Building the Digital World – Architectural Design Methods Based on the Use of Digital Tools – Performance Based Form Generation and Optimization

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Yasha Jacob Grobman

Submitted to the Senate of the Technion – Israel Institute of Technology

Elul, 5768. Haifa, September 2008

Dedicated to Avigail and Emma

Acknowledgements

The research Thesis was done under the supervision of Dr. Abraham Yezioro and Dr. Guedi Capeluto in the Faculty of Architecture and Town Planning.

The generous financial help of the Technion is gratefully acknowledged.

Table of contents:

List of figures

List of diagrams

List of tables

Abstract

*There has already been one loss of innocence in the recent history of design; the discovery of machine tools to replace hand craftsmen … now we are at a second watershed. This time the loss of innocence is intellectual rather than mechanical. (*Christopher Alexander, 1964)

The introduction of computers to architectural design has been making a significant impact on the way buildings are being designed and built. For the second time in the history of modern design, technology is advancing faster than the building industry. This time, the digital and information technology (IT) revolutions introduced technologies that allowed for the development of architectural designs and manufacturing tools on grounds other than need, and thus their influence on architecture is still largely unknown. This research examines changes in the architectural design process caused by the introduction of computers focusing mainly on computer-based form generation, simulation and evaluation. It suggests a new generative performance oriented design (GenPOD) model that use “performance envelopes” in a non linear generative design method.

As a preliminary stage a digital architecture database was developed. The database, which included projects that followed a new definition for digital architecture or computer-oriented design, has shown trends in computer oriented design project types, costs, geographical distribution and other details starting from the mid 1990’s.

Following the preliminary stage the research focuses on developments in architectural design tools/software, design methods and computer-based generation, simulation and evaluation tools and approaches. It discusses the increase in level of control that architects have over the designed architectural form, which is based on the increase of the amount of data the architectural form embeds, and its implications on the architectural design process. It also discusses, via a division into form- and performance based design, the shift towards performance-based architectural design, which introduced new possibilities in terms of using computers in the design process. The new possibilities derive, among others, from the ability to embed/add empirical quantitative data to the geometrical information regarding the architectural form and negotiate quantitative and qualitative data in the generation and evaluation processes of the initial architectural design.

Based on these analyses the research introduced the notion of multiple performance envelopes as a base for the GenPOD model. Performance envelopes are surfaces that connect points with similar information regarding performance (E.g. wind performance envelope will be defined by all the points in the design space with a similar wind speed). As opposed to traditional approaches in which computer simulation is used in an “after the fact“ manner in order to evaluate architectural form fulfillment of certain performative demands, the new model suggests to integrate multiple performance envelopes in a generative, “before the fact” approach.

The suggested model allows generating architectural initial form using a negotiation process of one or more performance envelopes that stand for similar or different performances, regarding different aspects of the design. The generation process is parametric and iterative in a sense that it allows to combine numerous rounds of generation using different performance envelopes that influence the entire form or only parts of the form according to programmatic demands and/or designer's preferences. The model generates in every run several design alternatives. The generated alternatives are evaluated in a new type of evaluation model that utilizes numerous fitness criteria, not necessarily used for the generation of the envelopes. These fitness criteria represent another layer of information (besides performance) that the designer receives of the generated form.

In order to select the most fitted solution each criterion is given a normalized priority by the designer. A total grade is then calculated for each design alternative. Both generation and evaluation process are parametric implying that it is possible to change the number and types of performance envelopes at any stage of the process and examine the effect of these changes immediately on the design alternative. It also implies that the designer can alter the preference regarding the priorities of the fitness criteria and thus examine changes in the total fitness grade.

The GenPOD model was tested in a case study that examines the applicability of the suggested method in a design of initial form for an office building. The research shows that designing with performance envelopes increases the general performance of the building form by increasing the amount of performance oriented information from which the buildings' form is generated while allowing to generate an architectural form that embeds a combination of user preference with empiric performance information. Moreover, since the initial form is generated using performance envelopes it adheres by definition to the performances that were used to generate it and does not necessitate an "after the fact" evaluation process, hence, guaranteeing the desired performance of the proposed building. Performance-oriented design approaches and the use of models such as GenPOD in architectural design are an important step toward a more efficient and sustainable environment.

Acronyms

BIM	Building information model
CA	Cellular Automata
CAD	Computer Aided Design/Drafting
CAM	Computer Aided Manufacturing
CBD	Computer-based Design
COD	Computer Oriented Design
CFD	Computational Fluid Dynamics
CNC	Computer Numerical Control
GenPOD	Generative Performance Oriented Design
GIC	Geometrical Insolating Coefficient
HVAC	Heating, Ventilation, Air Conditioning
RIBA	Royal Institute of British Architects
RP	Rapid Prototyping
SRE	Solar Rights Envelope
SCE	Solar Catch Envelope
SV	Solar Volume

1. Introduction

In architecture, the medium may not be the message, but it certainly has a profound effect on the message's form. (Anthony Vidler, 2001)

We are living in the midst of a cultural revolution brought about by the emerging telecommunications and information technologies. We are redefining the way we communicate, consume and spend our free time. Within this era of changes, architecture is redefining its boundaries, its language of representation and its approach toward form generation (note the similarity to Le Corbusier's description of the changes in the first "revolution" in the beginning of the 20th century (Le Corbusier, 1923).

For the second time in history, the practice of architecture is being overtaken by technology. The dramatic increase in computer processing power (see Figure 1) provides the foundation for the development of innovative computer oriented design (COD) [1] software. This software allows the contemporary designer a high degree of form manipulation and dynamic means of representation (Agrest, 2000, Benjamin, 2000). The computer has gone far beyond its initial role as a tool for drafting and representation, becoming a powerful engine for design and building (manufacturing).

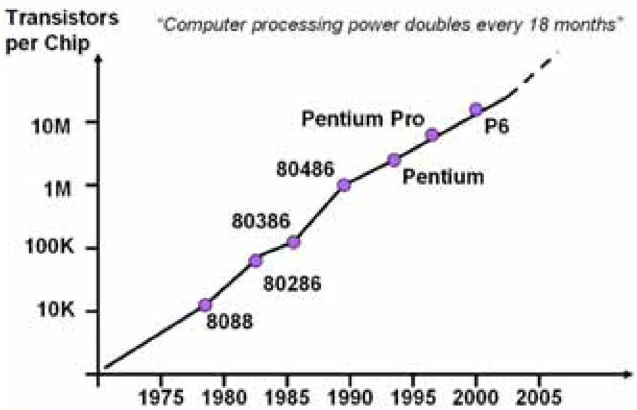


Figure 1 – Moore's law: Computer processing power doubles every 18 months (Gallupe, 2003)

Like many new technologies, new computer-oriented design tools were used initially in the same way as a drafting board or 2-D computer design (Andia, 1997). It took some time until architects and academics began to try to explore the possibilities of the new tools to create new computer-oriented approaches to design. These approaches use the computer's formal capabilities both to challenge 90-degree architecture introducing non-Euclidian, curvilinear spaces as a more complex new formal language, and to examine ways to use computer processing power to redefine the architectural design process.

---

[1] Computer-Oriented Design (COD) refers to an evolution of Computer-Aided Design (CAD) – the term COD is suggested by the author.

The new formal complexity rendered the plan and the section no longer sufficient as key representation tools for digital architecture (Zellner, 1999), since they are incapable of describing the geometrical complexity of the new objects (one example is the blob-like design seen in Figure 2). State-of-the-art 3-D applications, on the other hand, represent and probe the object being designed dynamically and from countless angles. Thus, a gradual transition is taking place from conventional line-oriented drawing representation to a more complex new convention using 3-D dynamic representation. Furthermore, many computer-designed buildings are neither represented nor built using traditional conventions.
Using the emerging 3-D software, buildings can now be fully formed in three-dimensional modeling, profiling and prototyping manufacturing software. Frank Gehry's Guggenheim Museum in Bilbao was modeled and manufactured using CATIA, an aeronautic and automotive design and manufacturing software program. Gehry was able to design every facet of the titanium and stone cladding before delivering the details to the contractor in CATIA format (Imperiale, 2000).

Figure 2 – Kas Oosterhuis, design with three-dimensional software (Oosterhuis, 2002)

Moreover, a new generation of CNC (computer numerical control) milling, laser cutting and three-dimensional rapid prototyping (RP) machines is being developed. This will enable the architectural designer to close the gap between the design and its realization – much as it has already done for the designers of automobiles, ships and airplanes. Thus, a craftsman or a builder will no longer be needed to negotiate the design process through paper plans and sections, and the building site will become an assembly site where computer-manufactured building parts will be assembled to create the new building. The new machines will usher in a new era in architecture and design, which will function entirely without an intermediary and in which the physical form is created directly from its representation on the new drafting board, namely, the three-dimensional software.

Since “architects draw what they can build and build what they can draw” (Mitchell, 2001), and the new technologies in digital design and fabrication substantially diminish the formal limitations in drawing and building buildings, it is necessary to re-examine the architectural design process.

While digital design and manufacturing tools can be employed to follow traditional design methods, they also pose an important question: Should new abilities to create and manipulate form change the way architects design? This research concentrates on the changes in the design process following the introduction of the computer. It focuses on computer-based form generation and optimization in the architectural design process. It examines the domain of the computer form generation approaches, which is marked by two conceptual boundaries that can be represented by the following two citations: “Objects are no longer designed but calculated” (Cache, 1995) and "A digital computer is, essentially, the same as a huge army of clerks, equipped with rule books, pencil and paper, all stupid and entirely without initiative, but able to follow exactly millions of precisely defined operations….in asking how the computer might be applied to architectural design, we must, therefore, ask ourselves what problems we know of in design that could be solved by such an army of clerks….at the moment, there are very few such problems" (Alexander, 1967). The following research will examine these boundaries and suggest a new theoretical model, software tool and design method that uses computers in a new generative performance based design.

2. Research Hypotheses

This research rests on an initial hypothesis: Design with computers or computer-oriented design (COD) represents a fundamental change in architectural design in general, and in architectural design process and methods in particular.

The second hypothesis suggests that computer generation, simulation and evaluation methods and tools can be embedded in the architectural design process, thus helping designers reach a higher level of control over the design.

The third hypothesis suggests that it is possible to create a design method and tool in which form is generated from one or more performance envelopes.

The fourth hypothesis suggests that it is possible to create a parametric computer evaluation process that negotiates empirical quantitative fitness criteria with designer's preferences (qualitative criteria).

3. Research aims

The first aim of the research is to characterize and analyze the changes that have occurred in the architectural design process due to the introduction of computers and to suggest expected directions for further development in the use of computers in architectural design.

The second aim is to critically examine computer-based form optimization and generation methods that have been developed to date for architectural design and to define gaps and directions worth pursuing in the future development of generative design methods. Particularly, the research will check whether multiple performance envelopes can be used to generate an architectural initial form/design envelope, negotiating empirical qualitative fitness criteria with designers' preferences (qualitative criteria).

By critically examining the current use of performative quantitative parameters as a base for computer-based form generation methods, this research will try to suggest a new type of generative design method that combines performative aspects with user inputs.

4. What is digital architecture?

4.1 Definition of digital architecture

> ***Digit*** *– 1) Any one of the ten Arabic numerals 0 to 9. 2) A finger or a toe*
> ***Digital*** *– Of, using digits.*
> ***Digital Computer*** *– One using combinations of characters in a special form (language) that are represented by the digits 0 and 1.*
> *(From the Oxford Students' Dictionary*)

Bill Gates has predicted that the present decade will be known as "The Digital Decade." He believes that by the end of the decade, all aspects of our life will be influenced by the digital realm (Leach, 2002). The term "digital" is used increasingly in our everyday lives, usually in connection to innovation and technology. Digital TVs, digital cameras and digital washing machines are only a few examples of the many devices that existed before the digital revolution and were given the "digital" prefix only recently. Not many people are aware of the difference between a digital and analog washing machine, but it is clear to all that digital stands for something new as opposed to the old concept of "analog."
According to Manovich (2001), the term digital representation encompasses three unrelated concepts: analog to digital conversion (digitization); a common representational code; and numerical representation.
Nonetheless, as in other fields, the term digital in architecture represents something new and innovative that is usually connected to computers (the digital machine). Lacking one accepted meaning, digital architecture is one of those terms that we all seem to understand without a clear definition. However, to create a common ground for this research, a clear definition is needed. Therefore, a definition that will be used throughout the research is developed in the following paragraphs.

Digital architecture can be defined from two aspects: the semantic/semiotic aspect and the practical/technological one.

Semantic/semiotic – The term "digital architecture," and especially the word "digital," implies innovation and being up-to-date, a step beyond "analog," which is associated with modernism and the beginning of the 20th century. Therefore, digital architecture, from a semantic point of view, simply means new and innovative architecture or that which is not conservative. This definition is too wide and vague and therefore will not be used in this research.

Practical/technological – Digital architecture can be defined in practical/technological terms from two different aspects:

a. The first aspect focuses on the architectural design and representation. According to this approach, digital architecture uses the computer as a design tool and not just as a tool for drafting and representation[2]. De Luca and Nardini suggest that “the possibility of creating, copying, shifting and deforming complex graphic entities is the truly innovative aspect of computer aided design” (De Luca, Nardini, 2002). The problem with this approach is in the definition of the term ‘design.’ The majority of architectural practices use computers in the design process for drafting in the same manner they used the traditional drawing board (Mirza & Nancy, 1999). Computer drafting usually starts from a sketch or a conceptual handmade drawing, which at a certain stage is “computerized” (drawn in CAD program) and developed further on the computer until the final design solution and the documentation drawings are done. Thus, in the process of architectural design, some design decisions must be made while drafting on the computer. Therefore, according to this definition, one can claim that the majority of architectural projects today could be defined as “digital architecture.” This definition’s scope is also too wide for this research; therefore, it will not be used.

b. The second aspect refers to the building form or the geometric complexity of the computer-designed projects. According to this idea, digital architecture is based on free-form, curvilinear and other complex geometries that could not be developed, represented and/or constructed without using computers. This definition emphasizes the formal aspects of digital architecture and the fact that only computers are able to represent and manipulate these forms within an acceptable time frame for architectural design[3]. This thesis will use this definition for digital architecture.

4.2 Digital architecture database

The digital architecture database is a data source for information on projects that conform to the definition of digital architecture starting from the mid-1990s. The database's initial aim was to understand the impact of the digital revolution on practical architectural design in terms of location, building type, budget and other data relating to the construction process of digital projects. It was also expected that the database would operate as a tool to investigate and compare design methods used in digital architectural projects (see appendix A for more information; see Figure 3 for the database interface).

---

[2] This aspect differentiates between the two meanings of the acronym CAD: computer-aided drafting and computer-aided design.

[3] It can be claimed that everything performed by computers in terms of design and manufacturing in architecture could be done without the use of computers given the proper resources (time, money, etc.), as was done in several highly complex buildings in the past (for example, Gaudi’s buildings). This claim is theoretically valid, although it is no longer logical to assume that these resources could be arranged today, given the fact that computers do the same work so much more quickly and cheaply.

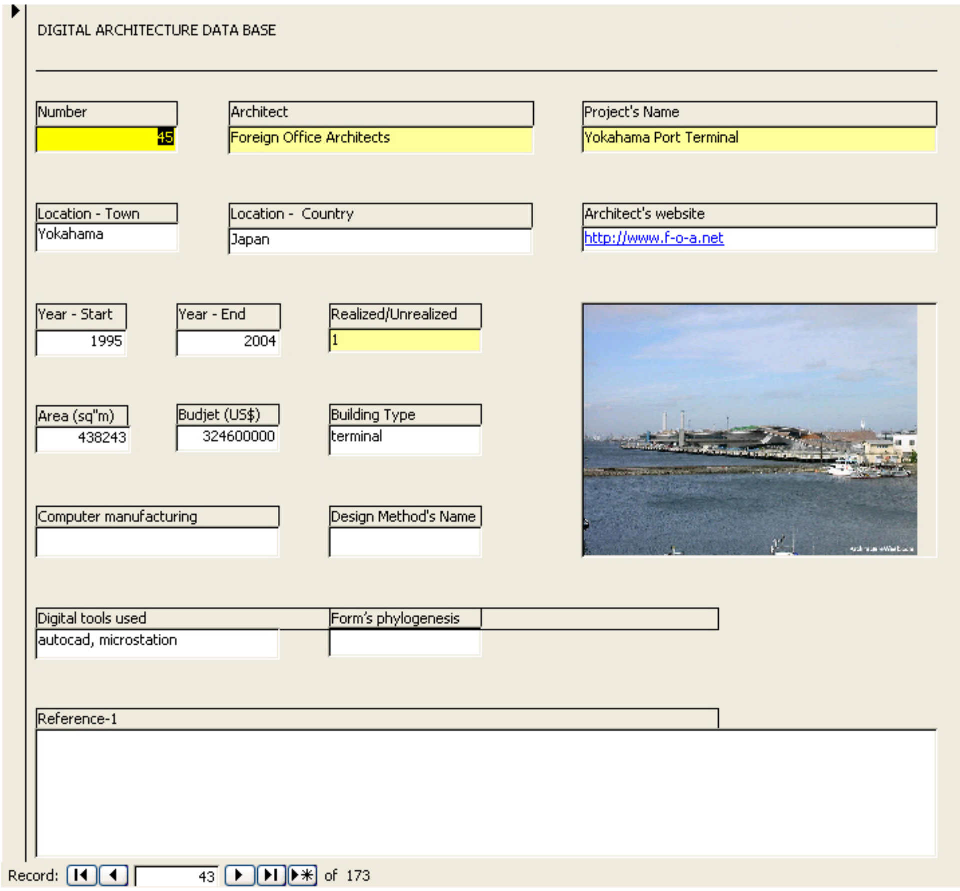


Figure 3 – Database interface

4.2.1 Project selection

The database includes only built projects and those that were designed to be built. Competition entries and other un-built projects that were designed with the intention to be built (that is, not conceptual) are also included. Virtual and conceptual projects were not included.

Projects were considered digital when the use of a computer was essential to the design or manufacturing process (see the definition of digital architecture in the beginning of this chapter). Thus, most of the projects were designed using complex geometry that requires a computer for design, representation and manufacturing. Nonetheless, projects that are based on straight lines and orthogonal geometry were also included when the complexity of the forms required computer-based design or manufacturing.

4.2.2 Database - main conclusions

The database made it possible to reach preliminary conclusions regarding digital architecture in practice. In general, it was found that the number of buildings that conform to this research's definition of digital architecture has been constantly increasing. This can be explained by the advances in the development of digital tools for design and manufacturing, and architects' increasing familiarity with these tools.

In terms of the distribution of digital projects, it was found – as expected – that most such buildings are concentrated in the Western countries that lead the digital discourse in contemporary architecture: the U.S., the U.K., the Netherlands, France and Germany. Next come countries that are less dominant in terms of their role in the digital discourse, such as Japan, Australia, Austria and Italy, and then follow countries in which one or two projects were built (see appendix B).

As for the types of digital buildings, the database shows that they are mainly high-profile public buildings, museums and cultural centers. The number of buildings of this type is almost double the number of buildings in the following category. Interestingly, the next category of building types relates mainly to the private domain (interiors, private homes, office buildings and pavilions). This can be explained by the high profile and innovative connotation associated with digital buildings (see appendix B).

In terms of cost, the database shows that the average cost per square meter of some digital building types (mainly public buildings) is close to double the price per square meter of conventional building ($2,000 per square meter). Nevertheless, the database shows that in general, the price per square meter decreased substantially from the beginning of the 1990s to the beginning of the 2000s (from around $9,000 per square meter to around $2,500 per square meter) (see appendix B). This can also be attributed to the increasing integration of digital design and manufacturing technologies in the building industry.

In relation to the architectural form's phylogenesis and the definition of performance-oriented design methods used in the selected projects, three main directions were noted: The first includes a design process in which the development of the architectural form did not involve the computer as a generative tool. The majority of the database projects belong to this category. The second direction includes projects in which the computer was used as a generative creative tool. The third direction is much the same, except for one vital difference that has to do with the type of information used to generate the form. In the third direction, only empirical information related to performance is used to generate the architectural form, as opposed to the second direction in which any type of data can be used to generate the form[4]. Of the 171 projects included in the database, 158 were defined as belonging to the first direction, 7 to the second direction and 6 to the third.

[4] See section 6.5 for a wider discussion of the difference between these directions.

4.3 Closing remarks

This chapter developed a preliminary definition for digital architecture that will be used in this research. It also presented the database of digital projects that was constructed in the early stages of this research in order to understand and define directions in the design and construction of digital projects. The initially defined directions in digital design and construction form the basis of the discussion on the influence of the computer on the architectural design process in chapter6. The following chapter discusses the technological background and implications of the evolution of digital design tools and technologies.

5. Technology/tools

5.1 Evolution and use of 3-D CAD modeling tools

The beginning of CAD software can be traced back to the early 1960s. The first software was capable of producing simple two-dimensional drawings. By the end of the 1960s, new entities like splines, patches and polygonal meshes were developed (Monedero, 1997). Solid modeling was also born in the beginning of the '60s, but the first important commercial packages (Romulus, Autocad) appeared in the beginning of the 1980s (Walter, 2003). Early 3-D modeling relied on equations that define points on a surface, which generate what is called polygon meshes (Walter, 2003). Polygon meshes are useful for modeling and rendering orthogonal shapes. When used to model curvilinear forms, polygon meshes produce poor tessellated results.

The term spline refers to a flat strip used to help in drawing complex curves on paper. In CAD software, the term refers to a smooth curve that runs through a series of user control points that are called nodes, vertexes, anchors and control points, etc. The curve can be modified by moving the control points. A spline surface is defined by a net of connected spline curves. The user can modify the surface by changing the position of one or more control points, which causes the computer to recalculate the surface. The frequently used B-spline is named after Pierre Bezier, a French automobile designer who developed the mathematical formula to calculate this spline in the 1960s (Walter, 2003). NURBS (Non-Uniform Rational Bezier Spline) refers to a further development of the B-spline that enables smoother and more complex curves (see Figure 4).

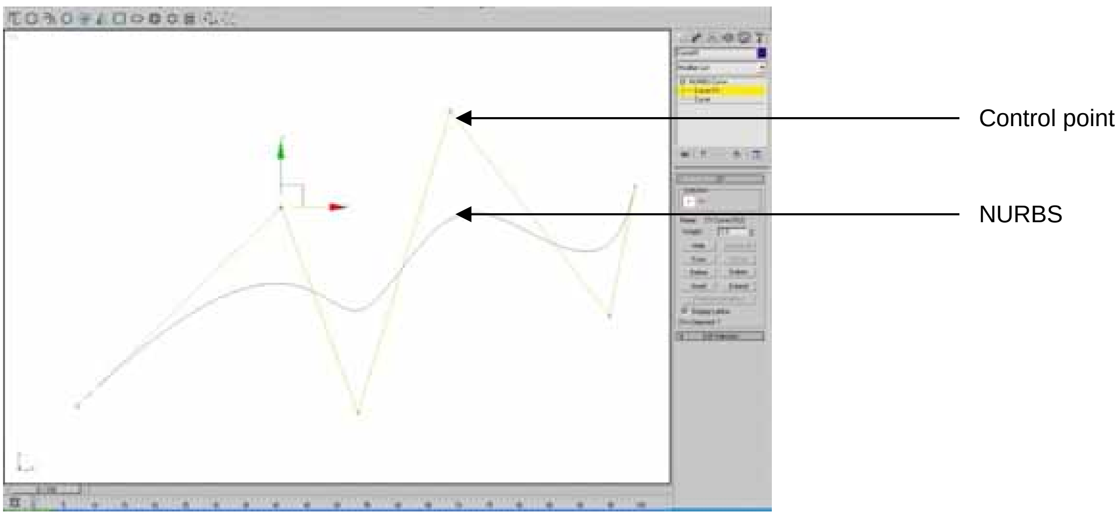


Figure 4 – NURBS-based line defined by control points in 3ds Max

In architecture, 3-D models are elaborated mainly by the following techniques: polygonal meshes, solid model or parametric models such as NURBS[5]. Most architectural models still use the first technique combined with solid modeling due to the planar and orthogonal nature of architectural design. Many architects still work in what is called 2.5-D, which is based on extrusion of lines or polylines to a particular height (Monedero, 1997).
The use of free-form parametric models that rely on NURBS-based surfaces is constantly on the rise. The increase in computer processing power and the introduction of computer-aided manufacturing is steadily decreasing the costs of manufacturing free-form architecture. Although the first digitally designed building to be realized[6] – the water pavilion built by NOX[7] – was completed only in 1997 (Rosa, 2001), more than 100 digitally designed and partly computer-manufactured buildings have been constructed since then. Moreover, leading schools and architectural departments have established design studios whose research agenda concentrate on the use of digital tools[8] .
According to research about the use of CAD in the United Kingdom (Mirza & Nancy, 1999), only 50 percent of architectural practices model in 3-D. This rate drops even more in small practices, of which only 40 percent use any CAD at all. It is reasonable to believe that since 1999, the number of CAD users and the use of 3-D modeling has increased, but it is clear that the 3-D CAD model has not yet become the primary means of communication between architects and builders and that 2-D project documentation is still the norm for the majority of building projects.

5.2 3-D tools in architecture

The 3-D tools used by architects can be defined using five categories:
a. Drafting and modeling software – This category features two types of software covering most of the tools used by architects. The first type, drafting software, has limited 3-D modeling capabilities and is used mainly to develop 2-D designs and documentation, which

---

[5] In this context, parametric model refers to a model built from curves that can be changed in a parametric way, such as moving the control points. This meaning differs from parametric design that refers to the introduction of constraints between the different objects in the model.

[6] Both Gehry's "Fish" and Guggenheim Museum Bilbao, which were built in 1989-92 and 1991-97, respectively, and Future Systems' Natwest Media Centre, built in 1994, involved computers in the design and manufacturing process. Nevertheless, their forms were not developed via a digital design method but rather via the traditional method that concentrated on plan, section and physical models.

[7] Future Systems' Natwest Media Centre (1994), Gehry's "Fish" in Barcelona (1991-1994) and Guggenheim Museum Bilbao (1991-1997) are not considered digital in this context although computers were used in their manufacturing process, because all of them used traditional design method in the design process.

[8] Columbia University's paperless studio, AA DRL, SIAL in RMIT and several other schools have design studios whose research agenda concentrates on the use of digital tools in architecture.

still serve as the main communication method with builders or the producers of buildings. The 3-D modules included in this software usually offer mainly orthogonal modeling options based on solid modeling and polygon meshes, presenting only limited options for free-form design. The leading software programs are Autocad (www.autodesk.com), Microstation (www.bentley.com), Datacad (www.datacad.com), Arc+ (www.aca-europe.com) and Vectorworks (www.nemetschek.net). In addition, there are many other applications that have limited market presence.

The second type is modeling software designed for architectural use (design, modeling, representation and the production of models and building elements). In some cases, it is a limited version of animation software. Among the commercial software in this category are 3D VIZ, a limited version of 3ds Max (www.discreet.com), SketchUp (http://www.sketchup.com) and Rhino (http://www.rhino3d.com/).

b. Parametric software programs for architects – The first parametric applications were introduced in the 1980s. By that time, the efforts to develop an application that solves “space allocation problems” by computer-based generation of a project’s plan had been reduced (Shaviv, 1987). The initial parametric applications concentrated on developing human-machine interactive programs in which a design is developed interactively by a designer using a computer program that acts as a design companion (by introducing geometric constraints) instead of a drawing board (Shaviv, 1987). Most of the applications developed using this approach were 2-D and were limited by the number of entities they could handle. Therefore, it was understood that a combination of methods was needed. Shaviv (1987) suggested using a combination of the generative method and an evaluation method (an application that evaluates the design using predetermined parameters). The efforts to develop an architectural parametric application did not produce any significant commercial application until the end of the 20th century.

In the 1990s, a second generation of parametric software for architects was developed. This software offered a different approach, suggesting a solution for the needs of both 3-D modeling and 2-D documentation. In this software, the plans and the sections derive from the same 3-D model. This design method is also called “one-model design” or “one-model building”. The software includes tools for documentation as dimensioning and text (see Figure 5). Free-form design techniques have not yet been fully integrated in this kind of software, which acts as a limitation in terms of the design’s formal expression.

Figure 5 – Revit interface – One 3-D model is used to create plan and section documentation

The leading software programs of this type are Architectural Desktop (ADT) (www.autodesk.com), Revit (www.autodesk.com), Archicad (www.graphisoft.com) and Microstation Triforma (www.bentley.com).

Bentley is currently developing parametric software for architects called Generative Components, which is able to define complex constraints. The condition and rules that define the constraints are applied as a second step after creating the object's geometry (Greetham, 2002; Aish, 2000).

This development has the potential to introduce the next step in parametric software in architecture, which is the ability to generate and work with complex geometric constraints (see Figure 6). However, the current user interface demands programming knowledge beyond that of the average architect, and a commercial version of the software appears to be some way off.

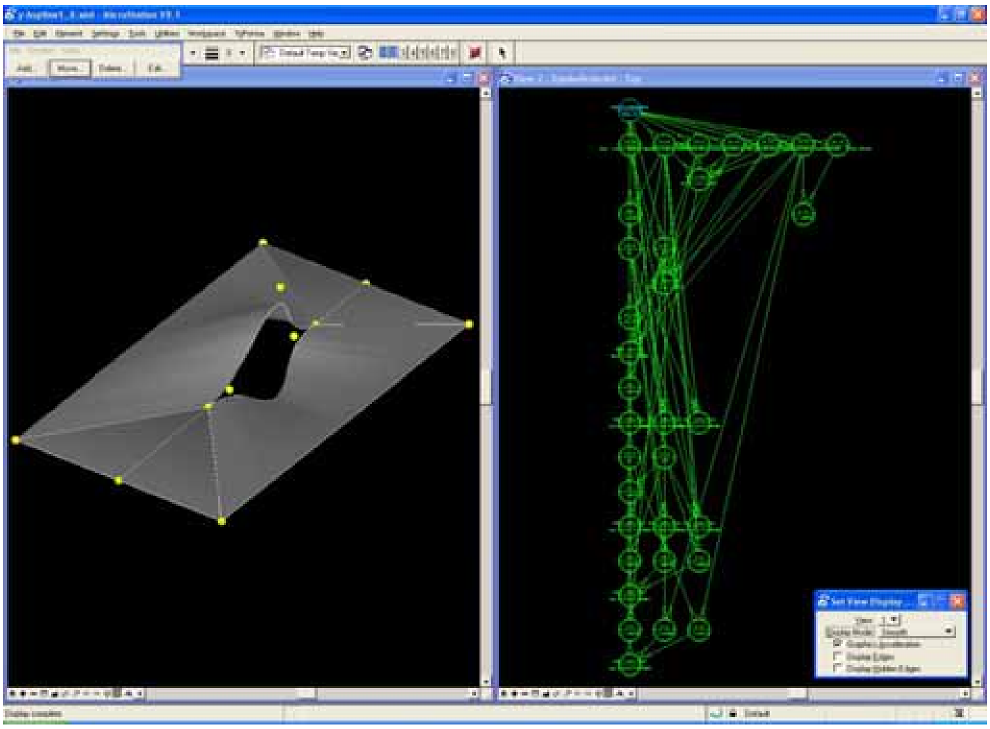


Figure 6 – Generative Component's interface – 3-D model with control points and a graph view presenting the interconnections between the objects

c. Modeling software originally designed for other professions – Most of the modeling software used by architects was initially developed for other fields like animation and mechanical/industrial design. The average architectural firm uses these software programs mainly for modeling and representation, either as a part of the design process or for representation of its final product (Mirza & Nancy, 1999). Another important use is the design and manufacturing of free-form elements. Software programs that were created for industrial and mechanical design can create and manipulate complex forms. These programs can also directly export to CNC and RP machines and more importantly, contain integral modules that calculate real material performances. Among the leading software programs in this category are: SoftImage (www.softimage.com); Rhino (www.rhino3d.com); TrueSpace (www.caligari.com); Form Z (www.formz.com); Solid Edge (www.solid-edge.com); SolidWorks (www.solidworks.com); CATIA (www.catia-ibm.com); Pro/E (www.ptc.com); and Maya, Alias Studio (www.alias.com).

d. Simulation and evaluation software – This category refers to software that is usually used by professional consultants to calculate the design's adherence to programmatic demands that have to do with performance factors such as acoustics, wind, energy conservation and efficiency and others. This category is discussed in detail in chapter 8.

e. Generation software – This category refers to software that generates architectural forms/plans/elements and is discussed in detail in chapter 8.

5.3 3-D editing

One of the most important aspects of computer-based design is the ability to modify or manipulate complex 3-D forms, which can be done in four basic ways via the existing modeling software programs:

a. Modifying single a object's vertex, anchor and grips position. A vertex or several vertexes are moved manually by the designer (see Figure 7).

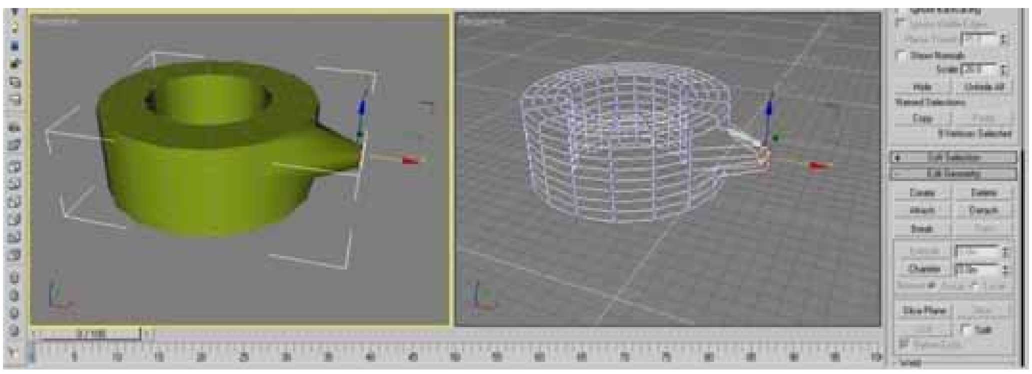


Figure 7– Modifying a 3-D single object by moving the vertex, thus changing position parameters (Autodesk VIZ 4)

b. Parametrically changing the shape of a simple object (usually called the “primitive” object, although in some software programs, the primitive list include topologic objects like Torus) (see Figure 8)

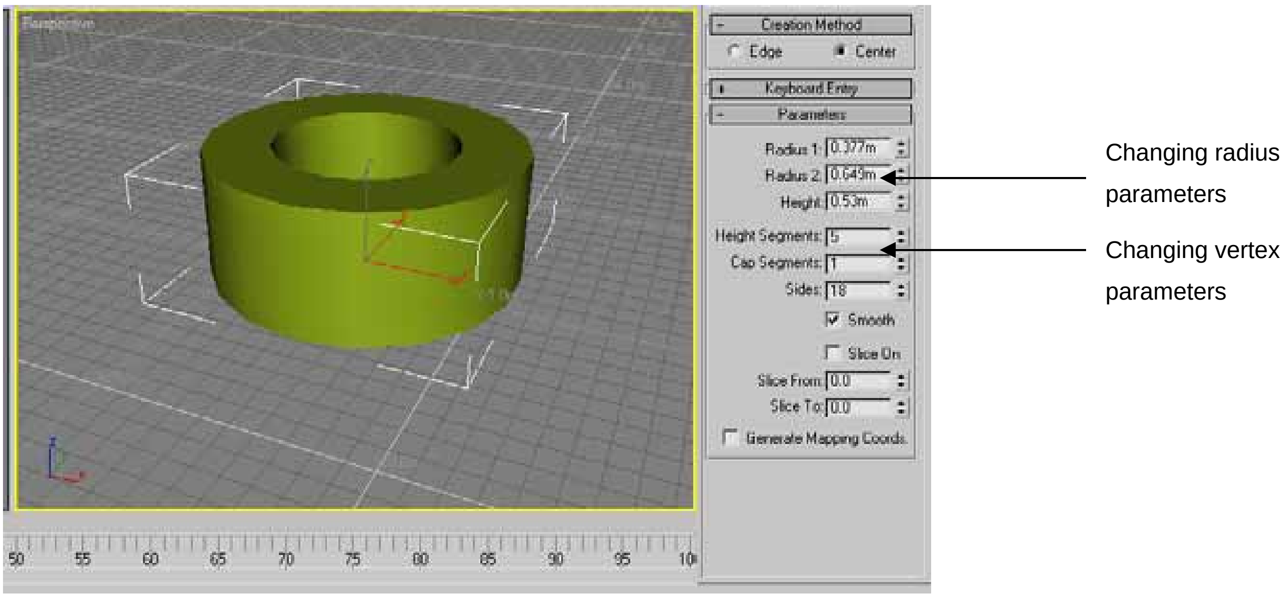


Figure 8 – modifying parameters of a simple parametric 3-D object (Autodesk VIZ 4)

c. Applying a modifier from a modifier list (included in any 3-D modeling software). A modifier is a predefined set of actions that change 3-D form. It can be a singular, stand-alone process or a dynamic parametric process in which the degree of the modification is controlled by parameters. The modifier list usually includes parametric modifiers (bend, taper, skew, stretch, ripple, squeeze, slice, displace and many others), Boolean operations (subtraction, union, intersection, etc) and mesh editing operations (optimize, tessellate and many others).

d. Parametrically modifying a set of objects with defined geometric relations to each other (Figure 9, Figure 10).
The ability to introduce geometric constraints to the modifying methods is limited to basic linear constraints such as being fixed in place or free to move, as well as looking at a certain object and following the way it moves. The author is not aware of any existing software which offers 3-D manipulation capabilities that is limited by predefined geometric constraints[9].

[9] This refers to the existing menus; some software has the ability to create macros using internal programming that defines constraints. Microstation and Microstation- based CustomObjects have limited capability to make 3-D modifications, which will be discussed in the next section.

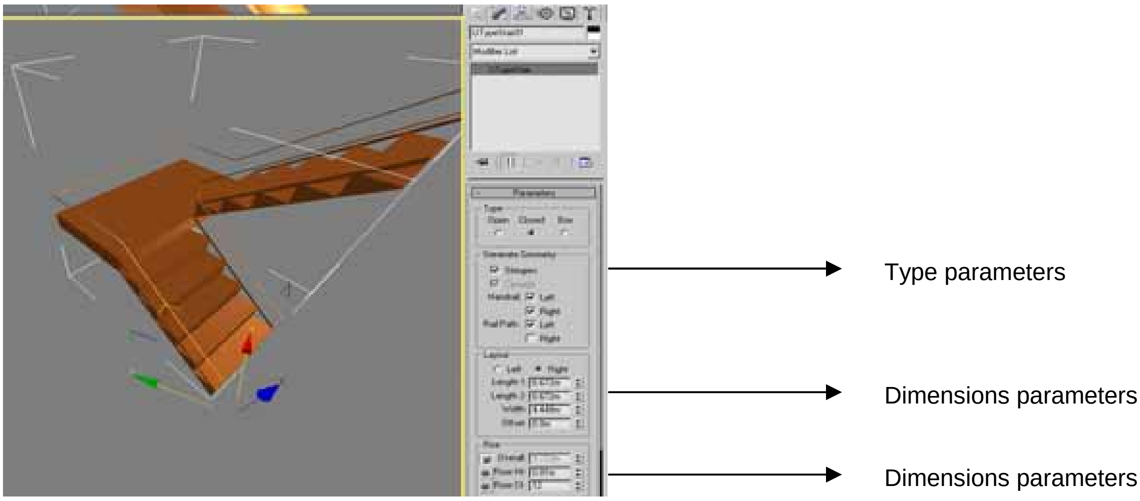


Figure 9 – Autodesk VIZ 4 interface for parametrically modifying a 3-D complex object – stairs

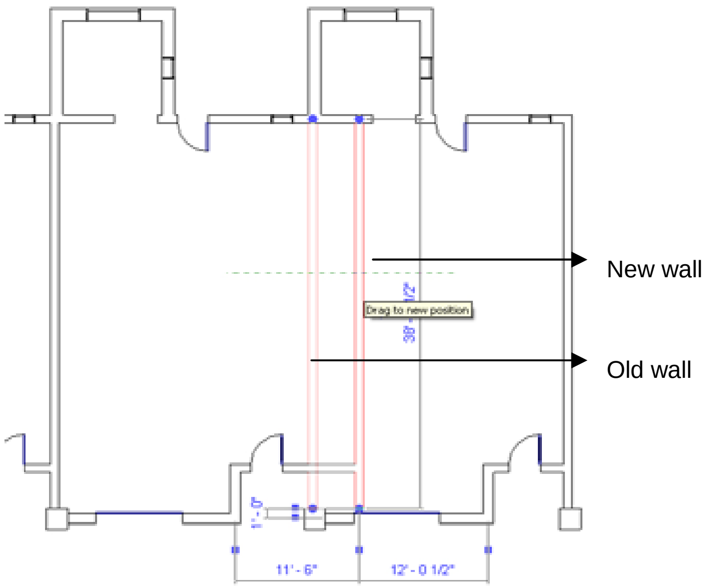


Figure 10 – Parametrically modifying a 3-D wall within a plan (Revit)

5.4 Parametric design/modeling

The term parametric design implies using parameters to define form[10]. But is every form that is parametrically defined a parametric design? A line drawn in any CAD software can be defined by two parameters (length and direction). A box is modeled in most 3-D modeling software by inserting parameters of height and length. These examples, although they use parameters to create form, are not considered to be parametric design. What parametric design does is to create functional/geometric relationships between objects (forms or a group

[10] Parametric design is also known as relational modeling, variational geometry/design, constraint-based design, associative geometry and rule-based design.

of forms/families) so that one can change certain features of the forms while retaining the basic predefined functional/geometric relationships. Changing features while retaining a basic logic gives the designer a dynamic tool to examine design alternatives, which is one of the biggest advantages of parametric design. In traditional CAD design or “explicit design”, a new alternative is created through a “sequence of erasures and redrawing (Burry and Murray, 1997)." A parametric model can be updated “automatically” by changing one of its parameters. This “shows great potential for rapid investigation of performance-driven design variants within the bounds of defined geometrical constraint” (Szalapaj, 2001).
A further refinement is using constraining parameters to preserve geometric or programmatic relationships, which is a valuable tool for architects. For example, in a function-driven project, such as a stadium, one can determine important constraints such as line of sight that must be kept in order for the building to “perform well." Another aspect of parametric design has to do with inserting material properties as constraints. For example, in the design of a free-form façade, introducing constraints could limit the building’s curvature according to the material properties. Changing the façade material (inserting different constraints) can open up different curvature possibilities. In an explicit design, this would mean designing and drawing an entire new façade, while in a parametric design, this could be done just by changing the set of constraints.

5.5 Parametric design precedents in architecture

The idea of constraints and parametric design software can be traced to Ivan Sutherland's pioneering thesis published in1963 (Sutherland, 1963). In the late 1970s, the first application that allowed specification of geometric constraints was introduced (Monedero, 1997). Since then, many applications based on parametric design and constraints have been created, mainly for research purposes.
Today parametric design applications are used in architecture mainly for the following purposes:
a. Design of building elements – Building elements, as opposed to entire buildings, do not require a complex set of constraints in order to follow all the interconnections defined by the brief[11]. Therefore, building elements can be designed entirely parametrically, without limitations of computer power or human perception. In this sense, building elements are designed in a similar manner to mechanical and industrial design objects. Moreover, no architectural parametric software that integrates material properties has been developed so far, which means that the design has to be performed using applications developed for industrial and mechanical designers. Software programs often used for this purpose are

[11] Another factor that discourages the use of existing parametric software for building design is the amount of time needed for developing a parametric model, which is not economically justified for a single building as opposed to, say, a model of a car or an airplane that will be mass-produced.

CATIA, Pro-engineer and Solidworks.

b. Optimization of building elements – This refers to the optimization of elements like facades to fit materials and manufacturing demands using parametric applications mainly designed for industrial design or mechanical, structural engineering.

c. Development of design alternatives – Parametric design has been employed in a limited way for several years, mainly in the U.K., to develop design alternatives using custom self-made macros in Microstation (Foster and Partners, KPF, ARUP, N. Grimshaw and Partners and others). Usually used for the design of geometrically complex buildings, parametric design is a tool that enables the modification of the geometry without the need to redraw the entire building. In terms of cost effectiveness, the time invested in developing the custom-made parametric model is worthwhile because it saves time that would have been spent on redrawing the building after geometrical changes and on drafting sections and plans that are produced automatically from the parametric model. Moreover, the parametric software programs are able to export data directly to manufacturing machines as CNC cutter/router and RP machines.

5.6 Parametric design and the implications of moving over to one-model building for the design process

One of the leading assertions concerning the use of computers in architecture suggests that architectural design will be developed by a shared computer model that includes the architectural design along with all the consultants' systems designs (Schodek *et al*, 2005). A similar design process is already used successfully in the automobile, shipbuilding and aviation industries[12]. In shared computer model design or one-model building, as it is often described, the computer architectural model is developed in 3-D using parametric objects that embed information on descriptive and performative traits.

The transition to one-model building facilitates collaborative work, which is expected to influence many aspects of the building process including[13]:

Building performance – The shared model makes it possible to perform simulations directly upon the developing architectural model in all stages of the design process, including structural analysis, energy simulation, computational air flow studies, equipment performance simulation and lighting and acoustic simulation.

---

[12] The first example of using a 3-D shared model is the design of the Boeing 777 aircraft. In the late 1980s, Boeing started design of new 777 Boeing plane using a 3-D model-based system supported by Fly Thru in-house application. Due to this change in the design and manufacturing process, "The Boeing 777 is being assembled faster, more accurately, and less expensively than any previous airplane in Boeing". (Abarbanel, 2000). The move to one model design in Boeing decreased the number of engineering changes during the design process in about 400%. The move to one model design in Boeing decreased rework at 60%-90% (Abarbanel, 2000)

[13] Based on information in http://www.gehrytechnologies.com/company-digital-practice.html

Digital contracting and cost estimation – The digital 3-D model provides accurate and comprehensive data concerning geometric and non-geometric project information that is directly connected to the building's 3-D elements. An accurate digital project model greatly decreases the ambiguity and potential disagreements about project quantities and can be used to produce quantity takeoffs and piece counts, and for automatically extracting quantities to spreadsheets for calculating cost estimates.
Digital fabrication – The model can be exported directly to manufacturing machines, whose shop drawings may be submitted in 3-D form back to the master project database. This allows for tight quality control of the construction process from design through fabrication.
Site integration – It enables integration with digital surveying technologies and the placing of digital information directly in the hands of construction personnel as they work in the field.

The transition to one building model also implies several fundamental changes in the architectural design process:
Architectural communication language – Contemporary architects communicate mainly by plans, sections and elevations. The number and location of both vertical sections and horizontal sections (plans) are connected with the amount of information needed for design development and manufacturing. The decision regarding the amount of information is always limited by the time one can invest in producing these drawings. In 3-D computer design, sections and plans can be generated directly from the 3-D model. The location and the number of sections can be changed with a click of the mouse. Moreover, various types of views are possible: Perspective, sections and isometric views can be extracted with the same amount of effort as plans and sections. Therefore, in terms of design development, plans and sections lose their leading role in communicating the design.
Another reason for the shift in the language of architectural communication is the increase in the formal complexity of architectural form. Complex forms, especially curvilinear ones, cannot be described by plans and sections within the constraints of reasonable time and cost[14]. Therefore, these types of forms have to be sent to other designers and consultants via 3-D model.
The shift in architectural communication language has not yet been studied. The scope of this research does not allow for further investigation in this direction, but we see potential for research in this topic.
Architects' position within the building's design process – It is reasonable to assume that as architects' 3-D models become the direct source of information for CAM, thus obviating the need for translation by a builder, architects' role and responsibilities in the design process will grow. The increasing responsibility might affect the centrality of the

[14] Describing these forms would require a larger number of sections/plans than are likely to be drawn in a contemporary practice, since it would not be affordable.

architects’ role in the design process, which has diminished due to the rising complexity of architectural projects and the amount of professional consultant it calls for. While the scope of this research does not allow for further development of this topic, we see potential for research in this direction as well.

5.7 Closing remarks

This chapter primary discussed the evolution of computer-based design tools. It described the way these tools developed, starting from an efficient substitute for the drafting tables and moving toward 3D-based parametric design applications. Within the 3D-oriented applications, five categories were identified: drafting and modeling software; parametric software for architects; modeling software originally designed for other professions; simulation and evaluation software; and form generation software. The second part of this chapter discussed the implications of parametric design. Parametric-based software allows not only the representation and deformation of complex forms as the conventional 3-D design applications, but also makes possible the definition of relationships between objects (constraints) by programming. The shift toward parametric software is changing architectural communication language from its traditional reliance on plan and section toward dependence on the 3-D model. This shift, together with the new technologies of computer-aided manufacturing, is expected to change the design and building processes and with it to increase architects’ responsibilities and strengthen their status and leadership within the building industry.

The next chapter will examine the implications of these changes on the design process and define the scope of this research within these changes.

6. Design process

"….Instead of trying to validate conventional architectural thinking in a different realm, our strategy today should be to infiltrate architecture with other media and disciplines to produce a new crossbreed. Reducing everything to flows of data and quantities, the computer offers us exactly this possibility." (Bart Lootsma, 1999).

6.1 Introduction

In his book ,"The Architecture Machine," Negroponte observed back in1970 that computers will assist the design process in three possible ways: "1) current procedures can be automated, thus speeding up and reducing the cost of existing practices; 2) existing methods can be altered to fit within the specifications and constitution of a machine, where only those issues are considered that are supposedly machine-compatible; 3) the design process, considered as evolutionary, can be presented to a machine, also considered as evolutionary, and a mutual training, resilience, and growth can be developed." (Negroponte, 1970)

As discussed in the previous chapter, early computer software concentrated on the way computers would assist the design process. With the implementation of the new design and manufacturing technologies, it is reasonable to argue that the traditional design process will be changing, as Negroponte predicted, to adapt to these new technologies.

It is evident that the design process and design methods vary in accordance with the different scales of the design that is building, urban or regional scale. Moreover, different countries – and especially different cultures – show considerable variation in terms of the design process. Nevertheless, the discussion on changes in the design process with the introduction of the computer, and especially the arguments concerning computer-based form generation and evaluation, have ramifications that go beyond the geographic location of the design and the building scale.

The following chapter examines changes in the design process from two perspectives: first, from the standpoint of the architectural design process and second, from the amount of information the designer has about the design and the resulting increase in his/her control over the design.

These ramifications will be discussed in terms of the time and work flows, participants and their relationships. The discussion will also delineate the stages within the design process where changes have occurred or will occur with the introduction of advanced computer applications to architectural design.

6.2 Architectural design process

According to Segers *et al* (2000), the architectural design process consists of three basic parts: analysis, synthesis and evaluation. In the analysis phase, the designer explores, assimilates, orders and structures various kinds of information that might inspire him or her.

The synthesis should result in a formulation of design objectives in the form of sketches, models, etc. These results are then evaluated against some explicit or implicit criteria. From a procedural point of view, the architectural design process is often divided into the following stages[15]:

a. Research
b. Outline proposals
c. Detailed proposals
d. Final proposals
e. Production information
f. Tender documentation
g. Site supervision

Nir and Capeluto (Nir, Capeluto, 2005) abridged these stages to the following:
a. Brief
b. Conceptual design (exploring different design alternatives)
c. Schematic design
d. Detailed design
e. Execution (documentation and construction)

The design process involves communication and negotiation with several external contributors that consult, evaluate or have a decision-making role. These contributors include[16]:

a. Client – The person or entity that orders the project. The two basic types of clients are private and public. Generally speaking, in privately owned projects, decisions are negotiated between the architect and the client, while in public projects decisions are negotiated between the architect, the public authority and, increasingly, also involve negotiations with the public itself (the real "customer") that is being represented by community representatives and public, local, state and sometimes global organizations.
b. Consultants – Professional advisers that can supply consultancy in one or more fields. Of the two basic types of consultants, the first is required to meet the country's building laws (for example, mandating a construction engineer who will be responsible for designing the construction and signing off on the construction documents). The second type has to do with the burgeoning complexity of the architectural program. The architect can no longer specialize to the level of being able to define specifications and needs of many of the mechanical, electrical and electronic systems in the modern building. Moreover, in many cases the building has to respond to certain environmental, urban and other demands[17].
c. Authorities – Most urban and building plans have to be submitted to various local authorities for approval before commencing the construction process.

---

[15]http://www.riba.org/go/RIBA/Member/Practice_306.html, www.building.org.il, www.architecture.com.au

[16] This information is based on the author's personal experience in the design process in Israel and Europe.

[17] These could include, for example, acoustic, wind, solar and HVAC simulations.

Based on the previously defined design stages, Diagram 1 describes a traditional design process in terms of connection to external contributors. The complexity of the process limits the ability to make changes in the design that step back more than one stage in the design process.

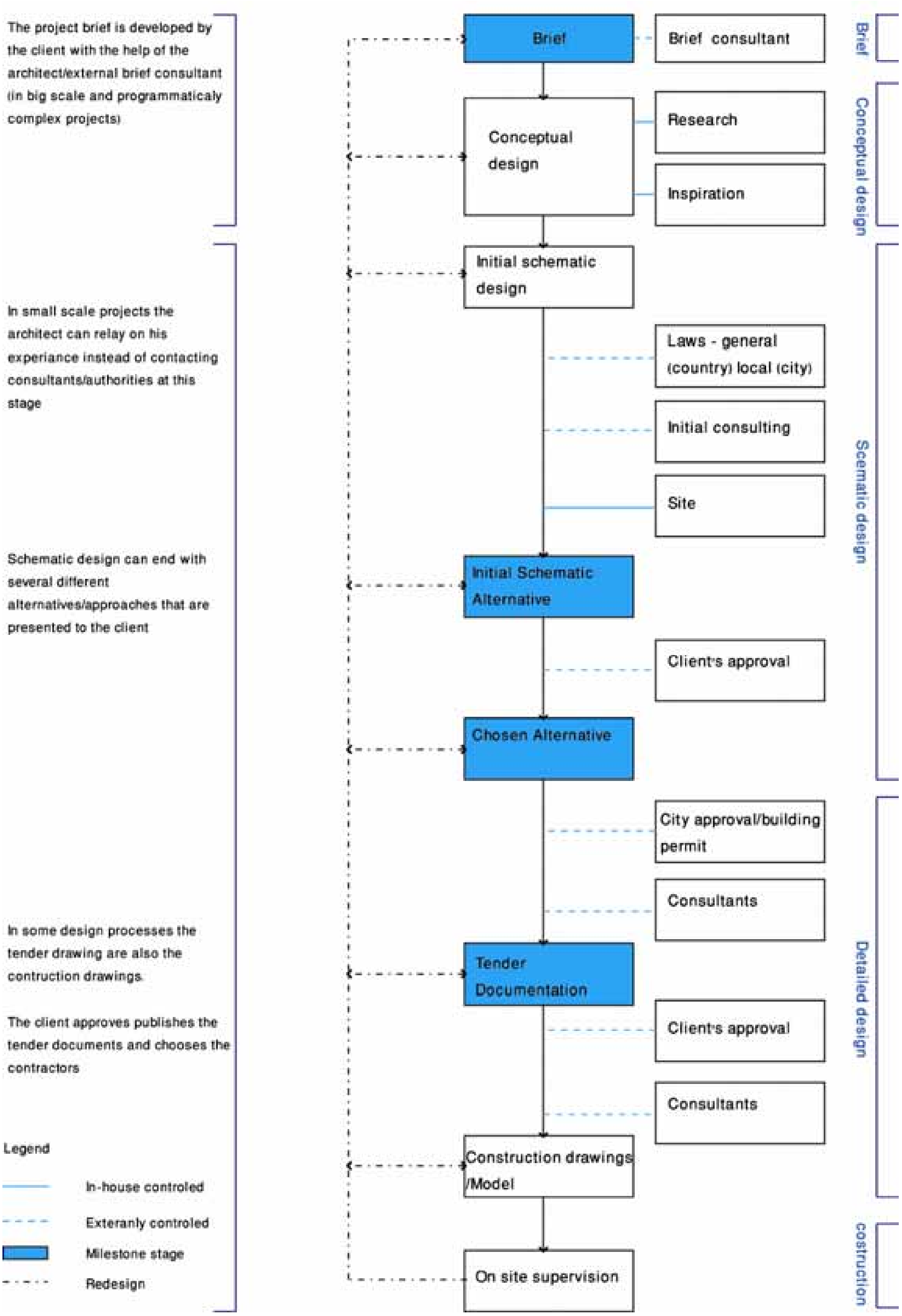


Diagram 1 – Architectural design process

6.3 Changes in the architectural design process caused by the introduction of computers

The computer's impact on the architectural discipline in general, and specifically on architectural design, can be distinguished in the following fields[18]:

a. Computer-based communication
b. Computer-based manufacturing
c. Intelligent house appliances
d. Social changes
e. Design process

a. Computer-based communication

The ability to manage and distribute information over computer networks such as the Internet has had a considerable impact on all stages of the design process. It influences design starting from the initial research stage and the search for precedents and inspiration, to the hunt for manufacturers and online catalogues and the search for specific details. Computer communication facilitated long-distance design by making it easy to share and send files and drawings. It also improved significantly the ability of different offices in different cities, countries or continents to work collaboratively on designs[19].

b. Computer-based manufacturing

The introduction of CNC subtractive machines, forging and bending machines, and various additive RP machines is challenging the traditional manufacturing methods, the building process flow and the notion of standards in architecture. Also, it holds far-reaching implications regarding architects' responsibilities and position in the building industry, by virtue of their new role in producing information from which the building is directly built/manufactured as opposed to the traditional drawings that need to be negotiated by the builder[20].

c. Intelligent house appliances

The introduction of microcomputers to home appliances enabled communication between appliances and their connection to a central computer control (smart/intelligent house). Moreover, new appliances are challenging old conventions about the distribution of spaces in

---

[18] This division is based on a synthesis of arguments made by Negroponte (1970), Mitchel (1990), Andia (1997), Kalay (2004).

