Exploiting Morphological Metaphors for Self-Organization of Unmanned Aerial Vehicles

Kiwon Yeom

Human Systems Integration Division, NASA Ames Research Center, Moffett Field, CA 94035-0001, USA

Abstract: This paper exploits a multitude shape formation method of simplistic modular agents which can self-organize their positions. We introduce a mobile molecular agent with binary, triad, or polymer interactions acting among individual molecules, taking the responsibilities for embodying its structure and determining where further units can be attached and detached. The process of attachment and detachment of each molecular agent are induced by morphogenetic properties of molecular cells, and the shape formation is controlled by cellular automata. Development of morphological features such as the fine formation of particles, hollow particles, cracks in particles, and the evolution of the structure are used as illustrative examples. We also describe a analytic aspect of our model comparing with behavioral performances.

Kiwon Yeom. **Exploiting Morphological Metaphors for Self-Organization of Unmanned Aerial Vehicles**. *Life Sci J* 2013;10(3):992-1000] (ISSN:1097-8135). http://www.lifesciencesite.com. 144

Keywords: Self-organization, Federation of agents, Modular UAV agents.

1. Introduction

Self-organizable or reconfigurable flying agents (i.e., Unmanned Aerial Vehicles) are agents which can maintain the shape built from many uniformly independent and mobile modules. Each modular agent is able to have actuators, sensors, processing power, memory, and means of communicating to its neighbors. If agents are autonomously able to change the way which agents are connected with and maintain their own formation on a certain level, the agent is a self-reconfigurable or self-organizable agent and the process, which agents arrange their position and maintain their shape, is called as reconfiguration or self-organization. Through reconfiguration process, self-reconfigurable agents can change their shape and adapt to the specific task environment([1]-[9]).

In general, establishing a suitable self-organization mechanism for maintaining and controlling overall behaviors of each agent is not trivial. However, many biological systems in nature have presented solvable clues to such complicated problems [10], [11], [12], [13], [14], [15].

Biological organisms, which are able to accomplish their coherent, reliable and complex behavior through local information and interaction among of identically 'programmed' cells, have inspired researchers trying to solve these kinds of problems. In particular, the diffusion of chemicals among organisms and the behavioral characteristics, which can be driven by the locally sensed information of diffused chemicals ('morphogen gradients'), due to its simplicity, seems appropriate to apply user-defined cell organization or particle pattern formation[16], [17].

In this perspective, we are able to imagine the possibility to exploit multi-cellular based computational organisms, made up of millions of interacting autonomous agents, capable of assembling and dynamically re-assembling themselves into a variety of complex shapes.

We present an approach to automatically construct user-specified structures in three dimensions through inspiration from biological systems. Our model resembles morphogenesis in that molecular agents can make a network to manage the spatial formation via self-organization in an ensemble of agents. In particular, our goal is to study how and to which extent a group of simple autonomous molecular agents can be programmed to coordinate their respective movements and create variety of global shapes.

The key contribution here is to show how a variety of shapes (from regular to non- regular ones, also involving differentiation in agents) can be achieved even in the absence of those capabilities (e.g., global perception, distance and direction sensing) that are required by most other approaches.

2. Molecular Mobile Agent

The concept, elaborated in this work, represents the particle as an agglomerate of molecular cells with force interactions acting among the individual molecular cells. Such molecular cells could be of various types [18], [19], [20]. For example, they can represent the catalyst carrier particles, active catalyst particle sites, polymer particle generation phase(s), and even the gas if the particle is small enough to occupy the void space [21], [22], [23]. Here we restrict ourselves to only one type of molecular cell.

The swarm of molecular agents can be segregated into several parts of small molecules [24], [25]. The molecules can be of different types, such as the polymer particle phase and carrier particles. The reaction, transport, and mechanical deformation processes are considered to take place in the interconnected network of molecules (see Figure 2). For the sake of simplicity, all molecules have a spherical shape.

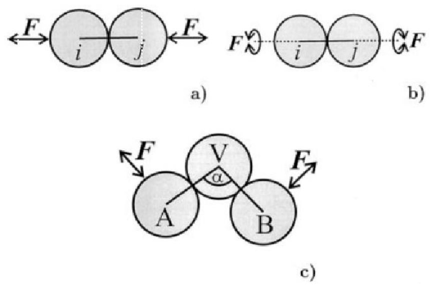


Fig. 1. Illustration of force interactions among molecules. (a) Resistance against 'push' and 'pull' (b) resistance against 'torsion' (c) resistance against 'change of bonding angle' (that is, ternary interaction).

Agglomerates of molecules could well represent particle morphologies other than the multigrain morphology because the molecules could partially interpenetrate or the connections among molecules could break [26], [27], [28].

3. Mechanical Force Model of Molecular Agent

Mechanical (force) interactions among molecules (see Figure 3) are calculated at each time step of dynamic simulation, and the translational movement of molecules is up- dated according to

$$\frac{dx_i}{dt} = v_i, \quad \frac{d(m_i v_i)}{dt} = \sum_j F_{ij}^{binary} + \sum_{j,k} F_{ijk}^{ternary}$$

(1)

where is time and xi, vi , and mi are position vectors of the center, velocity, and mass of the ith microelement, respectively [29], [30].

