Engineering Development of Agents using the Cooperative Behaviour of their Components

Noélie Bonjean, Carole Bernon, Pierre Glize
Institut de Recherche en Informatique de Toulouse
University of Toulouse III
118 route de Narbonne, 31062 Toulouse cedex 9, France
Email: Firstname.Name@irit.fr

***Abstract*—The objective of the work undertaken here is to endow an agent-oriented methodology (such as ADELFE) with a semi-automatic tool for helping designers when drawing up the agents composing an adaptive multi-agent system (AMAS). This tool acts as a guide for enabling designers to influence the emergent global behaviour of an AMAS by acting on the local behaviour of its cooperative agents. The preliminary approach proposed in this article can be seen as a feasibility study aiming at developing a textual guide by considering the principles of the AMAS theory. Simulation of the behaviour of healthy and cancerous cells is used as a base for this study.**

I. Introduction

Multi-Agent Systems (MAS) are a recognised paradigm for designing and implementing complex applications and several agent-oriented methodologies were proposed to guide engineers in such a task [1] [2]. However when complexity prevents designers from discovering an *a priori* algorithm, designing MAS may also be a complex problem and new approaches may be adopted, such as bottom-up ones. Agents composing a MAS are identified, their behaviour and interactions defined to let them build the actual organisation of this MAS and let the global collective behaviour emerge from these interactions. This approach is adopted when building Adaptive Multi-Agent Systems (AMAS) in which every agent has a cooperative attitude [3]. However this emergence at the macro-level does not prevent engineers from having difficulties for finding the right micro-level cooperative behaviours and helping them is still an issue. ADELFE[1] was proposed as a guide dedicated for designing AMAS, but it has still some lacks. Therefore, additional guidelines and tools have to be provided for enriching it.

The objective of the work undertaken here is then to endow ADELFE with a semi-automatic tool for helping engineers when drawing up the agents composing an AMAS. This tool acts as a guide for enabling designers to influence the emerging global behaviour of an AMAS by acting on the local behaviour of its cooperative agents. The approach proposed in this article is still a preliminary one. It can be seen as a feasibility study aiming at developing a textual guide to facilitate designing agents. This textual guide is developed by studying an application related to the biological domain and simulating the behaviour of healthy and cancerous cells.

[1]Atelier de Développement de Logiciels à Fonctionnalité Emergente

Section II presents the context for this study and Section III positions it according to existing works. The simulation adopted as a base for the proposed approach is presented in Section IV. This enables expounding how developing such a simulation may be guided in Section V before concluding with some prospects.

II. Context

The aim of this study is to facilitate the design of agents composing a specific type of MAS, adaptive ones. This section introduces the concepts to which this study is related.

A. AMAS Theory and Cooperative Agents

Adaptive multi-agent systems at the heart of this study apply self-organisation principles according to the AMAS theory detailed in [3]. According to it, for designing a system whose functionality is adequate with what is expected by the designer, it is sufficient to drawing this system up with parts which have a cooperative attitude. This cooperative attitude means always trying to avoid, otherwise repair, situations that are judged, from the own point of view of an agent, as non cooperative. An agent does this by changing its relationships with other agents. This also changes the internal organisation of the multi-agent system it belongs to and as a result transforms the collective function the system is performing, making it adaptive.

A behavioural model of a cooperative agent was proposed and used during the microMega[2] project which aim was to model and simulate the behaviour of a unicellular microorganism [4]. Adopting such a model has firstly simplified the visualisation of the different parts composing the behaviour of a cooperative agent (see Fig. 1). Indeed, this model separates the nominal behaviour of a cooperative agent from its adaptive (or cooperative) one, this latter being itself broken into tuning, reorganisation and evolution behaviours. A designer may then work on each part of the agent's behaviour almost independently and implement and test them in a gradual way. These behaviours may be described as follows:

- the nominal behaviour represents the basic behaviour of an agent, what it does for achieving its local function without necessarily coping with Non Cooperative Situations (NCS),

[2]National ANR-funded project, 2005-2008

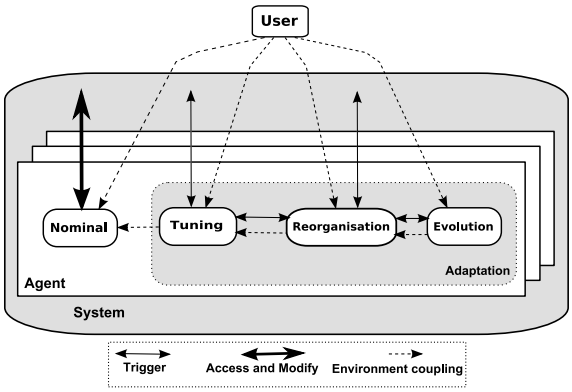

Fig. 1: Behavioural model of a cooperative agent.

