Evolutionary Swarm Design

How can swarm-based systems help to generate and evaluate designs?

von Mammen, Sebastian; Novakowski, Scott; Hushlak, Gerald; Jacob, Christian

Introduction

We consider design an act of intellectual craftsmanship which follows a holistic approach to conquer our arising challenges in engineering or architecture, or to simply enrich our lives culturally. As such, artists, designers, architects, engineers, and computer scientists drive the development and exploration of novel design tools and methodologies. Creative options offered through new tools become the catalysts that define new thinking and invention. In tandem, new tools result in new knowledge. The capacity of the computer to help with the conceptualization and organization of design space redefines the meaning of the tool from submissive server to an affirmative and creative partner. Together the designer and computer dream and play in an attempt to find the mindset for design that will subsequently lead to pragmatic solutions. Through steady technological advancements, computer-aided design has surpassed the idea of a three-dimensional drawing board. Instead, when provided with basic procedural and structural building blocks, an algorithmic system can generate solutions to predefined problems by itself. This approach allows the designer to focus on the identification and formulation of contextual interdependencies that drive the design process. The corresponding computational concepts are known by many names, for instance as “planning systems” in the discipline of artificial intelligence (Russell, Norvig et al. 1995) or as “computational developmental models” (CDMs) in the context of biological modeling (Kumar and Bentley 2003). Through the integration of CDMs with ideas gained from studies of complex systems, novel design tools are now ready to be as expressive and powerful as they are applicable to real-world design tasks.

In this article, we address this challenge with the presentation of a bio-inspired design framework that combines swarm grammars (a swarm-based CDM) with the means of evolutionary computation. The next section briefly introduces artificial swarms as a model for complex systems. The third section introduces swarm grammars that are then subjected to computational evolution. Two additional sections are dedicated to a discussion of the significance of CDM and evolutionary computation for design processes. Before we conclude this essay, a visionary scenario of a swarm grammar-supported design process is presented, which addresses potential future work.

Swarms as Complex Emergent Systems

Emergence is the phenomenon of a system exhibiting features that cannot be inferred from any of the features of its constituents alone (Holland 1998). Instead, the inherent interaction networks of the system's parts are responsible for —in many cases, unexpected— global effects. Systems that exhibit emergence are also called complex. Through closer analysis of numerous natural and man-made complex systems, it was realized that their constituents are often interconnected according to a power-law degree distribution (Hidalgo and Barabasi 2006). This means that the number of units that are interconnected with few others is very large, whereas the number of constituents with many connections is very low, and the transition in-between decreases according to a power-law. Examples of complex systems in which such degree distributions could be measured are social networks and gene regulatory networks (Barabasi and Albert 1999). Although it is still controversial which exact mathematical laws generally govern the interconnectedness in complex systems (Bader 2006), one may assume that their inherent topology plays a key role in the occurrence of emergent phenomena.

In schools of fish, nests of ants or flocks of birds, large numbers of individuals interact and intricate interaction patterns emerge. In 1987, "boids" were introduced as a simple model to graphically simulate basic flocking behavior (Reynolds 1987). Flocking patterns can emerge when each swarm individual only adjusts its acceleration with respect to its neighbors. Originally, three distinct urges determined the agents' behaviors, and thus possible flock formations: (1) Alignment, an orientation towards the average direction of the agents' neighbors, (2) cohesion, an urge towards the center of its neighbors, and (3) separation, to maintain minimal distances from each other. Several variations of this model have been investigated in order to identify the local behaviors responsible for emergent flight patterns (Huepe and Aldana 2008). The symmetry of noisy, random movement can, for example, be broken solely by the individuals' urges to align according to their locally perceived peers (Vicsek, Czirók et al. 1995).

Two aspects render artificial swarms an especially interesting model for investigations of complexity in two- or three-dimensional space. (1) A simple boids flocking model is purely based on spatial states and interdependencies: The individiuals' states are expressed through their locations, their perception is spatially limited and their actions result in changes of their locations. (2) The spatial movement of swarm individuals continuously re-configures the systems' interaction networks, rendering swarms as highly dynamic complex systems (von Mammen and Jacob 2008).

