Exploring the use of interacting morhporlogies as alternative design tools.

Olujide Esan
September 2003

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Virtual Environments from the University of London

Bartlett School of Graduate Studies
University College London

Acknowledgments

I would like to thank my supervisors, Lesley Gavin and Alasdair Turner, for their advise and support during the course of this research. I am grateful to Chiron Mottram for his extensive aid concerning the programming aspect at critical moments during the research.
I thank Ava Fatah for providing support and prompt advice when necessary during the course of the research.

Abstract

In approaching design two methods are often subscribed to; these are the top-down and the bottom-up design approaches. The top-down design gives consistent, predictable results, while the bottom-up design gives unexpected, more novel configurations. The trade-of is that the greater freedom and novelty gained from the bottom-up approach results in more processing time and steps to be invested.

The focus of this paper is to explore the interactive behaviour of morphologies in a system as creative entities within an active space.

The hypothesis is that recurrent interaction between transforming geometries within a system can result in the creation of form, and its use as an alternative design tool.

The research combines techniques and approaches from artificial intelligence and biological concepts of natural creation, and software agents represented as transforming geometries within an active space of interactions in a system, with a view to transforming their physical properties and employing their decision making capabilities to create design oriented form.

An agent based particle system to simulate the process and evaluate the outcomes is described. The details of the implementation of this system are presented as well as the various concepts supporting the proposed idea.

A theoretical base for the study is presented, with definitions and discussions of biological models of creation. A detailed description of the system is given and the results of the test carried out presented. The concluding chapter presents a case for the system as an alternative design tool based on the results derived from the test.

Keywords
Emergence, physical properties, Interaction, Reaction, Morphogenesis, Agents, Artificial-Life, Meta-Balls and Isomorphic Architectures

Table of Contents

Chapter One – Introduction

Introduction

In approaching design two methods are often subscribed to; these are the top-down and the bottom-up design approaches. The top-down design gives consistent, predictable results, while the bottom-up design gives unexpected, more novel configurations. The trade-of is that the greater freedom and novelty gained from the bottom-up approach results in more processing time and steps to be invested.

It is observed that the basic difference between the two methods is from the position which order is seen to proceed from. Architectural theorists Mark Wigley and Robert Venturi relate these methods to complexity and simplicity respectively by explaining that one subscribes to the theory of reduction and the other subscribes to emergence.

The search for alternative design procedures has lead to the adoption of biological models of development into the design processes. Of particular interest is the basic model of complex wholes and global behaviour derived from the interaction of a population of simple elements or emergent systems, which are derived by using evolutionary computation methods. The model is based on the idea that a whole may be more than the sum of its parts this is exemplified in other disciplines like biology where emergent behaviour of insect colonies results from decentralised local actions of many individuals.

Simulations of this model for design has been done with interaction of basic geometries within an active space, resulting in forms generated from transformation of basic geometries **P.Eleni et al [19]**.

According to **Wigley [10]** "The architect dreams of pure forms from which all instability and disorder has been excluded". Proponents for change in the way design processes are approached are challenging preconceived notions like this. In his essay on "architectural curvilinearity" published in 1993, Greg Lynn offers examples of new approaches to design that move away from the deconstructivism's "logic of conflict and contradiction" to develop a "more fluid logic of connectivity." He refers to design strategies that move away from Euclidean geometry of discrete volumes represented in Cartesian space, and is based on more fluid and continuous surfaces like NURBS. An example of commissions where such strategies have been employed for design is Bernard Franken's BMW Pavilion in Munich.

According to **Gregg Lynn [20]** he presupposes that "A shift from the present design procedures would equally imply a change in the instrumentalities for achieving the same objectives, thus freeing would be adopters of the model from the implicit limitations and built-in prejudices of both geometry and organizational analogies upon which present procedures are founded.

This paper will present a case for alternative design tools, exploring the use of the interactions of geometries in a particle system as a tool for form generation. The core of the particle system logic contents of this section.

The dictionary (Merriam Webster) defines a system as "a group of interacting or interdependent items under the influence of similar forces forming a unified whole."

There are many varieties of systems, on the one hand the interactions between the parts may be fixed (e.g. an engine), at the other extreme the interactions may be unconstrained (e.g. a gas). The system function depends upon the nature and arrangement of the parts and usually changes if parts are added, removed or rearranged. The system has properties that are emergent, if they are not intrinsically found within any of the parts, and exist only at a higher level of description.

1.1.3 Emergent Systems and Self-Organization

Systems that exhibit emergence are often driven by self-organization. A system is self-organizing if, left alone, it tends to become more organized. Examples include biological and artificial evolution, the development of urban structures, cellular automata, Conway's Game of Life, etc.

Holland [18] offers a definition for Emergence that emphasizes the existence of simple elements at the lowest levels of a system. In his view, elements interact according to simple laws to produce a seemingly diverse range of outcomes that, when examined carefully, in fact exhibit some level of regularity.

Emergence is what happens when an interconnected system of relatively simple elements self-organizes to form more intelligent, more adaptive higher-level behavior.

It's a bottom-up model; rather than being engineered by a general or a master planner, emergence begins at the ground level. Systems that at first glance seem vastly different -- ant colonies, human brains, cities, immune systems -- all turn out to follow the rules of emergence. In each of these systems, agents residing on one scale start producing behavior that lies a scale above them: ants create colonies, urbanites create neighborhoods.

Langton [17] states that we cannot predict the outcome of emergent systems from studying only the fine details. The local interaction of small, "simple" pieces, or agents, gives rise to global behavior. Global behavior "emerges" as a result of "simple" parts interacting on a lower "level." This global behavior results from the interactions of the parts. The global behavior that is not evident in the parts alone.

Emergent solutions effect the feeling of surprise in their observers precisely because of its novelty Any dynamic complex system, if it is able to do so, will try to organize its complexity so as to effect optimization within itself. As already explained a system or "organized whole" would emerge as a creation of a collection of interacting parts.

A self-organizing system thus, is any dynamic system from which order emerges entirely as a result of the properties of individual elements in the system, and without external influence(s). Good examples are found in most natural systems exhibiting fractal form or patterns; i.e. show organization on every level of magnification.

1.1.4 Morphogenesis

Morphogenic processes can be related to or sited as of emergent systems or self-organizing systems paradigms. Biological morphogenesis describes the embryological development of the structure of an organism in a whole or in part. It is a theory about how the actual ontogenic form comes into being in a given existential being. Examples are the growth of an animal from a fertilized ovum, biological evolution, learning, and societal development.

Morphogenesis literally means 'creation of shape'. It refers to the cell growth and movement generating the physical form during development. Morphogenesis exhibits several concepts common to other complex systems, like auto-organization, symmetry breakdown or emergence. A morphogenic system is capable of maintaining its continuity and integrity by changing essential aspects of its structure or organization. **(Von Bertalanffy, GST, pp. 148-9)**

Nowadays morphogenesis is becoming a commonly used term in more interdisciplinary fields. This term reflects general ideas of structure formation governed (directly or indirectly) by sets of rules or algorithms **[Mjolsness et al., 1991; deBoer et al., 1992; Cangelosi et al., 1994; Dellaert and Beer Kitano, 1994].**

The main idea of this approach is to encode rules that will themselves self-organize to produce a phenotype, as is the case with biological morphogenesis. However, the biological understanding of morphogenesis was inspired by concepts of "Mechanics of Development" that have been worked out in experimental embryology in the first half of our century. The biological conception of morphogenesis includes, apart from the idea of self-organization, so-called "regulation" principles. Biological morphogenesis, as a distinct self-organizing process, requires effective non-linear feedback between its dynamic components **[Beloussov et al., 1994].**

1.1.5 Collective intelligence

"It seems that intelligence, natural or artificial, is an emergent property of collective communication."
Corliss [26].

