Multi-level agent-based modeling with the Influence Reaction principle

Gildas Morvan, Daniel Jolly

Univ. Lille Nord de France
1bis rue Georges Lefèvre 59044 Lille cedex, France

LGI2A, U. Artois, Technoparc Futura 62400 Béthune, France.

email: first name.surname@univ-artois.fr

Abstract

This paper deals with the specification and the implementation of multi-level agent-based models, using a formal model, IRM4MLS (an Influence Reaction Model for Multi-Level Simulation), based on the Influence Reaction principle. Proposed examples illustrate forms of top-down control in (multi-level) multi-agent based-simulations.

Keywords: multi-level simulation, influence reaction model, micro-macro link, cellular automata.

1 Introduction

Four main *theoretical* issues emerge in the literature on multi-level[1] agent-based modeling: the conception of a meta-model allowing a non ambiguous characterization of a multi-level model at the conceptual level [18, 23, 29, 33], the introduction of a dynamic level of detail [25, 31], the detection and reification of emergent phenomena [2, 3, 5, 21, 30, 34] and the representation of aggregated entities [27]. This paper focuses on the first one, with respect to the Influence Reaction (IR) principle, shortly, action as a two step process: (1) agents produce "influences", *i.e.*, individual decisions, according to their internal state and

[1] The term multi-scale may also be found. Intuitively, a level and scale are similar concepts that both mean viewpoint. However, this notion should be clarified. In the following we assume that two agents are not at the same *scale* iff they represent processes that have different *spatial and/or temporal extents*. Two agents are not in the same *level* iff they cannot *interact directly*, *i.e.*, with a single interaction function. It should follow from the previous definitions that it exists multi-scale models that are mono-level and conversely.

perceptions (2) the system "reacts", *i.e.*, computes the consequences of influences, according to the state of the world [10]. This model has been extended in several ways, and notably for multi-agent based-simulation (MABS), by adding an explicit representation of time [19]. An IR-based meta-model, IRM4MLS (an Influence Reaction Model for Multi-Level Simulation), and its Java implementation are introduced in the section 2. Using, and then extending a simple example: the Conway's game of life, two toy-models of increasing complexity are presented in the section 3, illustrating forms of top-down control in (multi-level) MABS. Results are discussed in the section 4.

2 IRM4MLS: an Influence Reaction Model for Multi-Level Simulation

In this section, IRM4MLS, an Influence Reaction Model for Multi-Level Simulation, is introduced [23]. It extends IRM4S (an Influence Reaction Model for Simulation) in order to deal with multi-level models [19]. From a technical perspective, levels can be viewed as interacting IRM4S simulations[2].

A multi-level model is defined by a set of levels L and a specification of the relations between levels. Two types of relations are specified: an influence relation (agents in a level l are able to produce influences in a level $l' \neq l$) and a perception relation (agents in a level l are able to perceive the dynamic state of a level $l' \neq l$), represented by directed graphs denoted respectively $< L, E_I >$ and $< L, E_P >$, where E_I and E_P are two sets of edges, *i.e.*, ordered pairs of elements of L. Influence and perception relations in a level are systematic and thus not specified in E_I and E_P (cf. eq. 1 and 2).

The in and out neighborhood in $< L, E_I >$ (respectively $< L, E_P >$) are denoted N_I^- and N_I^+ (resp. N_P^- and N_P^+) and are defined as follows:

$$\forall l \in L, N_I^-(l) \text{ (resp. } N_P^-(l)) = \{l\} \cup \{l' \in L : l'l \in E_I \text{ (resp. } E_P)\}, \quad (1)$$

$$\forall l \in L, N_I^+(l) \text{ (resp. } N_P^+(l)) = \{l\} \cup \{l' \in L : ll' \in E_I \text{ (resp. } E_P)\}, \quad (2)$$

The set of agents in the system at time t is denoted $A(t)$. $\forall l \in L$, the set of agents belonging to l at t is denoted $A_l(t) \subseteq A(t)$. An agent belongs to a level iff a subset of its physical state ϕ_a belongs to the state of the level. Thus, an agent belongs to zero, one, or more levels. An environment models the *natural dynamics* of level properties and can be shared by different levels (fig. 1).

