Y. Ikeda, C. M. Herr, D. Holzer, S. Kaijima, M. J. Kim. M, A, Schnabel (eds.), *Emerging Experience in Past,Present and Future of Digital Architecture, Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA 2015,* 000–000.

ISOMORPHIC AGENCY

A digital environment for agent based modelling

IGOR PANTIC[1], SOOMEEN HAHM[2]
1 www.igorpantic.net, London, UK, igor@igorpantic.net
2 www.soomeenhahm.com, London, UK, s.hahm@soomeenhahm.com

Abstract. This paper deals with the topic of agent based design systems and their application in the process of generative design, with the goal of creating a series of volumetric arrangements of varying surface qualities. To test this approach, we created a custom made tool for Agent Based Modelling - Isomorphic Agency - whose design and functionality are described in this paper. Furthermore, we conduct a series of design exercises which combine different parameters, examining the resulting geometries and underlying organizational principles.

Keywords. multi agent systems; generative design; design research

1. Introduction

We strongly believe that agent based design and distributed design intelligence have an important place in the contemporary architectural discourse. This design approach, which is inherently algorithmic, allows for the creation of a myriad of highly differentiated outputs, rich in information and complexity, which would be impossible to create by other means of production.

In this paper we discuss the application of multi-agent systems as a form-finding and geometry generating tool for the creation of highly complex volumetric formations with varying porosity of enclosure. This is done through the break-down of an ongoing research project and a custom made software in Processing, titled Isomorphic agency (abbreviated IsA) - designed for the use within the Agent Based Design Systems.

2. Multi-agent systems in the context of architectural design

Agents, intelligent computational objects, with the ability of autonomous decision making, and Multi-Agent Systems (MAS), have been used in numerous disciplines, from description of complex biological and social behavioural models (Craig Reynolds' flocking model), to complex problem solving in computer science and agent-based computational economics. Since the late 90s, their potential has also been recognised in the field of computational architecture and design (Krause, 1997; Coates and Schmid, 1999), and in recent years, MAS have been used in number of ways in the process of design and simulation (Baharlou and Menges, 2013). According to Lyon (2004), due to their capacity to capture information about objects and ability to self-organise and make decisions through the use of internal protocols, multi-agent systems are seen as an adequate tool in the design processes. As observed by others (Leach, 2004; Maxwell and Pigram, 2010, Stuart-Smith, 2012), this kind of approach allows for the creation of programmable matter, establishing complex forms of higher order.

Notable research on this topic can be found in the works of practices such as Biothing/Alisa Andrasek, Kokkugia (Snooks and Stuart-Smith) and Supermanoeuvre (Maxwell and Pigram), which use agent based design systems in their work. Here we can distinguish several main approaches (which are sometimes combined) in which agents are used to generate architectural form, such as agent aggregations (agents represented in the form of discrete geometrical components), fibrous assemblages (traces of agent movement are converted into linear geometries and bundled assemblages), mesh deformations (initial meshes or forms are deformed through the influence of agents) or surface articulations (where agents are used on top of existing meshes to achieve further articulation or ornamentation).

3. Research goals and approach

In this context, we are interested in using Multi Agent Systems as means to create highly differentiated structures with complex geometries. In doing this, we approach this topic from several different aspects and design methods, which are later applied in IsA software. We argue for the design system which allows for generation of results that can answer to different design and performance criteria, such as density of generated structure, porosity of outer shell/mesh or space making properties.

With this in mind, our goal is to produce an agent based design system for creation of new geometries and volumes, which have the capacity to form architectural space. In this research we are not dealing with the capacity of the agents to mutate and deform previously generated objects and surfaces or to achieve structures through self-organisation and aggregation of predefined geometrical elements. Likewise, our goal was to create a software (application) whose functionality and structure allow it to be used in number of ways for different design methods - thus allowing for a wide variety of design outputs and exportable data types (meshes, lines, point clouds...)

To achieve this, we propose a computer program which generates space and forms by specifying the behaviour and mutual relationships of its functional elements. We set up an elaborate simulation environment in Processing, controlled through a central interface with a set of precisely defined parameters (Figure 1). Initial output can be directly viewed in the program itself, or it can be exported as files for various rendering applications.

The project itself is trying to expand on the existing agent-based agenda by focusing on environmental interaction as the key component of the system, and properties of the resulting three-dimensional spaces. In addition to this, we incorporate a physics library, which allows us to address questions of structural stability.

Finally, we argue for the openness of this system in a way that it can function and produce high-quality results with drastically different agent populations (from as little as 10 up to thousands), as well as not to rely solely on the extremely high numbers of agents and super-computing, but rather to have the richness of a result dependant on the complexity of scripted behaviours.

