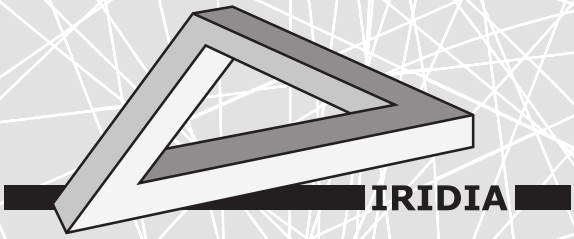


Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Analysing Robot Swarm Decision-making with Bio-PEPA

Mieke Massink, Manuele Brambilla, Diego Latella,
Marco Dorigo, and Mauro Birattari

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2012-005

April 2012

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:
IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2012-005

Analysing Robot Swarm Decision-making with Bio-PEPA

Mieke Massink[1], Manuele Brambilla[2], Diego Latella[1],
Marco Dorigo[2], and Mauro Birattari[2]

[1] Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo' (ISTI), CNR
Pisa, Italy {massink,latella}@isti.cnr.it
[2] IRIDIA, CoDE, Université Libre de Bruxelles,
Brussels, Belgium {mbrambil,mdorigo,mbiro}@ulb.ac.be

Abstract. We present a novel method to analyse swarm robotics systems based on Bio-PEPA. Bio-PEPA is a process algebraic language originally developed to analyse biochemical systems. Its main advantage is that it allows different kinds of analyses of a swarm robotics system starting from a single description. In general, to carry out different kinds of analysis, it is necessary to develop multiple models, raising issues of mutual consistency. With Bio-PEPA, instead, it is possible to perform stochastic simulation, fluid flow analysis and statistical model checking based on the same system specification. This reduces the complexity of the analysis and ensures consistency between analysis results. Bio-PEPA is well suited for swarm robotics systems, because it lends itself well to modelling distributed scalable systems and their space-time characteristics. We demonstrate the validity of Bio-PEPA by modelling collective decision-making in a swarm robotics system and we evaluate the result of different analyses.

1 Introduction

Swarm robotics is a novel approach to multi-robots systems. Swarm robotics systems (SRSs) are composed by tens or hundreds of robots which cooperate to perform a task, without a centralized controller or global knowledge. The goal of swarm robotics is to develop systems that are robust, scalable and flexible [1].

Analysing large and complex SRSs using physics-based simulations or directly with robots is often difficult and time consuming. For this reason, a common way to study these systems is by using models [2]. Models allow the developer to abstract from the complexity of a system and its implementation details and focus on the aspects that are relevant for the analysis. Different approaches are available to model a SRS. *Macroscopic modelling* [2] is commonly used for describing the collective behaviour of a system. Another approach, namely *microscopic modelling* [3], focuses instead on the behaviour of individual robots. Finally, *model-checking* has been used to verify formal properties of a SRS [4]. These approaches allow a developer to obtain different "views" of the system

behaviour. However, for each of these views, a different model is necessary. Producing different models greatly increases the complexity of the analysis process. Moreover, when dealing with different models, the issue of mutual consistency must be addressed.

In this paper we present a novel approach to model SRSs based on Bio-PEPA [5] which allows to obtain different consistent views of a system from the same formal specification. Bio-PEPA is a process algebraic language for biochemical and distributed systems. It has also been used[1] to analyse emergency egress [6] and crowd dynamics [7] which are systems characterized by a high number of individuals and lack of a centralized controller, aspects common also to SRSs. Bio-PEPA is well suited to analyse and develop SRSs; it provides for a clear specification at the microscopic level while providing also primitives for spatial description (e.g. *locations*) and for composition of individual robots (e.g *cooperation* operator). Moreover, Bio-PEPA allows to easily define *species*, which can be used to characterize groups of robots with specific attributes and actions; for instance, they can be used to differentiate between groups of robots performing different tasks at the same location. We use Bio-PEPA to develop a formal specification and analyse a collective decision-making behaviour which has been extensively studied in [8, 9]. The case study consists of a swarm of robots that have to collectively identify the shortest path between two possible choices. We validate our results against those presented in [8].

The outline of the paper is as follows. In Section 2, we present related work. In Section 3, we give a brief presentation of Bio-PEPA. In Section 4, we present the case study and its Bio-PEPA specification. In Section 5, we present and validate our results. Some conclusions are drawn in Section 6.