[19]More information on this field can be found in Sanders (1996), Kalay (2004).

[20] More information on this field can be found in Rotheroe (2000), Callicott, (2001), Ryder *et al* (2002), Pham *et al* (2003), Schodek *et al* (2005).

our homes and the circulation within them, such as the centrality of the television set in our living room as opposed to the emerging home cinema[21].

d. Social changes

The use of handheld personal and computers networks is changing the way we live, work, travel and communicate. These changes affect both our needs in terms of space and the way we "consume" it.

e. Design process

In reference to Marshall McLuhan's famous aphorism “the medium is the message,” McCullough argues that "the medium one is working in and the tools one uses, necessarily change the design process, which in turn, necessarily changes the message, the literary work, or the architectural design in question" (McCullough, 2004). Like many new tools and technologies, new computer modeling and representation software was used initially in the same way as the traditional drafting board or 2-D computer design. It took some time until architects began to explore the possibilities of the new tools using new design processes and methods (Andia,1997; Terzidis, 2006).

Kalay (2004) divides the roles played by computers in the design process as follows: design tools, means of communication, design assistant, design environment and virtual environment.

Based on this division, Diagram 2 describes the impact of computers on the design process in terms of the type of change each design stage is experiencing. The diagram expands and transforms Kalay's abstract roles into singular activities that are connected to the design process stages. In this sense, "design tools" translates into productivity, accuracy, form generation and manipulation. Means of communication encompasses research, communication and productivity. Diagram 2 defines the changes in the design process in relation to three types of changes:

The first is **emergence,** which refers to new fields that did not exist prior to the introduction of computers to the design process. The diagram presents two such fields: computer-based direct form manufacturing and computer-based form generation. Although it can be argued that creative design prior to the computer did generate form, the designer who generated this form was clearly in control of the process and "expected" the results of the generation process. Computer-based form generation, though, follows Eisenman's notion of losing control (Eisenman, 1992), which begins when the result of the generation process could not be calculated by human within a reasonable time/cost.

---

[21] More information on this field can be found in Zehavi, Carmon (1996) and http://www.ercim.org/publication/Ercim_News/enw47/millar.html

The second type is **partial emergence**. This relates to fields that existed prior to the introduction of computers to the design process but were endowed with new possibilities. In this sense, collaborative design (network-based collaboration) in the early stages of the process became possible, while collaboration in the later stages became much easier mainly in terms of joint decision-making. In the same manner, the emergence of the Internet as a data source for architects changed significantly the amount of resources available to architects.

The third type is **changes** – referring to existing fields that experience a change with the introduction of the computer to the design process.

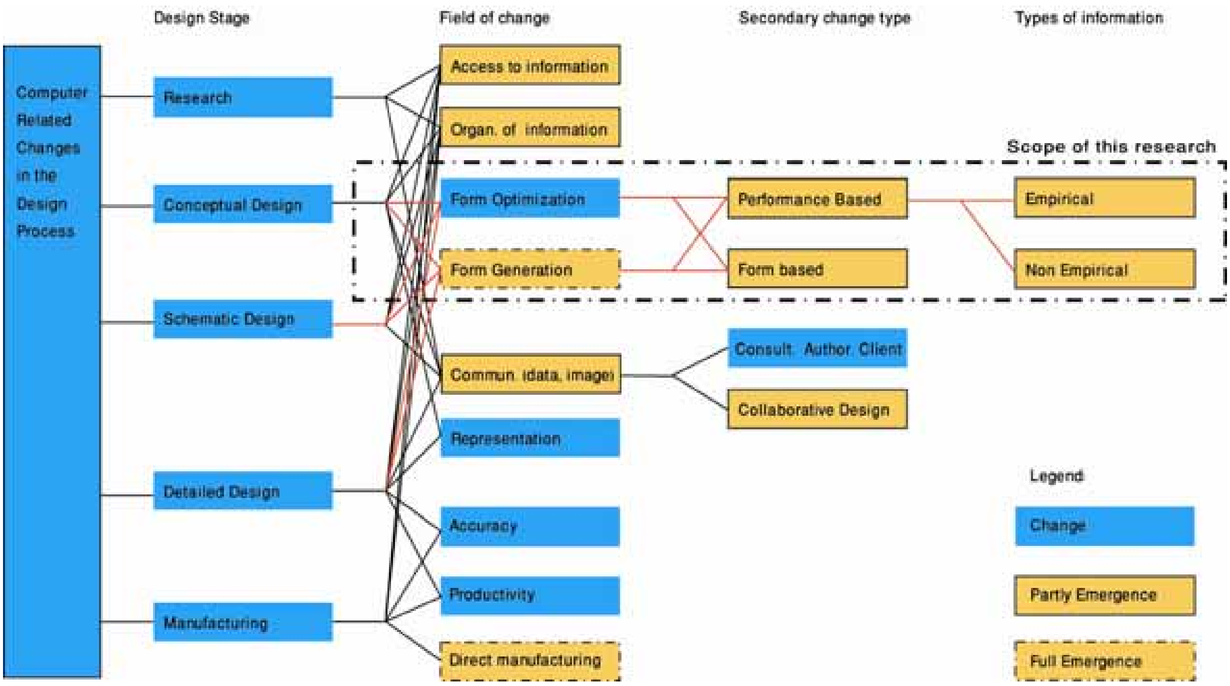


Diagram 2 - Impact of computers on the design process

Diagram 2 emphasizes that use of computer influence to a certain extent all the design stages. Nevertheless, from the fields of change described in the diagram only two fields fully emerge by the use of computers. Out of these two fields computer direct manufacturing has been researched widely within the architectural discipline and also in other disciplines. Therefore, we chose to concentrate in this research on computer-based form generation.

6.4 The increase in the amount of control/information architects have over the architectural form

Knowledge has always meant power and control. Many of the new form manipulation tools in contemporary design software produce formal results that could not be envisioned by the designers due to the complexity of the form (Beauce, Cache, 2003). This is, to a certain extent, “losing control” over the design process or collaboration with a new and highly capable partner – the computer[22]. Gaining control in this sense would mean having certain knowledge or being able to foresee the outcome of the design process or methodology (De Landa, 2003; Picon, 2006).

A new level of formal design knowledge is being developed using the computer as a generative design partner. Given the worldwide blossoming of creative digital design in the wake of the Information Technology (IT) revolution, the need for specification of “form processing” is increasing. Moreover, the need for a “multilevel model of design knowledge” that would permit both top-down and bottom-up operation – thus giving a designer a higher level of control over the design – is stronger than ever (Oxman & Oxman, 1990).

Architectural form, physical or virtual, can be described by two general types of information: descriptive and performative. Descriptive information has to do with dimensions and materials, and performative information relates to the form's performance. Performance in this context is used in its broader sense, which combines perceptual and phenomenological-related aspects with environmental and physical aspects.

Although the notion of control relies both on information and on action, it is logical to argue that increasing the amount of information about the design alone also increases the designer’s level of control.

But does increased control necessarily imply better design?

There is no straightforward answer to this question. The quality of design encompasses many factors that are only partly objective. Nevertheless it is fair to postulate that within the boundaries of the capability and talent of a single designer (or design team), the increase in the amount of information over the design is more likely to improve the design outcome (Hartog *et al*, 1998).

The present chapter will describe the evolution of the amount of descriptive and performative information designers have about the design following the introduction of the computer to the design process.

Echoing the launch of many new technologies, new computer modeling and representation tools were used initially in the same way as a drafting board in traditional design (Terzidis, 2006). Nevertheless, even in its initial role as drafting tool, computer-aided design introduced an important change in terms of the level of information it presented to architects and the fact that this information is within immediate reach. Moreover, since computers increased the efficiency of the drawing process, the time that was once spent on the more technical side of the design process can be now directed to other parts, thus improving the quality of the end solution.

---

[22] See also the discussion in Terzidis (2003, p 70) and Eisenman (Eisenman, 1992) on losing control over the architectural design.

The performative information introduced by computers about architectural forms or objects can be divided into two main groups: the first is related to dimensions, while the second is related to the object's physical properties. When drafting by hand, an experienced designer can calculate/retract dimensional information from the drawing. However, this process can be very time-consuming, especially when dealing with curvilinear or complex orthogonal forms or large quantities of objects. With computers, on the other hand, this information is usually only a mouse click away. In this sense the immediate information on the design's spatial preferences raises the designer's level of control over the design by giving him/her more information in less time.

The first generation of architectural software gave designers an immediate approach to descriptive information. Figures 11-13 present descriptive dimension-related information as shown in Autodesk products.

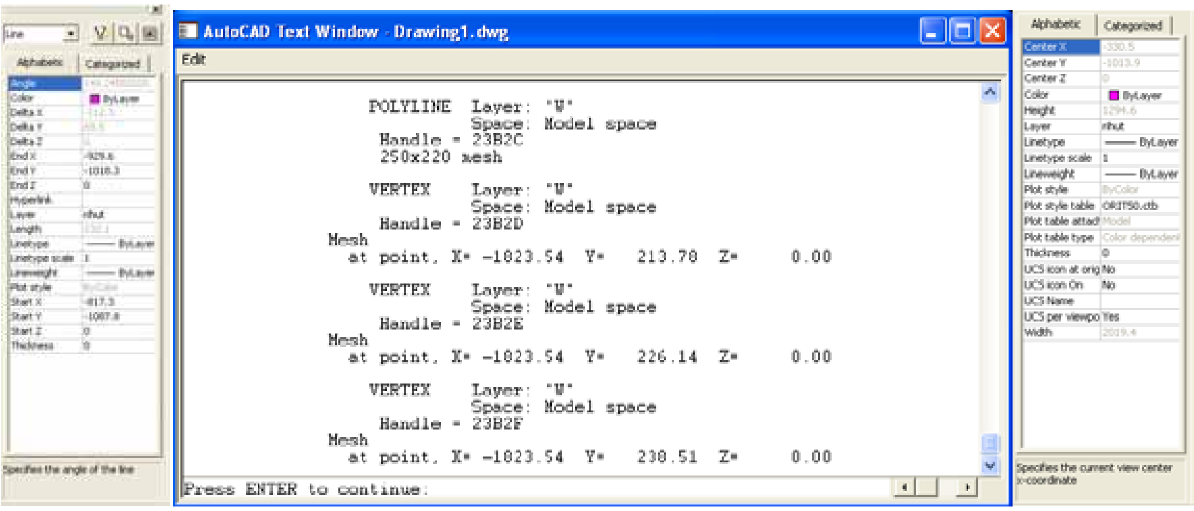


Figure 11 – Information on line segment (left), Polyline (middle) and Block (right) objects in Autodesk AutoCAD 2002

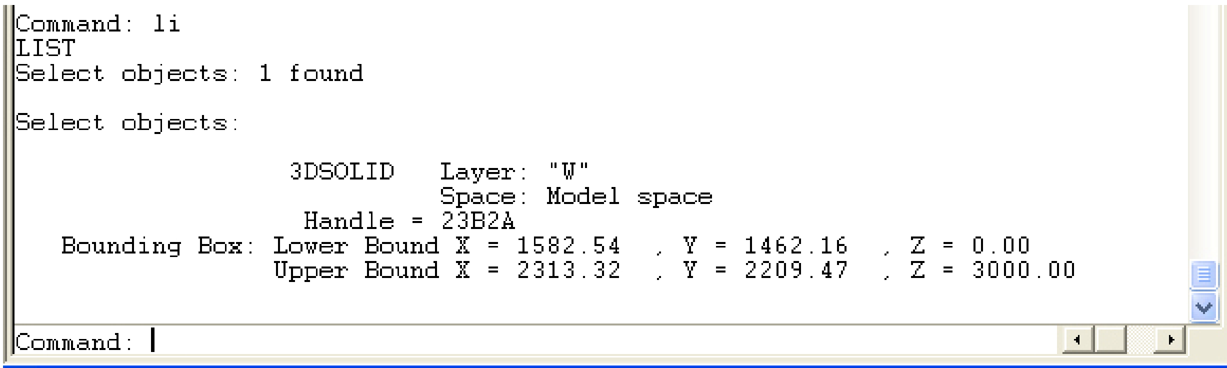


Figure 12 – Information on 3-D solid Autodesk AutoCAD 2002

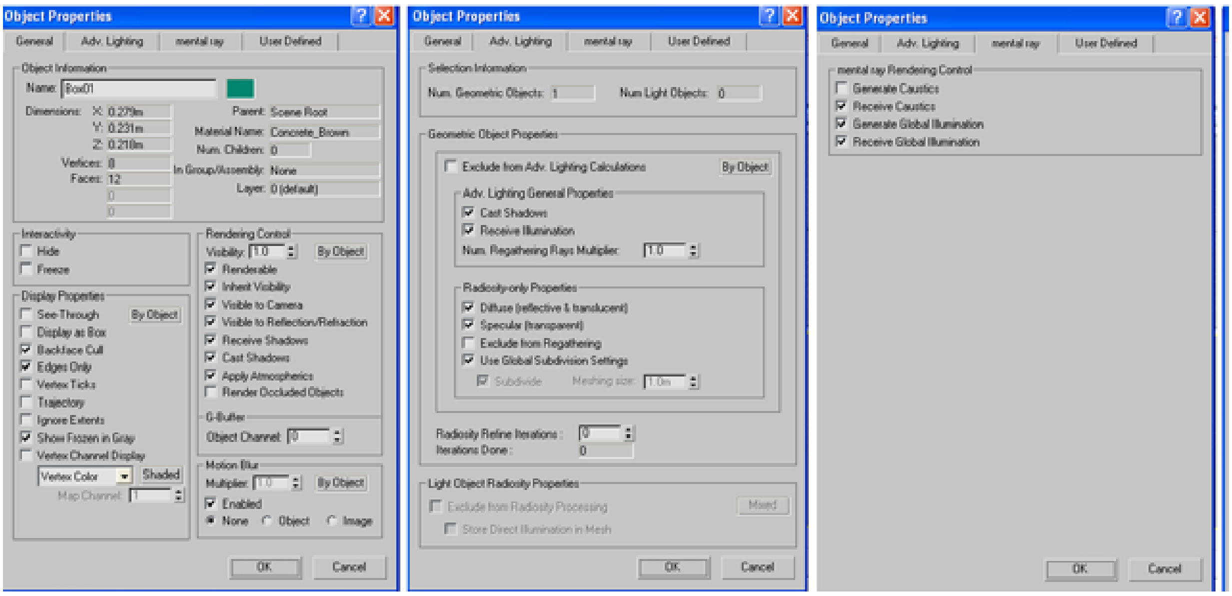


Figure 13 – Information on 3-D solid in Autodesk 3ds Max6

The second group, consisting of information related to the object's physical properties, was introduced to architectural design in the last decade with the advent of the new parametric software (see chapter 5). The development of descriptive information made available to architectural design by the new generation of parametric software can be described by the following changes:

6.4.1 Object-oriented design and building information model (BIM) [23]

In object-oriented design, the object is attributed with descriptive properties as size, location, orientation and material. These properties enable calculating, presenting and exporting different types of information that helps architects orient the generation of the design (Eggink *et al*, 2001). Object-oriented design implies shifting the architectural communication language from the traditional section and plan to a 3-D model. It facilitates exporting descriptive form information to fabrication machines as CNC and RP machines to be directly manufactured. Architecture software's transition to the model based on BIM and object-oriented design made it possible to base a design process on one 3-D model (one model building design is described in section 5.6).

6.4.2 The introduction of the notion of time or dynamic information

The ability to change parameters over time introduced the notion of animation to architectural design. Animation is used as a mode of representation and as a tool in computer-based design methods (see chapter 7). Additionally, and no less importantly, the introduction of

[23] BIM operates as a database that stores all building information. It facilitates a new kind of objects that embeds information on the physical world and translate it to an abstract computer representation (Magdy *et al*. 2001)

time presented a new capability to program dependencies and constraints between vertexes[24], singular objects and groups of objects (see Figure 14), which transforms the inert object into a "smart object" (Ibrahim *et al*, 2001).

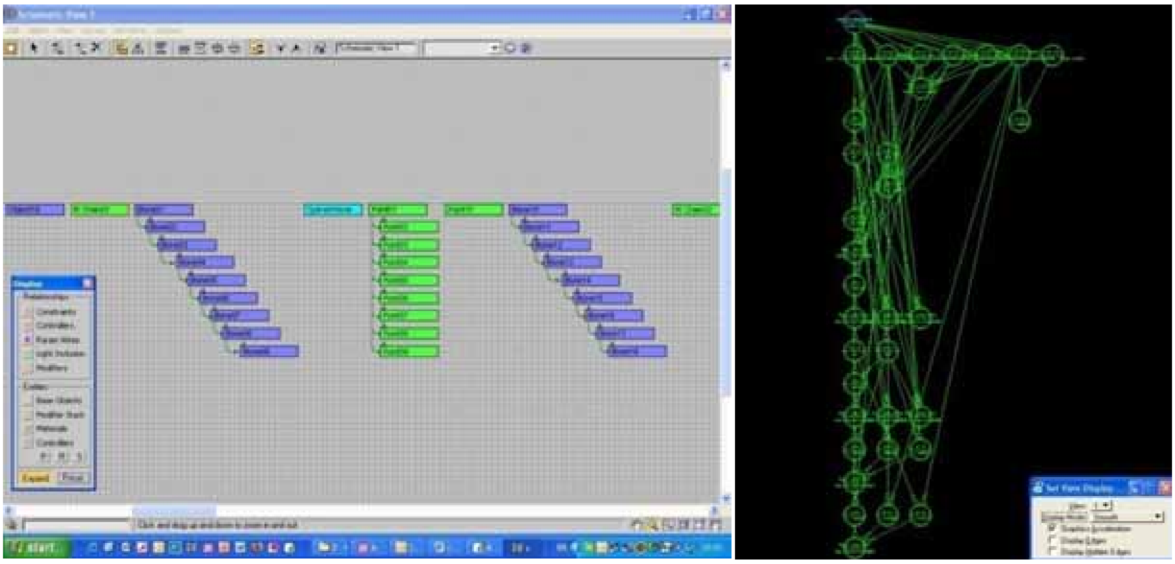


Figure 14 – Graph editor - Schematic view presenting relationships between objects in Autodesk 3ds Max6 (left) and relationships between objects in Bentley's Generative Components (right)

6.4.3 Real physical traits

Since the introduction of the notion of time to the design process and the ability to change objects' parameters in time, it became possible to develop algorithms that combine this capability with descriptive information on the objects to simulate the genuine behavior of real physical objects under various conditions. These algorithms are being developed continually to simulate and optimize the examined parameters with greater accuracy. The notion of computer-based form optimization will be discussed in chapter 8.

In terms of the design process, some researchers argue that these features will cause considerable changes to the entire design process and will not be limited to the scope of any design stage. The fact that the geometry is connected via parametric roles and constraints will facilitate modifying geometrical decisions that were taken at any stage of the design process. This will increase flexibility in terms of the ability to modify the design, decrease the costs of design modifications and help adhere to criteria that will be programmed within the relationships between the various objects (Nir, 2006). Although theoretically valid, this argument is not likely to fully withstand the demands of architectural practice. The complexity of the design and the amount of information it contains increase significantly as the design

[24] As in the Smart Cloud of Points model (SmCP) (Nir, 2006).

process advances. In the shipbuilding, aviation and automobile industries[25] (where parametric design and "one model design" have been implemented for more than a decade), it is usual to 'freeze' the configuration at the conclusion of certain design stages. This is done because the design involves collaboration with various subcontractors/designers that base their work on the preliminary data given by the main designer. Thus, it is not likely to assume that in architectural design, which is no less complex in terms of details and external collaborators, the design process could be performed without the need to freeze the configuration at certain stages during the design process. This freeze could apply to the entire form or only to certain of its parts or aspects. As in the naval, aviation and automobile industries, parametric design can be used to examine design alternatives and perform changes in the design within a certain design stage or within two adjacent design stages. Thus, it logical to assume that changing parameters beyond two design stages will occur only in small, specific parts of the building that would not influence the entire configuration, or under highly specific circumstances as in the case of a critical mistake in the design. Diagram 3 presents both the theoretically possible changes and the practicable changes in the design process using parametric design.

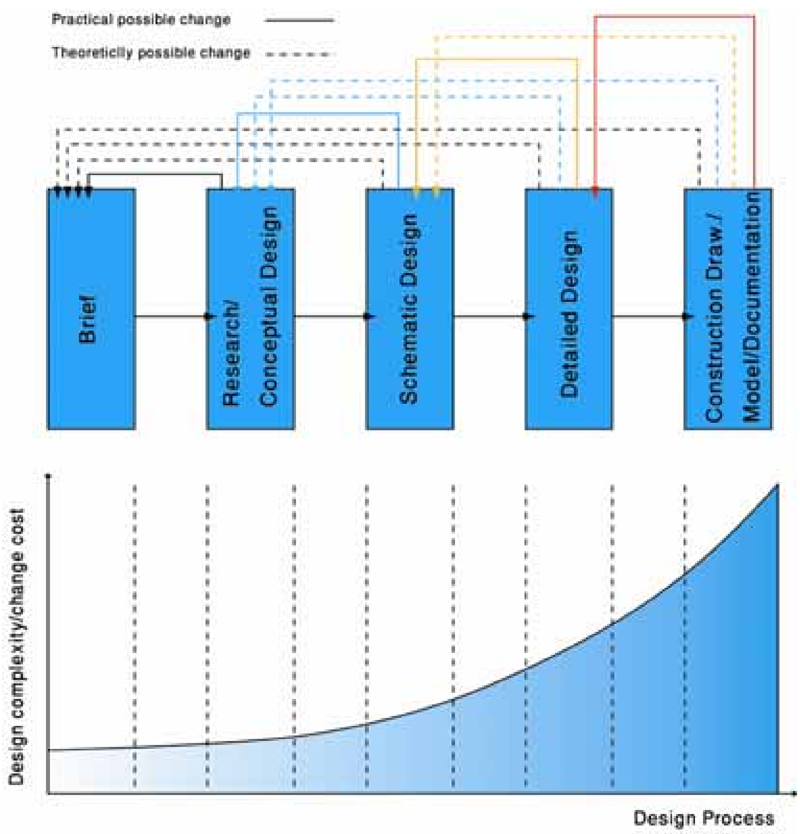


Diagram 3 – The possibility and implications of changes in the design process in parametric design (based on Yezioro, 1994)

[25] Only the shipbuilding industry is similar to architecture in terms of the singularity of the design product. Nevertheless, it can be argued that the complexity of buildings and the building design process will force designer to "freeze" configuration and not allow changes beyond a certain point in the design process.

6.5 Form-based design vs. performance-based design

There are two main approaches to considering the notion of performance in architecture. The wide approach suggests that performance includes physical, perceptual and cognitive aspects. If form follows function to a certain extent, then performance, in this sense, is the means to connect form and function.

The narrow approach differentiates between qualitative and quantitative criteria to define performances. Criteria are considered quantitative only if they can be empirically measured. Criteria that cannot be measured empirically, like most of the cognitive and perceptual aspects, are considered qualitative. Under the narrow approach, performance includes only quantitative aspects. The notion of performance in this research follows the narrow approach. Thus, perceptive and cognitive oriented criteria will be regarded in this research as form-related (not performance) design. Accordingly, design based on quantitative criteria will be defined as performance-based design. Hypothetically, every design process can be regarded as a design method (design method in this sense is a reason-based process that controls form deformation[26]). A design method can be based solely on intuition or be based on qualitative and/or quantitative information. A designer can follow one design method, combine several methods and thus create new methods or work in a completely intuitive manner, as in the case of Coop Himmelblau's architects who occasionally create the project's initial scheme sketching with their eyes closed. It can reasonably be argued that every design process has both formal and performance aspects. This research does not try to contradict this claim but argues that the increase in the amount of information architects have over their design inevitably moves the design toward the performance side.

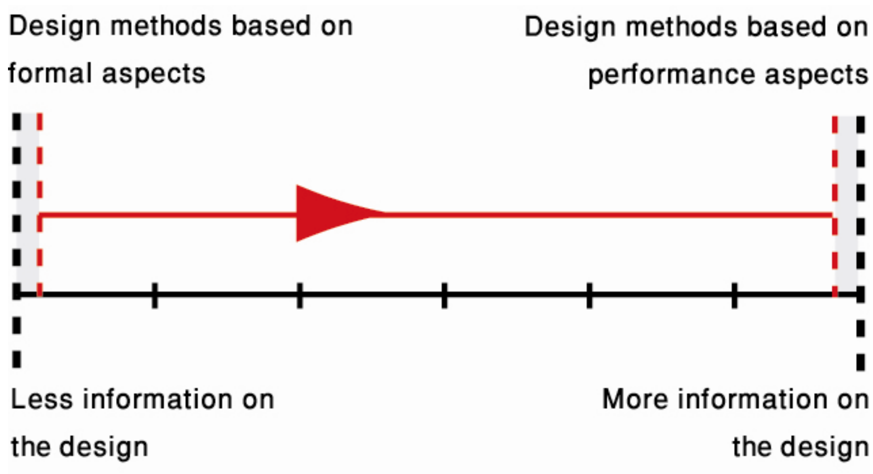


Diagram 4 – The influence of the computer's introduction upon design methods

---

[26] A design process can be based on a single method, incorporate several methods or be totally intuitive (without any coherent method). Computer-oriented design in this sense resembles any other architectural design, in that it can be intuitive or follow one or several design methods.

Nevertheless, moving toward performance-based architecture does not mean that architecture will reach a point where the final form will be based only on performance.
As already stated by some researchers, architectural projects are too complex to be reduced to parameters or mathematical formulas (Schmitt, 1992). As well, it is also impossible to create architecture based on only formal aspects since the structure has to stand in the physical world, which implies that it has to resist forces and thus adhere to some performance-based rules. Diagram 4 describes the tendency toward more performance-based architecture and its boundaries and the increase in the amount of information on the design it conveys.

6.6 Empirical and non empirical criteria in form and performance based design

Following the above division into form-based[27] and performance-based design, it is possible to divide in a similar way the information used by architects to generate architectural form. Since "there is indeed a fundamental difference between the quantitative nature of computation and the abstract holistic nature of human thinking" (Terzidis, 2006), the aim of this division would be to point out possible fields where computers can be used in the design process.
Diagram 5 presents types of information that can be used to generate architectural form, divided into form-based and performance-based. It also differentiates between information types that can be empirically measured and other types that are not empirical and can be subjectively defined.
The diagram’s first level divides the information used in the creation of the architectural form into form-based design and performance-based design. Form-based design includes three parts, the first of which is information that comes from the designer’s reaction to the design problem. It consists of the following:
Site – formal intuitive decisions connected to the designer’s reaction to the site
Form manipulation – formal decisions made intuitively by the designer
Style/canon – formal expression related to the designer’s decision to adhere to a certain style.
The second part relates to information that creates limitations that force the architect to choose a specific formal direction. It consists of the following:
Brief – form or formal expression directions that are instructed by the brief
Cost – formal expression directions and limitations that are driven by cost demands
Manufacturing – formal expression directions that are driven by manufacturing demands**.**

[27] See another definition for form-based design in Terzidis, 2003, p 48.

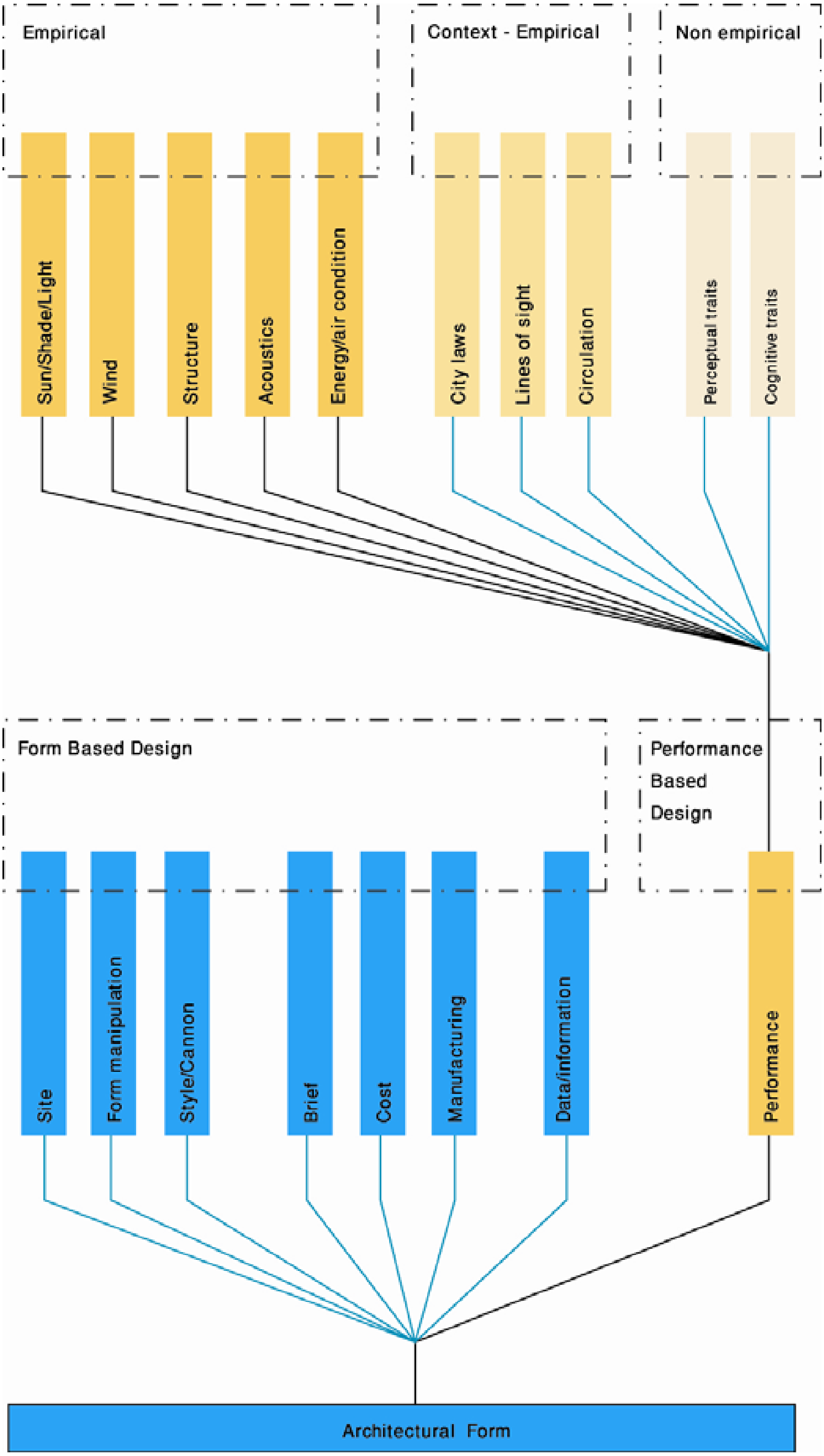


Diagram 5 – Generators of initial form – division into form and performance orientation

The third part of form-based design has to do with generation of form based on information that is not connected to performance. This type of form generation will be discussed in chapter 7).

The diagram's second level argues that there are three main types of performances from which one can empirically generate architectural form using their information. The three types are empirical information, contextual empirical and non-empirical.

Empirical information relates to traits that can be measured in a straightforward, objective numeric manner. These traits consist of light (sun and shade), wind, structure, acoustics and matters related to energy. Discussion of these traits will be developed in chapter 8.

Contextual empirical refers to traits that can be used in an empirical manner to generate architectural form in a specific contextual condition. The three types are city laws, lines of sight and circulation.

City laws – boundary information regarding the dimensions of the future project

Lines of sight – demands or limitations regarding specific lines of sight, as in projects for performing arts centers and stadiums

Circulation – demands or limitations regarding specific types of circulation and flow rate in particular projects, such as airports and hospitals.

Non-empirical performances refer to qualitative traits that were discussed earlier in this chapter.

It is important to mention that simply because the above diagram divides the information used to generate architectural form into different types does not mean that architects use a just one type in the design process. On the contrary, it is more logical to assume that different types of information are used in every design process and it is unlikely to find a design process based on a single type. Nevertheless, the division is necessary in order to understand where and how the quantitative nature of computers can be used in the design process.

6.7 Closing remarks

The chapter presented the contemporary architectural design process in terms of the different stages it contains, the work flow, the participants and the relationship between these elements. It then defined five domains that affect the architectural design process in which the introduction of the computer led to changes: computer-based communication; computer-based manufacturing; intelligent house appliances; social changes; and design process. From these domains, the design process was defined as the main scope of this research.

Examining changes in the design process revealed that due to the introduction of computers, two new fields emerged: computer-aided manufacturing and computer-form generation. Since computer-based manufacturing has been widely discussed in academic research, it was decided to concentrate on computer form generation in this research.

The chapter also discussed changes in the design process in terms of the amount of information and control architects have over the design since the introduction of computers. It suggested a division into descriptive and performative information and argued that the increase in the amount of information architects have about the design with the use of computers improves the design results. Following the division into descriptive and performative, the information on which the architectural form is based was divided in this chapter to form-based and performance-based. An argument stating that architectural design is shifting toward performance-based design was developed. Within performance-based information, a secondary division was made into empirical and non-empirical information and several empirical fields were identified.

The next chapter will discuss design methods, which are another aspect of the design process. Then, drawing on the insights from these two chapters, the following chapters will concentrate on computer-based form generation and optimization.

7. Design methods

7.1 Introduction

According to Cross (1989), design methods are “any procedures, techniques, aids or ‘tools’ for designing.” He defines two features common to the many existing design methods: They formalize certain design procedures and they externalize design thinking in terms of making the designer work using external tools such as charts and diagrams. Jones (1984) defines design method as “a means of resolving a conflict that exists between logical analysis and creative thought." He claims that design methods are intended to have two effects: to reduce the amount of design error, redesign and delay; and to make possible more imaginative and advanced designs.

Kalay (2004) argues that "design methods are intended to provide designers with rational means that may help them initiate the design synthesis process and bring it to a successful solution."

Rittel (1972, in Cross, 1984) defines two generations of design methods. The first emphasizes the “expert knows best” attitude toward design process, namely, defining a logical linear process that should be followed step by step in order to achieve a worthwhile design. Broadbent (1979, in Cross, 1984) presents design methods by Asimov (1962), Jones (1963), Archer (1963/4) and Alexander (1964) as members of the first generation and states that all of them tried to apply the Cartesian method to design (breaking down the problem into parts and solving each part before trying to perform a “grand synthesis").

The second generation in design methodology, according to Rittel, has two main traits. First, the design process is no longer considered to be a sequence of activities that should be carried out linearly. Secondly, it is argumentative, meaning that the “statements made are systematically challenged in order to expose them to the viewpoints of the different sides, and the structure of the process becomes one of alternating steps on the micro level” (Rittel, 1972, in Cross, 1984).

Broadbent (1979, in Cross, 1984) assets that the two generations of design methodology did not actually achieve the goal of being used by designers. He suggested a third generation in design methods based on the Popperian view, according to which designers “do not know how people should live. They merely offer possibilities which people can take or leave."

Archer (1979, in Cross, 1984) asserts that one of the problems of design methods is their use of an alien language. “Design activity is not only a distinctive process, comparable with but different from scientific and scholarly processes, but also operates through a medium called modeling, that is comparable with but different from language and notation," Archer states.

Goldschmidt (2001) posits that the “design methods movement,” which has been trying to develop a design science, is still far from attaining that goal. Moreover, she suggests that in

“most if not all design domains, rigorous design methods based on well-defined algorithms do not yield the expected improvement in design quality. In some cases, particularly in architecture and in industrial design, it is very hard to get designers in the real world to even try them out." Goldschmidt (1997) also suggests that in design problem–solving, the solutions are almost never predictable because the design problems are “ill-structured.”[28]
Kroes (2002) states that the aim of design methods is to improve the design process, which is the main reason it has always focused on the nature of that process. He believes that design methods should become more product-oriented because “the design process and design product are so intimately related to each other that an understanding of the nature of the design process requires insight into the nature of the product designed and vice versa.” Another reason is that the normative stance taken by design methods toward the design process implies it has to address questions concerning the quality of the outcome of the process and/or the product.

The orientation toward the product is highly significant in the case of computer-based design, especially when it is associated with computer-based manufacturing. This research carries on with the notion that expert-based design methods cannot attain the aim of improving the design quality. Moreover, the orientation toward computer-based manufacturing when practicing computer-based design has a clear product orientation and thus cannot be considered within the existing process-oriented design method.

Therefore, the approach chosen for developing design methods in this research is bottom-up instead of top-down (the “expert knows best” approach), that is, examining and developing methods used by designers and design consultants and suggesting ways to implement and improve existing technology within these design methods. Nevertheless, the top-down approach was dominant in the computer-oriented domain until the beginning of the 1990s, when the computer started to be used extensively in architectural practice. The main directions in this domain will be discussed in chapter 8.

Computer processing power allows the handling and deformation of complex forms incomprehensible to the human mind[29]. Developing these forms can either be done intuitively on the computer screen or developed according to concrete design methods. The previous chapter (Chapter 6) discussed the influence of new computer tools on the design process

---

[28] Archer (1984) defines an ill-defined problem as “one in which the requirements, as given, do not contain sufficient information to enable the designer to arrive at a means of meeting those requirements simply by transforming, reducing, optimizing, or superimposing the given information alone." According to Goldschmidt, in a well-defined problem “the initial state is given, the goal state is either specified or it can be determined using stop rules, and the operators are controlled by known algorithms." In an ill-defined or ill-structured problem, "one or more of these constituents is either unknown or ambiguous” (Goldschmidt, 1997). In relation to the above definitions, Rowe’s (1987) definition of problems as "wicked" emphasizes the lack of definitive formulation and the lack of stop rules.

[29] See also Peter Eisenman's differentiation between the mechanical paradigm and the electronic paradigm (Lenior T., Alt C, 2002).

and the types/amount of information architects have on the design. In the following chapter we discuss the influence of the computer on design methods.

7.2 Linear and non-linear design methods

Although architectural design ends with a single built design, during the design process more than a single alternative is usually examined. In traditional design, prior to the introduction of computers to architecture, developing alternatives meant that individual labor had to be dedicated to every design alternative. The shift to computer-oriented design introduced two main changes in this domain; the first had to do with the increased ability to handle and deform architectural forms. This change allowed designers to produce more alternatives faster. It includes both 2-D drawings, which can be copied and manipulated easily to create alternatives, and 3-D virtual and physical models[30].

The second change has to do with the move to nonlinear design process, which uses computer processing power to generate design alternatives. In this sort of process, the designer controls the type and degree of deformations between the alternatives and the number of generated alternatives. Accordingly, alternatives can be generated in any design stage and several design alternatives can be developed at the same time.

7.3 Static and animated computer-based design methods

Although considered by some researchers as a paradigmatic change[31], the introduction of computers did not negate existing design methods. On the contrary, as Negroponte envisioned already in 1970, it both enhanced existing and introduced new design methods (Negroponte, 1970). Following the arguments presented in chapter 6 concerning the influence of the computer on architectural design process in general, and specifically the influence on architecture in terms of information, the following chapter will discuss the emergence of new computer-based design methods in architectural practice, in relation to static and animated design[32]. The division into static and animated design derives from the conclusions of the earlier discussion on types of information introduced to architecture by computers (see section 6.4).

7.3.1 Static design

The following main directions in design methods were identified within this domain:

---

[30] The possibility of producing additional 3-D physical model alternatives refers to direct manufacturing by rapid prototyping and CNC machines.

[31] Kuhn's definition of paradigmatic change is problematic in the architectural context since architecture cannot be considered as a science and since the current change did not negate the previous "paradigm," which is one of the definitions of paradigmatic change, according to Kuhn (1962).

[32] As defined by Lynn (1999)

Boundary condition – Innovations in architectural software tools for manipulating surfaces and the introduction of complex surfaces made by NURBS (Non Uniform Rational B-Splines) enabled architects to integrate, develop and use surfaces as a generative element in the design process in a simple graphic way. Research in this direction was carried out already in the '70s by Shaviv and Greenberg, who developed an algorithm to define surfaces by their edge conditions (Shaviv, Greenberg, 1970). Designing surfaces cannot be regarded as a design method in this context because there is no generative method that determines the general form of the design. Nevertheless, when used in a more substantial way, in a specific design scenario (where program and site are defined), the calculation of surfaces can be used with different inputs of material to define the formal constraints of the design solution, thus creating an envelope of possible formal solutions. An example of such a method is the use of freeform sections to generate design; the designer draws generative sections that define the contours of the main spaces. Then, a secondary layer (surface) that closes the gaps between the sections is generated by the computer. Figure 15 describes a section-based method project by NOX. In terms of morphogenesis, a section is needed each time that new information or a change in space is created. Therefore, the number of sections correlates to the amount of spatial information needed to represent/construct the design or the level of complexity of the form.

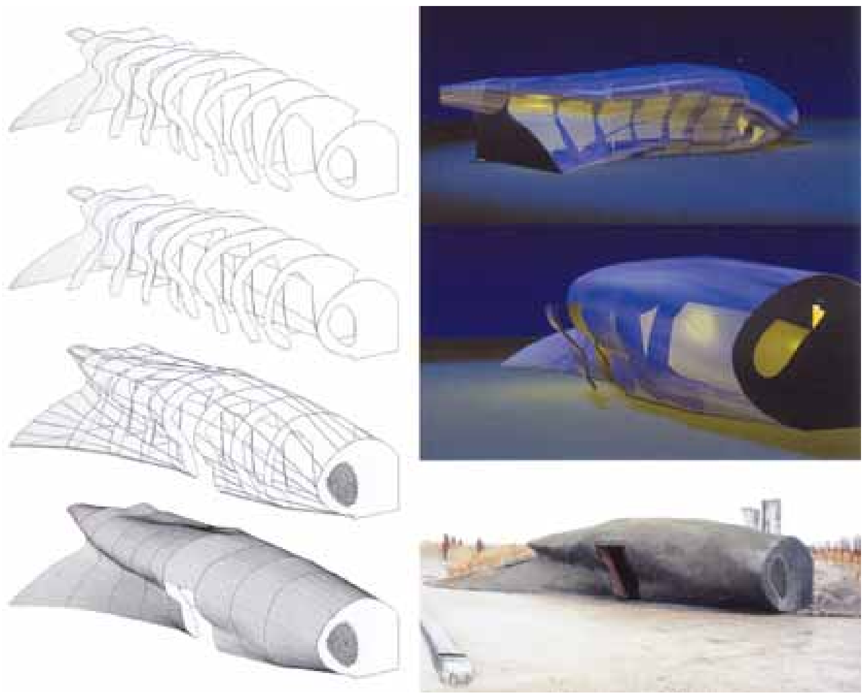


Figure 15 – NOX – section-based methodology design of a utility structure (Zellner, 1999)

In Figure 16, structural generative lines that were generated by a structural engineer were used as the base for the design form (Balmond, Weinstock, 2002).

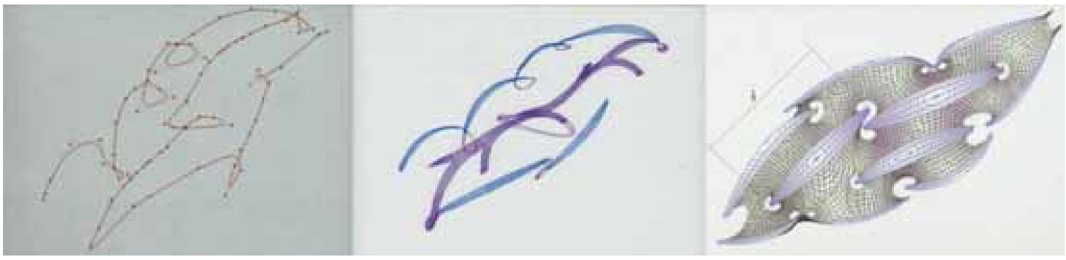


Figure 16 – Generative structural lines (Balmond, Weinstock, 2002)

System or tile-based design – This method refers to a formal approach to design, in which a 3-D tile or rule-based formal system is used to determine the project design. It can be regarded as an evolution of the notion of the 2-D grid into a new kind of grid based on a wetGRID, which is a 3-D deformable grid (Spuybroek, 2002) or a LEGO-based logic complex formal system. This method relies on the processor power of modern computers to create complex 3-D formal systems. The “tile” is a 3-D element that has the ability to connect to similar tiles to create 3-D structures. The computer power to dynamically handle 3-D forms enables designers to create and manipulate complex formal systems.

The most common use of tile-based methods is in the design of cladding systems, although it has been used in numerous buildings, building elements, furniture and installation design. The difference between tile- and system-based methods is that the tile-based method emphasizes the tile as a generator of formal order in the design, while the system-based method puts the emphasis on the formal rules that create the design.

Deformation design – This is a general name for several form-oriented design methods that involve modifying an initially chosen form by one or several modifiers[33]. The use of modifiers does not turn a process into design method. For the use of modifiers to become a design method, as opposed to intuitive design or technique, one must characterize a process in which meaning is given to the use of each modifier.

An example of deformation design method is Boolean operation digital design, a formal method used to create an initial form based on the site and program demands/constraints. This method uses Boolean operations such as union, subtraction and intersection to define the project form. The design process starts with an initial 3-D form (for example, one that is defined by the plot’s boundaries). Boolean modifiers are applied to this initial form. The size

[33] A general description of form modifiers that were introduced in 3-D software programs can be found in the paragraph describing digital tools in section 5.3.

and shape of what is usually called in the 3-D software Operand B[34] (given that the initial object is Operand A) is defined by various constraints derived mainly from the brief and the designer's concepts such as daylight needed, preferred views, traffic trajectories etc. The form created at the end of the process represents the 3-D boundaries of the future design.

7.3.2 Animated design

The notion of animation emerged from a development in visual arts at the beginning of the 20th century. It was an attempt made by visual artists to explore form relative to time. Artists like Umberto Boccioni tried to present the dynamic expression of movement in a train station, and Marcel Duchamp tried to describe the path of a man going down the stairs. Both tried to transfer the notion of time and movement to a 2-D surface. These attempts were encouraged by rapid developments in cinematic techniques and the invention of the motion picture. The difference between the orientation toward the recording of time in cinema and photography and toward its representation in the visual arts led to the creation of the art form called animation (More, 2001). The digital revolution in cinema is blurring the differences between cinema and animation. In fact, if we use the same line of thought to define digital cinema as the one used to define digital architecture, it can be inferred that "cinema can no longer be clearly distinguished from animation. It is no longer an indexical media technology, but rather, a sub-genre of painting" (Manovich, 1999).
Architectural animation started as cinematic recording of movement in space. The first architectural animations consisted of a camera moving in a digital 3-D space. This type of architectural animation, which is still probably the most popular, is done mainly for representation purposes. In his book "Animate Form," Greg Lynn differentiates between motion and animation (Lynn, 1999). According to Lynn, motion implies movement and action, while animation implies the evolution of a form and its shaping forces. The problem with the analogy to motion pictures, according to Lynn, is that "architecture occupies the role of the static frame through which motion progresses." Animate design is the implication of animation as an evolution of form. Animate design implies morphogenesis as equilibrium of a force field. In this sense, the form is created by dynamic forces that modify the initial form to its final stage. Thus, animate form does not have to move in order to be dynamic. The example cited by Lynn to demonstrate this idea is the design of a ship's hull, which represents the forces of water it will face. Animated design methods, as opposed to animation in design, rely on the premise of the new possibility to change the results of the form deformation process at any given time. Thus, when endowed with the ability to interactively modify the parameters of the form deformation, most of the static methods

[34] In a Boolean operation, an object is subtracted from another, intersected or joined (united) to create a formal solution. In the software, these objects are usually named Operand A and Operand B.

mentioned earlier in this chapter can turn into animated methods. The following main directions in design methods were identified within this realm:

Morphing – This term refers to a continuous transformation of one form/object into another where the computer calculates geometric steps that change one form to the other (see Figure 17). The designer decides on the number of "steps" between the two forms and the level of influence of each form on the result. Since the process is parametric, the final solution can be any of the steps between the initial forms or a combination of all the results. In this context, morphing is used in a wider sense, which includes not only geometrical changes but also changes in other parameters such as position and material.

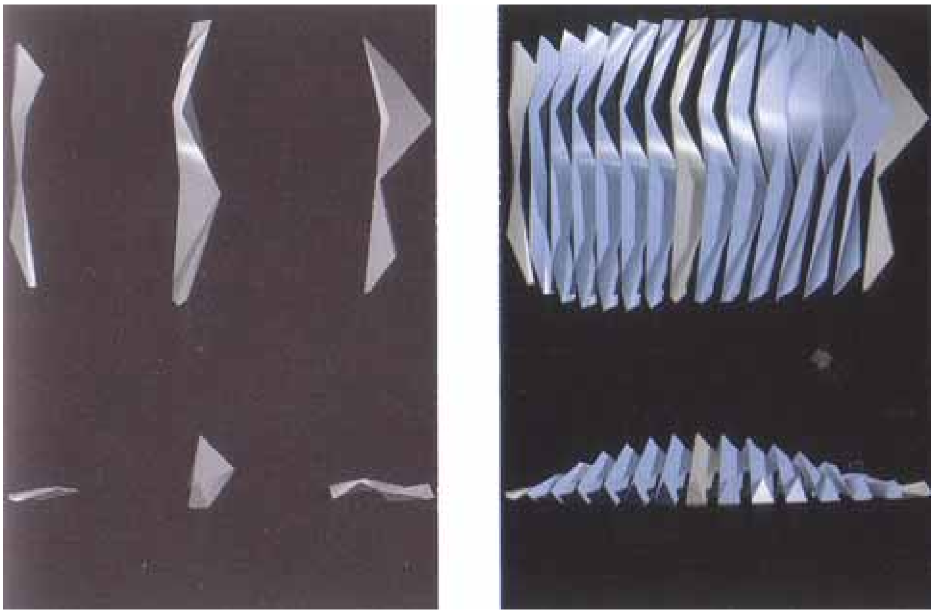


Figure 17 – Morphing – creating a form using transformation of 3 initial forms (De Luca, Nardini 2002)

Force-based design – This is a general name given to several methods that use rules or algorithms in an animated method. Two main directions of rule-based design used by architects were identified in this research; the first involves interaction between objects or forms with embedded "forces"[35] or interaction of objects with external "forces." In the first case for example, every object or form is assigned with a field of influence in which it applies a predefined or a programmed modifier. The interaction between the forms causes changes that reflect the physical and programmed relationships between them. One of the developers of this method is Greg Lynn, who discusses the use of forces and vectors as a generative force in architecture (Lynn, 1999). He compares this method to the design of a ship's hull where dynamic forces determine the final shape. He demonstrates the use of this method in

[35] "Forces" in this context refers to form deformation algorithms that can be assigned to an object or applied directly to an object.

architecture through the development of isomorphic polysurfaces (blobs) from spheres in the Artists Space Installation project (see Figure 18).

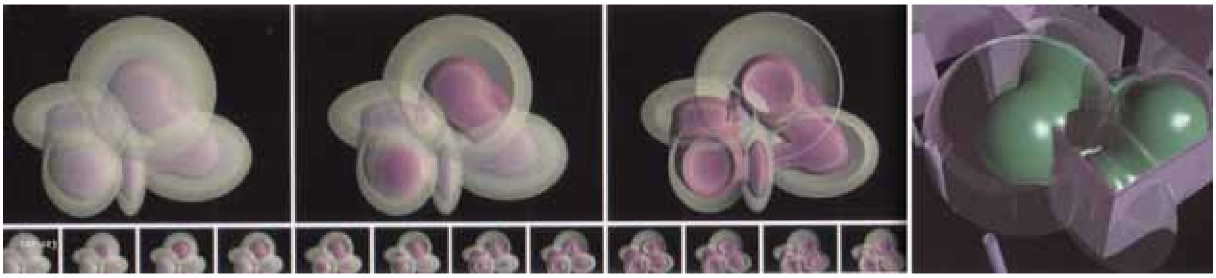
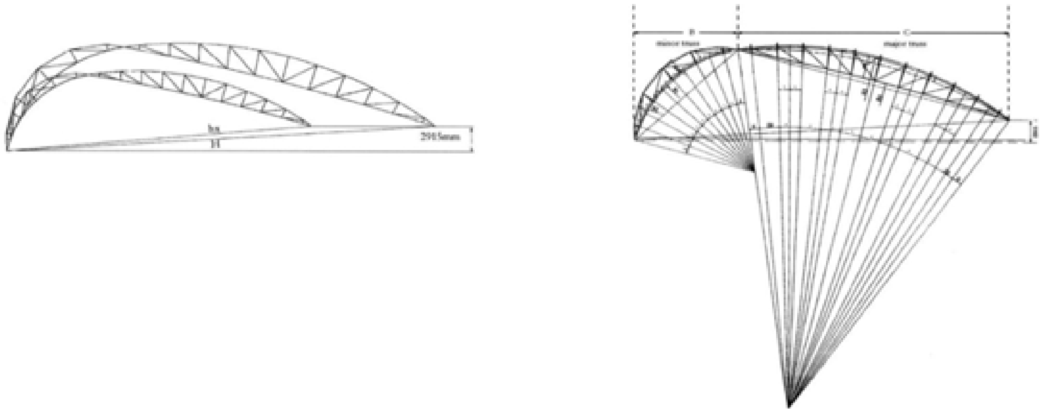


Figure 18 – Artists Space Installation – interaction between zones of influence of different objects defines the form (Lynn, 1999)

The second direction concerns the development of parametric relationships between objects in a 3-D model. These relationships represent programmatic or other demands. One of the earliest examples of this method is Grimshaw's Waterloo train station project in London, where a generative section was designed so that it could be modified to fit changing geometry and requirements (Figure 19)[36].



(a)

(b)

Figure 19 – Waterloo train station – rule-based design (a) – side view; (b) – realization (http://www.architectureweek.com/2001/0919/tools_1-1.html)

[36] See also the discussion on parametric design in section 5.5.

7.4 Closing remarks

This chapter provides an overview of design method theory and discusses the shift from process-oriented design to product-oriented design. It concentrates on the changes introduced by computers to architectural design methods. Within this domain, it argues that the increase in computer processing power and the introduction of new parametric-based design applications shift architectural design methods toward the nonlinear mode, in which it is possible to develop several design alternatives simultaneously. The chapter defines two types of computer-based design methods, static and animated, and presents directions and examples for each type.

In conjunction with the previous chapter, it establishes the theoretical basis for the attempt to define and develop new, nonlinear, animated design methods within the domain of computer-based form generation and optimization. The following chapter will concentrate on this domain, examining existing research and noting gaps and directions worth pursuing for this research's contribution to the body of knowledge.

8. Computer-based form generation and form optimization in architectural design

8.1 Introduction

The growing abundance of information and the increase in computer processing power are responsible for two parallel processes that pose a great challenge to architectural design. The first process is the emergence of dynamic simulation algorithms/software that are able to simulate real-life dynamic scenarios with increasing accuracy. These models are developed both for engineering/design purposes and for entertainment purposes (in the movie and computer game industries). The second process is the increasing connectivity between software and computer networks.

The ability to use computer simulation results as part of the design process, by both consultants and designers, is expected to have immense implications on the architectural discipline.

According to Gero (1996), typical computational models of design can be grouped under such processes as simulation, optimization, generation, decomposition, constraint satisfaction and, more generally, search and exploration. Some of these processes, however, have overlapping meanings in terms of the design process. Thus, decomposition can be a process performed within an optimization or generation process and constraint satisfaction can be a part of any of the other processes. A more precise grouping can differentiate two main modes of operation in terms of computer-based design: form generation (morphogenesis) and form optimization. Design optimization can be defined in this context as a form modification process to achieve performances defined by examined criteria (Kalay, 2004). It can be divided into three sub-processes: simulation, evaluation and modification. Simulation aims to imitate reality by creating mathematical models that dynamically represent the behavior of an object from a single performance point of view or the interaction between various performances. Evaluation can be defined as measuring differences between achieved and expected performances using predefined fitness criteria. Modification involves both developing a strategy concerning the changes that need to be performed in order to decrease these differences and the actual execution of these changes. From a linguistic point of view, the noun ‘generation’ has two main meanings: The first refers to the process of developing from an earlier type and the second refers to the process of coming or bringing into being[37]. In this context, the second meaning is more appropriate since the first meaning resembles the above definition of optimization. Nevertheless, it can be argued that most generative approaches do not stop after the initial form has been generated but continue to optimize it. Therefore, the difference between form generation and optimization is ambiguous. Since this uncertainty exists in the academic publications on the

[37] Merriam-Webster dictionary 2007 (http://www.m-w.com/)

subject, this research uses both terms (optimization and generation). For an approach to be defined as generative in this research, it must follow at least one of the following two rules: first, if it performs form generation according to the types of generation that are defined in the following section and second, if it is described as such by its developers in academic publications.

Consequently, the design process can be described as a process that starts with form generation; then, once the initial form is generated, it is developed or optimized further (through simulation, evaluation and modification processes) to adhere to prespecified fitness criteria.

The following chapter presents the latest developments in computer-based form generation, simulation, evaluation and optimization. It will also define gaps and directions worth pursuing in the application of these developments to the architectural design process.

8.2 Computer-based optimization – performance-based simulation in architectural design

It is already clear that computers and software are developing towards all-encompassing future of connectivity. Thus, by using standard protocols, it will be possible to import, export and work on information originally generated in various software programs that were developed for different types of users and disciplines. In architectural design, these developments initially were manifested in architects' use of software originally designed for other professions. The role of simulation in architectural design is gradually shifting from being controlled and used solely by professional consultants, generally at the end of the design process, toward being used throughout the different stages of the design process by all the professions within the building discipline. This can be explained by the gradual move to design in parametric software, the fact that parametric 3-D models are used both in simulation software and in some architectural software, and the ability to export and import files from different software.