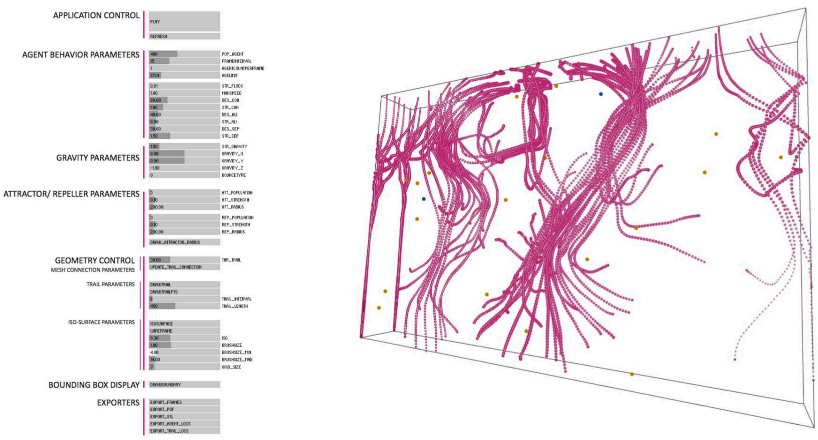

Figure 1. Example of the running script with GUI (left) and simulation in progress (right)

4. Design methodology and organisational principles

Generative logic results from embedded agent behaviour, interaction between different groups of agents, and their interaction with the environment. Agents are able to read the environmental data and react to it.

We are examining the 3 main parameters which influence the organisational principles:

- Agent behaviour
- Environmental influence
- Time component

Through the combination of these parameters, we try to create multiple layers of behaviour control at different scales - macro formations representing architectural spaces (macro volumes) and micro formations representing further surface articulation.

4.1 AGENT BEHAVIOURS

Agent behaviours are governed by several programmed functionalities based on the steering behaviours by Craig Reynolds. Main functions include flocking (cohesion, alignment, separation), target seek, follow the target, random walk - all of which affect immediate interactions between agents. As the basis for the steering and physics behaviours, we are using and extending Toxiclibs Processing library by Karsten Schmidt (2011, www.toxiclibs.org).

4.2 EXTERNAL INFLUENCES / INFLUENCE OF THE ENVIRONMENT

The behaviours are to be influenced by the environmental input, attractors and repellers and other external forces, which influence the agent movement and trigger or limit certain behaviours.

In this process we can distinguish two main types of external influences:

- Agent behaviour influenced by attractors or repellers
- Agent behaviour influenced by agents location in space / zone

4.2.1. Attractors and repellers

The IsA software allows for the import of point clouds which can be assigned to attractors/repellers. The primary role of attractors and repellers is to direct the agent movement (Figure 2), in order to create space/voids/enclosures. In addition to this, attractors and repellers can be used to trigger or limit certain behaviours once the agent is within the radius of attractor/repeller influence. These points can be static (locked), have animated behaviour (move within the scene) or have behaviours of their own (be agents as well).

It is important to note that in this approach, the attractors and repellers can be considered as poly-scalar constraints and drivers, that is, having the ability to influence macro / spatial-formations as well as micro / surface ornament formations. In addition to this, observing them as additional types of agents only adds to the complexity of the system.

4.2.2. Influence of the zone

Similar to how attractors and repellers can affect agent functionality, behaviours can also change when agents enter a defined zone of space. Functions turn on/off or values of the parameters change when agents reach a certain zone inside the computed world (Figure 3).

This method allows for combinations of different influences - i.e. attractors that affect the movement aspect, and agent location that affects functionality.