- the adaptive behaviour, added on top of the nominal one, aims at dealing with these cooperative failures in three different ways:
  - by trying to adjust the values of the parameters used during the nominal behaviour (tuning behaviour),
  - by changing its relationships with others for trying to solve dead-ends (reorganisation behaviour),
  - and finally by self-removing or creating other agents if NCS still remain (evolution behaviour).

B. ADELFE and Living Design

For assisting engineers when designing AMAS, ADELFE was proposed. During the design phase, a specific activity is provided as a guide for designing agents: every non cooperative situation has to be identified and every preventing or repair action has also to be defined. This textual guide is still insufficient because currently, nothing in ADELFE guarantees that this identification is the proper and complete one. It is therefore necessary to enrich ADELFE with a better tool.

Ideally this automatic (or semi-automatic) tool would enable to develop the adequate behaviour for a cooperative agent, the very one that would allow achieving the functional adequacy of the AMAS this agent belongs to. This tool should also take into account the benefits brought by the behavioural model presented in the previous section by enabling designers to separately act on the different parts composing such a behaviour.

The following section presents this issue and gives insight into existing works that have links with this one.

III. The Problem

Considering the model given in Fig. 1, each part of the behaviour of an agent has an action on the other one. A NCS occurring during the nominal behaviour may trigger a repair action that will be performed during the adaptive behaviour. Depending on the designer's degree of familiarity with the AMAS concepts, the behaviour of an agent may be devised in two ways:

- The designer succeeds in identifying non cooperative situations. He certainly designs the nominal behaviour of agents by including some actions that enable an agent to prevent or repair non cooperative situations it will encounter. Furthermore, he is also capable of separating the nominal behaviour from the adaptive one and implements an agent according to the behavioural model presented above.
- In the second case, this designer does not really know how to identify non cooperative situations for the agent he is designing, or this identification is incomplete. Furthermore, he does not know how to define the actions this agent has to perform for staying in a cooperative state. The nominal behaviour he designs essentially consists in the basic behaviour of this agent and does not concern the aspects related to its cooperative attitude. Such a designer encounters difficulties for finding the behavioural dichotomy the model lays down.

In both cases finding every non cooperative situation an agent may encounter is not guaranteed, let alone defining every action required for removing such situations. Theoretically this would result in the inability of agents to collectively achieve a functionally adequate global function: what emerges from their interactions will not suit the designer's expectations. Ideally ADELFE should help designers for verifying these points and/or complete the agent design they made.

A. Simulating for Designing?

In 2002, using simulation for AOSE and designing agents was a challenge as underlined in [5]. For a few years now, steps are made in this direction with several research works using simulation in the AOSE domain [6]–[8].

Some works were made in this sense for enriching ADELFE as mentioned above. Simulation has been used for automatically detecting NCS in an AMAS prototype [9]. Simulation has also be used for making agents self-adjust their behaviour by making their behavioural rules self-reorganise [10]. Although both approaches had conclusive results for rather simple applications, they nevertheless have some drawbacks. The kind of agents that was taken into account (which are situated in an environment and communicate in an indirect manner through this environment) and the underlying use of the SeSAm platform [11] (need to become familiar with it, performance problems when a great number of rather complex agents are simulated) may be considered as limitations. Furthermore enabling a designer to make a prototype of his system before really implementing it is interesting; however, our main aim is rather to make a first step towards enriching ADELFE with a "good practice guide" before (semi-)automating this guide through an appropriate tool.

B. The Adopted Approach

The issue here is therefore to study how general principles may be extracted from the features of AMAS, and more especially those of cooperative agents, in order to deliver this guide.

An AMAS simulating a behaviour is used as the base for this study. A designer of MAS, who is considered as being

unfamiliar with the way of implementing AMAS, follows ADELFE and designs agents for achieving a first functional version of this system. This version, probably only based on the nominal part of agents, will likely not be the proper one. Therefore, the second step is to try and improve this implementation by acting on the different parts of the adaptive behaviour an agent may have.