Heylighen [23] defines intelligence as the ability to solve problems. The term designates the cognitive powers; (perception, action planning and coordination, memory, imagination and hypothesis generation, inquisitiveness and learning abilities) of a group. Sometimes, as occurs in colonies of social insects such as termites and bees, intelligence is an emergent property, since each individual alone does not have the requisite neuronal capacity.

Collective intelligence means that families, groups, organizations, communities and entire societies can act intelligently as whole, living systems. Example scenarios like the workings of a built city illustrate relevance of this concept, a city works with complex mechanisms that together are too much for any individual human comprehension. A city built over time is the product of the collective intelligence of generations of people acting together, either in a spatial grouping or in a temporal perspective.

1.2 Agent Based Systems

Agent based systems represent practical application of collective intelligence; their most significant feature is their ability for interoperability. The remainder of this sub-section provides definitions and discusses the different types of agent systems available as they relate to this paper.

1.2.1 Definition of Agents

An agent system is a computational system that acquires sensory data from its environment and decides by itself how to relate the external stimulus to its behaviors in order to attain certain goals. Responding to different stimuli received from its task environment, the agent may select and exhibit different behavioral patterns, which may be carefully predefined or dynamically acquired by the agent based on some learning and adaptation mechanism(s).

Their interactions can be either cooperative or selfish. That is, the agents can share a common goal (e.g. an ant colony), or they can pursue their own interests. But because of their versatility in terms of use it is better to use the term as an umbrella term, meta-term or class, which covers a range of other more specific agent types, defined by their use and typologies.

Agent technology involves building complex applications from autonomous, interacting components. It is particularly suitable for modeling, simulating, and analyzing complex systems, as well as improving and optimizing their behavior. It can be perceived as a modeling paradigm, a problem-solving paradigm, or a software-engineering paradigm.

Three main ingredients necessary for agent based computing include

Agents: encapsulated computer systems that are situated in some environment and are capable of flexible, autonomous action in that environment in order to meet their design objectives.

Interactions: Such agents invariably need to interact with one another in order to manage their inter-dependencies. These interactions involve agents cooperating, negotiating and coordinating with one another.

1.2.2 Agent Typologies

Classification of agents is based on parameters these include; the role the agent would perform, the type of usage and the agent's attributes. **[1] (Nwana et. Al.).** Based on this assertion seven categories of agents are recognized namely;

- Collaborative Agents
- Interface Agents
- Mobile Agents
- Information/Internet Agents
- Reactive Agents

- Hybrid Agents
- Smart Agents

1.2.3 Reactive Agents

For the purposes of this paper we will be focusing on reactive agents. "*Reactive agents represent a special category of agents which do not possess internal, symbolic models of their environments instead they act/respond in a stimulus-response manner to the present state of the environment in which they are embedded.*" . **[1] (Nwana et. Al.).** Reactive agents behaviour are governed by simple rules, which have the ability to evolve into complex behaviour when observed as a whole or globally.

[27] Maes (1991a, p. 1) "*highlights the three key ideas which underpin reactive agents. Firstly, 'emergent functionality' which we have already mentioned, i.e. the dynamics of the interaction leads to the emergent complexity. Hence, there is no a priori specification (or plan) of the behaviour of the set-up of reactive agents. Secondly, is that of 'task decomposition': a reactive agent is viewed as a collection of modules, which operate autonomously and are responsible for specific tasks (e.g. sensing, motor control, computations, etc.). Communication between the modules is minimised and of quite a low-level nature. No global model exists within any of the agents and, hence, the global behaviour has to emerge. Thirdly, reactive agents tend to operate on representations which are close to raw sensor data, in contrast to the high-level symbolic representations that abound in the other types of agents*"

1.3 Isomorphic Architectures

"Isomorphic architecture is an assemblage of Isomorphic surfaces constructed as composite mutually inflecting parametric objects with internal forces of mass and attraction". **Greg Lynn [20]**
Blobs offer new possibilities for form creation by virtue of their amorphous properties.

"*Objects interact with each other instead of just occupying space; they become connected through logic where the whole is always open to variation as new blobs (fields of influence) are added as relations are made, creating new possibilities.* **Kolarevic, Branko [5]**"

1.3.1 Isomorphic Poly-surfaces

Meta-balls (or Blobs as they are often called), are types of implicit modeling technique that blends and transforms an assembly of spheres into a complex shape; it is suitable for modeling organic forms.

A meta-ball primitive is an object using spheres that attract and cling to each other according to their proximity to one another and their field of influence. It can be used on generic shapes. Its applications include creating organic objects and animation effects.

A meta-ball aggregate is defined as a single surface whose contours result from the interaction and assemblage of the multiple internal fields that define it.

Meta-Balls are nothing more than mathematical formulas that perform logical operations on one another (AND, OR), and that can be added and subtracted. This method is also called CSG, Constructive Solid Geometry.

The complete CSG area is divided into a 3D grid, and for each edge in the grid a calculation is made, and if (and more importantly where) the formula has a turning point, a 'vertex' to be polygonized is created there.

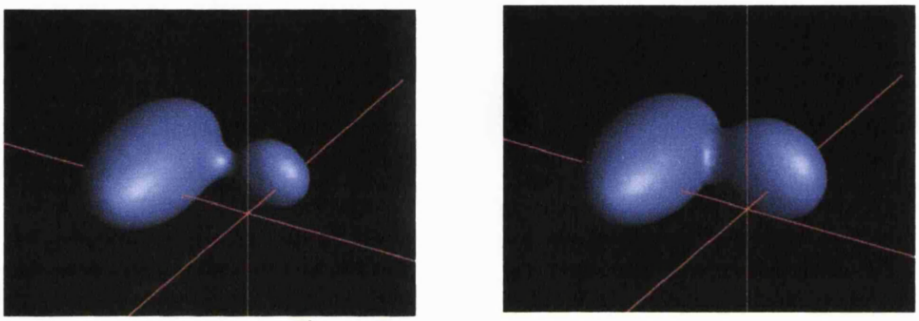


Figure 1

A meta-ball is defined by a so-called three-dimensional variable density field, radiating from a given center point. The value of the field can vary linearly with distance from the center, or in any other way expressible via a mathematical formula. For example, a field can have a negative density distribution, or even an eccentric distribution. A point on a meta-ball surface is constructed at all points in the field with the same density value, which is given by the modeler or derived from the modeling context. If two or more meta-balls are constructed in close proximity to one another so that they overlap, they coalesce and their fields are added in a process called fusion to produce a composite field, which is then evaluated to produce a composite surface. Meta-ball fields can be transformed in a variety of ways to produce organic shapes necessary to represent, for example, the human form. Meta-ball surfaces are usually rendered as polygons.

