Activation Energy-Based Simulation for Self-Assembly of Multi-Shape Tiles

Mostafa M. H. Ellabaan
Cairo University
Doki, Giz, Egypt
(+2) (018) 256-3816
mostafa.mhashim@gmail.com

ABSTRACT

Building artificial systems using self-assembly is one of the main issues of artificial life [17]. Scientists are trying to understand this process either using experimental approaches or computer simulation approaches. This paper aims at supporting research using computer simulation approaches mimicking self-assembly as it occurs in the real world using basic principles in physics such as Brownian motion and basic concepts in Chemistry such as activation energy required to build molecules at molecular level. In this paper, a simulator has been implemented to mimic the process of self-assembly. In this simulator, objects are modeled by shapes such as cubes, tetrahedrons, wedges and pyramids with colors in their faces. The objects move randomly in Brownian motion, and the faces' colors determine how the objects interact with each other. Each object has its own probability to move which depends on the energy that the object gains through its interaction with other objects. The higher energy an object has the less probability that the object has to move and vise verse

Categories and Subject Descriptors

J.2 [**PHYSICAL SCIENCES AND ENGINEERING**]: Chemistry; Physics

J.3 [**LIFE AND MEDICAL SCIENCES**]: Biology and genetics

General Terms

Theory, Design, Performance, Algorithms

Keywords

Artificial life, self-assembly, Self-organization, simulation.

1. INTRODUCTION

Self-assembly is a process by which simple objects autonomously assemble into complex systems. This process is ubiquitous in the natural world [13] [18]. It may occur in levels ranging from nanoscale to astronomical scale. Some scientists especially Astrobiologists such as Deamer consider self-assembly behind the existence of life on Earth.

Dreamer described the development of life as a sequence of self assembly processes. This sequence started by a self-assembly process which transformed a soap of molecules into bubbles or lipids. These bubbles or lipids self-assembled into cell-like membranous structures. By this structure the first cellular system began to appear. This cellular system by one way or another was able to capture energy from surrounding environments and began to grow and reproduce [15].

Nowadays, artificial life scientists try to mimic the process of creating living systems aiming to build artificial systems capable of replacing current living systems [4] [17]. This paper aims at introducing an advanced simulator of self-assembly achieving an autonomous production of artificially complex shapes hoping to pave the way for generating complex systems using self-assembly.

In this paper, the different approaches used to model or simulate self-assembly processes are quickly introduced in section two as well as the Wang cube self-assembly simulator in section three. In section four, the activation energy-based simulator will be explained in detail in the advanced self-assembly simulator section. In section five, the advanced self-assembly simulator has been compared to the Wang cube simulator. Section six includes the final conclusion of this paper.

2. BACKGROUND: APPROACHES TO SELF-ASSEMBLY SIMULATION

In this section, the background of already implemented simulators has been organized or classified using two ways: The first one is the structured approach in which simulators are classified according to how objects and relations between them are modeled into a tiling-based approach and a bond-based approach. The second approach classifies simulators according to how many dimensions these simulators can handle. This approach of classification is called the visualization approach [1].

2.1 Structured approaches

In these approaches, we will study how to differentiate between simulators according to how these simulators handle objects and relationships between the objects. There are two approaches for classifying the simulators according this rule.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-698-1/07/0007...$5.00.

The first approach is the tiling-based approach where objects represented as shapes consist of one or more polygons and relationships between these shapes are modeled as adjacency relations between these objects. Examples for tiling-based self-assembly simulators are [5], [6] and [7].

The second approach is the bond-based approach where objects are modeled as atom-like shapes and relationships between them are modeled as line-like shapes called bonds. Examples for bond-based self-assembly simulator are [3], [8], [9] and [10]

2.2 Visualization Approaches

Self-assembly simulators may differ according to the number of dimensions that this process targets. Self-assembly processes may target (1) one-dimensional shapes, (2) two-dimensional shapes, or (3) three-dimensional shapes. In the first approach where self-assembly targets one-dimensional self-assembled systems, objects assemble into linear complex systems such as simulator implemented by [11]. In the second approach where self-assembly targets two-dimensional self-assembled systems, objects autonomously assemble into complicated two-dimensional systems by randomly moving objects in a two-dimensional simulated environment such as simulators implemented in [6], [8], and [12]. In the third and last approach, objects autonomously assemble into complicated three-dimensional systems by randomly moving objects in a three-dimensional environment such as simulators implemented in [2], [5] and [10].