IV. Case Study: Simulating Cells

To stay in line with the application domain of microMega, and because biological phenomena are complex and dynamic, a simulation inspired from the biological world was chosen. Actually, simulation and modelling in biology are a very active field of research, notably when cells are involved [12]. Mathematical models, based on differential equations e.g., may be used [13] or less classical tools such as Petri nets [14], cellular automata [15] or neural networks. For a few years now, MAS are also contributing to this effort, mainly because they are able to scale and to model specific properties in a more comprehensible way than mathematical models [16]. Among these works, some are more precisely interested in simulating the behaviour of microorganisms or cells [17]–[19].

This section therefore describes the features of the target application and how it was implemented.

A. Description of the Simulation

In a living organism, mitosis enables division of a eukaryotic cell into two daughter cells and apoptosis makes such a healthy cell degrade and die. Mutations may also occur (due to radiations, viruses, genetic predispositions, etc.) by copying errors of genetic material during division. Most of the time, mutations are not harmful for the cell or are repaired by internal mechanisms. However, some changes may lead to malignant cells leading to cancer. Among other properties, a cancerous cell is not able to repair DNA anymore, divides in an unchecked way and does not die. Cells "communicate" with their environment by endocytosis, which enables them to absorb material from outside, or exocytosis, by secreting material to the extracellular environment. Molecules which may be released are moving through the interstitial tissue. Some of them play a role in the paracrine signalling by carrying information from a cell to another one [20]. Some molecules also supply cells with energy. In both cases, moving make them degrade with time.

According to this simplified biological context, the aim of the considered application is to simulate the behaviour of healthy and cancerous cells. The development process of ADELFE was applied for analysing the requirements, verifying the AMAS adequacy and identifying the agents involved in this MAS, and designing their nominal behaviour with respect to the basic biological knowledge previously presented. Due to lack of space and because the objective of the paper is not to focus on it, only some details are given.

Because cells are autonomous, have a local goal (survive for healthy ones and proliferate for malignant ones), have only a partial view of their environment, two types of agents were identified : healthy cells and cancerous ones. Molecules are moving in an autonomous way, have also a limited perception of their environment but they do not have a local goal and are not considered as agents. Message passing simulates the absorption of a paracrine molecule by a cell and the related communication. Healthy cells may divide, mutate, die, signal molecules, or detect and absorb molecules. Cancerous cells do not mutate and die.

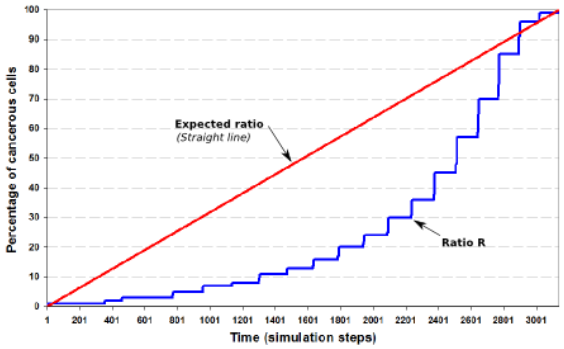

Fig. 2: Ratio of cancerous cells over time.

B. Collective Nominal Behaviour Obtained

The designer of this application is regarded as being able to deal with only the nominal behaviour of agents. As a result, the nominal behaviour of cells was implemented and the adaptive behaviour is what the guide has to help finding.

To see how the collective behaviour obtained for this MAS fits a user's needs, the ratio R of cancerous cells was studied. $R = \frac{C}{C+H}$ where C is the number of cancerous cells and H, the number of healthy ones. The curve obtained for R is shown on the right part of Fig. 2.

A healthy cell has a given probability of mutation and proliferates only if some free space is near it. On the contrary, a cancerous cell tries to push healthy ones if it does not have available space around it and divides itself if it succeeds. As a result, a slow increase of the number of cancerous cells can be seen on the curve at the beginning of the simulation because few mutations occur yet. However, once several malignant cells appear, they proliferate in a faster way than healthy ones, also because they do not die, and invade the tissue with time.

Let us suppose now that this collective behaviour is not the one expected by the end-user of this simulated system, e.g., a biologist. This user would like to obtain a curve with another shape; for instance, one that would express a more regular appearance of cancerous cells which is shown on the left part of Fig. 2. This shape has no biological reality and was chosen only for studying how the engineer could be guided to adapt the local behaviour of cells in order to obtain the new expected collective behaviour.

