ELSEVIER

Automation in Construction xx (2005) xxx – xxx

AUTOMATION IN CONSTRUCTION

www.elsevier.com/locate/autcon

Getting complexity organised
Using self-organisation in architectural construction

Fabian Scheurer *

Chair of Computer Aided Architectural Design (CAAD), Swiss Federal Institute of Technology (ETH) Hönggerberg HIL E15.1 CH-8093 Zurich, Switzerland

Abstract

The increasing number of the so called “non-standard” designs in architecture, enabled by the development of digital tools of design, analysis, and fabrication, poses new questions for the engineering part of the building process. The number of explicit design decisions required to define the constructive details of a non-repetitive structure can be overwhelming. This paper illustrates three design examples that use methods from the field of artificial-life to reduce the design decisions and organise complex non-regular structures: the design of a “forest of columns” for the *Groningen Stadsbalkon*, the construction of an adaptive quad-mesh for a *CAAD Swissbau Pavilion*, and the optimisation of a large irregular roof structure.

Keywords: Self-organisation; Emergent design; Artificial-life; Optimisation

1. Introduction

Irregular spatial structures are a rising phenomenon in contemporary architecture. Recent examples include the “bird’s nest” design for the Chinese National Stadium by Herzog and de Meuron [1], the “foam” structure of the Chinese National Swimming Centre by PTW Architects [2,3], and the “fractal” facade of the Federation Square complex in Melbourne by Lab Architecture Studio [7]. In early 2004 an exhibition in Paris assembled a selection of many more examples under the name “Architecture Non-Standard” – Non-standard architecture – which has become a general term for this evolving direction or “style” in building [10].

Aside from the aesthetic connotations, which shall not be discussed here, there are three major technical advances fostering this phenomenon. First, there has been tremendous development on the field of design tools for architects over the last decade. Free form design with Non-Unified Rational B-Spline Surfaces (NURBS) is now a standard feature of virtually every CAD package available on the market and today’s architects are able to “sculpture” forms on their desktop PCs, which only a few years ago would have required industrial strength workstations. A lot of these technologies are spin-offs from the entertainment industries, whose R&D departments have notoriously higher budgets than the architectural offices who are now integrating them into their design process. Since the decreasing hardware costs are also a result of the huge market for high-end graphics generated by the gaming industry, it is very likely that this development will continue. Second, the analysis tools available to engineers, which are able to simulate the performance of structures are becoming more powerful, cheaper and more easy to use at almost the same pace. Finite Element simulations allow evaluating complex irregular structures in whole instead of reducing them systematically into smaller, more manageable subsystems. Thus, there is less need to think in grids and modules to get a grasp of the structural behaviour of complex design. And third, Computer Aided Manufacturing (CAM) enables the production of parameterised individual parts for (almost) the cost of mass production, thus lowering the building budgets for non-repetitive structures to a reasonable level. In recent time digital manufacturing tools are becoming a common sight, “trickling down” [9] from large companies to the workshops of small and middle enterprises (SME) which are still forming the major part of the building industry. The availability of these manufacturing technologies is increasing, and the associated costs are decreasing.

* Tel.: +41 44 633 40 25.
E-mail address: scheurer@hbt.arch.ethz.ch.

0926-5805/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2005.10.006

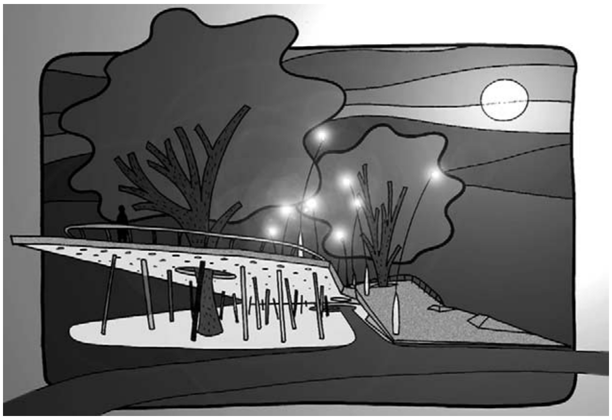

Fig. 1. The design of the Groningen Stadsbalkon (by permission of KCAP).

These advances in design, engineering, and fabrication technology obviously make it easier to build complex non-regular structures, but at the same time the building process itself is getting more complex due to the increasing amount of information, which is necessary to describe those structures. In his recent work Bill Mitchell defines the complexity of a design as the ratio of *added design content* to *added construction content* or simpler: “the number of design decisions relative to the scale of the project” [11,12]. For example he assigns a higher complexity to Gehry’s Bilbao Guggenheim than to Foster’s roof for the British Museum courtyard, because in the latter the form of every single (non-repetitive) node and member is implicitly defined by the constructive system and the shape of the roof, whereas in the former the shape is *not* controlled by simple rules but by explicit design decisions. Mitchell argues that the digital revolution in architecture makes it possible to handle increasing amounts of complexity and hopes that this may lead to a more appropriate design for the specific building site.