The summation of forces ∑Fij and ∑Fijk are carried out over all binary and ternary interactions between the ith molecule and its connected neighbors. The rotational movement of molecules displayed in Figure 1(b) is not considered herein for the sake of simplicity. The volume of the ith spherical molecule and its mass are

Vi=(4/3)πr3, mi=ρiVi (2)

where is the radius and ρi is the density of the molecule.

The evolution of the molecule volume and mass depends on its rate of growth dri /dt(the first order derivative), as follows

$$\frac{dV_i}{dt} = \frac{3V_i}{r_i}\frac{dr_i}{dt}, \quad \frac{dm_i}{dt} = \frac{3m_i}{r_i}\frac{dr_i}{dt} \qquad (3)$$

where the simplification of constant molecule density ρi is used throughout this article [31], [32], [33], [34].

The rate of growth of each particle depends on: (1) the amount of active sites inside the molecules and (2) the concentration of single molecules and other reactants at active sites. The rate of growth of the ith molecule is able to be described in a simple case of homo-polymerization which single molecules are combined with each other [18], described as

$$\frac{dm_i}{dt} = k m_{cat,i} c_i \qquad (4)$$

where cat, i is the mass of catalyst in the ith molecule and ci is the concentration of the monomer at the surface of the ith molecule [35], [36], [37].

The rate of growth of the particular molecule depends on the local concentration of monomer ci [see Eq.4]. Since the shape of polymer particle can evolve into an arbitrary shape, we define the center of the particle to be identical with the center of the gravity xCOG of the agglomerate of molecules. The characteristic radius of the particle R part is defined as the radius of the sphere with the center at xCOG, which contains95% of all molecules.

4. Self-Organizing Architecture

The proposed framework presupposes that each application is composed of one or more components as shown in Fig. 2. Each component has a mobile single-cellular structure (modular agent or motile cell), since it is self-comprised and self-roving. A collection of components is akin to pseudo-deformation in biology because such a combination can modify the structure of a group of components [38].

The framework provides interaction measures for migration of components. It controls the migration of components to other computers to accomplish particular requirements or tasks. The framework should be used to develop a unicellular application as a set of agent-based components for components or application, which is able to migrate to other devices and reorganize components while the application is running [39].

The movement of one component may affect other components. In other words, two components, which have different latencies in communication and velocities in movement and reorganization, are requested to simultaneously combine by a nearby computing device. For example, one is for the control of the combine hardware module and another is to networking synchronization module between the agents. Since each component has its own unique properties and control strategy, synchronizing a collection of components tends to be impossible over a distributed network environment.

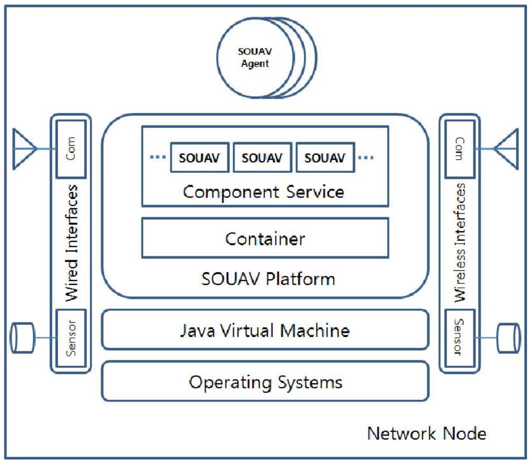


Fig. 2. SOUAV framework consists of modular components which are able to bind/unbind to other agents.

Therefore, components cannot efficiently coordinate with each other. Therefore, the framework ought to enable each component to explicitly specify its own restraints for migration and reorganization. A component has to migrate to one of the most suitable devices under their own controls and constraints to satisfy their requirements and tasks. However, it is not always possible to decide exactly which destination is the most suitable. For this reason, the proposed framework allows a component to simultaneously distribute its clones to multiple computing devices. The framework then selects the most suitable clone while the others naturally vanish.

5. Principal Attributes of SOUAV

Agents in the SOUAV framework are decentralized (Decentralization). There are no central entities to control and coordinate agents. Decentralization allows network applications to be scalable and simple by avoiding a single point of performance bottlenecks and failure, and any central coordination in deploying agents. Agents in the SOUAVareautonomous (Autonomy). Agents sense their local surrounding environments and they autonomously behave without any human intervention or from/to other agents, platforms.

Biological entities strive to seek and consume food for living (Energy Management).In SOUAV, agents store and expend energy for living. Each agent gains energy in exchange for performing its service to other agents and expends energy to use network and computing resources. Agents in the SOUAV framework are adaptive to dynamically changing environmental conditions (Adaptability). The adaptation is achieved from designing agent behavior policies to consider local environmental conditions. The abundance or scarcity of stored energy in agents affects their behaviors and triggers natural selection (Natural selection).

5.1. Behavior Process

In this section, the algorithmic aspects of the behavior selection mechanisms are visualized with the UML (Unified Modeling Language) sequence diagram.