V. Guiding the Simulation

Actually, changing the behaviour of an agent may be done by changing the basic function it performs, its nominal behaviour. However, this choice is contrary to the hypothesis

we previously made; only the adaptive behaviour has to be built or modified. This means modifying one or all the parts of this behaviour: tuning, reorganisation and evolution. Since this study is a first step towards our goal, only the tuning part will be reviewed.

A. Act on the Tuning Behaviour of Agents

In general, the tuning behaviour of an agent has to be modified in order to get the expected global behaviour (the expected shape for the studied curve, e.g.). Modifying the value of the parameters that are used by an agent is going to modify how the nominal behaviour, which uses these very parameters, performs and therefore is going to influence the global behaviour of its collective. The issue lies in identifying the right parameters that have to be calibrated in order to get the proper behaviour.

The approach proposed is to study first how the parameters of an agent may promote its possible actions. Which actions have to be influenced (fostered or encouraged) in order to get the expected behaviour is then determined. Since a parameter may influence one or several actions, these relationships have to be gradually propagated for determining the proper parameter to change in order to modify the actions previously identified.

B. Relate Parameters and Actions of Agents

By leaning on the application domain knowledge, the designer has to establish a table relating the actions agents may perform and the parameters these agents use. Actually, this table should be done before beginning implementation, it is also a means to see how agents behave and to reflect on what they perform according to what they have to do.

Such tables were built for healthy cell agents (see Table I) and cancerous ones (see Table II). In this application, 18 parameters may influence the 7 actions of healthy cells:

- A1 - Proliferate,
- A2 - Mutate,
- A3 - Die,
- A4 - Absorb a paracrine molecule,
- A5 - Signal a paracrine molecule,
- A6 - Absorb a "simple" molecule,
- A7 - Signal a "simple" molecule.

or the 5 ones of cancerous cells (A1', A4', A5', A6' and A7'). Due to lack of space, out of the 18 parameters, 12 parameters are considered in the tables built, those that are only required for the coming reasoning:

- P1 - Lifetime of a cell: lifetime in the means case,
- P2 - Energy of a cell: internal energy a cell has for living,
- P3 - Molecule concentration in the environment: number of molecules a cell may perceive,
- P4 - Environmental conditions: occupied spaces surrounding a healthy cell and free ones for a cancerous one,
- P5 - Proliferation speed: expressed in simulation steps,
- P6 - Energy cost for proliferating: energy required for this action,

TABLE I: Influence table for healthy cell-agents.

Param/Action	A1	A2	A3	A4	A5	A6	A7
P1	+		- -				
P2	+++	++			++	++	-
P3	+			+++	- -	- -	+++
P4	- -	+++	+++				
P5	++		-				
P6	-		++				
P7			+		-		
P8			+				-
P9			+			-	
P10			++				
P11		+		+++			
P12		+++					
. . .							

- P7 - Energy cost for signalling a paracrine molecule,
- P8 - Energy cost for absorbing a molecule of any type,
- P9 - Energy cost for signalling a "simple" molecule,
- P10 - Apoptosis signals: signals which tend to influence death when overpopulation occurs,
- P11 - Sensitivity threshold to paracrine: expresses the ability of a cell to react to paracrine molecules,
- P12 - Mutation rate: number of cells that may mutate during a lifecycle.

For each (parameter, action) pair, two criteria are represented in these influence tables:

- In which direction (increase/decrease) the given parameter may vary for promoting the action. This is expressed by $+$ or $-$.
- The influence of the given parameter on the action. This is expressed by the number of symbols $+$ or $-$ used.

For example, the parameter *P1-Energy of a cell* has to increase for promoting the action *A1-Proliferate* of a healthy cell and this influence on *A1* is high because three $+$ are used. *P1* has also an impact on the action *A7-Signal a "simple" molecule*. In order to encourage a healthy cell to make this action, *P1* has to be decreased and this influence is low because only one $-$ is used.

Once these tables designed, a reasoning has to be done in order to modify the behaviour of cell-agents.

C. Propagate Influences

A parameter may influence several actions (*P1* influencing *A1* and *A7*, e.g.), and by modifying it, the related actions will also be promoted or discouraged, depending on the impact of this parameter. Therefore before modifying a parameter, studying how its influences are propagated from an action to another one is required. The influence tables previously built will be a help for this study.