When it comes to actual construction of a complex building, the question arises: What is a reasonable quantity of explicit information for a specific design, and how does one communicate it in a reasonable fashion? How does one effectively design a thousand different parts, and how does one efficiently draw a thousand different plans? The second problem may be technically solved by establishing a continuous digital chain between designer, engineer, manufacturer and their respective computing environments. But the first question can become extremely difficult to answer if the design is intended to be irregular and its components are highly interdependent. An effective solution should organise vast numbers of inter-reliant components and do so in a more flexible and “intelligent” manner than those used in the digital design-systems of today.

Since 1987 research on *Artificial Life* has tried to develop a better understanding of processes which are based on local interactions and exhibit a self-organising behaviour without global control [8]. In biology, sociology, engineering, computer sciences and many other domains, many researchers are radically changing their viewing perspective to a bottom-up understanding of these processes. In the field of architecture there have been numerous approaches to use those methods, mostly for form-finding like *Genr8* or *MoSS* [6,15], the organisation of spatial layouts [4,16] or urban design [5]. Much less material is to be found on constructive issues, an exception here is the work of Kristina Shea on performance based design [13]. In the following three examples I will try to show how it is possible to apply methods from the field of artificial-life to constructive problems and so reduce the number of explicit design decisions by exploiting phenomena of self-organisation.

2. Growing a forest of columns: the Groningen Stadsbalkon

In the year 2003 the office of Kees Christiaanse Architects and Planners (KCAP) together with Ove Arup and Partners in Amsterdam and the chair of CAAD at the ETH Zurich experimented with self-organizing structures for the *Groningen Stadsbalkon* to create “a forest of columns”. Some 150 slender pillars with differing diameters, randomly dispersed and slightly inclined in different directions were supposed to hold up the large concrete slab of a pedestrian area in front of the train station in Groningen (Fig. 1). The structural implications of this design had been roughly calculated and approved by the engineers, but the challenging task was to define the exact location, diameter and inclination of every single column so that it would not obstruct the predefined walking and cycling paths. The column diameters, position and spacing, was immediately related to the loading, spanning and cantilevering capacity of the slab. Consequently every

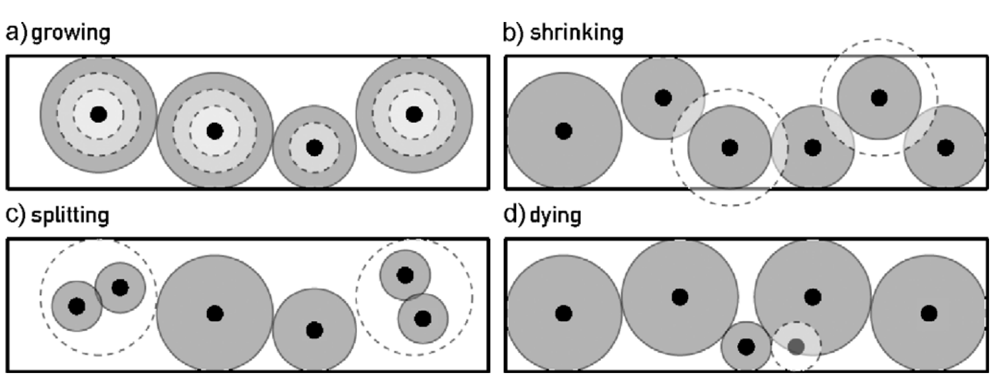

Fig. 2. (a–d) Growth and decline of the column agents within the habitat.

local change in one column would propagate through the whole structure because it affected the parameters of its neighbours. In addition, the total number of columns was not predefined but should be optimized. Since the placement of columns was only controlled by local rules, it seemed logical to self-organisation.

To simulate a process of self-organisation, the above definition was translated into a computer simulation model where the columns resemble simple *agents* that are "living" in a *habitat*. They can adapt to their local environment by moving, tilting and changing their diameter (which will increase or decrease the bearing capacity) within certain limits. If an agent reaches the maximum strength and still can't cope with the load, it may split into two agents of the smallest size, thus creating a new column in the habitat. If an agent reaches the minimum size it may eventually die, removing itself from the habitat (Fig. 2). The computer simulation was based on particle dynamics and its structure is shown in Fig. 3: the top and bottom ends of a column are particles connected by a spring, which tries to align them vertically (this spring resembles the actual column). A circular area around the top marks the bearing capacity of the column, which is dependent on the column diameter. The tops are repelling each other by force fields which try to keep the distance so that the circles of neighbouring columns just touch. The edges of the slab and the expansion joints are linear repellers, pushing the column heads away, the centrelines of the paths do the same for the column feet. The openings in the slab are defined as repeller points.