3. WANG CUBE SIMULATOR FOR SELF-ASSEMBLY

The Wang cube simulator [16] is a tiling-based, and three-dimension self-assembly simulator aping the process of real world self-assembly. In this simulator, objects are modeled as cubes with colors in their faces. These colors determine how the objects interact with each other, and relationships between the objects are modeled as adjacency between the objects. In this simulator, objects move randomly in Brownian motion. If the objects collide with each other, they stay together provided that the interaction between the colors in the faces where the collision occurs is powerful enough to keep them stable, otherwise they continue moving according their previous states (i.e. the moving objects continue moving, and stable objects continue stable).

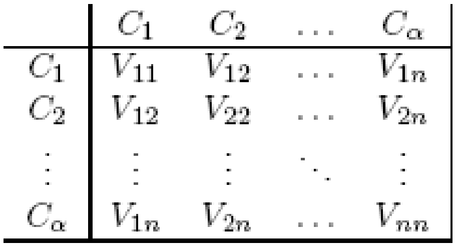
4. AN ADVANCED SIMULATOR OF SELF-ASSEMBLY

The advanced simulator is an extension to the Wang cube simulator. The simulator is implemented based on the tiling-based structured approach and the three-dimensional visualization approach. In this simulator, the self-assembly process is simulated using objects of different shapes such as cubes, tetrahedrons, wedges and pyramids. The objects move randomly in Brownian motion. If these objects collide together, their energy increases by a degree equal to the degree of the interaction between the colors on the faces where the collision occurs. Moreover, the energy that the object gains through its interaction with other objects determines the probability of the object to move. (See equation [1]). The more energy that an object has, the more stable the object will be. This concept is driven form the activation energy in Chemistry which is defined as the minimum energy that an object has to build a molecule or structure. By adding these concepts, the self-assembly simulator becomes more real in aping the real-world

In this section, we will describe the mathematical model that we used in our simulator, tools that we used to implement simulator, and the main implementation issues. Moreover, we will explain the experimental results that we have done to test the performance of the simulator.

4.1 Mathematical Description

In this simulator, objects are represented as shapes such as cubes, tetrahedrons, and pyramids with colors in their faces. These colors are taken from set A where A= {C1, C2… Cn} and Ci is a color and i is a number between 1 and n. The colors belonging to set A interact with each other forming what it is identified by the stickiness matrix or interaction matrix [6]. This interaction matrix is a symmetrical matrix consisting of n rows and n columns, and cell Vij where i is column number and j is row number determines the interaction between color Ci and color Cj (See Figure 1).



$$\begin{array}{c|cccc} & C_1 & C_2 & \cdots & C_\alpha \\ \hline C_1 & V_{11} & V_{12} & \cdots & V_{1n} \\ C_2 & V_{12} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_\alpha & V_{1n} & V_{2n} & \cdots & V_{nn} \end{array}$$

Figures 1. Interaction Matrix

In this simulator, objects move randomly in Brownian motion. This movement is constrained by the energy that the object gains through its interaction with the neighboring objects. In other words, every object has a probability to move, but this probability depends in the local interaction between the object and surrounding objects. The more energy that the object gains the more stable the object will become.

The Movement probability concept is driven from the activation energy which is defined as follows :

> *"It refers to the minimum average energy E which reactant molecules must have, in order to be able to produce product molecules. At a given temperature the fraction of molecules with this energy is usually proportional to the Boltzmann's population factor of exp (-E/k T)"* [14]

This statement states that the activation energy is the energy that the object should have to be capable to build a structure or a molecule. It is suggested to express movement probability as an exponential function expressed in the energy that the object has through its stickiness with other objects and the temperature multiplied by a specific constant such as the Boltzmann constant K as explained in the following equation:

$$P(O_J) = Exp(\frac{-E_J}{K \times T})$$

Equation [1]

$P\ (O_J)$ refers to the probability that the Object O_J moves. E_J is the energy that the object O_J has through its stickiness or interaction to other neighboring objects. E_J can be computed by the equation [2]. K is the Boltzmann constant, and T is the temperature. This equation shows that the more energy an object has the more stable the object will be. In other words, the stronger the stickiness the object has with neighboring objects, the less probability that the object has to move.

