

Decentralized Control and Interactive Design Methods for Large-Scale Heterogeneous Self-Organizing Swarms

Hiroki Sayama

Department of Bioengineering, Binghamton University
State University of New York, Binghamton NY 13902-6000, USA
sayama@binghamton.edu

Abstract. We present new methods of decentralized control and interactive design for artificial swarms of a large number of agents that can spontaneously organize and maintain non-trivial heterogeneous formations. Our model assumes no elaborate sensing, computation, or communication capabilities for each agent; the self-organization is achieved solely by simple kinetic interactions among agents. Specifications of the final formations are indirectly and implicitly woven into a list of different kinetic parameter settings and their proportions, which would be hard to obtain with a conventional top-down design method but may be designed heuristically through interactive design processes.

1 Introduction

Research on decentralized control mechanisms of artificial swarms has recently attracted an increasing attention because of its theoretical importance for understanding self-organization and emergence in complex biological systems [1] and its practical utility for creating distributed, robust, adaptive artifacts such as biomimetic algorithms for optimization problems [2, 3] and robotic systems for collective task achievement [4]. The latter merit manifests itself particularly in engineering applications, as self-organizing swarms may be substantially more flexible and less vulnerable to errors or attacks than conventional systems controlled by central units and hierarchical information flows.

A major technical challenge in this field is how to program and direct decentralized systems to a desired end. A number of different techniques have been proposed and tested to address this problem, e.g., by assuming hormonal communication signals in the environment [5, 6], by embedding top-down designed finite-state machines in each agent [7], by evolving neural network controllers of agents [8, 9], or by probabilistic control of aggregation patterns through kinetic parameter variations [10, 11]. Most of these studies (except for the last ones) demonstrated only with small swarm populations in the order of tens, and all of them used homogeneous agents only. Heterogeneous swarms have been the subject of investigation in recent studies [12, 13]; however, the focus of research

is largely on the task allocation among different types of agents, with limited attention paid to the dynamic structure of swarm populations themselves.

Here we consider dynamic self-organization of spatio-temporal patterns within a large-scale, heterogeneous swarm population, and present new methods of decentralized control and interactive design with which artificial swarms of a large number of agents (in the order of hundreds or thousands) can spontaneously organize and maintain non-trivial heterogeneous formations. This work is part of our ongoing *Swarm Chemistry* project [14], which aims to develop a novel theoretical framework for artificial chemistry research [15]. Our model is unique in that it assumes no elaborate sensing, computation, or communication capabilities for each agent. Rather, the self-organization is achieved solely by simple kinetic interactions among agents, as was originally modeled in Reynolds’ Boids [16], and more recently, physically demonstrated through the attraction-repulsion interactions of small magnetized disks by Grzybowski et al. [17].

Specifications of the final formations of swarms in Swarm Chemistry are indirectly and implicitly woven into a list of different kinetic parameter settings and their proportions, called a *recipe*, which would be hard to obtain with a conventional top-down design method but may be designed heuristically through interactive design processes. This paper presents a brief summary of our model and some preliminary results obtained so far.

2 Model

A swarm population in Swarm Chemistry consists of a number of simple, semi-autonomous agents. They can move in a two-dimensional continuous space, perceive positions and velocities of other agents within its local perception range, and change its velocity in discrete time steps according to the following kinetic rules (adopted and modified from the rules in Reynolds’ Boids system [16]):

- If there are no local agents within its perception range, steer randomly (**Straying**).
- Otherwise:
 - Steer to move toward the average position of local agents (**Cohesion**).
 - Steer towards the average velocity of local agents (**Alignment**).
 - Steer to avoid collision with local agents (**Separation**).
 - Steer randomly with a given probability (**Whim**).
- Approximate its speed to its own normal speed (**Pace keeping**).

These rules are implemented as shown in Algorithm 1. Kinetic parameters used in this algorithm are listed and explained in Table 1.