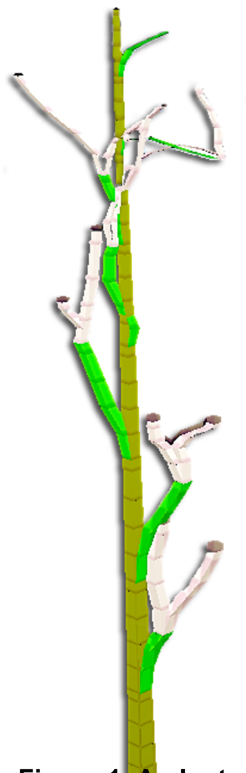


Figure 1. A plant-like swarm grammar structure.

The Conceptual Evolution of Swarm Grammars

In swarm grammars, individuals agents flock, reproduce and place construction elements in three-dimensional space. Thereby, swarm grammars combine the dynamic complexity of swarms with the abilities of developmental models (Prusinkiewicz and Lindenmayer 1996).

The concept of our swarm grammars has evolved in three steps. In its original version, swarm individuals (agents) leave continuous particle traces in space and periodically proliferate in accordance with a set of reproduction rules (Jacob and von Mammen 2007). This behavior results in unbroken, branched structures resembling plants (Figure 1). Reproduction also allows for specialization. A substitution A → BC is an example of a reproduction rule, which we use in our swarm grammars. An agent with type-A flocking behavior is substituted by two agents of types B and C. These rules are also referred to as grammatical rules to emphasize their affinity to formal grammars (Chomsky 1956).

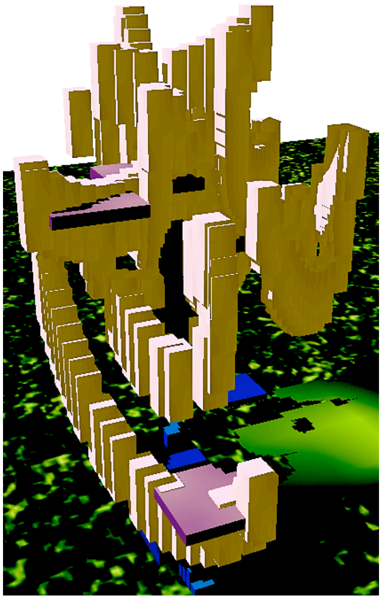


Figure 2. An example of a swarm grammar architecture.

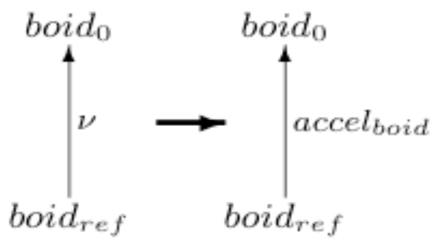
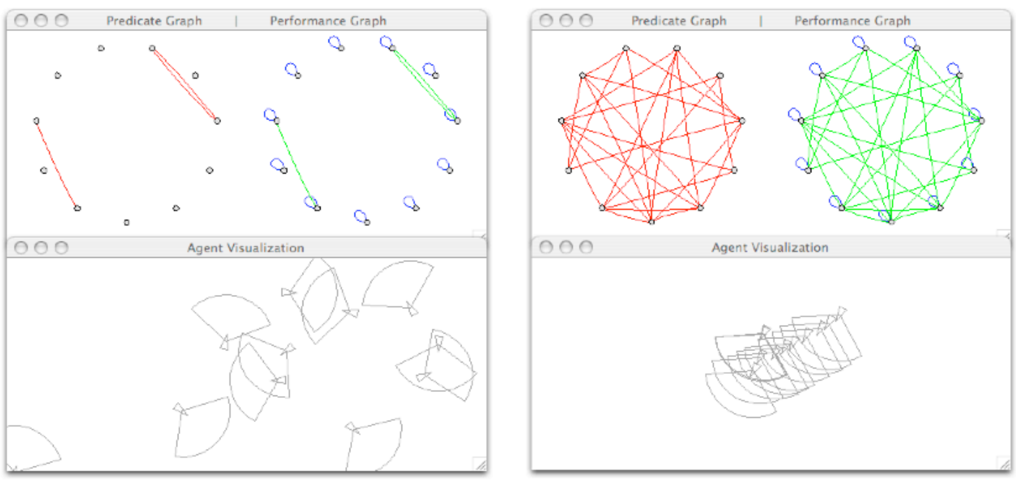


Figure 3. A boid agent reacts on a neighbor relation through acceleration.

