The IRM4S model: the influence/reaction principle for multiagent based simulation

Fabien Michel

The IRM4S Model: The Influence/Reaction Principle for Multi-Agent Based Simulation

Fabien Michel
CReSTIC/LERI, Universite de Reims, France
Rue des crayeres BP 1035, 51 687 Reims CEDEX 2
fabien.michel@univ-reims.fr

ABSTRACT

The IRM4S model (Influence Reaction Model for Simulation) is an adaptation of the formalism of [2] for multi-agent based simulations (MABS). The goal of IRM4S is to provide a framework that eases the use of the Influence/Reaction principle within MABS.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems—*Multi-Agent Based Simulation*

General Terms

Theory, Design, Experimentation

Keywords

Influence/Reaction, multi-agent based simulation

1. INTRODUCTION

Within MABS, the result of agents actions and interactions are usually modeled as a *transformation of a global state*. For instance, in [3], wherein Σ is the world state, the behavior of an agent a is modeled using $Behavior_a : \Sigma \mapsto \Sigma$ which is decomposed in three functions : (1) $Percept_a$: $\Sigma \mapsto P_a$ computes a percept, (2) $Mem_a : P_a \times S_a \mapsto S_a$ computes the new internal state s_a, and (3) $Decision_a$: $P_a \times S_a \mapsto \Sigma$ modifies the world according to the action of a. So, the **direct modification** of the environment **is** the means by which is computed the **deliberation result** (e.g. $\sigma = \{door(closed)\} \mapsto \sigma = \{door(open)\}$).

However, such an action modeling raises a number of problems at both the modeling and implementation levels.

Firstly, it does not allow to easily model simultaneous actions since the actions are validated one by one [2].

Secondly, by directly modifying the environment, it is not possible to model the uncertainty related to external dynamics. Indeed, it is not because an agent decides to do an action that the expected result will happen (e.g. the action means of a robot may be out of order). More generally, an agent cannot compute the result of its actions: It does not know exhaustively the environmental settings, especially the other actions taking place at the same time. This can be summarized as the *environmental integrity constraint*: An agent should not be able to directly modify its environment.

Thirdly, there is an issue related to agents autonomy. Indeed, agents interactions, and especially joint actions which conceptually involve independent decisions, are usually implemented so that only one agent, the initiator, does decide for the others, modifying the targeted agents and the environment at once (e.g. in a reproduction process [5]). Thus, the targeted agents goals are not considered. Is it normal that a targeted agent find itself involved in an interaction process not explicitly chosen? Is this entity autonomous? To be correctly modeled, some interactions require to consider all the involved behaviors, not only one [5].

Finally, in a modeling and simulation context, it is crucial that the results obtained from the modeling specifications do not depend on the implementation [7]. However, MABS models that rely on a classical action representation are extremely sensitive to the way they are implemented [4].

Ferber and Müller have proposed an original action model (noted FM in this paper) which is a solution to the simultaneous actions modeling issue [2]. This paper shows that it also represents a solution to the other preceding issues and presents an adaption of FM for MABS, namely the IRM4S model. The next sections describe our proposition and give examples of IRM4S use cases.

2. THE IRM4S MODEL

2.1 Influence/Reaction: a MAS action theory

This theory relies on two notions: (1) influences and (2) reaction to influences [2]. In this theory, an agent does not perform actions, in the meaning seen previously, but produces **influences** (e.g. "try to open a door"). The difference is fundamental. Influences do not directly modify the environment and, from an agent's point of view, nothing can be guaranteed about their result. This perspective enables to distinguish the individual gestures (agent level) from what actually happens considering the other gestures: **The environment reaction** to all the influences (multiagent level). So, the reaction cannot be computed without knowing all the influences which are produced at the same time.

Applying this theory at the implementation level thus requires a two phases mechanism that (1) collects the influences, *Influence phase*, and then (2) compute the result of their combination, *Reaction phase*. Based on an extension of [3], the formalism of FM relies on such a mechanism.