2 Related work

The most common approaches to modelling in swarm robotics are based on microscopic and macroscopic models. The main advantage of microscopic modelling is that it allows to study in detail the robot-to-robot and robot-to-environment interactions that are the key components of any SRS. Microscopic modelling, through stochastic simulation, can be used to analyse a system both in its equilibrium and far-from-equilibrium states. An example of a microscopic model of a SRS can be found in [10]. Macroscopic modelling, instead, considers only the swarm and its time evolution, ignoring the individual behaviour of the robots composing it. For this reason, it can be used to analyse systems composed by thousands of robots using fluid flow (Ordinary Differential Equations) approximation. However, the focus of macroscopic modelling is on the equilibrium analysis and thus it can be problematic to analyse a system in its far-from-equilibrium states or to understand specific problematic situations that could arise. A review on macroscopic modelling in swarm robotics can be found in [2]. A comparison between the microscopic and macroscopic models of a swarm robotics system is presented in [11].

[1] See http://www.biopepa.org for a complete list of publications.

A further way to model a SRS models is through mathematical logic. Models developed through mathematical logic can be used to formally verify given properties of a SRS by *automated* model checking (e.g. [12]). Up to now this approach has not been explored extensively in swarm robotics. Examples of model checking in swarm robotics can be found in [4, 13].

To perform stochastic simulation, fluid flow (ODE) approximation and model checking different models of a system are necessary. Our approach, instead, requires only a single Bio-PEPA specification permitting different kinds of system analyses.

3 Bio-PEPA

Bio-PEPA [5] is a process algebraic language that originally was developed for the stochastic analysis of biochemical systems. Bio-PEPA specifications consist of two main kinds of components. The first kind is called the *"species" component*, specifying the behaviour of individual entities. The second kind is the *model component*, specifying the interactions between the various species. In the context of this paper, the individual entities are the robots, and the model component defines how they interact.

The syntax of Bio-PEPA components is defined as:

$$S ::= (\alpha, \kappa)\ \mathtt{op}\ S \mid S + S \mid C \quad \text{with } \mathtt{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot \quad \text{and } P ::= P \underset{\mathcal{L}}{\bowtie} P \mid S(x)$$

where S is a *species component* and P is a *model component*.

The *prefix combinator* "op" in the prefix term $(\alpha, \kappa)\ \mathtt{op}\ S$ represents the impact that action α has on species S. Specifically, $\downarrow$ indicates that the number of entities of species S reduces when α occurs, and $\uparrow$ indicates that this number increases. The amount of the change is defined by the coefficient κ. This coefficient captures the multiples of an entity involved in an occurring action. We will see an example of its use in the next section. The default value of κ is 1, in which case we simply write α instead of (α, κ). Action durations are assumed to be random variables with negative exponential distributions, characterised by their *rates*. The rate of action α is defined by a so called functional rate or kinetics rate. Action rates are defined in the context section of a Bio-PEPA specification.

The symbol $\oplus$ denotes an *activator*, $\ominus$ an *inhibitor* and $\odot$ a generic *modifier*, all of which play a role in an action without being produced or consumed and have a defined meaning in the biochemical context. The operator "+" expresses the choice between possible actions, and the constant C is defined by the equation $C{=}S$. The process $P \underset{\mathcal{L}}{\bowtie} Q$ denotes synchronisation between components P and Q, the set $\mathcal{L}$ determines those actions on which the components P and Q are forced to synchronise. The shorthand $P \overset{*}{\bowtie} Q$ denotes synchronisation on all actions that P and Q have in common. In $S(x)$, the parameter $x \in \mathbb{R}$ represents the initial amount of the species. A Bio-PEPA *system* with *locations* consists of a set of species components, a model component, and a context containing definitions of locations, functional/kinetics rates, parameters, etc.. The prefix term $(\alpha, \kappa)\ \mathtt{op}\ S@l$ is used to specify that the action is performed by S in location l.

Bio-PEPA is given a formal operational semantics [5] which is based on Continuous Time Markov Chains (CTMCs). It is supported by a suite of software tools which automatically process Bio-PEPA models and generate internal representations suitable for different types of analysis [5, 14]. These tools include mappings from Bio-PEPA to differential equations (ODE) supporting a fluid flow approximation [15], stochastic simulation models [16], CTMCs with levels [17] and PRISM models [18] amenable to statistical model-checking. Consistency of the analyses is supported by a rich theory including process algebra, and the relationships between CTMCs and ODE.