The base class, migration schemes of components, and coordination strategy are described. The SOUAVframework provides the default implementation architecture which is automatically generated with basic class and implementation interfaces that can be observed in programming tools such as Visual C++, Net Beans, and IBM Architect.

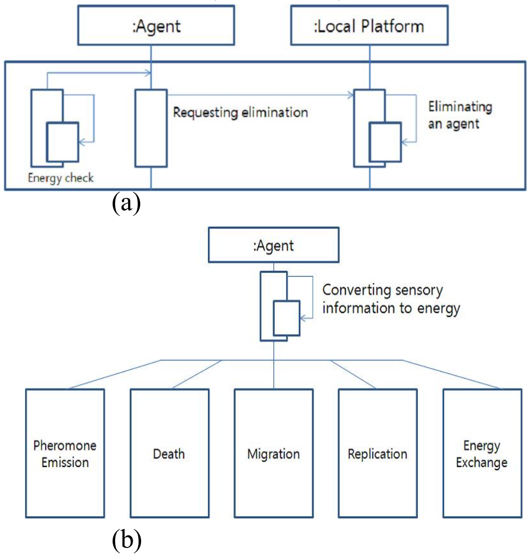


Fig. 3.(a) The basic class and behavior interfaces of the SOUAV agent. (b) Agent's death behavior.

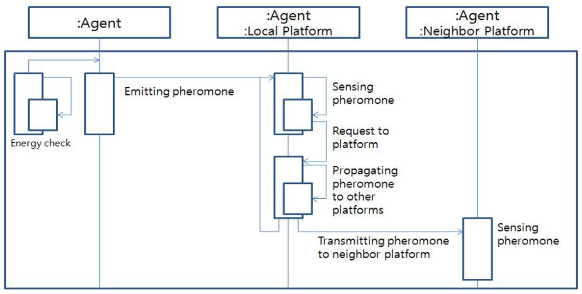
Fig. 2(a) shows the basic class and behavior interfaces. In SOUAVframework,an agent checks the energy level to determine the behavior. When the agent is generated at the first time, the agent has a certain amount of energy level. If the energy level of the agent becomes very low (i.e., below the death threshold), the agent is eliminated from the platform due to energy starvation (see Fig. 2(b)).An agent emits a pheromone (i.e., cytokine) if its energy level exceeds the threshold of pheromone emission. Agents continuously adjust their threshold value by CUSUM(cumulative sum control), which is a sequential analysis technique and typically used for monitoring change detection [40], of its energy level. When a pheromone is emitted on a platform, all the agents on the platform can sense it. It may stimulate their replication or migration behavior. Each pheromone has its own concentration (i.e., 1000) and fades away by lapse of time. The pheromone completely vanishes from the platform when the value becomes zero (see Fig. 4(a)).

$$S_m = \sum_{i=1}^{m} \bar{x}_i - \hat{\rho}_0 \quad (5)$$

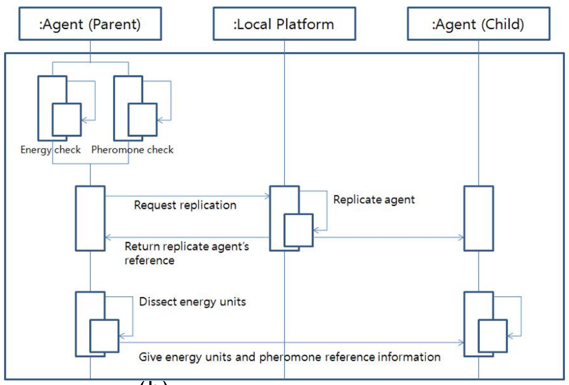
$$S_{hi}(i) = \max[0, S_{hi}(i-1) + x_i - \hat{\rho}_0 - k] \quad (6)$$

$$S_{lo}(i) = \max[0, S_{lo}(i-1) + \hat{\rho}_0 - k - x_i] \quad (7)$$

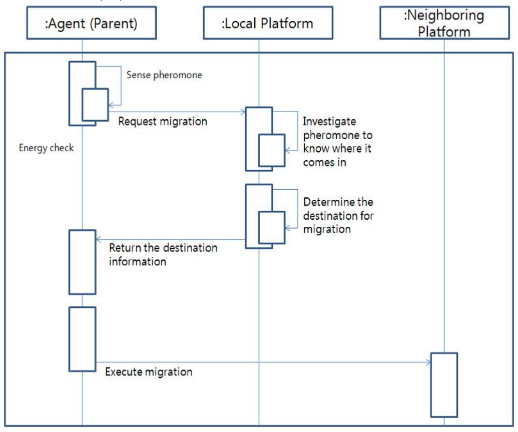
Where, m is the sample number, $\hat{\rho}_0$ is the estimate of the in-control mean.



(a)



(b)


:Agent (Parent)
:Local Platform
:Neighboring Platform
Sense pheromone
Request migration
Energy check
Investigate pheromone to know where it comes in
Determine the destination for migration
Return the destination information
Execute migration

(c)

Fig. 4.(a) Pheromone emission behavior. (b) Replication behavior. (c) Migration procedure.

