R. Stouffs, P. Janssen, S. Roudavski, B. Tunçer (eds.), *Open Systems: Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2013)*, 705–714. © 2013, The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong, and Center for Advanced Studies in Architecture (CASA), Department of Architecture-NUS, Singapore.

SKELETAL MODELLING

A developmental template for evolutionary design

Patrick JANSSEN and Vignesh KAUSHIK
National University of Singapore, Singapore
patrick@janssen.name, vigneshkaushik@gmail.com

Abstract. Evolutionary design is an approach that evolves populations of design variants through the iterative application of a set of computational procedures. For architecture and urban design, the developmental procedure typically needs to be capable of generating bounded variability, whereby design variants are both highly variable and highly constrained. This paper proposes a template for creating such developmental procedures. The template uses decision chain encoding techniques in order to generate a sparse skeleton model, and then uses standard parametric modelling techniques in order to generate a detailed form model. A demonstration is presented where the template is used to create a developmental procedure for generating design variants for a large residential project.

Keywords. Evolutionary; developmental; generative; design optimisation.

1. Introduction

Evolutionary design (Frazer, 1995; Bentley, 1999; Caldas, 2001; Bentley and Corne, 2002; Janssen, 2004) is an approach that evolves populations of design variants through the iterative application of a set of computational procedures. The developmental procedure generates design variants, one or more evaluation procedures assess the performance of design variants, and the feedback procedure drives the evolutionary process by applying selective pressure to the population. The selective pressure is applied by ensuring that design variants with low performance scores are more likely to be killed, while design variants with high performance scores are more likely to survive, and to be selected for reproduction.

The terminology is loosely based on the way that these terms are used in biology (Mahner and Kary, 1997). A *gene* is a variable that can have different values (sometimes referred to as *alleles*, but typically 'gene' is used to refer to both the variable and the value). The expression of an individual's gene values will give rise to observable characteristics or traits, referred to as *phenes*. The *genotype* of an individual is the set of gene values, while the *phenotype* of an individual is the set of phene traits. The *genome* describes the set of all possible genotypes in the population, while the *phenome* describes the set of all possible phenotypes in the population.

From a design perspective, the phenome describes the set of possible design variants. These variants will differ from one another with regards to their phenes, but they will also share a certain common identifiable character. The formative design model can be defined that describes both the variable and common characteristics of a design, referred to as a *design schema* (Janssen, 2004). The design schema represents a specific idea or strategy for tackling a particular design scenario. For an evolutionary design approach, the design schema needs to be encoded in the development, evaluation, and feedback procedures used by the evolutionary process. Although all three procedures are affected by the design schema, the developmental procedure is the one with the most direct impact as it is the procedure that actually generates design variants.

This paper will focus mainly on the developmental procedure. A template for creating developmental procedures is proposed that helps to overcome certain key issues related to noise and epistasis. A demonstration is presented where the template is used to create a developmental procedure for generating design variants for a large residential project.

Section 2 describes the developmental template, section 3 presents the demonstration, and section 4 briefly draws conclusions and indicates avenues of further research.

2. The Developmental Procedure

The developmental procedure generates a phenotype, which in evolutionary design is a design variant. This procedure is also referred to as an *embryogeny* (Bentley and Kumar, 1999) or as a *decoder function* (Eiben and Smith, 2003, pp. 216-217). The phenotype is generated under the influence of the genotype and the environment. For the genotype, the values of the individual genes will be linked to parameters in developmental procedure. For the environment, features in the environment such as the site boundary or other types of data such as weather data will be used within the developmental procedure (Frazer, 1995).

The mapping between genes and phenes can range from being highly direct to highly indirect. An example of a highly direct mapping could be a gene that controls a dimensional parameter in a model of the phenotype. An example of a highly indirect mapping could be a gene that defines a parameter in an iterative growth rule (such as the cellular automata rule) that gets triggered repeatedly together with other rules to generate the phenotype.

The main weakness of direct types of mappings is that they result in a phenome with very limited design variability. In order to increase variability, more indirect mapping techniques need to be used. However, such indirect mapping techniques can result in highly complex non-linear search spaces that may significantly hinder the progress of the evolutionary search process. In particular, two key issues are identified that add complexity to the search space: noise and epistasis.

The issue of noise relates to the fact that as mappings become more indirect, they tend to produce a higher proportion of invalid designs (Janssen, 2004). The reason is that with a higher level of indirection, it becomes harder to appropriately constrain the developmental procedure to only produce valid designs. Invalid designs may then quickly start to dominate the population, thereby adding a high level of noise to the search space.