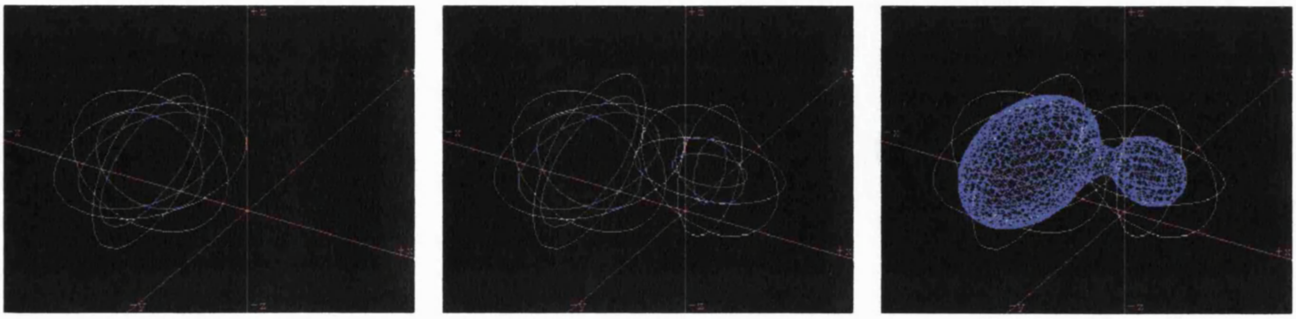


Figure 2

In this chapter, the main concepts and definition of terms supporting the topic of this study were presented and discussed with a view to elucidating and providing further justification for the questions and problems aimed to be solved by this study.

The next chapter presents examples and scenarios of researches where some or all of this concepts have been adopted. Problems encountered in these use cases are also highlighted. The subsequent chapters present details about the theoretical aspects supporting this study and follow with the description and implementation of the system proposed. Basic definitions of the relevant biological theories are given and aspects of artificial intelligence as regards software agents supporting this study are explained. Finally, some initial results are presented, and the future direction of the research is outlined.

Chapter Two - Related Research

The number of researches and studies employing the terms and concepts discussed in this paper are limited, however there exist isolated cases where some of the concepts have been used individually to create processes and solutions. The rest of the chapter introduces and discusses projects where each of these concepts has been used. Each project has been chosen for their relevance to the topic of the paper. Different researches employ either one or a combination of techniques to actualise the specific goals the studies are concerned with, this range from evolutionary to software agents as motivators for the creation of form within the different systems. The researches highlighted in this chapter are restricted to studies employing the methods and concepts, which best support and augment the topic proposed in this paper.

2.1 The design of morphogenesis

In this research by **C.Soddu, E.Colabella [22]**, the aim was to create an artificial intelligence software, to simulate the logical procedures design decisions through the use of some specific (and subjective) approach to the increasing complexity by generating a multiplicity of architectural or environmentally possible events. The use of morphogenesis further supports the argument for its use for creative purposes.

The research is based on the premise that design creativity is related to the variation of the random outcomes within an artificial environment. That The environmental design, as design of species, can directly operate the dynamic flowing of the system, identifying the sensitive parameters with the simulation of a connected sequence of possible scenarios.

The research software created called 'BASILICA' is a tool to design the morphogenetic code of the artificial environment that can be manipulated through a set of quantic parameters.

The first stage of the research was concerned with using artificial intelligence procedures to simulate the decision approach to design; subsequently additional functions were added to extend its use as an architectural and environmental design management tool.

A designer can build his own design paradigm and interactively change the relationships between events, change the geometry and stratify multiple possible geometries within the paradigm, define the quantity of possible exceptional events and the relationships between the events. Additionally BASILICA allows a designer to influence the creation of artificial events, which dictate the activities that occur within the system.

Live application of the BASILICA saw its use for designing the DNA of Italian medieval towns. Some of the problems encountered in this test were how to save the differences identified as morphogenetic code of the type of towns. **See Figure 3**

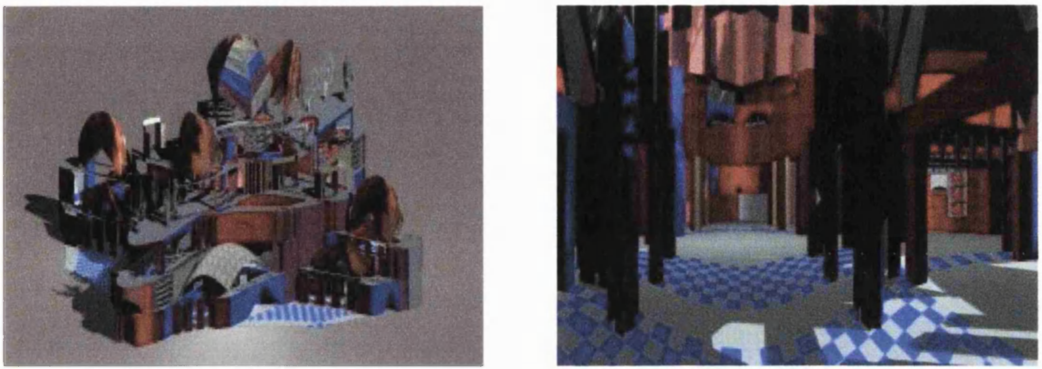


Figure 3 Views of Results produced

2.2 Interacting unities: An agent based system

In this research P.Eleni, A..Turner, R..Thum [19] investigate form generation through the pedestrian movement of software agents. The study investigates the role of movement as an external force in an active space of interactions by demonstrating that the recurrent interaction between agents and environment can lead to a structural coupling between those two elements; i.e. every time a change occurs in each one it triggers of the systems as an expression of its own structural dynamics, it triggers changes to the other one.

A series of experiments involving an agent-based system with simple rules that describe the behaviour of individual agents, were carried out to illustrate and test the stated assumptions. The experiments involved the random agent movement simulated as human pedestrian movement over a two-dimensional grided landscape made up of individual blocks.

Individual agents in the system modify their environment, which is a block of a two dimensional grid landscape, by translating each block they are standing on at the time, along with their height. Over time these transformations result in a global transformation of the landscape. In different iterations of the experiment different set of parameters that define the relationship between features of the agents and the landscape are manipulated, producing different conditions and visual effects for comparative purposes.

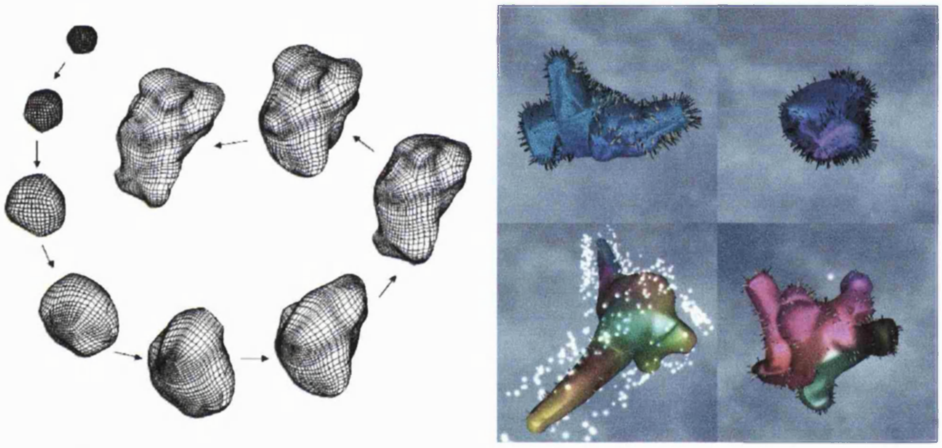
The figure below is a pseudo code illustrating the basic behaviour of each agent.

2.3 Aesthetic Selection of Morphogenetic Art Forms

This project by Dale Thomas [28] describes a system that allows a user allows a user to direct the evolution of three-dimensional shapes as a form of interactive art. The objects under evolution are defined by artificial DNA, which ultimately defines their final form and characteristics through an intermediate growth phase using interacting chemical reactions. The forms are part of a small population and the fitness is entirely defined by the user. The evolution continues and allows the user to design objects interactively or simply to explore the range of possible forms of the system. See Figure 4a

Users interact with the system through a touch screen interface and can select one of a number of individuals on the screen to mutate or breed. This creates a population of individuals similar in character to those selected. Users of the system can effect change in evolution properties by altering the fitness value. See Figure 4b

Physically the geometric primitives (forms) are defined in a simulated three-dimensional environment as a surface mesh structure consisting of springs. Exertion of forces on this springs result in the physical deformation on the geometries creating organic looking motion on the surfaces of the geometries.



a. Growth and form.

b. Forms possible with this system.