The dynamic state of a level $l \in L$ at time t, denoted $\delta^l(t) \in \Delta^l$, is a tuple $< \sigma^l(t), \gamma^l(t) >$, where $\sigma^l(t) \in \Sigma^l$ and $\gamma^l(t) \in \Gamma^l$ are the sets of environmental properties and influences of l. The behavior of an agent $a \in A_l$ is defined as $Decision_a^l \circ Memorization_a \circ Perception_a^l$, with

$$Perception_a^l : \prod_{l_P \in N_P^+(l)} \Delta^{l_P} \mapsto \prod_{l_P \in N_P^+(l)} P_a^{l_P}, \quad (3)$$

$$Memorization_a : \prod_{l \in L | a \in A_l} \prod_{l_P \in N_P^+(l)} P_a^{l_P} \times S_a \mapsto S_a, \quad (4)$$

[2] Therefore, each level has a microscopic side: the agent behaviors, and a macroscopic side: the reaction function. This aspect can also be found in holonic multi-agent systems [4].

$$Decision_a^l : S_a \mapsto \prod_{l_I \in N_I^+(l)} \Gamma^{l_I}{}'. \tag{5}$$

There is no memorization function specific to a level to preserve the coherence of the internal state of the agents. The environment ω of a level l produces influences through a function:

$$Natural_\omega^l : \Delta^l \mapsto \prod_{l_I \in N_I^+(l)} \Gamma^{l_I}{}'. \tag{6}$$

The reaction function computes next level state and time advance:

$$Reaction^l : \Sigma^l \times \Gamma^{l}{}' \mapsto \Delta^l \times \mathbb{T}^l. \tag{7}$$

The time representation is inspired by DEVS (Discrete EVent System specification) [35]. $\mathbb{T} = \bigcup_{l \in L}\{\mathbb{T}^l\}$ denotes the time vector of the simulation, such as $\forall l \in L, \mathbb{T}^l =< t^l, dt^l >$, where t^l represents when the current event (or step, depending on the simulation model) time and dt^l its lifespan. The final simulation time is denoted t_f. The algorithm 1 ensures the scheduling of these different functions with respect to temporal constraints of perception and memorization, influence production and reaction [23].

The implementation of IRM4MLS is based on the idea of *micro kernel*, taken from MadKit[3] [13]. Thus, in this approach, a *technical* agent, *e.g.*, an observer or a message broker, would be scheduled with respect to the IR principle (cf. algo. 1), the concept of level ensuring a clear separation between system and simulation agents.

The API is minimal (seven *high-level abstractions*) and specifies only the methods needed to schedule a model (fig. 2). Most methods are generic and then are implemented at an abstract level. Basically, to implement a model one only has to override perception, memorization, influence production, reaction and initialization functions.

Agent and behaviors (such as environment and natural dynamics) are represented by different entities to clearly distinguish the *core side* of an agent, its state and memorization function, and the *level sides* (perception and influence production functions) that can change according to simulations.

[3] `http://www.madkit.org`

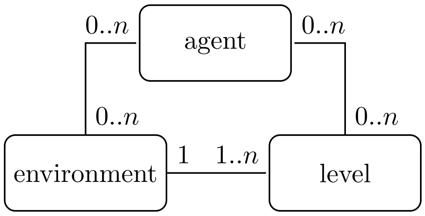


Figure 1: Main concepts of IRM4MLS (as a simplified class diagram)

```
Algorithm 1: simulation model of IRM4MLS
Input: $< L, E_I, E_P >, A, \delta, \mathbb{T}, t_f$
Output: $\delta(t_f)$
while $\exists t^l \leq t_f$ do
    foreach $a \in A(t)$ do
        foreach $l \in L : a \in A_l(t) \wedge \forall l_P \in N_P^+(l), t^l \geq t^{l_P}$ do
            $p_a(t^l) = Perception_a^l(< \delta^{l_P}(t^{l_P}) : l_P \in N_P^+(l) >)$;
        end
        $s_a(t^l + dt^l) = Memorization_a(p_a)$;
    end
    foreach $l \in L : \forall l_I \in N_I^+(l), t^l \leq t^{l_I} \vee t^l + dt^l < t^{l_I} + dt^{l_I}$ do
        foreach $l_I \in N_I^+(l) : t^l \leq t^{l_I} \wedge t^l + dt^l > t^{l_I}$ do
            $\gamma_\omega^{l_I}{}'(t^{l_I}) = Natural_\omega^l(\delta^l(t^l))$ ;
            foreach $a \in A_l(t)$ do
                $\gamma_a^{l_I}{}'(t^{l_I}) = Decision_a^l(s_a(t^l + dt^l))$;
            end
        end
    end
    foreach $l \in L : t^l + dt^l \in min(t + dt)$ do
        $\gamma^{l\prime}(t) = \{\gamma^l(t) \bigcup_{l_I \in N_I^-(l)} \gamma_\omega^{l_I}{}'(t) \bigcup_{a \in A_{l_I}} \gamma_a^{l_I}{}'(t)\}$;
        $<\delta^l(t^l + dt^l), \mathbb{T}^l >= Reaction^l(\sigma^l(t^l), \gamma^{l\prime}(t^l))$;
    end
end
```

