

Visualizing Bacteria Quorum Sensing

Maria Schwarz, Daniela Romano and Marian Gheorghe¹

Abstract. Large populations of bacteria communicate by sending into the environment some specific signalling molecules. A bacterium will sense the population density by the concentration of the signalling molecule. This process, known as “quorum sensing”, is used in this paper to show the emergent behaviour of a bacterium colony in various circumstances. The biochemistry associated with quorum sensing has been modelled and an agent based approach has been used to simulate the behaviour of the colony. In addition, a 3-D environment has been created to illustrate how the emergent behaviour occurs within the colony as a result of local immediate real-time interactions, while a data file is generated to visualise the behaviour of the colony over time as a 2D graph.

1 INTRODUCTION

There are numerous ways in which members of various communities communicate. Even simple organisms have different ways to pass information among them. Quorum sensing (QS) is one of these communication mechanisms, which has been known since the end of the 1960s. A review of various bacterium populations, the LuxR-LuxI family, and how they communicate with each other is presented in [1].

Bacteria use QS to coordinate different behaviours. For example the light emission in luminescent bacteria (the *Vibrio fischeri* or *Vibrio harveyi*), division in *Escherichia coli*, or biofilm formation in infective bacteria like *Pseudomonas aeruginosa*. The most important role in QS is played by signalling molecules, called autoinducers, which are produced by bacteria. These autoinducers diffuse through the bacterium membrane into environment. The accumulation of it in the environment takes place if there is a high concentration of bacteria in that space. If this occurs, a special gene will be activated and an increase in signalling molecules production is observed [2]. Therefore the behaviour of the bacteria change and the population density will be controlled [1], [3].

The QS process is widely studied today due to its importance in regulating the colony behaviour, but also as a computational paradigm applied to investigate computer networks, artificial life phenomena [4].

The QS mechanism represents one of the most important cell-to-cell communication processes [4], which requires fine grain knowledge of the complex (micro-)biological system involved and the associated communication procedures in place. In this

respect various approaches have been employed, ranging from mathematical models that address a global behaviour of the community to different agent based methods that represent every bacterium with its internal processes and local interactions between the community members. The agents, representing individuals in their environment, act in a totally autonomous manner [5] and coordinate their behaviour based on special rules. Thus, they interact with the other agents from their neighbourhood.

In order to study the behaviour of these colonies or to investigate certain properties of the computational paradigms associated with, adequate modelling and simulation tools have been provided. In the last few years improved computer technology has paved the way to visualise this process without major performance difficulties and it is possible to visualise a QS behaviour-using species of bacteria in three dimensions.

There have been built various agent based tools and software platforms. An example, which uses the agent-based method, is the visualisation of the motion and behaviour of a flock of birds [6]. Thereby each bird acts autonomously “to its local perception of the dynamic environment, the laws of simulated physics that rule its motion and a set of behaviours programmed into it by the ‘animator’” [6].

There are also open source environments allowing to create agent-based models and to simulate them on various platforms. NetLogo and SWARM are both agent-based modelling environments suitable to specify, study and simulate complex systems with a high number of agents [7], [8]. Regrettably, both frameworks have a pretty elementary construction so that there is no parallel processing available. This problem is addressed by another agent based toolkit, called MASON, which is written in java [9]. This modelling toolkit provides parallel processing, which is controlled by a scheduler. Flexible Agent Modelling Environment (FLAME, for short) has been developed by a team of the University of Sheffield [10], and has a great flexibility in its usage and has been extensively employed to model complex biological and economical systems [10]. Every agent is abstractly represented like a state machine, all the processes occurring inside are specified in an XML format and they share a common special memory component that allows agents to communicate and store information [5].

The paper aims to uncover aspects related to the emergent behaviour of a colony of *Vibrio fischeri* bacteria with respect to various parameters, like cell density, amount of signals produced, signal diffusion in the environment. The following topics are covered. Section 2 gives a brief description of the model used in the agent-based approach. In section 3 an overview of the implementation of the agent model within FLAME framework and the visualisation part are presented. Section 4 presents the experiments concerning the simulation and shows the results obtained and their significance for the

¹ Department of Computer Science, The University of Sheffield, Regent Court, Portobello Street, Sheffield S1 4DP, UK
Email: shef@schwarz-maria.de,
([d.romano](mailto:d.romano@dcs.shef.ac.uk), [m.gheorghe](mailto:m.gheorghe@dcs.shef.ac.uk))@dcs.shef.ac.uk

emergent behaviour. The last section evaluates the findings and proposes updates to the existing approach.