Replication occurs when the energy level of the agent exceeds its replication threshold, or pheromone stimulation threshold is bigger than pheromone concentration value. The agent keeps replicating itself until its energy level becomes less than its replication threshold value.

When the agent replicates itself, pheromone stimulation threshold decreases at the same time. When an agent replicates itself, the agent gives its energy units to its child agent (replicated agent). The child agent gathers the sensory information from its parent agent that stimulated the parent agent to perform the replication behavior (see Fig. 4(b)).

Because the pheromone decays on a hop-by-hop basis and includes where it is flowed from, the agent can sense where the original platform exist approximately, and move toward the original platform by climbing pheromone gradients. This allows agents to move to neighbor platforms (see Fig. 4(c)). In order for agents to determine which neighboring platform they have to move to, platforms periodically relay pheromone toothier platforms. The component references are critical for tracking moving components and invoking their methods. This framework provides the APIs for invoking the methods of components on local or remote computers.

Table 1. The protocol for message exchange.

Field	Value Set
Message Header	
Delivery Mode	The same as that specified on the request
Expiration	Derived from the request. Decide whether to degrade
Priority	Copied from the request
Correlation ID	Copied from the request Message ID
Destination	Copied from the Reply To property in the request
Message Properties	
Request URI	Copied from the request URI property in the request message
binding Version	Copied from the binding Version property
content Type	Inferred from the Envelope and presence of attachments
Message Body	
Body	Serialized message according to the type

It does not have to statically define any stub or skeleton interfaces through a precompiler approach because our target is a dynamiccomputing system (see the example invoking procedure). The component receives a reference at the current, which is given when the host is originally created.

The references compared with the GID of the host at the new location. Then other components can invoke the methods of the newly attracted component. In addition, based on Orb's publish/subscribe concept, our framework allows users to implement the generic remote publish/subscribe mechanism that enables subscribers to express their requirements or tasks for an event. As a result, components may easily recognize whether or not they can accomplish their requirements at the current host. Table 2 is brief summary for message protocol.

5.2 Programmable Control for Arbitrary Shape Formation

A component is a connected group of particles. The self-assembly of a component occurs when two smaller components combine each other. We denote the set of all component types in a system by C = {C1, C2, C3, ...}. The number of component types may be either finite or countable infinite. A macrostate describes the number of each type of component in the system at a given time.

The components in the systems, which are considered in this paper, are composed of some number of parts. If two components interact, they should communicate to do so via one part from each component, and these two parts can locally decide whether or not to bind according to their programmed knowledge or programming. In this section, we consider natural systems in which interacting parts always decide to bind. This paper suggests a simple model of polymerization and shows an example of the programmable parts deals with our test bed robots.

$$f \odot f(\overline{S_0}) = \text{TM}(\text{TM}\overline{S_0} + \overline{K}) + K$$
$$f \odot f \odot f(\overline{S_0}) = \text{TM}(\text{TM}(\text{TM}\overline{S_0} + \overline{K}) + K) + K$$
$$f \odot f \odot f(\overline{S_0}) = \text{TM}^3\overline{S_0} + \text{TM}^2\overline{K} + \text{TM}\overline{K} + \overline{K}$$

$$f \odot f(\overline{S_0})! \;=\; \text{TM}^t\overline{S_0} + \text{TM}^{t-1}\overline{K} + \text{TM}^{t-2}\overline{K} + ... + \text{TM}\overline{K} + \overline{K}$$

A. Indexing of Smaller Components

Two-dimensional cellular automata of identical cells can be used to mimic morphogenesis. Each cell of the automaton receives inputs from its four immediate neighbors, those to the north, south, east and west of itself. These inputs are 'morphogens' that, via a set of rules, determine both the state of the cell and the form of the output morphogen from the cell. The set of rules obeyed by each cell is identical for each cell.

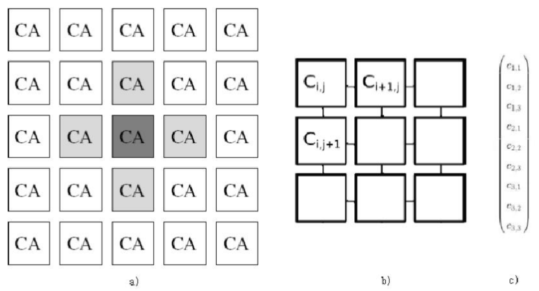


Fig. 5. This figure shows a grid of cells. In all cells identical cellular automata (CA) are running. (a) Cellular automata can change their state depending on the state of a neighborhood of cellular automata (b)Index of CA elements (c)a row-major vector equivalent.

$$f \odot f(\overline{S_0})! = \text{TM}!\overline{S_0} + (\frac{I - TM^{t-1}}{I - TM})\overline{K}$$

The biological analogue of the output of an automata cell would be the proteins it forms respective of its location within the body. Within the model, the organ will be a pattern of colors and each cell output will be a color. Each cell will communicate an integer state with its immediate neighbors. At each discrete time-step every cell computes its next state. Let us index each cell with the tuple (i,j), then describe the state of each cell at time t with an integer, ci,j,t and the pattern of the entire array as a matrix, Ct (see Fig. 5).