Since reproduction, differentiation and construction do not happen continuously in real swarms (for example, in social insects), but can be induced through external stimuli such as food supply, pheromone smells or construction configurations (stigmergic signals), we extended the swarm grammar model accordingly. Perceived events as well as construction and reproduction activities were integrated into the agents' behavioral rules (von Mammen and Jacob 2008), which led to the construction of architectural



(a) (b)

Figure 4. (a) The upper graphs show the perception and the reaction relations among the swarm individuals that are spatially represented below. (b) At a later stage of the simulation all individuals have clustered, as illustrated by the clustering relationship graphs.

structures governed by the interactions of swarming agents in 3D space.
In our most recent swarm grammar implementations, agent-nodes are graphically interconnected through relational edges (Figure 3). This graphical representation allows us to model any relations among agents. The agents' behavior is again encoded in grammatical rules. However, instead of substituting strings of symbols, an agent acts through replacement of contextual graphs that describe its relations to the environment, as inspired by the computational concept of relational growth grammars (Kniemeyer, Buck-Sorlin et al. 2004). Interaction graphs illustrate the changing relationships among interacting swarm agents over time (Figure 4).

Examples and Design Methodologies

Before discussing the significance of swarm grammars for design, we present several examples, detailing their emergence and possible conceptual extensions.

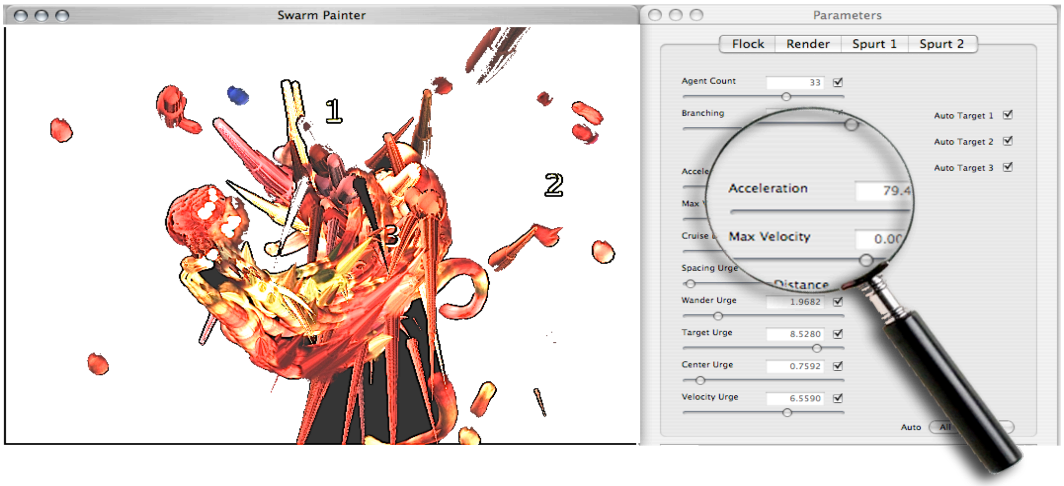


Figure 5. Drawing canvas and user interface of Swarm Painter.

The completion of the first swarm grammar implementation motivated us to manually configure swarms flocking in 3D space and to study how we could achieve design variations (Jacob and von Mammen 2007). For instance, we investigated the impact of different grammatical rules, the impact of various sets of flocking parameters and the possibility of the swarm argents to interact with their environment. Interacting with artificial swarms (for example, through video cameras and motion detection software) in a seamless manner has been driving our interactive swarm art installations for a long time. The newly formalized constructivity of swarm grammars was soon to be integrated into the visual design tool *Swarm Painter* (Figure 5). Swarm Painter is an application that provides an artist with intuitive means to influence the swarms' flocking formations and to utilize them as "brushes" in different simulated media.

