame*. Formally, an experimental frame is a specification of the conditions under which the system is observed or experimented with. An experimental frame is the operational formulation of the objectives that motivate a modeling and simulation project.

Stating the experimental frame is crucial. Firstly, it deeply influences how the modeling, validation and verification processes are done with respect to (1) the information available about the source system and (2) the level of abstraction at which the source system is considered with respect to the resources and objectives of the experiment*. For instance, for the study of a real ant colony, taking into account the velocity of the ants within the modeling could be of interest, or even crucial in some cases, while not relevant in others. Therefore, in a more general perspective, the experimental frame could be considered as the definition of the experimental context. Especially, taken in a broad sense, this includes the used modeling paradigm and its associated constraints as discussed later on.

1.5.3 Model

Model is an overloaded term. However, in the scope of the Framework for M&S, a computer simulation model has a precise meaning. A model refers to all the computational specifications (instructions, rules, equations, and so on) that entirely define how all the possible state transitions of a system operate. In other words, a model is the specification of the *Evolution* function (equation 1.1). This specification could be written using different simulation formalisms or languages.

1.5.4 Simulator

A simulator is any software architecture on which the model could be executed to generate its behavior. In other words the simulator is responsible for computing the *Evolution* function and producing the outputs of the model with respect to acceptable inputs. A simulator could also embed some verification and validation tools. Depending on its general purpose, a simulator could be designed for only one experiment or a wide class of models.

1.5.5 Modeling Relation: Validity

The modeling relation concerns the study of the relation between a model and a source system in the scope of a particular experimental frame. This relation thus globally raises issues about the validity of the experiment by comparing the model (and its behavior) with the source system (and its behavior database if it exists). Here, the main question is to know if the modeling which has been done could be validated as an acceptable abstraction of the source system, considering the chosen qualitative criteria and experiment objectives. To answer this question, Zeigler proposes three main levels of validity:

1. *replicative validity* is achieved if the behavior of the model which is observed matches the behavior of the source system considering an acceptability threshold;
2. *predictive validity* implies replicative validity but also requires that the model is able to predict unseen source system behavior, which requires that the state

*See [BS05] for a focused discussion on these issues in the scope of ABSS transition function does reflect how the source system actually reacts to any acceptable inputs;

3. *structural validity* implies predictive validity but also requires that the model mimics the way in which all the components of the source system do their state transitions.

1.5.6 Simulation Relation: Simulator Correctness

The *simulation relation* concerns the verification of the simulator with respect to the model. The question is to ensure that the simulator does generate correctly the model behavior. In other words, the simulation relation holds if the simulator guarantees that the consecutive states of the model are correctly reproduced. Conversely, it is interesting to note that this implies that the model specifications should entirely define the state transitions in both a (1) computerizable and (2) unambiguous way, otherwise the simulation relation cannot hold as it is possible to implement different interpretations of the model, thus implementing simulators relying on different state transition functions, which in turn produce different results for the same model.

Additionally, as the model specifications must be independent from the way they are executed by the simulator, the simulation relation also implies that the evolution of a simulated system must not depend on any simulator or hardware-related issues, enabling the replication of the experiment, which is fundamental from a M&S perspective. For instance, let us considered a simulated MAS executed on a Java Virtual Machine (JVM). If the evolution of any part of the system does depend on how the JVM manages the Java threads, replication is not achievable: the system is not simulated, but more simply executed.

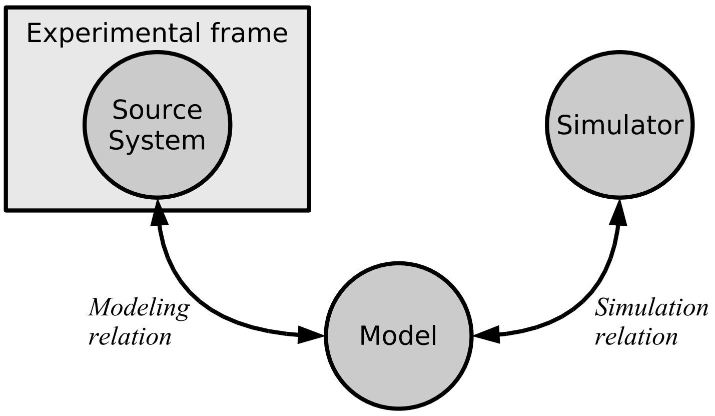

FIGURE 1.14 The Zeigler's Framework for M&S

1.5.7 Deriving Three Fundamental Questions

Considering the use of simulation, the Framework for M&S has several advantages. Firstly, it gives a precise meaning to the different keywords which are related to simulation. Secondly, by relating these different key words to each other, thanks to the modeling and simulation relations, the Framework for M&S highlights the basic issues and problems which are related to any M&S experiment. Especially, it gives some clear guidelines to study the problem of the validation and verification (V&V). Although the purpose of this chapter is not to directly discuss V&V issues in the scope of MAS simulations, it is interesting to study MAS simulation works with respect to the basic relations of the Framework for M&S. Especially, we propose to do this study according to three questions derived from the Framework for M&S:

1. Does the model accurately represent the source system? (modeling relation: model vs. source system)
2. Does the model accommodate the experimental frame? (modeling relation: model vs. experimental frame)
3. Is the simulator correct? (simulation relation: model vs. simulator)

So, studying how these questions could be answered in the scope of MAS simulations, the main goal of the next section is to extract and discuss some important issues which represent relevant directions for future research within the MAS community.

1.6 Studying MAS Simulations Using the Framework for M&S

1.6.1 Does the Model Accurately Represent the Source System?

Identifying the Nature of the Source System

One can easily see that the issues related to the study of the modeling relation (validity) strongly depend on the source system being actual or theoretical. Indeed, in the case the source system is purely theoretical, structural validity is a given property since the model does represent the actual state transitions of the system: They do not exist elsewhere.

On the contrary, if the source system is actual, then the modeling should be done with great care according to the behavior of the source system in order to, at least, achieve replicative validity. For instance, in a DAI platform that simulates the environment as a network, like in the DVMT experiment, the behavior of the simulated network has to match with real-world conditions (messaging time, resource access time, node failure, etc.) to some extent to be valid. So, doing a simulated MAS, one has to clearly state the nature of the source system which is considered: Actual or theoretical.

One could argue that it could be subtil to decide if a source system is actual or theoretical, which is true. For instance, although ABSS are inspired by real-world human societies, the level of abstraction used is sometimes so high that the model is considered as purely theoretical, while being inspired by real-world dynamics (e.g. SugarScape [EA96]). Validation issues related with an actual vs. theoretical perspective are discussed in papers such as [MDSC04, BS05] where continuum classifications of ABSS models are proposed according to different goals, levels of abstraction, and empirical data availability and gathering. The former proposes a general layered classification of paradigmatic models in ABSS with respect to their targets. The latter gives a focused discussion on related methodological issues and proposes a model maker perspective taxonomy of ABSS models (from case-based models to theoretical abstractions), suggesting various validation strategies according to the characteristics of the *empirical target matter*. So, by characterizing and layering the level of abstraction of ABSS models, these approaches help to decide how the modeling should be validated and such approaches could be useful in other domains.

Nonetheless, defining the level of abstraction of a model is a subjective task to some extent, and it would probably be possible to define hundreds of model classes if all application domains would be considered at once. In the scope of this chapter, we need a more general criterion to decide if a source system should be rather stated as actual or theoretical. In this respect, it is important to understand that such a criterion should not rely on the level of abstraction which is used in the modeling. And, in fact, it is possible to solve this problem in a general way with respect to the existence of the behavior database. Indeed, if a behavior database could be established, then the results could be checked according to it, thus the source system has to be considered as actual, whatever the level of abstraction.