The following section will discuss the use of computer-based simulation by architects, as well as provide an overview of simulation software and processes that are used by consultants in the architectural design and building process.

Computer simulation is already widely used by architects. Currently, the most common of the three main types of simulation used in architectural design is the representational. Realistic renderings and animations are made both for external presentation of the design to clients and for in-house design development purposes. In terms of performance in its wider sense, these renderings examine mainly qualitative aspects, such as esthetics and adherence to the site. Quantitative aspects in this type of simulation consist mainly of presenting light/shade conditions[38].

A second type of performance-based simulation used by architects employs physical forces modules in animation software, such as 3ds Max and Maya. Since these modules were originally designed for animated entertainment products, the focus here is more on the impression and less on the physical accuracy of the simulation. Therefore, these models cannot be used for accurate performance-based optimization and have been employed by architects mainly for creative form-based design.

The third type of simulation, which is the main interest of this section, consists of simulation software that were originally developed for other disciplines. Clarke (2001) divided the evolution of simulation tools into four generations:

1st generation – Handbook-oriented computer implementations aimed to provide the user with a general indication of some building performance criteria.

2nd generation – Developed in the mid-1970s, it introduced the dynamics of a building in an attempt to imitate the real physical conditions in a building. Early implementations were not applicable to the design process due to limitations in interfaces and computer processing power (Morbitzer, 2003).

3rd generation – The increase in processing power in the mid-1980s gave rise to new simulation software in which only time and space variables were independent. All the other parameters were dependent so that no single energy or mass transfer could be calculated in isolation (Morbitzer, 2003).

4th generation – Starting in the mid-1990s and still developing today, the fourth generation consists of changes in data modeling issues, user interface, application quality control and training. "In fourth-generation software, the built-in assumptions should be made explicit, they should undertake multi-variant analysis and they should be easy to use and interpret" (Morbitzer, 2003).

Recent years have seen the development of a large number of simulation tools and software programs. Indeed, the U.S. Department of Energy lists 335 different tools/software programs for simulation of energy performance alone[39], and this does not even include all the tools that emerged from the academic world, Based on the differentiation outlined in chapter 6, simulation tools can be categorized as following: sun shading and lighting, construction, wind, acoustics and energy. Since examining all the existing software is not possible within the scope of this research, it was decided to examine representative software/tools from each category, while focusing on their output and their suitability for use by architects.

---

[38] Light/shade in this context is regarded quantitative because the designer has information on the exact periods (day/month) when these conditions take place. A more advance application of light/shade simulation is discussed later in this section.

[39] See http://www.eere.energy.gov/buildings/tools_directory/ – February 2007.

8.2.1 Sun shading and lighting

The types of information required by architects during the design process can be summed up as follows:

a. Intensity and quality of light (daylight and artificial lighting) – includes local photometric calculation for all parts of the projects in various combinations of artificial and natural illumination conditions.

b. Intensity and geometry of shading – includes calculation of shading condition in various timeframes and calculations of shading coefficient factors.

The majority of commercial 3-D architectural design software offers some shading/lighting simulation modules. Some software offers physically photometric simulation modules for daylight, lighting and shading. The output of the calculations is a representation image of the light/shade condition. Numeric data concerning the level of illumination in the different areas of the examined model is not presented to the designer. Other software (such as AutoCad, Rhino and Revit) offers geometric calculations of shading conditions and less developed modules of lighting or only shading calculations, as in the case of SketchUp.

Nevertheless, special software for sunlight/shadow calculations is still used by academics and professional consultants. These software concentrate on a more complex level of shadow/light calculation to aid designers in the optimization of shadow/light conditions, mainly in cases when visual representation is not enough. Following are some examples of software that provides information not yet offered by commercial architectural software:

SUSTARC – generates solar rights envelope (SRE) and solar catch envelope (SCE) – the volume of possible solutions that consider either solar insolation or solar shading.

SRE – presents the maximum height of a building that does not violate the solar rights of any existing buildings during a given period of the year.

SCE – presents the lowest possible locus of windows and possible solar systems on the building under consideration so that they are not shaded by existing neighboring buildings during a given period of the year (generally winter). The user interface and example of an output is presented in Figure 64 (Capeluto and Shaviv, 1997).

SHADING – this application calculates numeric data on local shade conditions for specific open spaces, roofs, facades, windows or any other selected surfaces in a predefined time frame (year, month, hour of day). The user interface with an example of an output is presented in Figure 20 (Yezioro and Shaviv, 1994).

ECOTECT – Developed by Square One Company[40] , this application offers numeric data of illumination levels at any point (also in 3-D) in the architectural model. These data can be

[40] For information on Square One, see http://www.squ1.com/about

presented in various types of numerical grid, contoured in 2-D and 3-D lux and DF (Daylight Factors) images (see Figure 21).

RADIANCE - Suite of programs for the analysis and visualization of lighting in design developed by Greg Ward Larsonat Lawrence in Berkeley National Laboratory (1985 - 1997). It calculates values of spectral radiance (ie. luminance + color), irradiance (illuminance + color) and glare indices. Simulation results can be displayed as color images, numerical values and contour plots[41].

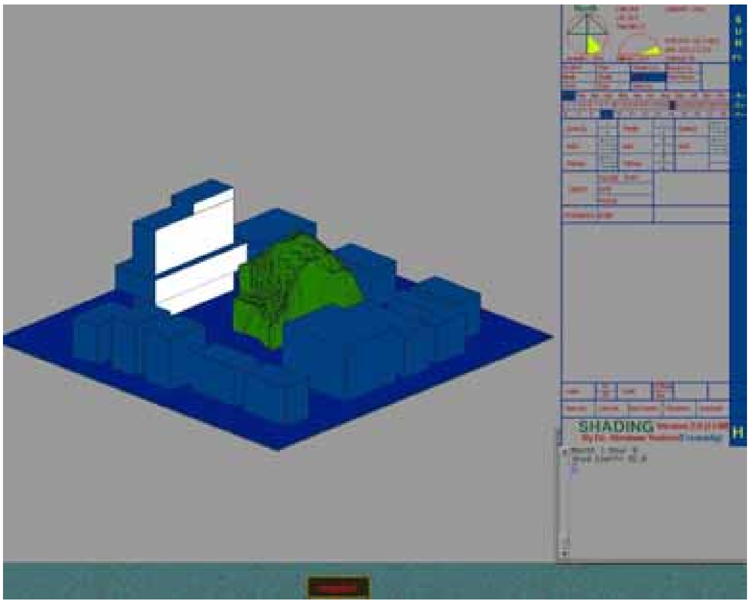


Figure 20 – Shading software interface

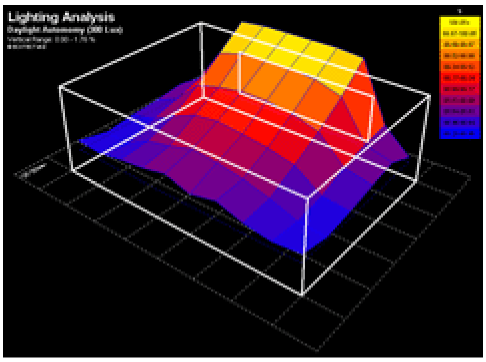


Figure 21 – Ecotect – 3-D numeric grid interior light simulation (http://www.squ1.com)

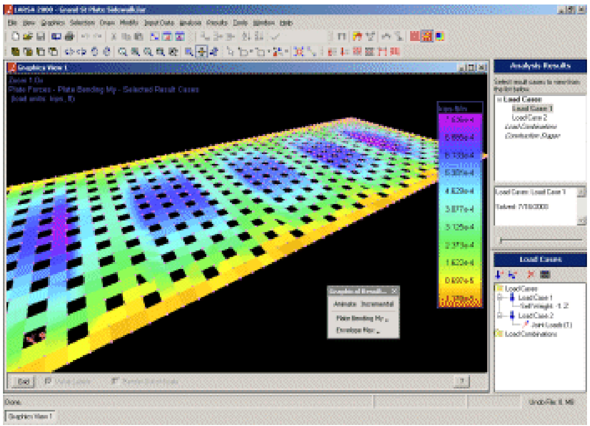
8.2.2 Structure – loads, stress simulation

Structural simulation software has been available for quite some time. In the last two decades, 3-D simulation software based on Finite Element calculation was developed. Broadly speaking, Finite Element calculations are based on the division of every element into

---

[41] http://radsite.lbl.gov/radiance/HOME.html

numerous smaller parts in which the load can be calculated for each part; the results are then interpolated to calculate the entire load. The Finite Element 3-D model facilitates examining the entire building model, taking into account reciprocal influences of different elements in terms of resistance to loads. Finite Element is employed to decrease the amount of material used to resist load forces and to help avoid mistakes in calculations (Oden, 1987). The main outputs of contemporary structural calculation/simulation software offer numeric and visual static and dynamic information of load/stress (see Figures 22-24).



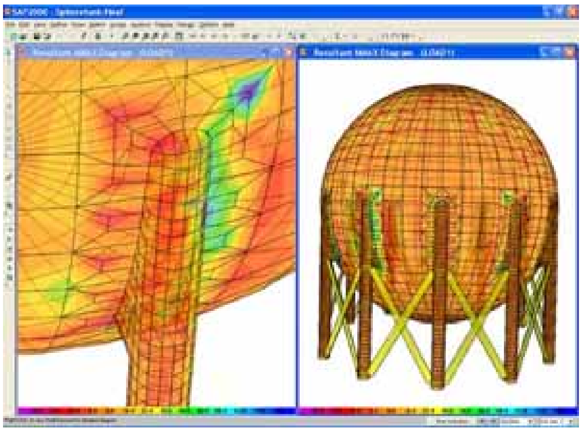
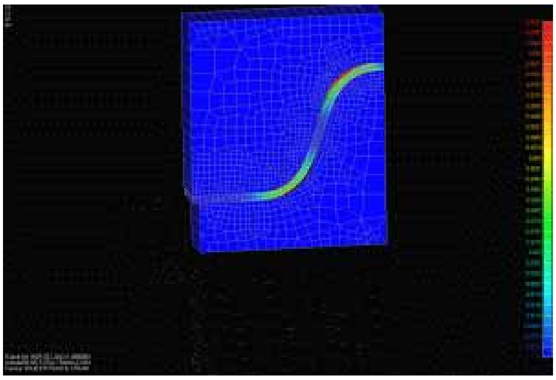


Figure 22 – On the left, LARSA software: Finite Element stress analysis in a surface model (http://www.larsa4d.com/). On the right, SAP software – color-coded stress simulation (http://www.csiberkeley.com/products_SAP.html)



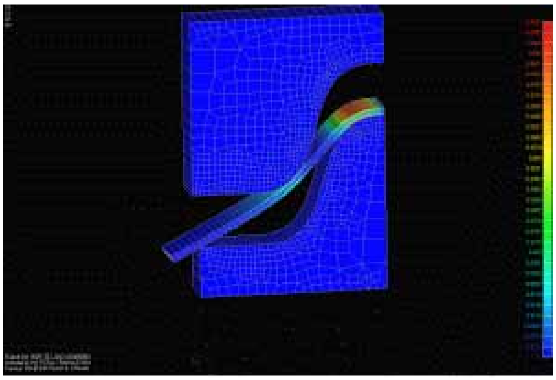


Figure 23 – NEi Nastran Finite Element Analysis software: Finite Element dynamic stress simulation (http://www.nenastran.com/newnoran/neiNastran.php)

As opposed to the sun light/shading simulation software described above, these structural simulation tools have not been widely employed by architects and they have not been embedded in architectural design software[42]. The form and the structure of the architectural design have been treated as separate processes, under the purview of separate disciplines.

[42] Software as Digital Product and Autodesk Revit that will be discussed later in this chapter include structural model, nonetheless, these model are not usually used by architects.

Several factors contribute to this situation:

a. The existing tools are very complex and most architects lack proper training in their use.

b. Since the architectural 3-D model cannot be directly used in Finite Element simulation software, it has to be rebuilt or modified, a very time-consuming activity that does not fit within the scope of normal design process.

c. Building laws usually demand that a structural engineer do the simulations for the architect.

Therefore, many architects tend to rely on the engineer's simulation results and are not motivated to invest time in learning and using these tools in their design process.

Figure 24 – On the left, Franken architects, Dynaform, tension analysis (Kolarevic, Malkawi 2005). On the right, the final result (www.franken-architekten.de)

The main motivations for embedding structural simulations in architectural software and design process are:

a. As buildings present increasingly intricate demands, in terms of both program and form, the ability to carry out in-house preliminary structural calculations could help architects avoid problematic design directions.

b. Since the use of computers in architectural discipline is moving toward 3-D parametric one-model design, the gap between design and structural simulation software is decreasing.

8.2.3 Wind

Wind generates three main types of considerations that must be addressed by architects during the design process. On an urban scale, wind regime must be considered in order to avoid places where wind does not allow people to stay or feel comfortable, or alternatively, to expose them to desirable winds so as to achieve natural ventilation. On a building scale, it has to be considered both for ventilation and for structural stability due to its generation of lateral forces. The simulation of the impact of wind on a building's structure is a dynamic load/stress simulation that is basically similar to the simulation processes discussed in the previous section. Computer wind-flow simulation tools that simulate wind on an urban scale have been developed and improved since the 1990s, yet this research did not find wind modules embedded in any architectural software to date. This can be explained by:

a. The existing wind simulation tools are complex and most architects lack the proper education to use them.

b. An architectural 3-D model cannot be directly used in wind simulation software that uses Computational Fluid Dynamics calculations (CFD). To be used in wind simulation software, the 3-D model has to be rebuilt or modified, a time-consuming task that cannot be executed within the scope of a normal design process.

c. The two main reasons to perform wind simulation in architectural design are to examine the influence of wind in terms of the building's structure and to examine its influence in terms of comfort (for pedestrians and people that open windows or go out to balconies) and ventilation. Structure-related wind simulation is mainly demanded by the planning authorities for the design of special projects such as high-rise building, bridges and so on. In these cases, as in the previous section concerning structure, special professional engineers are responsible for simulation. In terms of comfort and ventilation, wind simulation is not performed for most buildings; it is reserved for high- profile, very costly buildings such as high-rises or mega structures, or when an environmental assessment is required by authorities.

Consequently, similar to what was described regarding structure simulation, architects tend to rely on the engineer's simulation results, when performed, and are not motivated to invest time in learning and using these tools themselves when such a simulation is not performed.

The main motivations for embedding wind simulations in architectural software and in the design process can be summed up as follows:

a. In terms of structural wind simulation, as buildings present increasingly intricate demands, both in program and in form, the ability to perform in-house preliminary structural wind calculations could help architects avoid problematic design directions.

b. In terms of ventilation, as the world's energy consumption rises and its energy supply decreases, reducing the need for artificial cooling/heating becomes increasingly important.

c. In terms of comfort, embedding wind simulation to architectural design can increase usable space and fine-tune spaces to improve human comfort, thus also saving energy.

d. Since the use of computers in architecture is moving toward a 3-D parametric one-model design, the gap between design and wind simulation software is decreasing.

Figure 25 presents examples of the results of structure-related computer wind simulations done by structural consultants. Figure 26 shows examples of comfort-oriented computer dynamic simulation of wind velocities in an urban context.

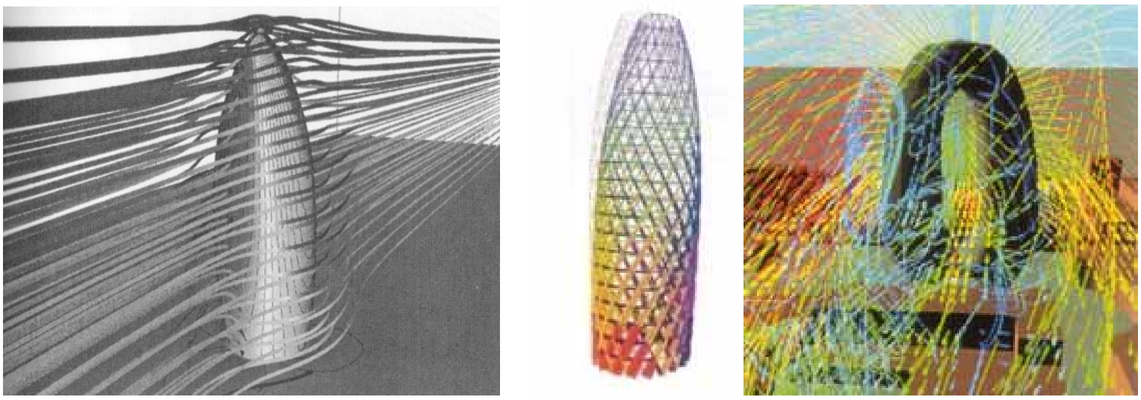


Figure 25 – On the left, Foster + Partners – wind force optimization for construction design (Szalapaj, 2000). In the middle, Finite Element optimization of lateral wind forces (Kolarevic, Malkawi 2005). On the right, Future systems Project Zed, ARUP CFD wind simulation (Kolarevic, Malkawi 2005)

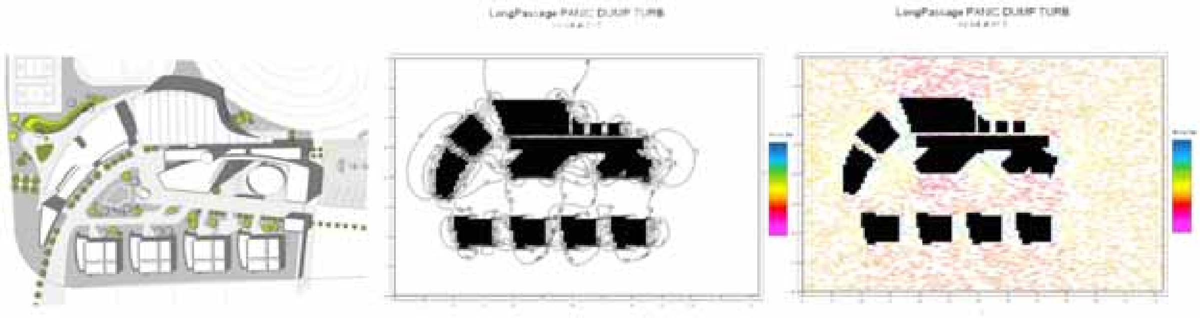


Figure 26 – Wind simulation using ENVI-met software (http://www.envi-met.com/). Images are taken from winning entry to Hefer High School competition (2004) designed by Grobman Architects in collaboration with Nir Chen Architects and architect Tami Lapidot. Climatic consultants: Guedi Capeluto and Abraham Yezioro. Wind simulations were conducted to optimize comfort conditions in the schoolyards.

8.2.4 Acoustics

Similar to wind simulations, acoustic simulations can be performed in two main scales, urban (exterior) and interior. On the urban scale, acoustics can influence decisions about the arrangement of buildings on a site and the need to design acoustic barriers, among other things. On the building scale, acoustic simulations are important in a wide variety of architectural designs including music halls, conference rooms, open-space working environments and many other situations. Acoustic simulation influences the project designers' decisions regarding geometry as well as the choice of materials. However, this research found no acoustic modules embedded in any architectural software to date. This can be explained by:

a. The existing acoustic simulation tools are complex and most architects lack the proper training to use them.

b. An architectural 3-D model cannot be directly used in acoustic simulation software that uses acoustic simulations. To be used in acoustic simulation software, the 3-D model has to

be rebuilt or modified, a very time-consuming task that cannot be executed within the scope of a normal design process.
c. In many cases when acoustic simulation is needed, a special consultant is appointed to perform the acoustic simulations. Thus, many architects rely upon the consultants' simulation results and are not motivated to invest time in learning to use these tools themselves.

The main motivations for embedding acoustic simulations in architectural software and design process are:
a. Urban scale – This type of simulation helps urban planners make decisions relating to, among other things, future land use and the location of new neighborhoods and new roads. It can also help define material codes for new developments and help decrease noise in existing urban environments (see Figure 27).

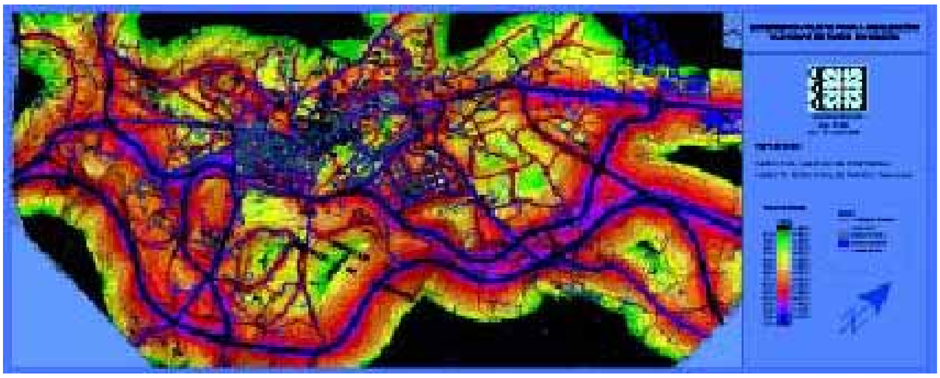


Figure 27 – Urban noise simulation using SoundPlan (http://www.soundplan.com/)

b. Small urban scale (neighborhood scale) – Analysis of noise from main roads can influence the layout of new developments as in the example presented in Figure 28, where the school's main administration building was used to block noise from reaching the school's inner courtyards and classrooms.
Figure 29 shows the distribution and levels of noise emitted from buses on an urban street. Having this type of simulation while designing an urban building can help the designer to consider building geometry that decreases noise levels inside the building while avoiding spaces with loud noise levels (caused by an interfering effect).
c. Building scale (interior design) – As designs become increasingly complex in terms of program and form, architects' use of acoustic simulation in the early stages of design can help to avoid both design mistakes (in geometry and materials choice) and problematic design directions. It also can help designers improve the final design's acoustic performance by using the acoustic consultant for fine-tuning the results rather than working on the initial solution.

Figure 28 – Urban noise analysis using SoundPlan (http://www.soundplan.com/). On the left, Acoustic simulation done by M.G. Acoustic for an entry to the Yokneam High school competition. On the right, entry to the Yokneam High school competition by Grobman Architects in collaboration with Lapidot Architects

The following images (Figure 30-31) show interior sound simulations in halls that can guide designer to modify the design for better acoustic performance. In the basketball stadium (Figure 30 on the left), the simulation was used to design and position deflectors and noise-desorbing elements in order to reduce disturbing noise levels. In the other halls (Figure 30 on the right, and Figure 31), the acoustic simulation was used to define the space's geometry and materials.

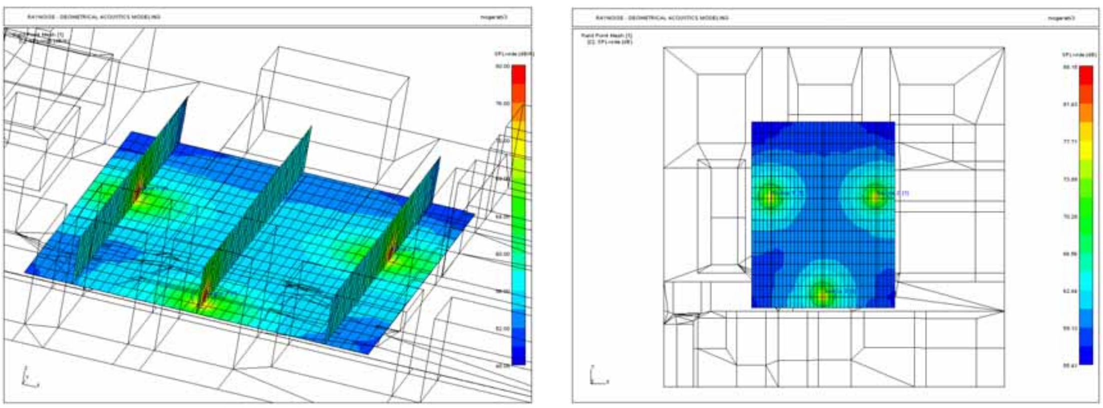


Figure 29 – Acoustic simulation using RayNoise[43]

[43] Acoustic simulation of noise emitted from buses driving on the streets that surround the site of the case study presented in chapter 10.6. The acoustic simulation was for this research by Livni Acoustic Consultants. Information on RayNoise can be found at http://www.lmskorea.com/home/products_cae_raynoise.php.

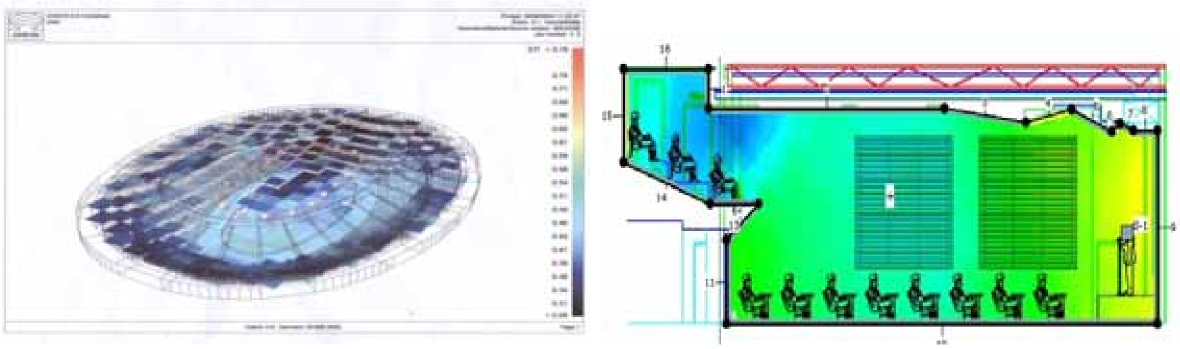


Figure 30 – On the left, acoustic color-coded simulation of noise in a basketball stadium[44] by Odeon (http://www.dat.dtu.dk/~odeon). On the right, acoustic color-coded simulation results for a hall by Soft DB (http://www.softdb.com)

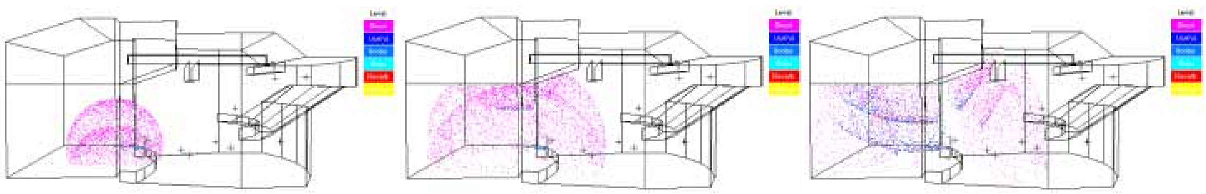


Figure 31 – Acoustic dynamic simulation of noise in a concert hall – Ecotect (http://www.squ1.com)

8.3 Integrated simulation software

Parallel to the ongoing advances in contemporary fourth-generation simulation software, an emerging new direction in simulation software places the emphasis on the integration of simulation tools. Oriented toward a wide range of disciplines, the integrated approach aims to use a one-model database for design, engineering, simulation, fabrication, project management and on-site construction, thus extending the previously mentioned notion of "one-model building" to include simulation an integral part of design.

The prospects and possibilities of this approach were discussed in two doctoral research dissertations (Bleiberg, 2003; Morbitzer, 2003) that examined and compared simulation tools. Both envisioned integrative interdisciplinary software as a vital part of the design process. Nevertheless, both concluded that simulation tools do not fully address the needs of building designers in terms of simplicity of use and adjustability of the simulation's accuracy and complexity to suit the various stages of the design process. Although some integrative tools were examined in these two dissertations, none was commercially sold and used extensively by collaborative design teams.

Since then, several architecture-oriented academic and commercial applications have been developed. Three different types of software were found:

---

[44] Image: Nokia basketball stadium by Lerman Architects. Acoustic simulation by M.G. Acoustics

a. Integrated remote simulation platform – This is represented by Semper II (a more advanced form of Semper) software, which is being developed jointly by the National University of Singapore (NUS), Carnegie Mellon University (CMU) and Temasek Polytechnic (TP, Singapore). Semper is an object-oriented design tool for integrated building performance simulation. It aims to allow users to access the SII system regardless of hardware, operating system or location on a network. Geographically distributed users can generate and edit building models via a platform-independent user interface. These building models can then be made subject to concurrent analysis by multiple simulation applications running on remote servers (Lam et. al. 2002). Although it is an intriguing idea, the SII is still under development and has not yet been fully tested, so it is difficult to determine at this point whether it can really fulfill its potential.
b. Integrated software for modeling simulation – This type is represented by Digital Project by Gehry Technologies, which is based on Catia, engineering design software that was developed for the aviation industry. Its adaptation to architectural design was initiated by Frank Gehry's architectural firm and based on its architectural design experience on several projects (http://www.gehrytechnologies.com). Digital Project is designed to perform as a platform that incorporates all aspects of the design process including modeling and simulation/analysis.
c. Integrated software for simulation – This integrated building performance assessment software aims to operate parallel to design software as an integrated simulation tool. Examples are Virtual Environment by Integrated Environmental Solutions (IES) (http://www.iesve.com) and DesignBuilder (DBS) (http://www.designbuilder.co.uk). These programs offers environmental simulation tools (thermal, solar, light and CFD), mechanical and electrical system design and evaluation tools, evacuation, code and cost assessment. The amount of information available on these software programs does not yet allow for an accurate assessment their performance and contribution to the architectural design process. Nevertheless, experience in integrated design processes in other disciplines, such as the aviation and naval industries seems to indicate that a design-oriented approach that incorporates modeling and simulation capabilities is more likely to suit the needs of architectural design.

8.4 Computer-based form optimization – evaluation and modification

For the most part, traditional architectural design methods can be regarded as form optimization[45]. Thus, architects have always modified/optimized initial design/form according to various demands/fitness criteria that had to be negotiated into a single design proposal.

[45] The definition of form optimization in this research follows that suggested by numerous researchers, including Schmitt (1992), Kolarevic (2003 b) and others, and resembles Kalay's notion of evaluation (Kalay, 2004, p. 302). There is, however, difference between Schmitt and the other researchers (including this research's definition to optimization) in term of the place of

Before discussing the implications and possibilities of computer-based optimization, it must be noted that the design optimization process in general has several preliminary difficulties:

a. Complexity of the problem[46], subjectivity of the solution – The first difficulty has to do with the fact that design involves complex and contradictory programmatic demands in terms of the form it suggests, its performance or its cost (Jockusch, 1992). This contradiction forces designers to define priorities for the level of adherence to each of the demands. Since the decision on priorities is made by the designer, it is clear that design solutions are subjective by nature, biased by the designer's preferences. Thus, the design solution is subjective and optimal in the sense that it is the least bad solution (Rittel and Webber, 1969). Moreover, it can be achieved only after the subjective decision regarding the relationship between the various criteria has been established by the designer.

b. Redundancy in architectural design – Weinstock (2005, 2006) compared computer-based design to natural morphogenesis and argued that since function and usability in design changes over time, all designs have to include redundancies. He argued that form optimization in architecture is time-dependent and that "It is necessary to think of the geometry of a biological or computational form not only as the description of the fully developed form, but also as the set of boundary constraints that act as a local organizing principle for self-organization during morphogenesis." Therefore, the second preliminary difficulty in design optimization is the temporary nature of the fitness criteria definitions that have to do with the optimization process in architecture.

c. Quantitative evaluation of qualitative problems – computer-based optimization is quantitative by nature. In order to optimize qualitative demands it has to be translated to quantitative scale. According to Chayutin (1975), this should be done by rationalizing the problem through economic analysis (scaling qualitative traits according to economic value). Another possibility is to use statistical data in such a way that the value/importance of a trait would be determined statistically by comparing results from numerous individuals (professional/experts or other individuals). For example, the amount of comfort in a given type of space would be determined by a statistical average of comfort values of many individuals on a percentage numeric scale.

Single-parameter computer optimization is a straightforward process in which the result of a simulation process is evaluated using three types of fitness criteria: threshold, range and discrete values (Schmitt, 1992). Then, a strategy/algorithm is used to modify the initial form in order to improve the results. This process stops when the needed criteria are achieved or when time/cost limits are reached.

---

optimization in the design process.Schmitt argues that evaluation is performed only at the end of the design process on the final design while other researchers do not limit the use of optimization to a certain stage of the design.

[46]See also the discussion on ill-defined/wicked architectural problems in chapter 7

Chayutin (1975) offers the following procedure for this type of design process:

a. Determining initial assumptions for the simplification of the problem

b. Choosing the dependent variable for which the optimization is needed (target variable)

c. Determining the independent variables that influence the target variable

d. Determining the dependencies between the dependent variable (target variable) and the independent variables

e. Determining the constraint/relationships between the independent variables – determining the problem limits

f. Choosing the optimization technique

g. Choosing a technique for result analysis and sensitivities analysis

h. Running the model, result analysis and sensitivities analysis

i. Re-addressing the initial assumptions and realization of the model in order to adjust it to real-life conditions. Continuing to build the model in an interactive process until the designer is satisfied with the result.

Multi-criteria optimization is highly intricate and poses great challenges to computer-based design. Linear approaches to multi-criteria optimization consider this approach to be an agglomeration of single-criteria procedures; thus, the order of the singular optimization (evaluation and modification) processes has to be determined according to the priority of the fitness criteria. Changing the order will change the final result. Fenves et al. (1992) combined linear and nonlinear processes by performing linear evaluation of single criteria, then modifying the architectural form according to this evaluation and then finally validating the solution by examining its adherence to all the other criteria.

Several nonlinear optimization approaches have been developed. In the Wiezel and Becker approach (1992), the results of all fitness criteria (from a simulation process) are negotiated to a single grade (or set of grades). The negotiation algorithm is defined by the designer, which makes the solution highly subjective. However, changing the algorithm parameters (ratios between the various criteria) can offer flexibility in terms of adjusting to the demands of different projects. A similar approach was further developed by Choudhary et al. (2003) and Gololov and Yezioro (2007).

Following the evaluation stage, in the second phase of the optimization process (the modification stage), a strategy/algorithm has to be developed for changes in the initial form parameters that will improve the grades. The more criteria (target variables) and dependent variables, the more possible changes have to be performed and evaluated. This process can be performed linearly for a single change each time or produce solution groups that have to be revaluated in order to find the single solution needed for continuing in the design process. The leading approaches within this premise are:

8.4.1 Complete enumeration

This approach is based on the premise that computers can generate all the possible alternatives to the modification of an examined dependent variable. Despite the existence of several academic software programs (see examples presented by Kalay [2004]), this research located no commercial software that uses enumeration.

8.4.2 Space allocation problems

Space allocation[47] aims to lay out spaces and activities in a building according to rational principles (Kalay, 2004). While space allocation approaches for design were developed mainly in 1960-1980[48], some research is still being done in this field (an example is the work by Michalek et al. 2002). Many of the earlier approaches concentrated on generating building plans from programmatic information. Kalay argues that space allocation is suitable for projects where circulation plays an important role and divides space allocation approaches into additive, permutational and constraint satisfaction (Kalay, 2004). The additive approach places the highly connected spaces first and then situates the other spaces around them. The permutational approach introduces the possibilities of swapping between spaces if the result is more satisfactory, as well as starting the allocation process randomly and then evaluating the results (Shaviv and Gali, 1974). Constraint satisfaction refers to attempts to include additional design criteria (besides circulation) in the decision-making process by adding constraints to the placement algorithms. According to Kalay, this is done “for buildings where distances are of lesser importance than other criteria (e.g., privacy in a house), and where additional criteria (e.g., lighting and site condition) influence as much as or more than circulation.” Various approaches to space allocation borrowed metaphors from other fields such as electricity (March and Steadman, 1974) and mechanics (Arvin et al, 2002) to define the algorithms. This research located no commercial software that uses space allocation and no attempts at space allocation 3-D approach were found, even in academic research.

8.4.3 Case-based reasoning/Cased-based design/Expert systems

A popular approach in CAD in the 1990s, case-based reasoning refers to an optimization method in which design decisions are guided by a single distinctive prior case (precedent, prototype, exemplar, or episode). Case-based reasoning seeks to determine a "source case" relevant to a given design problem or source case rules in expert systems. The process is

---

[47] Space allocation is also known as “automated floor plan generation,” “automated spatial synthesis” or “quadratic assignment formulation” (Kalay, 2004).

[48] A survey on space allocation algorithms was done by Simpson (1980).

thus separated into two parts: first, finding the appropriate source case and second, determining the appropriate parameters that need modification for the given design problem and developing an algorithm to perform these changes. Oxman (1992), Heylighen, Neuckermans (2001) and Kalay (2004) present many approaches that were developed over the years in academic research[49]. Nevertheless, Heylighen and Neuckermans (2001) argued that no convincing breakthroughs have yet been made. Moreover, none of the developed approaches seems to have deeply influenced architectural practice.

8.4.4 Evolutionary methods

Computer-based evolutionary methods are based on the idea of seeking the best solutions (using defined fitness criteria) from a population of solutions (phenotype) based on different genetic code (genotype) (Kalay, 2004).

Bentley (1999) defines four characteristics of the evolutionary process: reproduction, inheritance, variation and selection. Computer-based evolutionary programs that are based on algorithms also require initialization, evaluation and termination, according to Bentley. Evolutionary search algorithms are inspired by and based upon evolution in nature – evolving solutions to problems; instead of one solution at a time, these algorithms consider a large collection or population of solutions. Frazer (1995) posits that to achieve the evolutionary model, it is necessary to define a genetic code-script, set rules for the development of the code, map the code to a virtual model, determine the nature of the environment for the model's development and, most importantly, designate the criteria for selection. Bentley (1999) argued that before applying an evolutionary algorithm, we must define the boundaries of the solution space (specify the phenotype), define the search space (genotype), find the algorithm[50] most suitable to the problem and define fitness function. He also suggests that the computer does not evolve anything; it is currently impossible to program in evolution, since we do not fully understand how evolution works. Instead, computers are instructed to maintain population of solutions, allow better solutions to "propagate" and allow worse solutions to "die."

This approach has garnered increasing interest in the last decade. As opposed to earlier approaches that were developed and tested almost solely by scholars, this time new approaches that use evolutionary methods were developed and tested by designers as well. This change can be attributed to the ubiquity of computers, the increasing familiarity of

---

[49] Both Oxman (1988) and Heylighen (2000) submitted doctoral dissertations on expert systems/case-based reasoning. Both are sources for a deeper understanding of this domain.

[50] Bentley (1999) defines four main types of evolutionary algorithms: a) genetic algorithms – created by John Holland in the 1970s and made famous by David Goldberg in the late 1980s; b) evolutionary programming – created by Lawrence Fogel in the 1960s and developed by his son David Fogel; c) evolution strategies – created by Ingo Rechenberg in the 1970s and promoted by Thomas Back; and d) genetic programming – created by John Koza in the beginning of the 1990s.

designers with computers and the enhanced connectivity between software and computers. Nevertheless, for the most part, the designers' research that uses evolutionary methods involves form generation; therefore, it will be discussed in the next section (8.5). Although it seems that the focus of research into evolutionary optimization methods in architecture has shifted in recent years toward improving its applicability in practical design problems, so far this study has not found evolutionary methods embedded in any commercial design application. Nevertheless, some tools have been developed within this realm in academic research, an example is a tool developed by Malkawi et al. (2003) that combines morphing-based GA, CFD evaluation with user inputs for the optimization of the relationships between the measurement of a single room space, proportions and position of a window in that space and the specifications of the air conditioning supply in that space. The aim was to find the optimal window proportions based on the proportions and position of the air-conditioning supply. What is interesting about this approach is that the designer is given a chance to add his preferences, which are based on qualitative criteria, to the quantitative performance optimization.

8.5 Computer-based form generation (Morphogenesis)

While architects in previous decades have been preoccupied with examining the possibilities embedded in 3-D design software (complex form representation and modification), it seems that the last decade has seen a growing interest in developing these software programs to a higher level of involvement in the design process. Computer-based form generation by script/code writing as a design method has been examined since 2001 by leading master's degree programs including the AA DRL, Columbia University and Harvard's GSD. Prominent large practices such as Gehry and Partners, Foster + Partners and others employ people whose job is to write code that would improve and adopt the existing software to meet their needs not only in terms of the work flow and production, but also as part of the initial design stages. This shift in focus has been accompanied by a growing number of publications both in academic and practice-oriented magazines (Architectural Design magazine alone has published since 2003 three special issues dealing with computer-based form generation and code in architecture[51]). Nevertheless, it seems that so far the implications of computer-based form generation on the design process have not yet been fully examined. The following section will present the research that has been done in this field and will develop an argument regarding the possibilities and limitations of computer-based form generation and its influence on the design process.

---

[51] Architectural Design is one of the leading commercial design magazines – see http://www3.interscience.wiley.com/cgi-bin/jhome/109924136 for more information

8.5.1 Computer-based form generation and the architectural design process

Architects do not follow a single prescription in generating design initial form. Leeuwen et al. (2001) suggest that design can start with the following actions:
a. Organizing physical objects
b. Organizing abstract spaces with a possible materialization later (e.g., using a layout program, in which user activities are assigned to abstract spaces)
c. Planning user activities and afterward connecting them to specific spaces
d. Exploring shape without assigning a semantic meaning to it at the start

The introduction of computers to architectural design ushered in the possibility of using computers to generate architectural form. While form optimization acts on an initial form in an "after-the-fact" manner, in form generation the computer creates form from information (Kolarevic, 2003, Kolarevic, 2003 b).
Based on this definition, it can be assumed that form generation in architecture takes place only in the beginning of the design process while the rest of the process could be regarded as an optimization process (based on the definition of optimization in the previous chapter). This would probably be the case for a design process that deals with the design of one element as in some cases of industrial design, or the design of fairly simple structures that, for example, have no internal division to secondary spaces. Nevertheless, when it comes to the design of buildings, it is clear that several levels of form generation might take place within the design process, as follows:
a. Generation of initial form/envelope
b. Generation of secondary spaces/division to floors and sub-spaces
c. Generation of building elements (windows, doors, etc)
d. Generation of building details

The above four levels can be compressed into two main types: The first and more obvious has to do with the introduction of new forms, while the second has to do with changing the topology of the existing forms. Changing topology could be done by formal modifications such as adding openings[52].

8.5.2 From information to form

Kalay (2004) questions the direct connection between form and function as suggested in Sullivan's famous slogan "form follows function." He claims that the solution space for the

---

[52] From a topological point of view, there is no difference between a sphere and a box. One way to create a topological difference is to add hole to one of the forms.

“appropriate design solution” lay in the intersection between form, function and context. Using this division as a source of information to generate design alternatives, one can suggest that these three “building blocks” represent programmatic information: Form/space defines the specification of spaces and the relationship between these spaces. Function defines the expected performance from these spaces and context defines the relationship with the urban fabric and environment. However, the above division does not cover the entire scope of information that could be used to generate form. Another important layer of information derives from the designer. This layer consists of perceptual and cultural premises that are drawn from the designer’s knowledge/experience and intuition. Therefore, only a combination of all these layers can produce a complete design.

Earlier, this research delineated two general types of information used to generate form (see section 6.5): first, information that derives from performance-related criteria and second, information derived from formal aspects. The difference between using a computer to generate form and traditional form generation (without using computers) can be examined in relation to these types of information. In terms of form-based generation, computers allow designers to reach a higher level of formal complexity in a much shorter time, thus facilitating the examination of many design alternatives. Other more conceptual differences are based on “losing control” over the design and bypassing the limits imposed upon design creativity by the “human eye”; according to Eisenman, this can increase creativity and produce a new formal world unlimited by human perception (Eisenman, 1992).

Performance-based form generation, on the other hand, creates a "smart form" that embeds performance-related quantitative information, providing another layer of information parallel to the information on the form’s geometry (Capeluto, 2003b).

In the early ‘60s, the computer was considered to be an intelligent problem-solving machine that would eventually match, and perhaps even supersede, human intelligence. Researchers have developed a plethora of models and theories to automate the design process and optimize its product (Andia, 1997). Some believed that an entire “optimal” project could be computer-generated by translating the designer’s creative work and functional/performance-based criteria into quantitative information. This idea rested on the assumption that because the computer had superior intelligence, it would be able to generate the “optimal” design. In the previous section, we argued that the notion of “optimal” in computer-based optimization is problematic[53]. A similar line of argument explains why generating a complete “optimal” building is not possible:

a. One cannot define finite performance criteria for an architectural project’s final form because it is too complex and involves too many criteria (Venturi, 1977. Negroponte, 1970. Schmitt, 1992. Koutamanis, 2000. Kolarevic, 2004).

[53] See the discussion on the limits of optimization in section 8.1.

b. Performance criteria can contradict one another. This forces the designer to rank (using formal or calculated criteria) each performance within a hierarchy of influence. By not fully adhering to any criteria, the design solution stops being optimal and becomes subjective, which aligns it with traditional design.
c. Final building design consists of performance criteria that derive from perception/feeling, that is, cultural grounds. As of now, there is no known empirical way to translate these criteria into computer algorithms[54].

Nevertheless, it can be argued that computer form generation has significant advantages over traditional design when it comes to a small number of criteria and smaller well-defined problems such as the generation of a building's envelope/initial form or the generation of elements in the building. The following section will present the research that has taken place since the 1970s based on this premise.

8.5.3 Computer-based generative design methods/approaches developed in architectural research and practice

Computer-based form generation developed in a nonlinear manner along several parallel lines. In its early days, computer-aided design was expected by some scholars to replace conventional design by generating buildings from data. Thus, some generative systems aimed at complete spatial design that used the computer's processing power to overcome the designer's information-processing limitations and produce a final design, mainly plans. Less comprehensive approaches exploited the advantages of computer-aided design in helping the designer to lay out building plans, mainly for the type of building in which empirical criteria, such as the distances between programmatic functions and frequency of trip, could be defined (Frew, 1980). Other approaches argue that generation of form by computers is limited to narrow parts of the design process, mainly initial or conceptual design, geometry problems or other performance-related problems (Bentley, 1999). The following sections present and categorize the main existing approaches to computer-based form generation in academic research and practice:

8.5.3.1 Cellular automata

One of the earliest types of computer-form generation tools can be traced to Cellular Automata (CA). Based on von Neumann's theories from the 1940s about computed-based, self-replicating forms, cellular automata began mainly as 2-D growth-simulating

[54] See also the discussion by Woodbury & Burrow (2006) and Terzidis (2003) on the advantages and limitations of computers in design.

algorithms/software. One of the best-known examples of CA is Conway's Game of Life, developed in the 1970s, which allows the growth of 2-D patterns with three basic rules. Another well-known illustration is L-systems, an algorithm developed by the biologist and botanist Aristid Lindenmayer in the late '60s as a way to model the growth of plants by using a set of rules, constants and modifying parameters and by using different starting points (Rocker, 2006).
Over the last two decades, CA has been adapted to architecture in various ways; one of the more promising avenues is the use of CA to simulate and predict the growth of cities, as seen in the work being done at University College London Centre for Advanced Spatial Analysis (CASA)[55]. Another avenue is based on the use of CA to develop complex formal expressions as suggested by architects such as Karl Chu (2006) (see Figure 32) and Studio Rocker (Rocker, 2006).

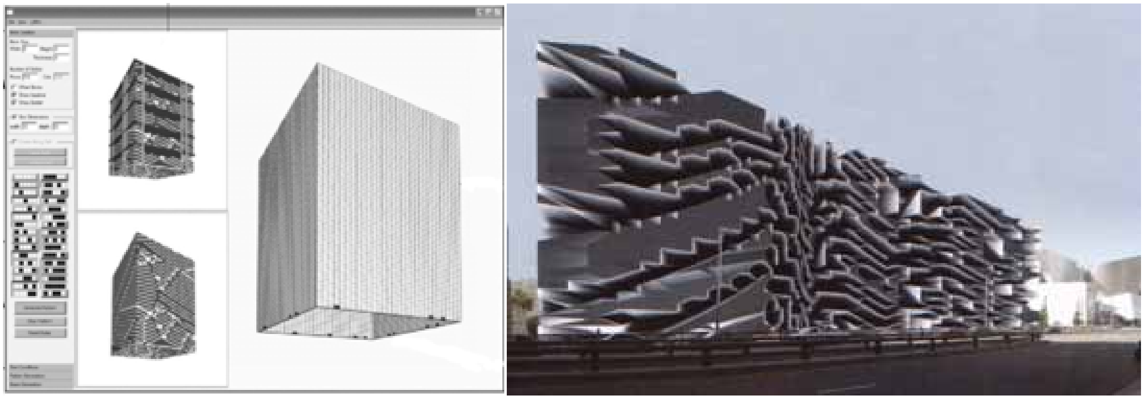


Figure 32 – On the left, building façade generation using CA (Silver, 2006). On the right, Proto-building – CA-generated complex façade pattern by Chu (Leach, Xu Wei Gue, 2004)

This work is interesting in terms of its potential to create highly complex conceptual forms that could be used for inspiration in the architectural form-finding process. Silver (2006) used CA-based algorithms to generate building façade patterns (see Figure 32). This approach was based on the understanding that computer power and simple rules can be used to generate a large number of complex formal patterns for the architect to evaluate.

8.5.3.2 Automated floor plan generation/Space allocation

Many of the earlier approaches concentrated on generating building plans from programmatic information. Automated floor plan generation and space allocation principles are described in section 8.4.2. Most of these approaches concentrated on the generation of 2-D plans. Although some of them suggested 2.5-D (extruded planes), this research found

[55] See the working papers published by Michael Batty, Paul Torrens and other members of CASA in http://www.casa.ucl.ac.uk

no 3-D based approaches[56]. While the approaches mentioned above and others produced prototype tools that demonstrated the ability to generate floor plans, this research found they had no significant influence on architectural practice, nor did it find any commercial application for the generation of building plans. This can be explained as follows:

1. Complexity of the problem – Space allocation can negotiate only small parts of the programmatic demands that are taken in consideration by architects in the generation of a plan.
2. Singularity of each building – Because every building program is unique, new definitions of rules are necessary for each project.
3. Three-dimensionality – To date, no way has been found to solve 3-D space allocation problems.
4. Cost – Defining space allocation parameters requires professional knowledge and training that is outside the scope of an average architectural practice.

8.5.3.3 Case-based reasoning/Expert systems

These approaches suggest using precedent cases (or precedent cases rules in expert systems) to optimize and generate similar building forms – see section 8.4 for a general description of these approaches.

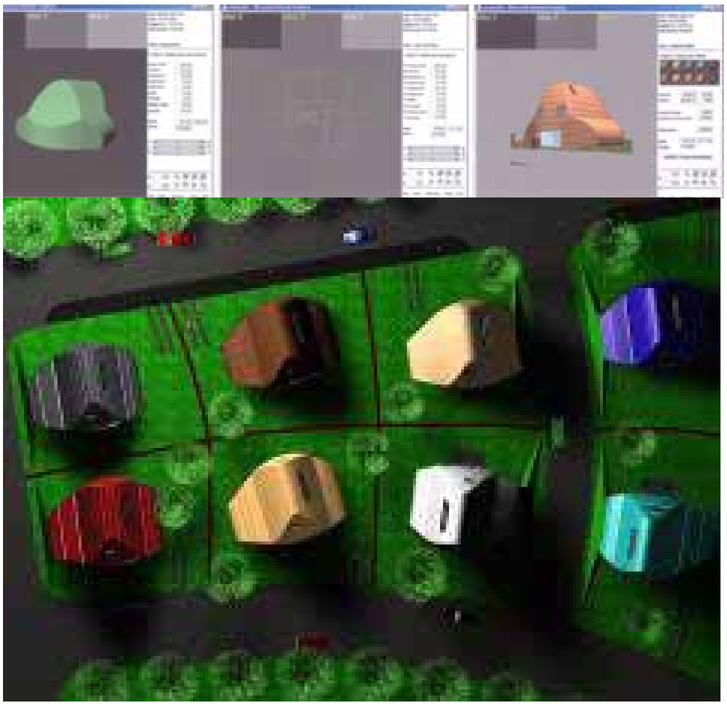


Figure 33 – Oosterhuis – Variomatics SM (Oosterhuis, 2002)

[56] 3-D based approaches should include the possibility of defining different heights for different programmatic demands, which by definition will generate complex sections of buildings, as opposed to the 2.5-D approach in which the generated building consists of flat floors.

In terms of form generation, in addition to the approaches identified by Oxman (1988), Heylighen and Neuckermans (2000) and Kalay (2004), this research identified another type of approach based on the idea that it is possible to define a parametric building template for certain building types. The template then can be used to generate different buildings from the similar type. One example of such a tool is Variomatics, developed by Oosterhuis (2002) – see Figure 33.

8.5.3.4 Shape grammar and formal rule-based form generation

Shape grammar theory was first presented by George Stiny and James Grip in the early 1970s (Stiny, Gips, 1972)[57]. They compared it to phase structure grammars, which were introduced by Chomsky in linguistics. While phase structure grammars use an alphabet of symbols to generate a one-dimensional string of symbols, shape grammars use singular shapes and produce multidimensional complex shapes. A shape grammar is defined in terms of shapes and rules. The rules in this case are strictly compositional. Shape grammar is oriented toward the initial stages in the design process where initial alternatives are generated (Stiny,1980. Knight, 2000). Several shape grammar-based form generation approaches were developed in the last two decades: Straightforward shape grammar approaches allow the generation of complex forms by defining basic formal rules. An example of this type of application is the Shaper2D tool, developed by Miranda C. McGill for her master's degree thesis (McGill, 2001). Similar approaches are found in 3DShaper by Wang and Duarte (2002), GEdit by Tapia (1999) and Xp-GEN by Pak et al (2003).
Analysis-based approaches try to define the formal rules of certain building types or building designs by an architect and then to generate a new building in a similar style. Koning et al developed an application that can generate Frank Lloyd Wright prairie style houses (Koning et. al. 1981), while in his PHD dissertation Duarte developed a way to generate buildings based on a grammar derived from the analysis of Alvaro Siza's buildings (Duarte, 2001).
Formal rules were utilized as a form generation mechanism in numerous approaches other than shape grammar. Shaviv et al developed a rule-based application that uses solid modeler as a "high hierarchy architectural language" to generate models of church basilicas (Shaviv et al. 1990). Oxman developed an application to generate kitchen layouts based on predefined elements (Oxman, 1992). Sei Watanabe (2002) developed an application used to design a spatial framework structure based on formal computer manipulations (see Figure 34).
Only a few of the rule-based applications have been distributed commercially; they include Genesis, an intuitive interactive and automated generation of complex 3-D designs in

---

[57] Further information about shape grammar (history, people, projects, etc.) can be found at www.shapegrammar.org

context, which has been applied to the design of Queen Anne-style houses and aircraft systems (Heisserman et al, 2000), ArcKaos by Tov Strikovsky (http://kaos4-design.com) and Automason (http://www.automason.com) (see Figure 35).

Another formal rule-driven approach derives its inspiration from nature. In Digital Botanic Architecture (D-B-A), Dennis Dollens presents a series of designs that were generated in Xfrog using rules that derive from nature (mainly botany) (Dollens, 2005). Two design labs are at the forefront of a wider-ranging approach that seeks genetic rules or code for the design of architectural forms: The first is the International University of Catalunya's ESARQ (Escuela Técnica Superior de Arquitectura) program, which promotes the Genetic Architectures line of research (Estevez, 2005) and the second is the Politecnico di Milano's Generative Design Lab headed by Celestino Soddu (http://www.generativedesign.com/). Both labs concentrate on defining formal codes/genes for the generation of "genetic" architectural forms.

In a different type of rule-based approach promoted in the late 1990s by Greg Lynn (Lynn, 1999), architectural form was generated by modifying an initial form with "forces," form modification rules coded to objects that were placed around the initial form (see section 7.3.2).

Shape grammar and the other rule-based "mechanical" approaches described in this section use computer processing power to generate intricate forms that could not be achieved within a normal design process. The fact that these forms were generated by computer rules makes their geometric logic easy to understand, thus facilitating better control, modification and construction. Nevertheless, since these forms are based only on formal rules, their impact on architectural design is arguably limited to realms such as creativity and inspiration. In terms of fitness criteria or the ability to differentiate between the generated forms, this approach alone does not offer any real solution.