The growing, shrinking, dying and splitting of an agent are triggered when the pressure it experiences from neighbours and repellers exceeds or falls short of defined threshold values.

The simulation was programmed in Java with the Java 3D API to visualize the results in real time and three dimensions (Fig. 4). The model can be influenced interactively by changing the various parameters and it is also possible to pick single columns and drag them to the desired locations while the simulation is running. When the model reaches a stable state, the results can be exported in various formats, ranging from a list of the column coordinates to a VRML-model.

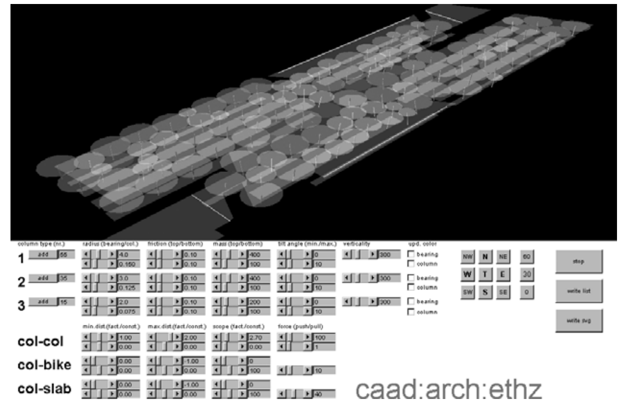

Fig. 4. Screenshot of the running simulation.

2.1. Results

The simulation model turned out to be very successful and enabled the architects to develop a number of possible solutions that were complying with the rules in very short time. The engineers only had to test a few alternatives in a finite element analysis, pick the best-performing and apply a few minor changes. However the simulation had a few flaws, one being the instability induced by the discrete growth of the columns, which added a lot of motion to the system and kept it from finding a stable state for a long time. In general though, the idea of modelling a rather simple generative system very closely after the given formal, functional and constructive constraints proved to be a good way of bringing together the needs of the engineers and architects in a crucial design stage, when the main decisions were already taken, but due to the deliberately irregular design concept the solution space was vast and difficult to probe by means of the available tools. The bottom-up concept of the columns "negotiating" their locations based on local information ensured that the constraints were fulfilled to a sufficient degree by all columns when the system reached a stable state.

3. Squaring a sphere: The Swissbau Pavilion

The second example is a small pavilion designed to show the potential of a continuously digital process from design

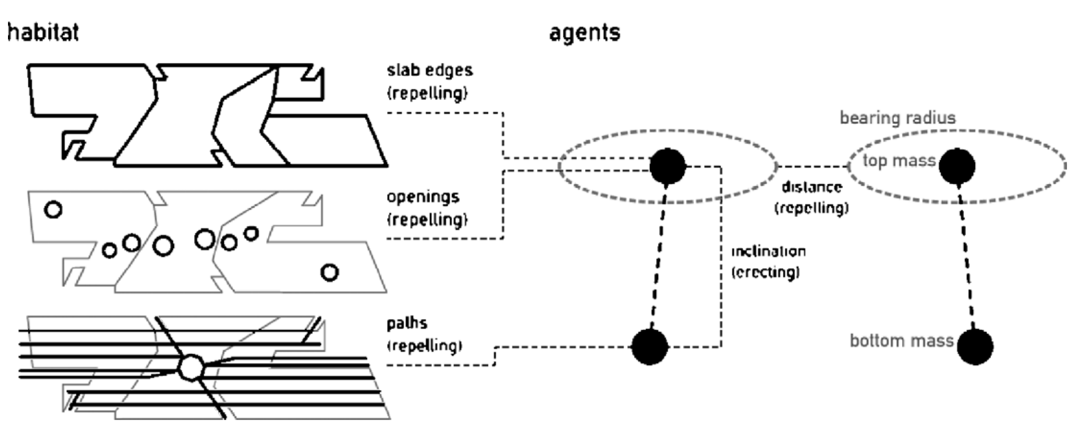

Fig. 3. Relations between habitat and agents in the Groningen model.