Each agent is assigned with its own kinetic parameter settings. We call a collection of identical agents that share the same kinetic parameter settings a *chemical species* (or just *species* in short). *Reaction* is defined as a process in which two or more species or their compounds are blended together and a new spatio-temporal pattern emerges through the kinetic interaction between different chemical species (Fig. 1).

Algorithm 1 The algorithm used to simulate the behavior of agents. \mathbf{x}_i , \mathbf{v}_i and \mathbf{v}'_i are the location, the current velocity, and the next velocity of the i -th agent, respectively. \mathbf{a} is a local variable temporarily used to represent an acceleration. r and $r_{\pm p}$ represent random numbers taken from $[0, 1]$ and $[-p, +p]$, respectively.

```

1: for all  $i \in \text{agents}$  do
2:    $N \leftarrow \{j \neq i \text{ that satisfies } |\mathbf{x}_j - \mathbf{x}_i| < R^i\}$ 
   // Finding other agents within its local perception range
3:   if  $|N| = 0$  then
4:      $\mathbf{a} \leftarrow (r_{\pm 5}, r_{\pm 5})$  // Straying
5:   else
6:      $\langle \mathbf{x} \rangle \leftarrow \sum_{j \in N} \mathbf{x}_j / |N|$  // Calculating the average position of local agents
7:      $\langle \mathbf{v} \rangle \leftarrow \sum_{j \in N} \mathbf{v}_j / |N|$  // Calculating the average velocity of local agents
8:      $\mathbf{a} \leftarrow c_1^i(\langle \mathbf{x} \rangle - \mathbf{x}_i) + c_2^i(\langle \mathbf{v} \rangle - \mathbf{v}_i) + c_3^i \sum_{j \in N} (\mathbf{x}_i - \mathbf{x}_j) / |\mathbf{x}_i - \mathbf{x}_j|^2$ 
     // Cohesion, alignment and separation
9:     if  $r < c_4^i$  then
10:       $\mathbf{a} \leftarrow \mathbf{a} + (r_{\pm 5}, r_{\pm 5})$  // Whim
11:    end if
12:  end if
13:   $\mathbf{v}'_i \leftarrow \mathbf{v}_i + \mathbf{a}$  // Acceleration
14:   $\mathbf{v}'_i \leftarrow \min(V_m^i / |\mathbf{v}'_i|, 1) \cdot \mathbf{v}'_i$  // Prohibiting overspeed
15:   $\mathbf{v}'_i \leftarrow c_5^i(V_n^i / |\mathbf{v}'_i| \cdot \mathbf{v}'_i) + (1 - c_5^i)\mathbf{v}'_i$  // Pace keeping
16: end for
17: for all  $i \in \text{agents}$  do
18:    $\mathbf{v}_i \leftarrow \mathbf{v}'_i$  // Updating velocity
19:    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$  // Updating location
20: end for

```

Note that individual agents remain exactly the same during this reaction process; only higher-order properties of the population as a whole (e.g., shapes, movement, etc.) may change. Such a nature of reactions in Swarm Chemistry may be a reasonable analog, at least conceptually, of what is going on in real chemistry, where the observed changes of chemical properties are due to the rearrangements of the participating elements and not due to the changes of the elements themselves. It should also be noted, however, that the kinetic interaction rules assumed in Swarm Chemistry have nothing to do with atomic or molecular interactions in real chemistry. The terminologies are borrowed from chemistry just for the purpose of making intuitive analogy.

3 Simulator

We developed a prototype of the interactive simulator of Swarm Chemistry using Java 2 SDK Standard Edition 1.5.0. It runs as a stand-alone application on any computer platform equipped with Java 2 Runtime Environment. Its source code is freely available from the author's website¹.