2 THE CONCEPTUAL MODEL

Many species of bacteria regulate gene expression in response to increasing cell population density. Bacteria produce and release signalling molecules (called autoinducers) that accumulate in the environment as the cell density increases. When a certain threshold is achieved a signalling transduction cascade is triggered and leads finally to a change in behaviour by increasing the production of the signalling molecules and leading to other individual changes. In *Vibrio fischeri* population a protein called LuxI is responsible to synthesise an acylated homoserine lactone signalling molecule (HSL, for short) [3]. The HSL signalling molecule diffuses out of the bacterium, into the environment, following a gradient concentration or it penetrates in when the environment surrounding it has a higher concentration of these molecules than the bacterium. If the concentration of this autoinducer reaches a special threshold, and therefore there are a high number of bacteria in a small space, a protein called LuxR reacts with the autoinducers by producing a complex, called LuxR-HSL. This complex is responsible for the activation of the transcription of a specific gene and the bacterium enters a new state, it becomes quorated and starts emitting light [3].

The agent model consists of two agent types: bacterium-agent and environment-agent. The bacterium-agent model is defined according to the rules defined by Figure 1. Usually each bacterium produces signalling molecules and proteins LuxR at a low rate (rules r1 and r2, respectively). The signalling molecules freely diffuse into the environment (r9) and both, signalling molecules and proteins degrade in time (r10, r11). Due to the low rate production of autoinducers, diffusion and degradation, the number of signalling molecules in the bacterium is too low and it will not start becoming quorated and producing light.

However, in specific circumstances when a certain threshold is reached somewhere in a specific part of the environment, near some bacteria, which only happens if a high number of bacteria are in one small space, the signalling molecules will diffuse back into those bacteria and increase more the concentration of autoinducers within them, triggering the quoration mechanism which leads eventually to producing light. This process is described in our model by the production of the complex LuxR-signal (r3), which in high concentration of the signalling molecule will not decompose back into its components and will next bind to a certain gene (r5) and will remain there and increase the production of signalling molecule and protein LuxR (r7, r8).

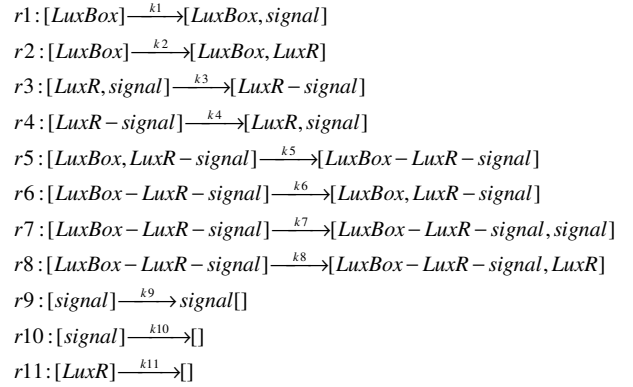


Figure 1. Equations describing all chemical interactions [11]

The bacterium-agent described above contains five different molecules that play an important role in producing the signalling molecule. The process of production, diffusion, degradation, combination of these chemical molecules is controlled by an exact stochastic method developed according to Gillespie algorithm [12]. Usually this algorithm iteratively selects the next rule to be applied and computes the time for this to be performed. In order to do this the current concentration of the chemicals involved in a rule and the kinetic rate (k1 to k11, in Figure 1) are considered.

Each environment field, modelled as an environment-agent, contains a defined number of signalling molecules (at the beginning zero). This number of signalling molecules is changed by the movement of them according to the concentration gradient.

3 IMPLEMENTATION

The general architecture of the implementation is given in Figure 2 and contains the agent-based system implementation and the graphical interface part.

The upper part of Figure 2 refers to the agent framework FLAME [13], [14] and the bottom part represents the user interface to the system.

To implement FLAME we had to build a XXML Model, which is written as an xml file. It defines all the agents, their status, associated functions (activities) and their dependencies with respect to messages sent between agents; messages owned by each agent are defined. Additionally, a C-based file, where all the functions and their features are set, needs to be provided. This file depends on a xml model defining the structure of the state machine. The initial state of the system must be defined (for example how many agents are in the system, their initial distribution, the initial concentrations of the chemicals present in each bacterium etc.).