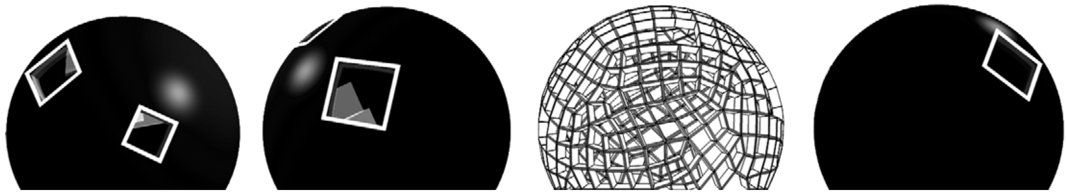

Fig. 5. Renderings of the pavilion's design and structure.

through construction until fabrication on the *Swissbau* fair in Basel. The pavilion has the form of a sphere with 2 m radius and reaches a height of 3 m. It is assembled from quadrilateral wooden frames, each consisting of four wooden boards standing perpendicular on the surface of the sphere. But while in a traditional coffered dome a regular structure dictates the placement of openings, here the structure was required to react to the deliberately asymmetric placement of windows (Fig. 5).

The generation of a quad-mesh is a well known task for everyone familiar with finite element systems; there are a number of algorithms and programs for this problem. In this specific case, however, there were some special boundary conditions: the mesh had to adjust to one large and four smaller openings so that the edges of the mesh were aligned with the edges of the openings. Also, the lower edges of the base frames had to align with the floor plane. The final structure could then be built by simply leaving away the frames below the floor level and inside the openings.

3.1. A growing mesh

To simulate an adapting mesh, again a system based on particle dynamics and local interactions was created (Fig. 6): Each *node* in the mesh is a particle that is connected to the centre of the sphere with a spring that defines the sphere's radius. *Edges* are springs between two nodes whose idle length corresponds to the ideal length of the wooden boards. *Meshes* are groups of four edges, two *diagonals* and a *diagonal connection*. *Diagonals* are springs connecting two opposite nodes of a mesh and adjusting their idle length according to the lengths of the mesh's edges, thus trying to keep the mesh rectangular. *Diagonal connections* are springs connecting the centre points of both a meshes diagonals and assuring that the mesh stays convex and as planar as possible.

The resulting mesh is flexible to a certain degree, allowing it to adapt not only to changes in the sphere's diameter but also to external forces applied to it. This is used to align the nodes and edges to the floor and the openings by introducing "magnetic" planes. The floor is such a horizontal plane cutting through the sphere 1 m below its centre, each opening defines a four-sided pyramid from its corners to the centre of the sphere. By attracting the nodes while at the same time repelling the centres of the meshes, the planes align the edges in their vicinity (Fig. 7).

While adjusting to the radius of the sphere and the openings, the topology of the mesh can be changed by three transformations, very much like a Shape Grammar [14]. This restriction assures that only quadrilateral meshes are created. The transformations are triggered by internal checks that the meshes perform in every update cycle:

- If the pulling forces that are applied to a node by its adjacent edges are exceeding a certain threshold, the node may split and create a new node and a new mesh for every adjacent mesh (Fig. 8a).
- If a mesh is deformed beyond a certain threshold the mesh may collapse by joining two opposite nodes into one (Fig. 8b).

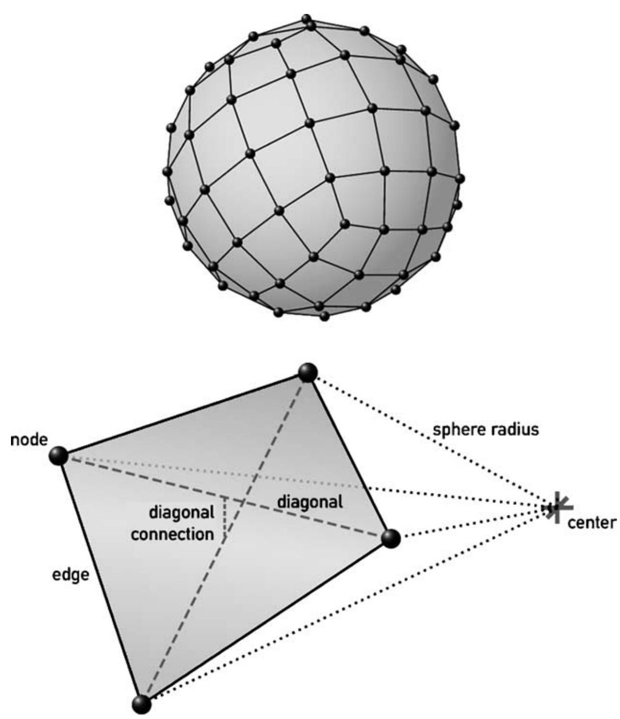

Fig. 6. The mesh and its components.