¹ <http://bingweb.binghamton.edu/~sayama/SwarmChemistry/>

Table 1. Kinetic parameters involved in the simulation of agent behavior. The maximum values are arbitrarily chosen for implementation purpose only. Unique values are assigned to these parameters for each agent i as its own dynamical properties.

Name	Min	Max	Meaning	Unit
R^i	0	300	Radius of local perception range	pixel
V_n^i	0	20	Normal speed	pixel step ⁻¹
V_m^i	0	40	Maximum speed	pixel step ⁻¹
c_1^i	0	1	Strength of cohesive force	step ⁻²
c_2^i	0	1	Strength of aligning force	step ⁻¹
c_3^i	0	100	Strength of separating force	pixel ² step ⁻²
c_4^i	0	0.5	Probability of random steering	—
c_5^i	0	1	Tendency of pace keeping	—

We used the *interactive evolution method* [18, 19] to enable interactive swarm design processes. Using our simulator, a human experimenter, or “alchemist”, can actively participate in the selection, perturbation and blending operations of swarm populations by subjectively selecting preferred swarm behaviors. Interactive evolution has already been applied to the designs of swarm behavior for scientific and educational purposes [10, 11]. Our work presented here is distinct from the earlier studies in that it evolves dynamic patterns of heterogeneous swarm populations.

We also note that conventional evolutionary computation techniques, such as genetic algorithms or genetic programming, could be implemented to acquire some types of swarm behaviors automatically. However, we did not choose to do so because setting an easily measurable metric for automated fitness evaluation would necessarily limit the diversity and novelty of potential outcomes.

Figure 2 shows a screen shot of our Swarm Chemistry simulator. Six different swarms are simultaneously simulated and demonstrated in their respective frames. A user can select up to two preferred swarms by clicking on the frames in which those swarms are simulated.

To produce a next generation of swarms, we used a unique set of evolutionary operators that are different from other typical genetic operators such as point mutation or crossover. If just one swarm is selected by the user, four perturbed swarms will be generated by repetitive random re-sampling of agents from the selected swarm up to a modified total population size (with $\pm 80\%$ possible variations). They will form the next set of swarms together with the original swarm preserved as is and a new swarm of randomly generated species, which is introduced to diversify the available options. On the other hand, if two swarms are selected by the user, three blended swarms, with agents’ initial positions completely randomized and mixed, will be generated at randomly determined ratios between the two selected swarms (ranging from 20:80 to 80:20), and they will be part of the next set of six swarms together with the original two as well as a new swarm of randomly generated species. In all cases, the maximum number of agents in a swarm is limited to 300. For the blending cases, the number of agents in a new swarm is set to the average of those in the two selected swarms. We

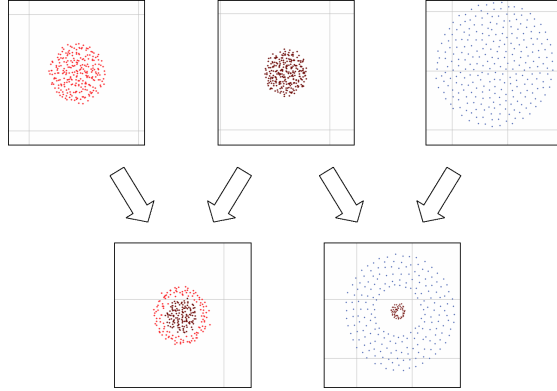


Fig. 1. Chemical reactions in Swarm Chemistry. Each dot represents one agent. Agents are colored by taking their $(c_1^i, c_2^i, c_3^i/100)$ values as their (R, G, B) values, respectively. Top: Three different chemical species (homogeneous populations made of identical agents). Bottom: Results of reactions (spatial patterns formed after blending two species). Blending different chemical species produce different patterns. Gray lines are references drawn at an interval of 300 pixels to show the scale of the image.

also implemented point mutation operators that would introduce random variations to each of the kinetic parameters, which was not included in the results presented in this paper in order to focus on the effects of chemical reactions in Swarm Chemistry.