Instances of the bacterium-agents and of the environment-agent are immersed into FLAME. Every bacterium-agent will run a slightly modified version of Gillespie algorithm to select the next rule to be used, adapted to match the requirements of the agent

framework. The initial concentration of the chemical molecules of the bacterium-agents will depend on various experiments made. The current Gillespie algorithm implementation computes only the next rule to be applied. All the agents of the system are able to send messages between them through a message board.

Thus the agents know independent of the other partners in the whole system what goes on in their close environment. Therefore every agent can behave irrespective of the size of the system or the collective number of agents [13], [14]. The only restriction is the dependency on the messages sent by agents in their close neighbourhood.

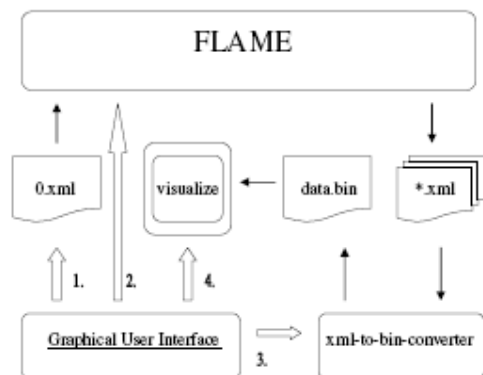


Figure 2. Overview of the implementation architecture

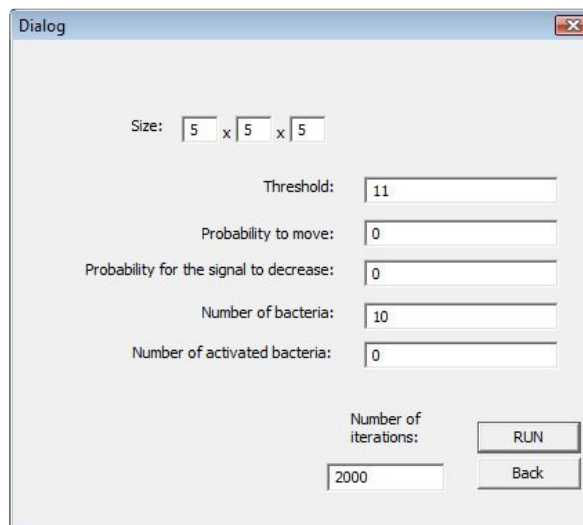
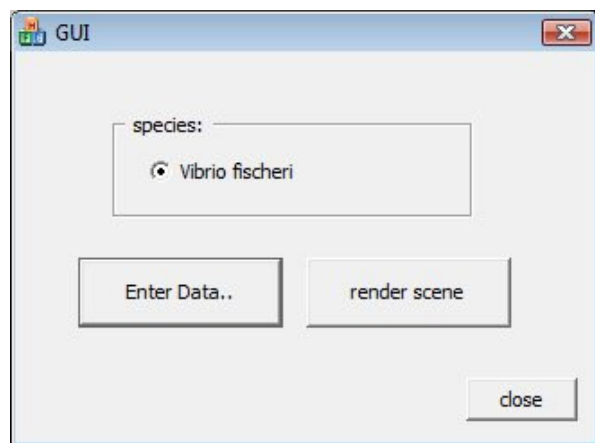


Figure 3. Screenshot of the graphical user interface and the child window of it

The Graphical User Interface (GUI), which is implemented with the Microsoft Foundation Class Library, provides four different functions. First it builds the 0.xml file which contains user defined data. Then, the second function deals with FLAME framework, which uses the initial xml file mentioned above, to run requested simulations and generate various output results into xml files. After running FLAME, the GUI executes the xml-to-bin-converter – the third function. Thus, xml files are converted to one single binary data file. This conversion helps improving the performance of the three-dimensional graphics process. The visualisation step, the last function, written in C++, using OpenGL library, displays the incoming data and outcomes of the simulation [15].

Figure 3 shows screen shots of the GUI. Firstly when the system starts, the top window appears. Through this window it is specified the species used in the simulation and visualisation (currently *Vibrio fischeri*), and, optionally setting various initial values, by pressing the “Enter Data” button. The “render scene” button may be used to render output data. After introducing the initial values requested by the current experiments a simulation phase is performed by hitting the “run” button. If a graphical output is requested then going back to the top window a new graphical representation may be obtained for the last simulation.

4 EXPERIMENTS & RESULTS

The agent model presented relies on a set of chemical reactions given by Figure 1.

k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	k11
0.002	0.002	0.009	0.001	0.01	0.002	0.25	0.2	0.01	0.03	0.02

Table 1. Used reaction rates

According to this model each reaction specifies the reactants and products as well as a kinetic rate. The values of these constants used in the following experiments are provided in Table 1.