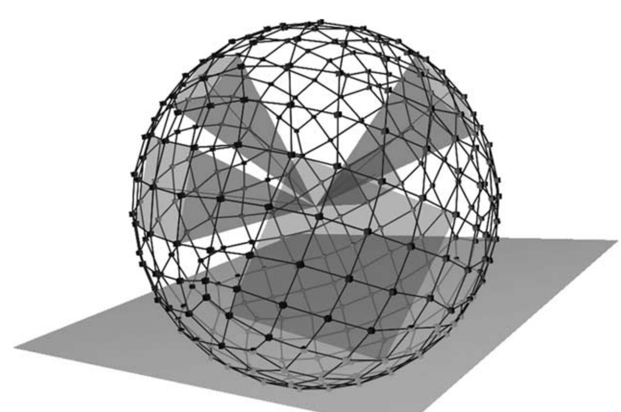

Fig. 7. Screenshot: mesh aligning with floor and openings.

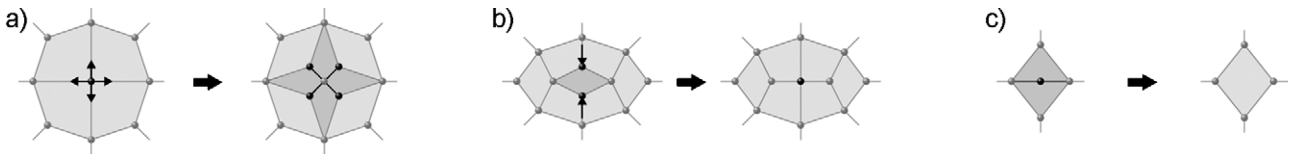

Fig. 8. (a–c) Mesh transformations: splitting a node, joining two nodes and removing a node.

– If two meshes are sharing three nodes, the node in the middle and its adjacent edges are removed and the two meshes are joined (Fig. 8c).

3.2. From simulation to the real world

The generation starts with a small rhombic dodecahedron, a regular polyhedron composed of 12 rhombic meshes (Fig. 9a). Since the edges of the starting configuration are too long, the nodes immediately begin to split and create new meshes until the strain on the surface is bearable (Fig. 9b). When the centre distance is slowly increased, more meshes are added until the sphere reaches its target size (Fig. 9c). Switching on the "magnetic planes" of the openings and the floor plane aligns the meshes with the desired lines. Like in the Groningen project, the simulation is interactive, so that the user can influence the process by picking and dragging single nodes.

The resulting mesh was exported to an XML file and further processed by an XSL transformation to the import format of second software programmed in the scripting language of a CAD package. Here the volumes of the wooden boards were generated based on the coordinates of the mesh. After checking the structure within the CAD program, another script automatically created the G-Code to control the cutting of the boards using a computerized five-axis milling machine. The assembly of the numbered frames took about one day on site (Fig. 10a and b).

3.3. Results

Although the concept for the self-organising process seems very similar to the Groningen project, this small pavilion drives the idea further in many directions. Most importantly, the digital chain from the generation of the geometry to the manufacturing of the boards was completely closed. No plans were drawn besides those for the window frames. It would have been completely impossible to meet the very tight deadlines for producing over 1200 parts without a direct CAD/CAM process. In addition the concepts for design, construction, and fabrication were developed in parallel and closely interrelated, so that they could influence each other for mutual benefit. With this project not only the locations of the parts were defined by a self-organising process but also the topology of the quadrilateral meshes was important. Finally, there were four times as many "agents" to be simulated in real-time, which clearly brought this type of simulation to the edge of its capacity. To examine larger structures with more complex constraints, a different approach was needed.

4. Evolving folds: study for a roof structure

In the two previous examples, the optimisation through self-organisation was based on entirely geometrical rules. The constraints were either already geometrical (e.g. the distance to the next defined path in the Groningen project or the length of a wooden board in the Swissbau Pavilion) or could be translated into geometrical rules (the bearing capacity of a column defines the distance to its neighbours). Geometry-based rules have the big advantage of being rather easy to compute, so it is possible to do interactive simulations with up to ten frames per second on a common PC. But in terms of structural analysis this is of course a high level of abstraction with a very low level of accuracy. For the structural constraints of the Groningen Stadsbalkon it was sufficient, but it is not always possible to find appropriate "rules of thumb". This third example will show an optimisation of a complex three-dimensional structure using the parameters of structural performance. By combining structural analysis software with a genetic algorithm it was possible to evolve possible solutions for a highly sophisticated problem. This process however is computationally intensive and is unfortunately far too slow for an interactive simulation.