By repeating this interactive selection process, the experimenter can explore a variety of dynamics and potentially create a novel, complex dynamic structure. This design cycle continues indefinitely until the application is manually quitted. The current version of our tool does not allow users to manually edit the kinetic parameter settings, which is planned to be implemented in the future versions.

4 Results

We have conducted preliminary exploration of possible dynamics in Swarm Chemistry and have found several characteristic outcomes of chemical reactions in this model world, which are summarized in what follows.

Spontaneous segregation: In most cases, agents spontaneously form clusters of the same species and segregate themselves from other chemical species when blended, even though they have no capability to distinguish themselves from other species. This is mainly because of the difference in their kinetic parameter settings that causes the difference in the local environments they “prefer”. It is generally very rare for two different species to remain mixed, with some exceptions where one or both of them are insensitive to local environmental conditions (this may happen if the perception range of agents is small).

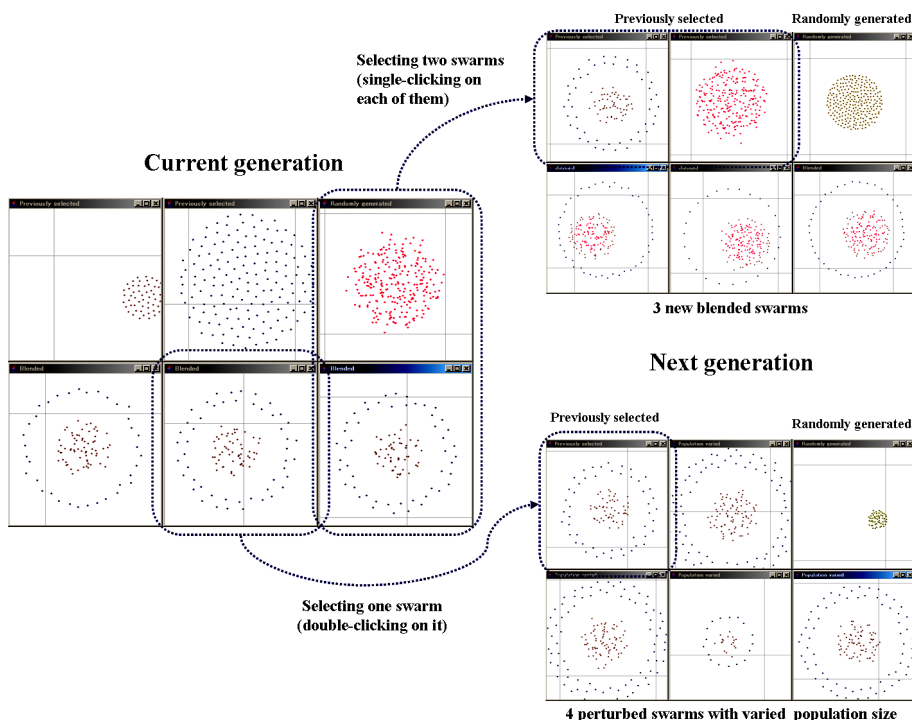


Fig. 2. Left: Screen shot of the Swarm Chemistry simulator. Six swarms are simultaneously simulated on a screen. The simulated space has no boundaries so that the distribution of agents could be infinitely large due to diffusion. The simulator dynamically rescales the viewing size so that the most significant cluster should always be tracked within the frame. Right: Results of selection operations in the Swarm Chemistry simulator. A user can select up to two swarms by clicking on the respective frames. If two swarms are selected, three new blended swarms will be generated at different ratios between the two (top). If just one swarm is selected, it will be replicated four times through random re-sampling with varied population sizes (bottom).

Production of movements: Reaction in Swarm Chemistry may produce a new macroscale dynamic movement of the swarm, which may be considered an example of the emergence of autonomous motion in prebiotic systems [20]. Movements typically arise when an asymmetric relationship is established between the two chemical species, one as a chaser and the other as an escaper. Such movements based on chasing may take a couple of different forms, as shown in Fig. 3.