4 Collective decision-making: a Bio-PEPA specification

In this paper, we analyse a collective decision-making system originally proposed by Montes de Oca et al. [8]. The task of the robots is to transport objects from a *start* area to a *goal* area. The objects to transport are too heavy for a single robot, thus the robots have to form groups of three in order to transport a single object. There are two possible paths between the start and the goal area and the robots can choose between the two. This is similar to what ants do in the well known double bridge experiment with the difference that ants use pheromones while in our setup robots use voting.

Each individual robot has a preferred path. When a group of three robots is formed in the start area, the robots choose the path that is preferred by the majority of them. The chosen path becomes the preferred one for all the robots in the group. More details are given in Sections 4.1. An analysis of the system is presented in Montes de Oca et al. [8] and in Scheidler [9].

This collective decision-making system is a good benchmark for testing Bio-PEPA since it displays two important aspects of swarm robotics: cooperation and space-time characteristics. Cooperation can be direct and indirect: the robots cooperate directly to transport the objects, and indirectly to select a path via the dynamics of their preferences. Space-time characteristics are displayed in the voting process itself, as it involves only the robots that are in the start area at a given time, and in the fact that the collective decision-making process depends on the time taken to navigate over the two different paths.

4.1 The Bio-PEPA specification

In the remaining part of this section we present the Bio-PEPA specification of the system. As shown in Fig. 1, the system is described by eight Bio-PEPA *locations*: two boundary locations, `start` and `goal`; two choice locations, `A` and `B`, where the robots decide which path to take; and two locations for each path, `L1` and `L2` for the long path and `S1` and `S2` for the short one. We also define a set of Bio-PEPA *species* to specify the behaviour of the robots. For example in `start` we distinguish two species of robots: those that last time returned via the short path, denoted as *Robo_start_fromS*, and those that returned via the long path, denoted as *Robo_start_fromL*. In the sequel we will refer to these two groups also

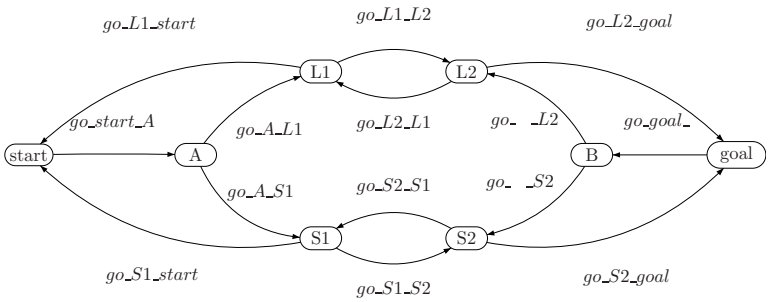


Fig. 1. Locations and transitions of robots in the Bio-PEPA specification.

as the *S-population* and the *L-population*, respectively. Similarly, other locations contain populations of teams of robots that move in the direction from the start area to the goal area and those that move in the opposite direction. For example, in location `S1` we can have *Teams_S1_StoG* and *Teams_S1_GtoS*, where *StoG* denotes the direction from the start area to the goal area and *GtoS* the opposite direction.

The Bio-PEPA fragment below specifies the behaviour of a robot. Robots leave the start area in groups of three. Each group is randomly composed by either three robots from the S-population, three from the L-population or two from S and one from L or two from L and one from S. These combinations are modelled as four different actions: *allS*, *allL*, *S2L1* and *S1L2*. In Bio-PEPA the formation of teams of robots is modelled by the coefficient that indicates how many entities are involved in an action. For example, upon action *allS* three robots of the S-population leave `start` (indicated by $(allS, 3)\downarrow$), to form an additional *team* of robots in choice point `A` (indicated by $(allS, 1)\uparrow$ in $Teams\_A\_S$) which is ready to take the short path when the team continues its journey towards the goal area (population $Teams\_A\_S@A$). Since action *allS* is shared between the species components *Robo_start_fromS* and *Teams_A_S* this movement occurs simultaneously with the rate of action *allS* that will be defined later on.