However, FM is complex and parts of it remain fuzzy [6, 4]. Moreover, since simulating a system relies on modeling its evolution from an instant t to the next $t + dt$ [7], the lack of a temporal variable in FM does not ease its use in MABS and notably leads to ambiguities on how its specification should be implemented [1, 4]. So, despite the interests of its theory, FM has not been used as it is and the works that try to apply FM are always an adaptation of it [1, 6].

2.2 The IRM4S equations

2.2.1 A two phases mechanism

Contrary to the previous cited adaptations, IRM4S does not formalize all the mechanisms which could be derived from the Influence/Reaction principle. The IRM4S purpose is to clarify the two phases mechanism underlying the Influence/Reaction theory, especially from a temporal point of view.

As in FM, we use the notion of *dynamical state* $\delta \in \Delta$ to represent the system state. δ is a 2-tuple $< \sigma, \gamma >$ where $\sigma \in \Sigma$ represents the environmental variables, and $\gamma \in \Gamma$ the influences. The system evolution is defined using a function *Evolution* such that $\delta(t + dt) = Evolution(< \sigma(t), \gamma(t) >)$. Thus, to apply the Influence/Reaction principle, we decompose *Evolution* in two functions, *Influence* and *Reaction*, that define the required two phases mechanism: *Influence* : $\Sigma \times \Gamma \mapsto \Gamma'$ then *Reaction* : $\Sigma \times \Gamma' \mapsto \Sigma \times \Gamma$.

Influence globally defines the influences $\gamma'(t) \in \Gamma'$ produced at the micro level (agents and environment). We note this set $\gamma'(t)$ (not $\gamma(t)$) to express that it is a temporary set which will be used immediately in the *Reaction*. *Reaction* defines how the world changes, considering $\sigma(t)$ and $\gamma'(t)$. So *Evolution* corresponds to a two phases mechanism:

$$\gamma'(t) = Influence(\sigma(t), \gamma(t)) \quad (1)$$

$$< \sigma(t + dt), \gamma(t + dt) >= Reaction(\sigma(t), \gamma'(t)) \quad (2)$$

2.2.2 Influence phase: agents and environment

Contrary to [3], the perception of an agent is now computed from a dynamical state Δ, and not only from Σ. This allows to model perceptions which express the dynamics of a situation (e.g. the fact that a ball is rolling). Such a perception is tricky to compute from Σ variables, while it is easy to model using influences. Moreover, the behavior of an agent now produces an influence $\gamma \in \Gamma'$. Thus, we have now $Behavior_a : \Sigma \times \Gamma \mapsto \Gamma'$, which can be decomposed in:

$$\begin{aligned} p_a(t) &= Perception_a(\sigma(t), \gamma(t)) && (3) \\ s_a(t + dt) &= Memorization_a(p_a(t), s_a(t)) && (4) \\ \gamma'_a(t) &= Decision_a(p_a(t), s_a(t + dt)) && (5) \end{aligned}$$

The environment has endogenous dynamics and thus also produces influences (moving objects, pheromone diffusion, etc.). Contrary to approaches that integrate these dynamics into the reaction computation, these influences belong to the *Influence* phase in IRM4S. Indeed, from a temporal point of view, all the influences are simultaneous: The environment and the agents are not temporarily independent (an agent and a rolling ball produce influences simultaneously). Besides, as for the behavior of an agent, the environment dynamics is also a consequence of the current system dynamical state. We thus represent the environmental dynamics with $Natural_w : \Sigma \times \Gamma \mapsto \Gamma$, similarly to $Behavior_a$. However, $Natural_w$ does not represent any autonomous behavior at all. So, we have $\gamma'_w(t) = Natural_w(\sigma(t), \gamma(t))$.