Encapsulation and restriction of movements: Chemical reaction in Swarm Chemistry can also form multiple closed layers in a population, where one species is encapsulated in a dynamically formed membrane made of the other species (Fig. 1). This sometimes restricts the movement of originally mobile species (Fig. 4), creating an internal rotation or oscillation inside the structure.

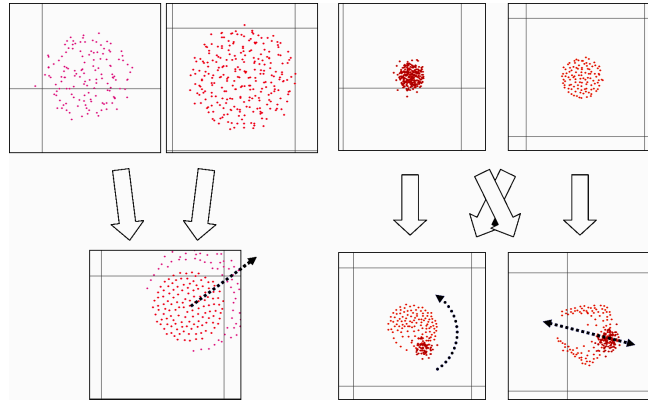


Fig. 3. Production of movements as a result of reaction in Swarm Chemistry. Left: Linear motion. Two originally immobile chemical species form a polarized cluster when blended, and their chasing/escaping behaviors create a linear motion of the entire swarm population. Middle: Rotation. The red cluster rotates around the orange cluster counterclockwise. Right: Linear oscillation. The red cluster is continuously attracted by the orange cloud and keeps moving back and forth through it. In this particular example, the last two movements arise from the same two chemical species. Their relative positions and proportions determine which movement will result.

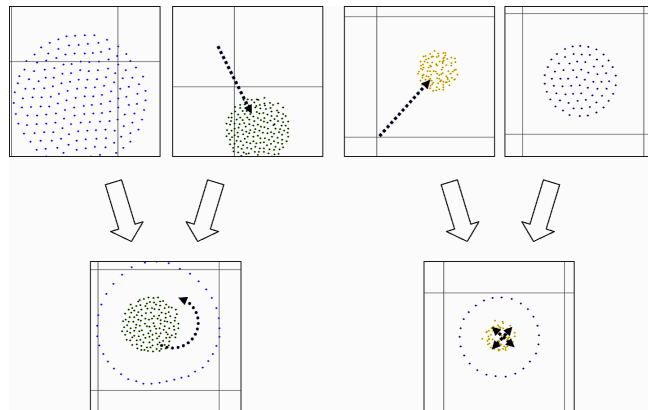


Fig. 4. Encapsulation of originally mobile species into an immobile membrane structure made of other species. The movement of the species are restricted and transformed into internal rotation (left) or oscillation (right).