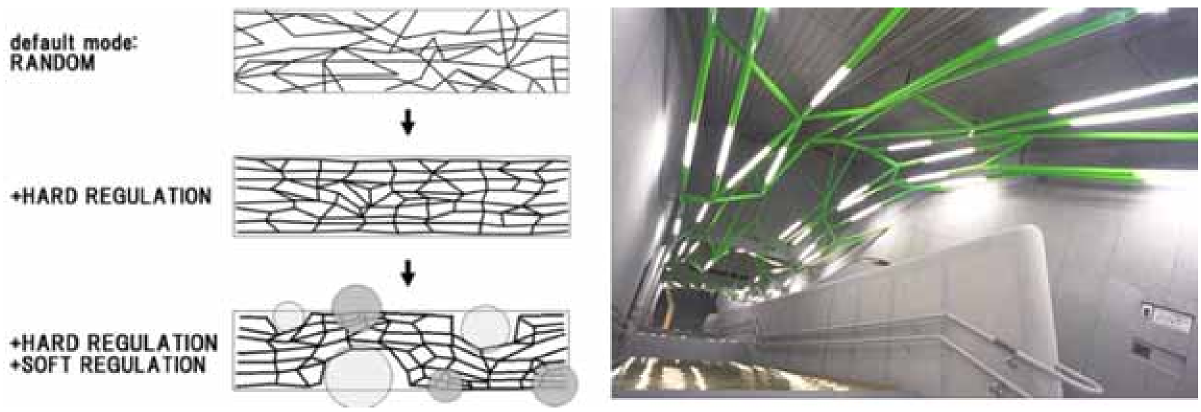


Figure 34 – Web frame, Sei Watanabe's computer-generated spatial framework in Lidabashi station, Japan (Sei Watanabe, 2002)

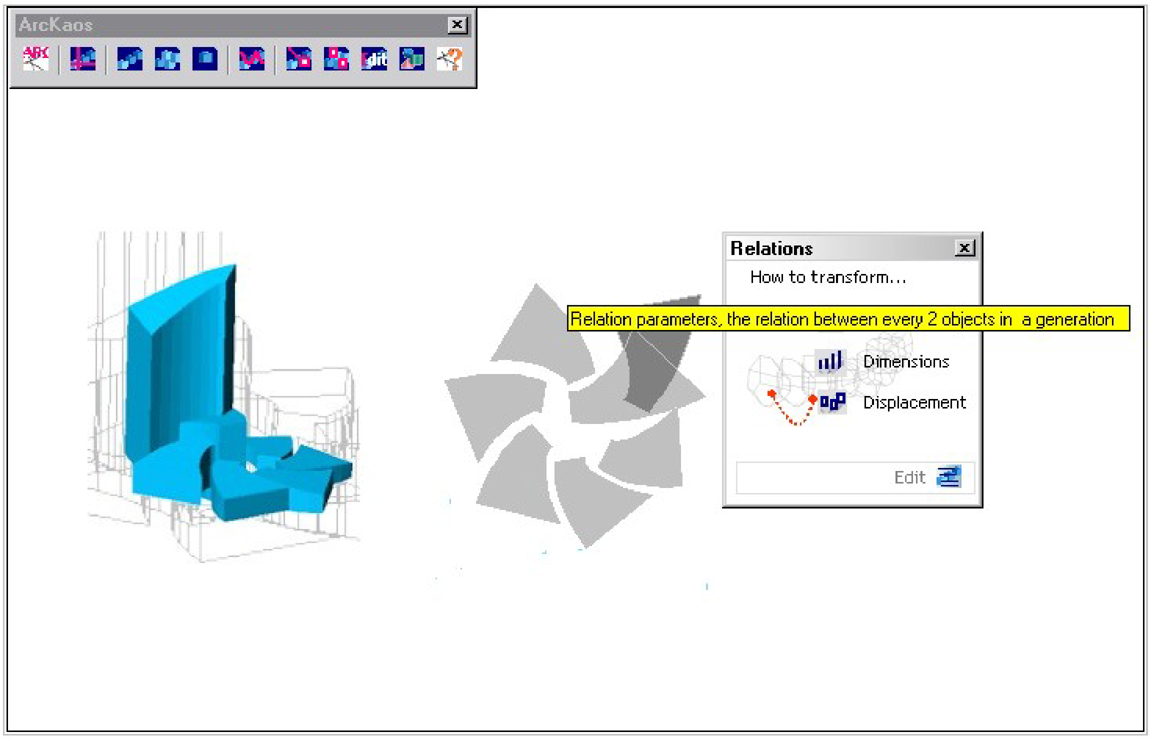


Figure 35 – ArcKaos – rule-based commercial software (image: http://kaos4-design.com)

To step beyond strict formalism, this approach must be associated with evaluation algorithms, as in the design of a coffeemaker by Agarwal et al in which shape grammar was associated with cost evaluation (Agarwal et al, 1999). A different approach could be to embed grammar modules in performance-based or another type of generative approach, as suggested by Shea and Cagan (1999).

8.5.3.5 Evolutionary methods

The main concepts behind evolutionary design are presented in the section that discusses form optimization (section 8.4). The following section presents several applications that represent various directions in evolutionary methods with a clear applicative approach[58]: GADES (Genetic Algorithm Designer) is a software program developed by Bentley (1999) that offers differentiation into genotype and phenotype. Each block of genes is a coded primitive shape and each gene a coded parameter. “A mutation operation is used within the genetic algorithm to vary the number of primitives in a design by adding or removing new blocks of nine genes from chromosomes” (Bentley, 1999). Following this random increase in the form’s complexity, the genetic evaluation algorithm examines the results and chooses the best solutions. Based on testing the design of a table, a boat, a hospital layout and a car, Bentley found that about 500 generations are needed to develop a solution (Bentley, 1999).

[58] See an explanation of the difference between optimization and generation in the introduction to this chapter.

GADES is the only tool located during this research that is capable of handling an entire design process, from zero to design prototype. However, the design scenarios in which it was tested were rather simple in terms of the fitness criteria used to evaluate the forms. It is reasonable to assume that in more complex problems (as in designing a building), it would be more difficult to define the fitness criteria[59]. An important feature of GADES is the possibility of adding user preferences to the evaluation, which could be helpful in managing complex fitness criteria scenarios.

Genr8 is a plug-in for Alias/Wavefront's Maya developed in 2001 at MIT. Based on an L-System growth algorithm, it generates surfaces, rule-based reflectors and retractors that are used to modify the initial surface and a genetic algorithm that "improves" the surface[60]. Genr8 was promoted and tested by the AA Emergent Technologies and Design program and the MIT master's program. Genr8 was used to produce some prototypes, but because it does not offer any way to introduce fitness criteria other than the one that controls the surface properties in the genetic algorithm, its use is limited to the form-based design domain and is not really different from manual manipulation of form.

Karl Chu argues for evolutionary design method in his X Phylum project (see Figure 36). These forms are generated by writing an initial algebraic formula that develops continuously through further generations (Chu, 2000; Imperiale, 2000). Although highly intricate, it is unclear from Chu's writing what fitness criteria are used to choose the appropriate solution in a certain generation. The criteria appear to be strictly formal, making the major advantages of these evolutionary methods their ability to generate many solutions.

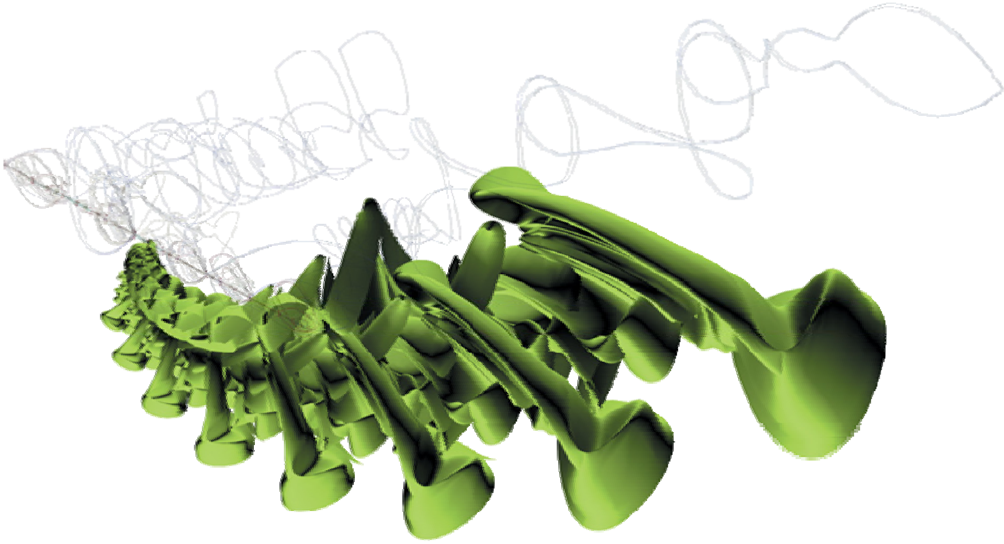


Figure 36 – Karl Chu, X Phylum. (image — ArchiLab collection)

[59] An important feature that is offered by GADES is the possibility to add user preferences to the evaluation process. This feature has a potential to be helpful in complex fitness criteria scenarios.

[60] Information on Genr8 can be found in http://projects.csail.mit.edu/emergentDesign/genr8/index.html.

Although evolutionary methods are widely used in dealing with optimization problems in other disciplines, this research found no sign of evolutionary-based applications that are embedded in architectural software or used on a regular basis by architects in practice.

8.5.3.6 Geometric constraints-based form generation

Geometric constraints-based form generation refers to design methods that generate form via writing coded parametric constraints that control the formal relationship between several geometric forms (see Figure 37). This tool permits the designer to set down basic elements of the architectural project's geometry, to define key control points and other parameters for these elements, and then to establish relationships among them such that alterations made to one or more of them will cause corresponding changes in the others. The development of this approach can be traced to the UK office of Foster + Partners (Whitehead, 2003). Employing Bentley's Microstation software as an interface for the new code, the office used this method to generate the initial form of several projects including the Swiss Re building, the London City Hall and the Gateshead Sage Music Center (see Figure 37). In the British Museum's Great Court design, instead of working with geometrical constraints, an algorithm was developed to negotiate the complex geometry of the basic design idea (see Figure 38) (Williams, 2004). This method is less interactive than the ones previously described in this section and demands greater knowledge of mathematics and programming.

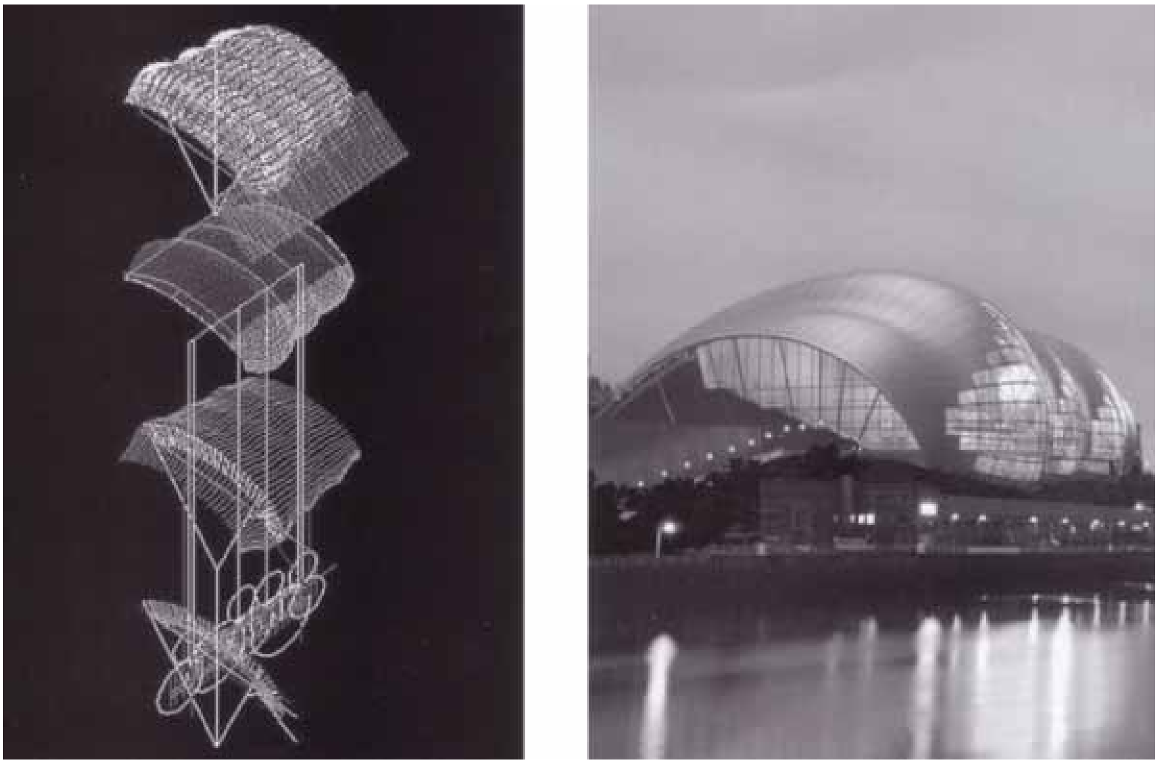


Figure 37– Foster + Partners – geometric constrain based form optimization (Kolarevic, Malkawi 2005)

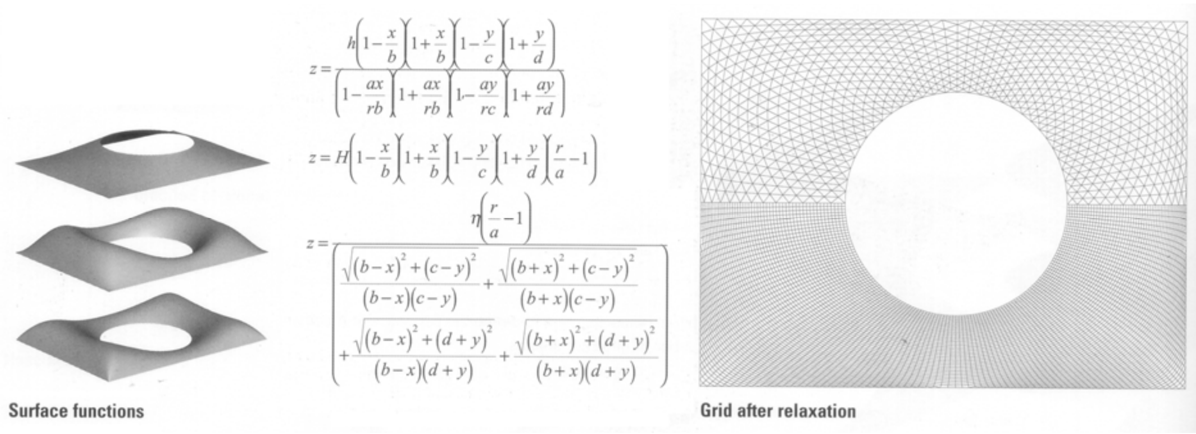


Figure 38 – Chris Williams, algorithm used to generate the dome of the Great Court at the British Museum, which was designed by Foster + Partners (Williams, 2004)

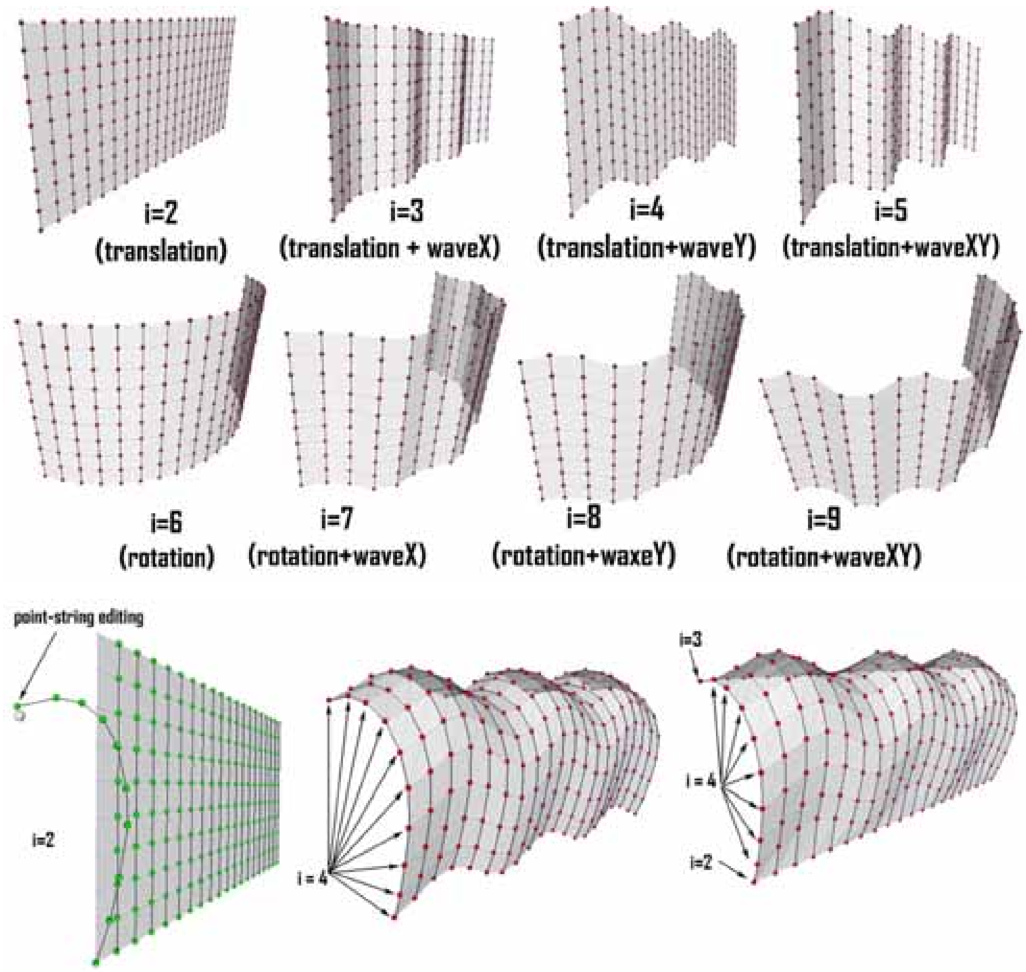


Figure 39 – Pictured above: Smart Cloud of Points (SmCP) – Applying various "i" values to the smart cloud of points model. Below: Exploring various typologies by assigning different "i" codes to the basic profile points (Nir and Capeluto, 2005)

This idea is being developed further in two main directions: the first is similar to the approach suggested by Nir in his doctoral thesis and later developed into a commercial tool called Paracloud. It addressed the “need for schematic representations which allow handling real-world problems with simplified interfaces and the ability to drive multiple representations from a single logical model ...” (http://www.paracloud.com). Instead of geometric constraints or direct algorithms as in the case of Foster + Partners, Paracloud is based on “smart cloud of points” in which an “i” parameter that can be coded with geometrical and performance information was added to the x,y and z Cartesian parameters – see Figure 39 (Nir and Capeluto, 2005).
The second approach is based on the work of the SmartGeometry research group, which argues that “Architecture is fundamentally about relationships. Many of those relationships are geometric in nature or find a geometric expression” (http://www.smartgeometry.com/). Generative Components, an application currently being developed by Robert Aish, a member of Bentley Systems’ SmartGeometry Group, aims to generate form by defining geometric form with parametric coded constraints (see also section 5.2).

8.5.3.7 Performance-driven form generation

Performance-driven form generation refers to the idea that performance data can be used to generate architectural form. As opposed to the previous geometric constraint-based form generation, which is oriented toward geometry in terms of its target function (surface division, finding the most suitable curve in terms of geometrical relations to other curves), in performance-driven form generation, performance simulation is used directly to generate the form; in many cases, it is also the target function/criterion. To date, this study has identified only single-performance criteria solutions in performance-based form generation.

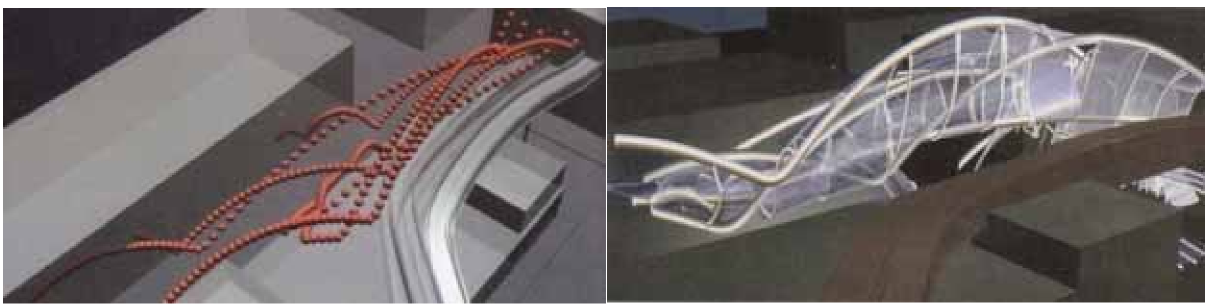


Figure 40 – Greg Lynn’s Port Authority Bridge design. At left, deflected particle flow. At right, translating particle trajectory to physical form (Lynn, 1999)

Of the two main approaches for this type of form generation, the first uses performance simulation as an inspiration for formal expression. This approach concentrates on the formal aspects of the generated form and does not argue for performance optimization. Examples are Greg Lynn's Port Authority Bridge project in which the shape of the bridge traced a simulation of trajectories of movements in the site (see Figure 40) and the H2 house, where the formal expression was generated by data on light and traffic (Rahim, 2006).

The second approach tries to generate an optimal formal solution for predefined performance-oriented target functions, as seen in the design of a competition entry for the Florence train station by Isozaki and Sasaki, a structural engineer. The structure in this project was generated using Sasaki's Extended ESO Method, an evolutionary process that both multiplies and deletes elements during the generation process, as opposed to the currently most common practice of deletions only. Sasaki used the method to generate the final shape of the project's support columns, given the loads and the columns' desired location (Sasaki, 2007) (see Figure 41).

Capeluto (2003) developed a similar approach that used SRE and SCE data as target functions. Further details on this approach can be found in section 8.2.1.

Another performance-based generation tool, eifForm, is based on a method called structural shape. Developed by Kristina Shea, eifForm aims to develop an "overall form of a structure, together with its triangulated breakdown into structural elements and joints … it works by repeatedly modifying an initial design with the aim of improving a predefined measure of performance, which can take into account many different factors, such as structural efficiency, economy of materials, member uniformity and even aesthetics" (Shea, 2004). The aesthetic criteria were based on the relations of the generated form to certain proportion systems such as the golden section. EifForm was developed as conceptual and experimental tool. In practice it was used to generate some prototype installations (see Figure 42).

Figure 41 – Florence train station competition entry designed by Isozaki and Sasaki, a structural engineer (image: www.zero-th.org)

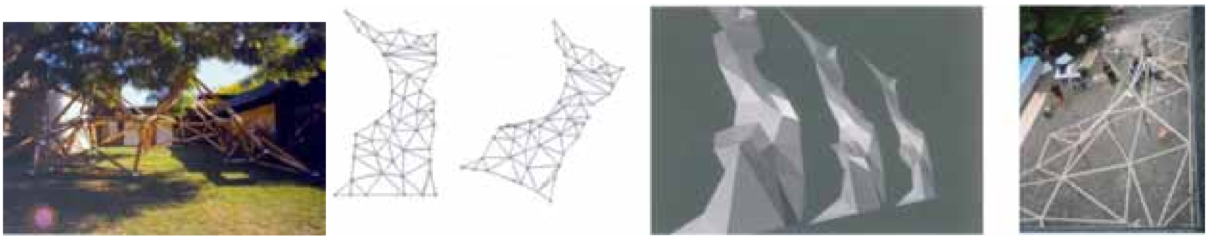


Figure 42 – On the left: Kristina Shea – EifForm generative software design prototypes (images: OSA Gen[h]ome exhibition installation, Los Angeles, 2006). On the right, Shea, Leach, Videcnik, van Mechelen installation at the Academie van Bouwkunst, Amsterdam, 2002

8.6 Summary of advantages and disadvantages of computer based form generation methods

The following table summarizes the main advantages and disadvantages of the various computer based form generation methods that were examined in this research:

Design Method/Approach	Utilization	Advantages	Disadvantages	Comments
Cellular Automata (CA)	**3,4,5**	**2,4,5,9*,10**	**2,3,7**	**CA is still being developed and used for generation and simulation of cities growth in urban scale and for generation of complex formal expression in building scale. * relates to using application based on CA, not to programming and developing new application**
Automated floor plan generation, Space allocation	**3,4,5**	**2,7**	**1,6,7**	**Space allocation based approaches where popular until mid 1990s. This research did not trace any significant recent research in this direction.**
Case Based Reasoning, Expert Systems	**4,5**	**4,5,9,10**	**2,4,5,7**	**Case based reasoning approaches where popular in the 1980s and 1990s.**
Shape grammar, Rule based form generation	**1,4,5**	**4,5,9**	**2,3,5,8**	**Shape grammar and rule based approaches introduced in the early 1970s and are being researched and developed up to date.**
Evolutionary methods	**2,4,5**	**2,3,4,5,6,8**	**6,7,8**	
Geometric constrains based form generation	**2,3,4,5**	**4,6,7**	**4,6,7**	**This approach is mainly promoted by practices in the UK (Foster and Partners and others) commercial software based on this approach is**

Design Method/Approach	Utilization	Advantages	Disadvantages	Comments
				offered by Paracloud. Another software is currently developed by Bentley (Generative Components)
Performance driven form generation	**3,4,5**	**4,6,7**	**4,6,7**	**Only single-performance based solutions**

Table 1 – Evaluation of computer based form generation main approaches

Utilization:

1. **Embedded in commercial architectural software used by architects.**
2. **Embedded in commercial architectural software used by other professions within the building industry.**
3. **Special software tool developed and used by architectural practices.**
4. **Special prototype software tool developed and used in academic research.**
5. **Theoretical model.**

Advantages:

1. **Uses computer processing power to calculate all possible alternatives within the given constrains.**
2. **Generation of design alternatives.**
3. **Applicable for several stages in the design process (can deal with differential detailing level of the generated form).**
4. **Possibility to generate/modify 3-D forms.**
5. **Allows user interaction during the generation/modification process.**
6. **Involves evaluation process for single fitness criteria.**
7. **Involves evaluation process for multiple fitness criteria.**
8. **Involves evaluation process for both qualitative and quantitative fitness criteria.**
9. **Easy to use, does not need special training/knowledge.**
10. **Suitable for various scales, i.e. urban scale, building scale.**

Disadvantages:

1. **Possibility to generate/modify 2-D/2.5-D forms.**
2. **Involves mainly formal aspects.**
3. **Usually does not involve evaluation process.**
4. **Linear approach (modifying a single solution)**
5. **Applicable mainly for specific building types.**
6. **Necessitates special training.**
7. **Applicable mainly for problems that involve small number of parameters.**
8. **Usually does not allow user inputs during the generation process.**

8.7 Gaps in the integration of computer-based optimization and generation in architectural design

It is clear that some of the focus of optimization and generation research has shifted toward improving its applicability to design. The previous sections presented approaches and implications of possible generation and optimization methods in architectural design. Whereas finding an optimum solution to a single-criteria, performance-based problem such as shading and acoustics or generating initial form based on a single criteria seems to be feasible, a general design problem (multi-criteria) is more complicated, and cannot be easily translated to parameters that can be expressed in a generation/optimization algorithm. The following directions have been identified by this research as worth pursuing both for embedding form generation and optimization tools into architectural design and for developing a new model within this premise:

Integration of simulation models into architectural software – Two main gaps exist within this premise; the first has to do with the fact that to date, most architectural design software does not include simulation tools. Although ill-defined and partly qualitative, some aspects of architectural design problems can be examined quantitatively (wind, load, acoustics, shading, light). As noted earlier in this chapter, during the last two decades many powerful simulation tools have been developed for disciplines in the building industry that are parallel to architectural design. It is time to shift the emphasis from making good tools that optimize/fix mistakes to creating tools that would allow designers to avoid mistakes in the first place. This will not eliminate the need for professional consultants; on the contrary, it will help raise the level of performance control on the designed building.

In order to use performance simulation as part of the design process, architects will have to increase their professional knowledge in these fields and develop new design methods. Since the architectural design process is extremely time-consuming, it is logical to assume that to be used by architects, integrative simulation tools must be embedded in architectural software or have a straightforward import/export interface. The need to use separate models for analysis/simulation and 3-D modeling, will arguably limit its usability to firms and projects whose scale allows a considerable investment of time in a design.

The second gap has to do with the type of the simulation needed for architectural design. The nature of the design process makes it essential for simulation modules to be able to work at varying levels of accuracy in terms of the level of information available about the architectural form. Generally speaking, during the early schematic design stage, when the initial form of the building is determined and some decisions, such as the building finish materials, are still left open, simulation software's main aim is to inspire and support decisions, that is, to help the designer avoid mistakes, save time and get more information on the various possible formal approaches. During the later stages of the design simulation, the software's main goal shifts to helping the designer fine-tune specific performance aspects.

Interactivity – Most architectural design software programs do not show in real time the results of a simulation process but present it at the end of the process. In shade simulation, for example, there are two types of representation of simulation results: the first shows a continuous simplified version of the real shade conditions on the designer's computer screen and the second requires the designer to activate the Render command in order to see a realistic simulation of the shade conditions. The second method was created because performing the simulation process continuously would consume too much processing power. Nevertheless, it can be argued that in architectural design, as opposed to other professions, it is important to be able to view simulation results constantly, especially in a multi-criteria optimization scenario, where several performances must be negotiated. The possibility of simultaneously examining several simulation results could greatly benefit designers when performing visual evaluations of the design. Therefore, a future integration of other simulation models should consider incorporating a simplified mode in which a result is always presented to the designer.

Design alternatives – Since it looks likely that complete (entire building) multi-criteria generation/optimization will not be possible in the near future, generation and optimization as design tools should be redirected toward design exploration and the development of alternatives and variations. This will boost the designers' creativity, rather than provide highly subjective answers to an oversimplified problem. Computer optimization and generation can harness the increasing processing power to allow ever-growing numbers of design alternatives to be created in a short time. Moreover, the emerging forms in this type of generation are often unexpected. A similar approach was suggested by Kim et al (2002) for optimization in structural engineering and by Frazer, Bentley and Koutamanis in their research on evolutionary design and computer form generation (Frazer, 1995. Bentley, 1999. Koutamanis, 2000).

Differential accuracy – Since the level of information on the project's formal expression increases during the design process, generation/optimization tools must be able to adapt to the various levels of inputs and demands. This could be accomplished in two ways: The first is to use different algorithms for different stages of the design. These algorithms should be highly specific and presented to the designer as a menu of algorithms divided as to type of parameters, stages in the design and desired solution. Robinson at el (1999) argued that this kind of approach would also save computer processing power since accurate optimization usually involves highly complex calculations and thus demands extensive processing power resources. This conclusion is supported by another argument suggesting that the processing power demands of intricate optimization tools will always be close to technological limits, since it will develop concurrently with the increase in computer processing power.

The second way calls for using the same tool in various stages of the design process by introducing preset or default values for data that has not been defined by the designer. This approach was demonstrated by Capeluto (1991) and Papamichael (1997).

User preferences and qualitative optimization – Mitchel argued already in 1992 that since design is “exploration of the interrelationships between beliefs and possibilities,” optimization should be regarded as a boundary case (Mitchel, 1992). In addition to the option of choosing from several alternatives based on subjective qualitative criteria, which is implied earlier in this section, the user should also be able to define priorities in term of fitness criteria. Since the evaluation of multiple criteria necessarily involves subjective decisions regarding the priorities of the various criteria, it would be logical to allow designers to change these priorities and compare the various solutions from the different evaluation runs. Moreover, it can be also argued that allowing for designers’ inputs, which are based on intuition, will increase creativity and narrow the design space to more accepted (for the designer) directions[61].

User inputs are starting to be embedded in software tools. A basic option for user inputs was embedded in Bentley’s GADES optimization software (Bentley, 1999) and suggested by Balcomb and Curtner in the MCDM software (Balcomb and Curtner 2000).

8.8 Closing remarks

This chapter presents an overview and analysis of computer-based generation and optimization research approaches and applications that were developed in architectural design. It examines the possibility of implementing computer-based generation and optimization methods in the architectural design process. It uses this study’s previously discussed categorizations and definitions regarding form and performance-based design to examine the feasibility of using empirical performance-based information in architectural design.

The idea of embedding optimization and generation modules does not suggest that architects will replace professional consultants. On the contrary, having "in-house" tools with which to generate and optimize the design (at least, to a certain extent) will mean that the initial model sent to the consultant for evaluation and optimization will be much more advanced in terms of adherence to performance criteria. This will necessarily lead to a better

[61] See Eckert et al (1999) for a discussion of the possibility of human-computer synergy in generative design.

optimization of the final design solution as the specialist consultant will be able to concentrate more fully on fine-tuning the solution. Moreover, mistakes born of a lack of empiric information about certain performative aspects of the design prior to the consultants' involvement could be avoided, thus saving time and money on changes and redesign.
In conjunction with the analysis of existing approaches, this chapter also delineates the main gaps that need to be bridged when trying to implement these tools in architectural design.
The next chapter presents a new proposed model for form generation in architecture, which is based on the use of performance envelopes. This model, which includes the development of computational tool and design methods (chapter 10), will tackle and try to fill the gaps that are outlined in this chapter.

9. Using performance envelopes in architectural design

9.1 Introduction

The following chapter discusses possible ways of using performance envelopes in a computer-based form generation process within the architectural design process. It puts forward a new model that uses multiple performance envelopes, which is based on existing computer tools and proposed future tools.

9.2 Performance envelopes and the logic behind using them in a design process

Performance envelopes are surfaces that connect points with similar information regarding desired, obtained or required performance. For example, the wind performance envelope will be defined by all the points in the design space with a similar wind speed. Figure 43 shows a section view of 1, 4 and 5 m/s wind performance envelopes.

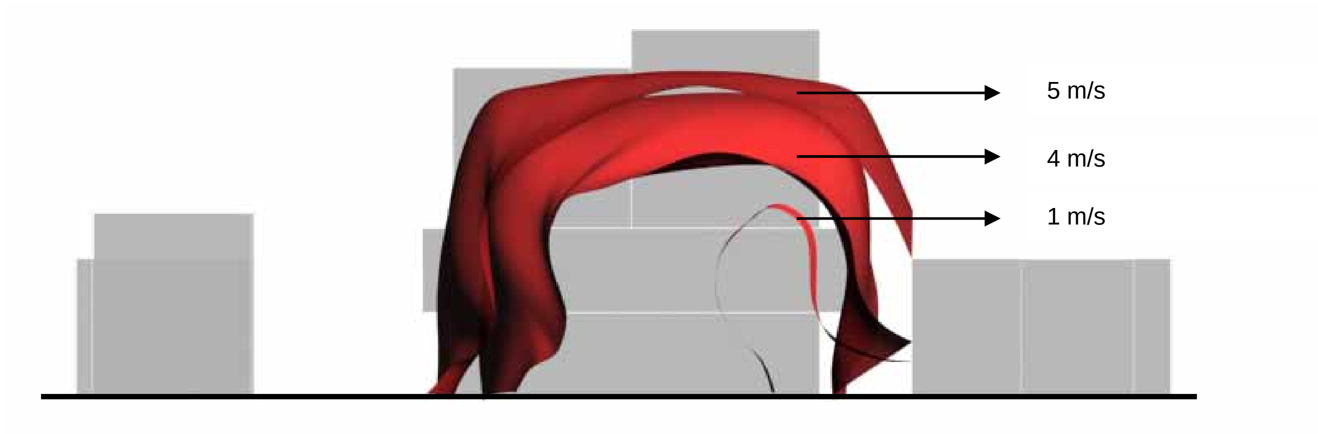


Figure 43 – Section view of 1, 2, 5 m/s wind performance envelopes

Currently it is possible to generate performance envelopes using several existing simulation software programs. Many of them are oriented toward building performance simulation and support graphic representation of performance envelopes. Some software, such as SUSTARC, supports the presentation and export of performance envelopes as surfaces. Similar to Schmitt's definition of types of fitness criteria (Schmitt, 1992), performance envelopes can be divided into three main types based on their use in design: discrete values, threshold, and range.

Employing a discrete value envelope suggests that the envelope's surface is used as an architectural form. Using a single envelope as a threshold defines a boundary for the solution space of the architectural form. A range type defines a solution space between two boundaries. Figure 44 presents a series of diagrams showing the various solution spaces that could be defined by different basic types of one or two performance envelopes.

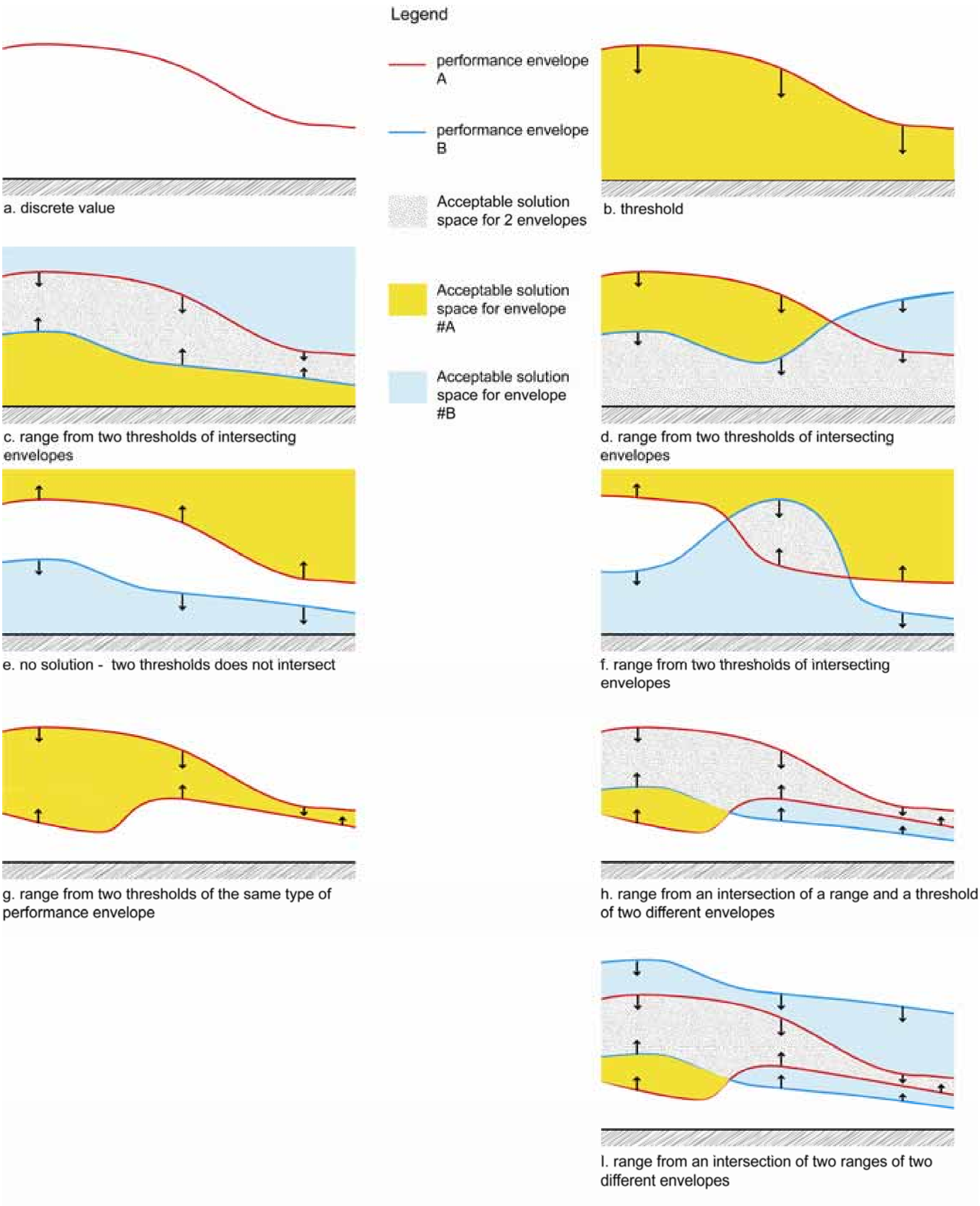


Figure 44 – Basic solution space from one or two performance envelopes

Many contemporary design processes in architectural practices use simulation results in an "after-the-fact" manner. In this kind of design process, the design solution's adherence to performance criteria is examined using simulation software and modified according to the result in order to improve upon it. This research proposes using multiple performance envelopes in a generative or “before-the-fact” manner, that is, to generate a 3-D design space or a building’s initial form from multiple performance envelopes. In this way, the generated form adheres by definition to the performances defined by the envelopes that were used to generate them. This can augment the general performance of the building form and save time in the design process, while producing a form that embeds a larger amount of information[62].

Generating a building form from a single performance envelope is a straightforward process. It can be done using a discrete value type of envelope, a threshold or a range (see Figure 44 a,b,g). In the first of two possible types of thresholds, the closer the result is to the envelope the better it is. In the second type, all the points within the solution space, which is defined by the threshold, have the same fitness value in terms of the solution’s quality. In this type of threshold and in the range type, the designer has to employ other fitness criteria in order to choose the best solution within the solution space (see Figure 44 b, g).

A multi-performance envelope scenario is much more complex (see Figure 44 c, d). First, performance envelopes must have a common ground to be able to generate a solution space. Figure 44e presents a situation with no common ground and the envelopes do not intersect or overlap. In this situation, one or two of the envelopes must be redefined (by changing the values of the requirements used to generate the performance envelope). Another possible situation is a local intersection (see Figure 44 f, h), in which only a local solution is possible for the two envelopes. In the areas where there is no intersection, the designer can use other performance envelopes or adhere to only a single performance. It is clear that introducing more envelopes increases the complexity of the route to the solution and the probability of conflicts (see Figure 44 l). Having more than one performance envelope necessitates a subjective decision by the designer regarding priorities and/or the employment of other external fitness criteria to be able to find a solution. Moreover, having a solution space implies that there is no one best optimal solution. The lack of a single solution suggests that we ought to think of a generation process that produces alternatives to be evaluated at a second stage rather than developing a single design in the traditional way. Being able to use and negotiate various performance envelopes in a multi-solution form generation process in architecture will help to propel the move to "lateral thinking" in design as opposed to "vertical thinking" (de Bono, 1970), a network or a rhizomatic mode of design rather than the traditional linear method (Deleuze, Guattari,1998).

---

[62] See the discussion of the increase in the amount of information in section 6.4.

9.3 Performance envelope types

Following the division into form-based design and performance-based design (see section 6.5) and the division into types of performance (see section 6.6), the following performance types were identified as usable for generating performance envelopes:

Sun shading and lighting – Sunlight and shade conditions can be represented as performance envelopes. It is possible to generate, for example, envelopes that represent solar rights requirements (a threshold envelope representing a volume that will not shade the surrounding buildings during a defined period of the year (see Figure 64). Other possible performance envelopes in this realm are solar catch (volume that is not shaded by surrounding buildings during a defined period of the year) and light envelope (volume that is defined by a threshold of a similar light level – see Figure 21).

Structure – In terms of performance, the building's structure design/calculations can be divided into several fields in which computers can be used in a simulation process. From these fields, load and tension have been identified so far as being representable by performance envelopes. Performance envelopes in this case are surfaces that connect all the points that experience equal load/ tension. Figures 22-24 present different simulation results that can be represented as performance envelopes.

Wind – A wind performance envelope is a surface that connects all the points with a similar wind velocity. It can represent extreme or average wind conditions in a specific context (see Figure 43). Wind can be examined in terms of three main effects: influence on the building structure, influence on the level of comfort people feel in external spaces and influence on the building's passive ventilation that is related to its energetic behavior.

Acoustics – An acoustic performance envelope is a surface that connects all the points with a similar level of noise. It can represent extreme or average acoustic conditions in a specific context. It can be used in an urban scale or a building scale (see Figures 28-31).

The following fields were identified as having potential to be used within this premise but were not examined in this research:

Energy (temperature) – Performance envelopes can be generated by creating a surface that connects points with equal temperature. Using a similar design method to the one presented later in this chapter, it might be possible to generate an interior space that based on temperature-related performance demands.

Visibility/lines of sight – Lines of sight from a single point in space can easily be presented as a spatial envelope. It is more difficult, however, to represent the same envelope for an area instead of a point. A possible solution can involve the calculation of the sky solid angle (SSA) (Capeluto, 2003) that examines the exposure to the view of every point in space. A surface that connects all the points with the same level of exposure can then be generated. A similar idea was developed by Ng (2003) for daylight design in Hong Kong.
Several approaches have been developed to analyze and evaluate lines of sight for an entire building (Fisher-Gewirtzman et al, 2005). However, none was identified by this research as representable as a performance envelope for more than a single point. Therefore, it seems that line-of-sight performance envelopes can be used only in the local type of generation process described in the next chapter.

Circulation – A circulation/flow performance envelope refers to a surface that describes the space used ("consumed") by people in movement. It is dependent on data regarding the number of people and their flow characteristics (speed, optimal distance between people, turn ratio, etc.). Flows have been used in the past by architects to generate forms (for example, Greg Lynn's Port Authority Bridge project and Asymptote's Dancer's Space installation). The way the flow data is collected/measured dictates whether the flow to become more than a conceptual form generator, as in both of the projects mentioned above. Nevertheless, because the processes suggested in this research are parametric, different data scenarios can be examined in a reasonably short time.
Circulation/flow optimization is important in projects such as hospitals, stadiums and transportation hubs. The use of flow performance envelopes could be a potentially important tool in the initial design of these building types.

Geometric constraints – Geometric constraint envelopes refer to surfaces that represent the formal threshold or range of an architectural form based on various sources of information as such city laws, material constraints and construction method. Therefore, each time a programmatic constraint can be described as a surface, it can be used as performance envelope in a negotiation/optimization process.

9.4 Initial approaches

Three main modus operandi were identified for employing performance envelopes for form generation of a design space or initial building:

a. Visual control – Based on the designer's ability to relate visually to performance envelopes during the design process, this method requires the insertion of performance envelopes into the design space. Then, the designer develops the building's initial form while referring to these envelopes; this allows for a constant awareness of performance during the entire course of the design process. Moreover, the designer can see and compare in a very straightforward and accessible way inputs from various performances and their influence on the design. Clearly, this method is not entirely precise in the control it offers over the generated form. Nonetheless, using this method does not necessitate any previous preparation or adaptation of performance algorithms and it can be used by practically any designer once the performance envelopes are inserted to the design space. Figure 45 presents two possible ways (wire frame mesh and cloud of points) to represent visual control; both alternatives can help the designer visually to limit the design form according to the chosen performance.

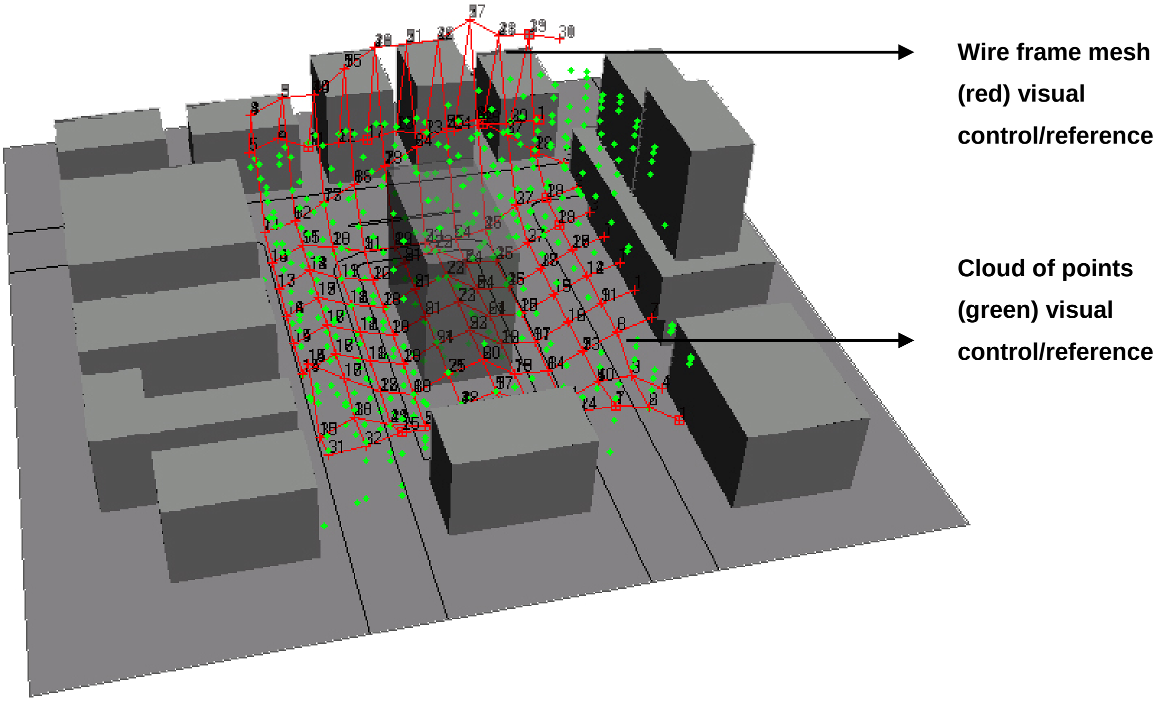


Figure 45 – Examples of visual control by performance envelopes: 1. Wire frame mesh 2. Cloud of points

b. **Parametrically controlled form generation** – This method is based on a special tool/algorithm that is able to negotiate several performance envelopes. Once the algorithm is developed, designers can parametrically control the level of influence of each performance envelope on the generated form. Two types of algorithms are suggested in this research: The first is a basic boundary algorithm, that is, an algorithm that generates an entire envelope/design in a single manner. It determines a desired ratio between the levels of influence of various performances and keeps the same ratio throughout the design solution. The second type is a local boundary algorithm, which is a variation of the previous algorithm that divides the design plot into areas of influence and designates different levels of influence of various performances to every area. An example for possible division is presented in Figure 46.

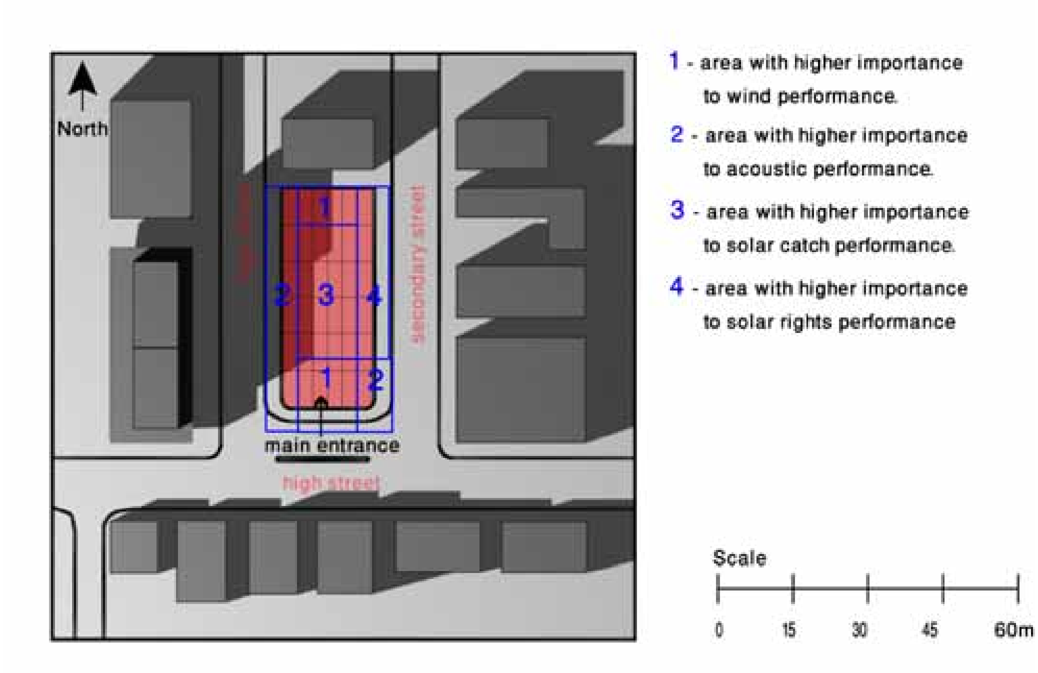


Figure 46 – Example of possible different local performance preferences.

Thus, one can modify separately different parts of the design using local performance criteria. The number of areas and their distribution (2-D division to areas or also more complex 3-D division) determine the precision of the designer’s control over the performance in the design. However, a large number of subdivisions require a more complex algorithm. Moreover, since no two projects are alike and performance criteria vary widely for every site, architect and developer, a special algorithm would have to be developed for every single project. It is doubtful that a typical architectural firm would have the capacity to develop such complex algorithms, in terms of both time and knowledge.

c. Combined parametric and visual control – Both parametrical generation and visual control are used in the initial form generation process. Two methods fall within this category: In the first, a basic algorithm is developed for the particular design scenario and used to generate an initial form that is then modified and developed by the designer employing visual control.

The second method involves employing the performance envelopes as a formal boundary for the initial form. The designer negotiates between several envelopes that represent one, two or more performance parameters. This negotiation can be done visually by parametrically changing the influence of each envelope or can be determined by an algorithm that controls the relations between the influences.

In this research, we chose to examine the use of three types of performance envelopes in the suggested computer-based form generation process:

The first two types are the Solar Rights Envelope (SRE) and the Solar Catch Envelope (SCE) – see description in section 8.2.1. These envelopes were examined separately as a threshold and together as defining the boundaries (range) of an accepted solution volume (see Figure 47).

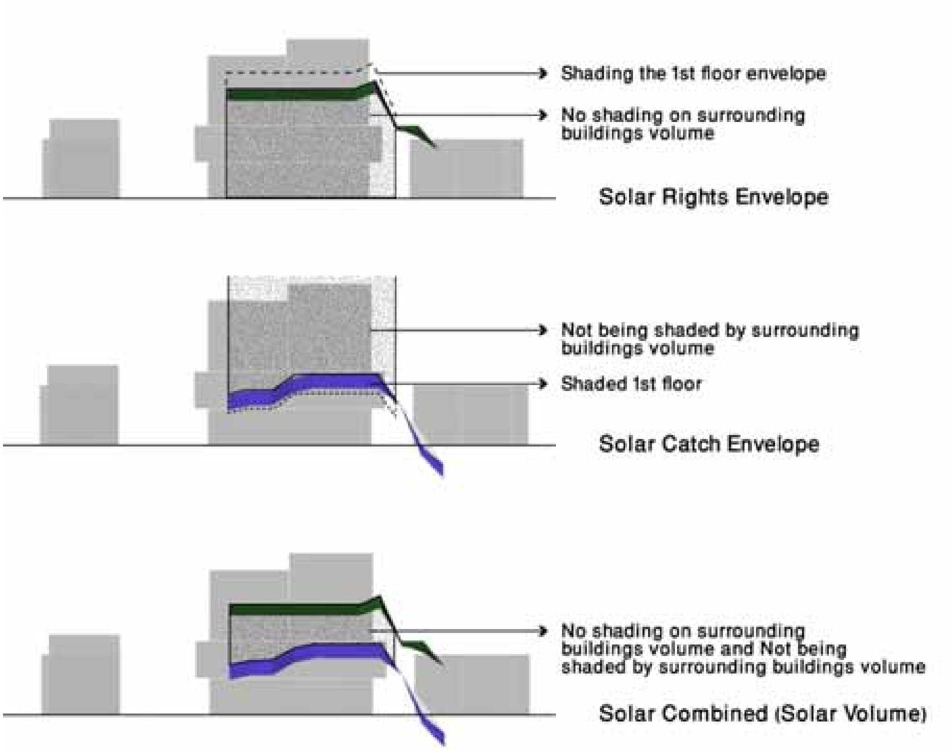
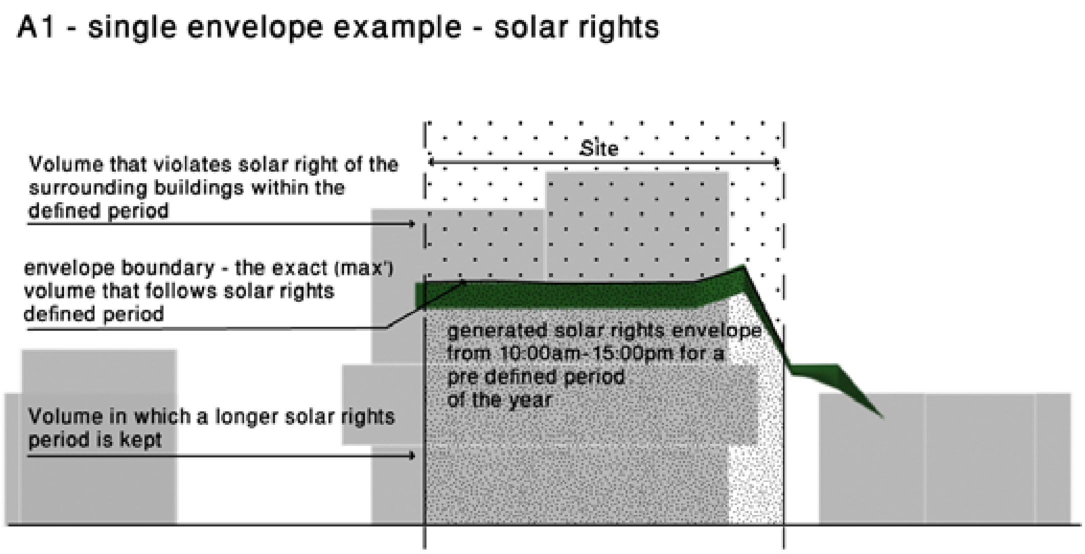


Figure 47 – Solar rights and solar catch envelopes – section view

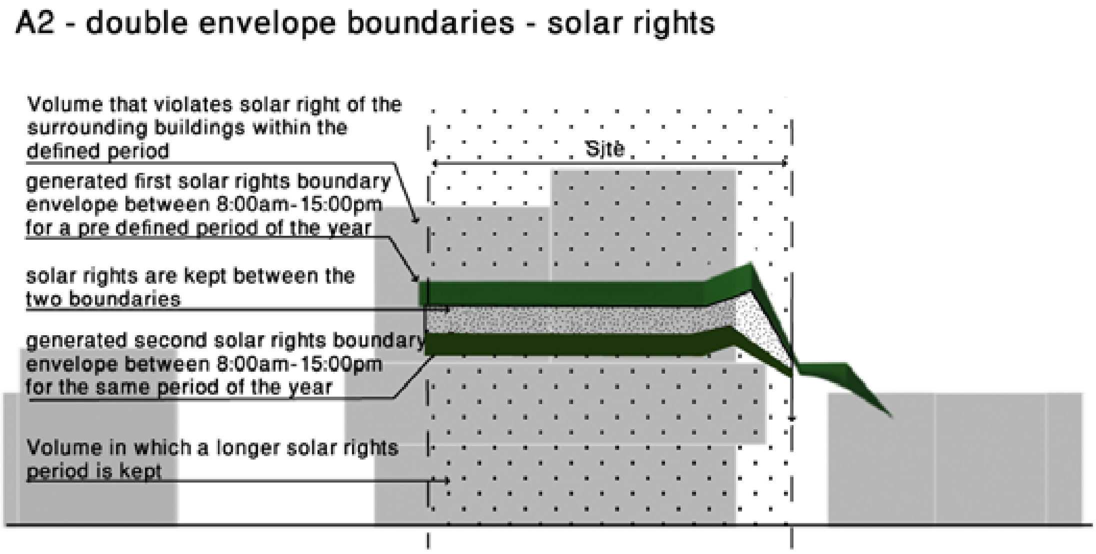
The third type is wind – an envelope that creates a surface from all the points in a chosen volume that experience equal wind velocity as a result of existing urban and environmental conditions (existing buildings, streets, topography, etc).