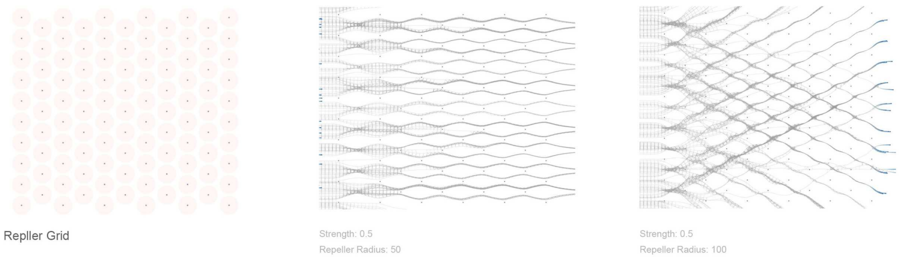

Figure 2. Different affects of attractors and repellers on agent trajectories

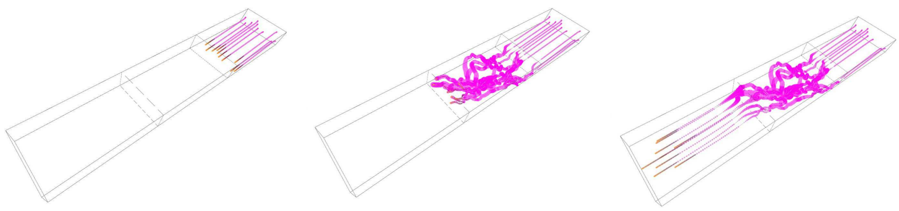

Figure 3. Agent behaviour affected by the zones of influence

4.3 TIME COMPONENT

Furthermore, we see time as an important components that governs operations in the system. Simulations are performed with a constant population of agents, or a population which is changing - continuously growing or dying out. In regards to this, we can determine if the agent population is:

- Constant - agent population is set before the simulation is run, and the number of agents remains the same throughout the simulation
- Increasing - new agents are created after a set number of frames. The user decides how many agents will be added, as well as what is the maximum total population of all agents in the scene.
- Dying / agent age limit - all agents have life spans which last for a certain number of frames. After the age limit has been reached, the agent disappears from the scene.

Similar to agent life span, agent behaviours and functions can be triggered periodically, or switched on/off after a certain period of time.

5. Resulting geometries

Through this process we are establishing three main types of generated geometry (Figure 4): agent trails, connections between these trails and iso-surface. This type of geometry classification allows for control on different levels and layers of creation.

5.1 AGENT TRAILS

Agent trails represent the basic form of geometrical output produced by IsA application. Main geometry guidelines are produced from agent trails, which describe the path of the agents. The resolution of the trail and the length of the trail can be numerically controlled. The resolution of the trail depicts the interval in which the trail control points are dropped. As the interval decreases, the resolution increases, and the representation of the trail becomes finer. This parameter directly influences the density of connections between the trails.

5.2 CONNECTIONS BETWEEN AGENT TRAILS

Trail points connect to neighboring trails to create thread-like meshes. On the most basic level, it works by drawing a line from an agent trail point to a number of points from the neighboring trails which are within the set threshold. The lower the threshold parameter becomes, the fewer connections will be created. By controlling this number, we have direct control over the porosity of the generated geometry.

We consider the connections between the trails to be the main type of generated geometry, due to it's ability to take the shape of any assigned/preprogramed geometry. This means that connections between agent trails can be in the form of the mentioned lines (Figure 4), surfaces (connecting multiple points between the neighboring trails or lofting between the previously created linear connections), and prescribed volumes, which are to be generated in place of linear and surface connections. Applying this technique further distinguishes created geometries from similar fibrous-like agent generated formations.

All three types of generated connections are further subjected to influence of the agents in the level of micro-formations/surface articulation.

5.3 ISO-SURFACE

Iso-surface is generated around the previously created geometry - trails and trail connections. We have the ability to control several topological traits of the Iso-surface, such as the resolution and the grid size of the surface, as well as control over the surface brush size.

Brush size, which operates between the set minimum and maximum range, determines the size (diameter) of the ISO-surfaces created around the linear geometry. This parameter, together with its minimum and maximum values can be connected to the influence of attractors/repellers and zone of the simulated world. Considering the importance which influence of external forces has on the resulting geometries has for us, this is a key attribute of the ISO-surface in the framework of IsA. Likewise, in addition to controlling the size of iso-surface cross section inside the IsA, this data can be exported as a .txt file and used in other purposes to enhance the richness of the design.

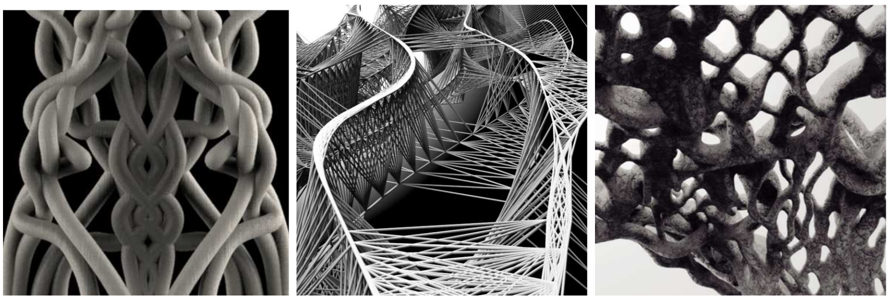

Figure 4. Resulting geometry types - agent trails, connections between trails and iso-surface

6. Importing geometry and other data

We allow for the number of geometrical types to be imported into the script, in the form of points, point clouds, lines and .obj mesh files, created using any other 3d software.