Figure 4

The genotype encodes a description of the interactions between a number N of abstract chemicals. Vertices of growing forms are composed of chemical concentrations, which react locally with one another and diffuse through the springs with rates also defined by the genotype. The structure of the genotype is divided into two parts: the first N alleles are the diffusion rates of the respective chemicals and the rest is divided up into a number of genes, each consisting of five alleles. Each gene represents an interaction and these five numbers correspond to the ‘from’ chemical, the ‘to’ chemical, the interaction weight and the upper and lower thresholds of the interaction.

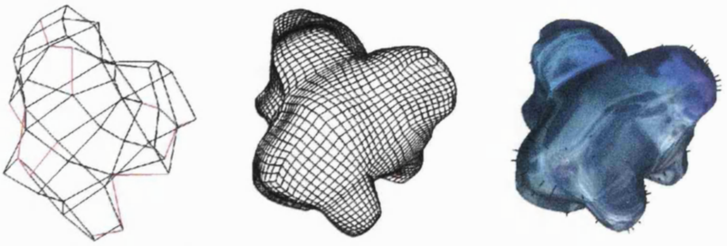


Figure 5 The three stages of object construction.

Discuss

This chapter presented some related researches employing concepts related to this paper and already discussed in the first chapter. The summary of the works chosen highlight the potential of these concepts in their area of application as they were used in each scenario and point out the pitfalls were applicable. The following chapter presents a simulation system to test the hypothesis proposed. Details of the system are described and outputs of experimental runs of the system are presented.

Chapter Three – Research Objective and Methodology

In this chapter a brief explanation for the motivation for this study is presented, the subsequent section outlines the methodology adopted in the simulation presented in chapter 4.

3.1 Motivation

"Presently, practicing architects are seeking new ways of design methods or procedures" **Megan Yakely [24]**; they require a shift from the current methods, which have been in use for the past eighty or so years. Critics have argued for a change from the Cartesian movement associated with isolation and the reduction of systems to their constitutive elements. Present day architects both academic and practicing have been toying with different ideas as to how design should be approached; this fact is evident in the work of architect like Bernard Tschumi, Daniel Libeskind and Peter Eisenman. Other proponents of change include the followers of Deconstructitvist Architecture who prefer to view architecture from its basic elements

In Darwinian fashion complexity is described as the gradual accumulation of differences that are in essence random in their combination and mutation. **Peter Zellner[6]** proposes that" architecture itself is mutating, reinventing itself, its essential coding, to adjust to an increasingly supple and volatile world. Architecture is recasting itself, becoming in part an experimental investigation of topological geometries, partly a computational orchestration of robotic material production and partly a generative, kinematic sculpting of space." He supports the idea that present day architecture need not be confined to the limitations inhibiting its procedures, but rather that modern architecture should be bred out of heterogeneity i.e. a composition of ideas and concepts rather perpetuating the myth of ideal form. He adds that such "an architecture evolves through the embodiment of competing identities. Unraveling and consuming opposed geometries and spatial postulates."

Greg Lynn [20] buttresses this view by describing complexity as the fusion of multiple and different systems into an assemblage, which behaves as a singularity while remaining irreducible to any, single simple organization. Thus a system exhibiting such complexity would be organised as a whole (singularity) and yet it would be distinguishable by its internal assemblage of discreet components (multiplicity).

It becomes apparent that alternative geometries are needed to illustrate these properties of multiplicity and singularity.

A complex system of the nature described above is created as a product of iterative reduction through variation and the emergence of order through interaction. The process of creation of a system of this type is synonymous with biological models of morphogenesis. Thus there's a need to engineer emergent behaviour in other to satisfy the requirements for the creation of such a system.

3.2 Research Methodology

The focus of the research is to investigate the use of a system of transforming geometries as a form-generating tool. A particle system of software agents is developed to simulate the required conditions.

The choice of a particle system as the mode of testing the simulation provides an avenue for merging the different concepts proposed by the system and most importantly enables a study of autonomous bodies in interacting in a truly random and unpredictable state. It is hoped that the simulation of this system would create the possibility of effecting processes like mutation and combination in their truly pure state without external interference.

It aims to create form by imitating biological model of cellular growth through the manipulation of reactive software agents physically represented as changing geometries

In this system the decision-making capabilities of software agents are combined with the transformative and interactive properties of isomorphic poly-surfaces (Meta-Ball primitives) to simulate cellular growth (morphogenesis) with a view to generating form.

The next chapter presents detail about the technical details supporting this study and follow with the description and implementation of the system proposed. Basic explanations of the implementation the biological concepts already defined in chapter 1 are given and aspects of artificial intelligence as regards software agents supporting this study are explained. In the final chapter, some initial results are presented, and the future direction for research is outlined.

Chapter Four – system description and Implementation

4.1 System Design and Description

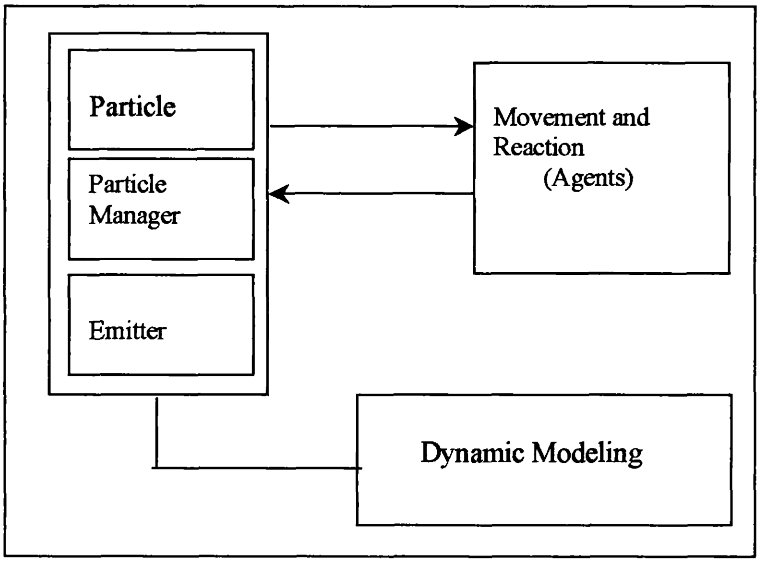
The objective of the project is to explore the possibility of evolving morphologies in a space of interaction as a form creation tool.

The system described is written in C++ with OpenGL performer libraries, basically it incorporates implementations of artificial-Life and geometric modeling concepts to create from.