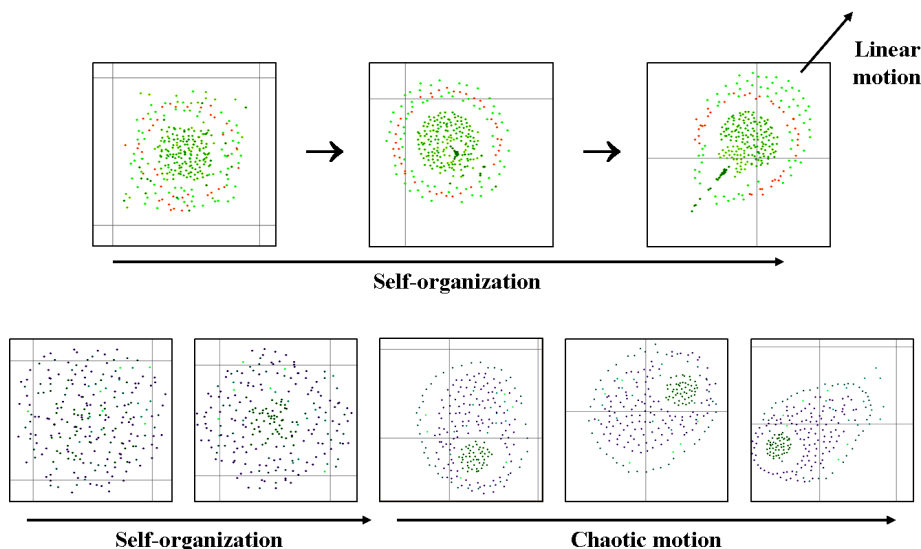


Fig. 5. Examples of complex structures made of several different chemical species, designed using our interactive simulator. The swarms self-organize from initially random states to a shape that looks like a horseshoe crab (top), or a biological cell-like structure that shows active chaotic movement after self-organization (bottom).

5 Interactive design of complex structures

Swarm Chemistry may be applied to the heuristic design of robust distributed systems. Our simulator enables one to interactively and incrementally build artificial chemical systems toward desirable structures and/or behaviors. Because the previously selected swarms (i.e., the best designs obtained so far) are always preserved and simulated on screen (Fig. 2), the designer can compare the new results of blending or variation with the previous achievement and decide whether or not the most recent changes should be incorporated into the design.

Sample products of such iterative design are shown in Fig. 5, where the designer tried to create some biological-looking structures. The first example self-organizes into a shape that resembles a horseshoe crab, and it actually moves toward the direction it heads at. The second example forms a biological cell-like structure, including nucleus and membrane, and shows a very chaotic, aggressive movement after its self-organization. The designs of these final products are specified in the form of a *recipe*, a list of the kind and the number of agents that describes the composition of the swarm population. What the designer is doing is to control the structure and the behavior of the product indirectly by modifying the contents of the recipe.

A remarkable feature of these products, and all other products that can be obtained in Swarm Chemistry, is that they are inherently *self-organizing*. Agents are initially mixed randomly, but they quickly find their own places to sit in and

collectively form the shape of the final product (Fig. 5). This self-organization process is found to be highly reproducible and robust to perturbations such as minor variations in system sizes and addition or removal of a small number of agents.

The shape of the final product must be indirectly woven into the recipe; however, there is neither an articulated blueprint nor a central controller that coordinates the arrangement of the agents. The entire process is driven solely by local kinetic interaction between agents. It would be difficult to predict the final outcome without carrying out explicit numerical simulation. While it is generally hard to manually design such a distributed self-organizing system, our approach with interactive, iterative heuristic design appears to be effective to reconcile the inconsistency between self-organization and manual design.

6 Conclusion

We introduced Swarm Chemistry and presented new methods for decentralized control and interactive design of spontaneously organizing and dynamically self-maintaining heterogeneous artificial swarms. We also developed a prototype of the interactive simulator and illustrated several characteristic phenomena frequently observed in the reaction processes in Swarm Chemistry, as well as its potential for heuristic design of robust, self-organizing distributed systems. We anticipate several practical applications of this work, including (1) introduction and dynamic control of self-organized heterogeneous swarms in particle swarm optimization, (2) distributed formation control of unmanned aerial vehicles (UAVs) in aerospace and/or military applications, and (3) interactive generation of aesthetically appealing dynamic patterns for artistic purposes.

Our model is unique in that agents we use are fairly simple. They have no lengthy self-regulating instructions like a genome in a biological cell; they always remain the same with no potential to differentiate. Also, the agents have no capability to identify or distinguish themselves from other species. They just look identical to each other, and therefore no sophisticated communication exists between them. This component simplicity would strengthen Swarm Chemistry as a model of distributed artificial systems, especially in view of real-world implementations where the simplicity of modules is of particular concern.