The energy that an object has is computed according to the following equation:

$$E_J = \sum St(Cjl, Cqf)) * B(j, l, q, f)$$

Equation [2]

Where E_J is the energy that an object *J* gains through its interaction with other objects, *Cjl* is the color on face *l* which belongs to object *j*, *St(N,H)* is the stickiness value between color *N* and color *H* which can be extracted from stickiness matrix row number *H* and column number *N*, and *B(j,l,q,f)* is a Boolean value that explains whether the face *f* belonging to object *q* is adjacent to face *l* belonging to Object *j* or not.

Objects will be capable to move in this simulator if they satisfy the following equation:

$$\Im(O_J) = \left\{\begin{matrix} 1 & & P(O_J) > \lambda \\ 0 & . & Otherwise \end{matrix}\right\}$$

Equation [43]

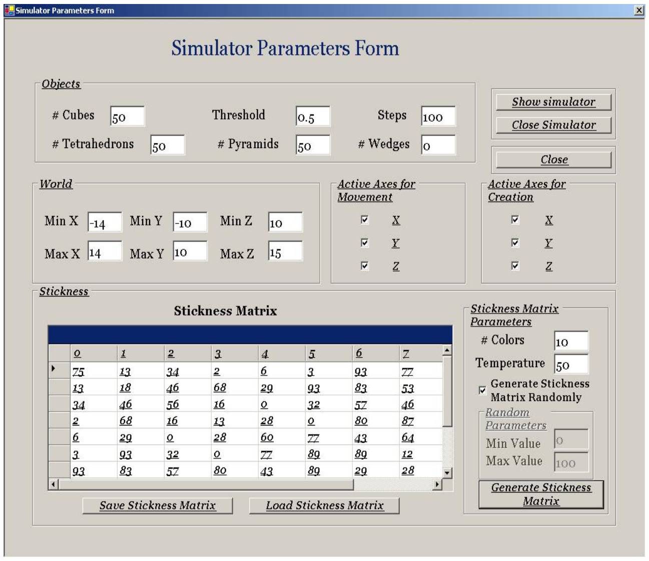
Where $\Im(O_J)$ describes the possibility for the object to move, and λ is a randomly generated number.

Equation [3] states that Object O_J will move in the case that $P(O_J)$ is greater than λ. Otherwise, Object O_J stays stable.

By adding the movement probability concept which is concluded from the *activation energy* concept in Chemistry, this model becomes more real than the previous one.

4.2 Implementation

We implemented a 3D simulator for self-assembly using DirectX 9.0 and C#. We represented shapes consisting of four, five or six faces. Each face consists of three or four vertices. These shapes or objects move randomly in a 3D environment: up, down, right, left, back or front based on the concept of the movement probability. To move an object, we implement three basic functions: the first function is to compute the energy that an object has (i.e. implementing equation [2]); the second function is to compute the probability of an object to move (i.e. implementing equation [1]); the third function is to determine whether the object will move or not (i.e. implementing equation [3]). If the object has $\Im(O_J)$ = 1 as stated in equation [3], the object will move, otherwise, the object will stay stable.



Simulator Parameters Form

Simulator Parameters Form

Objects

Cubes 50 Threshold 0.5 Steps 100

Tetrahedrons 50 # Pyramids 50 # Wedges 0

World

Min X -14 Min Y -10 Min Z 10

Max X 14 Max Y 10 Max Z 15

Active Axes for Movement

☑ X
☑ Y
☑ Z

Active Axes for Creation

☑ X
☑ Y
☑ Z

Stickness

Stickness Matrix

0	1	2	3	4	5	6	7
75	13	34	2	6	3	93	77
13	18	46	68	29	93	83	53
34	46	56	16	0	32	57	46
2	68	16	13	28	0	80	87
6	29	0	28	60	77	43	64
3	93	32	0	77	89	89	12
93	83	57	80	43	89	29	28

Stickness Matrix Parameters

Colors 10

Temperature 50

☑ Generate Stickness Matrix Randomly

Random Parameters

Min Value 0

Max Value 100

Figure 2: simulator parameters form

This simulator consists of two screens. In the first screen (see Figure 2), the basic required parameters should be inserted to represent the number of objects in terms of the basic shapes such as the number of cubes, the number of tetrahedrons, the number of wedges, the temperature, and the stickiness matrix. In the second screen (See Figure 3), the simulation process occurs where objects randomly move in Brownian motion in a predefined area called *"The world"* under the given parameters and according to the mathematical model defined in the previous section.