Paradoxically, this also holds when it is not possible to establish the behavior database before the experiment. For instance, a large number of DAI operational systems, such as DVMT, are first design using simulation models before being deployed in real-world computational structures (when they finally are). Therefore, no behavior database is available a priori. However, if the system could be deployed, thus a behavior database could be established and used to check replicative validity for the model. So, deciding if a source system is theoretical or actual, one has to check if it could be possible to establish a behavior database for the system. If it is not the case, the source system is theoretical.

Different Modules, Different Issues

Still, the previous analysis is not complete and should go further: In the scope of MAS, raising the question about the nature of the source system, it is important to make a distinction between the behavior module and the environment module. Indeed, it is possible to define a system with theoretical behaviors embedded in a very realistic environment. So, it is possible to distinguish between four situations as illustrated in Fig 1.15.

Each situation requires a different approach for validating the modeling: From fully theoretical systems for which structural validity is always true to fully actual systems for which both the behavior module and the environment module have to validated (at least with respect to replicative validity) according to the behavior of their targeted source systems. Thus each situation raises different issues when considering the modeling validation. Let us now discuss some of these issues according to the classification proposed in 1.15.

Environment Module Issues: The Need for Virtual Reality

From a validation point of view, the simplest case is to have a fully theoretical model. For instance, in [BMF06], a morphogenesis model for multiagent embryogeny is proposed. Although this model is inspired by real biological processes, the purpose of both the behavior and environment modules is not to mimic any real system at all, but only to define a system which is able to produce emergent phenomena having interesting properties such as self-organisation and the like. Therefore, there is no need to check the validity of the model with respect to the source system: The model is the source system itself, and therefore structurally valid. As we will see later on, that does not mean that the overall modeling could be directly considered as correct, but it clearly remains the less problematic case.

Next in difficulty are models where theoretical behaviors operate in an actual environment, namely in virtual reality (VR). For instance, in the Sims's work [Sim94], virtual creatures evolve in a 3D world. As the Sims's goal is to study the evolution of the virtual creatures in a realistic world, the environment model embeds complex dynamics such as

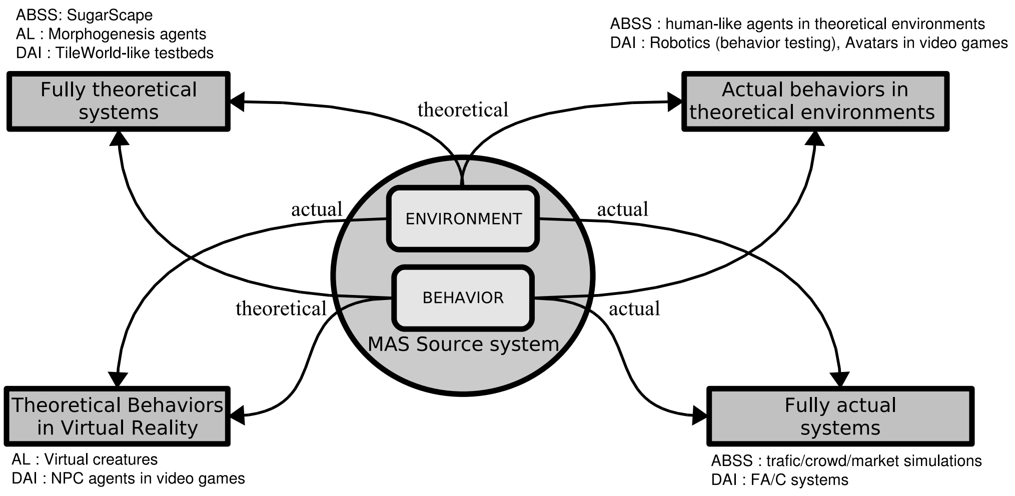
ABSS: SugarScape
AL : Morphogenesis agents
DAI : TileWorld-like testbeds
Fully theoretical systems
ABSS : human-like agents in theoretical environments
DAI : Robotics (behavior testing), Avatars in video games
Actual behaviors in theoretical environments
theoretical
actual
actual
ENVIRONMENT
BEHAVIOR
theoretical
actual
MAS Source system
Theoretical Behaviors in Virtual Reality
AL : Virtual creatures
DAI : NPC agents in video games
Fully actual systems
ABSS : trafic/crowd/market simulations
DAI : FA/C systems

FIGURE 1.15 Simulation and MAS with respect to the source systems of the behavior and environment modules

gravity, collisions, friction and so on. In such a case, structural validity is true for the behavior module as the agents are purely virtual (i.e. the source system of the behavior module is theoretical), while the model of the environment has to be validated according to real world dynamics (i.e. the source system of the environment module is actual). Therefore, from a validation point of view, the environment module is more important than the behavior module considering Simulated MAS applications that use VR. For instance, designing the agents AI in realistic video games, it is crucial that, like for the virtual creatures of [Sim94], "*the environment restricts the actions to physically plausible behaviors*": Observing and studying the behavior of the agents is not relevant if the modeling of the environment is not valid.

If having a realistic model of the environment is a strong requirement in the scope of VR-based applications, it is very interesting to see that the need for realism has also been pointed out as a primary concern in the DAI field relatively early, in the late eighties. For instance, studying teamwork and coordination of fire fighting units, a primary goal of the Phoenix testbed was to simulate an environment as realistic as possible* [CGHH89]:

> "*Our position is that most AI systems have been built for trivial environments which offer few constraints on their design and, thus, few opportunities to learn how environments constrain and inform system design. To afford ourselves this opportunity, we began the Phoenix project by designing a real-time, spatially distributed, multiagent, dynamic, ongoing, unpredictable environment.*"

*Studying forest fires management in the Yellowstone National Park, Phoenix defined a highly dynamic environment thanks to a complex simulation integrating several relalistic parameters such as weather conditions, physical features of the park, and burn rates of different land types.

Later, during the DAI testbeds explosion of the nineties, Hanks putted in the light the "*danger of experimenting in the small*" in [HPC93]. Hanks's point was that DAI testbeds such as the TileWorld do not really fulfil their objectives, i.e. making progress in both the engineering and the understanding of (efficient) DAI systems, because most of the time they consider small, controlled, and deterministic world:

> "*THe ultimate danger of experimentation in the small is that it entices us into solving problems that we understand rather than ones that are interesting. At best it gives the mistaken impression that we are making progress toward our real goal. At worst, over time it confounds us to the point that we believe that our real goal is the solution of the small, controlled problems.*"

Today, it is clear that experimentation in the small is still a reality, not only in the DAI field. Of course, from a pragmatic point of view, we always have to experiment in the small to some extent: We just cannot simulate all the details of reality and should focus on the aspects of the environment which are the most relevant according to the experiment objectives. However, one has to bear in mind that, with respect to the experiment objectives, the modeling of the environment could be crucial when evaluating the appropriateness of the overall modeling. In this respect, one major challenge for DAI researches in the years to come is to move from theoretical environments to actual ones: Putting DAI systems into VR, and thus *experimenting in the large*. Such a point of view also nicely bridges the gap between concerns from Artificial Life and DAI, highlighting the importance of the environment module for both [LA00]. Considering such an issue, the RoboCup [KAK$^+$97] is a really interesting step in the history of DAI testbeds because it represents a good deal between realism and DAI testing. Additionally, one can find in this book a chapter dealing with the fundamental role of the environment for software-in-the-loop simulations [HHW08].