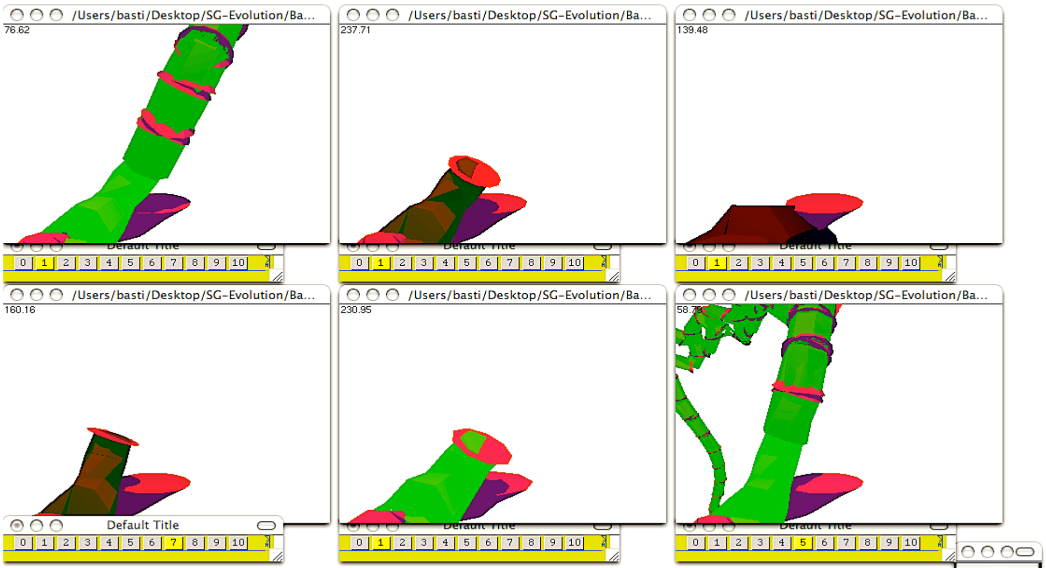


Figure 6. Interactive Evolution: The breeder rates a population of swarm grammar structures and lets the next generation compute.

Figure 7. Immersive Evolution: Like a gardener, the designer roams in 3D space to breed Swarm Grammar structures (von Mammen and Jacob

an an er to ve ad s.

(Figure 7).

Experiencing the unfolding space of innumerable variations of swarm grammar sculptures, we decided to narrow down the search and formulate specialized design tasks (von Mammen and Jacob 2008). In order to run an evolutionary search through the space of possible swarm grammar structures, a fitness function was required to rate the emergent structures. In addition, the

(a) (b) (c)

Figure 8. Automatically evolved "swirly" architectural idea models generated by swarm grammars.

interaction processes that yield the built structure, or the corresponding underlying behavioral programs, can then be investigated. Starting off with a simple volumetric measurement of the building and measurements of interaction during the construction process, we were able to promote diverse and interesting architectural design models (Figure 8).

Socio-Economic Design Responsibility

Complex, nature-inspired design elements have been a well-established part of post-modern architecture and design since the 1990s (Flagge, Schneider et al. 2004). Yet most of these examples have documented that form prevailed over function and true complex pretensions. We claim that a truly complex approach to design presupposes a theoretical foundation to address complex socio-economic design challenges. We present swarm grammars as a formalized means to model complex systems and to numerically generate complex design solutions, hence providing a basis for theoretical investigation of design through formalized mathematical models.

Relying on the swarm metaphor for the generation and analysis of complex systems has become a more widely pursued practice in recent years (Resnick 1997; Bonabeau, Dorigo et al. 1999). We accredit this trend mainly to the spatiality, the dynamic interaction topologies and the easily discoverable emergent pattern formations in swarms. Exactly these features are also of great importance for modeling and design processes: The role of the designer transcends isolated disciplines such as technology, engineering and marketing. The designer's viewpoint ideally expands beyond the accountant's desk and considers the immediate use for the client as well as the long-term impact of a new product (if we may rely on the example of consumer-oriented product design here). But how, indeed, can the designer gain a long-sighted viewpoint? Merely contemplating about the possible social and economic impact of novel products falls far short as the structures of the (post-)modern world are too complex and interwoven. The designer has to get a notion about the possible advantages and risks. One has to systematically 'project' the impact of the new product on the society as a whole. During this process, one can single out and analyze unexpected phenomena and their emergences and relate them to the product designs. Numerical, computer-based simulations support this approach.

Numerics for Complex Design

Computational evolutionary systems—i.e, defining design through fitness functions that can be easily massaged—offer the potential for breeding designs that successfully cope with complex requirements. The architect, often considered a singular authority arbitrating particular taste and branding, can use evolutionary computing to proactively engage the client in the participation of the design. At the same time, sociological, ecological and economical factors can be considered in a semi-automatic, evolutionary search process for complex designs. Like the French painter Mattisse, referring to the unconscious as the third hand or partner helping him to make his paintings, the designer can actually go beyond a helping hand and even quantify and measure decisions made by the evolutionary partner.