3 IRM4MLS in practice: multi-level games of life

3.1 Introducing a macroscopic parameter (top-down control)

In this section, a toy IRM4S (or 1-level IRM4MLS) model, is presented: a modified agent-based version of the Conway's game of life (or simply Life). This simple example illustrates where a macroscopic parameter, should (but should not) be introduced in an agent-based model that relies on the influence reaction principle: in the reaction function (but not in the behavioral functions of agents). Therefore, this parameter has a non-ambiguous semantics that does not depend on the updating scheme of the simulations, even in the case of *strong interaction*[4].

Each agent represents a cell that can be dead or alive and that has eight neighbors

[4] Strong interaction implies that agents agree on the outcome of the interaction [20]. Thus, such model should not be simulated with a STRIPS-like meta-model, *i.e.*, that views an action as a change of the state of the system. It would lead to problems of result replication [1], but also of parameter and result interpretation [8, 20].

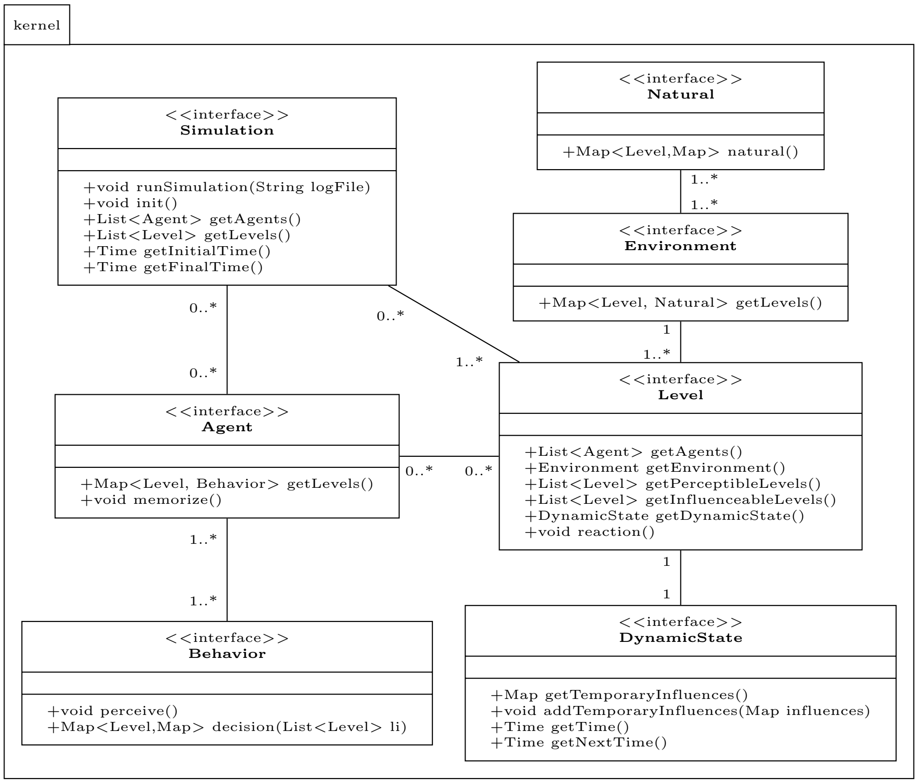


Figure 2: Java API of IRM4MLS

in a toroidal grid. The set of environmental properties is then:

$$\forall t, \sigma(t) = \bigcup_{a \in A} \{neighbors(a), alive(a)\}. \tag{8}$$

Cells evolve in parallel: the reaction function can then simply be defined as "applying agent influences" (algo. 2). If a cell is dead and has three living neighbors or is alive and has two or three living neighbors, it will be alive at the next step; in other cases, it will be dead. Let specify the behavior of the agents:

1. they perceive the number of living cells in their neighborhood (algo. 3),
2. memorize their internal state, *i.e.*, their next state (algo. 4),
3. and then, decide whether or not they will be alive at the next step according to their internal state (algo. 5).