According to the model presentation, given in Section 2, each bacterium produces signalling molecules and proteins at a basal rate, given by reactions r1 and r2. In certain conditions, when the concentration of the signalling molecule is high enough, in certain parts of the environment, the bacteria in that area will become quorated and instead of using the basal production rate for the above mentioned chemicals will switch to another state where reactions involving the use of the complex LuxR-signal, r7 and r8, are used. In a number of experiments below we will show how QS will emerge from local interactions when environment conditions are appropriate. To simulate the process of QS occurring within a bacterium colony, the right concentration of various chemicals is necessary to be identified. The concentration threshold is given by the amount of signalling molecule.

Firstly we started with five bacteria, spread randomly in a relatively big environment such as to avoid a high cell density population in some parts of the space. The initial values of the molecules are all zero except for LuxBox, which has the value one. This experiment is repeated five times and each time, 20000 iterations were performed.

	Max signal	Max protein	r1	r2	r7	r8
RUN 1	4	4	5318	4651	0	0
RUN 2	4	6	5301	4698	0	0
RUN 3	4	4	5349	4660	0	0
RUN 4	3	4	5340	4613	0	0
RUN 5	4	4	5371	4608	0	0

Table 2. Five isolated bacteria for 20000 steps

From Table 2 it follows that only rules r1 and r2 are used over 20000 iterations in each of the five experiments and none of the r7 or r8 is ever used. This shows all bacteria keep running the non-quorated state, producing signalling molecules and proteins at a basal rate. The maximum number of signalling molecule is 4 and of the protein is 6, across all experiments and iterations. These experiments clearly show that when the cell density is lower nothing happens even over a long period of time. This will be also reinforced by the latter experiments reported in Table 3. More than this, from Figure 4, it turns out that most of the time these values are between 2 and 3 and very rarely they reach values above 3.

initial concentration	activation after step range	highest concentration of signalling molecules	r1	r2	r7	r8
(500, 20)	2235 - 3855	63 - 88	593 - 989	500 - 915	70 - 156	66 - 123
(1000, 20)	734 - 966	58 - 95	195 - 309	179 - 281	194 - 241	168 - 196
(5000, 100)	680 - 888	73 - 112	182 - 261	146 - 249	203 - 238	159 - 185

Table 3. Behaviour of a bacterium colony

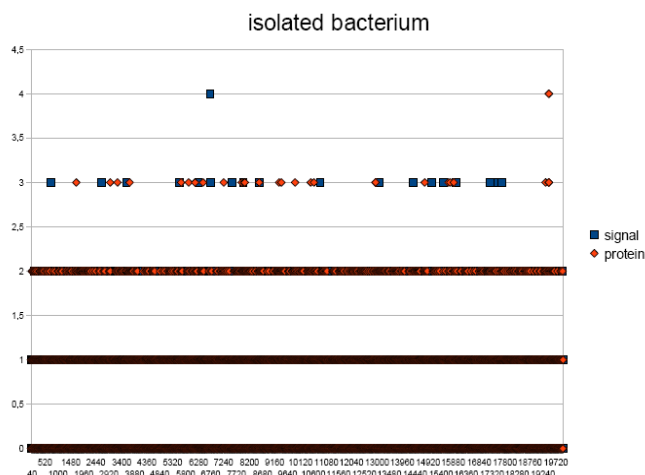


Figure 4. Signalling molecules and proteins in one isolated bacterium

From these experiments we have learned that if we consider 5 as a threshold for the number of signalling molecules that might trigger the QS, i.e., allowing rules r7 and r8 to be used, then this is coherent with individual behaviour of bacteria acting in isolation.

The second type of experiment we ran is with a colony with some special distribution of bacteria in the environment and different concentrations of the signalling molecule in the environment. All these experiments are run with 9 bacteria and 5000 steps. Five of these nine bacteria are very close to each other in one corner of the environment, whereas the other four are far away from that area. So we can compare the behaviour of the whole colony in these conditions and witness the occurrence of the emergent behaviour with respect to cell-density. In this colony we put into the bacterium that is in the middle of the group of 5 cells different initial values for the signalling molecule, so that this one, from the beginning, will be activated and will produce enough signalling molecules into the environment such as to saturate a certain part of it and to force the neighbour bacteria to accept the surplus of signalling chemicals. These initial values of the signalling molecule and protein are listed in the first column of Table 3 as a pair (x,y); x represents the number of signalling molecules and y the number of proteins. Three different experiments are reported – see Table 3. The other columns in Table 3 refer to data collected from the four bacteria surrounding the activated one and show when they become activated, the highest concentration of the signalling molecule and the number of times rules r1, r2, r7 and r8 are used. The other four bacteria are never activated, they keep producing the usual chemicals at a basal rate.