Behavior Module Issues: Toward the Participatory Design of Simulation

Although checking the validity of an actual environment module could be a hard and time consuming task, the general approach itself is relatively straightforward, because of our everyday life experience of the causality principle. Indeed, as we are living inside this real environment, we can directly observe it so that we are able to deeply analyse its functioning. So it is easy (relatively of course) to see that an environment module does not agree with real world dynamics, thanks to our own experience. Objects are supposed to be under the influence of the gravity force for instance. Comparatively, it is much more difficult to decide about the appropriateness of the behavior module when facing the modeling of real-world behaviors.

In fact, considering the modeling of actual behaviors, we have to distinguish between two different cases: (1) behaviors that refers to real-world computational structures (robots, software agents, etc.) and (2) behaviors of real-world living entities (insects, animals or human).

In the first case, checking the validity of the behavior module is not more difficult than for the environment because we deeply know the real structure of the related behaviors as we created them. For instance specifying the behavior module of a real robot simply consists in modeling how the behavior program of the robot runs on some concrete hardware in real conditions (software-in-the-loop). The same goes for an e-commerce agent as well.

However, when it comes to the modeling a living entities, we do not have such information. In contrast with the environment, we cannot observe the internal mechanisms that effectively conduct a living entities to make decisions: A living creature behavior is a black box which internal mechanisms could be only modeled using very high level abstractions.

Consequently, it could be difficult to decide about the validity of a behavior module with respect to the experiment objectives as we are, once again, always experimenting in the small to some extent: Validating a behavior module is a highly subjective task in such cases. Moreover, evaluating a human-like behavior module could be by a huge order of magnitude more problematic than one representing an ant-like behavior. It is obviously a reason why there is a growing trend toward a participatory design of simulation: We need to "put the human in the loop".

In [DVM02], Drogoul et al. highlight the huge gap which usually exists between the behaviors of the domain model (the behavior module) and the computational structure which are finally used to implement them:

> "*This means that the resulting "computational agents" (if we can still call them this way) do not possess any of the properties generally assigned to the agents used in MAS or DAI : they do not have any structural nor decisional autonomy, are not proactive, and cannot, for instance, modify their knowledge and behaviors through learning or adaptation.*"

The authors thus proposed a MABS methodological process that more concretely involves real stakeholders in the modeling activity, thus advocating the idea of the participatory design of simulations. Agent-Based Participatory Simulations could be considered as a more strong merging between MAS and Role Playing Game (RPG) [GH06]. RPG was used in [BBA01] to work on both the design and the validation of social MABS models. Still, in such works, the participants of the RPG do not directly control an agent of the simulation and the agents behavior are still encoded as computational structures. So, the main idea of agent-based participatory simulations is to have real humans playing the role and taking control of simulated agents to design and then refine the behavior module through an iterative modeling process (with steps with or without real human agents) [NDD07, GH06] . The main advantage is that, as all the events could be recorded, the behavior of the participants could be analysed in details during and, more importantly, after the simulation runs using the trace of the simulation execution. This helps to understand why and how the participants take decisions with respect to the information they gather during the simulation. So, the participatory design of simulation helps for the modeling of more accurate artificial behaviors, thanks to the refining of the perception/deliberation/action processes which is done at each iteration of the modeling loop proposed by this approach.

Although motivations for the design participatory simulation are numerous, in the scope of this chapter, we can summarize the major idea of such approaches as follows. Modeling the behavior of a real living entity is hard and structural validity could not be an objective at present time considering the behavior module. We need high level abstractions. Building these abstractions, rather than trying to model existing behaviors from scratch using complex deliberation models, and thus searching for an unreachable structural validity, we should focus on studying what we could observed and analysed from real living entities: The actions they produced with respect to their perceptions. In other words, rather than focusing on the structural aspect of the behavior module, we should first focus on its ability to achieve replicative validity, thus relating perceptions with actions, i.e. inputs with outputs.

If the relevance of participatory simulations is obvious considering social simulations, it could also be very interesting to use a similar approach for the modeling of non human species. The interest of the idea still holds: It could help for the design of the behavior of artificial entities as it is of course possible for a human to play the role of a predator agent for instance. Studying what makes a human decide what are the relevant perceptions/actions may significantly help for refining both the behavior module (modeling deliberation) and the environment module (adding perceptions/actions).

Merging Participatory Design and VR: a Promising Future for MABS

In the previous sections, we have discussed only about few issues which are related to the study of the modeling relation in SMAS. However we have highlighted two very interesting tracks of research, each of which being related to either the environment module or the behavior module: That is respectively VR and the participatory design of simulation. In this respect, it is worth noting that very recent works integrate both in a very innovating but unexpected manner.

In the pioneer work proposed in [INMN07], Ishida introduces the notion of *augmented experiment*, in the scope of real-world experiments on disaster evacuation. The idea of a Multiagent-Based Augmented Experiment is to do real-world experiments, with real-world subjects evolving in an *augmented reality*: That is, in a real-world environment where actions of simulated entities, evolving in a corresponding VR, could be perceived by real-world subjects thanks to the use of suitable sensor devices. In this perspective, the related design process of the experiment is particularly interesting as it involves a participatory design loop for modeling VR, which is in turn used to do augmented experiments. Doing so, such a work establishes a link between reality and VR as well as between humans and virtual entities.

Here, we are at the frontier of what we can do with our today technologies as augmented experiments raises a number of technical problems [INMN07]. However, this is obviously a forerunner example of how we could finally experiment in the large in a near future. In such a perspective, VR and participatory simulation are key concepts.

1.6.2 Does the Model Accommodate the Experimental Frame?

MABS are Supposed to Model MAS

One could think that considering a theoretical model implies that the overall modeling could be directly validated. Especially, as discussed by Troitzsch in the scope of ABSS, a simulation programme can be seen as a full model of a theory as it can be considered as a *structuralist reconstruction* of this theory as discussed [Tro04].

However, beyond this legitimate consideration, it is also crucial to not forget that the modeling relation is tripartite, implying that the source system and the model should always be considered in the scope of a particular experimental frame, which defines modeling constraints both explicitly and implicitly. Therefore these constraints must be taken into account in the validation process, whatever the nature of the source system. Notably, this is also true for fictitious worlds because they do also define environmental laws that impose consistency and coherence, as does the real world [VSSRA07]. For instance, if a theoretical source system is supposed to embed usual physical laws, then the modeling should not define state transitions which contradict these laws: It is not a valid modeling otherwise as it is an implicit constraint of the experimental frame.

So, as pointed out by Zeigler, it is crucial for the validity of a model that it at least coherently accommodates the experimental frame, thus verifying the *accommodation relation* [ZKP00]. Considering this relation in the scope of MAS is more challenging that it seems to be at first sight.

Verifying the accommodation relation, it is obvious that both the behavior module and the environment module must follow some modeling guidelines so that they do not integrate state transitions contradicting the experimental frame. For instance, it is obvious that an agent must not be able to perform an action that makes it go through a wall. Therefore, there is not really a need to put forward the importance of such concerns here.

Nonetheless, all MABS rely on an implicit, but strong, hypothesis: They use MAS as modeling paradigm. As such, their modeling should accommodate the properties of the agent paradigm, whatever the objective of the simulation study. Verifying this particular accommodation is related to the study of what we define as *paradigmatic validity/coherency*[*], i.e. the validity of the model with respect to the multiagent paradigm itself. In the following section, we only focus on two consequences underlying the consideration of such a level of validity.

Modeling Simultaneous Actions and the Autonomy Feature

Firstly, that means we should be able to represent simultaneous actions as agents are supposed to act in a concurrent manner on the environment. As Ferber has highlighted, implementing simultaneity with usual formalisms is not trivial [Fer99].