The Swarm Chemistry project is still at its launching stage, and there are a number of things yet to be done. Future to-dos include (a) analytical and numerical investigations of the properties of each chemical species and the effects of kinetic parameters on pattern formation, (b) exhaustive computational investigation of potential shapes and limitations of patterns made through the interactions of multiple species, (c) implementation and evaluation of dynamic self-replication and self-maintenance of complex patterns, and (d) hardware implementation and demonstration using mobile robotic modules.

References

1. Camazine, S. et al. (2001) *Self-Organization in Biological Systems*. Princeton Univ. Press.
2. Kennedy, J. & Eberhart, R. C. (2001) *Swarm Intelligence*. Morgan Kaufmann.
3. Engelbrecht, A. P. (2005) *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons.
4. Dorigo, M. et al. (2001-2005) *Swarm-bots Project*. <http://www.swarm-bots.org/> .
5. Dorigo, M. & Stützle, T. (2004) *Ant Colony Optimization*. MIT Press.
6. Shen, W.-M., Will, P., Galstyan, A. & Chuong, C.-M. (2004) Hormone-inspired self-organization and distributed control of robotic swarms. *Autonomous Robots* 17: 93-105.
7. Rodriguez, A. & Reggia, J. A. (2004) Extending self-organizing particle systems to problem solving. *Artificial Life* 10: 379-395.
8. Baldassarre, G., Parisi, D. & Nolfi, S. (2006) Distributed coordination of simulated robots on self-organization. *Artificial Life* 12: 289-311.
9. Groß, R., Bonani, M., Mondada, F. & Dorigo, M. (2006) Autonomous Self-assembly in a Swarm-bot. *Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, Springer, pp.314-322.
10. Kwong, H. & Jacob, C. (2003) Evolutionary exploration of dynamic swarm behaviour. *Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC 2003)*, IEEE, pp.367-374.
11. Sayama, H. (2006) Teaching emergence and evolution simultaneously through simulated breeding of artificial swarm behaviors. *Proceedings of the Sixth International Conference on Complex Systems (ICCS2006)*, NECSI. Available online at <http://necsi.org/events/iccs6/proceedings.html>.
12. Scheutz, M., Schermerhorn, P. & Bauer, P. (2005) The utility of heterogeneous swarms of simple UAVs with limited sensory capacity in detection and tracking tasks. *Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005)*, IEEE, pp.257-264.
13. Magg, S. & te Boekhorst, R. (2007) Pattern formation in homogeneous and heterogeneous swarms: Differences between versatile and specialized agents. *Proceedings of the First IEEE Symposium on Artificial Life (IEEE-Alife'07)*, IEEE, pp.311-316.
14. Sayama, H. (2007) Swarm chemistry. *Artificial Life*. Accepted pending revisions.
15. Dittrich, P., Ziegler, J., & Banzhaf, W. (2001) Artificial chemistries – A review. *Artificial Life*, 7, 225-275.
16. Reynolds, C. W. (1987) Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4): 25-34.
17. Grzybowski, B. A., Stone, H. A., & Whitesides, G. M. (2000) Dynamic self-assembly of magnetized, millimetre-sized objects rotating at a liquid-air interface. *Nature* 405, 1033-1036.
18. Unemi, T. (2003) Simulated breeding – a framework of breeding artifacts on the computer. *Kybernetes* 32: 203-220.
19. Rothlauf, F., et al., eds. (2006) *Applications of Evolutionary Computing: EvoWorkshops 2006*. Lecture Notes in Computer Science 3907, Springer. See papers of the EvoINTERACTION Workshop.
20. Ikegami, T. (2006) Emergence of autonomous motion. Keynote talk at the AlifeX Workshop on Simulation Models of Autonomous Systems. Workshop proceedings available online at http://www.alifex.org/program/wkshp_proceed.pdf.