For trying to obtain a more linear curve for R, starting from Fig. 2, R has to increase when R is below the straight line which represents its expected value and it has to decrease when it is above this line.

TABLE II: Influence table for cancerous cell-agents.

Parameter/Action	A1'	A4'	A5'	A6'	A7'
P1					
P2	+++		++	++	-
P3	+	+++	- -	- -	+++
P4	++				
P5	++				
P6	-				
P7			-		
P8					-
P9				-	
P11		+++			
. . .					

In a first step, R is going to be increased. Since R is a fraction, it increases when its numerator does and/or its denominator decreases. Since C cannot decrease, C has to increase or S has to decrease. To make C increase and S decrease, the action *A1-Proliferate* has to be promoted for cancerous cells, *A1'-Proliferate* has to be repressed for healthy ones and *A2-Mutate* as well as *A3-Die* have to be promoted for healthy cells.

Each action has to be analysed depending on the relationships it has with the parameters and how this influence propagates.

1) Promote "A1'-Proliferate" for Cancerous Cells: Proliferation of cancerous or healthy cells are promoted by the same parameters except *P4-Environmental conditions*. Therefore, promoting *A1'* by playing on *P1*, *P2*, *P3*, *P5* or *P6* will also promote *A1*, and *A1* has to be deserved. These parameters cannot be modified. The *P4* parameter represents the available space around a cell and is calculated by $(NCC + NHC + AS) - (NCC + NHC)$ where NCC is the number of neighbouring cancerous cells, NHC is the number of neighbouring healthy cells and AS is the number of available spaces around the cell. This formula should give the highest possible result for promoting *A1'*.

2) Promote "A1-Proliferate" for Healthy Cells: As above, some parameters cannot be changed. However other parameters have an impact on *A1* and do not influence *A1'*:

- *P1-Lifetime of a cell* has a low influence on *A1*. Deserving it should be obtained by decreasing *P1* and this would strongly promote *A3-Die* which is the expected effect.
- *P4-Environmental conditions* has a high influence on *A1*. It represents the occupied spaces around an healthy cell. *P4* is increased by promoting *A1/A1'* for each kind of cells or by deserving *A3*. The former alternative is contrary to the expected effect for healthy cells and redundant for what was done above for cancerous ones. The latter alternative is the opposite of the expected effect. Another solution would be to compute the value of *P4* in a different way. The current formula is $(NCC + NHC)/(NHC + NCC + AS)$. By adding the respective coefficients a, b and c to NCC, NHC and AS in order to reflect their importance, a new formula may be used: $(a \times NCC + b \times NHC)/(a \times NCC + b \times NHC + c \times AS)$. Furthermore, increasing this parameter would also promote, with the same importance, the actions *A2-Mutate* and *A3-Die* and this goes in the right direction.

3) Promote "A3-Die" for Healthy Cells: This action is influenced by *P1-Lifetime of a cell* and *P4-Environmental conditions* which were studied before. Some other parameters are also involved (from *P5* to *P10*). *P10-Apoptosis signals* influence solely this action and acting on them may be easy. Energy costs (from *P6* to *P9*) could be increased for promoting *A3-Die*. However, by studying propagation, these costs have also an impact on other actions such as absorption and signalling of molecules, and this could modify *A2-Mutate*; on the other hand, *A1-Proliferate* could also be deserved, and so on. Not modifying these parameters seems to be a good choice to try and avoid a too great influence on the system.

4) Promote "A2-Mutate" for Healthy Cells: Among the parameters that were not examined before, *P12-Mutation rate* has no influence on any other actions than *A2*, therefore it may be modified as needed. Adjusting *P11-Sensitivity threshold* would be propagated on *A4-Absorb paracrine* without any harmful effect on the system.

Once this propagation studied, some parameters may vary while others cannot. According to these new conditions, modification of some of the potential modifiable parameters has then to be done.

D. Tune Parameters

Table III shows the parameters that may vary (in bold font) or not (normal font) for *promoting* ($+$ before the name of an action) or *deserving* ($-$) actions of healthy or cancerous cells (their specific parameters or actions are distinguished from those of healthy cells by a ' following their name).

Parameters for which propagation is not opposite to the expected result are first modified. *P1*, *P4*, *P11* and *P12* are chosen, especially because they have the highest influence. The formula related to *P4* is thus modified for making cancerous cells have a higher impact and becomes $(2 \times NCC + NHC)/(2 \times NHC + NCC + AS)$.