The following table summarizes the parametrically controlled form generation initial possibilities using the above envelopes as reference**:**

Number of parameters	Number of envelopes per performance	Type of envelope used for the example	Image	Method	Remarks
1	1	Solar rights	A1	**Threshold envelope** – An envelope generated by simulation software is used as a boundary for the initial form generation. The envelope represents a state of 100% adherence to the chosen performance criteria (preserving the solar rights of the surrounding buildings within a predefined period). The designer decides on the amount of adherence to the performance envelope. Breaching the envelope in this example means that a violation of solar rights will occur at certain hours or dates.	This envelope does not present the designer with accurate data on solar rights status when outside of the boundary line. It only presents binary status (within the envelope or outside it).

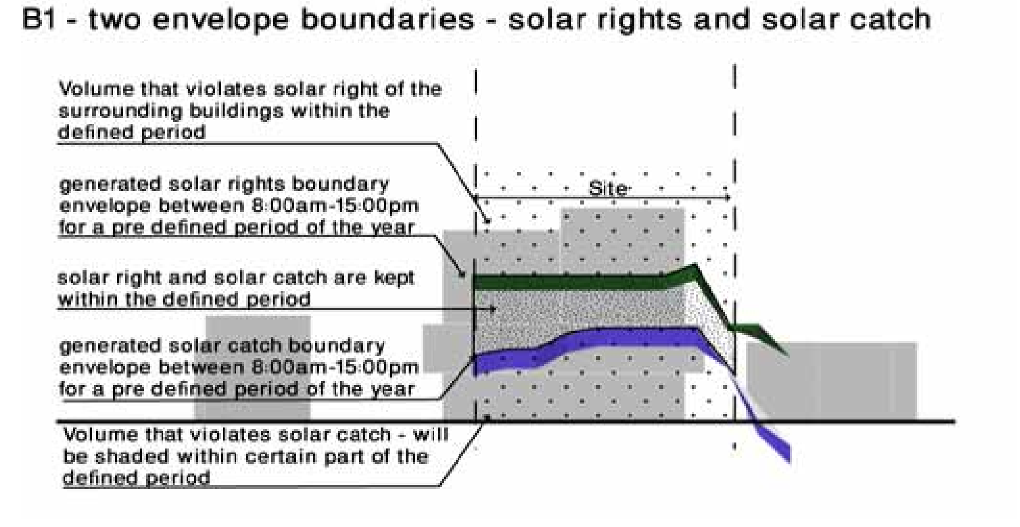


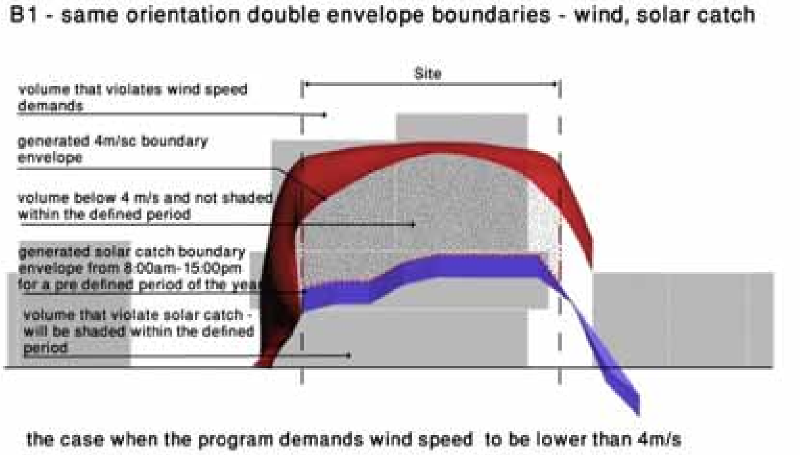
Number of parameters	Number of envelopes per performance	Type of envelope used for the example	Image	Method	Remarks
1	**2**	**Solar rights**	**A2**	**Range envelope** – Two envelopes of the same kind (performance) are generated by simulation software. Each envelope represents different period (time, date). The designer generates the initial building form using the two envelopes as boundaries to a morphing modifier. **Local boundary envelope –** opens the possibility of employing different relations between the envelopes for different areas in the design.	Does not define one optimal solution. The difference between the envelopes determines the resolution of the solution.

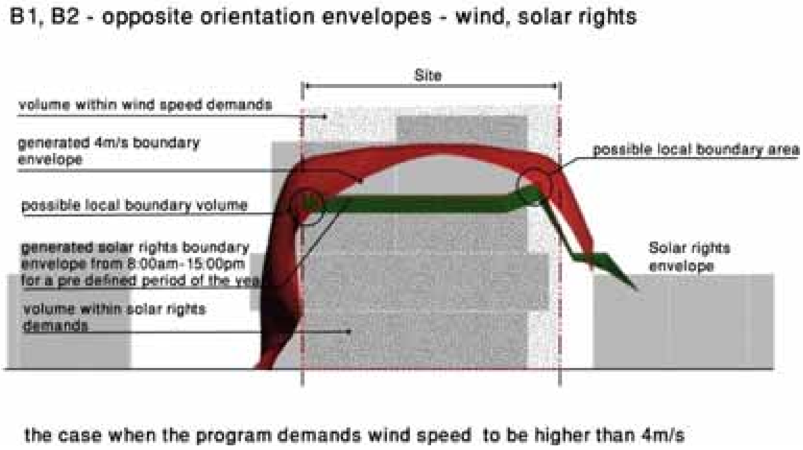


1	**3**	**Solar rights**		**Discrete value envelope + range envelopes** – Three envelopes of the same kind are generated by simulation software. One envelope represents the desired performance and the other two represent the accepted deviation from the desired performance. Thus, the designer will modify the desired envelope within the predefined boundaries. Or, when changes to the desired envelope are allowed only in one direction, two of the three boundaries can be used to define levels of change in the parameter**.** **Local boundary envelope –** see the previous approach.	
1	**More**	**Solar**		The number of envelopes influences the	

Number of parameters	Number of envelopes per performance	Type of envelope used for the example	Image	Method	Remarks
	than 3	**rights**		level of control/accuracy the designer has over the generated form. **Local boundary envelope –** see the previous approach.	
2	**2**	**1. Solar catch/ solar rights** **2. Wind/ solar rights**	**B1**	**Threshold envelopes** – two desired performance envelopes for two different parameters are generated by simulation software. Orientation possibilities: 1. Envelopes have the same orientation – the initial form is generated through negotiation between the two envelopes. 2. Envelopes have opposite orientations – this setting is not valid. The designer must redefine the performance criteria for one or both envelopes. 3. Envelopes have opposite orientations but intersect; this necessitates employing local boundaries. In the areas of intersection, one can negotiate between the envelopes and in areas of non-intersection, different envelopes must be defined.	Both envelopes must have the same orientation in order to yield a solution (see Figure 44).



<table>
<tr><th>Number of parameters</th><th>Number of envelopes per performance</th><th>Type of envelope used for the example</th><th>Image</th><th>Method</th><th>Remarks</th></tr>
<tr><td colspan="6">




</td></tr>
<tr><td>More than 2</td><td>2</td><td>Wind/ Solar rights</td><td>B2</td><td>Increasing the number of parameters increases the complexity of the problem but does not change the method. Negotiation between the various envelopes is done as in the previous method.</td><td>The number of envelopes per performance determines the level of information about performance available to the designer.</td></tr>
<tr><td>More than 2</td><td>More than 2</td><td></td><td></td><td>Increasing the number of performance envelopes and parameters increases the complexity of the problem but does not change the method. Nevertheless, it is likely that three or more envelopes simultaneously will shift the emphasis of the process toward local volume generation as opposed to entire volume generation.</td><td>The number of envelopes per performance determines the level of information about performance available to the designer.</td></tr>
</table>

Table 2 – Basic methods of using performance envelopes in a computer-based design generation process

The methods described above present a way to generate an initial form or performance-oriented formal solution space for architectural design based on empirical performance parameters. In the majority of cases in which two or more performance envelopes are negotiated, a range of possible solutions is generated. Hence, the designer has to examine these possible solutions and select the most satisfactory solution using additional fitness criteria. The parameters for the additional fitness criteria may vary between different designs and designers. To present a complete design method for the initial design stages a generative tool was developed to include both parts of the generative process (initial form via performance envelopes with additional fitness criteria from other realms). The process of developing such a tool and the final product are presented in chapter 10.

9.5 Static and animated performance envelope or re-evaluating the generation results

The methods described above use static performance envelopes to generate the initial form. The difference between static and dynamic envelopes lies in the update ratio of the envelopes. Static envelopes are generated in a single simulation run (and are not updated later), while dynamic or animated envelopes are updated according to a predefined period, or to developments (changes) in the design. This gives the designer live accurate data on the performance of the generated form. To date, most architectural software does not offer simulation capabilities (except sunlight/shade). Designers wishing to use the proposed generation method not only have to import the performance envelopes to their software's design space, but then they also need to export back the results to confirm that the generated form did not significantly influence the initial performance envelopes.

9.6 Design methods diagrams

The following diagrams present two types of design methods for the application of performance envelopes to generate architectural initial form. Diagram 6 describes a suggested design process based on existing tools. Diagram 7, on the other hand, describes a suggested design method that uses a new kind of architectural design software that embeds simulation modules. In the first process, performance envelopes must be generated externally in simulation software and then exported to the CAD environment. Furthermore, after the initial form is generated, the designer has to export the results to the simulation software in order to verify that the generated form did not change substantially the initial data on which the form generation relies.

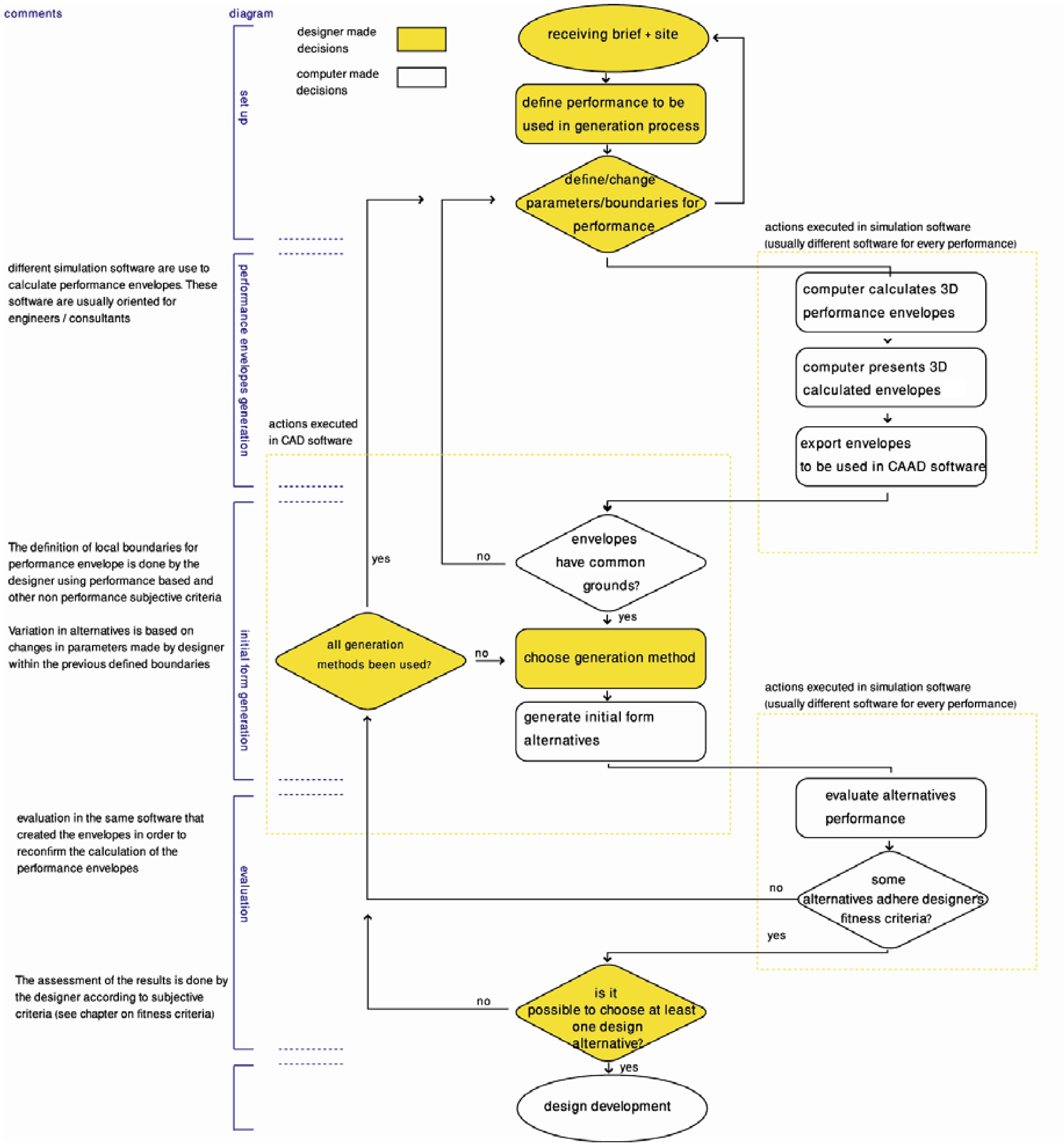


Diagram 6 – Form generation using performance envelopes, based on existing software

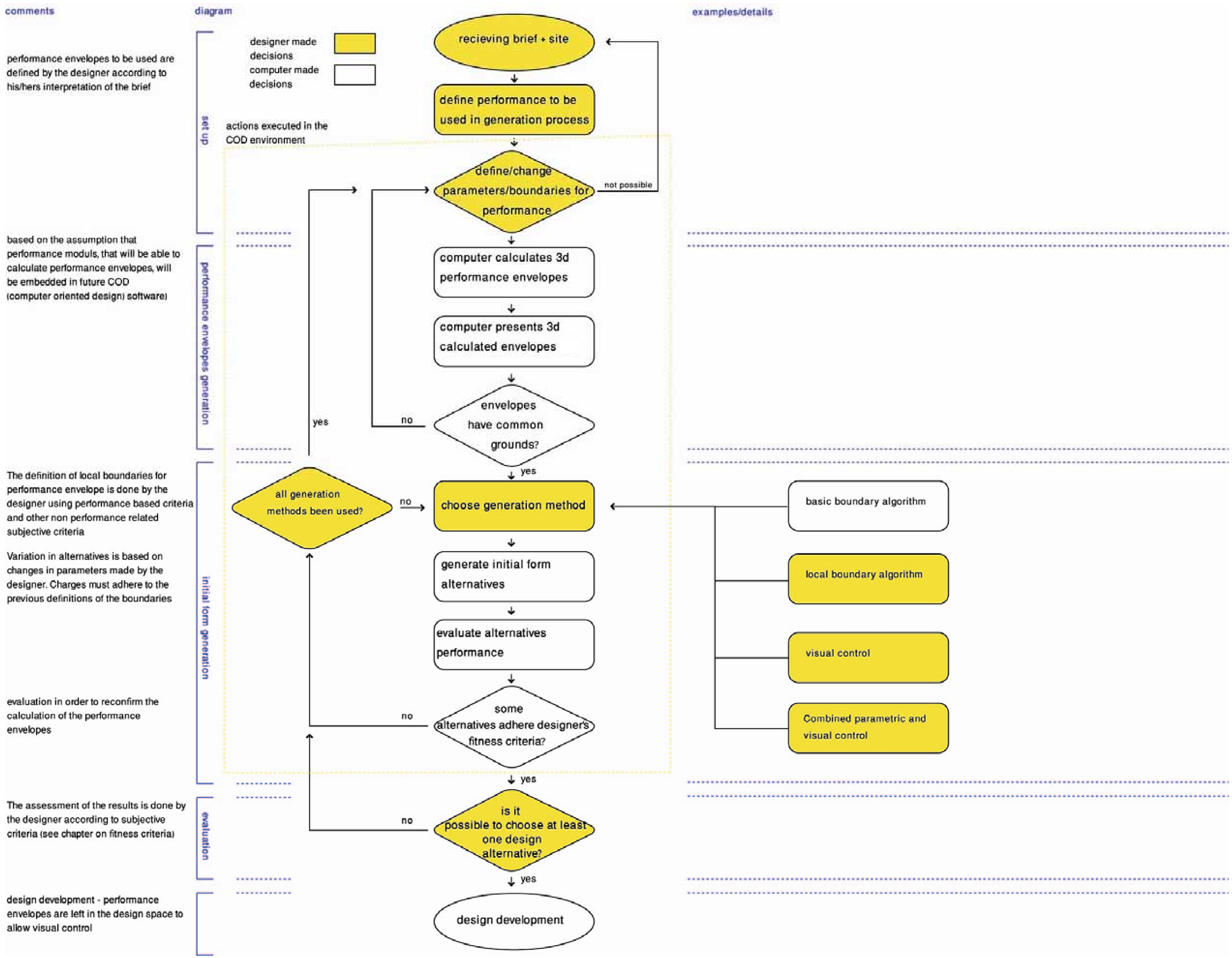


Diagram 7 – Form generation using performance envelopes, based on proposed software

9.7 Closing remarks

This chapter defines the notion of performance envelopes and its relation to the architectural design process. It defines types of performances that can be represented as performance envelopes and used in architectural design. Following these definitions it suggested and discussed possible directions for employing performance envelopes in an architectural design method. Finally this chapter presents two initial design methods that employ performance envelopes. The first method describes a future situation where the architectural software embeds simulation and evaluation modules and the second method relates to the current situation, which necessitate importing performance envelopes from simulation software before the generation process starts and exporting the form to simulation/evaluation software after the generation process ends in order to examine the results.

The next chapter discusses the development of a model and a tool that employs performance envelopes to generate architectural form.

10. Developing generative performance-oriented design (GenPOD) model

10.1 Introduction

The following chapter explains the development of the GenPOD model that uses performance envelopes in a generative computer based architectural design process. It presents several approaches that were pursued during the model development process and discusses the advantages and disadvantages of each approach. The chapter's main section discusses the proposed GenPOD model and describes the software tool that was developed as part of the model. The last section of the chapter proposes directions worth investigating in further development of the suggested GenPOD model.

10.2 Generating building's initial form using control sections and points (Approach 1.0)

As a surface, a performance envelope can also be generated by connecting 3-D lines in which all points have the same value of a certain performance. The initial approach, that was pursued to develop a method for negotiating several performance envelopes, suggested employing a regular grid of planes that cuts the performance envelope's surfaces and creates a regular grid of section lines. Then, in order to control the line parameters, control points are defined on the section lines. These points control the movement of the line in the 2-D plane of the section line. Moving a control point modifies the line on which it is located, which then changes the performance envelope's surface. Therefore, the control point will facilitate detailed control of the necessary relationship between the envelope and the building program/designer preferences, which will determine the amount of change in the movement of each control point.

An initial example of control points in relation to a single performance envelope (a threshold) is shown in Figure 48. The example shows a solar rights envelope section and a set of control points. The division into five units for the *z* axis movement presented in this example has no real significance at this stage[63]. In this example each unit represents the height of a single floor that would be shaded when crossing the defined threshold.

The main difficulty in relating to a single threshold performance envelope is that it allows a large solution space in which other criteria have to be used for making a decision regarding the "best" position. To narrow the solution space, two or more boundary envelopes can be placed around the chosen envelope, thus creating a range or a solution space limited by two

[63] From the 6 control points that were used for this example 3 points were put above the envelope (over the threshold) and the other 3 were defined on the envelope or under in (within the solution space) – this division was used in order to show that the threshold can be crossed when it is possible to measure the implication of this act. The image shows solar rights envelope and broken horizontal lines above the envelope (threshold). Each broken line represents the height of a single floor in the surrounding buildings that would be shaded if the designed building would reach this level.

boundaries. Figure 49 presents a range that is created by three wind envelopes (1m/s, 3m/s and 5m/s). The units in this example represent wind velocity – 1m/s for each unit.

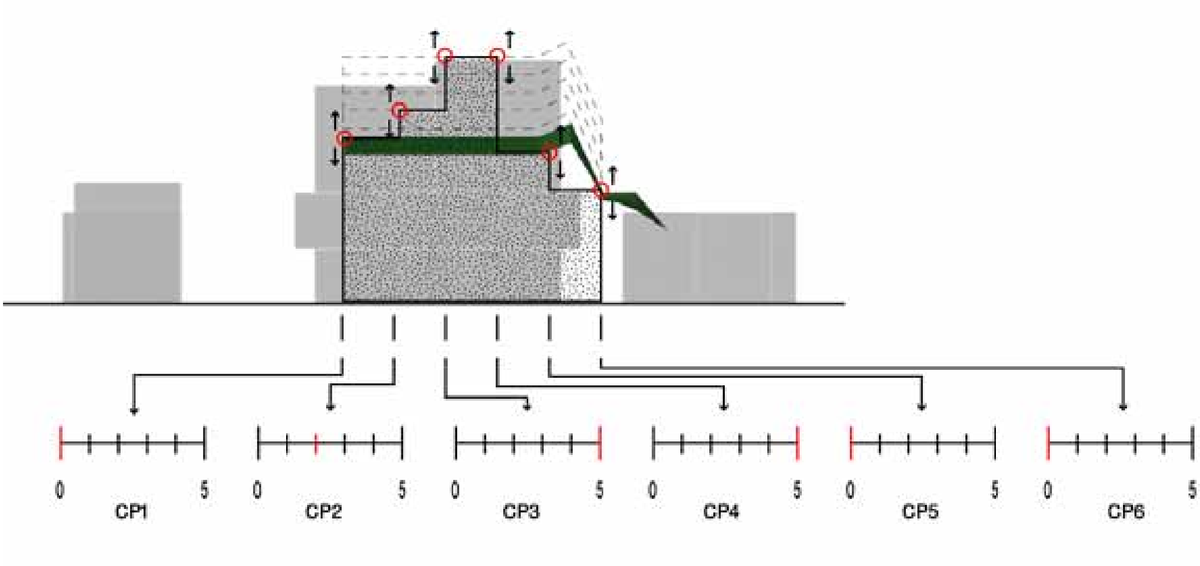


Figure 48 – control sliders for changing an initial building form in relation to a single performance envelope (a threshold)

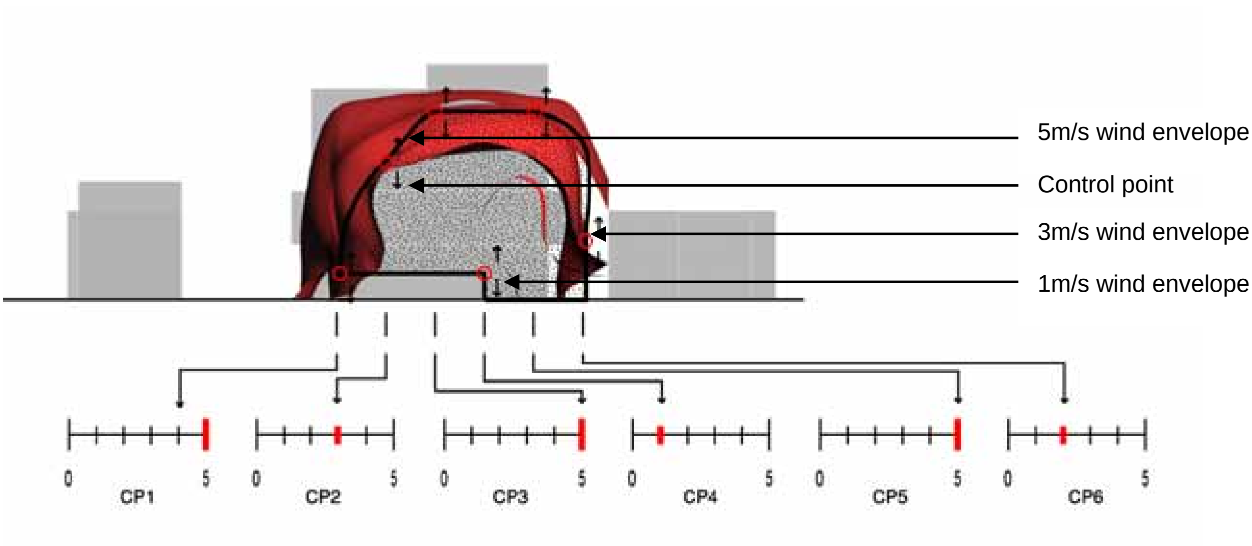


Figure 49 – control sliders for changing an initial building form in relation to the three wind envelopes (range)

In range-based solution spaces, the designer has to define the desired position of the control point within the solution space. The initial strategy examined to negotiate different envelopes within a range used a constant (*k*) factor. These factors defined the level of importance of each envelope normalized to 0-1 scale within the predefined range or solution space (the total value of adding all the factors is 1). Each envelope's level of importance is defined by the designer according to programmatic demands. The sum of the multiplication of the *k* factors and the current position values for each axis (*x, y, z*) defines the new position of the solution. An example of this approach describes using this strategy for the *z* axis in a three-performance envelope scenario where the *z* axis position of each control point (*La*) is defined by the combination of the influence of the three *k* factors (*ka*1, *ka*2, *ka*3) (see Figure 50). In this case, the *k* factor represents the level of influence of each of the envelopes (solar rights, solar catch and wind) on the end result. Therefore, in a three-envelope scenario, the location of the control point (*L*) in each axis will be defined by the following formula:

La=<u>a1*ka1+a2*ka2+a3*Ka3</u>

Where *a*1, *a*2 and *a*3 are different envelopes (in this example, wind, solar right and solar catch envelopes) and *a* represents the selected axis (*x, y, z*) axis position and *a*1+*a*2+*a*3=1.

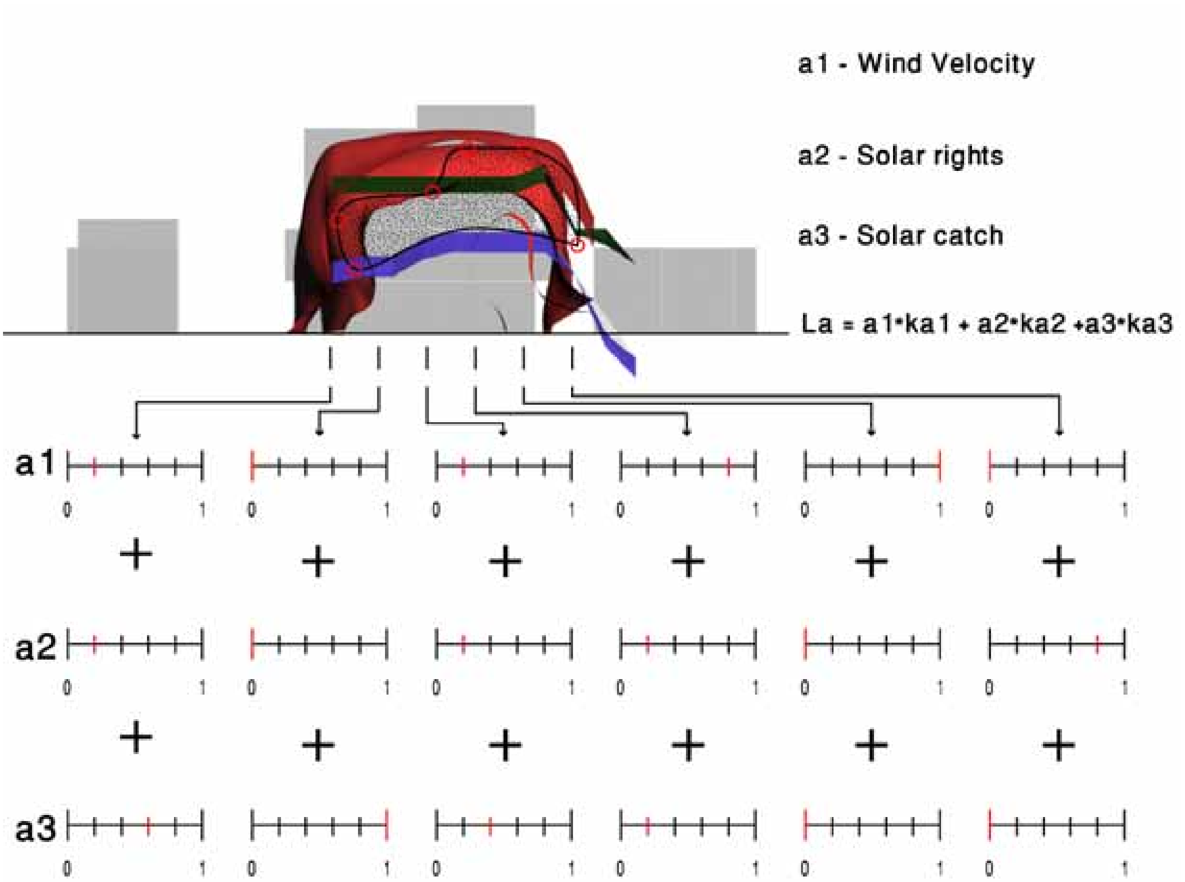


Figure 50 – negotiation of three performance envelopes using the *k* factor in a range

Figure 51 shows a more general description of the same approach. In this example a range created by the solar catch and wind envelopes is negotiated by assigning a *k* factor for each envelope. The addition of both envelopes' location multiplied by the *k* factor (according to the number of the envelopes) determines the new (*L*) location of the examined control point in that axis.

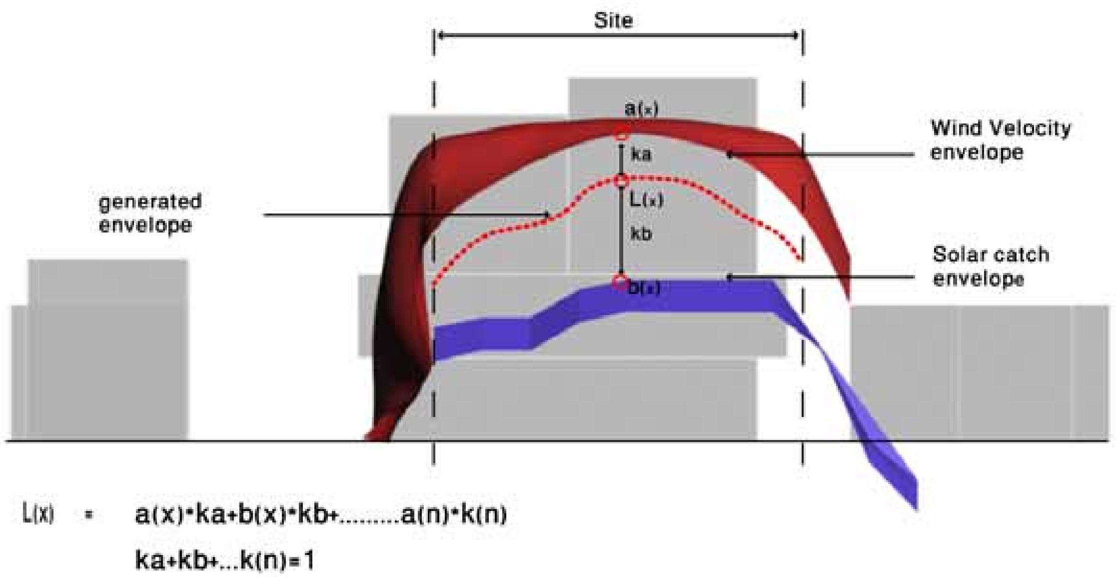


Figure 51 – generating a building envelope from two performance envelopes – general description of the initial approach

To examine the plausibility of this approach in an architectural design environment, an initial 3-D model was developed in 3ds Max, which was chosen because:
a. It has a robust script interface.
b. It has been used commercially by architectural practices; thus, it would be easy to examine the tool in a real design scenario and get comments and feedback from users.
c. It supports importing and exporting performance envelopes in formats used by simulation software.

In this case, the set of controllers was created using existing modifiers and script code. Each controller was connected to a specific area in the initial form (a chosen performance envelope) and to a specific parameter of a specified area such as overall scale, rotation and position of the entire form or the same parameters but for only a single axis (*x, y, z*). The level of each controller's influence over the initial form also can be defined, so that it can be used in much the same way as the previously described *k* factor (see Figure 52).
Attempts to use this approach and model in architectural design revealed two main problems: First, the initial form had to be decided by the designer beforehand. Thus, there was no real form generation stage in this approach but only modification; this deviated from the initial aim of generating the form directly from performance. Second, in order to control numerous parameters one must create numerous controllers (one for each specific parameter and area in the initial form). The expected number of controllers would increase complexity and fill the designer's desktop that would cause difficulties in controlling the generation process in a normal design. These two problems led us to look for a different approach that would involve the direct generation of an initial form from performance

envelopes and would be controllable in a way that would not necessitate a higher level of knowledge (i.e. climate and energy calculations, computer code, etc), experience and time than what is typical in a normal design process.

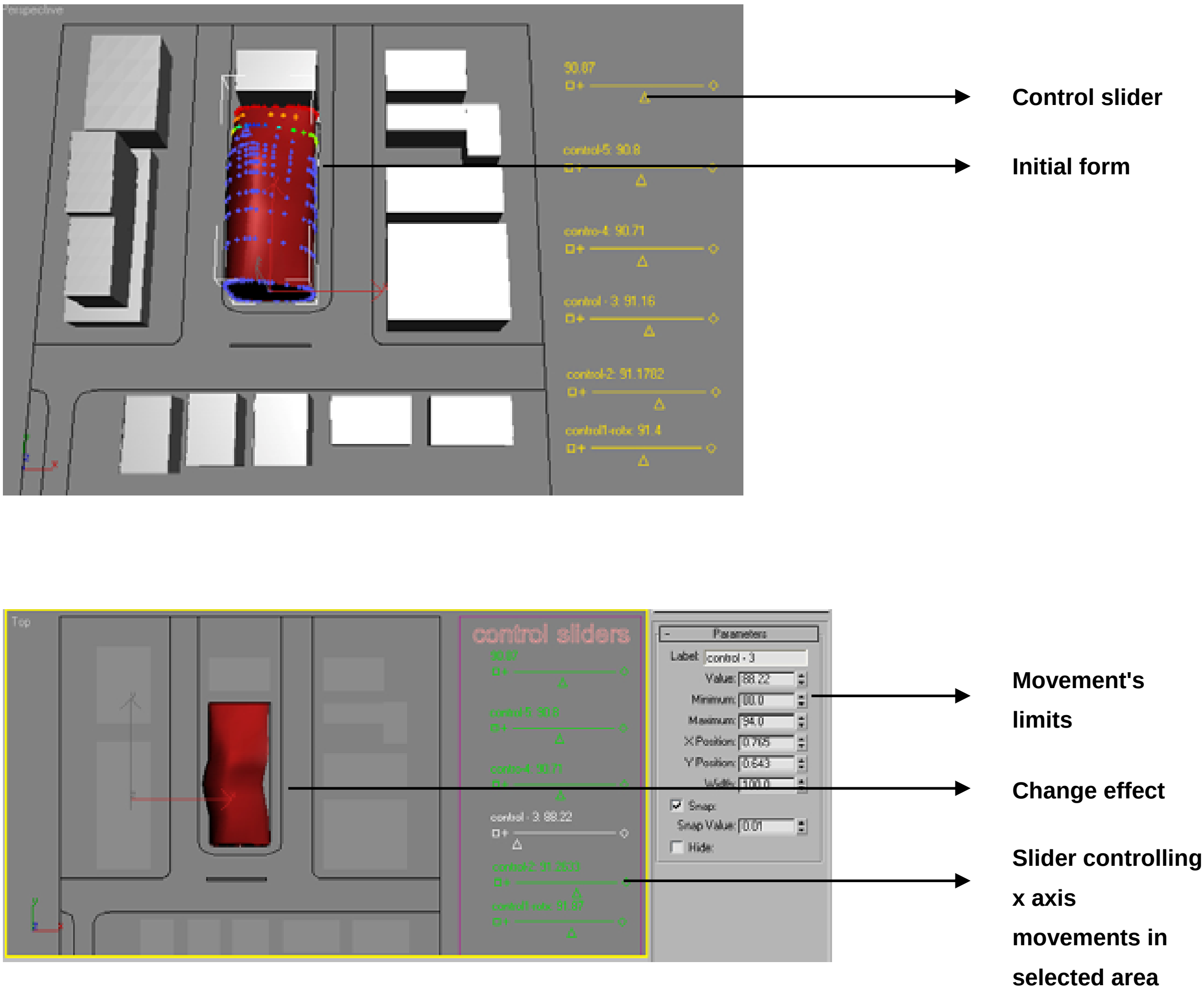


Figure 52 – initial approach layout

10.3 Working directly with the performance envelopes (Approach 2.0)

The idea beyond the second approach was to use a morphing algorithm that is capable of negotiating different surfaces. The morphing algorithm that was used works on forms with a similar number of vertexes and calculates the average point in space between any two points in a given number of surfaces according to the level of influence (defined by controllers) of each surface, which is defined by the designer. Therefore, as opposed to the previous approach, where the designer had to choose an initial form and modify it using the performance envelopes, in this approach the initial form is generated directly by the performance envelopes. The initial interface for this approach allowed negotiation of up to 20 envelopes, which can be used individually to define a threshold condition. Nevertheless, it was clear that finding a solution space for more than 3-4 envelopes would be difficult in terms of control. Moreover, it is not likely that more than 3-4 performances would be defined simultaneously as the main focus of a building's programmatic needs in a real design scenario. Therefore this approach was developed and examined using three performance envelopes: solar rights, solar catch and wind. In this set-up, each envelope was given one controller that controls the deformation of the initial form on a 0-100 scale (not important – very important). In order to negotiate more than one envelope, the total value of the controllers of all the envelopes has to equal 100. The different level of each controller defines the specific envelope's influence on the solution form. Figure 53 shows an example of the interface in a set-up of three envelopes that are given an equal level of influence (100/3).

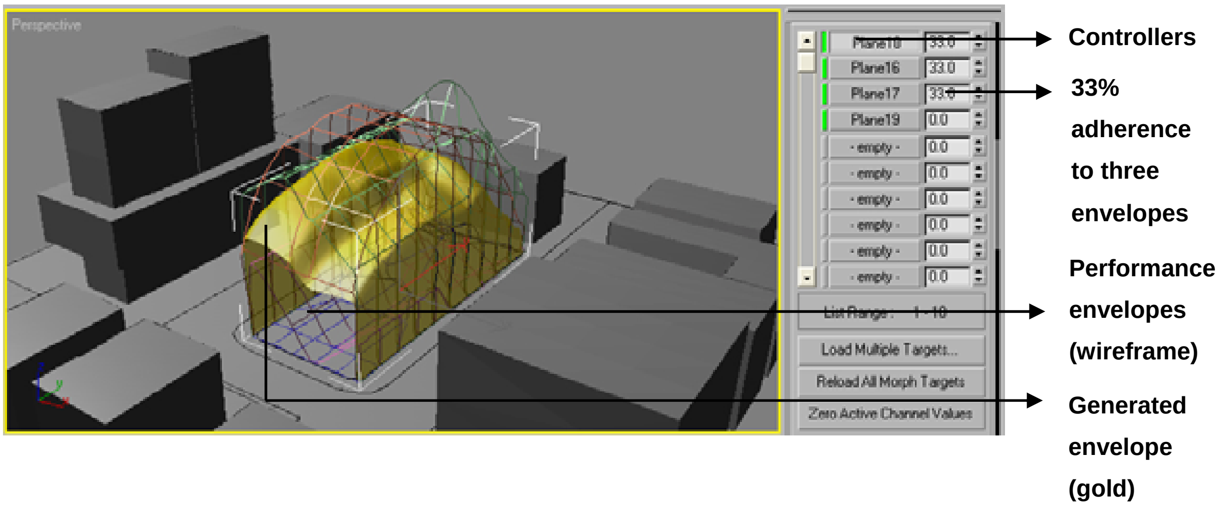


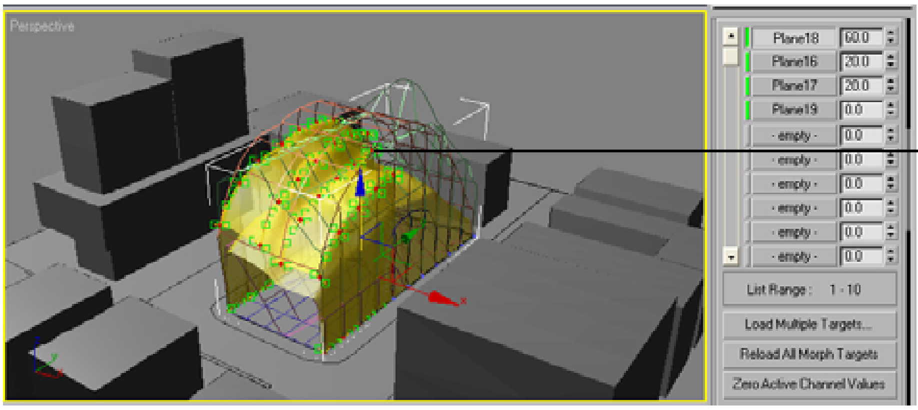
Figure 53 – using three performance envelopes in an equal adherence scenario

Another advantage of this approach is the ability to define local areas of influence. In the previous approach, the building's initial form was treated as single form/surface. In a real-life design scenario, different areas of the initial form would have different performance needs. For example, the front façade, where the main entrance is located, should be treated differently in terms of its durability against wind forces than the rear façade (see Figure 46 for possible division into different programmatic areas in a building site). To facilitate this sort of differentiation, the morphing modifier was introduced also to the sublevel of the generated form. Thus, the designer is able to define the areas to which a certain performance scenario should apply. An initial model is presented in Figure 54.

The main difficulties found so far for this approach are:

a. The generation process is limited to using the envelopes as thresholds (when using more than one envelope).

b. The only possible evaluation method available to the designer, aside from the data embedded in the performance envelopes, is visual.



Only the selected (local) parts the form are modified according to the controllers

Figure 54 – model for local scenarios in generated envelopes

To overcome these difficulties, it was decided to improve the model in two directions. The first is to allow for the use of two envelopes per performance, which defines a range, as opposed to the previous limit of relying on a threshold. The second is to add a real-time presentation of additional fitness criteria for the generated form that will allow the designer to evaluate the form using fitness criteria that are not necessarily related to the performance envelopes but can influence the designer’s preference regarding the chosen result.

Therefore the following features were added:

a. A new control panel/user interface

b. External fitness criteria - a secondary layer of information on the generated form.

c. The ability to define a range by using a second generative boundary performance envelope for each performance type.

The suggested interface (see Figure 55) introduced two main changes to the generation process: Addition of external fitness criteria; floor division and area calculation. This enables in addition to the previous qualitative evaluation a quantitative assessment of the alternatives.

External fitness criteria – Based on the definition in section 6.6 regarding types of fitness criteria, the generated form in the suggested process can be evaluated using 3 types of fitness criteria; a. the first type examines the way the form adheres to the performances used to generate it. In this stage this assessment is done visually by the designer. The starting point of the suggested generation process offers adherence to the performance used as envelopes. However, in order to determine the exact compliance with the requirements an external simulation tool has to be used. b. The second type is adherence to qualitative performances that are connected to perceptual and cognitive aspects. This type has also to be evaluated visually by the designer. c. The third type has to do with information on the geometry and dimensions of the generated form. This type of information could have influence on the designer preferences regarding the general evaluation/preference of the form as it can affect the building cost and adherence to the brief.

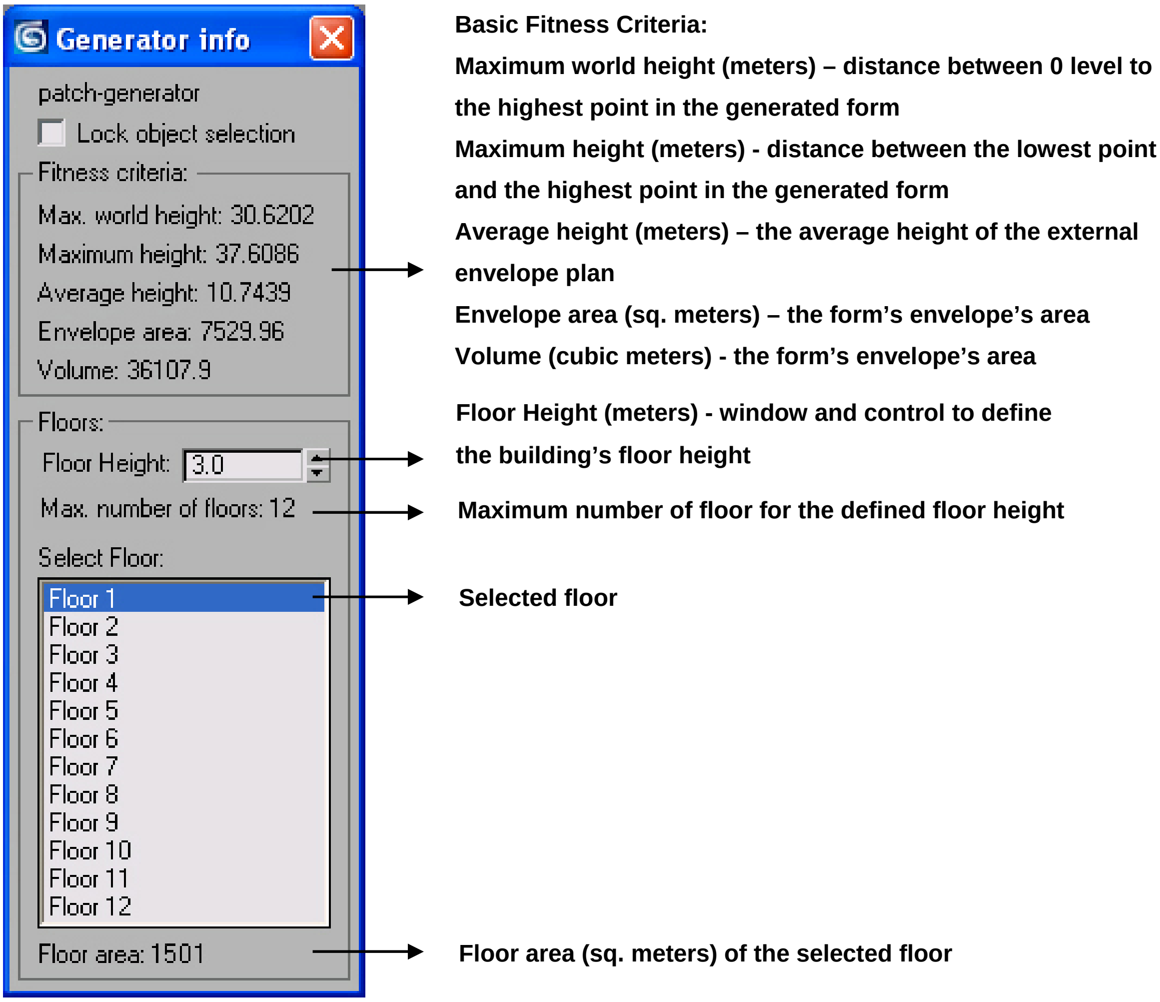


Figure 55 – control panel

In order to examine the applicability of the model for the last type of assessment, six fitness criteria were used in order to evaluate the achievements of each generated alternative: maximum world height (the distance from the lowest to the highest point) maximum height (From the building's 0 level), average height, envelope's area, envelope's volume and total floor area. Nonetheless, it is clear that the complexity and diversity of design problems will require the development of additional fitness criteria. Since every design problem is unique the interface for the evaluation of design alternatives has to be flexible enough in terms of choosing the various fitness criteria, selecting the more relevant for each situation.[64]

[64] The starting point of the suggested generation process offers adherence to the performance envelopes. However, up to date, an external simulation tool must be used to determine the exact value of the adherence to each performance. A future development of the model should consider embedding simulation module.

Floor division and area calculation – Another criterion which is examined is the floors division and floor area calculation. Following a designer's decision regarding the height of a typical floor an area value for each floor and a total expected floor area are calculated and displayed. Also, according to the defined floor height the maximum possible number of floors is shown. The information on the expected/maximum floor area of a building is one of the leading considerations in developing design alternatives since it has large effect on the building cost and revenue.

The model generates the geometry of the different floors, which can be used later on for future development and (see Figure 56).

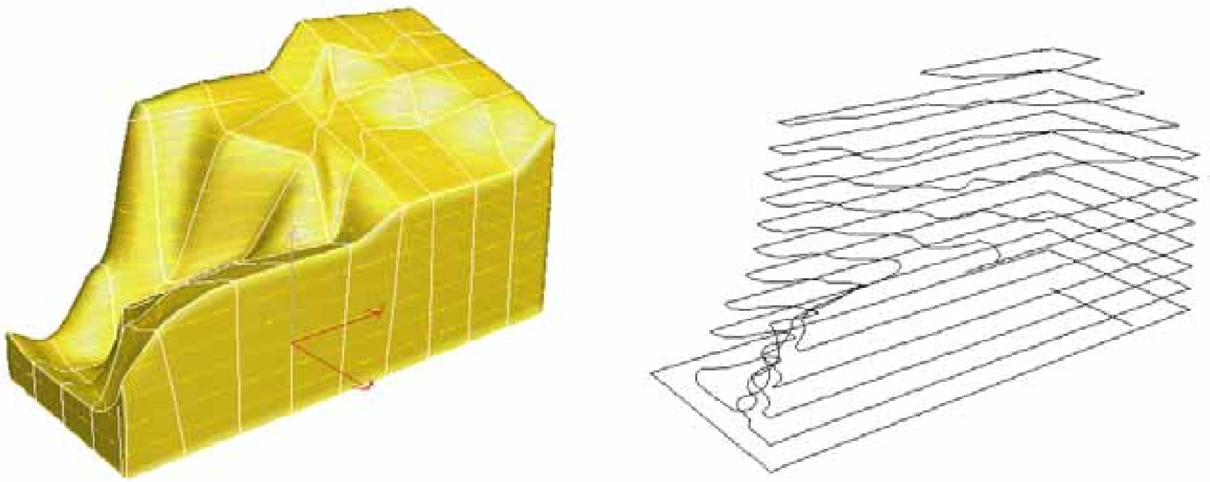


Figure 56 – division into floors according to the designer's decision on floor height

Defining a range condition for the negotiation process or introducing a second generative boundary performance envelope – In the previous stage, each performance envelope defined a threshold condition. At least two envelopes were needed to define a range condition. In many cases, the required performance is defined by two boundary conditions (low and high range) instead of only one (threshold). Figure 57 shows an example of a range (solution space) for form generation created by two wind performance envelopes (2m/s and 5m/s). Every point within the volume between these two envelopes has an initial[65] condition of wind between 2m/s and 5m/s that can represent a desired comfort range (see appendix c).

[65] We define the condition as "initial" because the exact wind conditions will change along with the development of the design. Future generative applications should include embedded simulation modules, which will facilitate online performance simulation. This research concentrates on the initial form/solution space and therefore presents a method of using performance envelopes with or without such capabilities.

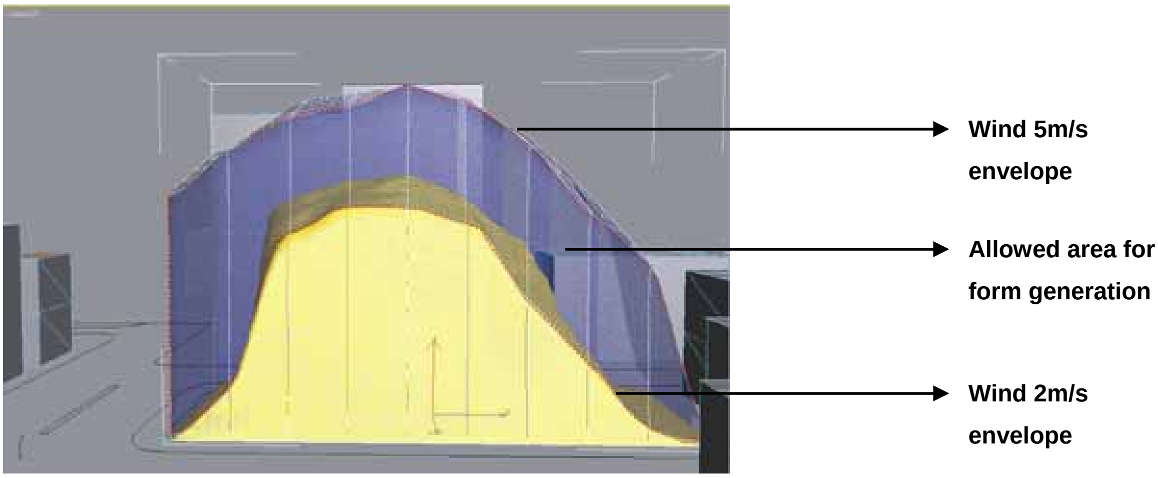


Figure 57 – solution space (range) generated by two-threshold wind performance envelopes

An examination of the approach at this stage in an architectural design scenario revealed two main drawbacks:

The linearity of the design process – The process is linear because during the generation phase the designer receives information only on the single form he sees on the computer screen. Therefore, comparing different alternatives is difficult.

Differential evaluation – The level of importance of different fitness criteria in a design changes according to the different architectural briefs and the designer's preferences. Moreover, in some cases, several different combinations of the fitness criteria's importance can be valid. Therefore, in each project the designer has to decide about the importance value given to each fitness criteria. The approaches discussed so far, however, are capable only of presenting the fitness data and do not allow for defining combinations of fitness criteria.

10.4 Non linear form generation approach (Approach 3.0)

To overcome these drawbacks, the linear design process was developed further into a nonlinear one. This shift suggests moving to a computer-based design exploration that uses the computer to generate design alternatives as opposed to the single-solution approach[66] discussed above.

In this type of design process, the designer defines the boundaries in terms of the performance envelopes and the number of alternatives to be generated. Then, the computer generates design alternatives according to these constraints and presents the results in a visual catalogue. In order to define the parameters of the examined performance envelopes, this approach introduces channels through which the designer can narrow the alternative generation process to discrete sections of the entire solution space. It is anticipated that at

[66] A discussion on other advantages of design exploration can be found in Woodbury and Burrow, 2006.

the begining of the generation process, when the solution space is wide, the channels' entire range (0-100%) will be used. In the later stages or when fine-tuning is needed, the range definition of each channel can be narrowed to generate alternatives only in specific areas of the full range.

Another control feature introduced at this stage is the number of alternatives. Defining a large number of alternatives will help the designer to evaluate the solutions when there is a big difference between the alternatives. This situation is likely to occur at the beginning of the generation process or when the solution volume is very large (Figure 59 presents the enhanced control interface).

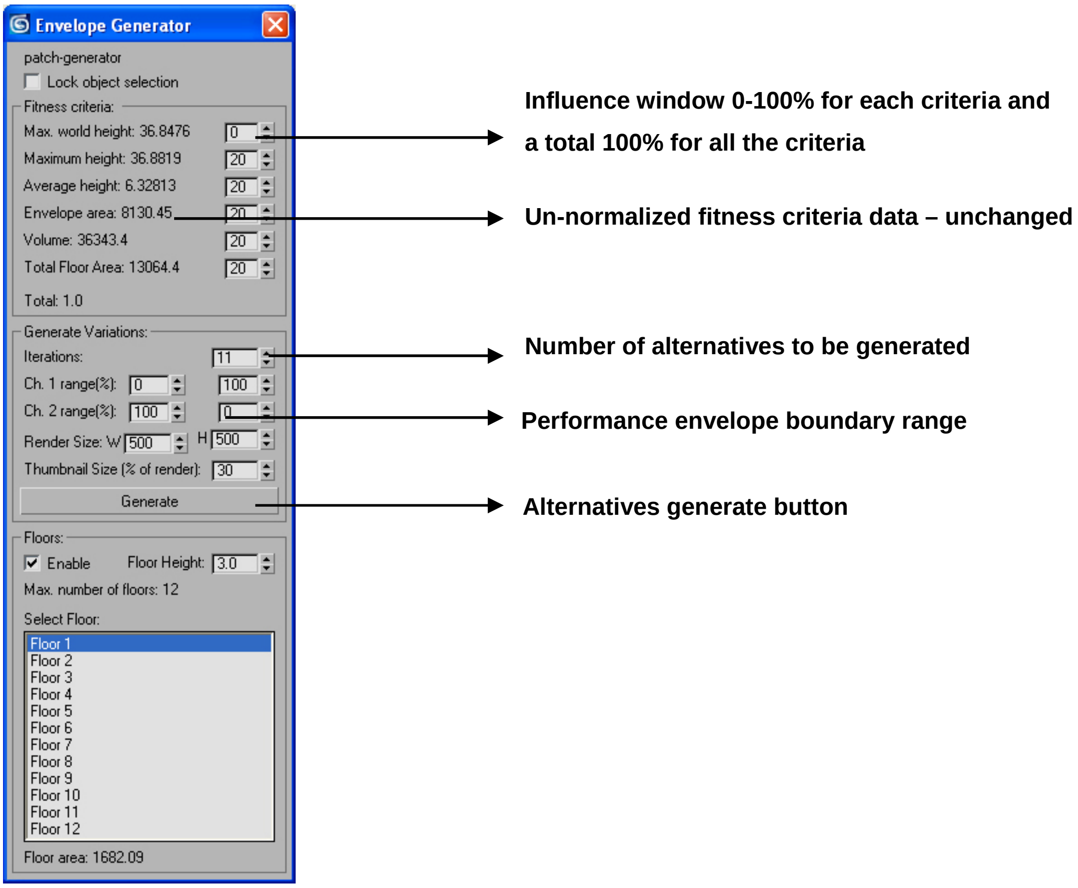


Figure 58 – enhanced control interface

The second improvement introduced in this approach is a grading algorithm. The importance of each of the fitness criteria that were presented earlier and the new total area value are controlled by an importance value which is defined by the designer using an influence window interface. In the influence window, the designer can assign a value between 0-100% that represents the influence of this specific fitness criterion on a total grade value. The sum of all the influence values cannot surpass 100%. A total grade value is then calculated normalizing all the criteria data to 0-1 scale. A grade is calculated for every single criterion and a total grade is given for every alternative. The results of the generation and grading processes are presented to the designer in a visual interactive catalogue in which the alternative with the best grade sits at the top of the chart and surrounded by a red frame. In addition to the values of the criteria mentioned earlier, the catalogue also presents numeric data on the influence of each channel that was used in every generated alternative (see Figure 59).

