Computer Oriented Design Methods in the Digital Design Studio

Yasha J. Grobman, Eran Neuman

Yasha J. Grobman, Faculty of Architecture and Town Planning, Technion, Israel Institute of Technology.

Eran Neuman, Faculty of Architecture and Town Planning, Technion, Israel Institute of Technology.

abstract

In opposition to the common condition in the architectural office, the digital design studio in academic environments is free of market and commercial constraints and allows exploring and experimenting with software in architecture. The studio proposes an ideal environment in which boundaries can be stretched and software can be exploited to examine the changes in the design process induced by the introduction of computer. The design studio culture allows developing new digital design procedures and methods, intervening in software trough writing code or scripting in order to articulate a new type of architecture.

T_CODE[1] (Technion_Computer Oriented DEsign) is an experimental computer oriented design research laboratory based at the Faculty of Architecture and Town Planning at the Technion – Israel Institute of Technology. The laboratory offers an environment for advanced research in design, architecture and urbanism. Its main explorative platform is a final project studio, which includes academic research and extracurricular activities.

In the studio, students are guided to develop digital design methods that would reflect, intervene and transform existing architectural conditions. To that end, the underlying perception of the studio maintains a tight connection between the design method and the end product. The means to conceive architecture necessarily influences the features of the end product. In that fashion, the digital tools were considered as what Gilles Deleuze and Félix Guattari termed "abstract machines" – a complex operation that trespasses a phenomenon into a new and singular condition of consistency.

The article will present the possibilities of examining digital tools exiting in the design laboratory. It will concentrate on the contribution of computer tools in developing new design methods and their potential for the academic and professorial realms.

Keywords: Computer Oriented Design, Design Methods, Scripting, Smart Forms

Acknowledgments: The case study projects presented in this article were done by the students Yaniv Ophir, Diana Voyshvillo, Zhenya Khilkevich, Moshit Fidelman and Guy Austern.

[1] http://tx.technion.ac.il/~tcode/

Introduction

It seems that Marshall McLuhan's argument in Understating Media – namely, that "the medium is the message" – has never been more relevant for the architectural discipline as in recent years (McLuhan, 1994). Digital media not only reshapes architecture as media, but it becomes the message, extending its effect on architecture beyond the function of digitization as a tool. Referring to McLuhan's notion, T_CODE examines the ways in which digital tools redefine architectural thinking, design and production as a whole, and become a message.

The examination takes place concurrently in two parallel domains. The first examines the influence of computers on the urban environment and the way people "occupy" it, and the second looks into changes within the architectural practice and especially into changes in computer oriented design methods, or the way architects' design with computers.

In terms of the changes in the urban environment, in our view, the implementation of digital tools in architectural design will not necessarily result in shifting towards digital environments. Rather, it suggests that the end product would reflect a digital culture in a way clearly indicating that the design was conceived and developed with digital tools and would not have been possible to design otherwise. In that sense, the digital tools are considered to be what Gilles Deleuze and Félix Guattari termed "abstract machines" – a complex operation that transforms a phenomenon into a new and singular condition of consistency (Deleuze and Guattari, 1987). In that respect, digitization does not refute old occupations in the discipline. Nor does it reduce the complexity of architecture into singular articulations – formal, material or textural. Digitization offers architecture a means of rethinking its intra-disciplinary circumstances, hybridizing old notions with contemporary articulations.

In terms of computer based design methods, our main motivation is to suggest ways for further incorporating the architecture of computation into the computation of architecture (Chu, 2005). We use the emerging definitions of the new information structures[2] as a data

[2] The notion of the new information structures in this context relates to Steele's (1998) discussion on the connection between the changing information structure and the search for new computer-based design methods in the research studio.

source in our search for ways to use computers in design, which go beyond the limits of the existing computer tools. In that respect, intervening in design software with code could elevate a new equilibrium between the quantitative nature of the computer and the qualitative nature of human perception is proposed.

New possibilities in computer oriented design methods

According to Cross (1989), design methods are “any procedures, techniques, aids or ‘tools’ for designing". Cross defines two common features for the innumerable existing design methods: They formalize certain procedures of design and they externalize design thinking in terms of making the designer use 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 a design method is intended to have two effects: to reduce the amount of design error, redesign and delay; and to make possible more imaginative and advanced designs. Computer based design methods do no deviate from these definitions. The quantitative nature of the computer and its superior processing power allows the generation and deformation of complex forms incomprehensible to the human mind[3]. This could be done either intuitively on the computer screen or developed according to concrete design methods.