E. Verify the Impact of Tuning

Once the above reasoning done and parameters modified in the simulation code, a new kind of curve for R was obtained (see Fig. 3). The first part of this curve is more linear than previously and gets closer to a straight line. However, its second part (from step 800) does not properly fit expectations.

Therefore, the same reasoning has to be done in order to make the top curve as linear as possible by decreasing R without cancelling the work done previously. An opposite operation has to be done: deserve *A1'-Proliferate* for cancerous cells, *A2-Mutate* and *A3-Die* for healthy ones, and promote *A1-Proliferate* for healthy cells. Since the benefits brought by the previous adjustment have to remain, the parameters that may be modified are those that were still untouched. In this case, only *P4'* was not modified and the value of environmental conditions for cancerous cells is then turned down.

TABLE III: Influence table for making R increase.

Parameter/Action	− A1	+ A2	+ A3	+ A1'
P1	-		- - -	
P2	- - -	++		+++
P3	-			+
P4	+++	+++	+++	
P4'				++
P5	- -		-	++
P6	+		+	-
P7			+	
P8			+	
P9			+	
P10			++	
P11	+			
P12		+++		

F. Observe the Collective Behaviour Obtained

After these two adjustments, a typical final curve obtained is shown in Fig. 4. It is more linear and looks like the one end-users expected. Therefore the collective behaviour of the cells is more in line with the expected one. As expected, the shape of the curve shows that cell-agents have modified their behaviour in the right direction.

It is also worth observing that the collective behaviour before and after modifications made on the tuning behaviour of cells does not give the same spatial distribution for cancerous cells, as shown in Fig. 5. In this figure, healthy cells are empty hexagons, cancerous ones are black hexagons and paracrine molecules are dots between cells. To simulate an "infinite" tissue, cells on opposite borders are neighbours. These snapshots were taken when around 10 cancerous cells appeared. They show that cancerous cells tend to be more scattered in the simulated tissue when only the nominal behaviour was concerned. After tuning, they tend to form clusters because these cancerous cells can mainly be created at the frontiers of clusters, and this prevents their exponential development. Of course as time goes by, the number of cancerous cells increases and their distribution tends to be more dense in both cases. However the initial disparity is consistent with the results and the aim of obtaining a straight curve.

Furthermore, this disparity and clusterisation was obtained without reasoning on this macro-level; only by studying relationships between parameters and actions *inside* the agents. Here also, there is emergence of a phenomenon at the macro-level which was controlled by cooperation at the micro-level. More precisely, there are three levels:

1) The macro-level which corresponds to the cell tissue. It is used to observe the global behaviour and to enter feedbacks according to the end-user wishes.
2) The meso-level constituted by the individual cells, where *ad hoc* feedbacks arrive according to their type (cancerous or not).
3) The micro-level corresponding to the cell components, where cooperative "negotiations" (as the "good practice guide" will explain in ADELFE) allows the determination of the relevant adjustments to do.

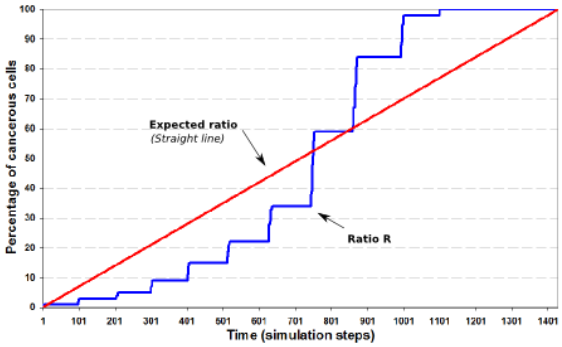

Fig. 3: New ratio obtained after the first tuning.

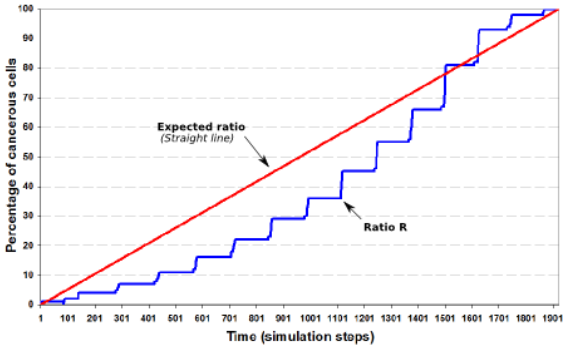

Fig. 4: Final ratio obtained.