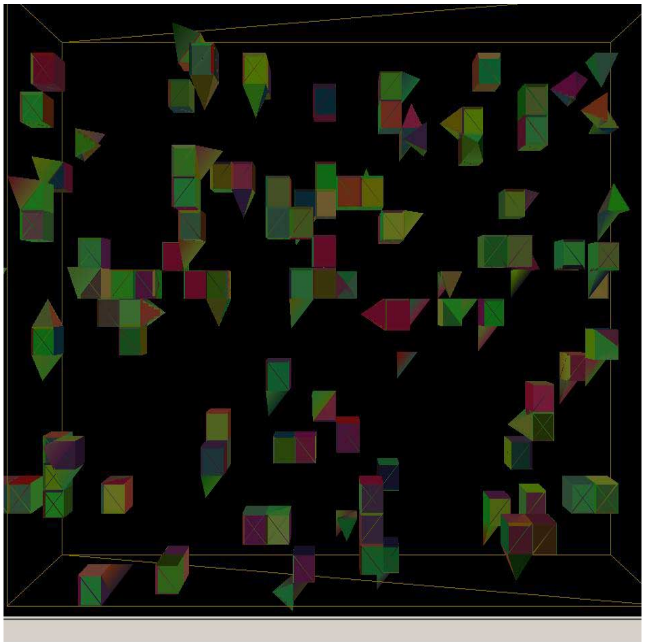


Figure 3: simulator form.

4.3 Performance

In this subsection, the performance of the advanced simulator will be tested by measuring the period of time that the simulator consumes to achieve a specific number of steps where the step is the single movement of all objects in the simulator. We tested the performance of the advanced simulator against two main parameters: (1) the number of objects simulated and (2) the temperature. To achieve that, we executed 750 experiments, 600 of which were executed to test performance against number of objects of different shapes, and the other 150 were executed to test performance against temperature. These experiments are explained in the following subsection

4.3.1 Performance and the number of objects

In this section, performance of the advanced simulator was tested against number of objects. 600 experiments were done, 150 of which were executed for each type of shape. In each experiment the temperature was 50, the number of steps was 100, and the stickiness matrix had 10 columns and 10 rows, where each cell of this matrix had a randomly generated value between 0 and 100.

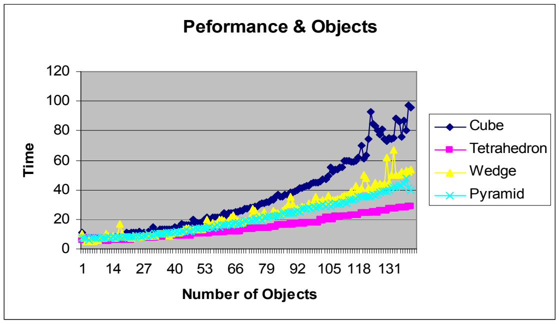


Figure 4: performance of the Wang Cube simulator against number of Objects.

Figure (4) shows that the performance of the system was affected be the number of objects simulated. The more objects, the more time was required to handle the creation, movement and collision of these objects, and the longer the period of time it took the simulator to execute.

Moreover, figure (4) also shows that performance was affected by the complexity of objects. For example, the performance of the simulator for tetrahedrons (only four triangular faces shape) is better than the performance of the simulator for cubes (only 4 square faces)

To sum up, this result states that the performance of the simulator is affected by the number of faces per shape as well as the number of vertices per face.

4.3.2 Performance and Temperature

In this section, performance of the advanced simulator was tested against temperature. 150 experiments were done. In each experiment, the number of cubes was 20, the number of tetrahedrons was 20, the number of wedges was 20, the number of pyramids was 20, the stickiness matrix had 10 columns and 10 rows where each cell of this matrix had a randomly generated value between 0 and 100, and the number of steps was 100.

Figure (5) shows that the performance of the system was slightly affected by the temperature when temperature was very small. Being unaffected by the temperature is one of the main advantageous features of the advanced simulator.

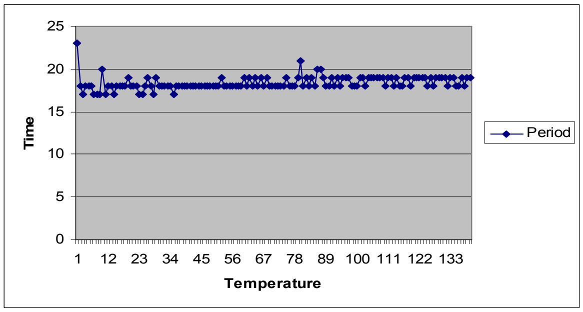


Figure 5: performance of system against temperature

5. COMPARISON

We will compare both the Wang cube simulator and the advanced self-assembly simulator against the following: (1) what shapes are included in the simulator? (2) How is the movement of objects handled? And (3) which simulator performs better?