$$\begin{aligned} Robo\_start\_fromS = {} & (allS, 3)\downarrow Robo\_start\_fromS@start + \\ & (S2L1, 2)\downarrow Robo\_start\_fromS@start + \\ & (S1L2, 1)\downarrow Robo\_start\_fromS@start + \\ & (go\_S1\_start, 3)\uparrow Robo\_start\_fromS@start; \end{aligned}$$

$$\begin{aligned} Teams\_A\_S = {} & (allS, 1)\uparrow Teams\_A\_S@A + \\ & (S2L1, 1)\uparrow Teams\_A\_S@A + \\ & go\_A\_S1\downarrow Teams\_A\_S@A; \end{aligned}$$

In a similar way, upon action $S2L1$, which is present in three components (*Robo_start_fromS*, *Teams_A_S* and *Robo_start_fromL*, the latter is not shown), all three components synchronize, resulting in two robots from the S-population and one from the L-population leaving the start area and forming at the same time 1 new team in choice point A in the population $Teams\_A\_S$. The synchronization pattern of the components is given by the model component shown later

on. The excerpt above only shows the behaviour of teams voting for the short path. The behaviour of those voting for the long path is similar and omitted for reasons of space. For the same reason also the behaviour of teams moving between different locations is not shown.

The actions denoting groups of robots leaving the start area need to occur with appropriate rates. For example, a group of three robots all from the S-population has a probability of $\frac{(RSS)}{(RSS)+(RSL)} * \frac{(RSS-1)}{(RSS-1)+(RSL)} * \frac{(RSS-2)}{(RSS-2)+(RSL)}$ times the rate of leaving the start area, where RSL (RSS resp.) abbreviates $Robo\_start\_fromL@start$ ($Robo\_start\_fromS@start$ resp.). A similar probability can be defined for a group of three from the L-population.

When considering mixed groups also the order in which the elements are extracted from the population in the start area is of influence. This is particularly true when relatively small populations of robots are considered. For example, the probability to extract two robots from the S-population in the start area and then one from the L-population is:

$$pSSL = \frac{(RSS)}{(RSS)+(RSL)} * \frac{(RSS-1)}{(RSS-1)+(RSL)} * \frac{(RSL)}{(RSS-2)+(RSL)}$$

Similarly we define probabilities for $pSLS$, $pLSS$, $pLLS$, $pLSL$ and $pSLL$. The rates of actions $S2L1$ and $S1L2$ can now be defined as $(pSSL+pSLS+pLSS)*move$ and $(pSLL+pLSL+pLLS)*move$, respectively. Note that the sum of these six probabilities and the probability of the combination of three S or three L is 1. So the total rate at which groups of robots leave the start area is constant and given by the parameter 'move'. The rate at which groups move from A to $S1$ and to $L1$ is also dependent on the number of groups present in A and are $walk\_normal * Teams\_A\_S@A$ and $walk\_normal * Teams\_A\_L@A$, respectively. The rate parameter $walk\_normal$ specifies the time it takes a group of robots to move from choice-point A to the first section of a path.

Finally, the overall system definition shows the initial size of robot populations in each location. The overall robot behaviour is defined using multi-part synchronization on shared actions:

$$\begin{array}{l} Robo\_start\_fromS@start(SS) \underset{*}{\bowtie} Robo\_start\_fromL@start(SL) \underset{*}{\bowtie} \\ Teams\_A\_S@A(0) \underset{*}{\bowtie} Teams\_A\_L@A(0) \underset{*}{\bowtie} \\ Teams\_S1\_StoG@S1(0) \underset{*}{\bowtie} Teams\_S1\_GtoS@S1(0) \underset{*}{\bowtie} \\ Teams\_S2\_StoG@S2(0) \underset{*}{\bowtie} Teams\_S2\_GtoS@S2(0) \underset{*}{\bowtie} \\ Teams\_L1\_StoG@L1(0) \underset{*}{\bowtie} Teams\_L1\_GtoS@L1(0) \underset{*}{\bowtie} \\ Teams\_L2\_StoG@L2(0) \underset{*}{\bowtie} Teams\_L2\_GtoS@L2(0) \underset{*}{\bowtie} \\ Teams\_goal\_fromS@goal(0) \underset{*}{\bowtie} Teams\_goal\_fromL@goal(0) \underset{*}{\bowtie} \\ Teams\_B\_fromS@B(0) \underset{*}{\bowtie} Teams\_B\_fromL@B(0) \end{array}$$

where SS (resp. SL) is the initial size of the S (resp. L)-population in the start area. There is a further issue to consider which is the way to model the length of the paths. This can be done in two ways. The first is to model each path by two sections, as illustrated above, and vary the time it takes teams to traverse these sections by choosing a different rate for the movement between sections on the short and the long path. However, as also discussed in [8], this model

has the disadvantage that the duration of path traversal is essentially modelled by a short series of exponential distributions which in general approximates the average duration well, but not the variability. It therefore does not reflect very well realistic robot behaviour.