B. Model for Programmable and Programmed Systems

If S0 is the initial shape of St ,f(S0) is its subsequent pattern after one time step, and f(f(S0)) or f⊙(f(S0)) is its pattern at t = 0, where the function f() describes the transition function.

The matrix TMt (transition matrix) is first transcribed into a row-major vector, Pt (Fig. 5 (b)) in order for f() to be a linear function of matrix algebra. Let us now define a simple transition function from one time step to the next:

(8)

where, e, s, w and x are coefficients of the state of neighbors of each cell and of the state of the

$$S_{i,j,t+1} \;=\; nS_{i,j-1,t} + wS_{i-1,j,t} + eS_{i+1,j,t} + sS_{i,j+1,t} + xS_{i,j,t+k}$$

cell itself. A transition function for the entire array can be formed from (1) such that f(St)= TMSt + K where K is a constant and the transition matrix (for a 3 by 3 CA), TM, takes the form:

$$\mathrm{TM} = \begin{pmatrix} x & e & 0 & s & 0 & 0 & 0 & 0 & 0 \\ w & x & e & 0 & s & 0 & 0 & 0 & 0 \\ 0 & w & x & 0 & 0 & s & 0 & 0 & 0 \\ n & 0 & 0 & x & e & 0 & s & 0 & 0 \\ 0 & n & 0 & w & x & e & 0 & s & 0 \\ 0 & 0 & n & 0 & w & x & 0 & 0 & s \\ 0 & 0 & 0 & n & 0 & 0 & x & e & 0 \\ 0 & 0 & 0 & 0 & n & 0 & w & x & e \\ 0 & 0 & 0 & 0 & 0 & n & 0 & w & x \end{pmatrix}$$

(9)

To determine the rates for reactions between programmable part components, we use a high-fidelity mechanics-based simulation of the system [6] and fit the stochastic process described by the Master Equation to the data. We repeat the essential details here for completeness. The spacing of the coefficients n,e, s,w and x within TM depend on the size of the CA. By the repeated application off(), the transition from S0 to St (where t >1) becomes a non-linear function: (10) Using the geometric series equation this can be simplified to form: (11)

Eq.(11) determines the pattern formed after t iterations of the transition function (1) have been applied to every cell synchronously. Given a sufficiently large t, in order for the dynamic nonlinear system to converge, the final pattern, **Error! Reference source not found.**must be independent of the initial pattern, **Error! Reference source not found.**. Thus, no matter the starting pattern (where t = 0 refers to the initial patter nor any pattern that might be the result of system corruption),the pattern of cell states will always return to the same stable pattern. To satisfy this constraint TMt, the coefficient of **Error! Reference source not found.**must equal zero. For this to be so, referring to the coefficients of the states of the cells above, below, left and right and of the cell itself respectively, the following three constraints must hold:

1) Either n or s must equal zero
2) Either e or w must equal zero
3) x must equal zero.

This tells us that each cell must determine its next input according to the state of one neighboring cell per axis. By expanding this analysis to an alternative sum-of-products modulo-two transition function, it can be shown that this conclusion also holds true for combinatorial transition functions.

6. Experimental Results

In this section we show several experimental examples. Of course the whole experiments are inspired from [31]. They described simple heuristic algorithms for shape generation using bary center which each particles senses gradients propagated by all other particles. However, as shown in their results, they had to frequently changed (adding or removing something) to form arbitrary patterns. It means that it requires the agents to be modified to form a specific shape.

In contrast, our model described in this paper illustrates formula methods for self-assembly of robots. The experiments have been performed in the simulation environments that we have developed. The overall resulting shapes are able to be similar to [31] but each simulation shows difference. As shown in Self-Repairing subsection, our algorithm can be adopted to shape restoration because each agent form (from any starting configuration) and hold a swarm in a class of shapes, and in the event of damage, reform the shape. This is all done without any centralized functions such as seed or leader. This shape is fully given to every robot ahead of time, in the form of the function f().

A. Circular Shaping

In this example, molecular robots run a distributed algorithm to assume a circle shape. Each robots runs the barycenteralgorithm described in [31].

This situation may cause considerable latency in service and cause the system to stall. However, our framework is able to avoid such issues through the introduction of thread pooling. Incoming components are allocated to the most inactive thread. This ensures that all services will be served within a certain amount of time. The resulting center particle (dark dotted) will serve as the circle center. This particle propagates a CIRCLE gradient which increases its value by one at each step. All the other particles sense the morphogenetic gradient. If they sense a value greater than R (intended circle radius) they move along the decreasing propagation direction of the CIRCLE gradient. Eventually, all particles outside e intended circle radius will collapse toward it.

Fig. 6. Different stages of the circle formation. As the bary center starts propagating the circle gradient, some particles (in black) already recognize themselves as being at the correct distance from the center and do not move; the other particles gradually collapse toward the circle circumference.