Secondly, one has to bear in mind that agents are supposed to be autonomous entities. As pointed in [MGF04], it is easy to model systems wherein agents are finally not autonomous, since some of their behaviors are directly controlled by other agents.

Without going into the details of the problems which are related to each case, it is interesting to see that the Influence/Reaction model has been considered as a solution for both. Indeed, the Influence/Reaction model [FM96] relies on modeling the agent actions as influences that do not modify the environment (and the other agents as well), but which are treated as a whole to compute the reaction of the environment to these influences, and thus the next system state. So, it is possible to easily model simultaneous actions by combining the influences to compute the actual result on the environment. Works such as [HVUM07, Mic07, WH04, DT00] show the interest of such an approach for modeling simultaneous actions in different contexts. Moreover, as the agents do not directly modify the others' internal variables, it is easy to check that the autonomy feature is taken into account [MGF04].

Toward Paradigmatic Validity: The Need of Linking the Micro and Macro Levels

In the previous section, we discussed about the opportunity of using the Influence/Reaction model for simulating MAS. From a more general point of view, this shows that we need approaches and formalisms specifically dedicated to the ABM approach in order to accommodate the experimental frame. All this is about studying issues related to paradigmatic validity.

Considering this study, one difficulty relies on the fact that MAS define at least two levels of dynamics: the micro level and the macro level. Most approaches only consider the micro level specifications and do not evaluate their impact on the macro level dynamics (e.g. the possibility of representing simultaneous actions for instance). So, most of the time, there is a discrepancy between the system we want to model (i.e. the source system) and the model we finally produced. In this respect, [DSC02] shows the interest of specifying both the micro and macro levels for ABM. Notably, [DSC02] proposes modeling specifications at different levels of granularity: *atomic agentified entities (AE)* are specified at the micro level, and then aggregates of AEs (macro-AE) are specified at different levels of aggregation

[*]The word *validity* is intentionally used to make a parallel between paradigmatic validity and the other forms of validity defined in the Framework for M&S.

(first order macro level, second order, and so on). The Influence/Reaction model also makes a difference between the micro level (influences) and the macro level (the reaction of the environment to the influences) within the modeling.

Nonetheless, it is clear that many efforts have still to be done in such directions. One major challenge for ABM approaches is indeed to reduce the gap between what we are supposed to model (i.e. MAS) and the modeling abstractions and formalisms we use to do it. In the scope of paradigmatic validity, this is a necessary requirement if we want to have a clear understanding of the models we build, knowing they are becoming more complex everyday. In this perspective, the problem is not only to represent simultaneous actions or autonomy, but also to be able to express complex concepts such as stigmergy, self-organisation, emergence and the like, in the model specifications. To this end, making an explicit link between micro level and macro level specifications is definitively an avenue for future research. As we will see now, such considerations also impact the study of the simulation relation.

1.6.3 Is the Simulator Correct?

We now switch our focus from the modeling relation to the simulation relation, that is, to the study of concerns related to the simulator correctness.

A New Day, a New Simulator

From a MAS perspective, it possible to classify MAS simulators according to three categories according to their capability and focus:

1. limited to a single class of model: *One shot*;
2. able to accept a whole class of models related to a particular domain: *Domain related* (e.g. Ecology: Cormas [BBPP98], Echo [HJF97]);
3. designed for implementing MAS simulations using a particular agent software architecture, independently of a particular domain: *Generic* (e.g. Swarm [MBLA96], MadKit [GFM01], Jason [Bor08], SeSam [Kl8]).

It is worth noting that there are far more One-shot MAS simulators than the two other kinds. This clearly shows that there still no consensus on the way agent-based systems have to be modeled and implemented. Indeed, one question is: Why a new simulator would be required almost every time? One could say that it is due to the ability of MAS to address numerous application domains, each of which having their own modeling requirements. This is probably true to some extent. However, another answer is possible if we consider the problems raised in the previous section: The gap between the MAS paradigm and the modeling tools we used. This gap does not allow a clear understanding of MAS simulation frameworks. Consequently, it is not surprising that researchers prefer to define their own simulation tools in order to (1) fulfil their requirements and (2) control entirely the simulation process.

Having a majority of one shot simulators raises several problems for the community. Firstly, it severely reduces the possibility of reusing previous works, which is an undisputed drawback from a software engineering point of view. It is a sure bet to claim that similar things (agent and environment models, simulation engines, etc.) have been coded hundreds of time: The community hardly capitalizes experiences. Of course, it also increases the possibility of facing bugged simulators. So, one major challenge for the community is to find ways of doing MABS so that we can reuse and capitalize experiences in a more efficient manner.

Secondly, it raises the problem of reusing the simulation results themselves. Indeed, the simulation relation is rarely studied and few MAS works do consider the simulator as a first order entity of the experiment. Most of the time, the implementation of the model is considered as a step of the experiment which correctness would be true a priori, which is obviously a mistake. Moreover, this decreases the confidence we have in published simulation results. Therefore, not considering the simulation relation is obviously a weak point that should be addressed in the years to come because it dramatically limits how simulation experiments can be evaluated: Verifying the simulation relation is not less important than validating the modeling relation in the simulation process. To emphasize more on that point, let us now consider a fundamental consequence of this state of affairs: The difficulty of replicating existing models.

The Fundamental Problem of Replication

In section 1.4, we have shown that ABM implementations could raise many problems, in many different ways. So, it is clear that going from model specifications to the implementation is not a trivial task in the scope of MAS. Not surprisingly, simulation results are highly sensitive to the way the model is defined and implemented (see [MST08] for a recent and very interesting study of that issue). Works such as [Axe97, LP00, Rou03, EH03] also clearly address this issue by showing all the difficulties we face when trying to replicate published results. Especially, they show that, since simulation results are usually published without detailing enough the model and its implementation, replications have a high probability of producing results which are (very) different from the originals.

This problem is identified as the *engineering divergence phenomenon* in [MGF04], i.e. the possibility of implementing different simulations, considering a unique model. Such a problem contradicts at the heart a basic hypothesis of the simulation relation of the Framework for M&S : A model should be defined so that the produced results do not depend on how it is implemented (cf. section 1.5.6). In the previous section we claim that there is a gap between the MAS paradigm and the modeling tools/formalisms we used. We can make a similar statement here: There is a gap between the agent-based models which are defined and the computational structures we used to implement them. In fact, the complexity of the implementation, which is dramatically real, is usually simply bypassed when agent-based models are designed.

The scheduling module is particularly neglected during the modeling phase because it is often considered as only related to the implementation. This is a mistake: The scheduling module is fundamentally part of the modeling. Indeed, if this part is fuzzily or not defined at all, thus the model specifications are ambiguous and could be implemented following different interpretations, which in turn may lead to huge differences in the simulation results.

So, as highlighted by Axelrod in the scope of ABSS [Axe97], one major challenge for the community is to find ways of sharing modeling specifications in an efficient way, enabling the replication, and thus the study, of published experiments. To this end, it is clear that making an explicit link between computational structures and models is the key. In this respect, the Swarm platform design is of great interest [MBLA96]. Indeed, establishing a Swarm-based model explicitly relies on defining, not only the behavior and the environment modules, but also the order in which they are finally executed by the simulator. In other words, a swarm-based model is directly related to its implementation, so that it is easy to replicate it.