The environment is static: there is no natural dynamics and thus, $Natural_\omega$ returns $\emptyset$.

One cruel and ironic aspect of Life is that a cell has generally little chance to remain alive in the long run (fig. 3(a)). Moreover, what you get most of the time,

Algorithm 2: *Reaction*

```
Input: $\sigma(t), \gamma'(t)$
Output: $\delta(t+1)$
foreach $a \in A(t)$ do
    $alive(a) = \gamma'_a(t)$ ;
end
```

Algorithm 3: *Perception$_a$*

```
Input: $\delta(t)$
Output: $p_a(t)$
$p_a(t) = \sum_{n \in neighbors(a)} alive(n)$ ;
```

Algorithm 4: *Memorization$_a$*

```
Input: $p_a(t), s_a(t)$
Output: $s_a(t+dt)$
if $alive(a) \wedge p_a(t) \in \{2,3\} \vee$
$\neg alive(a) \wedge p_a(t) \in \{3\}$ then
    $s_a(t+dt) = 1$ ;
else
    $s_a(t+dt) = 0$ ;
end
```

Algorithm 5: *Decision$_a$*

```
Input: $s_a(t+dt)$
Output: $\gamma'_a(t)$
$\gamma'_a(t) = s_a(t+dt)$ ;
```

is a board composed of small still lifes and 1-period oscillators. This behavior is predictable knowing the λ parameter of the game of life[5]. λ is a complexity measure of cellular automata introduced by [15] that depends on the number of cell states K, the neighborhood N and the number n of transitions to a quiescent state S_q in the transition function such as

$$\lambda = 1 - \frac{n}{K^N}, \tag{9}$$

with $K = 2$, $N = 9$ and $n = \binom{8}{2} + 2 \cdot \sum_{i=0,1,4-8} \binom{8}{i}$ for Life.

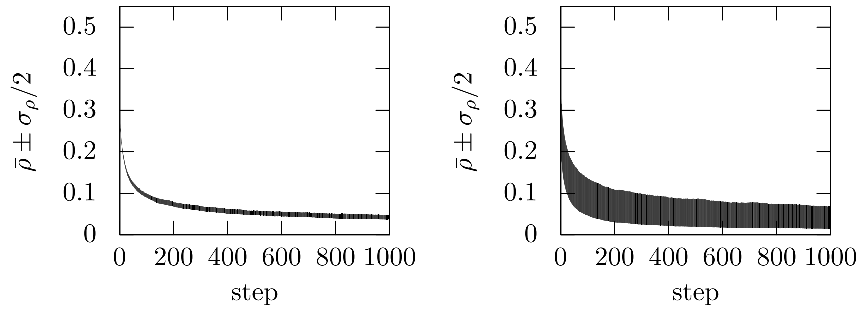
To improve λ_{life} to a value λ^+, one needs to change the rules. In this example, λ^+ is regarded as a macroscopic parameter, explicitly introduced in the model and independent from cell behaviors[6]: influences of dying cells are not taken into account by the reaction function with a probability p such as

$$n = \binom{8}{2} + (2-p) \cdot \sum_{i=0,1,4-8} \binom{8}{i}. \tag{10}$$

Thus, their is a simple linear relation between p and λ^+: $p = (\lambda^+ - \lambda_{life})/0.3359$. For $\lambda^+ \in [0.48, 0.6]$, the number of dying cells tends to decrease in time and large structures of vertical or horizontal rows eventually emerge, shaped by moving groups of switching state cells that seem to work at their boundaries, and eventually vanish when the board becomes a dense still life of density ≈ 0.5 (fig. 4).

[5] $\lambda_{life} = 0.2734375$. For $\lambda \approx 0.25$, *"structures of period 1 appear. Thus, there are now three different possible outcomes for the ultimate dynamics of the system, depending on the initial state. The dynamics may reach a homogeneous fixed point consisting entirely of state S_q, or it may reach a heterogeneous fixed point consisting mostly of cells in state S_q with a sprinkling of cells stuck in one of the other states, or it may settle down to periodic behavior"* [15, p. 17].