4.1. Stabilising a folded plane

This project was developed in collaboration with Bollinger+Grohmann Engineering in Frankfurt. It is a study for a large roof assembled from steel members. Topologically, the

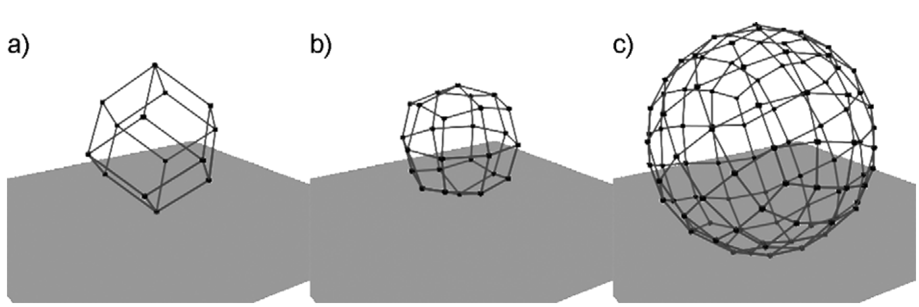

Fig. 9. (a–c) Starting configuration and growing mesh.

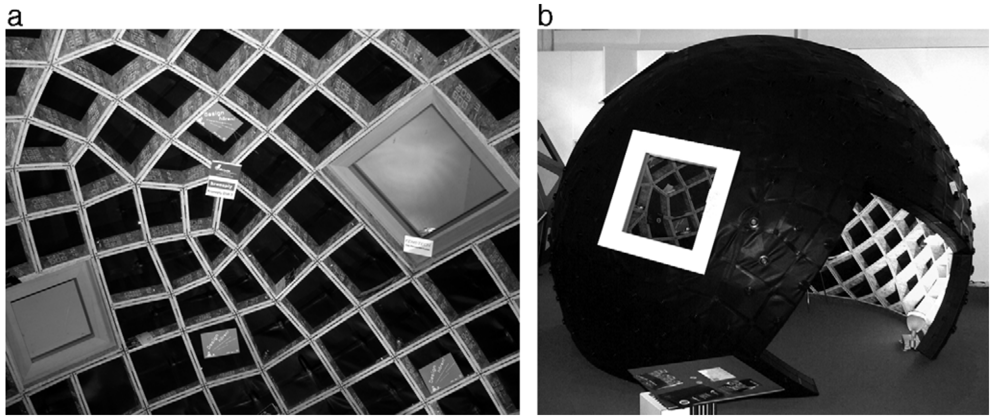

Fig. 10. (a, b) Inside and outside view of the finished pavilion.

structure is a single plane, triangulated in a regular pattern of 289 nodes and 800 members. The third dimension is achieved by folding this plane: the z-coordinates of the nodes are irregularly changed within certain boundaries. The roof stands on a kind of foot where the plane is folded down to the ground (Fig. 11). Structurally this poses a tricky problem. To achieve the necessary cantilevering capacity, it would be best to have deep folds running from the centre to the rim of the roof. But since there are no members spanning across the hole where the foot is going down, circular rings of members around the hole are the only way of keeping it from being drawn apart. Obviously those are two contradicting concepts.

Since it seemed impossible to find a set of geometric rules for stability, the only way to determine the structural performance was to simulate the behaviour of the whole structure by help of an engineering analysis software – in this case *RStab* by Dlubal – and find a way to optimise it based on the results. Fortunately *RStab* provides a programming interface, so a complete structural model of the roof could be generated from a controlling software. It could then be analysed, and the results read back to the controller automatically.

By encoding the z-coordinates of all nodes into a genome and using a genetic algorithm which allowed for crossover and mutation, the performance of the structure could be significantly improved. As a fitness criteria, the displacement of the nodes under self-weight was calculated by the analysis software, the worst node defining the inverse fitness for each individual. After 200 generations with 40 individuals each, the displacement repeatedly reached a minimum of 129 mm — at a cantilever of 25 m and with a diameter of 193 mm for the members an impressive value, according to the engineers (Fig. 12).

Where geometry based rules are not available, the exact simulation of a construction's performance might be necessary for optimisation. This of course is daily business for engineers, but methods employing artificial-life principles – like genetic algorithms – are rarely used, or trusted, in the building industry. Genetic algorithms are known to optimise for a specific given "ecological niche", evolving optimal results for the exact environment they are "living" in. This might in fact lead to exploitation of flaws in the structural analysis software, so that a genetic algorithm is building its best performing case on the grounds of a software bug in the simulation. A second corroborating analysis, performed by an alternative structural software program is strongly recommended as proof of the validity of any performance-generated design solution.