The system is implemented as a particle system comprising of; three distinct parts namely;

- Creation and Destruction (morphogenesis)
- Movement and Reaction (Agents)
- Dynamic Modeling (Meta-Balls)

The relationship between the different parts is illustrated in the schematic diagram below
see diagram



Diagram

4.2 Composition

The rest of this section presents a breakdown of the compositional parts of the simulation program samples of codes and pseudo-codes to further illustrate the functions of parts of the program are provided where applicable. An overview is presented in each section before proceeding to the implementation details

4.2.1 Creation and Destruction (Morphogenesis) As already defined in chapter 1, refers to the cell growth and movement generating the physical form during development

Taking into consideration the need to simulate the growth of form in a cellular manner, it was indicated that there be a need to introduce aspects of biological development methods in a non-

indicated that there be a need to introduce aspects of biological development methods in a non-complex way to illustrate the organic growth of form. morphogenesis is the
This aspect of the program simulates the biological process of morphogenesis, essentially its function is to create new particles every now and then and remove the old one from the 3D environment. **See Figure 4a**

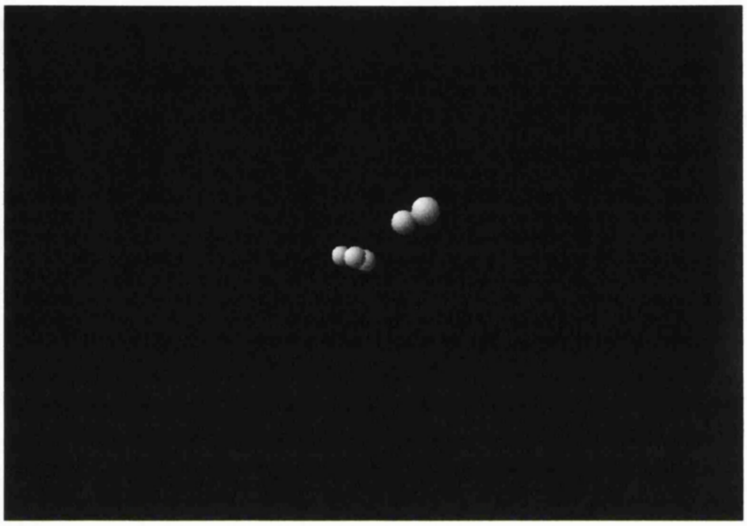


Figure 4a Examples of Geometries produced by the system controlled by the emitter class

The main program class for managing this process of the simulation is the eEmitter class, all variables concerning the creation of geometry in the system and the managing of their attributes are controlled from within the this class. The class is made up of a looped link list, which is created during initialization of the system The class controls the creation and destruction of particles, variables within this class control the rate at which particles are created, and monitor the number of active particles within the virtual space at a time. **See Appendix A.**

```
int loop;
particlePool = new partManager;
partManager *last = particlePool;;
for (loop = 0; loop < MAX - 1; loop++)
{
            partManager *pm = new partManager;
            particlePool->next = pm;
            pm->prev = particlePool;
            particlePool = pm;
```

Sample of Linked List code

```
eEmitter::eEmitter()
{
            eEmitter *eM;
            eM = this;
            eM->prev = NULL;                Initialization for the linked List loop
            eM->next = NULL; ;              Initialization for the linked List loop
            eM->totalPart = MAX;            Controls the Total Number of Particles in the system
            eM->partCount = 0;              Counts the number of particles in the 3D environment
            eM->life = 4;                   Determines the life-span of each particle created
            eM->lifeVar = 5;                Adds variance to the particle creation for randomness
            eM->emitsPerFrame = 20;         Determines the number of emissions per frame
            eM->emitVar        = 10;        Adds variance to the particle emission
```

4.2.2 Movement and Reaction (Agents)

As already described in chapter one multi-agent systems are described systems in which several autonomous acting entities work together to reach a given goal. In the system the use of agents is focused on physical modification and interaction, agents are represented as individual spherically shaped particles, which interact and respond to activity from their immediate surrounding.

Each particle exist as an autonomous entity with a range of behaviours available to it. They are continually created with varying properties and life times to exist within the system, thus increasing the chances for novelty. During the lifetime of each particle, its physical properties are transformed as responses to interaction within its environment (active space of interaction). Each particle moves in a random manner until it encounters another particle within its space. Interaction between individuals is typified by collisions, which result in transformations of both particles to create more complex form(s).

The behaviour attributable to each particle is governed by a set of simple rules which make the decision making process of the individual agents.

Each agent is physically transformed by its collision and fusion with other agents to create more complex forms and consequently modifying its environment. Variables defining relationships between features of the agents and the environment were modified in continuous runs of the systems with a view to exploring more outcomes and fulfilling experimental expectations. Details of these are provided below with accompanying images of the outcomes.

```
Loop
    look around, alter direction according to longest line of sight
                if counter is 0 and the movement speed is –1
                    Turn to face direction of longest distance
```

```
void caAgent::move()
{
  bool blocked = false;
  static pfMatrix mat;
  getMat(mat);
    if (pfuRandomLong() % 10 == 0 || m_speed == -1.0f) {
    look(mat);
  }
  pfVec2 preloc( (float) floor(mat[3][0] + 0.5f), (float) floor(mat[3][1] + 0.5f) );
  mat.preTrans(m_speed,0.0f,0.0f,mat);
  pfVec2 postloc( (float) floor(mat[3][0] + 0.5f), (float) floor(mat[3][1] + 0.5f) );
  if (preloc != postloc) {
    if (!g_surface->includes(postloc)) {
      blocked = true;
    }
    else {
      float heightdiff = g_surface->getHeight(postloc) - g_surface->getHeight(preloc);
      positive
      if (fabs(heightdiff) > 0.75f) {
        blocked = true;
      }
      else {
        g_surface->replaceAgent(preloc,postloc);
        mat[3][2] = g_surface->getHeight(postloc) + 0.875f;
      }
    }
  }
  if (!blocked) {
    setMat(mat);
```

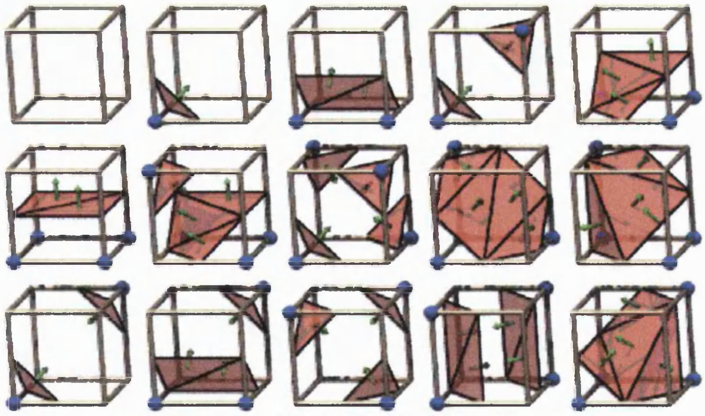
4.2.3 Dynamic Modeling (Isomorphic poly-surfaces)

The introduction of meta-balls to the systems was to achieve the dual purposes of simulating cellular growth and organic form transformation. The use of meta-balls provided the possibility to model cellular growth mechanisms such as adhesion, which are based on simple and local rules provided from the agent implementation.

There are various methods for modeling meta-balls, for this paper the marching cubes algorithm was used. The meta-balls were modeled using the Marching Cubes designed by William E. Lorensen and Harvey E. Cline.

Marching Cubes is an algorithm for rendering iso-surfaces in volumetric data. The basic notion is that we can define a voxel (cube) by the pixel values at the eight corners of the cube. If one or more pixels of a cube have values less than the user-specified iso-value, and one or more have values greater than this value, we know the voxel must contribute some component of the iso-surface. By determining which edges of the cube are intersected by the iso-surface, we can create triangular patches, which divide the cube between regions within the iso-surface and regions outside. By connecting the patches from all cubes on the iso-surface boundary, we get a surface representation. In this algorithm the user first specifies a threshold value. For this value, some voxels will be entirely inside or outside the corresponding iso-surface and some voxels will be intersected by the iso-surface. In the first pass of the algorithm the voxels that are intersected by the threshold value are identified. In the second pass these voxels are examined and a set of one or more polygons is produced, which are then output for rendering.

