Richard Schaffranek

OPTIMISING ARCHITECTURE – A TASK FOR THE ARCHITECT?

Algorithm, script, code, parametric design, optimisation, free form, simulation, field, crowd and swarm, these words have been around in architecture for a while. Today this field is available not only for a few brave but to any architectural student who knows how to switch on a computer. Limits have been narrowed over the years by exciting projects like Generative Components, Grasshopper, Processing, etc. which constantly introduce new features.

The newest release of Grasshopper features a 'Genetic Algorithm' object, enabling the user to 'optimise' a problem (not like De Landa imagined the genetic algorithm to produce the new [(0)]) without any deeper understanding of the processes running. Evolute[(1)], an Austrian based geometry consulting firm, just released an new 'Rhino3d Plug-in', helping architects to experiment with panelling solutions for their free form architecture and communicate with Evolute. Freeform seems to be the task of computers when looking at architectural magazines like Arch+ or AD. Optimising it, make it come real, the smoothest form vs. the cheapest panelling solution, is a field where not architects are active players but mathematicians like Evolute, not questioning the architecture, interpreting it as a 'vision' and trying to find solutions to build it. As flattering the idea of the architect's 'vision' might be, this will not push the limits in architecture. Besides Freeform optimisation algorithms are tested on office layouts and space allocation in buildings by computer scientists [(2)(3)], arguing their results with the shortest path, the most efficient work layout.

In fields like urban planning the idea of optimisation manifests itsself in another way since relations are more complex. Here the computer gets used to analyse a suggested development (as happened with the Flugfeld Aspern [(4)] with Community Viz [(5)]) based on statistical data or, as with the 'Space Syntax' method, the geometrical layout of a development and it's integration into the city. Space Syntax takes up a unique position combining a computer program with a strong idea about space and a long intensive research into this idea.

We as architects have to be aware of such developments, so that we are not only marginalised consumers of new 'optimisation' technology but are the ones that can argue what optimisation in architecture is about.

References:

(0) Deleuze and genetic algorithms in architecture, M. Delanda: in Designing for a digital world; ed. by Neil Leach; Wiley-Academy, Chichester; 2002;

(1) Evolute; http://www.evolute.at; last visited 14.10.2010;

(2) Bland, J.A. (1999) `Space-planning by ant colony optimisation', Int. J. of Computer Applications in Technology, Vol. 12, No. 6, pp. 320-328.

(3) E.K. Burke, P. Cowling, J.D. Landa Silva, and Barry McCollum; Practice and Theory of Automated Timetabling III, Lecture Notes in Computer Science, 2001, Volume 2079/2001, 254-273, DOI: 10.1007/3-540-44629-X_16

(4) Ressourcenplaner Flugfeld Aspern: interaktive Entscheidungsunterstützung, Clemens BEYER, Gregory TELEPAK, Josef LUEGER, Michael ROSENBERGER, Manfred SCHRENK; www.corp.at/archive/corp2007_LUEGER.pdf; last visited 18.11.2010;

(5) CommunityViz; http://placeways.com/communityviz/; last visited 18.10.2010;