In this case, the results generated have not (yet) used to build the structure, but were used as a "proof of concept" and to develop an understanding for the mechanisms which could

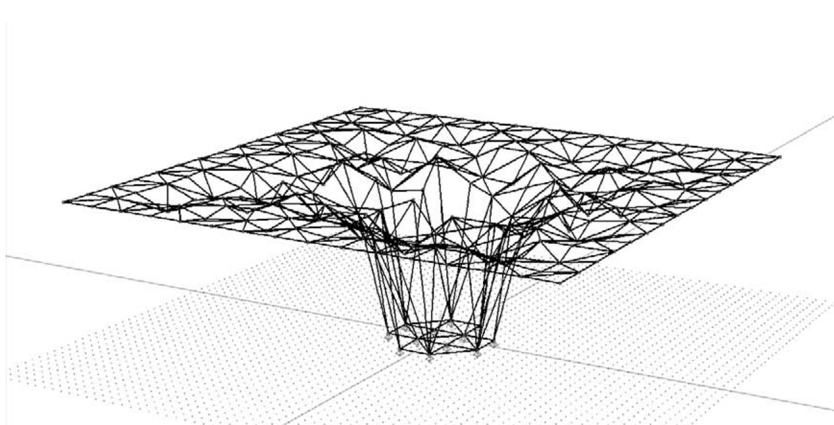

Fig. 11. Screenshot from the structural analysis software.

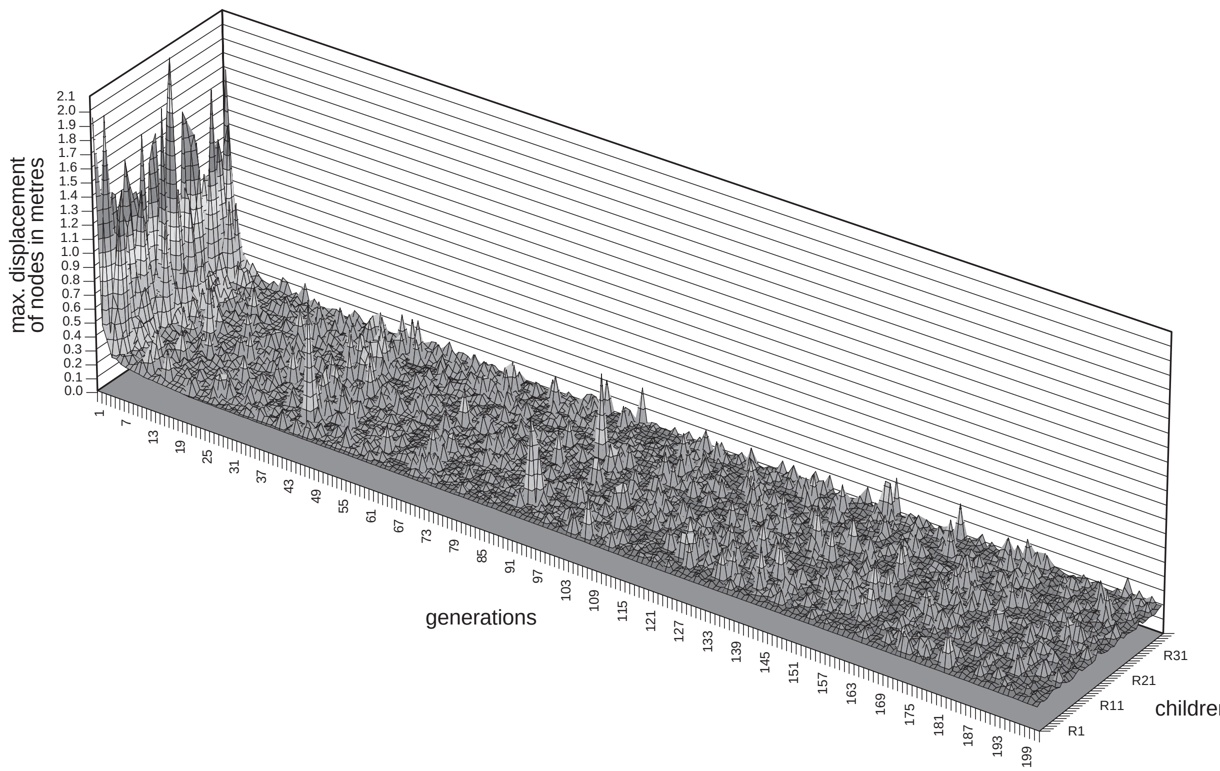

Fig. 12. Node displacement values of 200 generations with 40 individuals.

be used to create an optimal design. Interestingly, not one of the engineers on the project, with an impressive amount of experience between them, would have come up with the same engineering concepts that evolved from the A-life algorithm.

5. Conclusions

In three very different projects it has been shown, that the use of methods from the field of artificial-life can be used to solve constructive problems in architecture. All three examples are "real-world projects": Construction work on the Groningen Stadsbalkon is scheduled to be finished by mid 2006, the Swissbau Pavilion was designed (and programmed) within three months and fabricated in 60 h. The study for the roof structure is currently being used to further develop a project at Bollinger + Grohmann.

The use of artificial-life algorithms in design has a high potential for creating highly individual and complex projects; however attempts to "optimise everything" have a higher probability of generating total chaos rather than emergent patterns. Controlled, simple models such as the meshing algorithm in the Swissbau Project, create stunningly aesthetic results, which are the result of a limited interaction of parameters. The key issue for the application of artificial-life models to any design is to clearly identify the requirements and constraints for flexibility, functionality, construction, and structural behaviour.