Below is sample code from the CUBE _GRID class



The 15 Cube Combinations

[ ] “Each of the voxel's 8 vertices can be either inside or outside of the iso-surface value, thus, there are 2^8 = 256 possible ways in which a surface can intersect the voxel. By symmetry these 256 ways can be reduced to 15. The exact edge intersection points are determined and the polygons are created. Use a Table Lookup to Reduce Edge Intersection Tests “

4.3 Implementation

The system is tested through a series of experimental runs; in each run varying conditions are derived from the manipulation of parameters that influence relations between different components of the system. The aim is to simulate different sets of conditions in the system for the purpose of comparison.

The following sets of results presented from two runs of the simulation are based on random motion in different dimensional axes.

The first run simulates the motion of the particles in two dimensions (X and Y), particles start out in a specified location the centre of the grid are allowed to roam in different directions at a constant speed. A particle is transformed when it comes within proximity of another particle. The physical properties of a particle are determined by the value of its threshold and its weight. The weight translates to the radius of influence of energy surrounding the particle. The manner of interaction (fusion) with a nearby particle is determined by the value of the threshold. For this simulation the value of the threshold is set over a range of positive values to produce only a fusing effect between adjacent particles. Each particle is given a life variance value, which determines how long the specific particle is present for before it dies. Adjusting this value has the effect of speeding-up or slowing down the speed of the creation of particles in the system. Adjustment of this variable to a high value, produced a somewhat chaotic effect in the system as shown in figure??

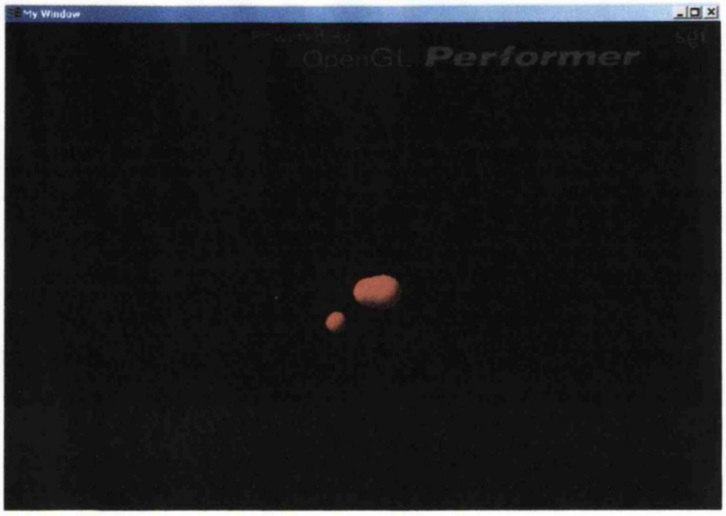
The geometric primitives representing the particles are polled from a pool created by the emitter class; the modularized structure of the program makes it easy to change the type of primitives used by the system in any iteration. Although for this run of experiments only the isomorphic poly-surfaces were used.

The movement and behaviour of the particles for the simulation using the grid was based on a simple agent model. The movement was forwards and restricted within the grid. The change of direction is based on the line of sight; the agents are programmed to look around randomly and change their direction based on the longest line of sight observed. The pseudo code below illustrate the mode of movement of each agent

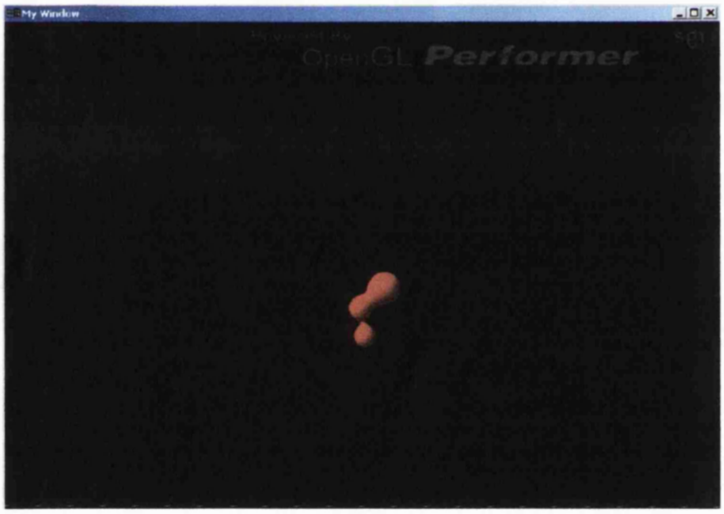
A default threshold value of 1.0f is set for each particle at the start of a run of the simulation; this produces a fairly stable result as seen in the image below,

particles tend to roam further maintaining their original attributes. Interactions are brief and less frequent, resulting in the creation of isolated clumps of amorphous forms distributed over the grid landscape. An increase in this the threshold value increases the radius of influence surrounding each particle and thus increasing the possibility of interaction with a close by neighbour. This results in less disparate creation of form over the landscape as particles tend fuse over a greater space. This result in bigger clumps of forms being created and with fewer isolated forms created on the fringes of the landscape.

A second run of the simulation saw the introduction of stationary particles onto the landscape



figure



figure

Stationary particles were introduced in a second run of the simulation influenced the outcome a different set forms on the grided landscape. The images in figure?? Show the results.

A free space for active interactions was used, in contrast to the restrictive grid landscape employed earlier. At the start of the run the stationary particles are locate randomly within the space, other particles are then created and removed by the program period of interactions with other particles. Simulation of life and death helps to accentuate the biological model of morphogenesis upon which part of the hypothesis is based. The introduction of the morphogenetic concept as it would seem helps to maintain the variation and novelty in the system. It should be of note that every particle is created with a different set of parameters such as life variation, initial starting position, and life value, thus continually changing internal conditions of the system at the genotypic level with a view to influencing the outcome or phenotype.

The chapter introduced the system developed to simulate the proposed ideas. It also provided a description and discussed the different components making up the systems as well as the technologies used finally it presented results from runs of iteration of the system. The Final and concluding chapter discusses the areas future development and proposes uses for the system.

Chapter Five – Future Work and Conclusion

5.1 Discussion

In summary the project explored transforming geometries in an active space as a tool for form creation. The system developed to illustrate the assumptions proposed, leveraged on artificial life, and evolutionary computing concepts and isomorphic poly-surfaces methods for modeling. Before proceeding further it is important to note that the present state of the system is that of a dynamic nature, this is due to the fact that geometries used (meta-balls) by their nature have to be in motion to produce the effects that make them unique i.e. fusion. Thus it must be added that most of the results deduced from the simulations were acquired by taking snapshots of the system at intervals. Alternative means of collecting results from the system is a topic for future research and is discussed later in the future section of this chapter.

Although not fully exploited in the pilot system presented in this project, the power of artificial life computing techniques cannot be understated.

The multi-agent system used coupled with the biological model (morphogenesis) employed encouraged emergent behavior in the outcomes of the result. The most significant outcomes were derived from the first simulation iteration, which used the grid. Reduction of the length of the line of sight meant the individual agents needn't look so far to effect their reactive behaviours. Over time this configuration it was observed that a similar type of form was created repeatedly, although it is quite difficult to recognize and compare results from the successive forms, due to their amorphous nature, nonetheless a pattern was identified.

As a design tool a user of the system would have the choice of picking from alternative outcomes thus providing the opportunity of finding innovative solutions for design task. The bottom-up approach to developing solutions enables the generation of broad result sets which should range from the simple to the complex, this would serve the dual purposes of properly analyzing a solution and having a better understanding of the origination and composition of the solution, secondly it serves to enrich the vocabulary of solution possibilities available to the user.