[6] An other macroscopic parameter, the asynchrony, has been previously introduced in Life is such way [9].



(a) Macroscopic level

(b) Mesoscopic level

Figure 3: Dynamics of the density of living cells, ρ, starting from a random grid with 100 replications (a) Macroscopic dynamics: expected density and variability of living cells in the whole grid. (b) Mesoscopic dynamics: expected density and variability of living cells in 10×10 cell clusters.

3.2 Top-down feedback control

In the previous example, while the macroscopic parameter λ^+ has an influence on agents, it is not related to the state of the system and therefore, there is no need to observe it. The reaction function can then be viewed as an *open-loop* controller. In the next example, a top-down feedback control is introduced.

The goal of the multi-scale model presented in this section is to keep Life boards at the desired density (ρ^+), by controlling the proportion of dying cells at the mesoscopic level, to account for the *natural* variability of density between regions of the grid (fig. 3(b)). Moreover, the control should affect *as less as possible* simulations at the microscopic level and, to keep it simple, should be tuned by a single linear parameter such as λ^+ in the previous model.

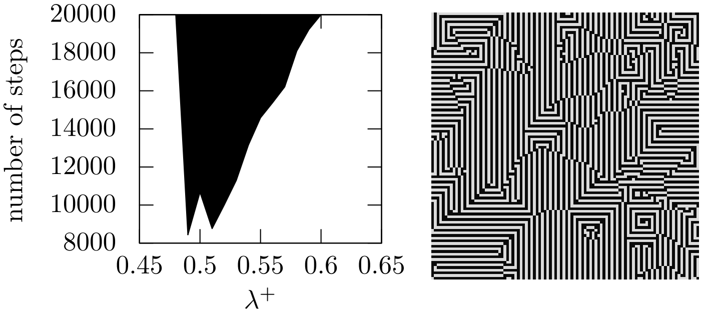
Two levels are considered: the cell (or microscopic) level, l_m and the cell region (or mesoscopic) level, l_M. At the mesoscopic level, the model properties are the expected density and the cells in each region:

$$\forall t, \sigma^{l_M}(t) = \{\rho^+\} \bigcup_{a^{l_M} \in A^{l_M}} \{cells(a^{l_M})\}. \tag{11}$$

The cells behave according to the game of life rules (algo. 3, 4 and 5). However, $Reaction^{l_m}$ depends on mesoscopic influences (algorithm 6). E_P and E_I are equal to $\{l_M l_m\}$. Mesoscopic agents have a proportional control behavior. They

1. perceive the density of living cells in a region,
2. memorize their internal state, *i.e.*, the difference ϵ between expected and actual densities,
3. and then decide the influence sent to agents of l_m:

$$\forall a^{l_M} \in A^{l_M} \forall a^{l_m} \in cells(a^{l_M}), command(a^{l_m}) = k_P \cdot \epsilon. \tag{12}$$



(a) Convergence conditions

(b) Simulation example

Figure 4: (a) Mean number of steps needed to converge to a steady state starting from a random grid (100 replications), simulations are stopped after $2 \cdot 10^4$ steps (simulations converge in the dark area). (b) Example of still life found for $\lambda^+ = 0.5$.

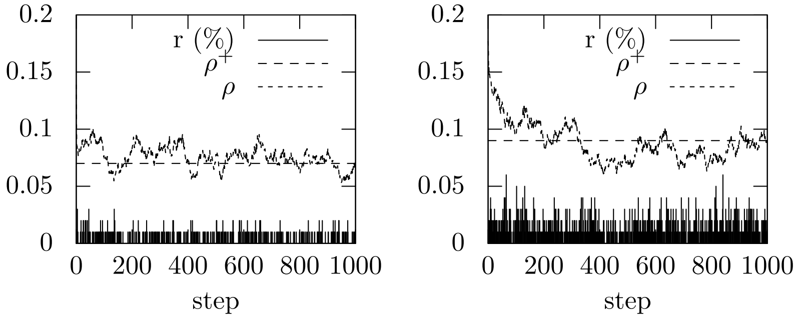
```
Algorithm 6: $Reaction^{l_m}$
Input: $\sigma^{l_m}(t), \gamma^{l_m\prime}(t)$
Output: $\delta^{l_m}(t+1)$
foreach $a^{l_m} \in a^{l_m}$ do
    $rand \in [0,1[$ from pseudorandom uniform distribution ;
    if $command(a^{l_m}) > rand \wedge alive(a^{l_m}) \wedge \neg {\gamma'}_{a^{l_m}}(t^{l_m})$ then
        $alive(a^{l_m}) = \top$ ;
    else
        $alive(a^{l_m}) = {\gamma'}_{a^{l_m}}(t^{l_m})$ ;
    end
end
```