Although considered by some scholars as a paradigmatic change[4], the introduction of computers did not negate existing design methods. On the contrary, as Negroponte (1970) envisioned already in 1970 it enhanced existing and introduced new ones.

Goldschmidt (2001) posits that the “design methods movement” that tried to develop a design science is still far from attaining its 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

---

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

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

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”[5]. Kroes (2002) states that design methods aim to improve the design process, which is the reason it has always focused on the nature of the process and not the design product. 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”. The orientation toward the product is highly significant in the case of computer-oriented design, especially when it is associated with computer-based manufacturing. Therefore, the interest of our laboratory continues 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-oriented design has a clear product orientation and thus cannot be considered within the existing process-oriented design method.

Thus, the chosen approach for examining and developing computer oriented design methods in T_CODE 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 existing and future technology within the design process.

Two main directions for computer oriented design methods, which look beyond the discourse concerning the new capabilities in form manipulation and representation were pursued:

---

[5] Archer (1979, in 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 problem or an ill-structured problem, "one or more of these constituents is either unknown or ambiguous” (Goldschmidt, 1997).

Smart forms – smart form refers to the possibility to define intricate relationships between forms or group of forms using code. It introduces, among other, the notion of form generation (Morphogenesis), and search for new complex dependencies, self organizing rules that can be based on performance in its wider sense (which includes empiric physical and also cognitive and perceptual performance), or search for new complex formal expression.

Code based design methods could bring up several issues regarding the role and education of contemporary architects. Among these issues is the question or "danger" of loosing control over the design, as pointed out by Terzidis (2006) and aspired by Eisenman[6]. Another important issue in this context, which could not be developed within the scope of this article, is related to the education and training of future architects in association with the emerging need to develop a computer code.

Simulation and optimization – Advances in both architectural and simulation software make it possible to embed simulation in the architectural design process. Moreover, simulation process can be used as part of an optimization process where the architectural form is optimized according to a selected target function. Up-to-date simulation and optimization processes were performed by highly professional expects. Today, the emerging connectivity between software and the redundancy of information calls for examining a new type of employing simulation and optimization by designers. This does not necessarily negate professional optimization and simulation by consultants, but calls for a step towards a new kind of performance oriented design.

Case studies

Each academic year, we investigate specific aspect of the digital revolution influence on the architectural culture and discourse. The pedagogical dimension of our research laboratory capitalizes on the digital media and uses them as a device for data-sharing, collection and analysis of information in and about digital architecture.

During the 2004-05 academic year, the studio addressed the question of the architectural

[6] By canceling the limiting effect of the human eye (Eisenman, 1992)

form in relation to spatial organization of new information structures. Intensifying formal architectural properties – surface, volume, scale – in the studio, students examined changes in traditional programs introduced by social and technological changes and developed new strategies for the articulation of programmatic organization. The overall perception was that form should correspond to diagrammatic analysis of the changes in the information structure of the examined building type, and generate complex programmatic solutions in increasingly complex designs. The strategies developed in the laboratory harnessed the digital tool to simultaneously generate programmatic complexities and extreme formalism, as a codependent procedures.

For example, one of the projects in the laboratory, HybridHiRiz, examined the spatial organization in a high-rise building using a scripted agents system to determine the relations between the building's form and morphology (i.e. surface and volume) and its use[7]. Each set of agents was programmed according to a specific use. Applying digital procedures of injection and projection, the various particle sets were organized in relation to their own use and in relation to each other, resulting in complex morphology (see Figure 1).

The outcome was a proposal for a high-rise building based on a non hierarchical configuration and distribution of its program throughout the structure. Historically, high-rise buildings were based on a division of the programmatic components according to levels of public use of each and every section. The most extensive spaces in terms of public access – commercial spaces, for examples – were located in the lowest section of the high-rise. Second in rank in terms of public accessibility – offices – were located on top of the commercial spaces. Finally, on top of the two programs, residential spaces were located.

[7] HybridHiRiz project was designed by the student Yaniv Ophir during the academic year 2004-2005.