Advantages of a progressive future need to be judicially weighed against risk. Therefore, in a post-modern world of incredible complexity, it seems appropriate to tender lucidity that is not compromised. Because some say that "time is money" the abandonment of preciousness in exchange for enhanced options does not have negative implications. In a computational evolutioanary setting, previous iterations can be summoned at any time, from which a new investigative design branch can be started. Design failure in an iterative computer process wastes the time of the machine as opposed to the designer's time.

Various challenges render numeric approaches in the context of socio-economic simulations less attractive than they deserve. First, there has always been and will always be a lack of computing power. By ignoring the factors which are relevant to emergent phenomena in complex systems, we could simply base our computations on each and every building block of the world that we have measured, from buildings and bricks down to the level of molecules. In this context, knowing too much is a curse! It renders the resulting computational model incomputable. The alternative, an abstraction from reality, the formulation of theoretical models, might risk the loss of important relations among the involved agents—after all, emergent phenomena are those that we haven't expected based on our limited insights into the dynamics of the system under investigation.

But even if the designer wants to establish an elaborate socio-economic simulation, which computational design tools would be available and suitable for modeling? Surely, there are many programming environments that support various modeling intents. For instance, there are agent-based modeling frameworks (Klein 2008), libraries for machine learning algorithms and for the analysis of vast numerical data (Kohavi, Sommerfield et al. 1996), physics engines (Smith 2007), and visualization frameworks (Junker 2006). It turns out that the majority of computational frameworks are not universal in respect to the simulated subject matter; they do not provide the basis for different analytical methodologies and they force the designer to comply with very restrictive conventions of data representation.

We believe, the designer should be given the possibility to focus on an intuitive description of the agents and the relationships involved in a model. Swarm grammars are being developed to particularly address this need through their underlying graphical representation in combination with the formation of interaction networks as the computational step. Their graphical representation provides an intuitive and uniform modeling language and immediately extends the scope from local interactions to global emergent effects. Swarm grammars hold the promise to be applicable to modeling and design tasks of various scales and scientific fields.

Beneath the Projective Design Approach

Despite the fact that computational developmental models grow structural patterns that resemble, for instance, spreading bacterial colonies (M. Hoar, K. Penner et al. 2003; Penner, Hoar et al. 2003) and blossoming plants (Prusinkiewicz and Lindenmayer 1996), the generative character of the respective models does not imply autonomy of the performing machine. Like in other simulations of natural phenomena, for example the weather forecast, the underlying algorithm is based on a theoretical model. On top of the theoretical model, the programmer has to make numerous algorithmic design decisions, e.g. how to balance simplicity versus comprehensiveness and efficiency versus accuracy. The resulting algorithmic design is also determined by the available hardware, by the programmer's experience, by the expressiveness of the utilized coding language, and, most importantly, by the simulation's investigative purpose. Obviously, the simulation itself is just as much the product of an elaborate, theory-driven design process as the artefacts produced within the created virtual realms.

The different breeding strategies described earlier are applicable to all kinds of design processes. The concept of evolutionary computation adds to the autonomous 'feel' of a software program, for in evolutionary developmental (evo-devo) models two stages of biological development are combined: the long-term adaptive processes through evolution and the 'unfolding' process that generates living organisms through morphogenesis. In the case of swarm grammar evolution these two stages are: (1) The simulation of swarm activity resulting in structural pattern formation, and (2) an evolutionary process that changes the swarm grammar configurations over time. Although evo-devo systems might be considered even more independent of human input compared to sole pattern generation simulations, their additional autonomy requires numerous supplementary design decisions.

We need to emphasize that man-made design does not necessitate the immediate conceptualisation of a desired product. Instead, a generative, numeric approach first requires the identification and description of the goals, the requirements and the available means. After several iterations of adjustments to a resulting algorithmic framework, the designer might be convinced of the soundness and practicability of the computationally generated assortment of solutions. The next steps resemble traditional design processes more closely: one may discuss the computationally generated outcomes and eventually make a choice among them. Of course, the offered solution might be functionally and aesthetically improved or it might only serve as an inspirational idea model.