Generalizing such an approach is crucial for the years to come: Enabling replication is what the community should go for. To this end, we claim that two things should be placed on the agenda of multiagent simulationists: (1) Promoting the reuse of existing works as far as possible and (2) defining unambiguous models. The former will enable the community to go toward the elaboration of a real simulator software engineering process, ultimately making the reuse of existing works possible. The latter will help to really study the simulation relation in ABM by making an explicit distinction between the model and its implementation, which in turn will help to better understand the true dynamics of MAS as they will be explicitly modeled. Only such approaches will enable us to ultimately answer the fundamental question of this section: Is the Simulator Correct?

1.7 Conclusion

MAS and simulation are two research fields having a close relationship. Studying the intersection between these two fields, it is important to keep in mind that simulation is not only a tool, but a real scientific discipline which already defines some design guidelines. This is the reason why we have argued on the interest of considering MAS simulation works in the scope of a pure M&S perspective such as Zeigler's. Indeed, the Framework for M&S permitted us to study MAS simulation works according to the modeling and simulation relations, which are essential for any M&S experiment.

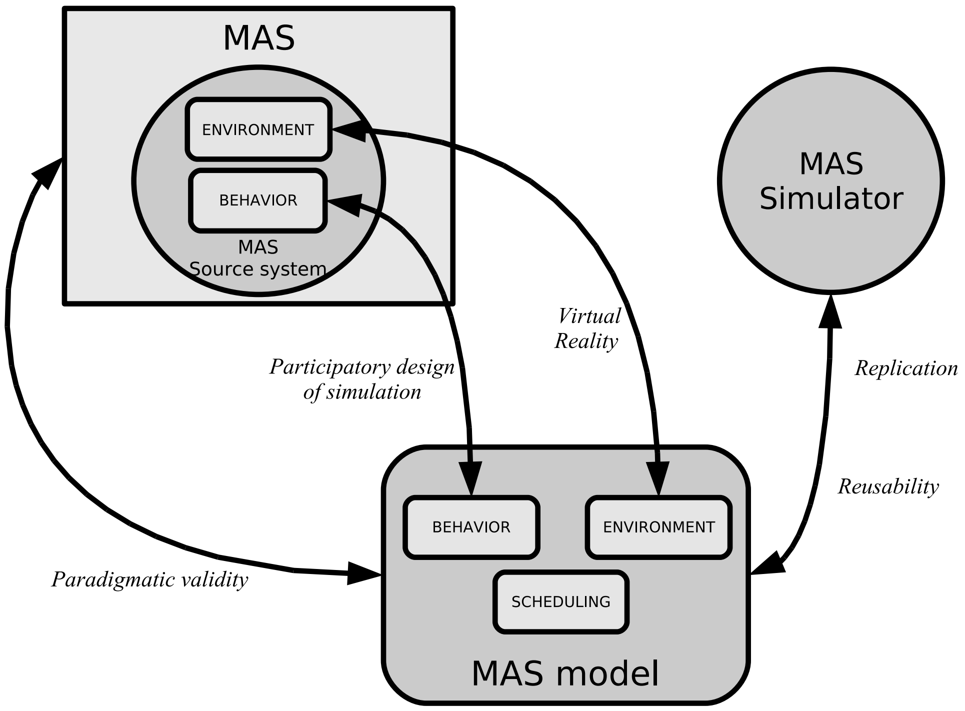

FIGURE 1.16 Definning MAS simulation issues using the Framework for M&S

Figure 1.16 illustrates the different challenges we have highlighted by studying MAS simulation according to Zeigler's Framework for M&S.

First of all, we have seen that VR is a concept of primary importance with respect to the environment module. Experimentation in the small has been useful for exploring some characteristics of the MAS paradigm in the past two decades. Today, the available technologies urge us to make the step toward VR-based models of the world to prove the usefulness of the existing DAI architectures and develop new ones, more adapted to real-world applications. Additionally, considering software-in-the-loop approaches, VR is the only way to go: Without an accurate model of the deployment environment, the system cannot be developed in an efficient way. The reader will find more about the crucial importance of the modeling of the environment in this book (ref Helleboogh). For the MAS community, VR is the key for experimenting in the large.

About the behavior module, we have emphasized the difficulty of building representative models of real living entity behaviors, either they are human or animal beings. Indeed, contrary to software agents (e.g. robots), the real structure of the decision process cannot be observed with our today technologies, so that we use theoretical approaches to model them. Therefore we have to find ways to reduce, as far as possible, the inescapable gap which exists between what we want to model (i.e. real decision processes) and what we could model using the theories and computational modeling abstractions available today. In this respect, we have discuss the relevance of the participatory design of simulation. There is major idea in this trend of research: Rather than model real behaviors so that they somehow fit our computational structures, we should make the computational structures evolve so that they can reflect how a real living entity is motivated to act (which perceptions for which actions). In other words, we should not directly use computational structures to model real behaviors, but use real behaviors to model new computational structures, which will in turn be more suited to our needs. Doing this shift of perspective is a major challenge for the MAS community, considering behavioral modeling.

The two previous directions of research represent paths toward the accommodation of future MAS models with the MAS experimental frame. Indeed, they force us to make progress toward simulation formalisms that take into account the specificities and features of the MAS paradigm, which in turn enables the study of issues related with paradigmatic validity. For instance, the Influence/Reaction formalism permits to model MAS basic concepts such as action simultaneity and autonomy. In a more general perspective, a major feature of ABM relies on the fact that it requires at least two fundamental levels of modeling: The micro (agent) and the macro (multiagent) levels. Most of the approaches do only consider the modeling of the micro level entities and do not clearly specify how their dynamics should be composed, leaving the developer free to choose during the implementation phase. Providing abstractions for the macro level dynamics, such as simultaneous actions, is a first step toward a concrete linking between these two fundamental levels.

Ultimately, studying the simulation relation, we have emphasized on a crucial issue for the MAS community: Capitalizing passed experiences. This is necessary if we want to speed up the research on MAS simulation in general, whatever the objectives. In this respect, it is obvious that the software engineering perspective is neglected by most of the MAS simulation works as the emphasize is always put on modeling issues. The MAS community has to find the way of concretely sharing implementation experiences, promoting reusability as a must. Without such concerns in mind, reinventing the wheel will continue to be a reality for the community. More than that, it is the only way through which we will be able to really do verification on MAS simulation experiments as it will ease replication, which is fundamental from a M&S perspective. Trusting simulator implementations a priori is not a viable option.