The issue of epistasis relates to the fact that as mappings become more indirect, they tend to have genes with a higher level of epistasis. Epistasis is the phenomenon where the effects of one gene are modified by one or several other genes. Developmental procedures with a higher level of indirection tend to use complex rule based procedures, with a significant level of interaction between the rules. The resulting high levels of epistasis obfuscate the relationship between genes and phenes, thereby making the search space highly non-linear (Draghi and Wagner, 2008).

The aim is therefore to create a developmental procedure that results in a phenome with sufficient design variability while at the same time limiting noise and epistasis to the minimum. Phenomes for architectural and urban designs are typically both highly variable and highly constrained. They are highly variable in the sense that there is no fixed organisational plan, but instead entities can be organised in space in a wide variety of ways. At the same time, these organisations are highly constrained by various rules delineating the validity of possible designs. This type of phenome that is both highly variable and at the same time highly constrained is described as having *bounded variability*.

2.1. THE DEVELOPMENTAL TEMPLATE

For phenomes displaying bounded variability, a template for creating developmental procedures is proposed that uses an intermediate skeleton model. The proposed template, shown in Figure 1, consists of two stages: skeleton generation followed by form generation.

The skeleton generation stage starts with a base model that contains fixed elements of the design not subject to evolution. A skeleton model is then constructed

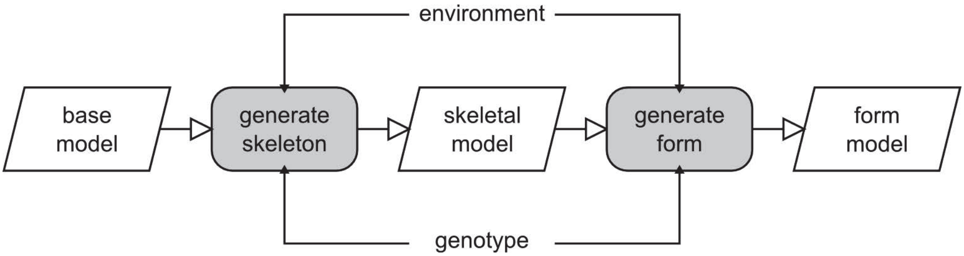

Figure 1. The template for developmental procedures.

that represents the overall form of the design. The skeleton model may consist of basic geometric elements with attributes that add semantic labels. The form generation stage starts with the skeleton model and then constructs form around the skeleton, using the geometric elements and attributes as a guide. Both skeleton generation and form generation can be influenced by the genes in the genotype or by features in the environment.

The skeleton model is a minimal three-dimensional structure that is lightweight and sparse while the form model is a more detailed model that may be large and complex. However, the skeleton generation process typically has to deal with more complex constraints, and as a result this process also tends to be more complex than the form generation process.

For the skeleton generation process, an encoding technique has been defined called *decision chain encoding* (Janssen, 2004; Janssen and Kaushik, 2013). The key feature of this encoding technique is its ability to control variability by handling complex sets of constraints, thereby minimising noise. In addition, certain techniques are suggested to limit the disruptive effects of epistasis.

2.1.1. Decision Chain Encoding

The decision chain encoding technique structures the skeleton generation process as a sequential chain of decision points. Each decision point involves choosing one option from a list of options. The list of options is created by a set of rules that generate and filter options. The genotype consists of a list of real-valued genes in the range {0,1}. For each decision point, a gene will be used to select an option by mapping it to an integer value in the range {0,n-1}, where n is equal to the total number of valid options for that decision point. Note that for each decision, the total number of valid options may not be known and may depend on the previous decisions.

With regards to noise, the decision chain encoding technique allows design variability to be carefully controlled. This allows a wide variety of valid design variants to be generated, while at the same time avoiding the creation of invalid designs.

With regards to epistasis, it is noted that with the decision chain encoding technique, a gene is not epistatically linked to any of the genes used earlier in the decision chain (since the options available at some earlier point in time are not affected by the current decision), but is likely to be epistatically linked to many of the genes used later in the decision chain (since the options available at a later point in time may be affected by the current decision). This suggests that such genotypes will have a mixed level of epistasis, with some genes having high epistasis and other having low epistasis.

3. Demonstration

In this section, the implementation of a developmental procedure for a complex building configuration is described, and a set of design variants are presented as examples of the types of forms that can be generated. The process of evolving design variants using this developmental procedure is not described in this paper.