Relative to (1) the shapes, The Wang cube simulator represents objects as cubes only, while in the advanced simulator, objects can be represented as shapes such as cubes, tetrahedrons, pyramid, and wedges.

Relative to (2) the movement of objects, the movement of an object in the Wang cube simulator for self-assembly occurs after generating a random integer between 1 and 6 determining the direction of movement. If this direction has no space to accommodate the cube, then no movement will occur, waiting for the next step which may generate a different movement direction. From this point, all cubes have the same probability to move while each cube has its own conditions or circumstances. Moreover, if two cubes collide together, and the stickiness value between them is greater than the temperature, they stop moving forever. In the real world, that does not occur. The stickiness values between objects or cubes reduce the possibility of movement of these cubes but do not make them stable forever. In the advanced simulator, all the objects have the probability to move, but this probability differs according to the local conditions of an object. By that way, even the attached objects or collided objects with a stickiness value greater than the temperature have a probability to move but a lower probability than the free objects. So the advanced simulator becomes better in aping the real-world than the Wang cube Simulator.

Relative to performance, we will compare the performance of the advanced simulator and Wang cube simulator against the number of objects and temperature.

Figure 6 compares performance of both simulators against the number of objects. It shows that performance of the Wang cube simulator is better than the performance of the advanced simulator. Why? Because in the case of collision in the Wang cube simulator if the two objects collide and they have a stickiness value greater than the temperature, they become stable forever, while in the case of the advanced simulator they stay stable until the movement probability becomes less than a randomly generated number (See equation [3]). So the stability rate in the case of the Wang cube simulator is greater than the stability rate in the advanced simulator.

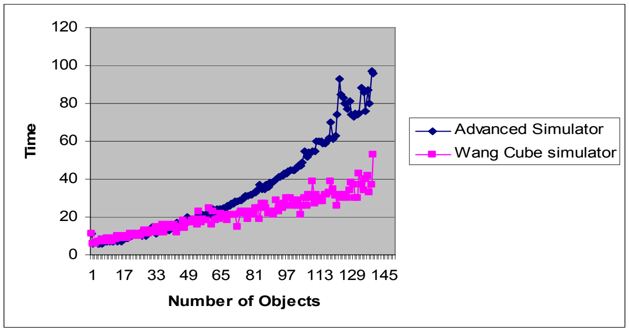


Figure 6: a comparison between performance of both the Wang cube simulator and the advance simulator against number of objects

Figure 7 shows that the performance of the Wang cube simulator is greatly affected by temperature, while the performance of the advanced simulator is slightly affected by the temperature when temperature is too small.

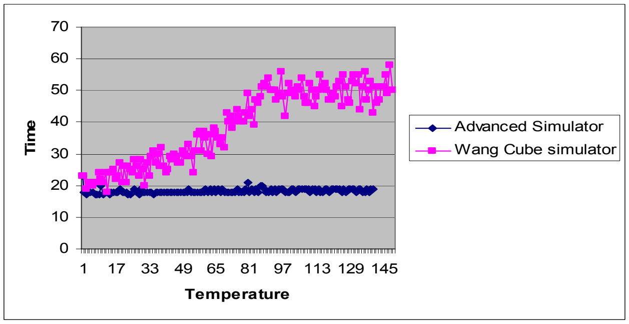


Figure 7: a comparison between performance of both the Wang cube simulator and the advanced simulator against temperature

Although the performance of the advanced simulator is less than the performance of the Wang cube simulator when comparing them against the number of objects, the advanced simulator is still considered better than the Wang cube simulator because the advanced simulator truly simulates the real world better than the Wang cube and the performance of the advanced simulator is not affected by the temperature as the Wang cube simulator is.

6. CONCLUSION

Self-assembly is an important process. It requires interdisciplinary collaboration to improve the understanding of this process or phenomenon. The more understanding gained about self-assembly, the more problems will be solved as well as the more applications will be gained. In this paper we have successfully mimicked some parts of the self-assembly process as it occurs in the real world, and we have also introduced the advanced simulator as a system in which objects are represented by shapes such as tetrahedrons, wedges and pyramids which are frequently used to represent objects at a molecular level. Moreover, this simulator handles the concept of movement probability which is driven from the activation energy in Chemistry. This concept states that all objects have the probability to move, but this probability differs from one object to another according to the energy that the object gains through its interaction with other objects. In other words, the more energy the object gains the more stable the object will become. By including more shapes and the concept of movement probability, the advanced simulator becomes better in simulating the self-assembly process as it occurs in the real world.