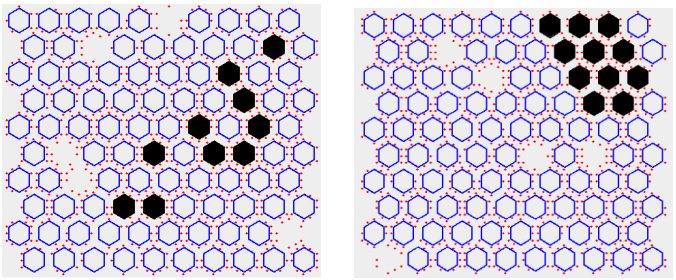

Fig. 5: Example of distribution obtained before (left) and after (right) tuning.

VI. Conclusion

In this article a first step was made towards integrating a methodological guide into ADELFE for helping engineers when designing the behaviour of cooperative agents. This preliminary textual guide may be summed up as follows. The designer has first to devise the nominal behaviour of agents involved in the AMAS he has to engineer. Depending on the feedbacks given by end-users, he has then to act on the adaptive behaviour of these agents to improve the collective behaviour of this AMAS. For the time being, only the tuning part of this adaptive behaviour was studied. The designer has then to establish relationships between parameters involved in the AMAS and actions agents may do (and this may be done

before, as a help for finding the nominal behaviour also). Then, these relationships have to be quantified and studied, especially by propagating effects a parameter may have on different actions (promote or deserve them). This enables finding which and how some parameters have to be changed for positively influencing the collective behaviour towards what is expected by end-users.

A MAS simulating cancerous cell proliferation in a tissue was used for grounding this demonstration. Although it cannot be considered as having a strong biological reality, it accounts for an interesting complex use case, at least for tuning, because it has several interrelated actions and parameters. The proposed approach was applied on this cell simulation for playing on the evolution of the ratio of cancerous cells over time and trying to influence the shape of the related curve. By changing some parameters used by healthy and cancerous cells, this curve was actually changed in the right direction. As a consequence the distribution of cells in the tissue was also modified which shows that emergence of a phenomenon at the global level may be influenced by changes in the local behaviours.

However, a lot of work has to be done yet. First, it is necessary to study how reorganisation and evolution behaviours could be used for guiding the engineer, then enriching ADELFE with this (still textual) guide could be done, once formalised. The last step would be to automate this guide by implementing the related tool as an AMAS. Indeed, discovering the parameters and actions to adjust in a complex adaptive system clearly corresponds to a specific agentification level, not directly required by the end-user problem. In this tool, for at least the tuning part, actions and parameters would be considered as cooperative agents. Their collective goal would be to find the right parameters to modify depending on the feedbacks given by end-users which would be considered as NCS that these agents should avoid and solve. Furthermore, all the results observable by an end-user would have to be agentified (e.g. the curve shown on Fig. 2 would be an agent in order to reason cooperatively on function R when an external feedback occurs). Consequently, engineering semi-automatically the development of a complex adaptive system must be AMAS-compliant, even when this complex system itself is not AMAS-designed.

Consequently this tool is not ADELFE-dependent and could be included in any methodology devoted to the development of complex adaptive systems. Nevertheless, a lot of work and tests are required before considering such a kind of deployment.

REFERENCES

[1] F. Bergenti, M.-P. Gleizes, and F. Zambonelli, *Methodologies and Software Engineering for Agent Systems*. Kluwer Publishing, 2004.

[2] B. Henderson-Sellers and P. Giorgini, *Agent-Oriented Methodologies*. Idea Group Pub, June 2005.

[3] D. Capera, J.-P. Georgé, M.-P. Gleizes, and P. Glize, "The AMAS Theory for Complex Problem Solving Based on Self-organizing Cooperative Agents," in *12th IEEE International Workshops on Enabling Technologies, Infrastructure for Collaborative Enterprises, Linz, Austria*. IEEE Computer Society, June 2003, pp. 383–388.

[4] C. Bernon, D. Capera, and J.-P. Mano, "Engineering Self-Modeling Systems: Application to Biology," in *International Workshop on Engineering Societies in the Agents World*, ser. LNCS, A. Artikis, G. Picard, and L. Vercouter, Eds., no. 5485. Springer, 2009, pp. 236–251.