Influence thus gives a set $\gamma'(t)$ that contains the influences which already are in the system and those which are produced by the environment and the agents:

$$\gamma'(t) = Influence(\sigma(t), \gamma(t)) = \{\gamma(t) \cup \gamma'_w(t) \bigcup_a \gamma'_a(t)\} \quad (6)$$

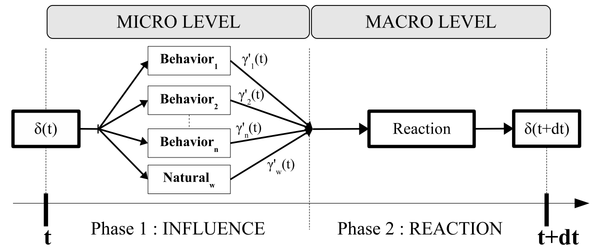

Figure 1: System evolution in the IRM4S model.

2.2.3 Reaction phase: macro level

In the previous section, we have decomposed the *Influence* phase to show how the micro level could be managed. Considering the *Reaction* phase, the same must not be done. Indeed, MABS models could be related to very different domains and the influences could be very heterogeneous: movements, speech acts, reproduction processes, etc. So, it would be not relevant to propose a unique solution to the reaction computation. It is up to the modeler to define the desired system dynamics.

We can however notice that this computation raises an issue: The complexity of combining all the influences. To reduce this complexity, two main solutions can be considered: (1) distribute the reaction computation and (2) classify the influences. Firstly, it is indeed possible to take into account that, a priori, an agent only influences its surroundings (e.g. a movement does not need to be considered at a global level). Secondly, it is interesting to classify the influences according to their type. An agent that wants to move will not have, a priori, any impact on a neighbor that consumes a resource.

2.2.4 Schematic representation of IRM4S

IRM4S defines a two phases mechanism: *Influence* **then** *Reaction*. So, our temporal vision of the Influence/Reaction cycle makes a clear distinction between (1) the influences and (2) the reaction to these influences (figure 1).

3. MODELING EXAMPLES

3.1 Soccer robots

Let be the following scenario. (1) two robots (Bot_1 and Bot_2) are in position so they can shoot a ball. (2) Each robot decides to shoot the ball (Influence phase). (3) The shoots are combined and make the ball moves (Reaction phase). (4) The movement of the ball then produces an environmental influence that models the frictions to which the ball is subjected (Influence phase). (5) The location of the ball has been changed and its speed reduced (Reaction phase). This scenario could be implemented as follows:

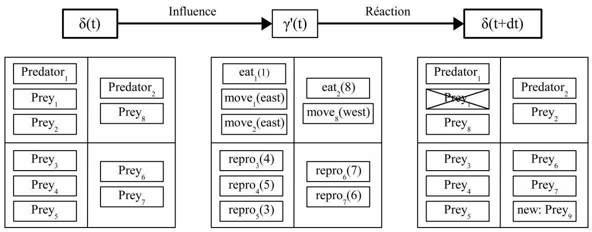

Figure 2: Influence then Reaction.

1 $< \sigma(0) = \{Bot_1, Bot_2, Ball\}, \gamma(0) = \{\} >$

2 $\gamma'(0) = \{shoot_1(1,1), shoot_2(-1,1)\} : (force\ vectors)$

3 $< \sigma(1) = \{Bot_1, Bot_2, Ball\}, \gamma(1) = \{move_{ball}(0,2)\} >$

4 $\gamma'(1) = \{move_{ball}(0,2), slow_{ball}(0,-0.5)\}$

5 $< \sigma(2) = \{Bot_1, Bot_2, Ball\}, \gamma(2) = \{move_{ball}(0,1.5)\} >$

This simple example highlights several IRM4S aspects. First of all, the modeling of the simultaneity is eased thanks to the distinction between the agent and multiagent levels. Indeed, thanks to the recovery of the influences (2), we have all the information which enables the computation of the shoots combination (3). Secondly, this example also illustrates how the endogenous dynamics of the environment could be modeled (4). Finally, it also shows the possibility for the agents to simply perceive the fact that the ball is rolling because it is modeled as an influence (3). As previously said, such a perception could be very tricky to model using classical approaches.