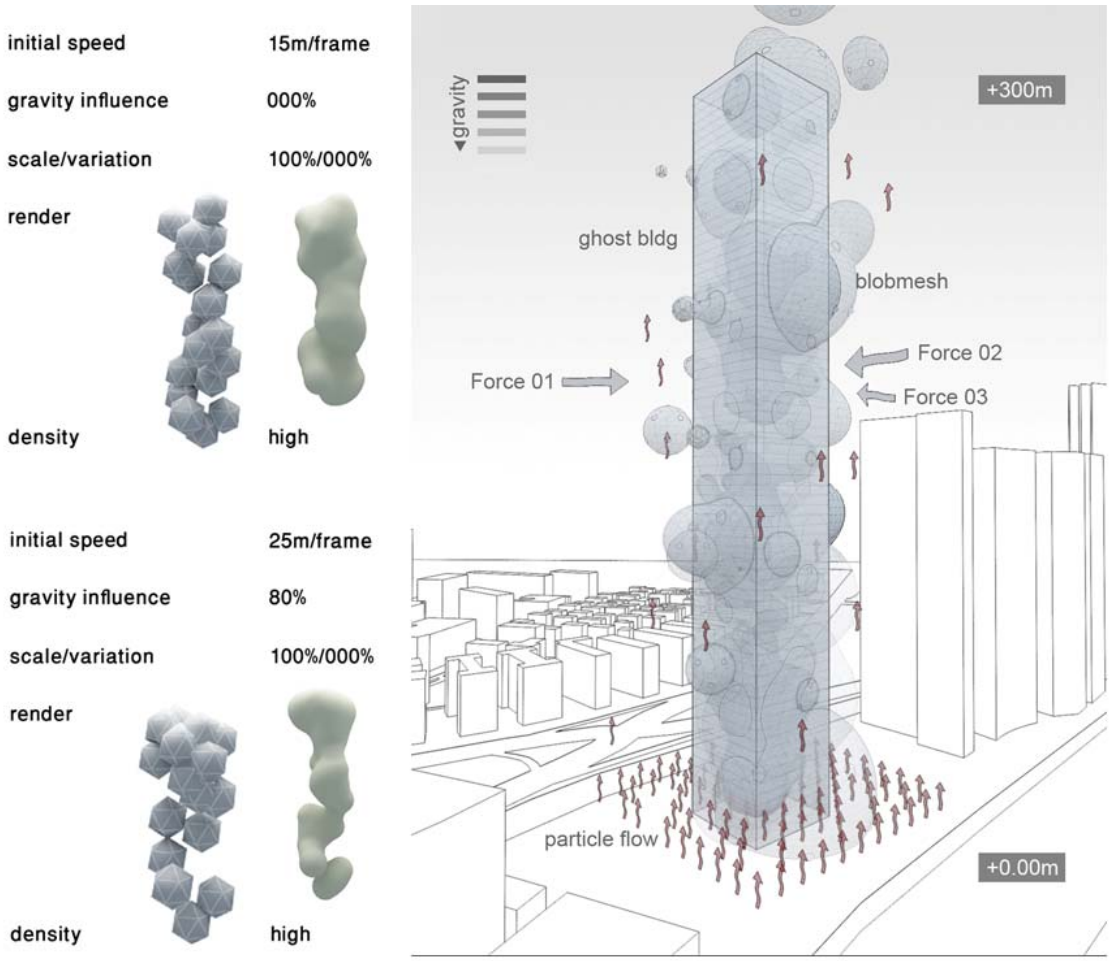

Figure 1 – Using practical simulation based on gravitational scripted model to determine program distribution in high-rise building (image: Yaniv Ophir)

Examining the relationships between accessibility, privacy and programmatic distribution, the project suggested rethinking the relations between the three programmatic components. The implementation of a scripted agent system yields a rearrangement of the components in a non-traditional fashion while maintaining the basic needs of privacy and accessibility (see Figure 2).