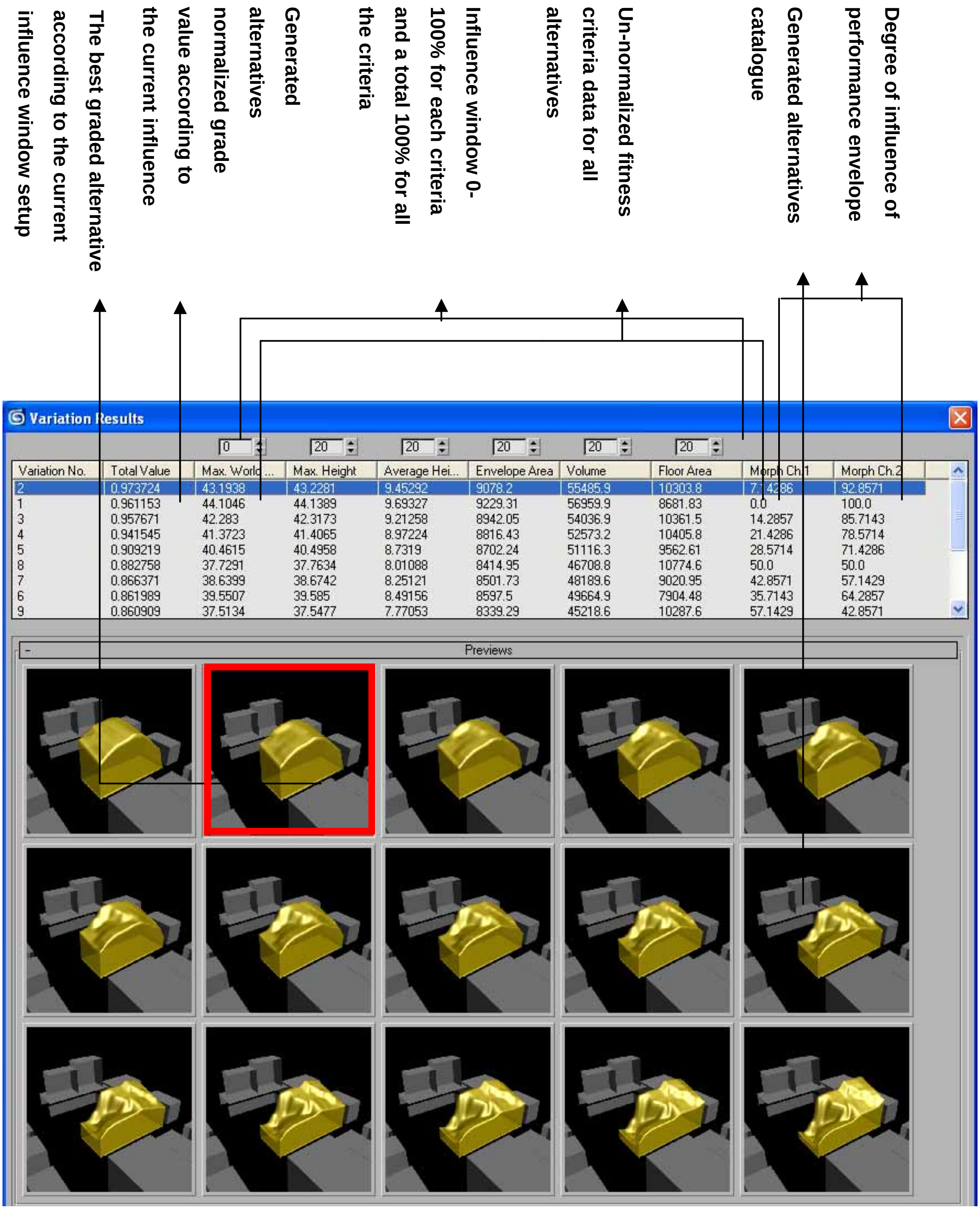


Figure 59 – visual catalogue of generated forms

To facilitate an interactive examination of various grading scenarios, the interface also allows changing the level of influence of every criterion after the initial generation of alternatives. This can change the grading of the various alternatives and may change the alternative that receives the highest score, thus leading the designer to modify his decision about the selected solution. Figure 60 shows an example of a change in the level of importance from equality among all the criteria (20% in Figure 59) into a situation of 70% for floor area criterion and 30% for the envelope's area criterion. This change the best graded alternative (No. 2) to alternative No. 8 as presented in Figure 60.

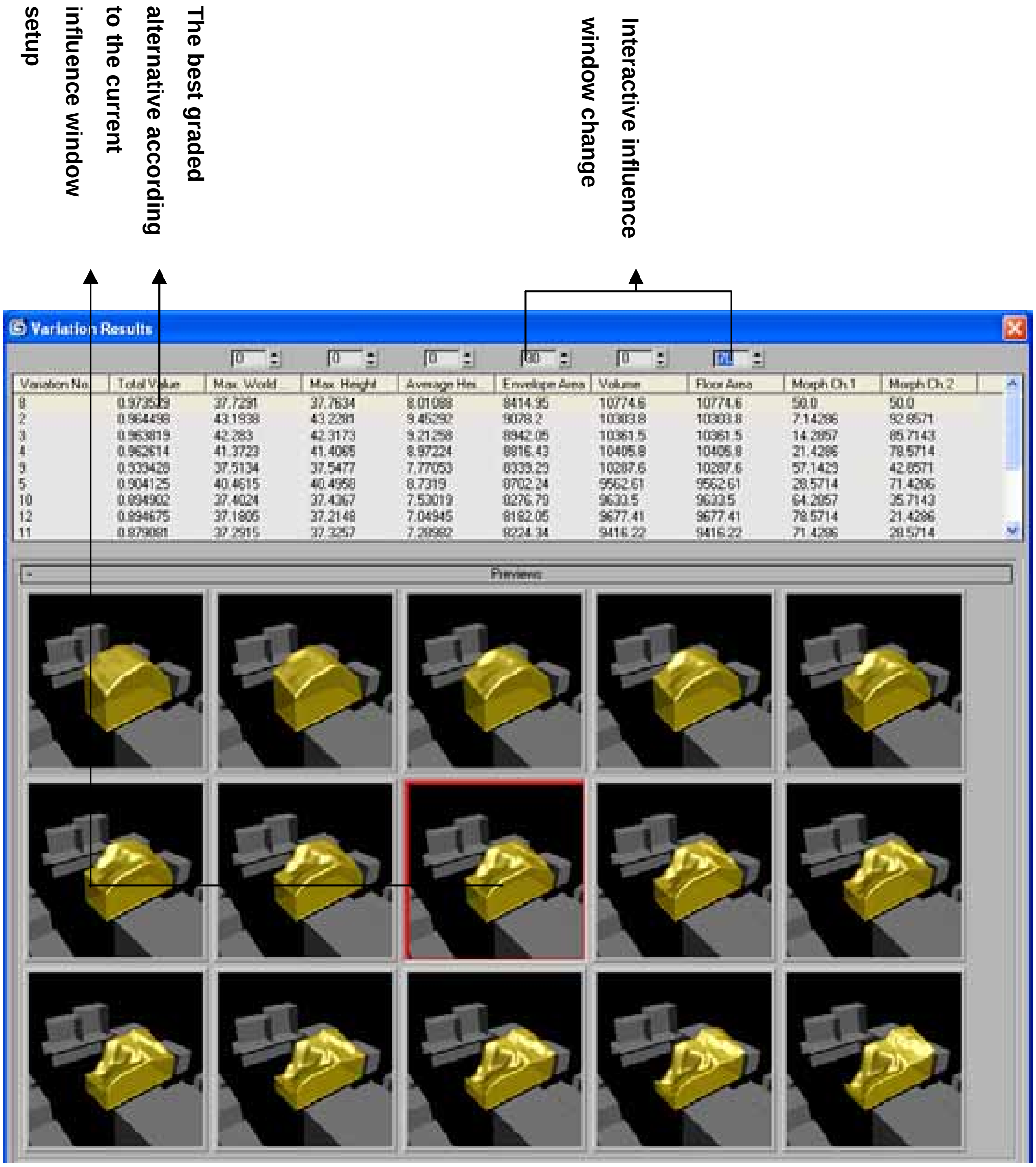


Figure 60 – interactive grading algorithm

In order to examine the applicability of the model, it was decided to develop only six basic fitness criteria in the grading algorithm. A possible future direction might involve developing a reservoir of fitness criteria algorithms from which the designer will be able to choose various combinations of criteria to be used in the specific design problem and design stage. An example of a possible interface is shown in Figure 61[67].

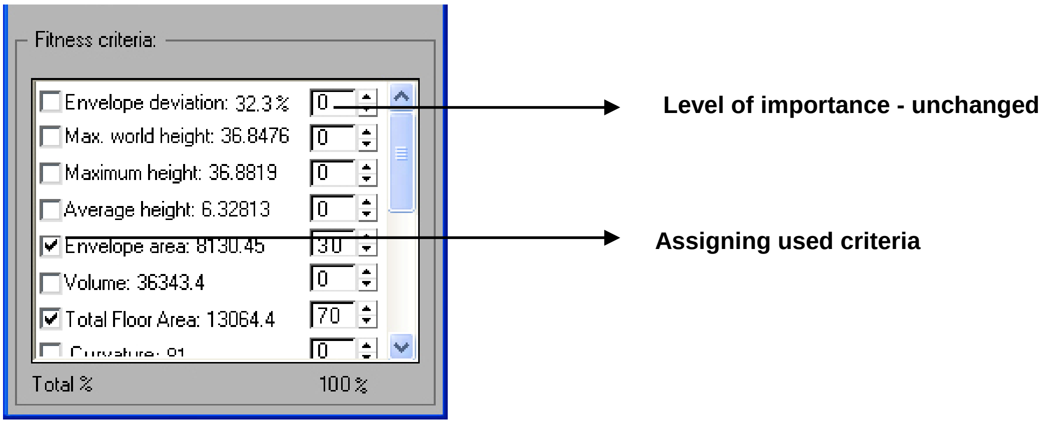


Figure 61 – suggested future fitness criteria interface

The following directions are defined as worth pursuing in future development of fitness criteria algorithms in this type of design process:

a. Degree of deviation from a preferred performance envelope – The generated design alternative uses one or many performance envelopes in the form generation process. Since the generated form is likely to be an optimization of several envelopes, it can be important to know the degree of deviation from each envelope. The degree of deviation can be presented as the average of the difference between the location of the used envelopes and the generated form. It is possible to present the deviation for one envelope or for more (as a calculated weighted average value). The deviation can be calculated as an average for the entire generated form or for local/pre-defined areas within the envelope.

b. Envelope compound curvature – It is anticipated that this design process will be used to generate a solution space/initial building form. Nonetheless, the generated envelope could be perceived as the building form. In such cases, examining the degree of curvature of all or part of the envelope is important when it comes to the choice of material and cost[68]. Curvature analysis is already embedded in some commercial software (see example in Figure 62)[69].

[67] See also another possible direction in developing the grading algorithm in Gololov, Yezioro. 2007

[68] For a practical example of the importance of curvature analysis, see Gehry's use of curvature analysis in *Digital Gehry* (Lindsey, 2001).

[69] For example: Rhinoceros, Catia and Pro-Engineer.

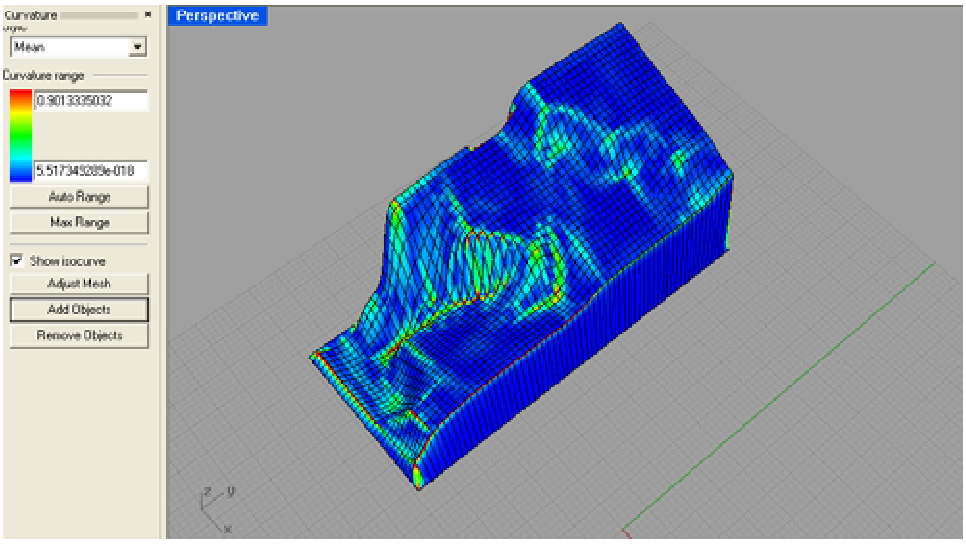


Figure 62 – complex curvature analysis in Rhino 3.0

c. Energy – Energy evaluation is appropriate mainly when the generated alternative is treated as an initial building form and a decision regarding the building's materials has been taken. A basic version of one of the energy simulation software tools described in 8.2 could be used.

d. Construction – Stress/ load evaluation is appropriate mainly when the generated alternative is treated as initial building form and the designer has already decided on building materials.

The following directions are defined as worth pursuing in future development of the model and the software tool:

a. Embedding simulation modules in design software/Interactive examination of performance envelopes – The developed design process uses generated performance envelopes from simulation software that is not currently embedded in software used by designers. Therefore, the generated form must be examined using the same simulation software in order to verify that the performance envelopes used in the initial process are still valid. It is our belief that future design software will embed various simulation modules that will allow the calculation of performance envelopes in real time.

b. Number of performance envelopes – For the purpose of proving the concept, the design interface allowed using maximum of four performance envelopes (two per channel) at any given stage. Future approaches should not be limited in the number of envelopes available for use.

c. Recording the process parameters – The proposed design method is based on a parametric process. Thus, it is possible to go back at any time to reexamine and modify any choice that was made during the design process. However, the current tool supports only the basic "Undo" command and does not support recording all the parameters that were used during the process. This means the designer has to memorize or write down the parameters that were used during the generation process. A future development of the proposed approach should be able to record all the parameters that were used or changed during the generation process and order in which they were used.

10.5 Approach analysis summary

The following table summarizes the various stages in the development of the main parameters and the capabilities of the proposed model.

Approach No.	Visual control/Cloud of points	1	2	3	Future approach
The initial form is generated by computer	**-**	_	**+**	**+**	**+**
Number of possible performance envelopes	**Unlimited (more than 3-4 envelopes on the design area can be difficult to handle and may disturb the design process)**	**2 per control slider**	**2-4**	**4 (two per channel)**	**Unlimited**
Use of one boundary envelope	**+**	**+**	**+**	**+**	**+**
Possibility of using more than one boundary envelope	**+**	_	_	**+**	**+**
Number of fitness criteria	**-** **(visual)**	**-** **(visual)**	**-** **(visual)**	**6**	**Unlimited**
Possibility of changing the influence ratio of fitness criteria	**-**	**-**	**-**	**+**	**+**
Generation of design alternatives	**-**	**-**	**-**	**+**	**+**
Local area generation	**-**	**+** **(local only)**	**+**	**+**	**+**
Embedded simulation module	**-**	**-**	**-**	**-**	**+**

Table 3 – approaches analysis summary

The table compares the developed approaches in the areas of greatest significance and also presents the initial thinking regarding a possible future approach.

10.6 Closing remarks

This chapter presents the development process of the GenPOD model and software tool. It delineates the considerations that informed the process and their implications for the expected design process. It also discusses directions worth pursuing in any future development of the model and the software tool. The next chapter will examine the application of the developed model and tool in a design case study.

11. Case study - Generation of building's initial form using performance envelopes – Office building in Mugrabi Square, Tel – Aviv

11.1 Introduction

The case study's main aim is to examine the GenPod model which uses several empirical performance envelopes' data to generate an initial design form or a solution space. The suggested model facilitates designers with a numeric parametric control of environmental and programmatic fitness criteria in the initial form generation process. It also provides designers a higher level of control over the design's form by allowing immediate visualization of changes and their causes in different ratios of both single and multiple fitness criteria scenarios.

The model is expected to be suitable for use in both urban and building scales. In an urban scale the model is expected to generate initial guiding 3-D model based on performance envelopes which could be used as an initial solution space for an urban plan. In a building scale the model is expected to generate an initial building form or a performance oriented solution space that could be used as a spatial boundary for future design development. The scope of this case study focused on the building scale. This scale was chosen for this case study for two main reasons; the first one was that it seems that designers use more often computers during the initial stages of the design in building design rather than in urban design process. For that reason, the potential of designers to use the proposed design method and tool seemed higher in this scale. The second reason had to do with the number of parameters and the complexity of the performance envelopes needed for an urban plan model. It was expected that the complexity of the computer model in the urban scale would surpass the resources and scope of this research. Nevertheless, it is expected that insights from the building scale will be valid to a certain extent also in the urban scale.

This chapter will describe in details a possible design process that uses the suggested GenPOD model and tool, to design initial form/solution space for an office building in Tel Aviv.

11.2 Urban Context

The site is an empty plot located in Alenbi Street, which is one of the main commercial and business streets in the city of Tel Aviv (Lat=32.5 deg. N, Long=35 deg. E) (see Figure **63**). The site is three blocks east from the Mediterranean Sea. The surrounding buildings are: high-rise building (17 floor) on the west side, residential buildings (4-5 floors) on the north and north-east sides and commercial buildings (4-7 floors) on the south-east and south sides.

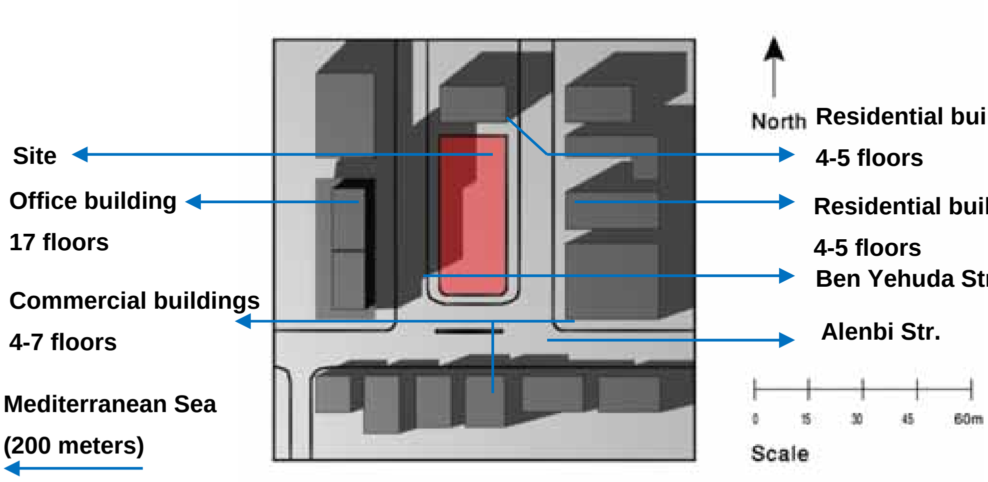


Figure 63 – case study site

11.3 Building's brief

The following constraints were defined for the building's brief:

a. The building height will not be higher than 35 meters (about 10 floors) - Based on the city regulations for the area.

b. The building will be designed as an office building for large and medium size companies. The floor height will be 3.5 meters.

c. Climatic considerations - the building will add minimal shade on the commercial and residential buildings around the site in winter time, being it the critical period in terms of solar rights and sun radiation in Tel Aviv. Therefore the solar rights envelopes were defined between November and February from 10:00 am to 14:00 pm.

The building form and position will allow maximum natural ventilation in the open public areas within the limits of comfort during the main working hours. Specific attention should be given to the building’s entrance where wind velocity should be minimized - no more than 2 m/s desirable for thermal comfort reasons (for reference about wind effects and comfort scale limits see BRE, 1994 in appendix c).

d. The building’s form should offer maximum floor area in order to gain the highest renting revenue.

These outlines define the contextual constraints for the generation process. The specific definitions are subject to change according to the buildings location, brief and the designer’s preferences. Since GenPOD is a parametric model it supports the option to change these definitions and examine the results of different scenarios.

11.4 Performance envelopes used in the generation process

As discussed earlier in chapter 8 up to date software generally used by architects do not include simulation modules nor is it common for architects to use simulation software during the initial design stages. It is recommended by this research, however, that future architectural software should include simulation modules to be used by designers in the design process. In order to simulate this design approach we used in this case study separate software for creating the performance envelopes and other for the form generation process.

Three types of performance envelope were initially considered for the case study:

Solar envelopes - both solar rights and solar catch envelopes were imported from a simulations done in SUSTARC[70]. In the early stages of the envelope generation process in SUSTARC the designer defines the time period (months and hours) for which the envelopes are calculated. The time frame used in the case study brief included winter periods which was defined between November to February and 10:00am to 14:00 pm. This period correlates with hours of maximum sun radiation for thermal comfort in open spaces and passive solar heating of buildings, and with the main working hours in the most problematic period in winter in terms of blocking the direct sun light to the surrounding buildings. The envelopes were exported to 3ds Max. Figure 64 shows an example of the interface and the generated envelops in SUSTARC. Figure 65 present an example of the same envelopes imported in 3ds Max environment.

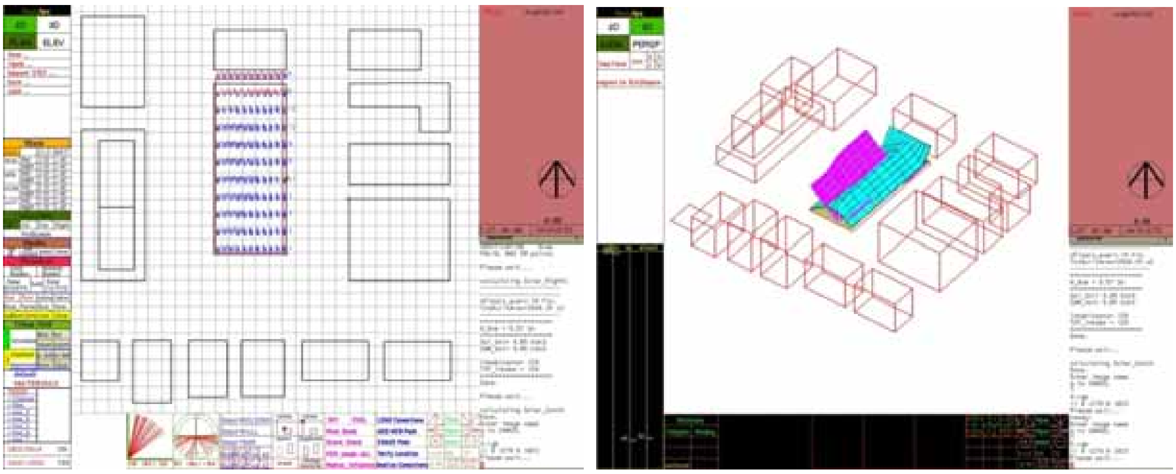


Figure 64 – SUSTARC solar rights and solar catch envelopes

[70] Capeluto, I. G. 1996. And Capeluto, I. G. Shaviv, E. 1997

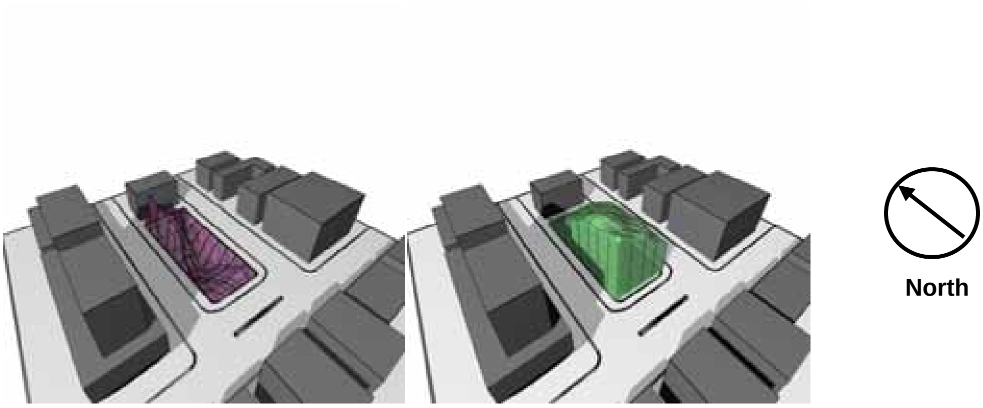


Figure 65 – solar rights and solar catch envelopes site boundary

Wind envelopes – Wind envelopes were generated in ENVI-met 3.0. The climatic Input data for the simulation software was taken from the Climatic Atlas of Israel (Bitan, Rubin.1991). Five envelopes (1-5m/s envelopes) were generated by employing the typical wind scenarios in the site for both summer during winter. The influence of wind during summer period was treated according the Building Research Establishment (BRE) definitions described in the appendix D. According to these definitions, in relation to people feeling of comfort in exposure to wind in summer, 4m/s wind (B4(Beaufort number): 4.1-5.9m/s) is recommended for short exposures. It is just below the agreeable wind speed limit which is less than 6-8m/s (B5). It also allows open windows in a building without the risk of damage to the windows and too strong wind in the rooms. Since it is on the limit of comfort it is also the maximum wind speed that can be used for natural ventilation.

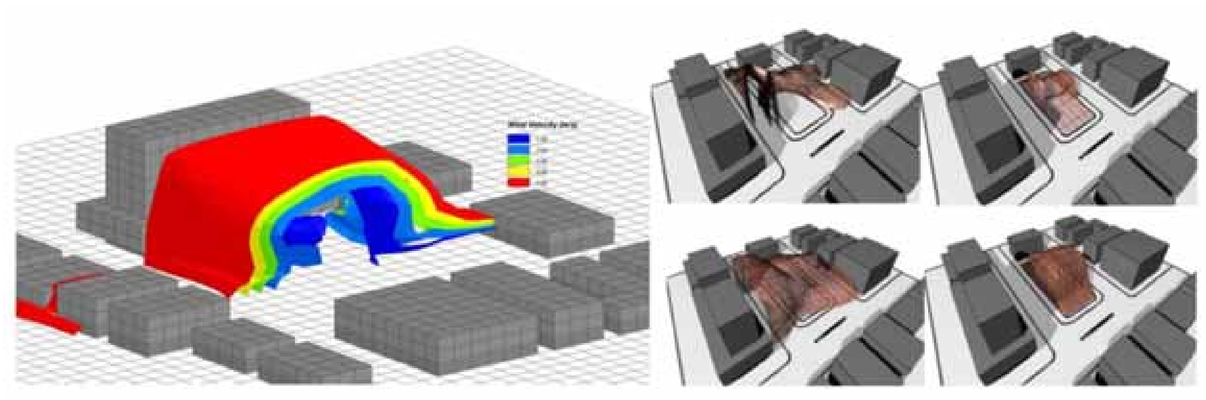


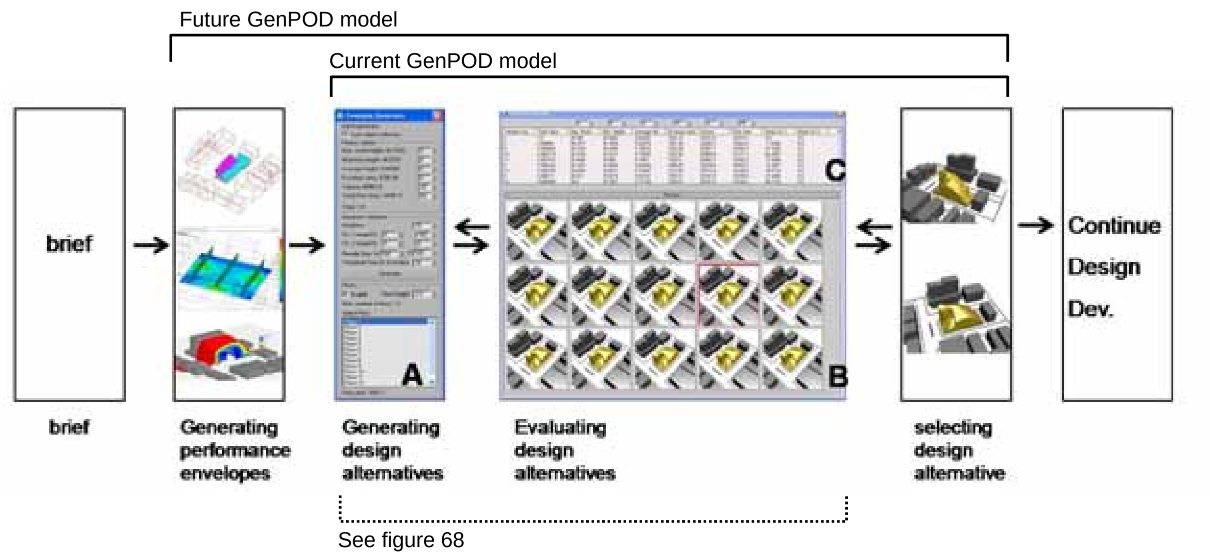
Figure 66 – ENVI-met wind envelopes simulation (left), full envelope and cropped to site boundary envelope of 1m/s and 2 m/s envelopes (right)

Higher wind speed, which could allow better ventilation, can be considered for spaces that are not inhabited. In places that are designed for longer exposure to wind as balconies, roof terraces and entrances a light breeze (B2: 1.2-2.5 m/s) or light air (B1: 0.22-1.1 m/s) is appropriate. In winter, according to the BRE, at lower temperatures than 10˚C, the relative comfort level is reduced by one Beaufort number for a 20oC fall in temperature – a condition which is not likely to occur in Tel-Aviv. Therefore, we related to 4m/s wind performance envelope as the upper threshold and 1m/s, 2m/s wind performance envelopes as lower threshold in the case study. Following the generation the envelopes were exported to 3ds Max.

Figure 66 shows and example of both the output of the generated envelopes and the imported envelopes.

11.5 Design method

The design method used in the case study follows the method described in Diagram 6 (Chapter 9). It uses performance envelopes generated by different software to generate an initial building form/design space. A schematic view of the design process is described in Figure 67. The suggested design process starts with an interpretation of an architectural brief as performance envelopes. Then, design alternatives are generated, presented (in a visual catalogue – see B in Figure 67) and evaluated in a repetitive parametric process in which the chosen alternatives are evaluated and fine tuned (see B and C in Figure 67). The process ends when the designer decides that one or more alternatives adhere to the criteria defined by the brief.



A - GenPOD user interface – defining the performance envelopes and fitness criteria parameters

B - Visual catalogue of the generated alternatives

C - Fitness criteria values/grades for further evaluation and classification of the generated alternatives

Figure 67 – schematic view of the case study's design process

The design process can be divided to four main stages;

11.5.1 Preliminary generation (Stage 1)

In this stage the initial performance envelopes are generated using data from context and programmatic demands. After verifying that the generated performance envelopes have common grounds (see section 9.2) an initial set up is performed. The data needed in this stage consist of basic fitness criteria, the specific range of the each morphing channel, the height of each floor and the number of alternatives that the generation process should generate.

Setting the Initial fitness criteria parameters[71]: The developed tool offers numerous fitness criteria possibilities. Each alternative receives a score for each fitness criteria and a total score value is calculated according to the initial setup (in percentage) of influence of the selected criteria. The generation results of total score value is arranged from the highest to the lowest score (from 0-1) in the variation window. Nevertheless, fitness criteria settings parameters can be changed during the entire working process making it possible to examine different generated results while checking various scenarios.

The number of alternatives: set according to needed accuracy of the results. A low number will usually generate solutions which are very different while a high number will diminish the differences between two successive alternatives. Using high number is recommended when there is a significant difference between the negotiated envelopes or when higher precision is needed.

Range of the morph channel: the difference between any two envelopes (range) is defined as 100%. The range channel is used when the designer wants to narrow the generation of the alternative to part of the entire range. It is used mainly for fine tuning the generated solution after the generation and evaluation of the initial generation (see also the explanation in chapter 10).

Height of floors: the decision on the building floor's height determines the amount of floors the building would have and the maximum achievable floor area. After setting up the above data the designer has to initiate the initial alternative generation process.

11.5.2 Evaluation (Stage 2)

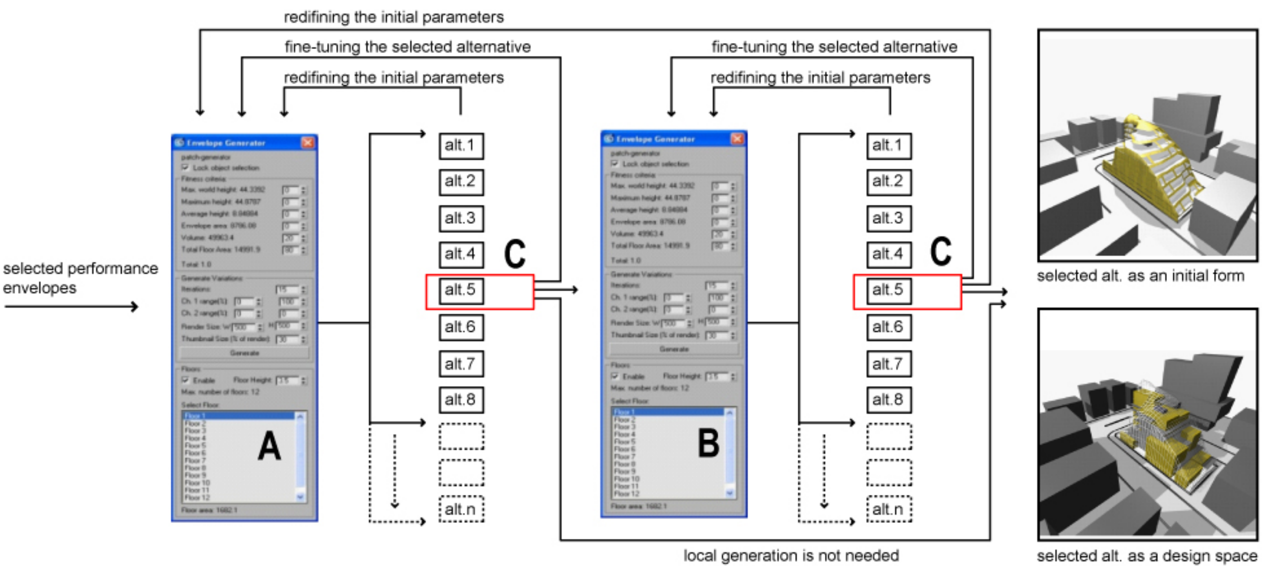
After the *initial generation* the most appropriate solution among the generated alternatives should be selected by employing single fitness criterion or any combination of the existing

[71] For general discussion on fitness criteria and specific discussion of fitness criteria in the new tool see chapters 9 and 10

criteria. In case of not finding a suitable solution in the *initial generation* the initial setup should be changed. If there is still no appropriate solution after several rounds of generation the performance requirements that were used to generate the alternatives or the fitness criteria that were employed in the evaluation need to be reconsidered. (see Diagram 6 in chapter 9).

11.5.3 Secondary (local) generation/optimization (Stage 3)

After choosing a preliminary solution a secondary, fine tuning, optimization process is performed. This process can be performed on the entire form or on parts of it (local generation). Also, it can fine-tune the chosen alternative by refining the morphing channel range while employing the same performance envelopes. Moreover, it can refine it by introducing new performance envelopes appropriate for these parts.
Local generation applies on a certain portion of the form the same process that was described in the two previous stages. The generation process ends when a satisfying solution is found. Nevertheless, this design method is open ended - it is possible at any given time during the generation process to go back and forth, erase secondary stages, introduce new performance requirements, change preferences and regenerate new solutions according to the brief (see Figure 68).



A – General generation (modifying the entire form)

B – Local/secondary generation (modifying the selected parts of the form)

C – Evaluation – selecting an alternative using the calculated criteria values/grades and the designer's visual preference

Figure 68 – general and secondary/local form generation

11.5.4 Performance evaluation (Stage 4)

The main aim of this stage is to verify that the performance of the chosen solution complies with the requirements stated for its generation process. It involves a simulating process that assesses the performance of the chosen form in the same conditions that were used to generate the initial performance envelopes. Providing that different simulation tools will be embedded in future versions of the GenPOD model, this stage will be unnecessary since it will be performed simultaneously with the generation process.

11.6 Case study design process and results

The case study presented a possible design process within the building scale based on the design method described in the previous section and the given architectural brief. The detailed case study design process and results are described in Appendix C.

The following main subjects are examined and demonstrated in the case study:

a. The relationship between the brief's demands and the performances/performance envelopes used in the design process – The suggested model is intended for mainly projects with a performance-oriented brief in which the performance could be translated into performance envelopes (see also the discussion on types of envelopes in Chapter 9). The initial step in using the model involves examining the common ground that the various selected envelopes may share, and possible contradictions that might arise. The examination defines the strategy of using the model on the selected envelopes. The simplest examination-result scenario takes place when the selected envelopes share common ground in all the envelopes' volume. In this scenario, the model can be applied on the entire form.

However, in many possible scenarios (as in the case study's example), not all the envelopes have common ground (see Figure 76). In such cases, the designer has two main possibilities: the first is to perform, at a certain moment during the design process, a local/secondary generation in the areas that deviate from the common ground using different performance definitions. This possibility is shown in the case study's detailed example in Appendix C. The second possibility involves a different approach to using performance envelopes. While the main approach emphasizes working within the range or common ground created by the performance envelopes, the second possible approach calls for working both inside and outside of the common ground. The logic behind this rests on the idea that insofar as the designer is aware of the performance level that the results achieve, he can deviate from the initial common ground to respond or adhere to other parameters that were found more important at that stage.

b. The possibility of using various performance envelopes and influencing/modifying parts of the designed form as opposed to modifying the entire form - The case study demonstrated the possibility of using various performance envelopes in a single design process to modify the entire form or only parts of the form. The initial building form can be generated within the range that is defined by two or more different types of performance envelopes. The case study demonstrated the generation of initial form by using a range created by a single solar rights envelope and a range from two wind envelopes (see Section 1 in Appendix C). Changing performance envelopes and working on parts of the form within the design process is necessary mainly for adjusting the performance of areas within the form that have special demands in terms of performance. This possibility is exemplified in the case study in two different parts: the first is use of a second 1m/s wind performance envelope for the designated entrance area of the designed building while applying a 2-4m/s wind performance envelope (which is needed for ventilation), for the generation of the entire building's form (see Section 1.1 in Appendix C). The second is the use of a single solar rights envelope to modify the form in an area that deviates from the common ground (see Section 1.1.1 in Appendix C)

c. The implications of having numerous generated design alternatives to work with during the design process – The case study presents a design process in which numerous design alternatives are generated in each iteration. The number of alternatives to be generated can be determined by the designer according to the expected variability between the alternatives; in the case study, 15 design alternatives in each run was found to be satisfactory.

The possibility of working with numerous design alternatives in a nonlinear mode, as opposed to the traditional linear approach of developing and examining one alternative at a time, allows the designer to explore a greater number of alternatives and directions within the limited time frame of the design. Moreover, the fact that all the alternatives are simultaneously evaluated by the grading algorithm and visually by the designer helps to narrow the solution space and reach a better solution; in the case study's initial generation run, it was noticed that some of the generated alternatives did not adhere to one of the programmatic demands (maximum number of floors). Due to the large number of alternatives, it was considerably simpler and faster to determine a new range for the next generation, which explored alternatives that adhered to previously problematic programmatic demands.

d. Types of fitness criteria, the possibility of using various combinations of criteria – The case study demonstrated the possibility of combining visual and quantitative criteria. It also demonstrated the possibility of combining different kinds of criteria – those related to the performance envelopes that generated the initial form and other more general "architectural" criteria, such as floor and envelope area, built volume, etc. This is demonstrated in the case study by controlling the parameters of the performance envelopes while being able to view the immediate effect of changes on the other, more general criteria. Moreover, it demonstrated the possibility of examining a solution using different sets of criteria (see Section 1.1 in Appendix C). Also, the fact that this process is parametrically controlled makes it possible to modify at any time during the process the criteria preference (as with any input data in the process), which accordingly could change the end result. All of the abovementioned possibilities, which are incorporated in the suggested model, help to increase the amount of information the designer has about the selected result. Equally important, it also increases the amount of information on all the other generated results that were not selected.

e. Evaluating the performance of the selected result – In order to assess to what extent the generated results adhere to the performance parameters that were defined in the brief and expressed in the performance envelopes, the results were examined using external simulation tools. The detailed examination is described in Section 1.3 in Appendix C. The results confirmed the initial assumption and did not deviate significantly from the performances that were initially defined. The results also emphasized the need for an exact reading of performance levels. Using the current model, the designer can get a good estimate of the level of adherence to each performance parameter by examining the percentage of the solution's adherence to each envelope used in the generation process. The ability to know the exact level of performance for each parameter will be achieved only when a simulation module is embedded in the GenPOD model. This will allow greater accuracy in determining the performance level and thus, a higher level of control over the design by the designer.

f. Types of solutions – The generation process ends when the designer finds a solution that adheres to the brief's demands. There could be a situation in which several solutions adhere to the demands. In this case, the designer can choose from among the following:
1. Choose the solution that receives the best grade. This choice is demonstrated in the case study in Appendix C.
2. Redefine more accurately/in greater detail the brief's demands in order to narrow the range of solutions.
3. Introduce different criteria to the evaluation process.

4. Continue to the next design stage with several alternatives.

The selected alternative can be conceived as an initial form of the building that needs to be further developed in the next design development stage (see description of the architectural design process in Diagram 1, section 6.3) or as a solution space that represent the formal boundaries of a buildings form that will be developed in the following design stages. When conceived as an initial form it is possible to use the generated division to floors which is created in every generation run. Figure 69 shows a possible usage of the selected alternative as an initial building's form.

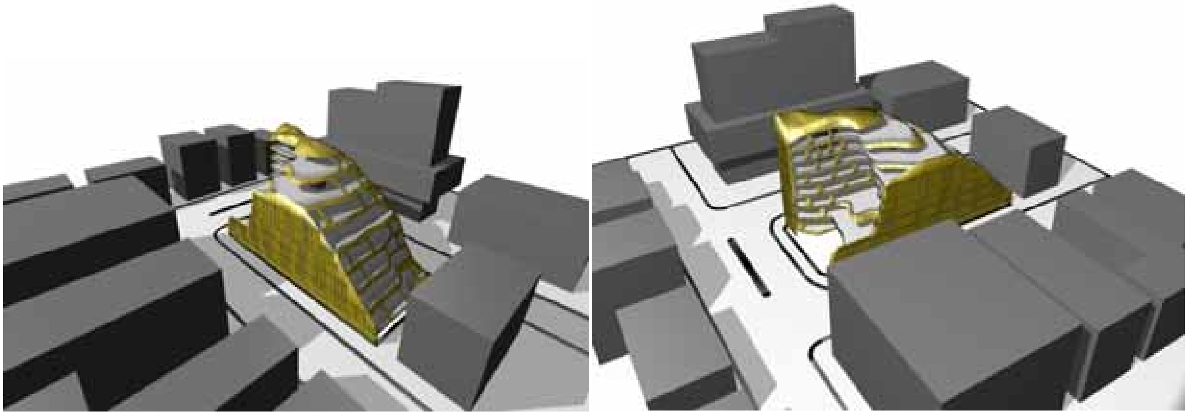


Figure 69 – using the selected alternative as an initial building's form

When conceived as a solution space the selected alternative is used as a volumetric boundary in which the designer has to develop the designed building. Figure 70 describes a possible initial design of a building within the boundary of the selected envelope. Obviously, a combination of both approaches can be followed by the designer as well.

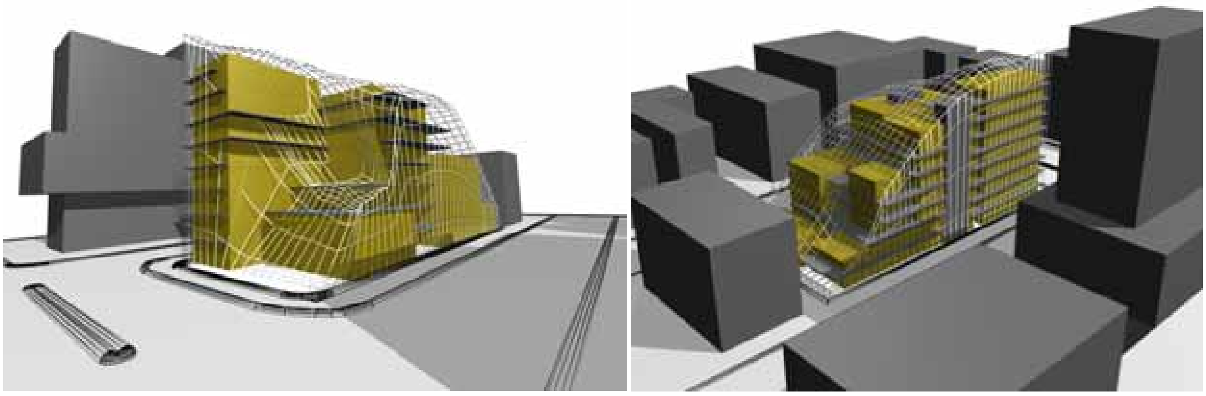


Figure 70 – using the selected alternative as a solution space

11.7 Closing remarks

The case study presented a possible design process within the building scale that that examines the applicability of the suggested method in a design of initial form/design space for an office building. The suggested design process uses and negotiates several performance envelopes in a computer-based form generation process. The presented process generates many design variations from which the designer can learn and choose the best variation according to several types of fitness criteria. The case study shows that designing with performance envelopes increases the general performance of the building form by increasing the amount of performance oriented information from which the buildings' form is generated while allowing to generate an architectural form that embeds a combination of user preference with empiric performance information.

12. Conclusions

12.1 Introduction

Within a wider scope of the influence of the introduction of computers on architecture discipline[72] this research has examined the influence and implications of the introduction of computers on the architectural design process. Its main motivation was to suggest ways of further incorporating the architecture of computation into the computation of architecture (Chu, 2005). The preliminary stage saw the development of a database of projects that followed a new definition for digital architecture, or computer-oriented design. The database showed trends relating to types, costs, distribution and other factors in computer-oriented design projects, starting from the mid-1990s.

After constructing the database, the research focused on developments in architectural design tools and software; design methods and computer-based optimization; and generation tools and approaches. It discussed the rise in the level of control by architects over the designed architectural form, which stems from the increase in the amount of data the architectural form embeds, and its effect on the architectural design process.

Through a division into form-based and performance-based design, it discussed the possibility of employing empirical data in the optimization and generation of the initial building design.

Based on the insights gleaned from the above analyses, the research introduced the notion of multiple performance envelopes as a generative design tool. It then developed a model that served as the basis for a design tool and a design method that use multiple performance envelopes to generate a building's initial form. The design tool enables the generation of form using one or more performance envelopes that can be related to similar or different types of performances. The developed tool proposes a nonlinear design method in which the computer generates several design alternatives that can be evaluated by one or several fitness criteria. A total grade is calculated for each alternative based on the ratio of influence of each fitness criterion, which has been decided by the designer.

The generated form in the proposed design method adheres by definition to the performances defined by the envelopes that were used to generate it. This can increase the general performance of the building form and save time in the design process while producing a form that embeds more information.

[72] See the discussion on chapter 5.3

12.2 Changes in architectural design process following the shift toward digital architecture or the introduction of computers to the architectural design process

12.2.1 Form and performance in computer-based architectural design

This research divides the contribution of computers to the architectural design process into two main domains: performance-based design and form-based design. It suggests that the increase in the amount of information architects have about the design drives architectural design toward being more performance-based. It proposes a division into two types of performances: those that support empirical design and those that cannot be empirically measured or analyzed.

The empirical performances usually relate to environmental and physical data such as strength, temperature, lighting levels, etc. The non-empirical performances relate to cognitive and perceptual realms, and focus, in this context, on the way human cognition/perception can be translated into space and, conversely, the way space can be translated into human cognition/perception.

The empirical performances, as shown in this research, are translatable into computer language in various ways. However, translation into computer language of the cognitive and perceptual performances, which currently can be measured mainly by statistical methods (which, for example, numerically examine the preferences or the aesthetic evaluations of a group of people in a particular space), still constitutes a complex problem for which there are no immediate solutions.

This research suggests that given the move toward performance-based design and the increase in the connectivity between software, the architectural design process should embed simulation, evaluation and optimization processes. This will further increase the architect's control over the design, which can open the way to the creation of a better architectural design.

Regarding form-based design, this research follows earlier approaches positing that computer form-based design increases architects' creativity by using computer processing power to generate form unprecedented in its complexity and quantity (design alternatives). Currently, approximately 20 years after the introduction of 3-D tools that allow designers to manipulate complex form, and about 15 years since the first commercial architectural projects that used the computer for manufacturing (Frank Gehry's Fish and Guggenheim Museum Bilbao, Lars Spuybroek's Fresh Water Pavilion and other projects that are mentioned in the database), the architectural discourse is shifting from concentration on the formal aspects to the performative aspects[73] (Neuman, Grobman. 2008).

The emergence of the computer was one of the major reasons behind the diversion of the architectural discourse to modes of thinking that go beyond form or function, in a way that does not discard the discussion of these factors, but attempts to define the connection between them. It can be argued that the connection between form and function is meant to define the way in which the former sustains the latter, and that a connection of this kind could be actualized through an examination of the performances, so that by means of the performances required by the function, it becomes possible to arrive at the form (Grobman, 2008).

12.2.2 Toward a higher level of information embedded in the architectural form

Recent developments in computation have changed the architectural form from an inert representation of purely geometrical data into a “smart form” that embeds information on material properties and communicates by coded constraints with the surrounding design space. This research argues that the next important step in this direction should involve the integration of building optimization tools. Optimization, according to the definition in this research, includes simulation, evaluation and modification modules. At present, two types of integration can be developed. The first embeds existing or new simulation models into architectural design. This can be accomplished in the near future, but it will be of limited use due to its inapplicability to all the design stages and to the level of knowledge it demands from architects. The second type, which could be developed specifically for architects further into the future, should be adaptable regarding its simulation’s level of accuracy. This simulation or optimization module would fit various stages in the architectural design process and would run in the background providing continuous, dynamic readings on predefined performative aspects of the developing architectural design.

Embedding simulation and optimization processes in architectural design does not mean that architects will stop using professional consultants. On the contrary, the importance and precision of performance aspects in the building design will increase, and consultants will be asked to fine-tune the initial form that was roughly optimized by the designers.

[73] This assertion is also one of the main concepts underpinning “Performalism – Form and Performance in Digital Architecture,” an international exhibition presented at the Tel Aviv Museum of Art in June-September 2008. The concept is articulated in the curatorial article in the exhibition catalogue by Neuman, Grobman (2008).

12.2.3 The architectural representation language

One of the most immediate and noticeable changes in the architectural design process is the change in the architectural representation language. The complexity of the digitally generated form, the shift toward nonstandard form and computer-controlled direct manufacturing has made the traditional 2-D representation language (mainly plans and sections) insufficient and inefficient in terms of both information and time (the information needed to represent the form and the time to produce sufficient 2-D representation). The currently emerging representation language is more complex and relies simultaneously on both 3-D models and 2-D drawings, according to the specific relevant needs. Moreover, the move toward the nonstandard calls into question the need for representation conventions at all, positing instead a nonstandard, one-off representation method for each specific project. Nevertheless, this hybrid method of representation will probably use a mix of existing representation methods and would not suggest a new one.

12.2.4 The use of computer code by architects

The transition to computer-based design indicates that the architects of the future will require a greater mastery of mathematics and of computer languages to enable them at least to adapt existing tools to their own needs, if not to improve their skills for writing new code.

On the whole, these skills are not part of current architectural education. This raises two main questions: the first involves whether it is necessary for architects to master code as opposed to using professional programmers, and the second has to do with the education of architects and the level of code mastery they need to attain. There are no straightforward answers to these questions, mainly because this domain has not been widely examined by architects and researchers. Nevertheless, it seems logical to postulate that while architects will not require the qualifications of a programmer or a mathematician, they will need to achieve an understanding of and the ability to use computer-based parametric processes that already are being employed on interdisciplinary levels. Architects will also need to be able to communicate with professional programmers and mathematicians in their own language.

12.2.5 Form generation in the architectural design process

Computer-based form generation has not been widely used in architectural practice because it demands knowledge and an investment of time and resources beyond what the average architectural practice can afford. Since the architectural problem is ill-defined and involves qualitative data that cannot be empirically measured, computer-based form generation should concentrate on two main directions: first, the generation of form for the early design stages in which the main programmatic demands are confronted and the level of detailed demands is limited. The second direction deals with specifics in the more advanced stages of the design process where the target function can be easily defined; examples include the generation of façade pattern, which is based on a geometric pattern or climatic empirical data, and the generation of details or designs in which empirical performance plays an important role, such as theaters, stadiums and hospitals. Nevertheless, since the architectural form is subjective by definition, the designer should also be more involved during the form-generation process and not only at the evaluation stage toward the end. This can be done by embedding the designer's input as part of the evaluation criteria that control the generation of the form. Since it seems that in the near future computer-form generation will not reach the level of generating the final building's form, at a certain point in this type of design process architects would need to choose one alternative and switch to a more traditional design in which the computer is used for representation and modification of form. Also, form generation so far mainly concentrates on the generation of building envelopes. The generation of complex typological forms, which beyond the building envelope also include a division into secondary interior spaces, probably will remain one of the main challenges for the near future.

12.2.6 Form optimization in the architectural design process

Since it is impossible to define the totality of the architectural problem, it is also impossible to solve it empirically as is done in modern science. Hence, it is hard to speak of full optimization of form in the scientific/empirical sense. In an optimization process that entails more than one parameter belonging to the empirical dimension of performance, there must be a subjective definition of preferences in order to arrive at the "optimum. " And even then the optimum will always be specific, since, as already noted, the order in which the processes are activated, and the kinds of parameters chosen, change the final product. Hence, because of the subjective definition and the complexity of the architectural problem that entails reference to many parameters, the idea of optimization in architecture takes on a different meaning. The problem is even more difficult in the cognitive and the perceptual dimensions of the concept of performance, because the initial definition of the parameters is done subjectively by a statistical translation of human desires and impressions.

12.2.7 The position of the architect in the discipline

One of the foreseeable effects of the transition to computer-based design and manufacturing is a rise in the architect's status in the set of forces operating in the building discipline. If, before this transition, the architect was responsible for the design and production of plans that were the builder/contractor's job to realize, in computer-based design and manufacturing, the architect in fact produces the code that generates the form and the file from which the real object is produced, without any need for mediators.

In terms of the design process, one of the ramifications of the enhancement of the architect's position in the building discipline is the proliferation of possibilities during the design process of using tools and processes such as simulation and optimization, which until now were almost entirely the domain of researchers, advisers and engineers. The incorporation of the simulation processes as part of the design process performed by the architect is unlikely do away with the need for professional advisers, but it will probably lead to a professionalizing and a fine-tuning of the examined parameter/solution.

12.3 The use of performance envelopes in generative architectural design and developing computer generative tools based on performance envelopes

This research presents the possibility of using multiple performance envelopes as a generative tool in architectural design. It defines types of performances that can be used to produce performance envelopes. It also defines the type of performance envelopes that can be used in the form generation process and suggests a twofold design method for generating form: a general method for creating the initial building form and a local method for optimizing areas that have special local needs in terms of performance.

The tool developed in this research uses performance envelopes to generate architectural form and offers a unique design method in which:

a. One or more performance envelopes can be used to generate form.

b. Numerous design alternatives are generated at each stage (the number of alternatives is defined by the designer).

c. An interactive evaluation module enables the designer to grade fitness criteria according to their importance in the particular design stage and to change the grades at any stage in order to examine different evaluation scenarios.

d. A total grade is calculated for each alternative and an end solution is defined according to the grade rankings for each criterion set by the designer.

The entire design process is parametrically controlled. This enables the designer to change any decision that was taken at any stage of the process and to examine the implications of the change on the final solution.

The proposed method and the use of performance envelopes to generate an initial form are applicable when performance aspects that can be described as envelopes are of considerable importance in the projects' brief. In other cases, performance envelopes and the suggested method can be used to modify/generate local areas or parts of the design.

The model was demonstrated in a case study. We can conclude from this study the following:

a. Performance envelopes

1. The initial form that was generated by negotiating performance envelopes offers another layer of information, which is embedded in the increasingly “smart” architectural form.
2 The case study demonstrates the feasibility of negotiating several different performance envelopes in an architectural design form-generation process.
3. The case study demonstrates the feasibility of using performance envelopes to modify local areas in the developed solution.
4. While the tool was examined on a building scale, it seems likely that it can also be used on an urban scale; this point, however, requires further research.
5. The design method employed emphasizes the need to embed simulation modules in architectural software.
6. The use of the developed design model necessitates specific knowledge that in some cases can exceed architects’ formal education. Moreover, the anticipated increase in the importance and measurability of performance aspects in design is expected to go beyond architects’ knowledge and should be addressed in a discussion regarding future education of architects.

b. Fitness criteria

1. The case study demonstrates the feasibility of designers using interactive empiric fitness criteria as part of the design evaluation process.
2. The new approach to grading helps designers evaluate the generated forms both quantitatively and qualitatively by allowing the comparison of different scenarios according to the criteria importance.

c. Nonlinear design

1. By introducing the ability to generate and evaluate numerous design alternatives in every design step and the ability to change the controlling parameters of all the design stages at any point of the process, the suggested design method moves from the traditional linear design process, which concentrates on examining a single design at any design stage, to a new, nonlinear design approach.
2. The number of alternatives to be generated must be defined in relation to the stage of generation (initial generation = many alternatives) and the expected difference between the results (big difference = many alternatives).