As highlighted in chapter3, a change in the approach to design equally necessitates the use of alternative geometric primitives to illustrate properties and qualities of the approach used. Isometric poly-surfaces were made popular by digital architecture practitioners like Greg Lynn and Bernhard Franken. They provide an alternative to the Platonic solids and Cartesian space which traditional architecture is based on, they offer alternatives to the norms of regular design methods as earlier stated [ ]. Isomorphic objects by nature and composition are capable of continual, consistent and dynamic transformation of their physical properties thus freeing users from the limitations of geometry and organizational analogies inherent in the present design methods and tools. Designers would be afforded more flexibility with the options available to them in terms of composing elements with which to create. Though it is understood that initial use would be rather daunting especially for first user it is believed that continuous use and interaction with the system, would spur the flow of creativity thereby giving rise to new possibilities, leading to more innovative designs and creations.

5.2 Future Work

The objective of this project has been to explore the interaction between transforming morphologies a form creation tool. This has been approached as a simulation of interactive geometric primitives based on biological models of morphogenesis within a particle system or active space of interaction. Experimental runs of the system demonstrating the viability of the process have been presented. Suggestions for future developments envisaged for improvement on the system proposed are summarized as follows;

5.2.1 Implementing Optimisation

The present results of the system though interesting can at best be describe as ordinary to a large extent, it is believed that the introduction of an optimization mechanism would prompt the production of more interesting and complex outputs from the system. Such an optimization strategy would be applied at the genotypic level; i.e. the composing elements (agents) to manifest outcomes from at the phenotypic level or the form created. More importantly it would improve the usability of the system by allowing the user more flexibility of use. A user can easily change the system variables via the parameters of the fitness function or genotype properties, and thus easily observe variations in the outputs of different iterations of the system. Also incorporating a GA will allow the possibility of testing the limitations and range of the system performance.
It would be interesting to see the introduction of a fitness function to the system; this is encouraged by the fact that the system is based on evolutionary concepts (morphogenesis) and artificial-life concepts. It is of the belief that the introduction of optimization principles like genetic algorithms would be particularly useful.

5.2.2 Interactivity

Being a test system the present state of interactivity in the system is poor; the ability of the system to engage the attention of a user is suspect. It is of the opinion that better result can be achieved in terms of outputs from the system by users, through prolonged use and experimentation. This can be achieved by the introduction of a more intuitive and detailed interface to the system. Accessibility to internal variables through the interface put more direct control at the disposal of a user. Such an interface would equally allow for immediate visual manifestation of changes as effected by a user. Additonally it would aid the possibilityf collecting result which is a significant problem presently.

5.2.3 Environment Enhancement

In the real world evolution is effected in biological organism in response to different interactions to environmental changes and agents.

Other aspects that need to be enhanced are the physical environment of the system with could be enhanced with gravity and collisions

Conclusion

The objective of this project has been to explore the interaction between transforming morphologies a form creation tool. This has been approached as a simulation of interactive geometric primitives based on biological models of morphogenesis within a particle system or active space of interaction.

In conclusion it must be reiterated that the focus of the paper has been to present a set of concepts and ideas on alternative means of creating forms primarily for the purposes of design, the proposal presented and the simulation model described are by no means conclusive or finished products. Considerable amount work is still required to bring them to a state considered as stable and usable. As already pointed in the previous sections, there exist many possible directions for further research. However from the arguments and points highlighted in the paper, it must be agreed that the potentials of use of the system as a design tools do exist.

An immediate application area would see its application as an extension or a plug-in to some of the existing CAD application tools. Although some work as already been done in this area **(Peter Testa et al) [25]**, an advantage of this system is in the area of flexibility, which lies in the fact that a fitness function can be easily implemented. the use of a multi-agent model provides more alternatives and options.

Bibliography

[1] Nwana H. S. (1996) Software agents: An overview. Knowledge Engineering Review,

[2] Robert Chatley () Hybrid Deliberative/Reactive Agents

[3] Greg Lynn () An Advanced Form of Movement, Architecture after Geometry, A.D Profile 127, 1997. Academy Group.

[4] Michael Wooldridge. Introduction to Multi-Agent Systems. John Wiley and Sons;. 2002.

[5] Kolarevic, Branko Designing and Manufacturing Architecture in the Digital Age University of Pennsylvania, USAhttp://www.upenn.edu/gsfa

[6]Peter Zellner (October 1999) HYBRID SPACE Rizzoli

[7] Wooldridge, M. & Jennings, N. (1995a), "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review 10 (2), 115-152.

[8] Wooldridge, M. (1995), "Conceptualising and Developing Agents", In Proceedings of the UNICOM Seminar on Agent Software, 25-26 April, London, 40-54.

[9] Klaus Krippendorff's Dictionary of Cybernetics 80 p. unpublished report dated Feb. 2, 1986,

[10] Alexander V. Spirov Life Morphogenesis in Models

[11] Mark H. Bickhard with Donald T. Campbell Emergence Undated publication

[12] Mitchel Resnick and Brian silverman Epistemology and Learning Group MIT Media Laboratory Exploring Emergence

[13] Nikos A. Salingaros Design methods, Emergence, and Collective Intelligence. Science

[14] Frontiers ONLINE NO. 66 Nov-DEC 1989 ARMY ANTS: A COLLECTIVE INTELLIGENCE?

[15] David Gordon (2003) Collective Intelligence in Social Insects Introduction & Self-Organisation

[16] Prusinkiewicz, P. and Lindenmayer, A. (1990) The Algorithmic Beauty of Plants, Springer Verlag.

[17] Langton, Christopher G. ed. (1989) Artificial Life Addison-Wesley, Reading, Massachusetts

[18] Holland, John H. 1998. Emergence: from Chaos to Order. Perseus Books, Cambridge, Mass. p. 251.

[19] P. Eleni, A. Turner, R Thum (GA 2002) Interacting Unities : An Agent Based System

[20] Greg Lynn (June 1, 1989) Folds, Bodies and Blobs: Collected Essays La Lettre Volee; 0 edition

[21] Manuel de Landa (September 18, 2000) A Thousand Years of Nonlinear History : Zone books reprint Edition

[22] *C. Soddu, E. Colabella (1994)* The design of morphogenesis. An experimental research about the logical procedures in design processes Paper

[23] Heylighen F. (1992) : "Evolution, Selfishness and Cooperation", Journal of Ideas, Vol 2, # 4, pp 70-76.