As previously mentioned, points can be used as attractors/repellers or as the origin/seed points for agent creation. This similarly applies to lines/ splines, which in addition to this, can become the initial roots for the future geometrical network. Likewise, as seen in comparable projects, which deal with the application of Multi-Agent Systems, importing a mesh allows agents to build geometries around, beside and inside the existing mesh, or add finer levels of surface articulation to the imported mesh. However, as previously mentioned, this feature is not the main focus of our research, as Isomorphic agency's primary goal is the creation of geometry from scratch.

7. Design studies

The previously explained ideas, which establish a specific approach within the agent based design methodology, were tested and demonstrated through our research, educational and practical works. The following examples present a series of design studies performed by altering parameters of agent behaviours and influence of external forces. They show how the system can be adapted to control, among other things, the formation of enclosed spaces and the density of output geometries. By applying agency on multiple scales and layers, from volumetric formations to local surface conditions, we were focusing on the emergent spatial and ornamental qualities of the resulting heterogeneous meshes.

7.1. DESIGN STUDY 1

The first design scenario deals with the control of agent movement with attractors and repellers resulting in a series of elegant and intricate patterns. Agent trails are used as main geometry drivers, and the lines are exported and further used to create meshes in other applications, such as Rhino/Grasshopper.

The initial setup for each of these tests consisted of an array of attractors or repellers generated independently and imported into Processing as a text file (Figure 5). We varied the density of attractor setups, as well as the involvement of complex behaviours. In general, each set-up favoured one of the parameters. The denser and more controlled the attractor/repeller set-up was, the less influential the agent behaviour would be, and vice versa.

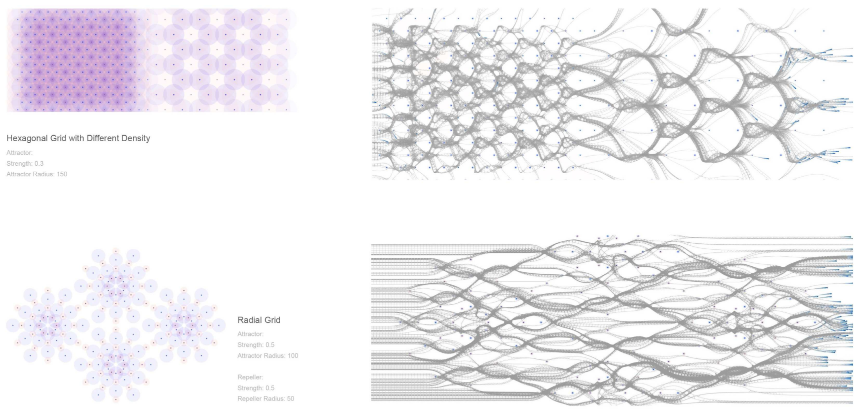

Figure 5. Testing scenario 1 - agent trajectories patterns

7.2. DESIGN STUDY 2

Similar to the previous example, this study deals with the control of agent movement with attractors and repellers in order to create enclosed and usable space. Strength of the repelling force is used in combination with the constant one-directional force, pushing agents around the repellers, creating pockets of space.

However, unlike in the previous example, this was used to a lesser extent, as the main focus of this exercise was on geometries built in between the trail lines and control of their density. The results were less uniform and more volumetric in nature (Figure 6). Likewise, attractors and repellers were also used to trigger behaviours in addition to directing the movement of the agents.

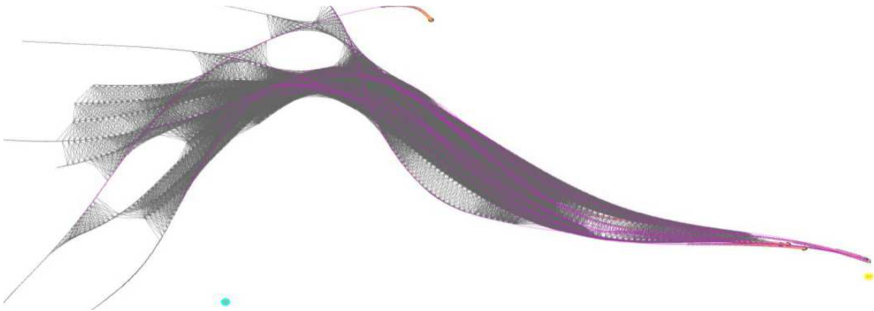

Figure 6. Testing scenario 1 - agent trajectories patterns

7.3 DESIGN STUDY 3

This design study shows the influence of different environmental setups on agent behaviour. The testing box (computational world) was divided into 3 different zones (Figure 7). Inside zones 1 and 3 the agents act as particles directed only by a one directional force, while in the middle zone 2, agent functionalities are turned on.