References

1. M. Arango, L. Bahler, P. Bates, M. Cochinwala, D. Cohrs, R. Fish, G. Gopal, N. Griffeth, G.E. Herman, T. Hickey, K.C. Lee, W.E. Leland, C. Lowery, V. Mak, J. Patterson, L. Ruston, M. Segal, R.C. Sekar, M.P. Vecchi, A. Weinrib, and S.-Y. Wuu. The touring machine system. *Communications of the ACM*, 36(1):69–77, 1993.
2. F.f.I.P. Agents. http://www.fipa.org. Accessed June 2008. FIPA Communicative Act Library Specification.
3. G. Agha. *Actors: a model of concurrent computation in distributed systems.* MIT Press, Cambridge, MA, USA, 1986.
4. R.C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.
5. J.L. Austin. *How to Do Things With Words.* Oxford University Press, 1962.
6. R. Axelrod. *Advancing the art of simulation in the social sciences*, pages 21–40. Volume 456 of Conte et al. [CHT97], 1997.
7. R.L. Axtell. Why agents? on the varied motivations for agent computing in the social sciences, csed working paper no. 17. Technical report, Center on Social and Economic Dynamics, The Brookings Institution, November 2000.
8. O. Balci. The implementation of four conceptual frameworks for simulation modeling in high-level languages. In M.A. Abrams, P.L. Haigh, and J.C. Comfort, editors, *Proceedings of the 20th conference on Winter simulation*, pages 287–295. ACM Press, 1988.
9. M.S. Bartlett. *Stochastic Population Models in Ecology and Epidemiology.* Wiley, 1960.
10. O. Barreteau, F. Bousquet, and J. Attonaty. Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to senegal river valley irrigated systems. *JASSS, The Journal of Artificial Societies and Social Simulation*, 4(2), 2001.
11. F. Bousquet, I. Bakam, H. Proton, and C.L. Page. Cormas: Common-pool resources and multi-agent systems. In *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial In telligence and Expert Systems*, pages 826–837. Springer-Verlag, 1998.
12. R.A. Brooks and J.H. Connell. Asynchronous distributed control system for a mobile robot. In W. Wolfe and N. Marquina, editors, *SPIE's Cambridge Symposium on Optical and Opto-Elecronic Engineering*, volume 727, pages 77–84, October 1986.
13. R. Beckers, J.L. Deneubourg, and S. Goss. Trails and u-turns in the selection of a path by the ant lasius niger. *Journal of Theoretical Biology*, 159:397–415, 1992.
14. B. Beaufils, J. Delahaye, and P. Mathieu. Complete classes of strategies for the classical iterated prisoner's dilemma. In *Proceedings of the 7th International Conference on Evolutionary Programming VII*, pages 33–41. Springer-Verlag, 1998.
15. A. Bretagnolle, E. Daudé, and D. Pumain. From theory to modelling : urban systems as complex systems. *CyberGeo: European Journal of Geography*, (335):1–17, March 2006.
16. G. Beurier, F. Michel, and J. Ferber. A morphogenesis model for multiagent embryogeny. In L.M. Rocha, L.S. Yaeger, M.A. Bedau, D. Floreano, R.L. Goldstone, and A. Vespignani, editors, *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 84–90, Cambridge, MA, USA, August 2006. MIT Press.
17. K.S. Barber, R. McKay, M. MacMahon, C.E. Martin, D.N. Lam, A. Goel, D.C. Han,

and J. Kim. Sensible agents: an implemented multi-agent system and testbed. In *Proceedings of the fifth international conference on Autonomous agents*, pages 92–99. ACM Press, 2001.

18. B. Bauer, J.P. Müller, and J. Odell. Agent uml: A formalism for specifying multiagent software systems. In P. Ciancarini and M. Wooldridge, editors, *Agent-Oriented Software Engineering, First International Workshop, AOSE 2000, Limerick, Ireland, June 10, 2000, Revised Papers*, volume 1957 of *Lecture Notes in Computer Science*, pages 91–104. Springer, 2001.
19. R.H. Bordini. Agent-based simulation using bdi programming in Jason. In Uhrmacher and Weyns [UW08], chapter 14.
20. G.M. Buettner and W. Siler. Variability in predator-prey experiments: simulation using a stochastic model. In *Proceedings of the 14th annual Southeast regional conference*, pages 194–201. ACM Press, 1976.
21. R. Boero and F. Squazzoni. Does empirical embeddedness matter? methodological issues on agent-based models for analytical social science. *Journal of Artificial Societies and Social Simulation*, 8(4):6, 2005.
22. P.H. Chang, K. Chen, Y. Chien, E. Kao, and V. Soo. From reality to mind: A cognitive middle layer of environment concepts for believable agents. In Weyns et al. [WPM05a], pages 57–73.
23. P.R. Cohen, M.L. Greenberg, D.M. Hart, and A.E. Howe. Trial by fire: understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):32–48, 1989.
24. R. Conte, N. Gilbert, and J.S. Sichman. MAS and social simulation: A suitable commitment. In J.S. Sichman, R. Conte, and N. Gilbert, editors, *Multi-Agent Systems and Agent-Based Simulation*, volume 1544 of *LNCS*, pages 1–9. Springer-Verlag, 1998.
25. R. Conte, R. Hegselmann, and P. Terna, editors. *Simulating Social Phenomena*, volume 456 of *Lecture Notes in Economics and Mathematical Systems*, Berlin, 1997. Springer-Verlag.
26. R.J. Collins and D.R. Jefferson. Antfarm: Towards simulated evolution. In C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 579–601. Addison-Wesley, Redwood City, CA, 1992.
27. J. Chapelle, O. Simonin, and J. Ferber. How situated agents can learn to cooperate by monitoring their neighbors' satisfaction. In *Proceedings of the* 15^{th} *European Conference on Artificial Intelligence ECAI'2002*, July 21-26 2002.
28. G. Deffuant, F. Amblard, G. Weisbuch, and T. Faure. How can extremism prevail? a study based on the relative agreement interaction model. *The Journal of Artificial Societies and Social Simulation, JASSS*, 5(4), January 2002.
29. K.S. Decker. Distributed artificial intelligence testbeds. In G.M.P. O'Hare and Nick.R. Jennings, editors, *Foundations of distributed artificial intelligence*, chapter 3, pages 119–138. John Wiley & Sons, Inc., 1996.
30. Y. Demazeau. Steps towards multi-agent oriented programming. slides Workshop, 1st International Workshop on Multi-Agent Systems, IWMAS '97, October 1997.
31. A. Drogoul and J. Ferber. Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In C. Castelfranchi and E. Werner, editors, *Artificial Social Systems*, volume 830 of *Lecture Notes in Computer Science*, pages 3–23. Springer-Verlag, 1992.
32. M. Dorigo and L.M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

33. B. Dumont and D.R.C. Hill. Multi-agent simulation of group foraging in sheep: effects of spatial memory, conspecific attraction and plot size. *Ecological Modelling*, 141:201–215, July 1 2001.
34. M. Dorigo. *Optimization, learning and natural algorithms*. PhD thesis, Politecnico di Milano, Italy (in Italian), 1992.
35. M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
36. K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of The Third International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS'2004)*, volume 2, pages 530–537. ACM Press, July 19-23 2004.
37. N. David, J.S. Sichman, and H. Coelho. Towards an emergence-driven software process for agent-based simulation. In c, editor, *Multi-Agent-Based Simulation II, Proceedings of MABS 2002, Third International Worshop*, volume 2581 of *LNAI*, pages 89–104. Springer-Verlag 2003, July 2002.
38. J. Dàvila and K. Tucci. Towards a logic-based, multi-agent simulation theory. In *International Conference on Modeling, Simulation and Neural Networks MSNN'2000*, October 22-24 2000.
39. A. Drogoul, D. Vanbergue, and T. Meurisse. Multi-agent based simulation: Where are the agents ? In c, editor, *Multi-Agent-Based Simulation II, Proceedings of MABS 2002, Third International Worshop*, volume 2581 of *LNAI*, pages 89–104. Springer-Verlag, July 2002.
40. E.H. Durfee, M. Yokoo, M.N. Huhns, and O. Shehory, editors. *Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*, New York, NY, USA, 2007. ACM.
41. J.M. Epstein and R.L. Axtell. *Growing Artificial Societies*. Brookings Institution Press, Washington D.C., 1996.
42. B. Edmonds and D. Hales. Replication, replication and replication: Some hard lessons from model alignment. *The Journal of Artificial Societies and Social Simulation, JASSS*, 6(4), October 2003.
43. L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy. The hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12(2):213–253, 1980.
44. J.M. Epstein. Agent-based computational models and generative social science. In J.M. Epstein, editor, *Generative Social Science Studies in Agent-Based Computational Modeling*, Princeton Studies in Complexity, pages 4–46. Princeton University Press, January 2007.
45. J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., 1999.
46. T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an agent communication language. In *CIKM '94: Proceedings of the third international conference on Information and knowledge management*, pages 456–463, New York, NY, USA, 1994. ACM.
47. P.A. Fishwick. Computer simulation: growth through extension. *Transactions of the Society for Computer Simulation International*, 14(1):13–23, 1997.
48. J. Ferber and J.-P. Müller. Influences and reaction : a model of situated multi-agent systems. In M. Tokoro, editor, *Proceedings of the 2nd International Conference on Multi-agent Systems (ICMAS-96)*, pages 72–79. The AAAI Press, December 10-13 1996.