The k_P parameter has to be carefully tuned to run realistic simulations: too small simulations do not achieve the desired solution ($\bar{\rho} = \rho^+$), too big the board density tends to oscillate around ρ^+ and the number of micro influences not taken into account by $Reaction^{l_m}$ becomes too important. However, for appropriate k_P values, this simple linear controller achieves good results and allows to find a good compromise between conflicting micro and meso knowledge (fig. 5).

4 Conclusion

4.1 Discussion

An important issue of multi-level agent-based modeling, only briefly discussed here, is to define the adequate methodology. Indeed, the *traditional* MABS methodology is purely bottom-up: microscopic knowledge is used to construct models while macro-



(a) $\rho^+ = 0.07$

(b) $\rho^+ = 0.09$

Figure 5: Simulation examples with $k_P = 10{\cdot}\rho^+$ and initial density $\rho(t_0) = 2{\cdot}\rho^+$. r represents the rate of microscopic influences not taken into account (in %) by $Reaction^{l_m}$.

scopic knowledge is used to validate models [6]. Thus, it seems irrelevant in a multi-level context. Three general conclusions can be drawn from the previous examples:

- a parameter should be introduced at its observation level in the model; therefore, each observed level should be explicitly represented in the model,
- an intuitive way to model an external control on a level l is to modify the reaction function of l, *i.e.*, to modify the way influences of agents of l are taken into account (but not agent behavioral functions), according to external influences; an external feedback control implies both observation and influence relations: $N_I^-(l) = N_P^-(l)$,
- such a controller can be viewed as a technical tool that aims to find a compromise between conflicting knowledge from the different studied levels and achieve realistic simulations from different studied points of view [16].

Pattern oriented modeling (POM) consists in *"the multi-criteria design, selection and calibration of models of complex systems"* [12]. Many aspects of this methodology, developed in the context of ecology, seem particularly relevant for multi-level agent-based models since "patterns" are generally observed at different levels of organization in complex systems. However, the problem is far from being solved. For instance, the introduction of a dynamic level of detail raises several questions regarding, *e.g.*, the validation of the model or the representation of *composite* agents [25, 27]. A case study of three real world multi-level agent-based models reveals other interesting methodological issues [11].

4.2 Related works

At least three works could be related to this one:

- ML-DEVS is an extension of DEVS that allows the simulation of multi-level models (and not only coupled models in which the behavior of a model is determined by the behaviors of its sub-models) [33]. Two types of relation between levels are defined: information propagation and event activation which are quite similar to those defined in IRM4MLS. However, ML-DEVS supports only pure hierarchies of models, *i.e.*, *interaction graphs* are viewed as *trees* [18]. DEVS, as a generic event-based simulation framework, has also been extended to support agent-based models [24]. A major design difference between IR and DEVS based approaches is the technical orientation of the latter leading to an important gap between conceptual and computational models.

- PADAWAN (Pattern for Accurate Design of Agent Worlds in Agent Nests) is a multi-scale agent-based meta-model based on a compact matricial representation of interactions: IODA (Interaction-Oriented Design of Agent simulations) [14, 29]. Moreover, authors analyze the structure of what is called here interaction graphs in multi-scale models (a relevant issue for IRM4MLS as well), and conclude they should be viewed as *upper semilattices* and not simply *trees* as suggested elsewhere. A major design difference between IR and IODA based approaches is that the latter constraints the definition of interactions, leading to a simple but restrictive simulation framework.
- GAMA[7] is a MABS platform with a dedicated modeling language, GAML, that offers multi-level capabilities [32]. Moreover, it includes a framework (a set of predefined GAML commands) to *agentify* emerging structures [34]. It is certainly the most advanced platform, from an end-user point of view, that integrates a multi-level approach.

4.3 Perspectives

The main perspectives of this work concern the implementation of existing works with IRM4MLS:

- the concept of *PolyAgent* [26, 28],
- multi-level organizational models widely used in engineering sciences such as holonic multi-agent systems (cf. footnote 2), system of systems and heterarchical control [4, 17, 22],
- multi-scale tools: generic scaling operators and emergence detection and reification algorithms [3, 25].