"Non-standard" design in architecture is rapidly evolving, and with the designs come a need for engineering and construction methodologies to support them. In discussions with engineers it becomes very clear that new methods are required to solve the complex issues inherent to the realisation of these proposals. The most appropriate position for these new tools seems firmly set between the two disciplines of architecture and engineering, helping each rationalize and realize the project. In the three examples presented, the generative digital tools assisted and enabled both the design and the subsequent engineering and fabrication of the project.

The development of these digital processes not only presents the professions with a new set of tools, but also presents new challenges to the traditional working methodology. Perhaps the biggest challenge for the "non-standard" designer will be to accept that in order to optimize these processes, the designer will no longer detail the form of a design, but will design the process which generates the details.

Acknowledgements

Andy Woodcock, project manager at KCAP Rotterdam, and Arjan Habraken, engineer at ARUP Amsterdam, lead the Groningen project and invited me to collaborate. My colleagues Christoph Schindler and Markus Braach at the CAAD chair completed the team for the Swissbau Pavilion, which was realised in collaboration with Bach Heiden AG, Contec AG and the Fensterfabrik Albisrieden. Manfred Grohmann und Arne Hofmann at Bollinger + Grohmann Engineering in Frankfurt invited me to work with them on the roof structure and helped with knowledge and expertise in engineering.

References

[1] ARUP, Beijing National Stadium, Olympic Green, China, vol. 2004, Ove Arup and Partner, 2004.
[2] ARUP, The Water Cube — National Swimming Center in Beijing, China, vol. 2004, Ove Arup and Partner, 2004.
[3] S. Bull, Steve Downing, Beijing water cube — the IT challenge, The Structural Engineer 82 (2004) 23–26.
[4] P. Coates, N. Healy, C.Lamb, W.L. Voon, The use of Cellular Automata to explore bottom up architectonic rules, Eurographics UK Chapter 14th Annual Conference, Imperial College London UK, 1996.
[5] T. Elezkurtaj, G. Franck, Evolutionary Algorithms in Urban Planning, CORP, Vienna, 2001, pp. 269–272.
[6] M. Hemberg, GENR8 — a design tool for surface generation, Master of Science Engineering Physics, Department of Physical Resource Theory, MIT, 2001.
[7] Lab-Architecture, Federation Square, vol. 2004, Lab Architecture, 2001.
[8] C.G. Langton, Artificial Life an Overview, MIT Press, Cambridge, MA, 1995.
[9] R. Loveridge, K. Strehlke, Form, surface and assembly — CAAD/CAAM research, projects and implications, in: S. Sariyildiz, B. Tuncer (Eds.), Innovation in Architecture, Engineering and Construction, vol. 1, Delft University of Technology, Rotterdam, 2005, pp. 151–162.
[10] F. Migayrou, Centre National d'Art et de Culture Georges Pompidou (Paris), Architectures non standard exposition présentée au centre Pompidou, galerie sud, Paris, 10 décembre 2003–1er mars 2004, Centre Pompidou, Paris, 2003.
[11] W.J. Mitchell, Constructing Complexity, in: B. Martens, A. Brown (Eds.), Computer Aided Architectural Design Futures 2005, Springer, Vienna, 2005, pp. 41–50.
[12] W.J. Mitchell, Constructing Complexity in the Digital Age, Science 303 (2004) 1472–1473.
[13] K. Shea, R. Aish, M. Gourtovaia, Towards integrated performance-driven generative design tools, Automation in Construction 14 (2) (2005) 253–264.
[14] G. Stiny, J. Gips, Shape Grammars and the Generative Specification of Painting and Sculpture, in: C.V. Freiman (Ed.), Proceedings of IFIP Congress, 71, North-Holland, Amsterdam, 1971, pp. 1460–1465.
[15] P. Testa, U.-M. O'Reilly, M. Kangas, A. Kilian, MoSS: Morphogenetic Surface Structure — A Software Tool for Design Exploration, Proceedings of Digital Creativity Symposium., Greenwich, 2000.
[16] M.S. Watanabe, Induction Design — A Method for Evolutionary Design, Birkhäuser, Basel, 2002.

Fabian Scheurer graduated from Technical University of Munich after studying Computer Sciences and Architecture. He is currently working on his Ph.D. thesis and lecturing at the Chair of CAAD at the Swiss Federal Institute of Technology (ETH Zurich). His research is focused on the practical aspects of artificial-life methods in architectural construction and has been applied to a number of collaborative projects between architects, engineers, fabrication experts, and the Chair of CAAD.