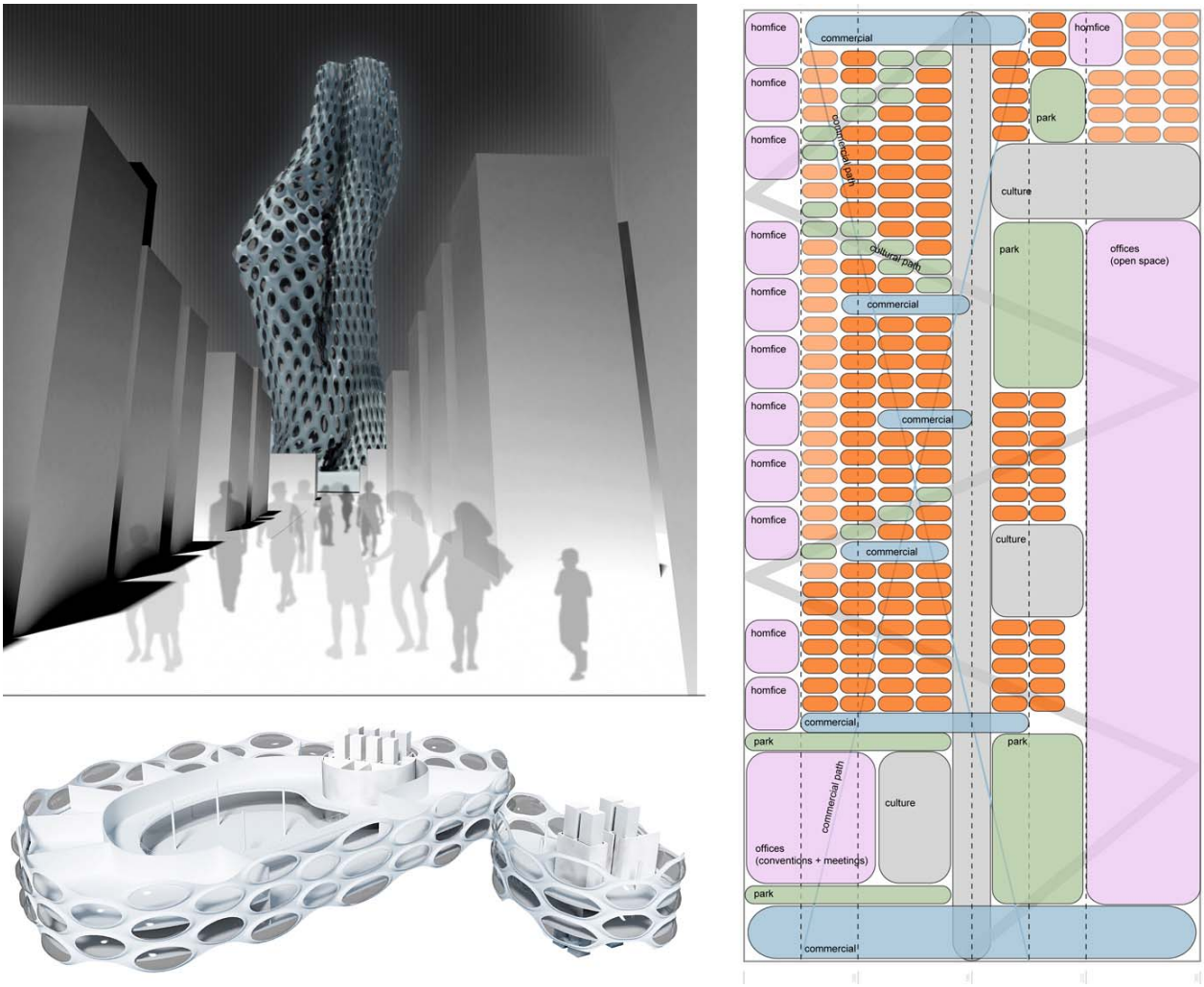

Figure 2 – HybridHiRiz project - elevation and isometric section of a floor plan (left), non hierarchical configuration and distribution of its program (right) (image: Yaniv Ophir).

During the 2005-06 academic year, the studio addressed the question of typology in the digital era. With the shift from mass standardization to mass customization, typology as a means of defining, categorizing and producing architecture came into question. Increasingly, each architectural project became a specified problem that could not be resolved by applying architectural types. This process was reinforced by the ability to design and produce custom-made architecture at the same cost as that of standard designs. Yet, even within the logic of transformation from mass standardization to mass customization and the decline of typologies as a means of architectural classification, we can still recognize buildings according to their use and type. The studio addressed this

paradox and tried to develop design tools that would operate in this tension. In designing a small-scale airport, one of the studio's projects (Airport) tried to negotiate between the airport as a building type and specified movement flows[8]. Developing codes of movement through localities of attractors and reflectors, the project referred to the design of the airport type as a codified entity which, on the one hand, tries to crystallize its generic type, but, on the other hand, creates a singular articulation of the same specific type.