3.2 A prey/predator system

Let be a system, composed of preys and predators localized on a 2D grid: $\Sigma = \{Prey_1(x,y), ..., Pred_n(x,y)\}$. Every agent can perceive the entities which are at the same place: $P_{prey} = P_{pred} = \{Prey_1, ..., Pred_n\}$. The predators can move and eat preys. The preys can move and reproduce with each other. So, we have the following influences: $\Gamma_{predator} = \{move_i(direction), eat_i(Prey_{id})\}$ and $\Gamma_{prey} = \{move_i(direction), repro_i(Prey_{id})\}$. Here are some examples of behaviors which could be obtained:

$$Behavior_{prey_2}(Prey_1, Prey_4) = repro_2(Prey_1)$$
$$Behavior_{pred_2}(Prey_2, Prey_3) = eat_2(Prey_3)$$

Here is an example of how the reaction could be done:

`reactionComputation(){`
`(1) For each influence` $eat_i(Prey_{id})$
$eat_i(Prey_{id})$ `is validated according to a coin flip probability. In case of competition (that is` $\{eat_i(Prey_k), eat_j(Prey_k)\} \in \gamma'$`) the predator with the highest energy is selected`
`(2) Suppression of the agents killed in (1)`
`(3) Considering influences` $repro_i(Prey_j)$
`If` $\{repro_i(Prey_j), repro_j(Prey_i)\} \in \gamma'$
`Then a new prey is created`
`(4) Validate the influences` $move_i(direction)$ `}`

Figure 2 is an example of the dynamic which could be thus obtained. This example illustrates all the interests of the notion of influence with respect to the modeling of the reproduction interaction. Indeed, contrary to the classical approaches, all the agents influences are taken into account and there is no violation of the autonomy property: We do not decide in place of the agents, but we do explicitly decide of the dynamics of the system. Besides, even if the computation of the reaction which has been proposed is only a solution among others (the scheduling of the reaction could be modified), it is however the expression of a dynamic which is entirely controlled and which cannot be different whatever the used implementation. This is a fundamental aspect of IRM4S: Whatever the order in which the agents produce their influences, the reaction completely specifies the dynamic of the system.

Moreover, the IRM4S model is not related to a particular simulation technique (event-based of discrete time simulation). Indeed, the produced influences may not be immediately consumed within the reaction and can persist during the time of the behavior. So, in an event-based simulation, two reproduction influences can overlap and thus succeed.

4. CONCLUSIONS

This paper has presented the IRM4S model which eases the use of the Influence/Reaction principle within MABS thanks to the use of an explicit temporal variable that clarifies the two phases mechanism embedded in this principle.

Moreover, IRM4S is also a solution to the other issues related to the classical action representation. Indeed, thanks to the notion of influence, IRM4S fully integrates the environmental integrity constraint: The agents do not directly modify the environment. Consequently, the agents cannot violate the autonomy property of the others as well. Besides, autonomous behaviors are always taken into account through their influences. Finally, IRM4S provides an efficient means to design modeling specifications which are fully independent from the way they are implemented.

5. REFERENCES

[1] J. Dávila 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.

[2] J. Ferber and J.-P. Müller. Influences and reaction : a model of situated multi-agent systems. In M. Tokoro, editor, *2^{nd} Int. Conference on Multi-agent Systems ICMAS-96*, pages 72–79. AAAI Press, Dec 1996.

[3] M. R. Genesereth and N. J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., 1987.

[4] F. Michel. Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents. Thesis at Université Montpellier II, 2004.

[5] 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 *LNAI*, pages 43–56. Springer, February 2004.

[6] D. Weyns and T. Holvoet. Formal model for situated multi-agent systems. *Fundamenta Informaticae*, 63:1–34, 2004.

[7] B. P. Zeigler, T. G. Kim, and H. Praehofer. *Theory of Modeling and Simulation*. Academic Press, Inc., 2000.