49. E. Fyanio, J.-P. Treuil, E. Perrier, and Y. Demazeau. Multi-agent architecture integrating heterogeneous models of dynamical processes : the representation of time. In J.S. Schiman, R. Conte, and N. Gilbert, editors, *Multi-Agent Systems and Agent-Based Simulation*, volume 1534 of *LNCS*, pages 226–236. Springer-Verlag, 1998.
50. G.F. Gause. *The Struggle for Existence*. Williams and Wilkins, Baltimore, 1934.
51. L. Gasser, C. Braganza, and N. Herman. Mace: A flexible testbed for distributed ai research. *Distributed Artificial Intelligence*, pages 119–152, 1987.
52. O. Gutknecht, J. Ferber, and F. Michel. Integrating tools and infrastructures for generic multi-agent systems. In *Proceedings of the fifth international conference on Autonomous agents, AA 2001*, pages 441–448. ACM Press, 2001.
53. P. Guyot and S. Honiden. Agent-based participatory simulations: Merging multi-agent systems and role-playing games. *JASSS, The Journal of Artificial Societies and Social Simulation*, 9(4), 2006.
54. N. Gilbert. Agent-based social simulation: Dealing with complexity. http://www.soc.surrey.ac.uk/staff/ngilbert/ngpub/paper165_NG.pdf. Accessed June 2008, 2005.
55. L. Gasser and K. Kakugawa. MACE3J: fast flexible distributed simulation of large, large-grain multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 745–752. ACM Press, 2002.
56. M.R. Genesereth and N.J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., 1987.
57. C. Goldspink. Methodological implications of complex systems approaches to sociality: Simulation as a foundation for knowledge. *The Journal of Artificial Societies and Social Simulation, JASSS*, 5(1), January 2002.
58. V. Grimm and S.F. Railsback. *Individual-based Modeling and Ecology (Princeton Series in Theoretical and Computational Biology)*. Princeton Series in Theoretical and Computational Biology. Princeton University Press, July 2005.
59. N. Gilbert and K.G. Troitzsch. *Simulation for the Social Scientist*. Open University Press, 2nd edition, February 2005.
60. Z. Guessoum. A multi-agent simulation framework. *Transactions of the Society for Computer Simulation International*, 17(1):2–11, 2000.
61. A. Harding. *Microsimulation and Public Policy*. North Holland Elsevier, Amsterdam, 1996.
62. B.A. Huberman and N.S. Glance. Evolutionary games and computer simulations. *Proceedings of the National Academy of Science USA*, 90(16):7716–7718, August 1993.
63. A. Helleboogh, T. Holvoet, and D. Weyns. The role of the environment in simulating multi-agent systems. In Uhrmacher and Weyns [UW08], chapter 6.
64. P.T. Hraber, T. Jones, and S. Forrest. The ecology of Echo. *Artificial Life*, 3(3):165–190, 1997.
65. D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, May 1995.
66. S. Hanks, M.E. Pollack, and P.R. Cohen. Benchmarks, test beds, controlled experimentation, and the design of agent architectures. *Artificial Intelligence Magazine*, 14(4):17–42, 1993.
67. J. Himmelspach and M. Röhl. JAMES II - experiences and interpretations. In Uhrmacher and Weyns [UW08], chapter 15.
68. J. Himmelspach, M. Röhl, and A.M. Uhrmacher. Simulation for testing software agents -

an exploration based on James. In S. Chick, P.J. Sánchez, D. Ferrin, and D.J. Morrice, editors, *WSC'03: Proceedings of the 35th Winter Simulation Conference*, volume 1, pages 799–807, Piscataway, NJ, USA, 2003. IEEE press.

69. A. Helleboogh, G. Vizzari, A. Uhrmacher, and F. Michel. Modeling dynamic environments in multi-agent simulation. *Autonomous Agents and Multi-Agent Systems*, 14(1):87–116, 2007.
70. T. Ishida, Y. Nakajima, Y. Murakami, and H. Nakanishi. Augmented experiment: Participatory design with multiagent simulation. In M.M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1341–1346, 2007.
71. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In W.L. Johnson, editor, *AGENTS '97: Proceedings of the first international conference on Autonomous Agents*, pages 340–347, New York, NY, USA, 1997. ACM Press.
72. F. Klügl. SeSAm: Visual programming and participatory simulation for agent-based models. In Uhrmacher and Weyns [UW08], chapter 16.
73. F. Klügl, C. Oechslein, F. Puppe, and A. Dornhaus. Multi-agent modelling in comparison to standard modelling. In F.J. Barros and N. Giambasi, editors, *Proceedings of AIS 2002: Artificial Intelligence, Simulation and Planning in High Autonomous Systems*, pages 105–110, San Diego, CA, April 7-10 2002. SCS Publishing House.
74. M. Luck and R. Aylett. Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, 14(1):3–32, 2000.
75. V.R. Lesser and D.D. Corkill. Functionally accurate cooperative distributed systems. *IEEE transactions on systems, machines and cybernetics*, SMC-11(1):81–96, 1981.
76. V.R. Lesser and D.D. Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, 1983.
77. B.G. Lawson and S. Park. Asynchronous time evolution in an artificial society mode. *The Journal of Artificial Societies and Social Simulation, JASSS*, 3(1), January 2000.
78. F. Michel, G. Beurier, and J. Ferber. The TurtleKit simulation platform: Application to complex systems. In A. Akono, E. Tonyé, A. Dipanda, and K. Yétongnon, editors, *Workshops Sessions, First International Conference on Signal & Image Technology and Internet-Based Systems SITIS' 05*, pages 122–128. IEEE, november 2005.
79. N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The Swarm simulation system: A toolkit for building multi-agent simulations. Technical Report 96-06-042, Santa Fe Institute, Santa Fe, NM, USA, 1996.
80. M.B. Marietto, N. David, J.S. Sichman, and H. Coelho. A classification of paradigmatic models for agent-based social simulation. In D. Hales, B. Edmonds, E. Norling, and J. Rouchier, editors, *Multi-Agent-Based Simulation III*, volume 2927 of *LNCS*, pages 193–208. Springer-Verlag, February 2004.
81. F. Michel, A. Gouaïch, and J. Ferber. Weak interaction and strong interaction in agent based simulations. In D. Hales, B. Edmonds, E. Norling, and J. Rouchier, editors, *Multi-Agent-Based Simulation III*, volume 2927 of *LNCS*, pages 43–56. Springer-Verlag, February 2004.
82. S. Mazouzi, Z. Guessoum, F. Michel, and M. Batouche. A multi-agent approach for range image segmentation with bayesian edge regularization. In J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems, 9th International Conference, ACIVS 2007, Delft, The*

Netherlands, August 28-31, 2007, pages 449–460, 2007.