For the machine to understand intelligent choice within its array of thousands of iterations, it is imperative for it to learn the human aesthetic of the designer. Frank Gehry uses the word "cranky" almost as an endearment. It would be blasphemy to call the Rem Koolhaus library in Seattle "slick". These words have precise meaning to the mature designer. Unfortunately, to the computer or to the student of design they are inexact. The upside of this process is the diversity of design choices offered; the downside is the wealth of image choices (designs) offered. For instance, the ubiquitous digital camera has transformed photography from a process of contrived aesthetic consideration to one of sorting through countless images that no longer have a cost attached to them. Because there is no cost per digital picture, we sample and overload. For the iterative process to serve us, it is important that the overload be reigned in by machines that have learnt to preselect according to the designers' predisposition.

No matter how the perspective shifts are perceived, generative software stems from a theoretical underpinning and is itself a product and an expression of man-made design. On the other hand, it cannot be argued that our design methodologies change with the advancement of tools and technology. Just as drawings have become part of architectural practice since the Renaissance (McQuaid 2002), computer-aided design has become common within the last decades and complex simulations will become common practice very soon. Instead of the obsolescence of theory in design and architecture due to the autonomy of generative machines, Frichot (2009) suggests to embrace the opportunities of "design intelligence" (Speaks 2000) offered by the new rising paradigm which is also refered to as "projective architecture" (Somol and Whiting 2002).

Future Work – A Virtual Swarm Grammar Design Walk-Through

Imagine the following scenario. An interdisciplinary team of researchers has gathered around a table. A holographic, three-dimensional still hovers several inches above the desktop. Before the researchers debate how to tackle the presented problem, each tries to capture the problem alone. After several minutes have passed, the experts describe what they see through their own eyes facing the challenge. They explain what an ideal solution to the problem might look like, point out the differences to the visualization in front of them and suggest ways to get from the status quo to the desired goal. Some members of the group realize that the spoken word defines and confines meaning within the parameters of language, others point out the need for additional semantics and offer visual clues to their unconscious that are outside of language. After listening to each others' opinions, the meeting is adjourned.

For the following session the team members prepare visual representations of the problem and ideal solutions defined in a pragmatic way. A resulting three-dimensional graph is blended into the projection that again occupies the space above the table. The most relevant units in the scope of the problem are represented as spherical nodes. The nodes are connected through spatial edges that indicate relationships among the involved units. While explaining the model, the original three-dimensional image of the problem fades out slightly in order to highlight the expert's theoretical, graphical model of the situation and its solution. Fading is good because thought becomes more abstract and is not confined to the constraints of the display. The experts begin to discuss the tangibles of the team's understanding of the design task. The team agrees to consider those corner stones that provide models that deviate from the normative expectations.

During the next meeting, sets of solutions are reviewed that were discovered in the implicitly defined space of possible solutions. The solutions that the team prefers require a re-adjustment from first principles. This process of model improvements may be repeated several times. In the meantime, small changes could be manually or automatically introduced into the model on various scales to analyze the robustness of the interaction networks of the solution. New aspects become obvious throughout the inspection and revision sessions. For instance, unbearable costs and realization times of the suggested solutions might call for improvements. Eventually, the team chooses among the presented solutions. Details on the transition from the status quo to the result are revealed by the 'projective' computational engine and studied by the team. Knowing what has to be done to master the challenge, the team maps the theoretical solution to a project plan and takes the appropriate actions.

The expert team can address anything in Anyville: an architectural design task, or the modeling of a novel drug. Of course, the outlined design process describes but one methodological 'projective' scenario that integrates interdisciplinary expertise, teamwork, graphical modeling, complex simulation and bio-inspired learning techniques. The presented computer-supported team-play is a viable and potentially fruitful scenario for tackling complex problems.

Conclusion

We introduced swarm grammars as a representation that integrates a bio-inspired CDM with a clear understanding of inherent complex and emergent phenomena. Swarm grammars allow intuitive, graphical and universally applicable modeling of multi-agent systems. In addition, we presented several evolutionary breeding techniques as examples of how to drive automated or semi-automated developments of design solutions. We emphasized, however, that despite proclamations of the "death of theory", the design focus is merely shifting towards framing the respective design challenges into formalized mathematical models. This is similar to the time when pen and paper drawings became an integral part of architectural design practice. The difference is that the mathematical models do not seem mathematical (in the strict sense of exact equations) but involve users through an intuitive interface into virtual design worlds. Evolving and unfolding concepts are accessed through simple entry points; results are promoted or demoted through interactivity on a multitude of levels. Swarm-driven, 'projective' design fosters a clearer understanding and thus better control of design-induced emergent effects on a global scale. The focus of design shifts and previously neglected challenges that exist because of their complexity become simple. Less has become more.