Moreover, the first model presented in this paper could be used to explore the relations between computational capabilities of a cellular automaton (λ^+), noise (a function of $\lambda^+ - \lambda$) and entropy. Moreover, finding the conditions under which cells arrange themselves in a steady state could be an interesting way to solve heuristically large instances of the *maximum density still life problem* [7].

The second model illustrates a form of simple proportional top-down feedback control. Such approach could be generalized to model more complex of cross-level feedback control, using integrations and derivates of observed variables.

References

[1] Bigbee, A., Cioffi-Revilla, C., Luke, S.: Replication of sugarscape using mason. In: Terano, T., Kita, H., Deguchi, H., Kijima, K. (eds.) Agent-Based Approaches

[7] `http://code.google.com/p/gama-platform/`

in Economic and Social Complex Systems IV, Agent-Based Social Systems, vol. 3, pp. 183–190. Springer (2007)

[2] Caillou, P., Gil-Quijano, J.: Simanalyzer : Automated description of groups dynamics in agent-based simulations. In: Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012) (2012)

[3] Chen, C.C., Clack, C., Nagl, S.: Identifying multi-level emergent behaviors in agent-directed simulations using complex event type specifications. Simulation 86(1), 41–51 (2010)

[4] Cossentino, M., Gaud, N., Hilaire, V., Galland, S., Koukam, A.: Aspecs: an agent-oriented software process for engineering complex systems. Autonomous Agents and Multi-Agent Systems 20(2), 260–304 (2010)

[5] David, D., Courdier, R.: See emergence as a metaknowledge. a way to reify emergent phenomena in multiagent simulations? In: Proceedings of ICAART'09. pp. 564–569. Porto, Portugal (2009)

[6] Drogoul, A., Vanbergue, D., Meurisse, T.: Multi-agent based simulation: Where are the agents? In: Sichman, J., Bousquet, F., Davidsson, P. (eds.) Multi-Agent-Based Simulation II, Lecture Notes in Computer Science, vol. 2581, pp. 1–15. Springer (2003)

[7] Elkies, N.: The still-life density problem and its generalizations. Voronoi's Impact on Modern Science, Book I. Proceedings of the Institute of Mathematics of the National Academy of Sciences of Ukraine 21, 228–253 (1998)

[8] Fatès, N., Chevrier, V.: How important are updating schemes in multi-agent systems? an illustration on a multi-turmite model. In: Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010). vol. 1, pp. 533–540 (2010)

[9] Fatès, N.: Does "life" resist asynchrony? In: Adamatzky, A. (ed.) Game of Life Cellular Automata, pp. 257–274. Springer (2010)

[10] Ferber, J., Müller, J.P.: Influences and reaction: a model of situated multiagent systems. In: 2nd International Conference on Multi-agent systems (ICMAS'96). pp. 72–79 (1996)

[11] Gil-Quijano, J., Louail, T., Hutzler, G.: From biological to urban cells: Lessons from three multilevel agent-based models. In: Desai, N., Liu, A., Winikoff, M. (eds.) Principles and Practice of Multi-Agent Systems, Lecture Notes in Computer Science, vol. 7057, pp. 620–635. Springer (2012)

[12] Grimm, V., Railsback, S.: Pattern-oriented modelling: a 'multi-scope' for predictive systems ecology. Philosophical Transactions of the Royal Society B: Biological Sciences 367, 298–310 (2012)

[13] Gutknecht, O., Ferber, J.: The madkit agent platform architecture. In: Agents Workshop on Infrastructure for Multi-Agent Systems. pp. 48–55 (2000)

[14] Kubera, Y., Mathieu, P., Picault, S.: Interaction-oriented agent simulations: From theory to implementation. In: Proc. of the18th European Conf. on Artificial Intelligence (ECAI'08). pp. 383–387 (2008)

[15] Langton, C.: Computation at the edge of chaos: Phase transitions and emergent computation. Physica D: Nonlinear Phenomena 42(1-3), 12–37 (1990)

[16] Li, J., Ge, W., Zhang, J., Kwauk, M.: Multi-scale compromise and multi-level correlation in complex systems. Chemical Engineering Research and Design 83(6), 574 – 582 (2005)