3.1. THE DESIGN SCHEMA

The design scenario for the demonstration is for a medium-rise residential project consisting of blocks of flats. The design schema uses a "staggered brick" pattern, where rectangular building blocks consisting of multiple floors are stacked up on top of one another in a brick pattern. The design schema is loosely based on a set of recent projects that have used variants of the staggered brick patterns: the OMA's Interlace project (The Interlace/OMA, 2013), the Celosia Residence MVRDV project developed in collaboration with Blanca Lleó (Celosia Building/MVRDV, 2013), the Bjarke Ingels Group Taipei City Wall project in Taiwan (Taipei City Wall/BIG, 2013), and the A-Lab Statoil project in Oslo (Statoil/A-Lab, 2013).

For the proposed design scenario, the design schema consists of a set of blocks of four different types, each between 4 to 8 stories tall. The blocks are stacked in a staggered brick pattern, and each stack of blocks is rotated around a pair of vertical axes, creating a complex interlocking configuration. The axes of rotation coincide with the location of the vertical cores of the building, thereby allowing for a single vertical core to connect blocks at different levels. The blocks are almost totally glazed, with large windows on all four facades. In addition, blocks also have a series of shades projecting out from the façade. An example is shown in Figure 2, where 7 blocks are stacked and rotated to form an interlocking configuration.

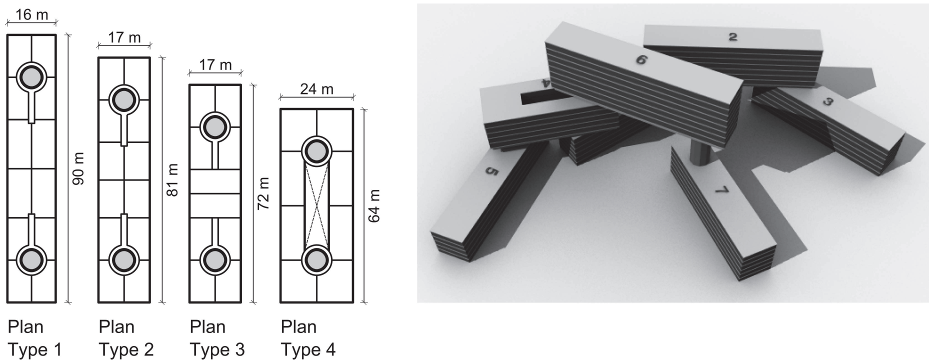

Figure 2. The four plan types and an example of how these plan types can be combined into a staggered brick configuration. In the model, the number on the block indicates the order of placement.

Three evaluation procedures are defined. The window daylight procedure calculates the number of windows receiving daylight below a certain minimum threshold on an overcast day. The façade cost procedure calculates the cost of the façade, including glazing systems and shades required to bring the heat gain through the facade to below a certain minimum threshold (Janssen and Kaushik, 2012). The core length procedure calculates the total vertical core length, representing lifts and other services. (This also acts as the penalty function for high buildings.) All three performance measures need to be minimised.

3.1.1. Hard and Soft Constraints

When generating design variants, a number of hard constraints have to be satisfied:

- The total floor area of the blocks must remain constant, based on the plot ratio of the site.
- The blocks must be positioned within the site boundary, including a setback of 10 meters.
- The blocks must be positioned so that a minimum horizontal spacing of 10 meters is maintained between blocks.
- The angle of rotation for adjacent blocks must be selected from a set of predefined angles (for example, 15, 30, 45, 60).

In addition to these hard constraints, one key soft constrain also has to be considered related to the height of the building. This soft constrain was added after initial experiment revealed that many of the design variants were unrealistically high. In the developmental procedure, the height of the building is only increased if no other options are available, thereby giving preference to low designs over high designs. In addition, the core length evaluation procedure is added that penalises very high buildings.

3.2. SKELETON GENERATION

The skeleton generation process uses a decision chain encoding technique in order to control variability. The skeleton model represents each block as a single longitudinal line at the base of the block that joins the centres of the two cores. Each line has attributes that define the block type and the block height. The skeleton model is therefore highly simplified, typically consisting of 20 or 30 lines floating in space.

The block lines are positioned within the site volume one after another. The placing of the block lines involves three decisions: choosing the block location, choosing the block type, and choosing the block height. The decision chain therefore consists of a sequence of block placing steps, where each step consists of three sequential decisions.