B. Formation of Ring

First the center particle has run, and the CIRCLE gradienth as been propagated. The particles on the circumference (i.e. those perceiving the CIRCLE gradient with value R) start propagating a RING gradient which increases its value by one at each step. This RING gradient, which also propagates to the inner parts of the circles, attracts particles towards the circumference, thus emptying the inside of the ring. The thickness of the ring can be tuned by having particles stop following the RING gradient when it reaches a value of T, where T will consequently be the thickness.

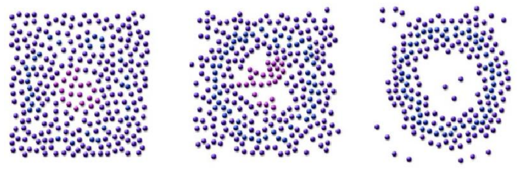


Fig. 7. As the particles at the correct distance from the barycenter self-recognize to be there, they start injecting a gradient that attracts inner particles.

In this simulation, we are able to find the difference from[31]. That is, the pattern of [31] is little bit irregular. On the other hand, in our experiment, the pattern is almost ring shape. In addition, outliers are closely placed from the ring boundary.

C. Formation of Limb

The overall idea of this simulation is to exploit particles density as the source to break the symmetry. For instance, when forming a circle, particles start to collapse toward the circle itself. If the number of particles compared to the circle size is very high, then the perimeter of the circle will be very crowded.

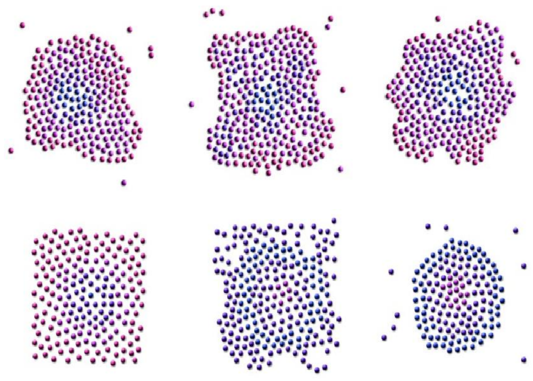


Fig. 8.The particles in the circle that detect a high density of particles, inhibit the propagation of the circle gradient, thus leading to the formation of lobes.

The idea is to force particles in very crowded areas to rearrange their positions so as to stay more separate from each other. This process ends up in a slight deformation of the circle(i.e., in the emergence of small 'limbs') in those part of its circumference where an excess of particles are accumulating. These small emergent limbs can be amplified via an additional mechanism of morphogen gradient inhibition that, in turn, makes larger limbs emerge. In this experiment, we find the difference from [31]. Especially, unlike its irregular pattern, our resulting shape is almost like '6-star' as shown in Fig. 8.

7. Discussion

The proposed model is based on three dimensional shape and we thought it mainly makes the difference. We also applied the two force model like binary and triad force interaction. It should make the difference too. In addition, our key mechanism for aggregation is cellular automat of particles using pheromone (i.e., cytokine). It induced the particles' autonomous and collaboration for making a shape, and in the event it made the much more regular shape than [31].

The total overhead of cytokine sensing is determined by the amount of time it takes for the cytokine sensing service to find the migrating agent's cytokine (by accessing cytokine list maintained by the cytokine emission service), contact a representative of the platform that the cytokine specifies (i.e. the platform that the agent migrated to),and locate the migrated agent on the remote platform (see Table 4).

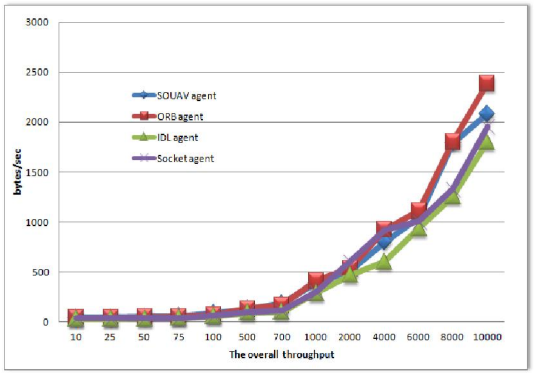
Fig. 9 (a) shows how the overhead changes when an agent senses cytokines emitted on remote platforms, multiple hops away. It also illustrates that the overhead increases linearly as the hop count to remote platforms increases, which indicates that the cytokine sensing service is scalable.

Fig.9 (b) shows the roundtrip time of a message between two components running on different frameworks. In this measurement, a single component is deployed on a platform and a disparate set of components (from 101 to104 receiver agents) is deployed on another platform. The sender randomly chooses one of the receivers and sends a component initialization message to the chosen receiver.

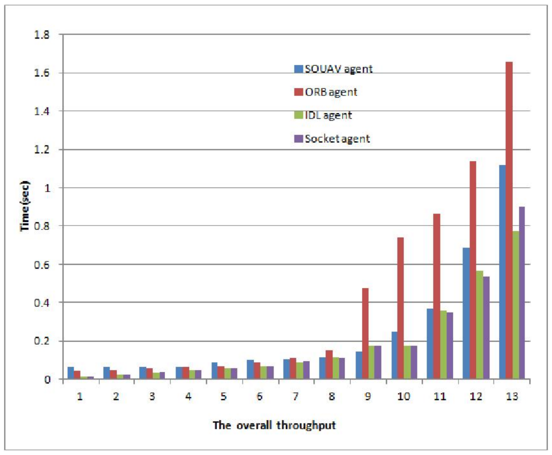
Then the receiver sends back an acknowledge message to the sender.