7. ACKNOWLEDGMENTS

I would like to express my faithful thanks to Mr. Ibrahim El-Tatway, Mr. German Terrazas Angulo and Dr. Natalio Krasnogor for their supportive help and useful discussion.

8. REFRENCES

[1] Ellabaan, M. “3D Simulation Of Self-Assembly”**,** M.Sc, Dissertation, the University of Nottingham, Nottingham, UK (2006).

[2] Zhang, T; Rohlfs, R.; Schwartz, R.; (2005): Implementation of discrete event simulator for biological self-assembly systems. Proceedings of the 2005 Winter Simulation Conference; P.2223-2231.

[3] Sohrabi, K.; Merrill, W. ; Elson, J.; Girod, L.; Newberg,F.; and Kaiser, W., Methods of for scalable self-assembly of Ad Hoc Wireless sensor Networks. IEEE Transactions on Mobile Computing, v.3 n.4 p.317-331, October 2004.

[4] Patwardhan, J.; Dwyer, C.; Lebeck, A.; and Sorin, D. (2004) :Circuit and System Architecture for DNA-Guided Self-Assembly of Nanoelectronics, Proceedings of the 1st Conference on the Foundations of Nanoscience, April, 2004.

[5] Gerasimov, V.; Guo, Y.; James, G.; Poulton, G.; and Valencia P.; (2004) Multiagent Self-assembly Simulation Environment, *proceedings of the* third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04), July 19-23, 2004, New York, New York, USA.

[6] Terrazas, G.; Krasnogor, N.; Georghe, M. and Kendall, G. (2005), Automated Tile Design for Self-Assembly Conformations; Proceedings of the 2005 IEEE Congress on Evolutionary Computation, September 2-5, 2005, Edinburgh, Scotland.

[7] Winfree, E.; Liu, F., Wenzler, L.A., and Seeman N.C. (1998) Design and Self-Assembly of Two-Dimensional DNA Crystals, Nature 394, p. 539-544.

[8] Li, L., Krasnogor, N., and Garibaldi, J., (2006), Automated Self-assembly Programming Paradigm: Initial Investigation. In Proceedings of the Third IEEE International Workshop on Engineering of Autonomic & Autonomous Systems, pages 25–34, Potsdamn, Germany, 2006.

[9] Shwartz, R. Shor, P. Prevelige, P., and Berger, B. 1998, local rules simulation of kinetics of virus capsid self-assembly, Biophysics 75: p.2626—2636

[10] Maillet, J. B. Lachet, V. and Coveney, P. V. (1999): Large scale molecular dynamics simulation of self-assembly processes in short and long chain cationic surfactants, Phys Chem Chem Phys, 1, 5277-5290 (1999).

[11] Winfree E. Schulman R. Lee, S. and Papadakis, N., (2003 B), One Dimensional boundaries for DNA tile self-assembly. DNA 2003, p. 108-126.

[12] Cook, M. Rothemund P. and Winfree, E (2003): Self-assembled Circuit Patterns. DNA 2003: P. 91-107

[13] Wikipedia, "Self-assembly definition", June 2006 URL: "http://en.wikipedia.org/wiki/Self-assembly"

[14] Wikipedia, "Energy Definition", Chemistry Section, June 2006 URL: "http://en.wikipedia.org/wiki/Energy#Chemistry"

[15] Deamer, D. (1999), the origin of life, Evolutionary theory conference November 14-19, 1999, URL: http://www.esalenctr.org/display/confpage.cfm?confid=5&pageid=50&pgtype=1

[16] Ellabaan, M. and Brailsford T, "Wang Cube Simulator for Self-assembly", the 4th International conference for information and communication technology, ICICT 2006, Cairo, Egypt.

[17] Brooks, R. "The relationship between matter and life by Rodney Brooks. Nature 409, 409-411 (2001) "Nature 409, 409-411 (2001)

[18] Whitesides, G. " Self-assembly and Nanotechnology", abstract for the fourth foresight conference on molecular nanotechnology, May 2006, URL: "http://www.zyvex.com/nanotech/nano4/whitesidesAbstract.html