From Table 3 one can get that the activation step of the QS mechanism is between 680 and 3855, depending on various chemical concentrations of the above mentioned chemicals. The use of rules r1, r2 on the one hand and r7, r8, on the other hand refers to the two behaviour stages of the bacteria involved, *non-activation* – only the first rules are used and *activation* – only the last two are employed.

To illustrate the evolution in time of the colony and the transformations occurring within the environment, in Figure 5, a three dimensional visualisation framework is provided. In order to keep the visualisation process as simple as possible, the agents are shown as cubes depicted with different colours. These cubes are not solid blocks, but only frames. Therewith it is easy to view all the agents in the background. The colour of a bacterium-cube is white for an inactivated bacterium and yellow for an activated one. The environment-agents are grey at the beginning of the simulation and when signalling molecules start to appear they change to a dark green colour. With this colour-change it is possible for the user to see the diffusion of the signalling molecules and how do they accumulate in some parts of the environment and then the gradual activation of various bacteria.

In Figure 5 few visualisation screenshots are represented. The first picture shows the initialised state of the simulation, when only the central bacterium in the left-bottom group is active and thereby yellow. The other bacteria are all white and they only produce signalling molecules and proteins at a low rate. Remember the active bacterium has quite a massive surplus of signalling molecule which is released into the surrounding environment and after a while will enter the bacteria nearby. The second picture shows a snapshot of the time when the environment around the central bacterium has plenty signalling molecules and one of the other four bacteria from that part becomes active, turns to yellow. The green parts around the bacteria are the environment agents, which contain signalling molecules. The brighter these cubes are, the more signalling molecules are in these places.

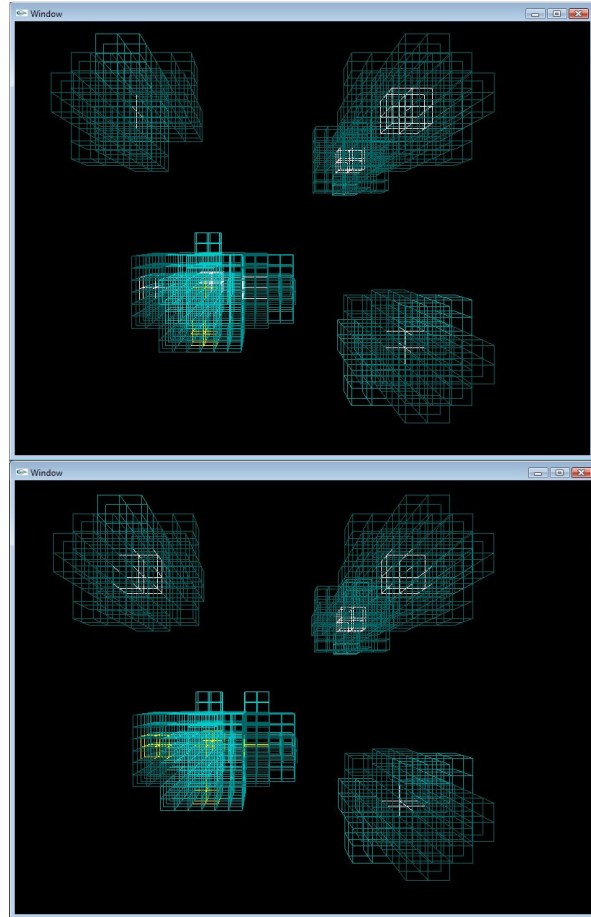
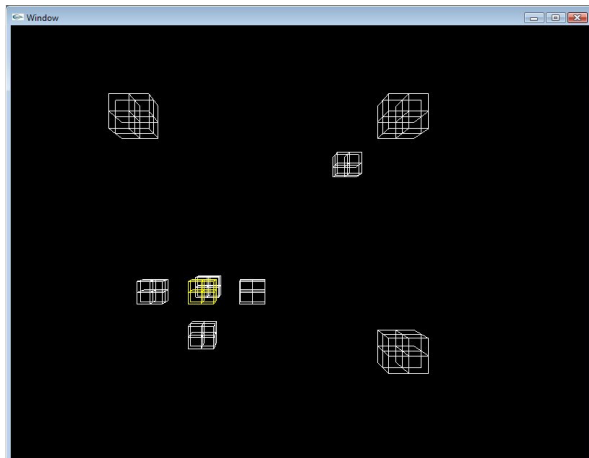


Figure 5. Screenshots of an example experiment with 9 bacteria

In the last picture all the bottom-left bacteria are activated whereas the other four remain inactivated and they will still be in that state even after 20000 steps.