An alternative is to choose the same rate for each section and to vary the number of sections on each path to model their difference in length. This way the traversal time of a path is modelled by a sequence of say m exponentially distributed random variables with rate λ, also known as an Erlang distribution, using the well-known method of stages [19] (p. 119).[2] We model the two paths of the environment with 8 S-sections and 15 L-sections. Each section takes, on average, ten time units to traverse. This is modelled in the system by defining the rate $walk\_normal = 0.1$. Considering also the movements from the choice points to the path and those from the path to the start area and the goal area, in this way the short path takes on average 100 time units to traverse, and the long one 170. This is comparable to the latency periods used in [8] (end of Section 4).

5 Analysis

For the analysis in this section we consider a Bio-PEPA voting specification with a population of 32 robots. In [8] the analysis results make reference to the number of teams, k, that are active in the system at any time. We specify this in Bio-PEPA by making sure that at any time at least min_start robots are in the start area, corresponding to $k = (32 - min\_start)/3$.[3] We furthermore consider the following parameters for the specification: $N = 32$, of which initially $SS = 16$ and $SL = 16$, $move = 0.28$, $walk\_normal = 0.1$.

In the following we illustrate three different forms of analysis of the same Bio-PEPA specification and validate their results with those from the literature [8].

5.1 Stochastic simulation

The first kind of analysis uses stochastic simulation to check the average number of active teams in the system over time for different assumptions on the minimal number of robots that remain in the start area. Fig. 2 presents two stochastic simulation results (average over 10 simulation runs) for $min\_start = 5$ (Fig. 2 left) and $min\_start = 2$ (Fig. 2 right). The figure shows that the number of active teams on the paths quickly increases to 9 (resp. 10) and then stabilizes at that level. This means that the rate at which robots leave the start area, i.e. $move = 0.28$, is sufficiently high to quickly reach a situation with the desired

[2] For m going to infinite, an Erlang distribution $[m, \lambda]$ converges to a normal distribution with mean m/λ and variance m/λ^2. So, in general, the larger is m, the better the Erlang distribution $[m, \lambda]$ approximates a normal distribution.

[3] In Bio-PEPA, one can make use of a predefined function H which takes a rate as argument. If the rate is zero, H returns zero, otherwise it returns 1. To guarantee a minimum number, min_start, in the start area, the rate of, e.g., action $S2L1$ can then be defined as: $S2L1 = (pSSL + pSLS + pLSS) * move * H((RSS + RSL) - min\_start)$.

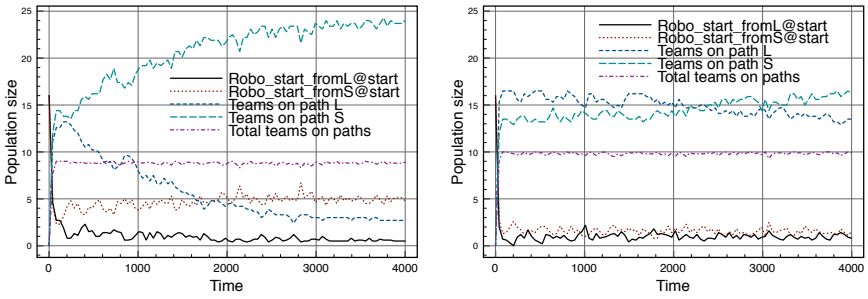


Fig. 2. Number of active teams for $min\_start = 5$ (left) and $min\_start = 2$ (right).

number of active teams. This makes it possible to compare the results of this analysis with the results obtained with the physics-based simulation and Monte Carlo simulation as reported in [8] which will be discussed later on. The figures show the number of robots present in the start area over time, both those in the S-population and in the L-population, and the number of teams on each path.