Fig. 9 (c)shows that the roundtrip time is comparable with well-known Java-based distributed object platforms (CORBA ORB and Java IDL), indicating that the networking interface for message transport is implemented efficiently. It also shows that the roundtrip time remains relatively constant as the number of receiver agents grows up to 10,000,which indicates scalability.

(a)



(b)



(c)

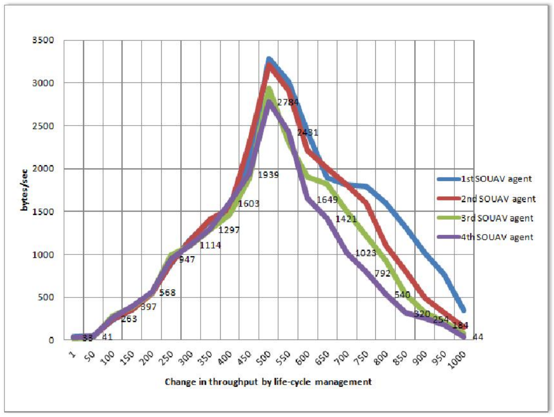


Fig. 9.(a) Overhead of environment sensing using cytokine emission services. (b) Roundtrip time of message between two components on different frameworks. (c) Comparing to other platform technologies.

8. Conclusions and Future Work

This paper presented a middleware system comprised of dynamic collections of components on a distributed system. With biologically inspired principles and mechanisms, network applications were created based on the proposed architecture satisfying the key requirements of future network applications such as autonomy, scalability, and adaptability. Empirical evaluation shows that the platform is efficient, scalable, and reusable. There are still further issues that need tube resolved. Although the current implementation focuses on the deployment of components, we plan to extend the framework so that it can be used to modify the behavior of each component during execution. The final goal of this architecture is to provide general test bed for various bio-inspired approaches for adaptive distributed systems. Also, since the current performance is not satisfactory, further measurements and optimizations are required.

Reference

[1]. Floyd S., Heiskanen T., Taylor T.W., Mann G.E., Ray W.H. Polymerizationof olefins through heterogeneous catalysis. VI. Effect of particle heat and mass transfer on polymerization behavior and polymer properties. J. ApplPolym Sci., vol.33, pp.1021-1065**(1987)**.

[2]Kosek J., Stepanek F., Novak A., Grof Z., Marek M. Multi-scale modeling of growing polymer particles in heterogeneous catalytic reactors. In: Gani R, Jłrgensen SB, eds. Proceedings of the European Symposium on Computer Aided Process Engineering 11 (ESCAPE- 11). Amsterdam: Elsevier; pp.171-176, **(2001)**.

[3]Grof Z., Kosek J., Marek M., Adler P.M. Modeling of morphogenesisof polyolefin particles: Catalyst fragmentation. AIChE J., vol.49,pp.1002- 1013, **(2003)**.

[4]Grof Z., Kosek J., Marek M. Principles of the morphogenesis of polyolefinparticles. Ind. Eng. Chem. Res. 2005, vol.44, pp.2389-2404.

[5] Tsuji Y., Kawaguchi T., Tanaka T. Discrete particle simulation of two dimensional fluidized bed. Powder Technol. vol.77, pp.79-87, **(1993)**.

[6] Young R.J., Lovell P.A. Introduction to Polymers. 2nd Edition. London: Chapman & Hall, (**1995**).

[7] Bay J. S., Unsal C. Spatial Self-Organization in Large Populations of Mobile Robots. IEEE Symposium on Intelligent Control, Columbus, **(1994)**.

[8]Coore D. Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer. PhD Thesis, MIT, (**1999**).

[9] Day S.J., Lawrence P.A., Morphogens: Measuring Dimensions: theRegulation of Size and Shape. Development, vol.127, pp.2977-2987, (**2000**).

[10]Fredslund J., Mataric M. A General Algorithm for Robot FormationsUsing Local Sensing and Minimal Communication, IEEE Transactionson Robotics and Automation., vol.18(5), pp.837-846, (**2002**).

[11]Guo Y., Poulton G., Valencia P. James G. Designing Self-Assemblyfor 2- Dimensional Building Blocks.

in Engineering Self-Organizing Applications, LNCS No. vol.1977, (**2004**).

[12]Nagpal R. Programmable Self-Assembly Using Biologically-Inspired Multirobot Control. ACM Joint Conference on Autonomous Agents and Multiagent Systems, Bologna (I), (**2002**).

[13]Shen W.M., Salemi B., Will P. Hormone-Inspired Adaptive Communication for Self- Reconfigurable Robots. IEEE Trans. on Robotics and Automation, vol.18(5), pp.1-12, (**2002**).