- The location choice decision requires a list of location options in space to be calculated. This consists of three stages: first, all locations are calculated based on the positions of the cores already placed; second, valid locations are obtained by filtering out all invalid locations that break any of the above mentioned hard constraints; third, the remaining valid locations are filtered according to soft constraints, by giving priority to lower locations over higher locations.
- The block type decision requires a list of block type options to be calculated. In total, there are four block types. However, due to the differing sizes of the blocks, they may not all fit in the chosen location. Therefore, the block type options are calculated by filtering out all invalid block types that break any of the above mentioned hard constraints.
- A block height decision requires a list of block height options to be calculated. Block heights may vary between 4 and 8 stories, assuming there is sufficient space. The block height options are calculated by filtering out all invalid block heights that break any of the above mentioned hard constraints.

Once all three decisions are taken, a new block line is then inserted into the model, and the process is repeated until the required floor area is reached.

3.3. FORM GENERATION

Once a skeleton model consisting of block lines has been generated, the form model can then be generated using standard parametric techniques. The process consists of three steps: creating blocks, creating cores, and creating facades. None of these three steps make use of any genes in the genotype. The first two steps also do not require any data related to the environment. Only the third step is influenced by the environment, as weather data is required.

- For the block creation step, a parametric model of each of the four block type is created, with the parameter being the number of floors. The block lines in the skeleton models are then replaced with the appropriate block models, with the floors parameter being extracted from the block line attribute.
- For the core creation step, the endpoints of the block lines indicate the positions of the cores. For each set of vertically aligned core points, a core is inserted, starting at the ground and ending at the highest point.
- For the façade creation step, the depths of the sunshades need to be defined and glazing systems need to be selected in order to reduce the heat gain to below a certain threshold. A a set of simulations are executed in order to calculate cumulative solar radiation on each windows for a period of a whole year and these values are then used to define sunshade depths and to select glazing systems.

3.4. GENOTYPE STRUCTURE

The genotype consists of a set of genes specifying global variables and a set of decision. All genes are real-valued numbers in the range {0,1}.

When considering the genotype structure, two issues need to be addressed. The first is how to organise the genes for the three types of decisions. One approach is to simply list all the genes in one long genotype, following the sequence of decisions. However, this means that a crossover operation would only always affect all three types of decisions. For example, it would not be possible to crossover only the block type decisions while leaving the other types of decisions the same as before. The genotype is therefore structured as four separate chromosomes: chromosome 1 is used for global variables, chromosome 2 is used for location choice decisions, chromosome 3 is used for block type decisions, and chromosome 4 is used for block height decisions. Chromosome can then be crossed over independently from one another.

The second issue to be addressed when considering genotype structure is the fact that the total number of decisions that need to be made is not known in advance. The number of blocks depends on which block types and block heights are chosen. For example, if large high blocks are chosen, then the maximum floor area will be reached with fewer blocks, and therefore fewer decisions need to be made. In order to deal with this uncertainty, the chromosome lengths are defined based on the maximum number of decisions that could be required. This means that genes at the end of the chromosome are redundant in cases where the required floor area is achieved before that maximum is reached.

3.5. DESIGN VARIANTS

The developmental procedure is able to generate a vast variety of design variants with significant variations in performance. Figure 3 shows three randomly generated design variants for a site in Singapore. The total built floor area for all three variants is 226,000 meters squared, which results in close to 1500 flats.