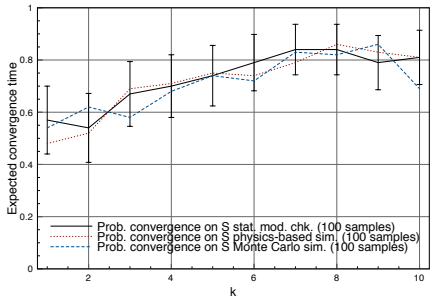
5.2 Statistical model checking

Another way to analyse the system is via statistical model checking provided, for example, by the model-checker PRISM [18]. In particular, the Bio-PEPA specification can be exported automatically to the PRISM input language. Statistical model checking is an analysis method in which a logical formula, formalizing a particular property of the system, can be automatically checked against a set of randomly generated simulation runs of a model of the system via statistical analysis. For example, if we denote convergence on the short path by the shorthand "Convergence_on_S", and convergence on the long path by "Convergence_on_L", the statement "what is the probability that the system did not converge on the long path until it converges on the short path" can be expressed in the logic CSL [20, 12] as:

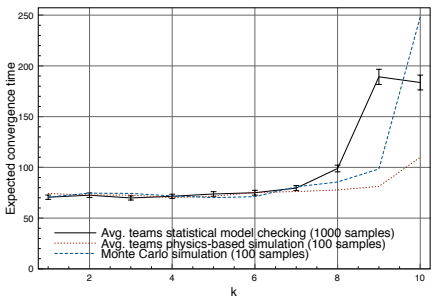
$$P =? \; [!\text{“}Convergence\_on\_L\text{”} \; U \; \text{“}Convergence\_on\_S\text{”}] \tag{1}$$

where $P =?$ is used to compute a probability, ! stands for negation and U reads as "until". "Convergence_on_S" can be defined as the situation in which all the 32 robots are either in a team on the short path or in the S-population in the start area or at the goal area. "Convergence_on_L" can be defined similarly, but requiring that the above sum is equal to 0 instead of 32.

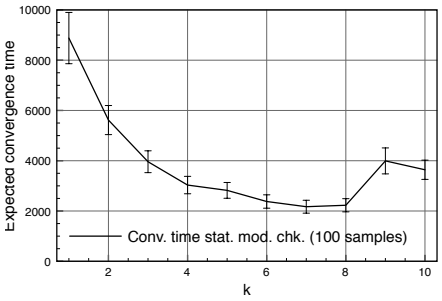
In a similar way, the model can be analysed to obtain the expected number of team formations and the expected time until convergence. For the number of team formations, one needs to count the number of times the actions 'go_A_S1' and 'go_A_L1' occur until convergence happens. Let us assume that the variable *teams* accumulates the number of teams formed until convergence, and the variable *total_time* the total time that passed until convergence in the various



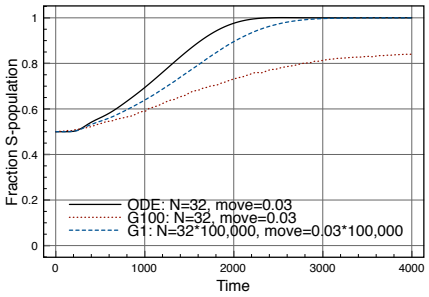
(a) Probability of convergence on the short path (100 samples)



(b) Expected number of team formations until convergence



(c) Expected convergence time (100 samples)



(d) Fraction in S-population

Fig. 3. Results with Erlang distribution of path length.

simulation runs.[4] The question "what is the expected number of teams formed until eventually convergence has taken place on the short or the long path" can then be answered by statistical model-checking with the logical reward formula:

$$R\{\text{“}teams\text{”}\} =? \ [F \ (\text{“}\_Convergence\_on\_S\text{”} | \text{“}\_Convergence\_on\_L\text{”})] \qquad (2)$$

where F reads as "eventually", $|$ denotes logical disjunction and $R =?$ is used to compute the expected value, commonly called 'reward', of specific events. A similar analysis using the same formula, but substituting *teams* with *total_time*, gives the expected time until convergence.

The following analyses have been based on 100 random samples and a confidence level of 99%, except where explicitly indicated. In the figures the confidence intervals are shown as error-bars.