[14]Stoy K., Nagpal R. Self-Reconfiguration Using Directed Growth. 7thInt. Symposium on Distributed Autonomous Robotic Systems, Toulouse(F), (**2004**).

[15]Bojinov, H., A. Casal, and T. Hogg. Emergent Structures in Modular Self-Reconfigurable Robots. IEEE Intl. Conf. on Robotics and Automation,(**2000**).

[16]Bonabeau, E. From Classical Models of Morphogenesis to Agent-based Models of Pattern Formation, Artificial Life, vol.3, pp.191-211, (**1997**).

[17]Fredslund, J., and M. J. Mataric. A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication. IEEE Transactions on Robotics and Automation, vol.18(5), pp.837-846, (**2002**).

[18] Gordon, N., I. A. Wagner, and A. M. Bruckstein. Discrete Bee Dance Algorithm for Pattern Formation on a Grid. IEEE International Conferenceon Intelligents Agents Technologies, (**2003**).

[19]Guo, Y., G. Poulton, P. Valencia, and G. James. Designing Self-Assembly for 2-Dimensional Building Blocks.Engineering Self-Organizing Applications, vol.1977, (**2004**).

[20]Kondacs, A., Biologically-inspired Self-Assembly of 2D Shapes Using Global-to-local Compilation. International Joint Conference on ArtificialIntelligence (IJCAI), (**2003**).

[21] Murata S., Yoshida E., Kurokawa H., Tomita K., Kokaji S. Concept ofSelf-Reconfigurable Modular Robotic System. Journal AI in Engineering, vol.15(4), pp.383-387, (**2001**).

[22] Yoshida E., Murata S., Kamimura A., Tomita K., Kurokawa H., KokajiS. A Self-Reconfigurable Modular Robot: Reconfiguration Planning andExperiments. The International Journal of Robotics Research,vol.21(10), pp.903-916, (**2003**).

[23]Rus D., Vona M. Self-Reconfiguration Planning with Compressible Unit Modules. Proc. IEEE Intl. Conf. Robotics and Automation (ICRA99),pp.2513-2530, (**1999**).

[24]Rus D., Vona M. Crystalline Robots: Self-Reconfiguration with Compressible Unit Modules. Autonomous Robots, vol.10(1), pp.107-124, (**2001**).

[25] Turing A. The chemical basis of morphogenesis. Philos, Trans. Roy. Soc., Ser. B 237, vol.37, (**1950**).

[26]Hogeweg P. Shapes in the Shadow: Evolutionary Dynamics of Morphogenesis. Artificial Life, vol.6, pp.85-101, (**2000**).

[27]Scheirs J., Bigger S.W., Delatycki O. Structural morphology and compactionof nascent high-density polyethylene produced by supported catalysts. J Mater Sci. vol.26, pp.3171-3179, (**1999**).

[28]Simonazzi T., Cecchin G., Mazzullo S. An outlook on progress in polypropylene-based polymer technology. ProgPolym Sci. vol.16,pp.303-329, (**1991**).

[29] Thiele E.W. Relation between Catalytic Activity and Size of Particle.Ind.Eng. Chem., vol.31, pp.916, (**1939**).

[30] Yi Jiang, B.S. CELLULAR PATTERN FORMATION. Ph.D Thesis, Department of Physics of the University of Notre Dame, Indiana, (**1998**).

[31]Mamei M., Vasirani M., Zambonelli F. Experiments of Morphogenesis in Swarms of Simple Mobile Robots. Journal of Applied Artificial Intelligence, vol.18(9-10), pp.903-919, (**2004**).

[32] Jones, D. and McWilliam, R. and Purvis, A. Mimicking morphogenesisfor robust behaviour of cellular architectures. in Proceedings of World Academy of Science, Engineering and Technology International Conference on Biosciences and Bioengineering, pp.59-61, (**2008**).

[33]M. Murata, Biologically inspired communication network control, International Workshop on Self-* Properties in Complex Information Systems, **(2004)**

[34]I. Satoh, Building reusable mobile agents for network management, IEEE Transactions onSystems, Man and Cybernetics 33 (3), **(2003)**.

[35]J. Suzuki, T. Nakano, K. Fujii, N. Ikeda, T. Suda, Reconfiguration of network applications and middle ware systems in the bio-networking architectureProc. of IEEE LARTES,**(2002)**.

[36]K. Yeom, J. Park, Bio-Inspired Adaptive Framework Enabling Self-Organization and Management for Distributed Modular Agents, J. of Internet Technology 11 (5), **(2010)**

[37] Y. Wang and Y. Chen, Multiple-Obstacle Avoidance and Role Reassignment in Robot Formation Control, Adv. Sci. Lett. 4, 2864-2868 **(2011)**

[38]G.Cerofolini, P. Amato, M.Masserini, and G. Mauri, A Surveillance System for Early-Stage Diagnosis of Endogenous Diseases by Swarms of Nanobots, Adv. Sci. Lett. 3, 345-352 **(2010)**

[39] M. Roth, S. Wicker, Termite: emergent ad-hoc networking, Proc. of 115the Second Mediterranean Workshop on Ad-hoc networks, **(2003).**

6/12/2013