These experiments clearly show a high dependency of the QS behaviour on the cell-density as well as a high stability with respect to parameters considered in these experiments.

The work reported in this paper may be also considered as a continuation of the investigation reported in [4] where QS phenomena are presented with respect to environment changes. This study represents a step forward by considering a broader range of experiments, with a well-established set of internal rules and fine-tuned parameters, within a stable range of values. The experiments clearly show when emergent behaviour occurs and how various rules and chemical concentrations contribute to this process.

5 CONCLUSIONS

The paper presents an agent based approach to simulate the behaviour of a bacterium colony with the aim of revealing some emergent properties with respect to QS phenomenon. A stochastic model based on Gillespie algorithm has been implemented for the agents running within a flexible agent platform, called FLAME. A graphical interface allowing to adjust various parameters of the model and to visualise the behaviour of the system is presented.

The current case study shows the potential of the approach to reveal emergent behaviour of a set of agents as a consequence of cell-density concentration in the environment and various ways of producing specific signalling molecules.

The model and the software platform described in this paper may be used to describe more complex systems and to identify specific behaviour patterns. This is one of our future plans and in order to make it achievable an intermediary step will be to investigate mechanisms to speed up the execution time of the simulator by migrating it on more efficient platforms, like clusters or graphical cards.

REFERENCES

- [1] W. C. Fuqua, S. C. Winans, and E. P. Greenberg. Quorum Sensing in Bacteria: the LuxR-LuxI Family of Cell Density-Responsive Transcriptional Regulators. In: *Journal of Bacteriology* 176: 269-275 (1994)
- [2] K. H. Neelson, T. Platt, and J. W. Hastings. Cellular Control of the Synthesis and Activity of the Bacterial Luminescent System. In: *Journal of Bacteriology* 104: 313-322 (1970)
- [3] M. B. Miller and B. L. Bassler. Quorum Sensing in Bacteria. In: *Annu. Rev. Microbiol.* 55:165-199 (2001)
- [4] N.Krasnogor, M. Gheorghe, G. Terrazas, S. Diggle, P. Williams, and M. Camara. An Appealing Computational Mechanism Drawn from Bacterial Quorum Sensing. In: *Bulletin of the European Association for Theoretical Computer Science.* (2005)
- [5] C. M. Macal and M. J. North. Introduction to Agent-based Modeling and Simulation, In: *Argonne National laboratory, MCS LANS Informal Seminar.* (2006) Available at <http://www-unix.mcs.anl.gov/~leyffer/listn/slides-06/MacalNorth.pdf> (access 07th December 2007)
- [6] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. In: *Computer Graphics*, 21: (4) 25-34. (1987)
- [7] U. Wilensky. NetLogo. (1999)
<http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston. IL
- [8] The Swarm Development Group Wiki.
http://www.swarm.org/wiki/Main_Page (access 7th December 2007)
- [9] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan. MASON: A New Multi-Agent Simulation Toolkit. In: *Proceedings of the 2004 SwarmFest Workshop.* (2005)
- [10] Mike Holcombe, Simon Coakley and Rod Smallwood. A General Framework for Agent-based Modelling of Complex Systems. In: *EURACE Working paper WP1.1* (2006)
- [11] F. Bernardini, F.J. Romero-Campero, M. Gheorghe, and M.J. Pérez-Jiménez. A Modeling Approach Based on P Systems with Bounded Parallelism. In: *Lecture Notes in Computer Science.* Springer. 4261/2006. Berlin/ Heidelberg. 49-65. (2007)
- [12] D. T. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. In: *The Journal of Physical Chemistry* 81: (25) 2340-2361. (1977)
- [13] <http://www.flame.co.uk> (access 27th November 2007)
- [14] <https://trac.flame.ac.uk/wiki/> (access 29th November 2007)
- [15] M. Woo et al. OpenGL Programming Guide. Third Edition., published by Addison-Wesley Longman, Amsterdam. Page 2 f. (2000)