[24] Megan Yakeley (April, 1993) Deconstruction for Everyone: The Myth Exploded

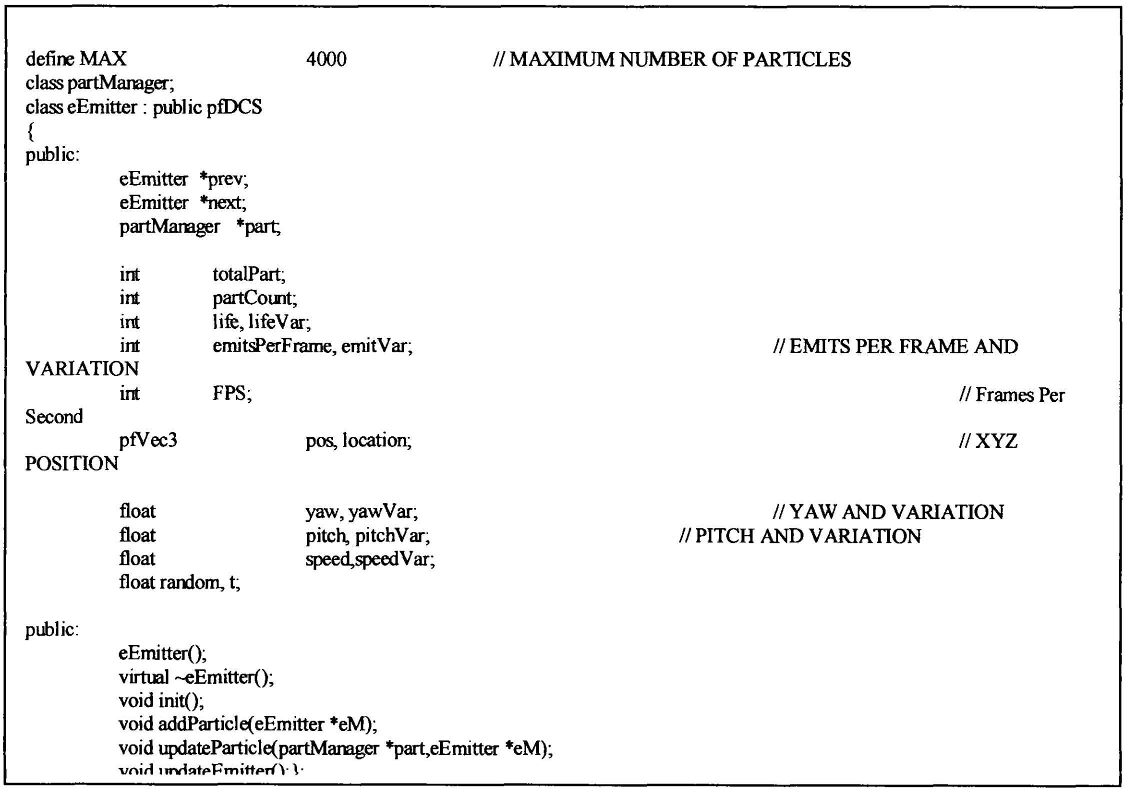
[25] Peter Testa MoSS: Morphogenetic Surface Structure A software Tool for Design Exploration

[26] William R. Corliss (Dec 1989) Science Frontiers #66

[27] Maes, P. (ed) (1991a), Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, London: The MIT press.

Appendix A

Class eEmitter



```
define MAX                    4000                    // MAXIMUM NUMBER OF PARTICLES
class partManager;
class eEmitter : public pfDCS
{
public:
          eEmitter  *prev;
          eEmitter  *next;
          partManager  *part;

          int        totalPart;
          int        partCount;
          int        life, lifeVar;
          int        emitsPerFrame, emitVar;                      // EMITS PER FRAME AND
VARIATION
          int        FPS;                                                 // Frames Per
Second
          pfVec3            pos, location;                                // XYZ
POSITION

          float             yaw, yawVar;                          // YAW AND VARIATION
          float             pitch, pitchVar;              // PITCH AND VARIATION
          float             speed,speedVar;
          float random, t;

public:
          eEmitter();
          virtual ~eEmitter();
          void init();
          void addParticle(eEmitter *eM);
          void updateParticle(partManager *part,eEmitter *eM);
          void updateEmitter();};
```

Header File For Class eEmitter

```
// eEmitter.cpp: implementation of the eEmitter class.
//
//////////////////////////////////////////////////////////////////////
#include "headerFiles.h"
partManager *particlePool;
//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////
//unsigned seed = time (NULL);q
//                              srand(seed);
void eEmitter::init()
{
        int loop;
        particlePool = new partManager;
        partManager *last = particlePool;;
        for (loop = 0; loop < MAX - 1; loop++)
        {
                partManager *pm = new partManager;
                particlePool->next = pm;
                pm->prev = particlePool;
                particlePool = pm;
        }
        particlePool = last;
        setTrans(0.5f, 0.0f, 0.0f);
}
eEmitter::~eEmitter()
{
}
eEmitter::eEmitter()
{
        eEmitter *eM;
        eM = this;
        eM->prev = NULL;
        eM->next = NULL;
        eM->totalPart = MAX;
        eM->partCount = 0;
        eM->life = 4;
        eM->lifeVar = 10;
        eM->emitsPerFrame = 1;
        eM->emitVar           = 10;
        FPS = 0;
        eM->part    = NULL;
}

void eEmitter::addParticle(eEmitter *eM)
{
        eM = this;
        partManager   *npart;
        unsigned seed = time(NULL);
        srand(seed);
        random = (float)rand()*10;//.0f
        float yaw,pitch,speed;
        if (eM != NULL && particlePool != NULL && eM->partCount < eM->totalPart && eM->partCount <
population)
                {
                        npart = particlePool;                           // THE CURRENT PARTICLE
                        particlePool = particlePool->next;          // FIX THE POOL POINTERS
                        if (part != NULL)
                                part->prev = npart;                     // SET BACK LINK
                        npart->next = part;                             // SET ITS NEXT
POINTER
                        npart->prev = NULL;                             // IT HAS NO BACK
POINTER
                        part = npart;                                           // SET IT IN
THE EMITTER
                        npart->init();
                        npart->m_geometry->setSize(0.3f, 0.3f, 0.3f);
                        npart->locate(location);
                        addChild(npart);

                        // CALCULATE THE LIFE SPAN
                        part->life = eM->life + (int)((float)eM->lifeVar * pfuRandomFloat()* 15);
                                //cout << "random =  " << random/100000;
                        eM->partCount++;                                // A NEW PARTICLE IS
BORN
        }

,
```

4.3 Implementation

The system is tested through a series of experimental runs; in each run varying conditions are derived from the manipulation of parameters that influence relations between different components of the system. The aim is to simulate different sets of conditions in the system for the purpose of comparison.

The following sets of results presented from two runs of the simulation are based on random motion in different dimensional axes.

The first run simulates the motion of the particles in two dimensions (X and Y), particles start out in a specified location the centre of the grid are allowed to roam in different directions at a constant speed. A particle is transformed when it comes within proximity of another particle. The physical properties of a particle are determined by the value of its threshold and its weight. The weight translates to the radius of influence of energy surrounding the particle. The manner of interaction (fusion) with a nearby particle is determined by the value of the threshold. For this simulation the value of the threshold is set over a range of positive values to produce only a fusing effect between adjacent particles. Each particle is given a life variance value, which determines how long the specific particle is present for before it dies. Adjusting this value has the effect of speeding-up or slowing down the speed of the creation of particles in the system. Adjustment of this variable to a high value, produced a somewhat chaotic effect in the system as shown in figure??

The geometric primitives representing the particles are polled from a pool created by the emitter class; the modularized structure of the program makes it easy to change the type of primitives used by the system in any iteration. Although for this run of experiments only the isomorphic poly-surfaces were used.

The movement and behaviour of the particles for the simulation using the grid was based on a simple agent model. The movement was forwards and restricted within the grid. The change of direction is based on the line of sight; the agents are programmed to look around randomly and change their direction based on the longest line of sight observed. The pseudo code below illustrate the mode of movement of each agent

A default threshold value of 1.0f is set for each particle at the start of a run of the simulation; this produces a fairly stable result as seen in the image below,

particles tend to roam further maintaining their original attributes. Interactions are brief and less frequent, resulting in the creation of isolated clumps of amorphous forms distributed over the grid landscape. An increase in this the threshold value increases the radius of influence surrounding each particle and thus increasing the possibility of interaction with a close by neighbour. This results in less disparate creation of form over the landscape as particles tend fuse over a greater space. This result in bigger clumps of forms being created and with fewer isolated forms created