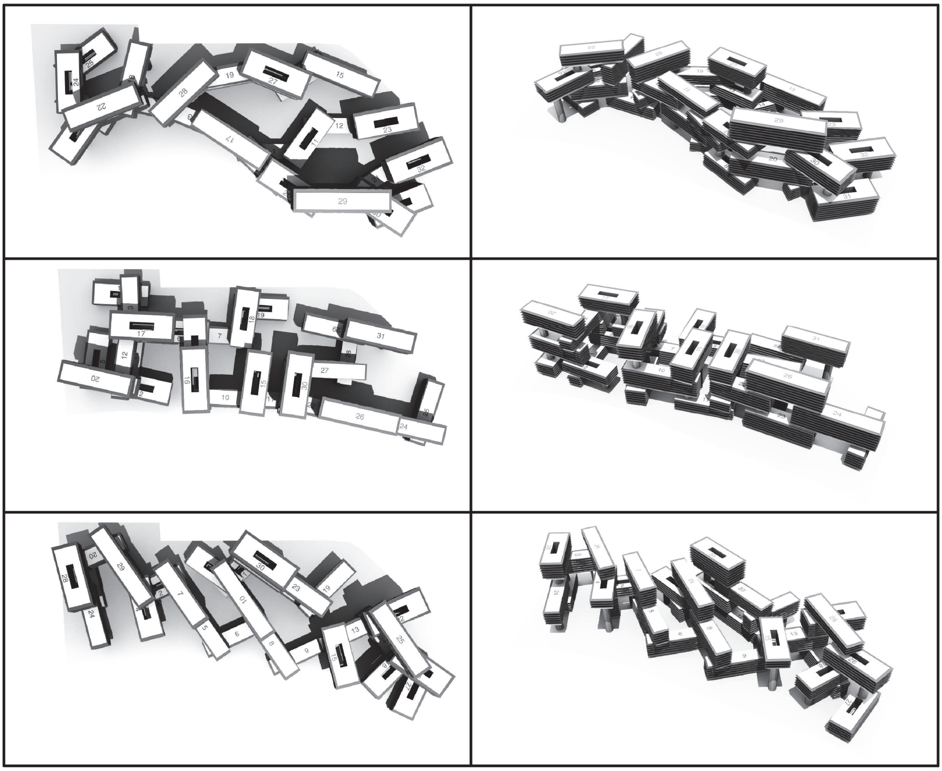

Figure 3. Three design variants.

4. Conclusions

A developmental template has been proposed for phenomes that display bounded variability. The template uses decision chain encoding techniques in order to generate sparse skeletons of design variants, and then uses standard parametric modelling techniques in order to generate models of the final forms.

Future research will further investigate the use of decision chain encoding techniques, focusing in particular on the issue of epistasis. Strategies for minimising the disruptive effects of epistasis have been proposed but have so far not been validated. Initial experiments have been conducted, where designs have been successfully evolved using decision chain techniques (Janssen and Kaushik, 2013), but further experimental data is required in order to substantiate these results.

References

Bentley, P. (ed.): 1999, *Evolutionary Design By Computers*, Morgan Kaufmann, San Francisco, CA.

Bentley, P. and Corne, D. (eds.): 2002, *Creative Evolutionary Systems*, Morgan Kaufmann, San Francisco, CA.

Bentley, P. and Kumar, S.: 1999, Three ways to grow designs: a comparison of evolved embryogenies for a design problem, *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 35–43.

Caldas, L. G.: 2001, An evolution-based generative design system: using adaptation to shape architectural form, PhD Dissertation, Massachusetts Institute Of Technology.

"Celosia Building/MVRDV": 2013, Available from: Open Source Repository < http://www.archdaily.com/29637/celosia-building-mvrdv-with-blanca-lleo/> (accessed 17 February 2013).

Draghi, J. and Wagner, G. P.: 2008, Evolution of evolvability in a developmental model, *Evolution: International Journal of Organic Evolution*, **62**(2), 301–315.

Eiben, A. E. and Smith, J. E.: 2010, *Introduction To Evolutionary Computing*, Springer.

Frazer, J.: 1995, *An Evolutionary Architecture*, AA Publications, London.

Janssen, P. H. T.: 2004, A design method and computational architecture for generating and evolving building designs, PhD Dissertation, The Hong Kong Polytechnic University.

Janssen, P., Chen, K. W. and Basol, C.: 2011, Evolutionary developmental design for non-programmers, *Proceedings of the 29th eCAADe Conference*, Ljubljana, Slovenia, 245–252.

Janssen, P. and Kaushik, V.: 2012, Iterative refinement through simulation: exploring trade-offs between speed and accuracy, *Proceedings of the 30th eCAADe Conference*, Prague, Czech Republic, 555–563.

Janssen, P. H. T. and Kaushik, V.: 2013, Decision chain encoding: evolutionary design optimization with complex constraints, *Proceedings of the 2nd EvoMUSART Conference*, Vienna, Austria, 157–167.

Mahner, M. and Kary, M.: 1997, What exactly are genomes, genotypes and phenotypes? and what about phenomes?, *Journal of Theoretical Biology*, **186**(1), 55–63.

Michalewicz, Z.: 1996, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin.

"Statoil/A-Lab": 2013. Available from: Open Source Repository < http://www.archdaily.com/166664/in-progress-statoil-a-lab/> (accessed 17 February 2013).

"Taipei City Wall/BIG": 2013, Available from: Open Source Repository < http://www.big.dk/#projects-wall> (accessed 17 February 2013).

"The Interlace/OMA": 2013, Available from: Open Source Repository < http://www.archdaily.com/34142/the-interlace-oma/> (accessed 17 February 2013).