83. S. Moss, H. Gaylard, S. Wallis, and B. Edmonds. SDML: A multi-agent language for organizational modelling. *Computational & Mathematical Organization Theory*, 4(1):43–69, 1998.
84. F. Michel. The IRM4S model: the influence/reaction principle for multiagent based simulation. In Durfee et al. [DYHS07], pages 903–905.
85. J.P. Müller and M. Pischel. The agent architecture inteRRaP: Concept and application. Technical Report RR 93-26, German Research Center for Artificial Intelligence, 1993.
86. U. Merlone, M. Sonnessa, and P. Terna. Horizontal and vertical multiple implementations in a model of industrial districts. *Journal of Artificial Societies and Social Simulation*, 11(2):5, 2008.
87. S.R. Musse and D. Thalmann. From one virtual actor to virtual crowds: requirements and constraints. In *AGENTS'00: Proceedings of the fourth international conference on Autonomous Agents*, pages 52–53, New York, NY, USA, 2000. ACM press.
88. M.J. North, N.T. Collier, and J.R. Vos. Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16(1):1–25, January 2006.
89. M. Nguyen-Duc and A. Drogoul. Using computational agents to design participatory social simulations. *JASSS, The Journal of Artificial Societies and Social Simulation*, 10(4/5), 2007.
90. D.J. Nowak, I. Price, and G.B. Lamont. Self organized UAV swarm planning optimization for search and destroy using swarmfare simulation. In *WSC '07: Proceedings of the 39th Winter Simulation Conference*, pages 1315–1323, Piscataway, NJ, USA, 2007. IEEE Press.
91. J. Odell, H.V.D. Parunak, M. Fleischer, and S. Breuckner. Modeling agents and their environment. In F. Giunchiglia, J. Odell, and G. Weiss, editors, *Agent-Oriented Software Engineering (AOSE) III*, volume 2585 of *Lecture Notes on Computer Science*, pages 16–31, Berlin, 2002. Springer-Verlag.
92. Guy.H. Orcutt. A new type of socio-economic systems. *The Review of Economics and Statistics*, 58:773–797, 1957.
93. N. Pelechano, J.M. Allbeck, and N.I. Badler. Controlling individual agents in high-density crowd simulation. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108, Aire-la-Ville, Switzerland, 2007. Eurographics Association.
94. H.V.D. Parunak. "go to the ant": Engineering principles from natural multi-agent systems. *Annals of Operations Research*, 75(0):69–101, January 1997.
95. M.T. Parker. What is ascape and why should you care? *The Journal of Artificial Societies and Social Simulation, JASSS*, 4(1), January 2001.
96. H.V.D. Parunak and S.A. Brueckner. Polyagents: Simulation for supporting agents' decision making. In Uhrmacher and Weyns [UW08], chapter 4.
97. D. Phan. From agent-based computational economics towards cognitive economics. In P. Bourgine and J.-P. Nadal, editors, *Cognitive Economics*, pages 371–395. Springer Verlag, 2004.
98. H.V.D. Parunak, R. Savit, and R.L. Riolo. Agent-based modeling vs. equation-based modeling: A case study and users' guide. In J.S. Schiman, R. Conte, and N. Gilbert, editors, *Proceedings of the* 1^{st} *Workshop on Modelling Agent Based Systems, MABS'98*, volume 1534 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, July 4-6 1998.

99. W.T. Reeves. Particle systems – a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, 1983.
100. M. Resnick. *Turtles, termites, and traffic jams: explorations in massively parallel microworlds*. MIT Press, Cambridge, MA, USA, 1994.
101. C.W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1987. ACM.
102. A.S. Rao and M.P. Georgeff. An abstract architecture for rational agents. In B. Nebel, C. Rich, and W.R. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 439–449. M. Kaufmann Publishers, October 25-29 1992.
103. S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition, 2003.
104. M.L. Rosenzweig. Paradox of enrichment: Destabilization of exploitation ecosystems in ecological time. *Science*, 171(3969):385–387, January 1971.
105. J. Rouchier. Re-implementation of a multi-agent model aimed at sustaining experimental economic research: The case of simulations with emerging speculation. *The Journal of Artificial Societies and Social Simulation, JASSS*, 6(4), October 2003.
106. P. Riley and G. Riley. SPADES — a distributed agent simulation environment with software-in-the-loop execution. In S. Chick, P.J. Sánchez, D. Ferrin, and D.J. Morrice, editors, *WSC'03: Proceedings of the 35th Winter Simulation Conference*, volume 1, pages 817–825, Piscataway, NJ, USA, 2003. IEEE press.
107. T.S. Schelling. Dynamic models of segregation. *Journal of Mathematical Sociology*, 1:143–186, 1971.
108. J.R. Searle. *Speech Acts*. Cambridge University Press, 1969.
109. O. Simonin and J. Ferber. Modeling Self Satisfaction and Altruism to handle Action Selection and Reactive Cooperation. In J.A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S.W. Wilson, editors, *From Animals to Animats 6, the sixth International Conference on Simulation of Adaptive Behavior SAB'00*, pages 314–323, Cambridge, Massachussets, USA, 11-16 septembre 2000. MIT Press.
110. R.E. Shannon. *Systems Simulation: the art and science*. Prentice-Hall, 1975.
111. K. Sims. Evolving 3d morphology and behavior by competition. *Artificial Life*, 1(4):353–372, 1994.
112. R.G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C29(12):1104–1113, 1980.
113. B. Schattenberg and A.M. Uhrmacher. Planning agents in james. *Proceedings of the IEEE*, 89(2):158–173, February 2001.
114. K.G. Troitzsch. Validating simulation models. In H. Graham, editor, *18th European Simulation Multiconference. Networked Simulations and Simulation Networks*, pages 265–270. The Society for Modeling and Simulation International, SCS Publishing House, 2004.
115. K.G. Troitzsch. Multi-agent systems and simulation: a survey from an application perspective. In Uhrmacher and Weyns [UW08], chapter 2.
116. T. Tyrrell. The use of hierarchies for action selection. *Adaptive Behavior*, 1(4):387–420, 1993.
117. A.M. Uhrmacher. Dynamic structures in modeling and simulation: a reflective approach. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 11(2):206–232, 2001.

118. A.M. Uhrmacher and D. Weyns, editors. *Agents, Simulation and Applications*. Taylor and Francis, 2008.
119. R. Vincent, B. Horling, and V.R. Lesser. An agent infrastructure to build and evaluate multi-agent systems: The java agent framework and multi-agent system simulator. *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, 1887:102–127, January 2001.
120. V. Volterra. *Variations and fluctuations of the number of individuals in animal species living together*. McGraw-Hill, 1926.
121. P. Valckenaers, J.A. Sauter, C. Sierra, and J.A. Rodríguez-Aguilar. Applications and environments for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):61–85, 2007.
122. D. Vanbergue, J.-P. Treuil, and A. Drogoul. Modelling urban phenomena with cellular automata. *Advances in Complex Systems*, 3, 1-4:127–140, 2000.
123. D. Weyns and T. Holvoet. Formal model for situated multi-agent systems. *Fundamenta Informaticae*, 63:1–34, 2004.
124. M. Woolridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2002.
125. D. Weyns, H.V.D. Parunak, and F. Michel, editors. *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, volume 3374 of *LNAI*. Springer, 2005.
126. D. Weyns, H.V.D. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems: State-of-the-art and research challenges. In Weyns et al. [WPM05a], pages 1–47.
127. B.P. Zeigler. Toward a formal theory of modeling and simulation: Structure preserving morphisms. *Journal of the ACM (JACM)*, 19(4):742–764, 1972.
128. K. Zeghal and J. Ferber. Craash: A coordinated collision avoidance system. In *European Simulation Multiconference*, 1993.
129. B.P. Zeigler, T.G. Kim, and H. Praehofer. *Theory of Modeling and Simulation*. Academic Press, Inc., 2nd edition, 2000.
130. F. Zambonelli and H.V.D. Parunak. From design to intention: signs of a revolution. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 455–456. ACM Press, 2002.