[5] A. Uhrmacher, "Simulation for Agent-Oriented Software Engineering," in *First International Conference on Grand Challenges for Modelling and Simulation, San Antonio, Texas*, W. Lunceford and E. Page, Eds. San Diego: SCS, 2002.

[6] G. Fortino, A. Garro, W. Russo, R. Caico, M. Cossentino, and F. Termine, "Simulation-Driven Development of Multi-Agent Systems," in *Workshop on Multi-Agent Systems and Simulation, Palermo, Italia*, 2006.

[7] M. Cossentino, G. Fortino, A. Garro, S. Mascillaro, and W. Russo, "PASSIM : A Simulation-based Process for the Development of Multi-Agent Systems," *Int. Journal of Agent-Oriented Software Engineering*, vol. 2, no. 2, pp. 132–170, 2008.

[8] L. Gardelli, M. Viroli, M. Casadei, and A. Omicini, "Designing Self-organising Environments with Agents and Artefacts: A Simulation-driven Approach," *Int. J. Agent-Oriented Softw. Eng.*, vol. 2, no. 2, pp. 171–195, 2008.

[9] C. Bernon, M.-P. Gleizes, and G. Picard, "Enhancing Self-Organising Emergent Systems Design with Simulation," in *International Workshop on Engineering Societies in the Agents World (ESAW), Dublin*, ser. LNCS, G. O'Hare, M. O'Grady, A. Ricci, and O. Dikenelli, Eds., vol. 4457. Springer-Verlag, September 2007, pp. 284–299.

[10] S. Lemouzy, C. Bernon, and M.-P. Gleizes, "Living Design: Simulation for Self-Designing Agents," in *European Simulation and Modelling Conference (ESM), Malte*, J. Sklenar, A. Tanguy, C. Bertelle, and G. Fortino, Eds. Eurosis, October 2007, pp. 432–436.

[11] F. Klügl, R. Herrler, and O. Oechslein, "From Simulated to Real Environments: How to use SeSAm for Software Development," in *Multiagent System Technologies - 1st German Conference MATES*, ser. LNAI, M. Schillo, Ed., vol. 2831. Springer-Verlag, 2003, pp. 13–24.

[12] T. Fisher and T. Henziger, "Executable Cell Biology," *Nature Biotechnology*, vol. 25, no. 11, pp. 1239–1249, November 2007.

[13] P. Mendes, "GEPASI: A Software Package for Modelling the Dynamics, Steady States and Control of Biochemical and other Systems," *Computer App. in the Biosciences*, vol. 9, no. 5, pp. 563–571, October 1993.

[14] C. Li, Q. W. Ge, M. Nakata, H. Matsuno, and S. Miyano, "Modelling and Simulation of Signal Transductions in an Apoptosis Pathway by using Timed Petri Nets," *J. Biosci.*, vol. 32, pp. 113–127, 2007.

[15] H. Silva and M. Martins, "A Cellular Automata Model for Cell Differentiation," *Physica A*, no. 322, pp. 555–566, 2003.

[16] E. Merelli, G. Armano, N. Cannata, F. Corradini, M. d'Inverno, A. Doms, P. Lord, A. Martin, L. Milanesi, S. Moller, M. Schroeder, and M. Luck, "Agents in bioinformatics, Computational and Systems Biology," *Briefings in Bioinformatics*, vol. 8, no. 1, pp. 45–59, 2006.

[17] M. d'Inverno and R. Saunders, "Agent-Based Modelling of Stem Cell Self-organisation in a Niche," in *Engineering Self-Organising Systems: Methodologies and Applications*, ser. LNCS, S. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, and R. Nagpal, Eds., vol. 3464. Springer, 2004, pp. 52–68.

[18] L. Dib, N. Bonnet, Z. Guessoum, and M. Laskri, "Multi-agent System Simulating Tumoral Cells Migration," in *18th Australian Joint Conference on Artificial Intelligence*. Springer Verlag, December 2005, pp. 624–632.

[19] K. Webb and T. White, "Cell Modeling with Reusable Agent-based Formalisms," *Applied Intelligence*, vol. 24, no. 2, pp. 169–181, April 2006.

[20] T. J. Weber, R. W. Siegel, L. M. Markillie, W. B. Chrisler, X. C. Lei, and N. H. Colburn, "A Paracrine Signal Mediates the Cell Transformation Response to Low Dose Gamma Radiation in JB6 Cells," *Molecular Carcinogenesis*, no. 43, pp. 31–37, 2005.