References

Bader, J. (2006). The Drosophila Protein Interaction Network May Be neither Power-Law nor Scale-Free. Power Laws, Scale-Free Networks and Genome Biology: 53--64.

Barabasi, A. L. and R. Albert (1999). "Emergence of scaling in random networks." Science 286(5439): 509--512.

Bonabeau, E., M. Dorigo, et al. (1999). Swarm Intelligence: From Natural to Artificial Systems. New York, Oxford University Press.

Chomsky, N. (1956). "Three models for the description of language." Information Theory, IRE Transactions on 2(3): 113--124.

Flagge, I., R. Schneider, et al. (2004). Die Revision Der Postmoderne: Post-modernism Revisited:[in Memoriam Heinrich Klotz], DAM, Deutsches Architekturmuseum.

Frichot, H. (To appear in 2009). On the Death of Architectural Theory and Other Spectres. Design Principles and Practices: An International Journal.

Hidalgo, C. A. and A.-L. Barabasi (2006). Scale-Free Networks. Scholarpedia: The free peer reviewed encyclopedia.

Holland, J. H. (1998). Emergence: From Chaos to Order. New York, Oxford University Press.

Huepe, C. and M. Aldana (2008). "New tools for characterizing swarming systems: A comparison of minimal models." Physica A: Statistical Mechanics and its Applications 387(12): 2809 - 2822.

Jacob, C. and S. von Mammen (2007). "Swarm grammars: growing dynamic structures in 3D agent spaces." Digital Creativity: Special issue on Computational Models of Creativity in the Arts 18(1): 54--64.

Junker, G. (2006). Pro OGRE 3D programming, Apress.

Klein, J. (2008). breve: a 3d Simulation Environment for Multi-Agent Simulations and Artificial Life., http://www.spiderland.org/.

Kniemeyer, O., G. H. Buck-Sorlin, et al. (2004). "A Graph Grammar Approach to Artificial Life." Artificial Life 10(4): 413--431.

Kohavi, R., D. Sommerfield, et al. (1996). Data mining using MLC++, a machine learning library in C++. Proceedings of the 8th International Conference on Tools with Artificial Intelligence (ICTAI'96).

Kumar, S. and P. J. Bentley (2003). Biologically Inspired Evolutionary Development. Evolvable Systems: From Biology to Hardware: 99--106.

Hoar, R., J. Penner, et al. (2003). Modelling Bacterial Signal Transduction Pathways through Evolving Artifical Chemistries, Canberra, Australia.

McQuaid, M. (2002). Envisioning Architecture: Drawings from the Museum of Modern Art. New York, NY, USA, The Museum of Modern Art.

Penner, J., R. Hoar, et al. (2003). Bacterial Chemotaxis in Silico. ACAL 2003, First Australian Conference on Artificial Life, Canberra, Australia.

Prusinkiewicz, P. and A. Lindenmayer (1996). The Algorithmic Beauty of Plants, Springer-Verlag.

Resnick, M. (1997). Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. Cambridge, MA, MIT Press.jams to economic systems, work the same decentralized

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. SIGGRAPH '87 Conference Proceedings.

Russell, S. J., P. Norvig, et al. (1995). Artificial intelligence: a modern approach, Prentice Hall Englewood Cliffs, NJ.

Smith, R. (2007). The Open Dynamics Engine (ODE), http://www.ode.org/.

Somol, R. and S. Whiting (2002). "Notes around the Doppler Effect and Other Moods of Modernism." Perspecta 33: 72--77.

Speaks, M. (2000). "Which Way Avant-Garde?" Assemblage(41): 78.

Vicsek, T., A. Czirók, et al. (1995). "Novel Type of Phase Transition in a System of Self-Driven Particles." Physical Review Letters 75(6): 1226--1229.

von Mammen, S. and C. Jacob (2007). Genetic Swarm Grammar Programming: Ecological Breeding Like a Gardener. 2007 IEEE Congress on Evolutionary Computation.

von Mammen, S. and C. Jacob (2008). Evolutionary Swarm Design of Architectural Idea Models. Genetic and Evolutionary Computation Conference (GECCO) 2008, New York, NY, USA, ACM Press.

von Mammen, S. and C. Jacob (2008). The Spatiality of Swarms --- Quantitative Analysis of Dynamic Interaction Networks. Proceedings of Artificial Life XI, MIT Press.