Likewise, the zone of influence also affects the creation of the 3D iso-surface by controlling the iso-surface brush size (Figure 7).

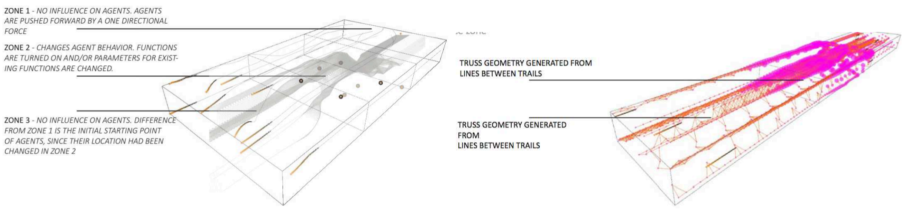

Figure 7. Testing scenario 3 - agent behaviours affected by zones of influence

This expands on the previous studies where the attractors were used as the functionality triggers. In this case, we are able to dedicate these properties to entire zones, while still using attractors and repellers to direct the movement of the agents.

8. Conclusions and future work

We see the current stage of the project as a phase of extended research into application of Multi Agent System in an architectural setting. The aim is to test the application in a real-life scenario, while exploring the additional means of translating agent movement into a structural system or a system of surfaces. Extruding along an agent's path and connecting between the paths are only a few of many possibilities that this kind of system offers. Further research on geometry would test conditions such as lofting between paths or replacing the lines between trails with objects.

Further research on behaviours would include the introduction of multiple species of agents, each with their own attributes and goals. Likewise, in order to expand the body of research beyond the current point, further work is to be done on creating environments which invite more interaction.

Acknowledgements

The script has been tested during the algorithmic design workshop 'Complex Morphologies' directed by I.Pantic and held at the Faculty of Architecture, University of Belgrade, Serbia (2014), as well as with the students at the Bartlett GAD, Research Cluster 6, directed by D.Widrig, S.Hahm and S.Bassing, (2013/2014 school year). All of the presented material has been produced by I.Pantic and S.Hahm or in collaboration with the students from the mentioned institutions, using the Isomorphic Agency (IsA) - application for Agent Based Modelling script.

References

Baharlou, E. and Menges, A.: 2013, Generative Agent-Based Design Computation: Integrating material formation and construction constraints, *in Computation and Performance, Vol.2 - Generation, Exploration and Optimisation, Proceedings of the 2013 eCAADe, Delft University of Technology, Delft, 165-174*

Coates, P. and Schmid, C.: 1999, Agent Based Modelling, *in Architectural Computing from Turing to 2000: Proceedings of the 1999 eCAADe,* Liverpool, 652-661

Krause, J: 1997, Agent Generated Architecture, *in Design and Representation: Proceedings of the 1997 Association for Computer Aided Design in Architecture (ACADIA),* Cincinatti, 63-70

Leach, N.: 2004, Swarm Tectonics, *in* N. Leach, D. Turnbull, and C. Williams (eds.), *Digital Tectonics,* Wiley, London, 70-77

Lyon, E.: 2004, Artificial Intelligence and Emergence in Architecture: A multi-agent based model for design processes, *in Proceedings of the 8th Iberoamerican Congress of Digital Graphics,* Porte Alegre, 404-406

Maxwell, I. and Pigram, D.: 2012, Algorithmic Typology: Towards an Operational Model, *in* N. Leach, and P. F. Yuan (eds.), *Scripting the Future*, Tongji University Press, Beijing, 107-112

Maxwell, I. and Pigram, D.: 2010, Supermanoeuvre - Inorganic speciation: Matter, Behaviour and Formation in Architecture, *in* D. Kottas (ed.), *Contemporary Digital Architecture: Deisgn & Techniques*, Links International, Barcelona, 208-227

Reas, C. and Fry, B.: 2007, Processing: A Programming Handbook for Visual Designers and Artists, MIT Press, Cambridge.

Schmidt, K.: 2011, Toxiclibs. <www.toxiclibs.org>

Stuart-Smith, R..: 2012, A Matter of Organization: Re-emphasizing Organization *in* N. Leach, and P. F. Yuan (eds.), *Scripting the Future*, Tongji University Press, Beijing, 157-162