Figure 3 (a), (b) and (c) show the result of statistical model checking of the above formulas for k active teams, where $k = \{1, 2, ..., 10\}$. In particular, Fig. 3(a)

[4] In terms of Markov theory such 'counting' is defined by *reward structures*. In statistical model-checking these numbers are used in the statistical analysis of the generated simulation runs.

shows the probability of convergence on the short path (i.e. Formula (1)). The data are compared to those obtained via physics-based simulation and Monte Carlo simulation of the same case-study reported in [8]. The latter are close to the results obtained with the Bio-PEPA specification and well within the error-margins. Fig. 3(b) shows results on the expected number of team formations until convergence on the short or long path (i.e. Formula (2)). The data correspond very well for k from 1 to 7, but diverge for higher values of k. Note that the data obtained via statistical model checking in Fig. 3(b) is based on 1000 random samples and a confidence level of 99%. Fig. 3(c) shows the expected convergence time. No data from the literature concerning this aspect is available for comparison. The total model checking time to produce the data in Fig. 3(a) was ca. 10 minutes, those in Fig. 3(b) ca. 48 minutes and those in Fig. 3(c) ca. 5 minutes. Due to space limits we limit our analysis to the shown properties. However, many other interesting properties of the system could be analysed this way. For example it can also be shown that for any value of k from 1 to 10 the probability that convergence occurs is equal to 1.

5.3 Fluid flow analysis

The third kind of analysis is a fluid flow approximation or numerical analysis of the ODE underlying the Bio-PEPA specification. Based on the Bio-PEPA syntax, the underlying ODE model can be generated automatically and in a systematic way [15]. This provides yet another view on the behavioural aspects of the system. One can, for example, explore numerically the sensitivity of the system to initial values and discover stationary points and other aspects related to stability analysis. As an example, here we show the relation between stochastic simulation and fluid approximation results.

In Fig. 3(d) a fluid flow analysis (ODE) is shown of the total fraction of robots in the S-population over time, i.e. both those present in the start area and those in the teams.[5] Clearly, the fluid approximation predicts that the system converges in 100% of the cases to the short path. Stochastic simulation over 100 independent runs (G100) shows that such convergence happens only in 85% of the cases, which corresponds to what we found with statistical model checking for a comparable value of k (see Fig. 3(a)). The difference can be explained by the small size of the population. The point is that for small populations, the probability that the system 'accidentally' converges on the long path is relatively high. For large populations, such a probability tends to zero. In fact, it is well-known that, when the specification satisfies certain scaling conditions, scaling the population (and in this case also the rate at which robots leave the start area)

[5] To avoid possible problems with discontinuities in the model, the fluid flow analyses have been performed on a Bio-PEPA specification in which the construction with the H-function was substituted by an appropriate value of the parameter *move* to regulate the average number of teams. To approximate a situation with on average $k = 7$, the rate at which robots leave the start area has been set to 0.03, i.e. $move = 0.03$.

with a large number, say 100,000, single stochastic simulation trajectories start to approximate the deterministic ODE solution (see [21]). This phenomenon can be observed in Fig. 3(d) from the curve labelled G1. This insight provides a way to interpret results obtained with fluid approximation.

6 Conclusions

Bio-PEPA [5] is a process algebraic language originally developed for the stochastic modelling and analysis of biochemical systems. In this paper we used Bio-PEPA to specify and analyse a robot swarm decision-making behaviour, originally presented in [8]. We showed that with Bio-PEPA issues relevant to SRS modelling can be addressed at the microscopic level. Among these issues are: robot team-formation, voting, spatial and temporal aspects, species of robots with particular behavioural characteristics, and direct and indirect interaction. At the same time, a single microscopic Bio-PEPA specification of the system lends itself to a variety of analyses methods such as stochastic simulation, fluid flow (ODE) approximation and statistical (stochastic) model checking. This provides an efficient way to obtain different views of the system behaviour, while preserving their mutual consistency. This consistency is due to the formal semantics underlying the Bio-PEPA language and analysis framework. The results were shown to be largely comparable with those obtained in [8] via physics-based simulation and Monte Carlo simulation of the same case study. In future work we will extend Bio-PEPA with more sophisticated spatial concepts and facilities to explore more easily non-linear behavioural aspects using numerical techniques. Our goal is to extend Bio-PEPA to ease the modelling and analysis process of SRS. We believe that this could facilitate the more widespread uptake of modelling and analysis in swarm robotics.