12.4 Future research

12.4.1 Performance envelopes in the architectural design process

The scope of this research allows for the examination of only a small number of performance envelopes from the list of possible envelopes that can be used in the method developed. Therefore, a direct continuation of this research would be to further examine different types of envelopes and the resultant possible negotiation scenarios. At this stage, it seems that structural envelopes based on finite element simulation results or more explorative methods such as in Shea (2004) or similar to the method used by Isozaki and Sasaki in their proposal for the new train station in Florence, can be also integrated in the model (Sasaki, 2007). Another promising performance envelope in this premise is visual exposure. A potential extension of this premise can be based on further developing the method to generate visibility graphs, as suggested by Turner et al (2001) or in the approach developed by Fisher-Gewirtzman and Wagner (2003) and Fisher-Gewirtzman et al (2005).

In terms of scale, the scope of this research allows for the examination of only the building scale. Neither the urban scale nor the small element scale has yet been examined, although the developed method is expected to be applicable to both scales, provided some adjustments are made. Further research could examine and devise specific characterizations for working with performance envelopes in these scales. Moreover, it is expected that different types of performance envelopes could be more applicable in these scale. Acoustic performance envelopes, for example, might be considered in the generation of an initial urban residential plan.

The suggested method, which uses performance envelopes, is based on the ability to generate performance envelopes in simulation software. Only recently, architectural software such as Digital Practice has started to embed simulation and evaluation modules. Although there has been a great deal of research on simulation and evaluation in architecture in general, only a few studies have concentrated on methods and implications of embedding simulation and evaluation modules in the architectural design process.

12.4.2 Fitness criteria and performance in computer-based generated architectural form

This research proposes a new grading approach that can negotiate several fitness criteria. The scope of this research allowed implementing a small number of fitness criteria. Future research is needed to examine both the general idea of fitness criteria in architecture and more specifically the possibility of coding criteria in computer algorithm; a method that will allow designers to interactively negotiate several fitness criteria should be further developed. Moreover, recently developed multi-criteria systems, as suggested by Gololov and Yezioro

(2007) and involving several different aspects considered in the design of the building's envelope, seem to have the potential to be easily embedded in the proposed tool.

12.4.3 Computer form-based design

This research offers a division of computer-generated architectural form into form-based and performance-based form generation. The study's scope allows it to focus only on the issue of performance-based form generation. It is important, however, to examine form-based design, both from an historic point of view (analyzing the development of methods to generate forms in architecture) and from an applicative point of view (analyzing and proposing future methods of using computers to generate form in a creative way that does not involve performative aspects). Nevertheless, form-based and performance-based design do not necessarily have to be examined separately. Another intriguing avenue for future research would be to propose and discuss ways to implement integrative methods of using computer-based generation.

In term of the proposed model, which was developed in orientation towards working with performance envelopes, it seems highly probable that it can be also used as a powerful morphing tool .As a morphing tool it will be able to examine both visually and quantitatively (using the grading algorithm) numerous formal alternatives which exist between two or more envelopes/forms.

12.4.4 Computer-based form generation and optimization for later stages of the design process

This research concentrates on the early stages of the architectural design process. Following the rapid developments in this field within the last decade, some of the limitations in computer processing power and in connectivity between architectural and other software appear to have disappeared or decreased substantially. This makes it possible to start examining the prospect of using the computer for form generation and optimization also in the later stages of architectural design, which are much more complex in their programmatic demands from the architectural form.

12.4.5 The position of the architect in the discipline

Although not within the main focus of this research, a noticeable finding emerged during the analysis of the computers' influence on the design process: The move to "one-model building" based on BIM (building information model) and direct manufacturing brings with it significant changes in the status and position of architects within the building discipline. Being responsible for producing files from which the building is directly manufactured is

expected to restore the architect’s stature after a long period of declining power, which derived from the increasing complexity of architectural projects and the rise in the number of participants in the design process.
This research did not find any comprehensive examination of this shift and its implications for the architectural practice and the entire building discipline, but it would be worthwhile to conduct some research in this direction.

12.4.6 Developing computer code/scripts by architects

Another noticeable issue that emerged during the analysis of the computers’ influence on the design process is the increasing accessibility of architects to both code/scripting modules of commercial design software and to generic code writing software. These codes or scripting tools are often used by architects to develop new design methods and to increase productivity.
This research did not find any comprehensive examination of the possibilities and limitations of code developing by architects, and its implications for the architectural practice and the future education of architects, but it would be worthwhile to conduct some research in this direction.

13. References

Abarbanel, R. 2000. 300 million megabytes flying in close formation. In *New Technologies in Architecture – Digital Design and Manufacturing Techniques*, eds. Bechthold, Groggs, Schodek, Steinberg. Proceedings of the First International Conference, Harvard University Graduate School of Design. October 2000.

Agarwal, M., Cagan, J. and Constantine, K.G. 1999. Artificial intelligence for engineering. *Analysis and Manufacturing 13*. Cambridge, UK: Cambridge University Press, 241-251.

Agrest, D. 2000. *Representation as Articulation between Theory and Practice*. *In Practice: Architecture, Technique and Representation By Stan Allen* Amsterdam: Routledge.176.

Aish, R. 2000. Custom Objects: A model-oriented end-user programming environment. Visual End User workshop, IEEE Visual Languages, Seattle

Alexander, C. 1964. *Notes on the Synthesis of Form*. Cambridge, Mass.: Harvard University Press. 9.

Alexander, C. 1967. The question of computers in design. *Landscape*, Autumn 1967. 8-12.

Andia, A. 1997. Managing Technological Changes in Architectural Practice: The Role of Computers in the Culture of Design. PhD dissertation, University of California at Berkeley.

Archer, B. 1984. Whatever became of design methodology? In *Development in Design Methodology*, ed. Cross, N., 347-349. Chichester, UK: John Wiley & Sons. Originally published in Design Studies, 1. 1979. 17-19.

Arvin, S. A. and House, D. H. 2002. Modeling architectural design objectives in physically based space planning, *Automation in Construction No.11*, 213–225.

Balcomb, J.D. and Curtner, A. 2000. Multi-criteria decision-making process for buildings. Paper presented at the American Institute of Aeronautics and Astronautics Conference in Las Vegas, Nevada, July 24-28.

Balmond, C. and Weinstock, M. 2002. The digital and the material: Contemporary techniques in architecture. *AD Vol. 72, No.1*, January, 46-51.

Beauce, P. and Cache, B. 2003. Towards a non-standard mode of production. In *Phylogenesis, foa's ark*. Barcelona: Actar. 391.

Benjamin, A. 2000. *Architectural Philosophy*. London: Athlone Press. 69**.**

Bentley, J. P. 1999. *Evolutionary Design by Computers*. San Francisco: Morgan Kaufmann.

Bitan, A. and Rubin S. 1991. *Climatic Atlas of Israel for Physical and Environmental Planning and Design. Section 7, Sde Dov (Hebrew)*. Tel Aviv: Ramot Press – Tel Aviv University.

Bleiberg, T. 2003. A model for enhancing the collaboration among the design participants in the process of computer-aided collaborative design. PhD dissertation. Haifa: Technion Israel Institute of Technology.

Burry, M. and Murray, Z. 1997. Architectural design based on parametric variation and associative
geometry. In *Challenges of the Future,* 15th eCAADe Conference Proceedings, Vienna.
Cache, B. 1995. *Earth Moves: The Furnishing of Territories*. Cambridge, Mass: The MIT Press.
Callicott, Nick. 2001. *Computer-Aided Manufacturing in Architecture – The Pursuit of Novelty*. Oxford, UK: Architectural Press.
Capeluto, I. G. 1991. A CAD System for Design and Evaluation of Solar Apartment Buildings. M.Sc. dissertation (Hebrew). Haifa: Technion Israel Institute of Technology.
Capeluto, I. G. 1996. Architecture and Environment - Modeling a Sustainable Architecture. D.Sc dissertation (Hebrew). Haifa: Technion Israel Institute of Technology.
Capeluto, I. G. and Shaviv, E. 1997. Modeling the design of urban grids and fabric with solar rights considerations. Paper presented at the ISES 1997 Solar World Congress in Taejon, Korea. 148-160.
Capeluto I. G, **2003a**. The influence of the urban environment on the availability of daylight in office buildings in Israel. *Building and Environment*, *Vol. 38/5*: 745-752**.**
Capeluto, I. G. 2003b. Energy performance of the self-shading building envelope. *Energy and Buildings 35*: 327-336.
Chayutin, M. 1975. Utilization of optimization technique for the evaluation of architectural systems. PhD dissertation (Hebrew). Haifa: Technion Israel Institute of Technology.
Choudhary, R., Malkawi, A.M. and Papalambros, P.Y. 2003. A hierarchical design optimization framework for building performance analysis. Eighth International IBPSA Conference in Eindhoven, Netherlands, August. 179-186.
Chu, K. 2000. *The Turing Dimension.*
http://www.archilab.org/public/2000/catalog/xkavya/xkavyaen.htm
Chu, K. 2005. Metaphysics of genetic architecture and computation. In *Genetic Architecture II: Digital Tools & Organic Forms*. Santa Fe: Sites Books/ESARQ. 158-180.
Chu, K. 2006. Planetary Automata. The Gen[h]ome Project. Ed. Peter Noever, Open Source Architecture. Los Angeles: MAK Center. 32-37.
Clarke, J, A. 2001. *Energy Simulation in Building Design*. Oxford: Butterworth-Heinemann.
Cross, N. 1984. *Developments in Design Methodology*. Chichester, UK: John Wiley and Sons.
Cross, N. 1989. *Engineering Design Methods*. Chichester, UK: John Wiley and Sons. 33.
Deleuze, G. and Guattari, F. 1998. *A Thousand Plateaus: Capitalism and Schizophrenia*. London: Athlone Press.
De Bono, E. 1970. *Lateral Thinking*. London: Penguin Books.
De Landa, M. 2003. Deleuze and the use of the genetic algorithm in architecture. In *Phylogenesis, foa's ark*. Barcelona: Actar, 521. Originally presented at the conference E-

Futures: Designing for a Digital World held at the RIBA, London and published in Neil Leach, ed., *Designing for a Digital World* (London: Wiley Academy. 2002).
De Luca, F. and Nardini, M. 2002. *Behind the Scenes. Avant-garde Techniques in Contemporary Design*. Basel, Boston, Berlin: Birkhauser. 9, 30, 41.
De Vries, B., Jessurun, A.J. and Van Wijk, J.J. 2001. Interactive 3-D modeling in the inception phase of architectural design. In Eurographics Short Presentations, Manchester, UK, September 4-7, 265-271.
Dollens, D. 2005. *Digital Botanic Architecture, D-B-A*. Santa Fe: Sites Books.
Duarte, J. 2001. Customizing mass housing: A discursive grammar for Siza's Malagueira houses. PhD dissertation. Massachusetts Institute of Technology, Dept. of Architecture.
Eckert, C., Kelly, I. and Stacey, M. 1999. Artificial intelligence for engineering design. *Analysis and Manufacturing 13*: 303–320.
Eisenman, P. 1992. Visions unfolding: Architecture in the age of electronic media. *Domus 734*: 17-21.
Estevez, A.T. 2005. Biomorphic architecture. In *Genetic Architecture II: Digital Tools & Organic Forms*. Santa Fe: Sites Books/ESARQ. 54-80.
Eggink, D., Gross, M.D. and Do, E. 2001. Smart objects: Constraints and behaviors in a 3-D design Environment. In Proceedings of 19th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe) in Helsinki, Finland, August 29-31. 460-465.
Fenves, S.J., Fleming, U., Hendrickson and C. Mahr, M.L. 1992. Performance evaluation in an integrated software environment for building design and construction planning. In *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*, ed. Kalay, Y., 159-170. Chichester, UK: Wiley & Sons, 2001.
Fisher-Gewirtzman, D., Shach-Pinsly D., Wagner, I.A. and Burt M. 2005. View-oriented three-dimensional visual analysis models for the urban environment." *Urban Design International*, *Vol. 10*: 23-37.
Fisher-Gewirtzman, D. Wagner, I.A. 2003. Spatial openness as a practical metric for evaluating built-up environments. *Environment and Planning B: Planning and Design, Vol. 30*: 37-49.
Frazer, J.H. 1995. *An Evolutionary Architecture.* London: Architectural Association Publications.
Frew, S. R. 1980. A survey of space allocation algorithms in use in architectural design in the past twenty years. Proceedings of the 17th Conference on Design Automation held in Minneapolis, Minnesota, June 23-25.165–174.
Gallupe, B. 2003. Based on a presentation: Knowledge management, relationship systems, and enterprise systems: The digital architecture of the 21st century enterprise. University of Auckland, New Zealand, May.

Gero, J.S. 1996. Computers and creative design. In *The Global Design Studio*, ed. Tan, M. and Teh, R. National University of Singapore Press. 11-19.

Goldschmidt, G. 1997. Capturing indeterminism: Representation in the design problem space. *Design Studies 18*: 441-445.

Goldschmidt, G. 2001. Visual analogy -- A strategy for design reasoning and learning." In *Design Knowing and Learning: Cognition in Design Education*, ed. Eastman, C., Newstetter, W. and McCracken, M. Elsevier. 199-215.

Gololov, I. and Yezioro, A. 2007. A computer system for multi-criteria comparative evaluation of building envelopes. IBPSA International Conference on Building Simulation, Beijing, China.

Greetham, L. 2002. Inside .. Research. www.bentleyuser.org June 24, 2002.

Grobman, Y. J. 2008. *The Various Dimensions of the concept of "performance" in Architecture* in Performalism – Form and Performance in Digital Architecture. Tel Aviv Museum of Art, cat. 8/2008. ISBN 978-965-7161-73-9, June 2008

Hartog, J. P. den, Koutamanis, A. and Luscuere, P.G. 1998. Simulation and evaluation of environmental aspects throughout the design process in H. Timmermans: Proceedings of the Fourth Design and Decision Support Systems Conference in Maastricht.

Heisserman, J., Callahan, S., and Mattikalli, R. 2000. A design representation to support automated design generation. In *Artificial Intelligence in Design 00*, ed. Gero, J. Dordrecht: Kluwer Academic.
545–566.

Heylighen, A. 2000. In Case of Architectural Design: Critique and Praise of Case-Based Design in Architecture. PhD dissertation, ASRO, K.U. Leuven.

Heylighen, A. and Neuckermans, H. 2001. A Case Base of Case-Based Design tools for Architecture. Elsevier: *Computer Aided Design 33*. 1111-1122**.**

Ibrahim, M., Krawczyk, R. and Schipporeit, G. 2001. CAD Smart Objects: Potentials and Limitations. eCAADe 21 Proceedings. 547-551.

Imperiale, A. 2000. *New Flatness-Surface Tension in Digital Architecture*. Basel: Birkhauser. 73.

Jockusch, P. 1992. How can we achieve a good building? In *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*, ed. Kalay, Y., 2001. Chichester, UK: Wiley & Sons. 49.

Jones, C. J. 1984. A method of systematic design. In *Development in Design Methodology* ed. Cross, N. Chichester, UK: Wiley & Sons. 9.

Kalay, Y. 1992. ed. *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance.* New York: John Wiley & Sons.

Kalay, Y. 2004. *Architect's New Media: Principles: Theories and Methods of Computer-Aided Design.*. Cambridge, Massachusetts and London, England: MIT Press.

Kieran, S. and Timberlake, J. 2004. *Refabricating Architecture*. New York: McGraw-Hill, 69.

Kim, H., Querin, O.M. and Steven, G.P. 2002. On the development of structural optimization and its relevance in engineering design. *Design Studies 23*: 85–102.

Knight, T. 2000. Shape grammars in education and practice: History and prospects. *International Journal of Design Computation 2.* www.arch.usyd.edu.au/kcdc/journal

Kolarevic, B. 2003 a. eCAADe 21. Digital Design conference proceedings. Basel.

Kolarevic, B. 2003 b. Digital Morphogenesis. In *Architecture in the Digital Age – Design and Manufacturing*. London, New York: Spon Press. 26.

Kolarevic, B. and Malkawi, A, M. 2005. eds. *Performative Architecture – Beyond Instrumentality*. London, New York: Spon Press.

Koning, H. and Eizenberg, J. 1981. The Language of the Prairie: Frank Lloyd Wright's Prairie Houses. *Environment and Planning B 8*. 295-323.

Koutamanis, A. 2000. Redirecting design generation in architecture. http://www.generativeart.com/2000/koutamanis/ga_2000.htm. Proceedings of the 3rd International Conference on Generative Art held in Milan, Italy. December, 2000.

Kroes, P. 2002. Design methodology and the nature of technical artifacts. *Design Studies 23*. 87-302.

Kuhn, T. S. 1962. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.

Lam, K.P., Mahdavi, A., Gupta, S., Wong, N.H., Brahme, R. and Kang, Z. 2002. Integrated and distributed computational support for building performance evaluation. *Advances in Engineering Software, Vol. 33, No. 4*. 199-206

Leach, N. 2002. Introduction. Designing for a Digital World. West Sussex, UK: Wiley Academy. 6.

Leach, N. and WeiGuo, X. 2004. Fast Forward >> Hot Spot - Brain Cells, Architecture Biennial Beijing, 2004. Beijing: Map Book Press.

Le Corbusier, originally printed in **1923**, Towards a New Architecture. Tel Aviv: Bavel (Hebrew version, 1998). 240.

Leeuwen, J.S., Hendricx, A. and Fridqvist, S. 2001. Towards dynamic information modeling in architectural design. Proceedings CIB W78 2001 in Mpumalanga, South Africa, May 2001.

Lenoir, T. and Alt, C. 2002. Flow, Process, Fold: Interactions in Bioinformatics and Contemporary Architecture. In *Science, Metaphor and Architecture*, Antoine Picon and Alessandra Ponte, eds. Princeton: Princeton University Press. 2002.

Lindsey, B. 2001. *Digital Gehry – material resistance/digital construction*. Basel, Boston, Berlin: Birkhauser. 71.

Lootsma, B. 1999. In *Hybrid Space – New Forms in Digital Architecture*, ed. Peter Zellner. London: Thames & Hudson. 7.

Lynn, G. 1999. *Animate Form*. New York: Princeton Architectural Press. 13.

Magdy, I., Krawczyk, R. and Schipporeit, G. 2001. Cad Smart Objects: Potentials and Limitations. Proceedings eCAADe 21. 547-551.

Malkawi, A. M., Srinivasan, R. S., Yi, Y. K. and Choudhary, R. 2003 . Performance-based design evolution: The use of genetic algorithms and CFD. Proceedings of the Eighth International IBPSA Conference held in Eindhoven, Netherlands.793-798.

Manovich, L. 1999. What is digital cinema? In *The Digital Dialectic: New Essays on New Media,* ed. Lunenfeld, P. Cambridge, Mass: The MIT Press. 175.

Manovich, L. 2001. *The Language of New Media*. Cambridge, Mass: The MIT Press. 52.

March, L. and Steadman, P. 1974. The geometry of environment. Cambridge, Mass: The MIT Press. In *Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design,* Kalay, Y. Cambridge, Mass: The MIT Press, 2004.

Margolius, I. 2000. *Automobiles by Architects.* West Sussex, UK: Wiley Academy Press. 5.

McGill, M.C. 2001. A Visual Approach for Exploring Computational Design. Master's thesis, Massachusetts Institute of Technology. Cambridge, Mass.

McCullough, M. 2004. *Digital Ground.* Cambridge, Mass**:** The MIT Press.

Mirza & Nancy research. 1999. Architects CAD Survey 99. UK

Michalek, J., Choudhary, R. and Papalambros, P. 2002. Architectural layout design optimization.

Engineering Optimization, Vol. 34(5). 461-484.

Mitchell, W. 1990. *The Logic of Architecture – Design, Computation, and Cognition*. Cambridge, Mass: The MIT Press. 179-181

Mitchell, W. 1992. The uses of inconsistency in design**.** In *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*, ed. Kalay, Y. 1992. New York: John Wiley & Sons. 12.

Mitchell, W. 1999. *E-Topia.* Cambridge, Mass: The MIT Press. 147-154.

Mitchell, W. 2001. Roll over Euclid: How Frank Gehry designs and builds. In *Frank Gehry Architects*, ed. J. Fiona Ragheb. New York: Guggenheim Museum Publications. 352-363.

Monedero, J. 1997. Parametric design: A review and some experiences. Proceedings of the 15th eCAADe conference held in Vienna. 611-624

More, G. 2001. Animated techniques: Time and technological acquiescence of animation. *AD, Vol. 71, No. 2*. April 2001. 22.

Morbitzer, C.A. 2003. Towards the Integration of Simulation into the Building Design Process. PhD dissertation. Energy System Research Unit, Department of Mechanical Engineering, University of Strathclyde, Glasgow, Scotland.

Negroponte, N. 1970. *The Architecture Machine*. Cambridge, Mass: The MIT Press. 7,119, Preface.

Neuman, E., Grobman, Y. J. 2008. *Performalism: A Manifesto for architecture Performance* in Performalism – Form and Performance in Digital Architecture. Tel Aviv Museum of Art, cat. 8/2008. ISBN 978-965-7161-73-9, June 2008

Ng, E. 2003. Applying computational simulation results to the development of a design method for daylighting design and regulation in high-density cities. Proceedings of the Eighth International IBPSA Conference held in Eindhoven, Netherlands. 943-950.

Nir, E. 2006. Smart Cloud of Points as Digital Media and Tools for Building Envelope Design. PhD dissertation. Haifa: Technion Israel Institute of Technology. 13-14.

Nir, E. and Capeluto, I.G. 2005. Smart cloud-of-points model: Point-based digital media and tools for architectural design. In *eCAADe(23) 2005: Digital Design: The Quest for New Paradigms. Education and Research in COMPUTER AIDED ARCHITECTURAL DESIGN in Europe*, Lisbon, Portugal. 687-694.

Oden, T. 1987. Some historic comments on finite elements. Proceedings of the ACM conference on history of scientific and numeric computation, Princeton, New Jersey. 125-130.

Oosterhuis, K. 2002. *Programmable Architecture*. Italy: L'arca Edizioni. 59**.**

Oxman, R. 1988. Expert Systems for Generation and Evaluation in Architectural Design. PhD dissertation (Hebrew). Haifa: Technion Israel Institute of Technology. Chapter 2, pp 15-38.

Oxman, R. 1992. Multiple operative and interactive modes in knowledge-based design systems. In *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*, ed. Kalay, Y. 2001. New York: John Wiley & Sons. 125-143.

Oxman R. and Oxman R. 1990. The computability of architectural knowledge. In *The Electronic Design Studio*, eds. McCullough, M., Mitchell, W.J. and Purcell, P. Cambridge, Mass: The MIT Press. 181.

Pak, B., Ozener, O.O. and Erdem, A. 2003. Xp-GEN: A randomized design tool for non-deterministic digital design methods in architecture and visual design. In ed. Dokonal, W. and Hirschberg, U., *Digital Design*, Proceedings of the 21st Conference on Education in Computer Aided Architectural Design in Europe, eCAADe, held in Graz, 2003. 485– 488.

Papamichael, K. 1997. Designers and information overload: a new approach. *Advanced Buildings Newsletter*. Volume 1, Number 18.

Pham, T. and Dimov, S. 2003. Rapid prototyping and rapid tooling— the key enablers for rapid manufacturing. *Journal of Mechanical Engineering Science Vol. 217, Part C,* January, 2003.

Picon, A. 2003. *Architecture and the Sciences – Exchanging Metaphors*. Ed. Picon, A. and Ponte, A. New York: Princeton Architectural Press.

Picon, A. 2006. in Terzidis, K. 2006*. Algorithmic Architecture*. Oxford, UK: Architectural Press. VII.

Rahim A. 2002. Introduction to contemporary techniques in architecture. *AD, Vol. 72, No.1*, January 2002. 5-8.

Rahim, A. 2006. *Catalytic Formations – Architecture and Digital Design*. London, New York: Taylor & Francis. 22-29.

Rakatansky, M. 2003. Envelope Please. In *The State of Architecture at the Beginning of the 21st Century*, eds. Tschumi, B. and Cheng, I. New York: The Monacelli Press. 76.

Rittel, H. and Webber, H. 1969. Dilemmas in General Theory of Planning. Reprint 86, Institute of Urban and Regional Development, University of California, Berkeley, USA. 160-167.

Rittel, H. 1984. Second -generation design methods. In *Development in Design Methodology,* ed. Cross, N. Chichester, UK: Wiley & Sons. 316-327. Originally published in the DMG 5th anniversary report: DMG Occasional Paper No, 1 1972. 5-10.

Robinson, G., El-Beltagy, M. and Keane, A. 1999. Optimization in mechanical design. In *Evolutionary Design by Computers*, ed. Bentley, P.J. San Francisco: Morgan Kaufmann. 147-167.

Rocker, I. M. 2006. When code matters. *Architectural Design 76, Issue 4*. Wiley Academy. 16-25.

Rosa, J. **2001**. *Folds, Blobs + Boxes: Architecture in the Digital Era*. Pittsburgh: The Heinz Architectural Center, Carnegie Museum of Art. 5-16.

Rotheroe, K. 2000. Manufacturing free-form architecture. Proceedings of New Technologies in Architecture symposium, ed. Bechthold, Griggs, Schodek and Steinberg. Harvard Graduate School of Design, 2001. 43.

Rowe, P.G. 1987. *Design Thinking*. Cambridge, Mass**:** The MIT Press.

Ryder, B., Ion, B., Green, G., Harrison, D. and Wood, B. 2002. Rapid design and manufacture tools in architecture. *Automation in Construction Vol. 11, Issue 3*, April 2002. 279-290.

Sanders, K. 1996. *The Digital Architect*. New York: John Wiley & Sons. 1-31**.**

Sasaki, M. 2007. *Morphogenesis of Flux Structure*. London: AA Publications.

Schmitt, G. 1992. Design for Performance. In *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*, ed. Kalay, Y. 2001. New York: John Wiley & Sons. 87

Schodek, D., Bechthold, M., Griggs, K., Kao, M. K. and Steinberg, M. 2005. *Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design*. Hoboken, New Jersey: John Wiley & Sons Press.

Schmal, P.C. 2001. *Digital Real*. ed.. Basel, Boston, Berlin: Birkhauser. 8-19.

Schwartz, A., Berry, D. M. and Shaviv, E. 1994. On the use of the automated building design system. *Computer Aided Design, Vol. 26, No.10*. 747-762.

Segers, N.M. Achten, H.H. Timmermans, H.J.P. de Vries, B. (2000). A comparison of computer-aided tools for architectural design. In *Proceedings of the 5th International Conference on Design and Decision Support Systems,* ed. Timmermans, H.J.P. and Vries, B. de, August 22-25 2000, Nijkerk, 325-340.

Sei Watanabe, M. 2002. *Induction Design: A Method for Evolutionary Design*. Basel, Boston, Berlin: Birkhauser.

Shaviv, E. and Greenberg, D.P. 1970. Funicular surface structures: A computer graphics approach. *Bulletin of the International Association for Shell Structures 37***.** Madrid, Spain: 1968. 15-26.

Shaviv, E. and Gali, D. 1974. A model for space allocation in complex buildings: A computer graphics approach. *Build International 7*. England: Applied Science Publisher Ltd. 493-517.

Shaviv, E. 1987. Generative and evaluation CAD tools for spatial allocation problems. *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*, ed. Kalay, Y. 2001. New York: John Wiley & Sons.192.

Shaviv, E., Gavish, O. and Amir, U. 1990. Implementations of Solid Modeling in High Hierarchy Architectural Language, *Planning and Design Vol. 17*. 205-220.

Shea, K. and Cagan, J. 1999. Languages and semantics of grammatical discrete structures. *Artificial Intelligence for Engineering, Analysis and Manufacturing 13.* Cambridge, UK: Cambridge University Press. 241-251.

Shea, K. 2003. Home Front: New Development in Housing. *Architectural Design, Vol 73, Issue 4.* West Sussex, UK: Wiley Academy. 119

Shea, K. 2004. Directed Randomness. In *Digital Tectonics*. eds. Leach, N., Turnbull, D. and Williams, C. West Sussex, UK: Wiley Academy. 96.

Silver, M. 2006. Building without drawings: Automason 1.0. *Architectural Design*, *Vol. 76, Issue 4*. West Sussex, UK: Wiley Academy. 46-51.

Simpson, R. 1980. A Survey of Space Allocation Algorithms In use in Architectural Design in The past twenty years. ACM 0-89791-020-6/80/0600/0165.

Spuybroek, L. 2002. wetGRID: The Soft Machine of Vision. In *Designing for a Digital World*, Leach, N. West Sussex,UK: Wiley Academy. 93.

Stiny, G. and Gips, J. 1972. Shape grammars and the generative specification of painting and sculpture. In ed. Freiman, C.V., *Information Processing 71*, Amsterdam: North-Holland, 1460-1465. Republished in ed. Petrocelli, O.R. 1972 *The Best Computer Papers of 1971*. Philadelphia: Auerbach, 125-135. In www.shapegrammar.org/biblio.html

Stiny, G. 1980. Introduction to shapes and shape grammars. *Environment and Planning B 7*, 343-351.

Szalapaj, P. 2000. *CAD Principles for Architectural Design: An Analytic Approach to the Computational Representation of Architectural Form*. Oxford, UK: Architectural Press**.**

Szalapaj, P. 2001. Parametric Propagation of Form, *Architecture Wee*k (www.architectureweek.com), 10 October 2001.

Sutherland, I. E. 1963. Sketchpad: A man-machine graphical communication system, PhD dissertation, Electrical Engineering Dept, MIT, Cambridge, Mass. (2003, web edition. Technical report no. 574 ISSN 1476-2986, University of Cambridge, Computer Laboratory)

Schwarz, A., Berry, D. M. and Shaviv, E. 1994. On the use of the automated building design system. *Computer-Aided Design, Vol. 26, Issue 10*. 747-762.

Tapia, M. 1999. A visual implementation of a shape grammar system. *Environment and Planning. B.* Planning and Design 26(1). 59-73.

Turner, A., Doxa, M., O'Sullivan, D. and Penn, A. 2001. From isovists to visibility graphs: a methodology for the analysis of architectural space. *Environment and Planning B: Planning and Design, Vol. 28*. 103 –121.

Terzidis, K. 2003. *Expressive Form: A Conceptual Approach to Computational Design*. London: Spon Press.

Terzidis, K. 2006. *Algorithmic Architecture*. Oxford, UK: Architectural Press. 14-35.

Thompson, D. 1961. *On Growth and Form. An Abridged Edition*. Ed. Bonner, J.T. Cambridge, UK: Cambridge University Press.

Venturi, R. 1977. *Complexity and Contradiction in Architecture*, revised edition. New York: Museum of Modern Art. 19.

Vidler, A. 2001. The Medium and its Message, or ``I'm sorry, Dave, I don't have enough information.'' New York: *Architectural Record*, May, 2001.

Wang, Y. and Duarte, J.P. 2002. Automatic generation and fabrication of designs. *Automation in Construction 11(3)*. 291-302.

Walter, J. 2003. *Blobitecture: Waveform Architecture and Digital Design*. Gloucester, Mass: Rockport Publishers. 55.

Williams, C. 2004. Design by Algorithm. In *Digital Tectonics*. Eds. Leach, N. Turnbull, D. and Williams, C. West Sussex, UK: Wiley Academy. 79-87.

Weinstock, M. 2005. On self-organization. In *genetic architecture II: digital tools & organic forms*. Site Books. 100-106.

Weinstock, M. 2006. Self-organization and the structural dynamists of plants. In *Techniques and Technologies in Morphogenetic Design*. Ed. Weinstock, M., Hensel, M. and Menges, A. Wiley Academy. 27.

Wiezel, A. and Becker, R. 1992. Integration of performance evaluation in computer-aided design. In *Principles of Computer-Aided Design: Evaluating and Predicting Design Performance*. Ed. Kalay, Y. 2001. New York: John Wiley & Sons. 171-182.

Whitehead, H. 2003. Laws of form. In *Architecture in the Digital Age – Design and Manufacturing*, ed. Kolarevic, B. London, New York: Spon Press. 81-100.

Woodbury, R.F. and Burrow, A.L. 2006. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*. Cambridge, UK: Cambridge University Press. 63-82.

Yezioro, A. 1994. Form and Performance in Intelligent CAAD Systems for Early Stages in Solar Building Design. PhD dissertation. Haifa: Technion Israel Institute of Technology. 31-44.

Yezioro, A. and Shaviv, E. 1994. Shading: A design tool for analyzing mutual shading between buildings. *Solar Energy, Vol. 52, No. 1*. USA: Pergamon Press. 27-37.

Zehavi, R. and Carmon, N. 1996. Home Telecommunication: Is It Really Going to Change Our Lives? The Center for Urban and Regional Studies. Haifa: Technion Israel Institute of Technology Press (Hebrew).

Zellner, P. 1999. *Hybrid Space – New Forms in Digital Architecture.* London: Thames & Hudson. 13.

Appendix A: Digital database – information sources

The information for the database derives from architectural publication (books and professional magazines), architects official websites and online architectural The database was built in Microsoft Access environment.

Information sources:

Architects websites:

Sheet1
Architect's website
http://195.171.22.30/uhtbin/cgisirsi.exe/Nn2DtcRQlB/0/49
http://www.archilab.org/public/2000/catalog/naga/nagaen.htm
http://www.archilab.org/public/2000/catalog/naga/nagaen.htm
http://www.archi-tectonics.com/
http://www.asymptote.net/
http://www.coop-himmelblau.at/coophimmelblau.html
http://www.dillerscofidio.com/
http://www.eea-architects.com/swf/eea_flashed.html
http://www.eisenmanarchitects.com/
http://www.f-o-a.net
http://www.franken-architekten.de/
http://www.garofalo.a-node.net/home.html
http://www.geaf.com/
http://www.glform.com/
http://www.grimshaw-architects.com/index_new.html
http://www.gsd.harvard.edu/people/faculty/cohen/index.html
http://www.joelsandersarchitect.com/
http://www.kdlab.net/
http://www.kolatanmacdonaldstudio.com/
http://www.labarchitecture.com/
http://www.makoto-architect.com/index.htm
http://www.makoto-architect.com/index.htm
http://www.massiearchitecture.com/
http://www.michelesaee.com/
http://www.morphosis.net/morph.html
http://www.mvrdv.nl/mvrdv.html
http://www.new-territories.com/
http://www.nio.nl/
http://www.nlarchitects.nl/
http://www.noxarch.com
http://www.objectile.org/
http://www.oceand.com/
http://www.oosterhuis.nl
http://www.plexus-architecture.com/
http://www.ptp-architects.com.hk/
http://www.reiser-umemoto.com/

Sheet1
Architect's website
http://www.renevanzuuk.nl/
http://www.roydesign.com/
http://www.rpbw.com/
http://www.s-e-r-v-o.com/
http://www.sial.rmit.edu.au/People/Professor_Mark_Burry.php
http://www.stevenholl.com/
http://www.unstudio.com/
http://www.via-arquitectura.net/13/13-050.htm
http://www.xefirotarch.com/
http://www.zaha-hadid.com/
http://www.archi-tectonics.com/
http://www.coop-himmelblau.at/coophimmelblau.html
http://www.eisenmanarchitects.com/
http://www.f-o-a.net
http://www.fosterandpartners.com/
http://www.glform.com/
http://www.glform.com/
http://www.kolatanmacdonaldstudio.com/
http://www.labarchitecture.com/
http://www.oceand.com/

Magazines:

The Architectural Review – 1993-2005

Architectural Record – 1993-2005

Architectural Design (A.D) - 1993-2005

Books:

Cook, P., Spiller, N., Allen, L. and Rawes, P. 2001. *The Lowe Lectures: The Paradox of Contemporary Architecture*. London: Wiley Academy.

Hadid, Z. and Schumacher, P., 2002. *Latent Utopias: Experiments within Contemporary Architecture*. Vienna: Springer-verlag.

Leach, N., Turnbull, D. and Williams, C. 2004. *Digital Tectonics*. West Sussex, UK: Wiley Academy.

Migayrou, F. and Brayer, M.A. 2001. *ArchiLab: Radical Experiments in Global Architecture*. London: Thames & Hudson

Oosterhuis, K. 2003. *Hyperbodies: Toward an E-motive Architecture*. Basel, Boston, Berlin: Birkhauser.

Riley, T. 1999.*The Un-Private House*. New York: Museum of Modern Art.

Rosa, J. 2004. *Next Generation Architecture*. London: Thames & Hudson.

Schmal, P.C., ed. 2001. *Digital Real – Blobmeister: First Built Projects*. Basel, Boston, Berlin: Birkhauser.

Appendix B: Database analysis results

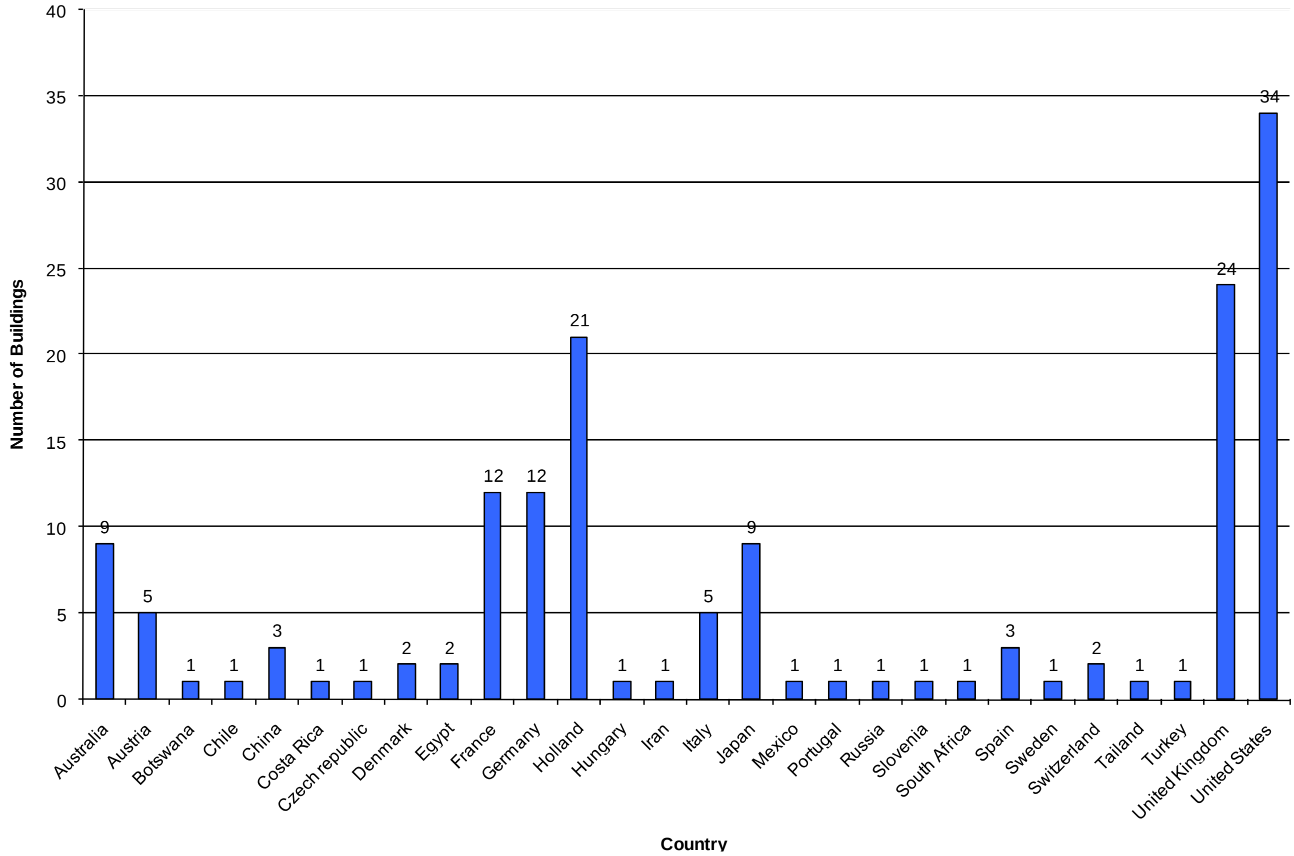


Figure 71 – number of digital projects per country

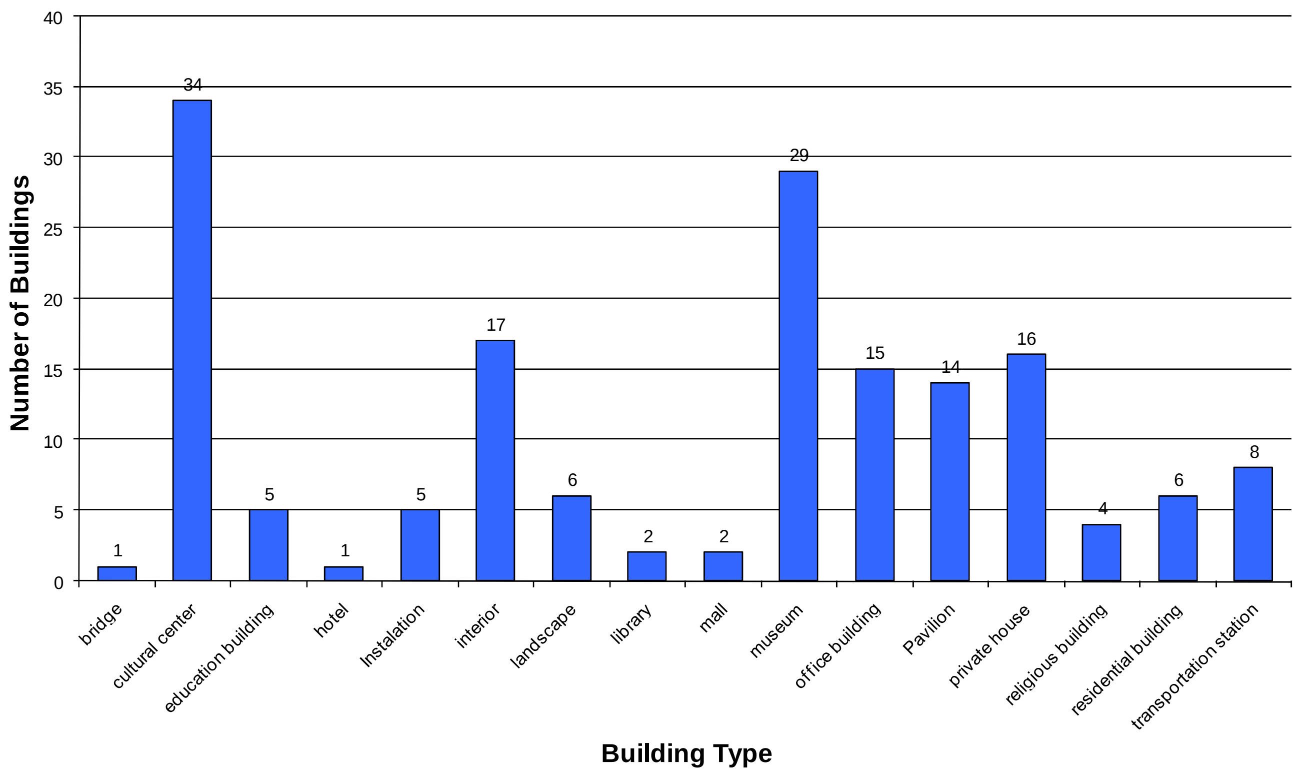


Figure 72 – digital projects' building type

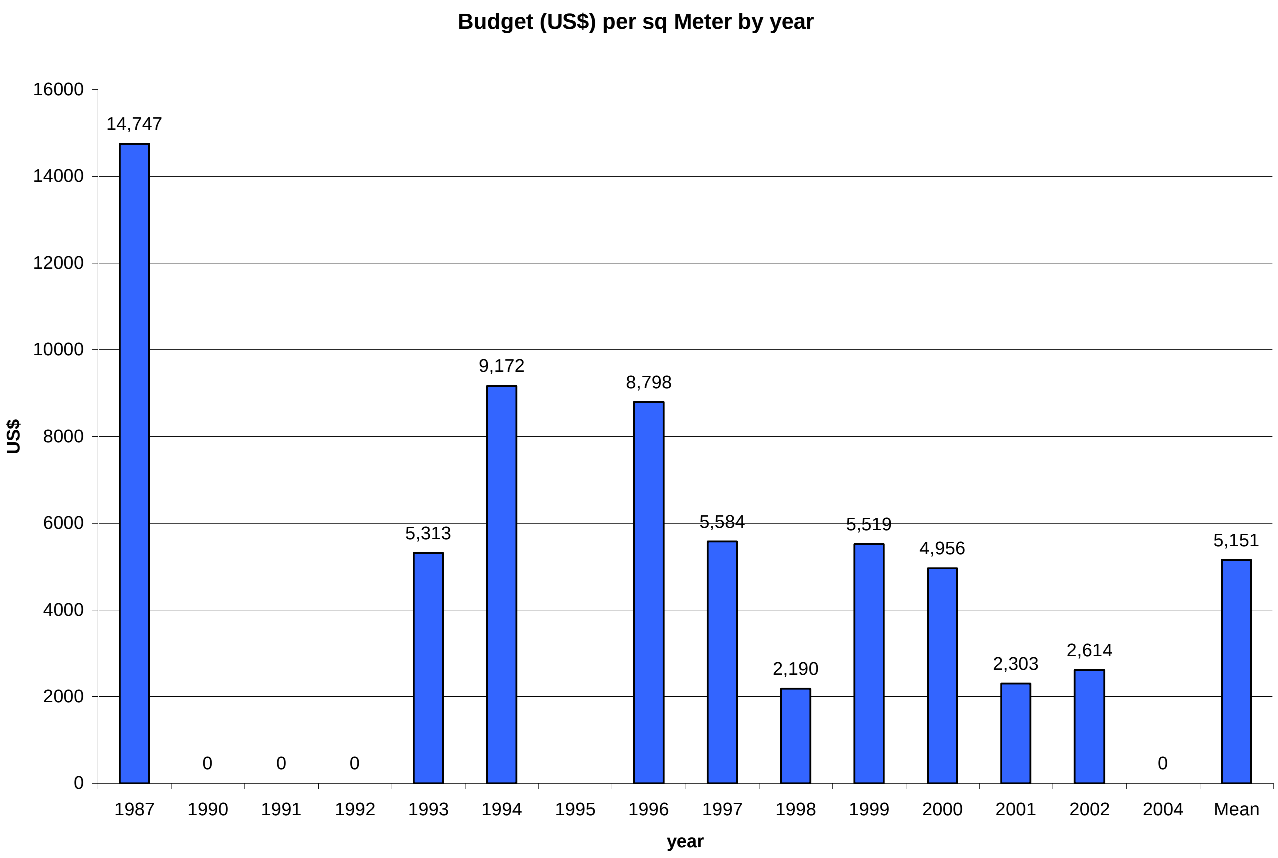


Figure 73 – digital projects budget per sq. meter

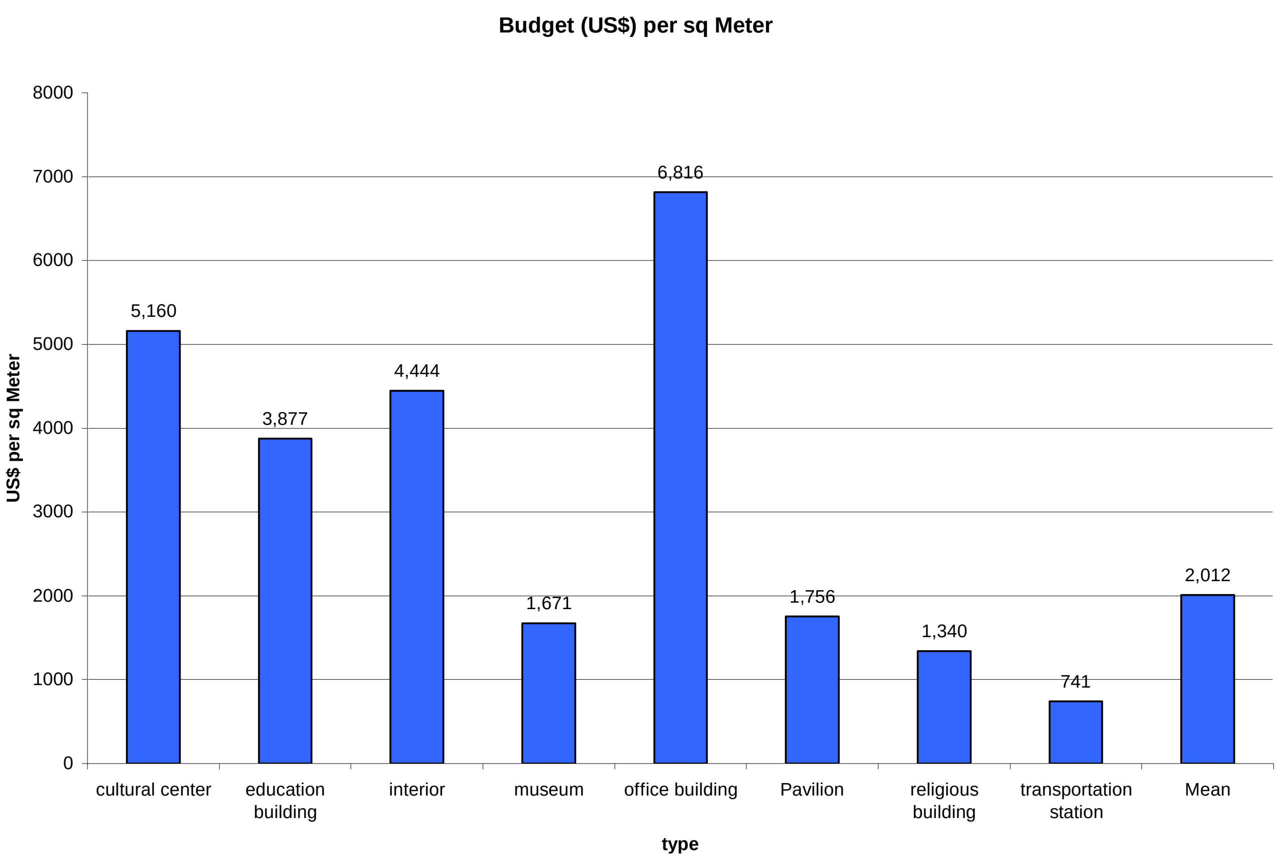


Figure 74 – cost per building type

Appendix C: Case study - form generation process

1. Initial Generation and evaluation process - stage 1 and stage 2

The initial setup was based on the demands presented in the brief regarding wind and shade conditions. Therefore solar right and solar catch envelopes were initially examined in order to define their common ground. The examination suggested that solar rights envelope can be used single-handedly, starting from the solar catch surface, since its volume includes almost completely the solar catch envelope (see Figure 75).

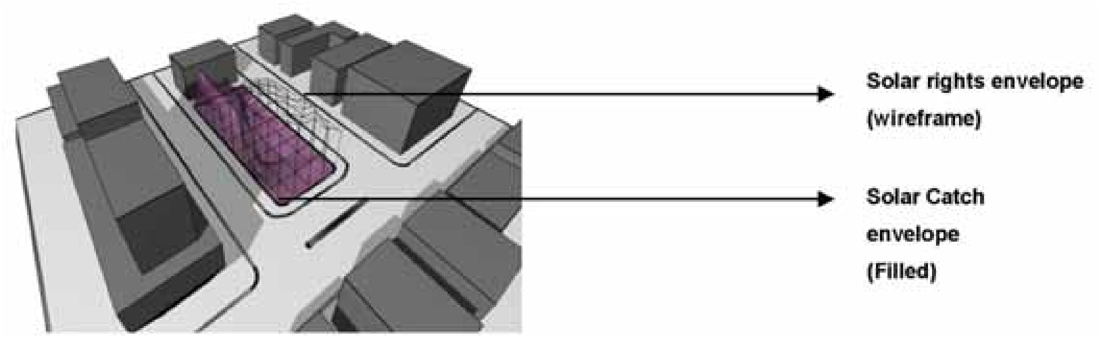


Figure 75 – finding common ground – solar catch and solar rights envelopes

The next step included an examination of the common grounds between solar rights envelope and wind velocity envelopes for 2 and 4 m/s. The definition of this range follows the BRE definitions for long exposure and short exposure (between Beaufort numbers B3-B4), which are described in appendix D and according to the considerations described in section 11.4.

The examination revealed 2 distinct volumes: In the first volume (marked A in Figure 76) the solar rights envelope is located in the middle of the range that is created between 2m/s and 4m/s envelopes. In the second volume (marked B in Figure 76) the solar right envelope is located up to 6 meters under the wind 2m/s envelope. This means that a decision has to be taken to either allow the new building to shade parts (1-2 lower floors) of the neighboring buildings or have insufficient ventilation in that area. Another important implication of the two different volumes is that following the initial generation process that deals with the entire form a local re-generation of one or the two volumes will be necessary in order to fine-tune the results since the envelopes' orientation is opposite in the two part.

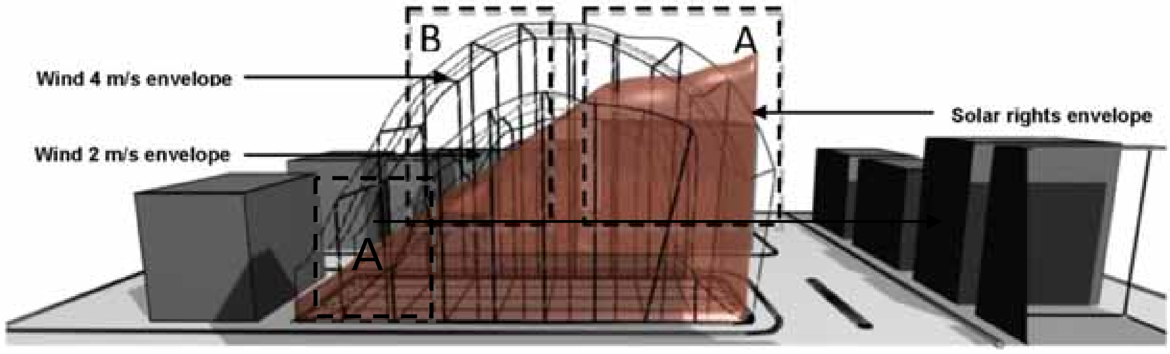


Figure 76 – finding common ground – wind 2, 4m/s and solar rights envelopes

As an initial generation run it was decided to generate 15 initial alternatives and use the entire range of the morphing channel. 15 generated alternatives would be presented in 3 rows of 5 alternatives each that would fit in the computer screen. The initial setup of the control panel is presented in Figure 77. The results of the initial generation showed a big difference in terms of floor area, volume and envelope area between the alternatives. The floor area and number of floors, as an example varied from 9 floors and a total of 8807 m^2 in the smaller alternative up to 11 floors (which contradict the brief's limits) and 15673 m^2 in the larger alternative. However, the deviation from the solar rights envelope in alternative 1-5 and 10-15 (below 35% and above 70% of the generation channel) would have cause shade in more than the first 2 floors of some of the surrounding buildings (Figure 78 presents the initial generation results). Therefore, it was decided to narrow the generation channel to 35%-70%.