[17] Maier, M.: Architecting principles for system of systems. Systems Engineering 1(4), 267–284 (1998)

[18] Maus, C., John, M., Röhl, M., Uhrmacher, A.: Hierarchical modeling for computational biology. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) Formal Methods for Computational Systems Biology, Lecture Notes in Computer Science, vol. 5016, pp. 81–124. Springer (2008)

[19] Michel, F.: The IRM4S model: the influence/reaction principle for multiagent based simulation. In: Proc. of 6th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2007). pp. 1–3 (2007)

[20] Michel, F., Gouaïch, A., Ferber, J.: Weak interaction and strong interaction in agent based simulations. In: Multi-Agent-Based Simulation III, Lecture Notes in Computer Science, vol. 2927, pp. 43–56. Springer (2003)

[21] Moncion, T., Amar, P., Hutzler, G.: Automatic characterization of emergent phenomena in complex systems. Journal of Biological Physics and Chemistry 10, 16–23 (2010)

[22] Morvan, G., Dupont, D., Soyez, J.B., Merzouki, R.: Engineering hierarchical complex systems: an agent-based approach – the case of flexible manufacturing systems. In: Service Orientation in Holonic and Multi Agent Manufacturing Control, Studies in Computational Intelligence, vol. 402. Springer (2012)

[23] Morvan, G., Veremme, A., Dupont, D.: IRM4MLS: the influence reaction model for multi-level simulation. In: Bosse, T., Geller, A., Jonker, C. (eds.) Multi-Agent-Based Simulation XI, Lecture Notes in Artificial Intelligence, vol. 6532, pp. 16–27. Springer (2011)

[24] Müller, J.P.: Towards a formal semantics of event-based multi-agent simulations. In: David, N., Sichman, J. (eds.) Multi-Agent-Based Simulation IX, Lecture Notes in Computer Science, vol. 5269, pp. 110–126. Springer Berlin / Heidelberg (2009)

[25] Navarro, L., Flacher, F., Corruble, V.: Dynamic level of detail for large scale agent-based urban simulations. In: Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011). pp. 701–708 (2011)

[26] Parunak, H.: Pheromones, probabilities and multiple futures. In: Multi-Agent-Based Simulation XI, Lecture Notes in Artificial Intelligence, vol. 6532, pp. 44–60. Springer (2011)

[27] Parunak, H.: Between agents and mean fields. In: Multi-Agent-Based Simulation XII, Lecture Notes in Artificial Intelligence, vol. 7124. Springer (2012)

[28] Parunak, H., Brueckner, S.: Concurrent modeling of alternative worlds with polyagents. In: Multi-Agent-Based Simulation VII, pp. 128–141. Lecture Notes in Computer Science, Springer (2007)

[29] Picault, S., Mathieu, P.: An interaction-oriented model for multi-scale simulation. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)

[30] Prévost, G., Bertelle, C.: Detection and reification of emerging dynamical ecosystems from interaction networks. In: Complex Systems and Self-organization Modelling, Understanding Complex Systems, vol. 39, pp. 139–161. Springer (2009)

[31] Soyez, J.B., Morvan, G., Merzouki, R., Dupont, D., Kubiak, P.: Multi-agent multi-level modeling – a methodology to simulate complex systems. In: Proceedings of the 23rd European Modeling & Simulation Symposium (2011)

[32] Taillandier, P., Vo, D.A., Amouroux, E., Drogoul, A.: GAMA: A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In: Desai, N., Liu, A., Winikoff, M. (eds.) Principles and Practice of Multi-Agent Systems, Lecture Notes in Computer Science, vol. 7057, pp. 242–258. Springer (2012)

[33] Uhrmacher, A.M., Ewald, R., John, M., Maus, C., Jeschke, M., Biermann, S.: Combining micro and macro-modeling in devs for computational biology. In: Proceedings of the 39th conference on Winter simulation. pp. 871–880 (2007)

[34] Vo, D.A., Drogoul, A., Zucker, J.D., Ho, T.V.: A modelling language to represent and specify emerging structures in agent-based model. In: Desai, N., Liu, A., Winikoff, M. (eds.) Principles and Practice of Multi-Agent Systems, Lecture Notes in Computer Science, vol. 7057, pp. 212–227. Springer (2012)

[35] Zeigler, B., Kim, T., Praehofer, H.: Theory of Modeling and Simulation. Academic Press, 2nd edn. (2000)