Acknowledgements. The research leading to the results presented in this paper has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n 246939, and by the EU project ASCENS, 257414. Manuele Brambilla, Mauro Birattari and Marco Dorigo acknowledge support from the F.R.S.-FNRS of Belgium's Wallonia-Brussels Federation. The authors would like to thank Stephen Gilmore and Alan Clark (Edinburgh University) for their help with the Bio-PEPA tool suite and templates.

References

1. Şahin, E.: Swarm robotics: from sources of inspiration to domains of application. In: Swarm Robotics. Volume 3342 of LNCS. Springer, Heidelberg (2005) 10–20
2. Lerman, K., Martinoli, A., Galstyan, A.: A review of probabilistic macroscopic models for swarm robotic systems. In: Swarm robotics. Volume 3342 of LNCS. Springer, Heidelberg (2005) 143–152
3. Dixon, C., Winfield, A., Fisher, M.: Towards temporal verification of emergent behaviours in swarm robotic systems. In: Towards Autonomous Robotic Systems. Volume 6856 of LNCS. Springer, Heidelberg (2011) 336–347

4. Konur, S., Dixon, C., Fisher, M.: Analysing robot swarm behaviour via probabilistic model checking. Robotics and Autonomous Systems **60**(2) (2012) 199 – 213
5. Ciocchetta, F., Hillston, J.: Bio-PEPA: A framework for the modelling and analysis of biological systems. TCS **410**(33-34) (2009) 3065–3084
6. Massink, M., Latella, D., Bracciali, A., Harrison, M., Hillston, J.: Scalable context-dependent analysis of emergency egress models. Formal Aspects of Computing (2011) 1–36 In press.
7. Massink, M., Latella, D., Bracciali, A., Hillston, J.: Modelling non-linear crowd dynamics in Bio-PEPA. In: Fundamental Approaches to Software Engineering. Volume 6603 of LNCS. Springer, Heidelberg (2011) 96–110
8. Montes de Oca, M.A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., Dorigo, M.: Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making. Swarm Intelligence **5**(3–4) (2011) 305–327
9. Scheidler, A.: Dynamics of majority rule with differential latencies. Phys. Rev. E **83** (Mar 2011) 031116
10. Ijspeert, A., Martinoli, A., Billard, A., Gambardella, L.M.: Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. Autonomous Robots **11** (2001) 149–171
11. Martinoli, A., Easton, K., Agassounon, W.: Modeling swarm robotic systems: a case study in collaborative distributed manipulation. The International Journal of Robotics Research **23**(4-5) (2004) 415–436
12. Baier, C., Katoen, J.P., Hermanns, H.: Approximate Symbolic Model Checking of Continuous-Time Markov Chains. In: Concur '99. Volume 1664 of LNCS., Springer, Berlin, Heidelberg (1999) 146–162
13. Brambilla, M., Pinciroli, C., Birattari, M., Dorigo, M.: Property-driven design for swarm robotics. In: Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012), IFAAMAS (2012) In press.
14. Ciocchetta, F., Duguid, A., Gilmore, S., Guerriero, M.L., J., H.: The Bio-PEPA Tool Suite. In: Proceedings of the 6th International Conference on Quantitative Evaluation of SysTems (QEST 2009), Washington, DC, USA, IEEE Computer Society (2009) 309–310
15. Hillston, J.: Fluid flow approximation of PEPA models. In: Proceedings of the 2th International Conference on Quantitative Evaluation of SysTems (QEST 2005), Washington, DC, USA, IEEE Computer Society (2005) 33–43
16. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry **81**(25) (1977) 2340–2361
17. Ciocchetta, F., Hillston, J.: Bio-PEPA: An extension of the process algebra PEPA for biochemical networks. ENTCS **194**(3) (2008) 103–117
18. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Proc. 23rd International Conference on Computer Aided Verification (CAV'11). Volume 6806 of LNCS., Springer, Heidelberg (2011) 585–591
19. Kleinrock, L.: Queueing Systems. Volume 1: Theory. Wiley, New York, NY, USA (1975)
20. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model checking Continuous Time Markov Chains. ACM Transactions on Computational Logic **1**(1) (2000) 162–170
21. Kurtz, T.: Solutions of ordinary differential equations as limits of pure jump Markov processes. Journal of Applied Probability **7** (1970) 49–58