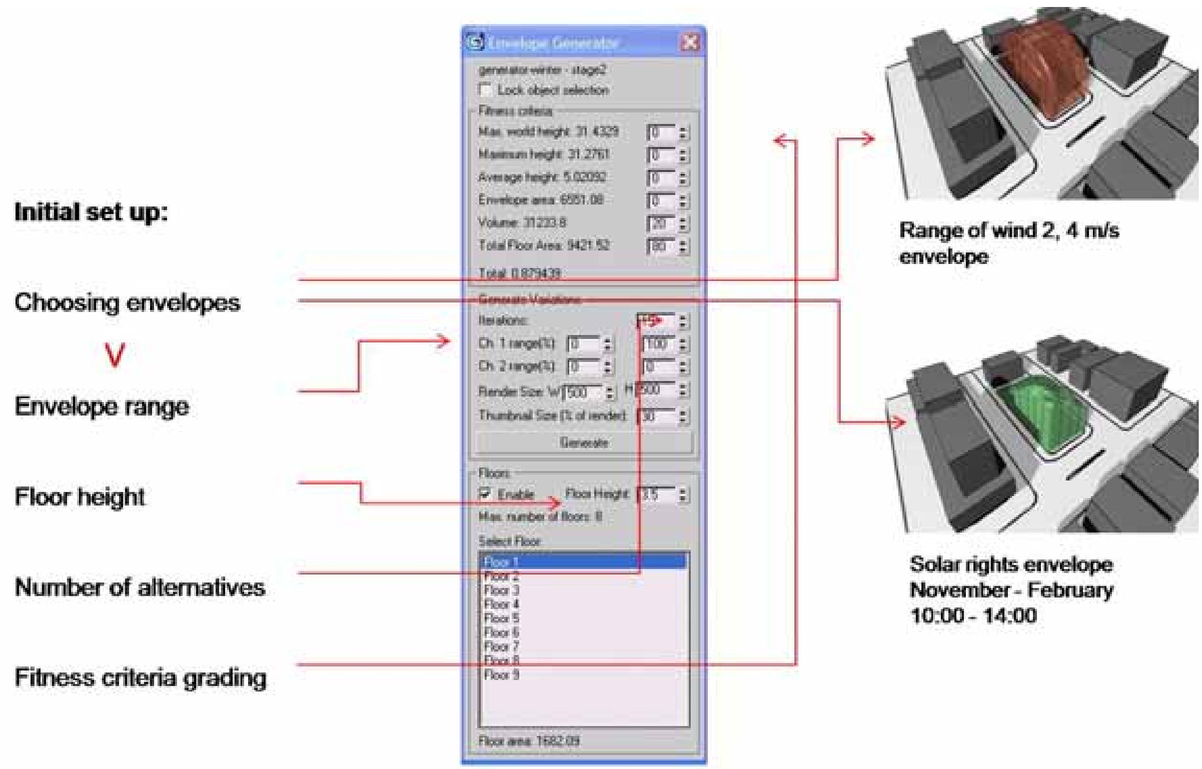


Figure 77 – initial setup (stage 1)

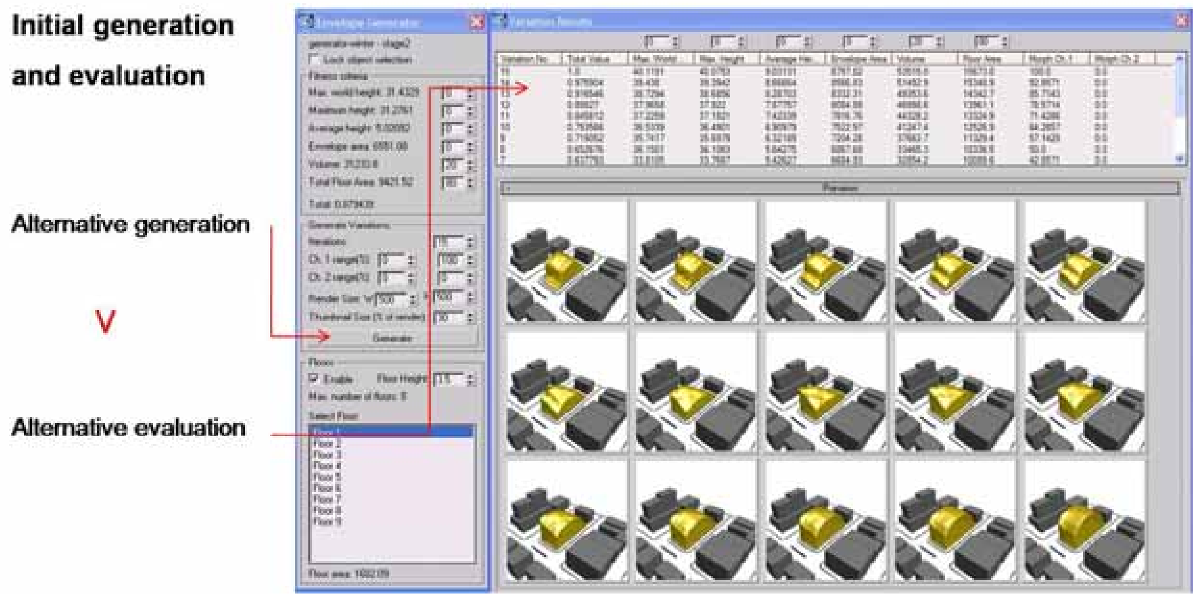
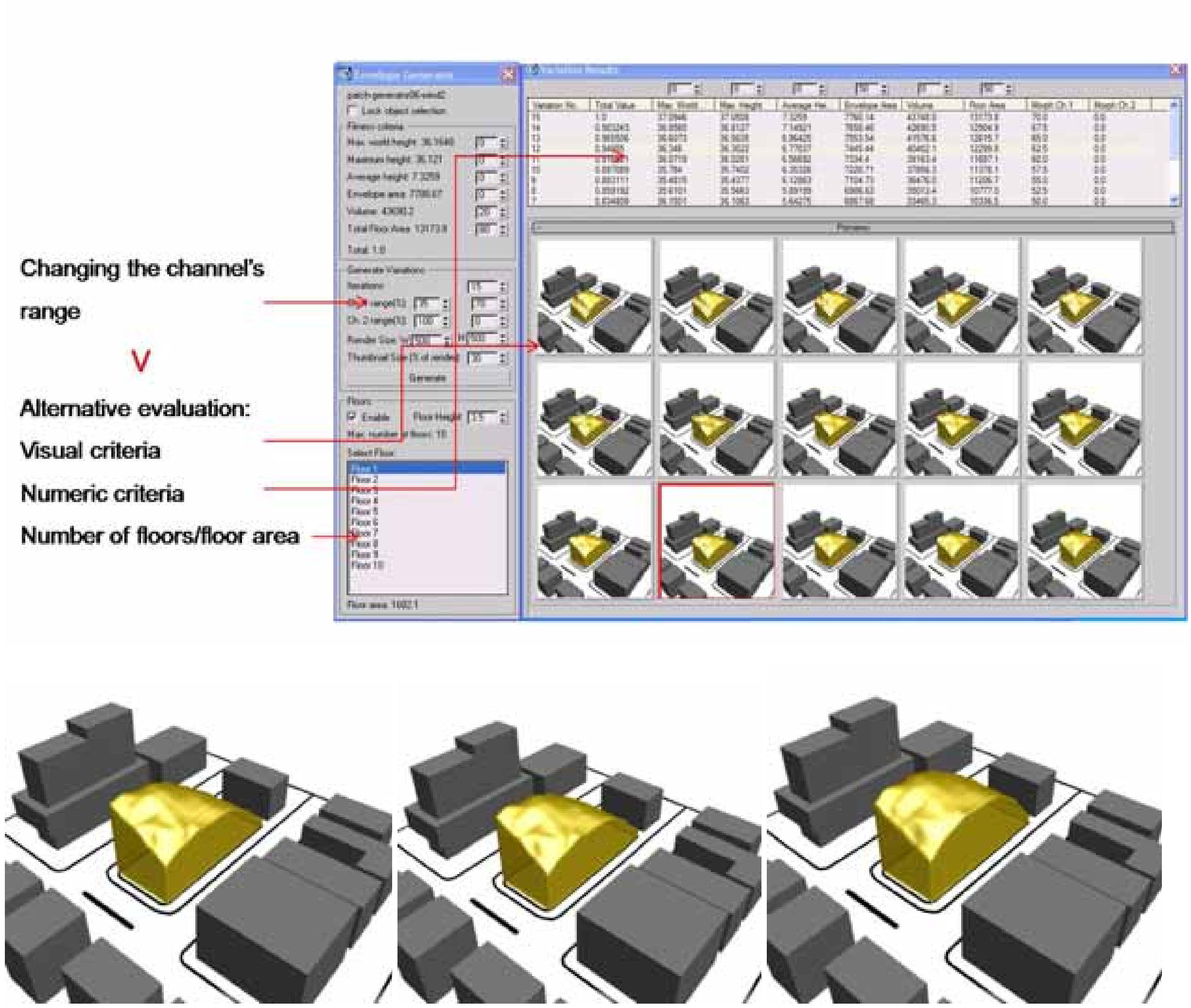


Figure 78 – initial generation and evaluation

The first type of the result evaluation examined the number of floors and the expected floor area of each alternative. It was noticed that starting from alternative 10 the building form volume could be divided to 10 levels instead of the 9 possible levels division of the other alternatives. From those alternatives that support 10 levels division only alternatives 12 and higher allow floor area of more than 300 m^2 in the 10th floor. The second type of criteria included the general grading (total value) that was calculated (see Figure 79). From the different variations that were generated, alternative number 15 received the best score (1.0), having also the highest results in all the criteria. Therefore, changing the fitness criteria ratios between different criteria will change the final grade but will not change the position/hierarchy of the alternatives in the score table. It was also noticed that there was slightly higher difference between the grades in alternatives 12 and 11 (about 0.3 against 0.15-0.2). This difference can be explained by the jump in floor area which is about 300 m^2 between all consecutive alternative except 11 to 12 were it jumps to 619 m^2.



Alt 11 Alt 12 Alt 15

Figure 79 – secondary generation (stage 2)

The third type of evaluation criteria was performed visually and involved aesthetics and visual comparison to the initial performance envelopes. The evaluation showed that alternatives 11 and 12 have the greatest potential in terms of their minimal deviation from both wind 2 m/s and solar right envelopes and the simplicity of the form. Following these results alternative 12 was selected to be further developed in the next stages.

1.1 Secondary generation/optimization process - stage 3

As mentioned earlier the third's stage's main aims in terms of the design process are to refine the selected alternative according to local conditions and examine performance information from new performance envelopes. This example also meant to demonstrate the possibility to work on selected parts of the form (local generation) as opposed to the previous stages where the whole form was modified (global generation). It concentrates on the local area facing Alenbi Street and Ben-Yehuda Street, where the future entrances of the building are planned. It utilizes 1m/sec wind performance envelope. The first step in local generation is to select the area that will be affected by the local re-generation process. Figure 80 presents the area defined in this example. The definition of the local area is based on a subjective preliminary design decision that the commercial area facing the street in the new building has to be around 20 m^2 and have double floor height. Nevertheless, it is possible to easily examine the implication of any other area definition simply by performing another generation run.

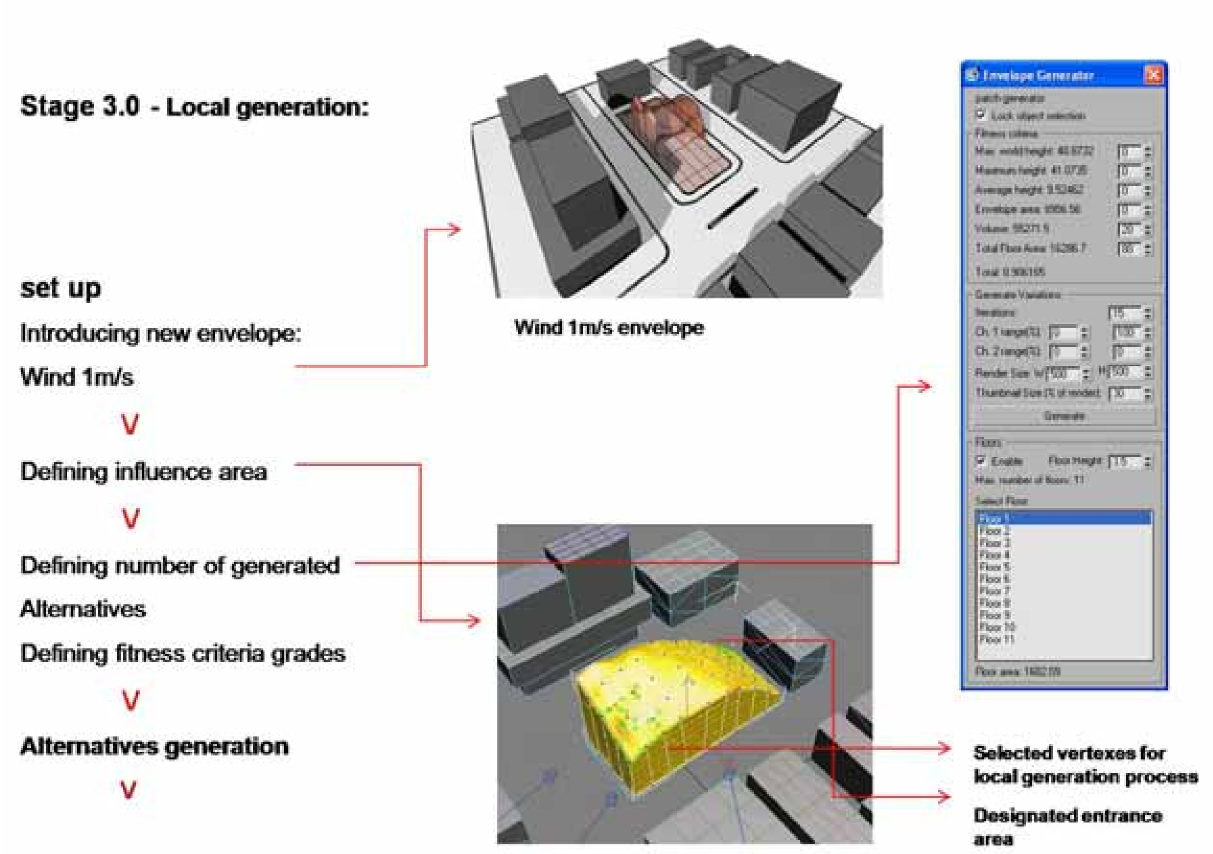
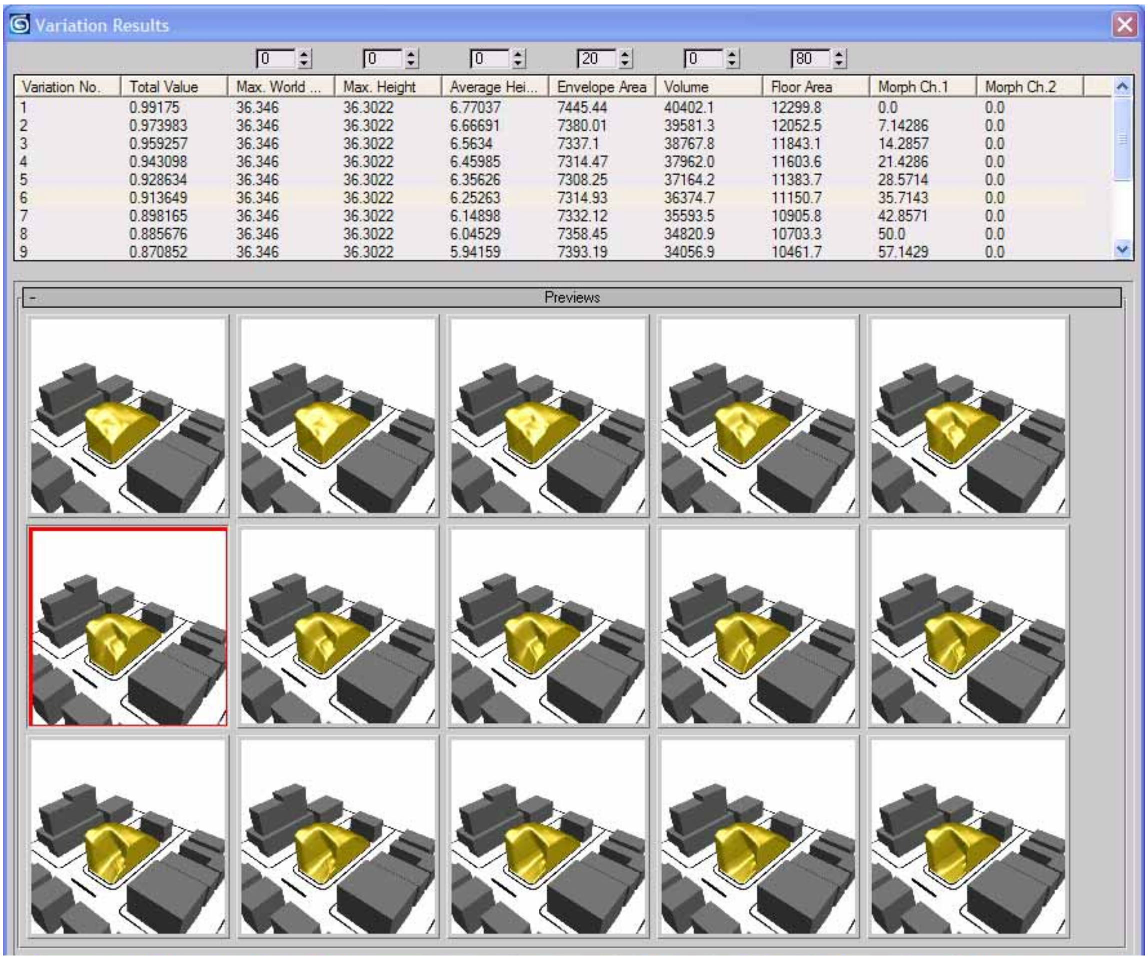


Figure 80 – stage 3 - Local area definition (wind 1m/sec envelope)

The results of the local generation process are presented in Figure 81. In order to differentiate between the results of the local generation two subjective design decisions were taken. The first was to search for a solution that will allow two double height stories in the entrance area in order to fit a future commercial use at street level. The second decision was to look for a solution that maximizes floor area in the upper floors of the future building. All of the variations presented in Figure 81 were examined using these criteria.



Variation No.	Total Value	Max. World ...	Max. Height	Average Hei...	Envelope Area	Volume	Floor Area	Morph Ch.1	Morph Ch.2
1	0.99175	36.346	36.3022	6.77037	7445.44	40402.1	12299.8	0.0	0.0
2	0.973983	36.346	36.3022	6.66691	7380.01	39581.3	12052.5	7.14286	0.0
3	0.959257	36.346	36.3022	6.5634	7337.1	38767.8	11843.1	14.2857	0.0
4	0.943098	36.346	36.3022	6.45985	7314.47	37962.0	11603.6	21.4286	0.0
5	0.928634	36.346	36.3022	6.35626	7308.25	37164.2	11383.7	28.5714	0.0
6	0.913649	36.346	36.3022	6.25263	7314.93	36374.7	11150.7	35.7143	0.0
7	0.898165	36.346	36.3022	6.14898	7332.12	35593.5	10905.8	42.8571	0.0
8	0.885676	36.346	36.3022	6.04529	7358.45	34820.9	10703.3	50.0	0.0
9	0.870852	36.346	36.3022	5.94159	7393.19	34056.9	10461.7	57.1429	0.0

Figure 81 – stage 3 - local generation results

Interestingly, changing the preference of the fitness criteria in the grading algorithm from the initial emphasis on maximizing floor area to maximizing envelope area as presented in Figure 82 changed the best graded alternative from alternative no. 1 to alternative 15. In the initial preference only 6 alternatives received grades higher than 0.9 while in the second preference all the alternatives are graded more than 0.9. The emphasis of envelope area can be important in designing surface area for cost related reasons, energy looses and placing photo voltaic cells or for commercial billboards.

Examining the generated forms pointed out that only one alternative (No. 10) offered the two needed commercial floors while partially embedding the local envelopes constraints. In the others the entrance area was either too low and did not allow double height commercial floors or too high which was too close to the starting form where the wind velocity at the entrance level was too strong, beyond the required limits.

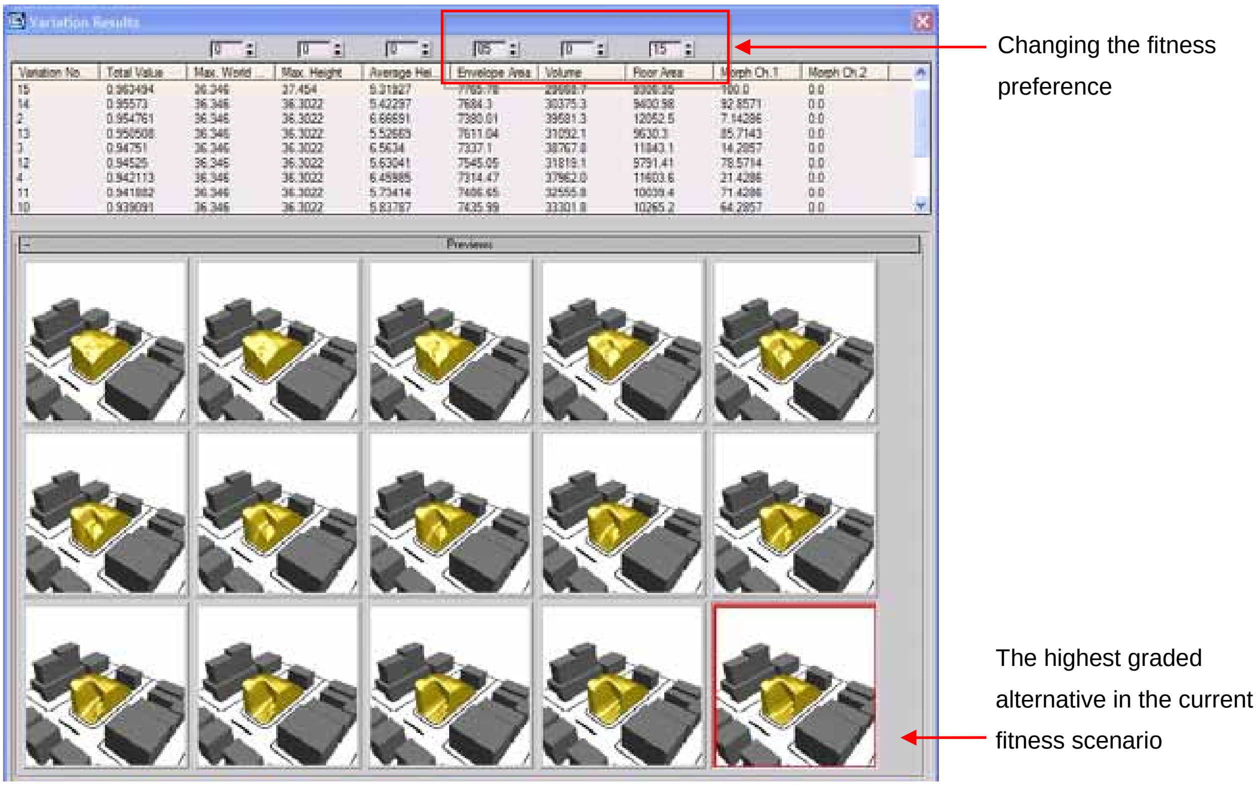


Figure 82 – changes in the grading preference

1.1.1 Second local optimization process

In terms of the design process this stage's main aim was to fine-tune part of the area that was designated as volume A in the initial generation. In terms of the case study, however, it had a second aim which was to prove the possibility to use more than one performance envelope in a local generation process. In order to perform this process the vertexes that were selected in the southern part of the building's form. The vertex selection is presented in Figure 83.

Following the area selection, local generation that utilizes solar rights envelope was performed. The solar rights envelope that was used kept the solar rights of the surrounding building according to the definitions of the brief. The generation results are shown in Figure 84. The criteria used to evaluate the results were the floor area (which was expected to decrease), the adherence to the envelopes (examined visually) and building envelope area. Alternative number 9 (marked with a red rectangle in Figure 84 and enlarged in Figure 85) was chosen. This alternative has the third largest floor area (10183 m^2) after alternatives 1 and 2 (10273 and 10263 m^2 respectively). However, alternatives 1 and 2 show a smaller effect in the new envelope (0% and 7.1% influence in comparison to 57.1% in alternative 9), thus is much closer to the desired performance at the designated area. In terms of an evaluation that combines floor area and envelope area (defining 50% influence for each criterion) alternative 9 has also the 3rd best grade (0.98) very close to alternatives 1 and 2 (1.0 and 0.99).

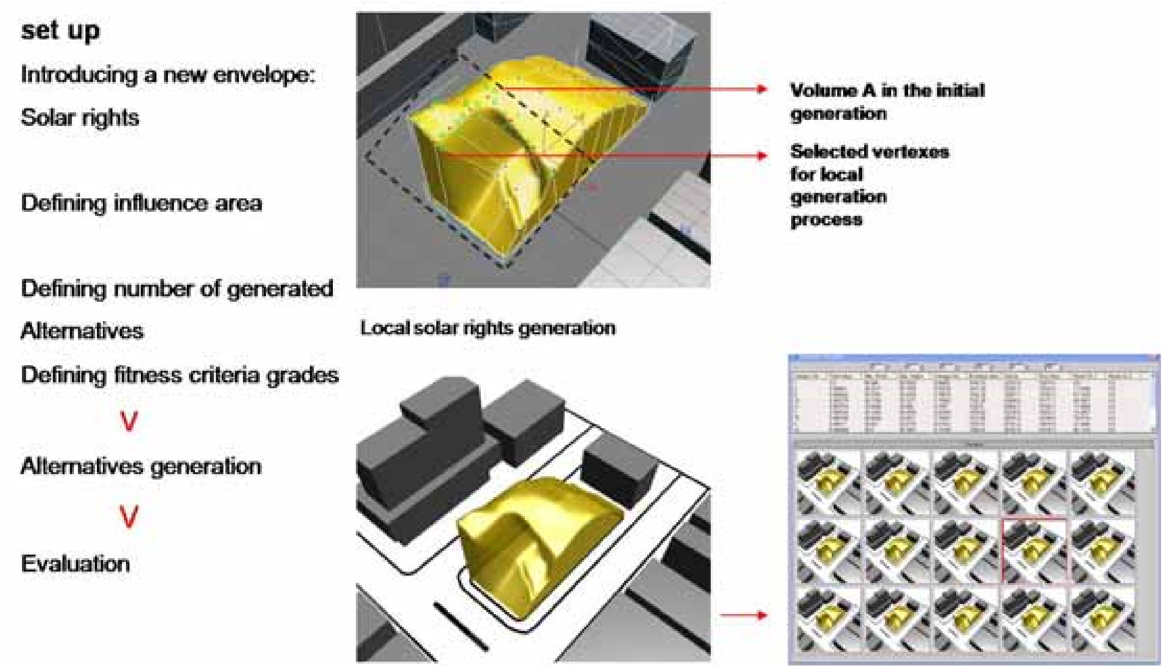
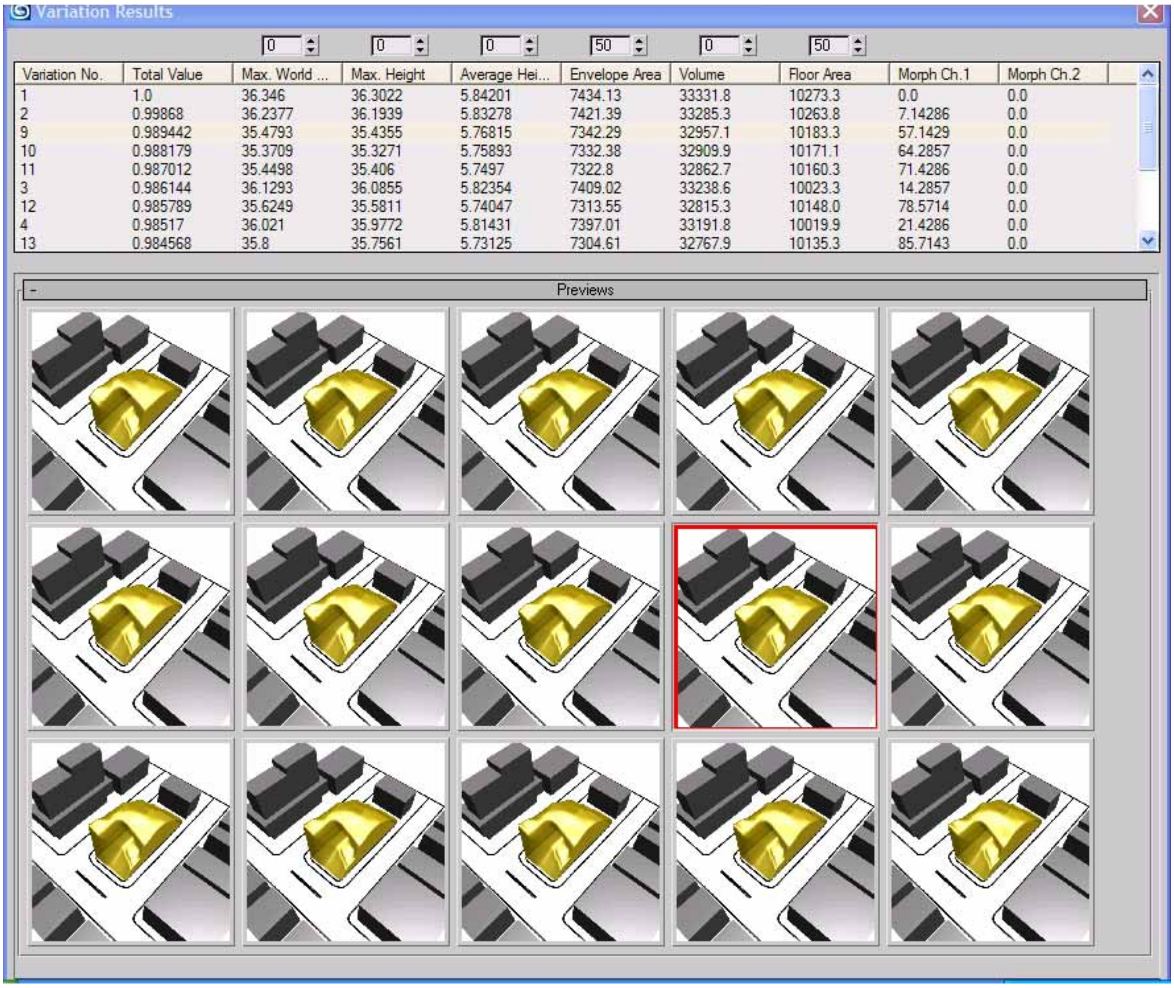


Figure 83 – second local area definition (Stage 3)



Variation No.	Total Value	Max. World ...	Max. Height	Average Hei...	Envelope Area	Volume	Floor Area	Morph Ch.1	Morph Ch.2
1	1.0	36.346	36.3022	5.84201	7434.13	33331.8	10273.3	0.0	0.0
2	0.99868	36.2377	36.1939	5.83278	7421.39	33285.3	10263.8	7.14286	0.0
9	0.989442	35.4793	35.4355	5.76815	7342.29	32957.1	10183.3	57.1429	0.0
10	0.988179	35.3709	35.3271	5.75893	7332.38	32909.9	10171.1	64.2857	0.0
11	0.987012	35.4498	35.406	5.7497	7322.8	32862.7	10160.3	71.4286	0.0
3	0.986144	36.1293	36.0855	5.82354	7409.02	33238.6	10023.3	14.2857	0.0
12	0.985789	35.6249	35.5811	5.74047	7313.55	32815.3	10148.0	78.5714	0.0
4	0.98517	36.021	35.9772	5.81431	7397.01	33191.8	10019.9	21.4286	0.0
13	0.984568	35.8	35.7561	5.73125	7304.61	32767.9	10135.3	85.7143	0.0

Figure 84 – stage 3 – Local generation results (solar rights envelope)

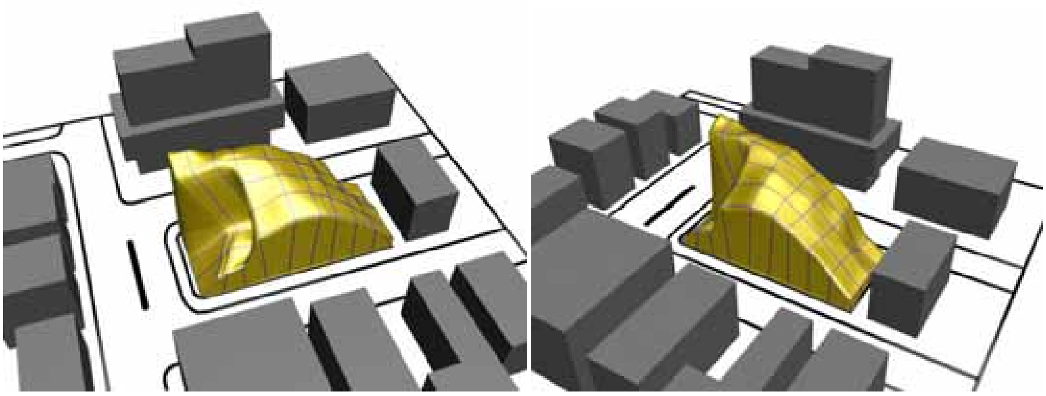


Figure 85 – chosen alternative

The case study brief is not detailed enough to decide whether this gap is big enough to stop the generation process at this stage, or, another local generation round is needed in order to make the end result more substantial in terms of the difference from other results. Nevertheless, the fact that the proposed design process is parametric suggests that it is possible at any time to add and subtract local and global form generation processes using various types of performance envelopes.

1.2 Results evaluation – stage 4

The generated form can be evaluated in two different ways; form based evaluation and performance based evaluation. Since the research concentrated on empirical performance criteria form based criteria will not be discussed in this example[74].

The performance based evaluation will inspect the alternative's adherence to solar rights and wind performance envelopes that were used to generate the solution. A quantitative evaluation of the selected alternative regarding solar rights impact was performed by comparing the sun exposure of the surrounding facades with and without the new building. The evaluation was done using the SHADING model that calculates the Geometrical Insolating Coefficient (GIC), which is the ratio between the insolated and the total surface area of the building facades (Yezioro and Shaviv, 1994)[75].

The results showed that the situation before and after the building's design proposal on the site in terms of the sun direct radiation and shade conditions in the designated period differs as follows (see Figure 86):

East facades - after around 12am the facades are self shade in all the examined period. From 10 am -11 am there is around 10% change in shade and from 13 pm-14 pm the

[74] See general discussion on form base and performance base design on chapter 7.

[75] 3ds Max7 that was used as an interface for developing the generation tool also offer simulation module that simulates shade. However, the user has to perform a render in order to see the shade condition while in SkechUp 5 the shade is displayed in real-time on the screen. For that reason we choose to perform the evaluation in SketchUp 5.

change grows to around 20%. South façades - in October and February the difference is close to 0%. In November and January the difference is around 15% and in December the difference is around 25%. West façades - until 11 am the façades are self shaded in all the examined period. Between 11 am-13 pm there is nearly no change, between 13 pm and 14 pm there is 0-5% additional shade.

These results demonstrate that the generated form closely adheres to the demands of the brief. The minor change in the amount of radiation in some of the facades between the initial situation (without the building) and the proposed situation can be explained by the fact that several performance aspects influenced the examined form's generation.

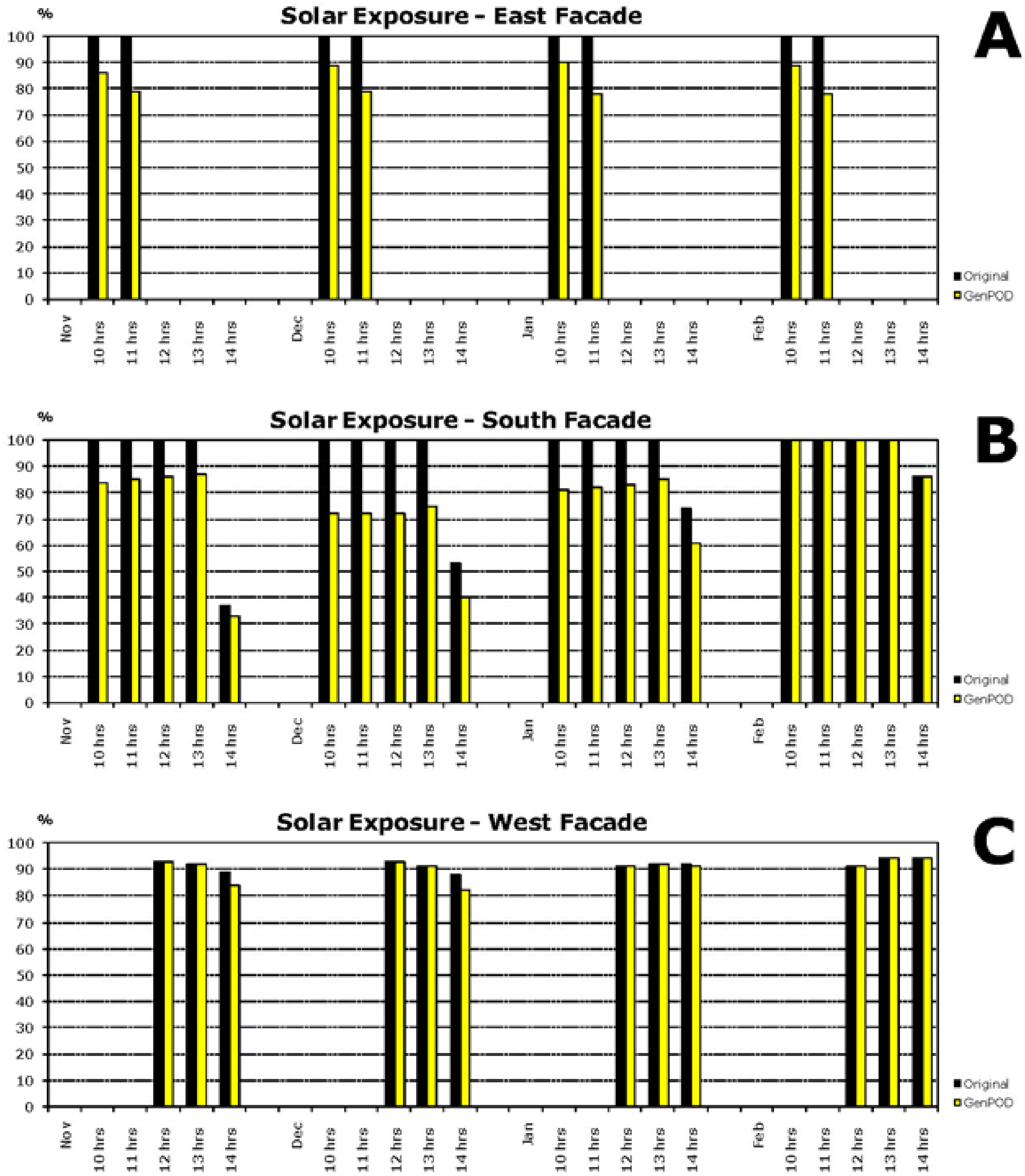


Figure 86 – quantitative evaluation of sun light and shade conditions with and without the new design proposal

The visual evaluation concentrated on the problematic period in term of violating the solar rights of the surrounding buildings by the generated form which was from 10:00am to 14:00pm from November to February. The images presented in Table 4 demonstrate that in all the examined cases in the predefined period the new building's initial form does not shade the surrounding building to a level higher than the ground floor level. Although the initial performance envelope was defined not to shade any floor the shade in the ground was expected due to the negotiation process with other performance envelopes. Therefore, this results correlate to the case study's design intentions.

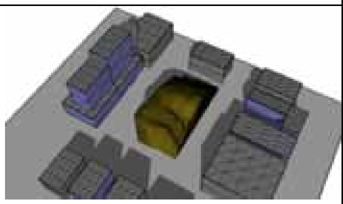
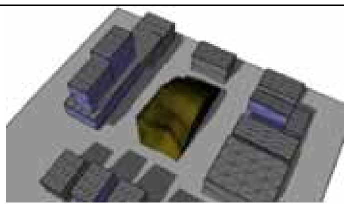
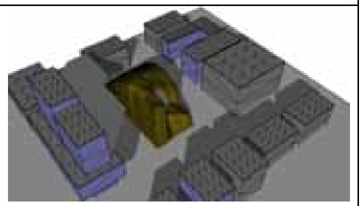
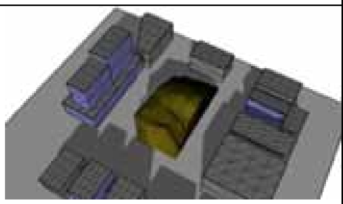
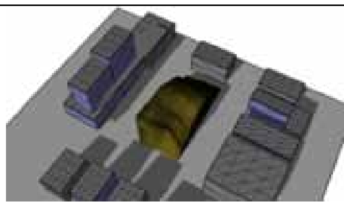
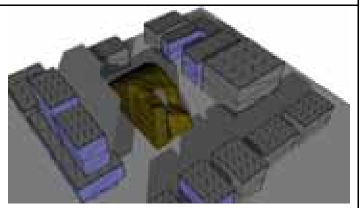
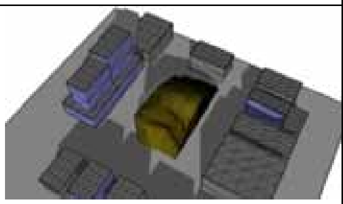
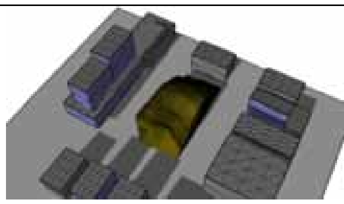
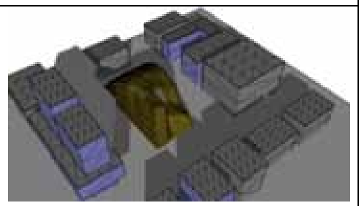
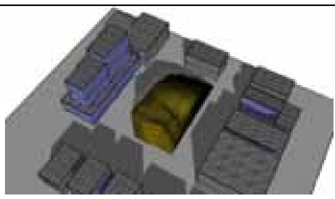
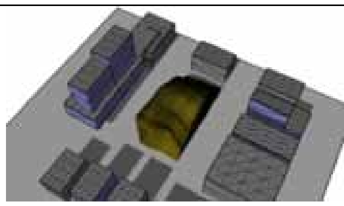
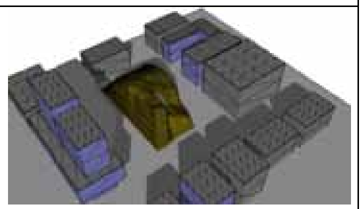
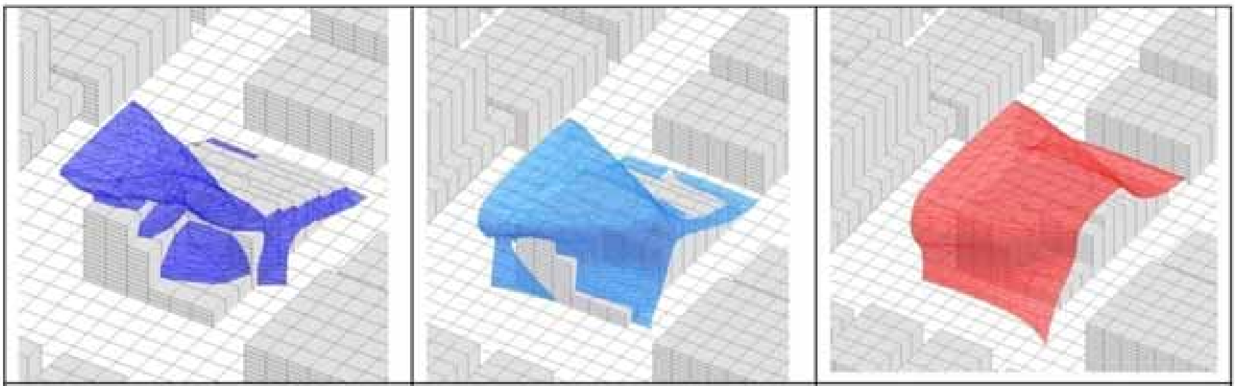
	10:00am	12:00am	14:00pm
November 1st			
December 1st			
January 1st			
February 1st			
	10:00am	12:00am	14:00pm

Table 4 – evaluation of the selected alternative – solar rights

The evaluation of wind velocity was done in ENVI-met 3.0. Since this software does not support importing 3-D complex models an approximate version of the final result was used. The results of the simulation results are presented in Table 5.

According to the results wind velocity envelopes of 1, 2 and 4 m/sec are very close to the initial envelopes used for the generation and generally follow the building volume. Both entrance areas have 1-2 m/s wind velocity which do not deviate from the brief and create comfortable access to the building. Also, the building's form does not deviate from the 4 m/s envelope and it follow its general form. However, small deviations occur at the upper areas of

1 m/s and 2 m/s envelopes. It is reasonable to assume that these deviations are caused by inaccuracies in the simplified model and the fact the envelope is highly curved in these areas[76]. This fact should be taken into consideration in terms of the level of confidence in the evaluation results in this area.



1 m/sec	2 m/sec	4 m/sec

Table 5 – evaluation of the selected alternative – wind

1.3 Model *constraints*

The degree of accuracy of the design method relays, among other, on the resolution of the performance envelopes that were used for the form generation process. These envelopes are made of surfaces between net of points with similar performance data regarding a certain design parameter. The amount of points and their distances determine the network accuracy. The accuracy of the performance envelope is enhanced with the increase in the number of points (vertexes) and the decrease in the distance between the points and vice versa. The case study model was constructed with a net of 10X10 vertexes. This net was modified to describe the generated envelope. The 10x10 net of vertexes was chosen after experiencing significant long processing time in the generation process of form based on higher resolution network. The second constrain has to do with the need to use several simulation software to generate the initial performance envelope. Since we were not aware of a single application that could generate different kinds of performance envelopes we had to generate each envelope in different software and import all of them to a single environment. Moreover, most of the simulation software do not support exporting the performance envelope. Consequently, exporting the performance envelope to single software is time consuming, reduces the envelopes accuracy (when tracing process is needed) and requires high level of compatibility between the applications.

[76] Envi-met do not support at the moment building 3-D models using curved lines. This fact should be taken into consideration in terms of the level of confidence in these results.

Appendix D: Beaufort wind scale

Source: BRE Digest 390, 1994. “Wind Around Tall Buildings”. Building Research Establishment. Garston, Watford, WD2 7JR.

Beaufort Scale	**Description**	**Mean wind speed range (m/s) at 10 m**	Mean wind speed range **(m/s) at 2 m**	**Effects**
B0	Calm	0 - 0.2	0.0 - 0.15	
B1	Light air	0.3 - 1.5	0.22 - 1.1	No noticeable wind.
B2	Light breeze	1.6 – 3.3	1.2 - 2.5	Wind felt on face.
B3	Gentle breeze	3.4 – 5.4	2.6 - 4.0	Wind extends light flag.
B4	Moderate breeze	5.5 – 7.9	4.1 - 5.9	Raises dust and loose paper. Hair disarranged, clothing flaps.
B5	Fresh breeze	8.0 – 10.7	6.0 - 8.0	Limit of agreeable wind.
B6	Strong breeze	10.8 – 13.8	8.1 - 10.4	Umbrellas used with difficulty. Force of the wind felt on the body. Wind noisy, frequent blinking.
B7	Near gale	13.9 – 17.1	10.5 - 12.8	Inconvenience felt when walking; difficult to walk steadily. Hair blown straight.
B8	Gale	17.2 – 20.7	12.9 - 15.5	Generally impedes progress; walking difficult to control. Great difficulty with balance in gusts.
B9	Strong gale	20.8 – 24.4	15.6 - 18.3	People blown over by gusts. Impossible to face wind; ear ache, headache, breathing difficult. Some structural damage occurs: falling roof tiles, tree branches etc, hazardous for pedestrians.
B10	Storm	24.5 – 28.4	18.4 - 21.3	Seldom experienced inland. Trees uprooted; considerable structural damage occurs.

Table 6 – summary of wind effects on pedestrians, based on the Beaufort Scale of wind force[1].

Relative comfort					
Activity	**Areas applicable**	**Perceptible**	**Tolerable**	**Unpleasant**	**Dangerous**
Walking fast	Pavements	**B5** (8.0 - 10.7 m/s)	B6 (10.8 - 13.8 m/s)	B7 (13.9 - 17.1 m/s)	B8 (17.2 - 20.7 m/s)
Strolling	Parks, entrances	**B4** (5.5 – 7.9 m/s)	B5	B6	B8
Standing, sitting:					

short exposure	Parks, plazas	**B3** (3.4 – 5.4 m/s)	B4	B5	B8
long exposure	Street cafes, theatres	**B2** (1.6 – 3.3 m/s)	B3	B4	B8
Acceptable is speed occurs less than:			once/week	once/month	once/year

Table 7 – comfort and safety criteria[2].

---

1 Wind speeds measured at 10 m above ground are reduced to about 75% at head height. The decimal values of wind speed arise from speeds being expressed in knots in the original tabulations (1 m/s = 1.94 knots = 2.24 mph = 3.6 Km/h).

2 Units are Beaufort numbers. Temperatures above 10oC. At lower temperatures, the relative comfort level is reduced by one Beaufort number for a 20oC fall in temperature.

בניית העולם הדיגיטאלי – שיטות תכנון אדריכלי המבוססות על שימוש בכלים דיגיטאליים - ייצור ואופטימיזציה של צורה המבוססת על ביצועים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

יעקב יאשה גרובמן

הוגש לסנט הטכניון - מכון טכנולוגי לישראל
אלול תשס"ח, חיפה. ספטמבר 2008

מוקדש לאביגיל ואמה

הבעות תודה

המחקר נעשה בהנחיית ד"ר אברהם יזיאורו וד"ר גדי קפלוטו בפקולטה לארכיטקטורה ובינוי ערים.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תקציר מורחב

אנו חיים במרכזה של מהפכה תרבותית הנובעת מהתקדמות משמעותית בטכנולוגיות התקשורת והאניפורמציה. בתקופה זו מוגדרים מחדש האופן בו אנו מתקשרים, עובדים, צורכים ומבלים את זמננו החופשי. במסגרת תקופה זו של שינויים, הדיסציפלינה האדריכלית מגדירה מחדש את גבולותיה, שפת הייצוג, גישתה לתהליך התכנון האדריכלי בכלל, והאופן בו נוצרת ומתפתחת הצורה האדריכלית בפרט. בפעם השניה בהיסטוריה של האדריכלות המודרנית, בעקבות התקדים של המהפכה התעשייתית במאה ה- 19, התקדמות המדע והטכנולוגיה, מקדימים את השימוש שנעשה בטכנולוגיה זו בפרקטיקה האדריכלית. בניגוד למהפכה ה"מכאנית" הקודמת שכונתה "עידן המכונה הראשון" והתבססה על תיעוש ומיכון, כיום, המהפכה היא אלקטרונית, או דיגיטאלית, ומתבססת בעיקר על התפתחות המחשב והשימוש הנעשה בו בדיסציפלינה האדריכלית. השימוש במחשב לתכנון וייצור אדריכלי התחיל על בסיס שאינו מונע בצורך אלא בזמינות טכנולוגית ושימוש בכלים שיוצרו בתחילה לדיסיפלינות אחרות כתעשיות המטוסים, המכוניות והקולנוע. לכן, קיימים עדיין פערים משמעותיים בין הבנתינו את יכולות המחשב לאופן בו נעשה במחשב שימוש בתכנון (design) אדריכלי (בניגוד לשרטוט-drafting) והשלכות השפעת המחשב על תהליך התכנון האדריכלי ותוצאותיו עדיין לא נבחנו במלואן.
מחקר זה בוחן את השינויים בתהליך התכנון האדריכלי הנובעים מהשימוש במחשב. המחקר מתמקד בעיקר ביצירה (generation), סימולציה והערכה (evaluation) של צורה האדריכלית מבוססת מחשב. המחקר מציע מודל חדש של תכנון גנרטיבי מבוסס ביצועים (GenPOD – Generative Performance Oriented Design) העושה שימוש במעטפות ביצועים כבסיס ליצירת צורה אדריכלית.

כשלב מקדים, פותח במסגרת המחקר מאגר מידע ממוחשב של פרוייקטים באדריכלות דיגיטאלית או אדריכלות מבוססת מחשב. מאגר זה כלל פרויקטים אדריכליים שתוכננו ו/או ניבנו החל מתחילת שנות התשעים של המאה העשרים העונים להגדרה חדשה של אדריכלות דיגיטאלית שנוסחה במחקר. המאגר נועד, בין השאר, לבחון באופן כמותי את השימוש במחשב לתכנון מבוסס מחשב באדריכלות. מניתוח המידע הוסקו מסקנות לגבי תמורות בסוגי פרוייקטים, עלויות, מיקום גאוגרפי ופרטים נוספים שחידדו את התובנות לגבי כיווני ההתפתחות הקיימים והצפויים באדריכלות מבוססת מחשב.

לאחר השלב המקדים בחן המחקר את השפעות המחשב על התכנון האדריכלי ובעיקר את השינויים בתהליך התכנון האדריכלי, מתודות תכנון, וכלים וגישות ליצור, סימולציה והערכה מבוססות מחשב של הצורה האדריכלית. במסגרת זו נערכה בחינה ביקורתית של גישות ומודלים שפותחו במהלך השנים לשימוש במחשב בפרקטיקה ובמחקר האקדמי בדיסציפלינה האדריכלית ודיסציפלינות אחרות. היכולת לעבור לתכנון מבוסס ביצועים נובעת, בין השאר, מהאפשרות לקשר מידע אמפירי, כמותי לצורה האדריכלית במחשב (בנוסף על נתונים גאומטריים) והאפשרות לפיתוח ושימוש במודלים לסימולציה, הערכה ואופטימיזציה כחלק מהכלים בשימוש אדריכליים ובתהליך הפיתוח של הצורה. מידע זה כולל, בין השאר, התיחסות לביצועים בנושאים כגון רוח, אור/צל, אקוסטיקה, חוזק ויציבות מבנה, תנועה, זויות ראיה (מבטים) ואנרגיה. שימוש במודלים אלה מתחילת תהליך התכנון אדריכלי מעלה את הביצועים של הצורה האדריכלית, מאפשר התאמה גבוהה יותר בין הדרישות הפרוגרמטיות והתוצר האדריכלי הסופי ויכול לתרום לחסכון במשאבים המושקעים בבניה ובאחזקת הבנין.
נדונו גם משמעויות נוספות למעבר לתכנון פרמטרי והעליה בכמות האינפורמציה האמפירית הקיימת על הצורה האדריכלית כעליה ברמת השליטה של האדריכל בצורה, המעבר לתכנון לא לינארי והתפתחות מתודות תכנון מבוססות מחשב.
מסקנות שלב זה גובשו לכדי הגדרת פערים בין השימוש שנעשה במחשב לתכנון אדריכלי ע"י אדריכלים לבין אפשרויות השימוש במחשב שעלו בבחינת המודלים הקיימים.

על בסיס מסקנות אלו מציע המחקר להשתמש במעטפות ביצועים (performance envelopes) כבסיס למודל חדש של תכנון גנרטיבי מבוסס ביצועים. מעטפות ביצועים הינן משטחים הנוצרים מחיבור נקודות בחלל להם

ערכים זהים של ביצועים מסוג מסוים (מעטפת ביצועים של רוח במהירות 5 מטר לשניה לדוגמא תוגדר עפ"י שיטה זו באמצעות חיבור כל הנקודות בחלל בהם קיימת מהירות רוח זו). במסגרת המחקר נבחנו וסווגו סוגי ביצועים שונים אותם ניתן להציג כמעטפות והוגדרו אפשרויות השימוש בתכנון לכל אחד מהם. בניגוד לגישות מסורתיות בתכנון אדריכלי, בהן נעשה שימוש בתהליך סימולציה בשלבים המתקדמים של תהליך התכנון על מנת לבדוק האם התוצר האדריכלי עומד בדרישות הביצועים שהוגדרו לפרוייקט, המודל החדש, המורכב ממודל תאורטי ואב טיפוס לכלי תוכנה מעשי, מציע להשתמש במעטפות ביצועים רבות באופן גנרטיבי החל מתחילת תהליך התכנון ובכך ליצור צורה אדריכלית העומדת בהגדרה בדרישות הביצועים של הפרויקט המופנמות במעטפות הביצועים בהן השתמשו ליצור הצורה.

המודל המוצע מאפשר יצירה של מרחב תכנון ו/או צורה אדריכלית ראשונית לפרויקט העונים על הדרישות התפקודיות שהוגדרו מראש, באמצעות משא ומתן צורני המתבסס על תהליך מורפינג (Morphing) של מעטפת ביצועים אחת או יותר, המייצגות תחום ביצועים זהה או שונה. תהליך התכנון המוצע הינו פרמטרי ורקורסיבי. הוא מאפשר שילוב של מספר סבבי יצירה תוך שימוש במספר מעטפות ביצועים המשפיעות על הצורה כולה או רק על חלק נבחר מהצורה עפ"י הגדרות פרוגרמטיות ו/או העדפות תכנוניות. בנוסף, המודל מאפשר חזרה לכל שלב בתהליך היצירה ובחינת השפעת השינוי של כל פרמטר בו השתמשו ליצירת והערכת הצורה על התוצאה הסופית. המודל מייצר בכל שלב מספר אלטרנטיבות בהתאם להגדרת המתכנן. האלטרנטיבות נבחנות בעזרת אלגוריתם להערכה חדש המשתמש בקריטריוני התאמה (Fitness criteria) שאינם קשורים בהכרח לפרמטרים בהם נעשה שימוש בתהליך היצירה (לפיהם נוצרו המעטפות) ומהווים שכבה נוספת של נתונים (מעבר לנתונים הקשורים בביצועים) אותם מקבל המתכנן כדרך להערכה של כל אחת מהאלטרנטיבות המוצעות.

על מנת לבחור את האלטרנטיבה המתאימה ביותר מגדיר המתכנן לכל קריטריון התאמה את רמת חשיבותו באופן מספרי יחסית לקרטריונים האחרים. בהתאם להגדרה זו מסודרים נתוני קריטריוני ההתאמה של כל אחת מהאלטרנטיבות ומחושב להן ציון משוכלל כללי. שינוי בחשיבות של כל קרטריון ההתאמה יוביל באופן מיידי לשינויים על הציון הכללי של כל אלטרנטיבה.

המודל GenPOD נבדק במקרה מבחן להערכת היתכנות וישימות השימוש בכלי ושיטת התכנון המוצעים בתכנון מבנה משרדים בסביבה עירונית באתר במרכז תל אביב. במסגרת מקרה המבחן הודגמה היכולת להשתמש במספר מעטפות ביצועים ליצור מרחב תכנון מבוסס ביצועים ו/או צורה ראשונית לבנין המוצע, באופן העונה להגדרות הביצועים שנקבעו בפגורמה לבנין. מקרה המבחן הראה, בין השאר, כי השימוש במעטפות ביצועים בתכנון אדריכלי יכול להעלות את ההתאמה לדרישת הביצועים הכללית המוגדרת לבנין. תוצאה זו מושגת באמצעות הגדלת רמת המידע המבוסס ביצועים בה נעשה שימוש ליצירת הצורה האדריכלית תוך שימוש בשילוב מודע של העדפות סוביקטיביות של המתכנן ומידע אמפירי המתייחס לביצועים.

השימוש במחשב לתכנון אדריכלי מבוסס ביצועים באמצעות מודלים כדוגמת GenPOD הינו צעד חשוב לקראת התאמת השינויים הטכנולוגיים לתוצרים האדריכליים בעולם בו משאבי הטבע הולכים ומתכלים ושלב נוסף בדרך לסביבה בת קיימה.