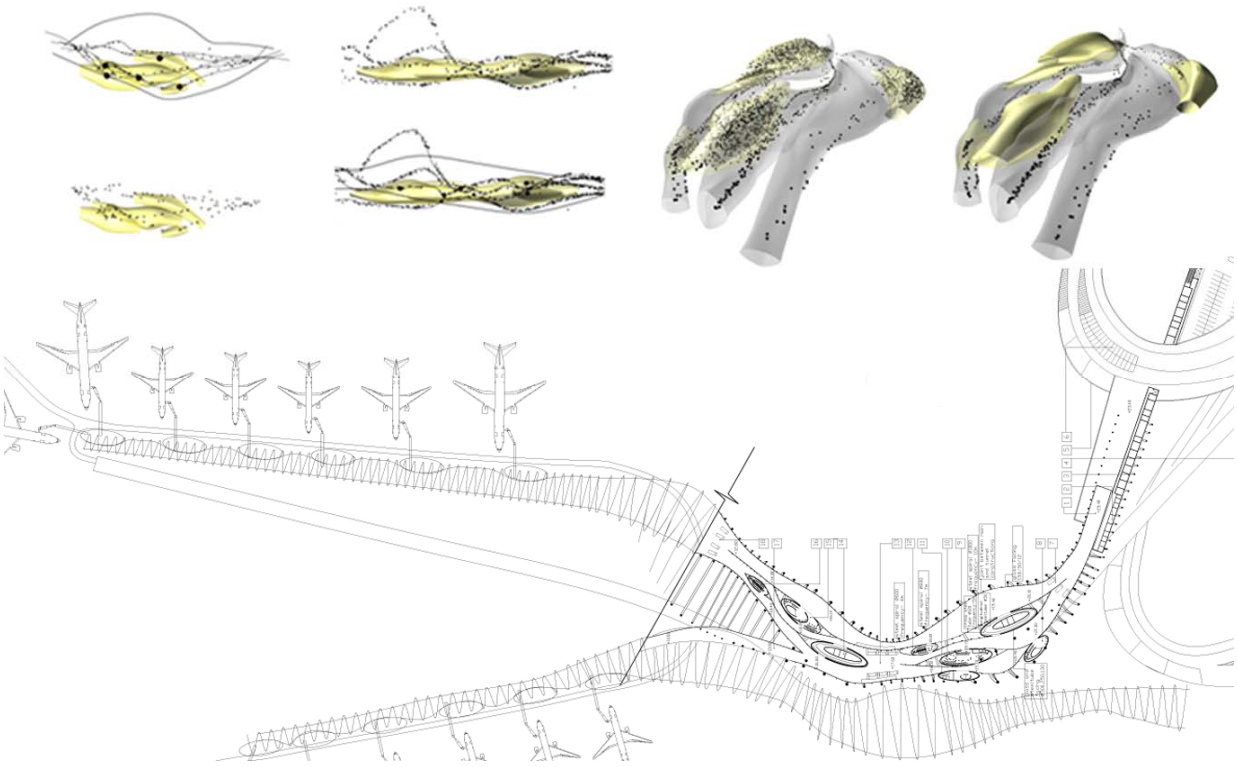

Figure 3 – examination of various spatial configuration influence on passenger flows using particle flow simulations (up), final main floor plan (down) (image Diana Voyshvillo and Zhenya Khilkevich).

Through the examination of movement and flow into and through the building, the digital code enables the creation of differences within the similarities of the type constraints. The code sets the parameters for the project and allowes flexibility for various articulations

[8] Airport project was design by the students Diana Voyshvillo and Zhenya Khilkevich during the academic year 2005-2006.

according to the specific context. Figure 3 shows a sample of the numerous particle flow simulations that were performed in order to reach the expected passengers flow in the airport.

This year (2006-2007), the studio is examining the question of performance in architecture. Expanding and criticizing architectural functionalism, the studio examines the ways in which the architectural function goes beyond prescribed usage and allows a flexible adaptation of the architectural entity. As a result, in the studio, we propose performance, rather than function, as a key concept for the evolution of the architectural project. Every project in the studio this year attempts to examine architectural functionality through its performance. This may be achieved by several means, including by evaluating performance through the integration of simulation and optimization tools into the architectural design process, and by developing generative design methods based on coding the relationship between performance and space in different scales. Working on a city scale, Evo_city project[9], for example, uses Cellular Automata[10] (CA) to develop a bottom-up method for the generation of different urban schemes according to various performance scenarios. The application of CA allows for the rapid development and comparison of complex land use and social scenarios based on complex 3-D (as opposed to 2.5-D, i.e. extruded plan) formal strategies. Figure 4 shows the interface of the developed software, which allows parametric control over the relationship between the key parameters of the generated city.

---

[9] Evo_city project was designed by the students Moshit Fidelman and Guy Austern during the academic year 2006-2007.

[10] Cellular Automata is one of the earliest types of computer form generation tools. Based on Von Neuman's ideas on computed based self-replicating forms from the 1940's cellular automata began mainly as a 2-D growth simulating algorithms/software. A well known implication of this idea is Conway's Game of life that was developed in the 1970's, which allows growth of 2-D patterns with three basic rules.

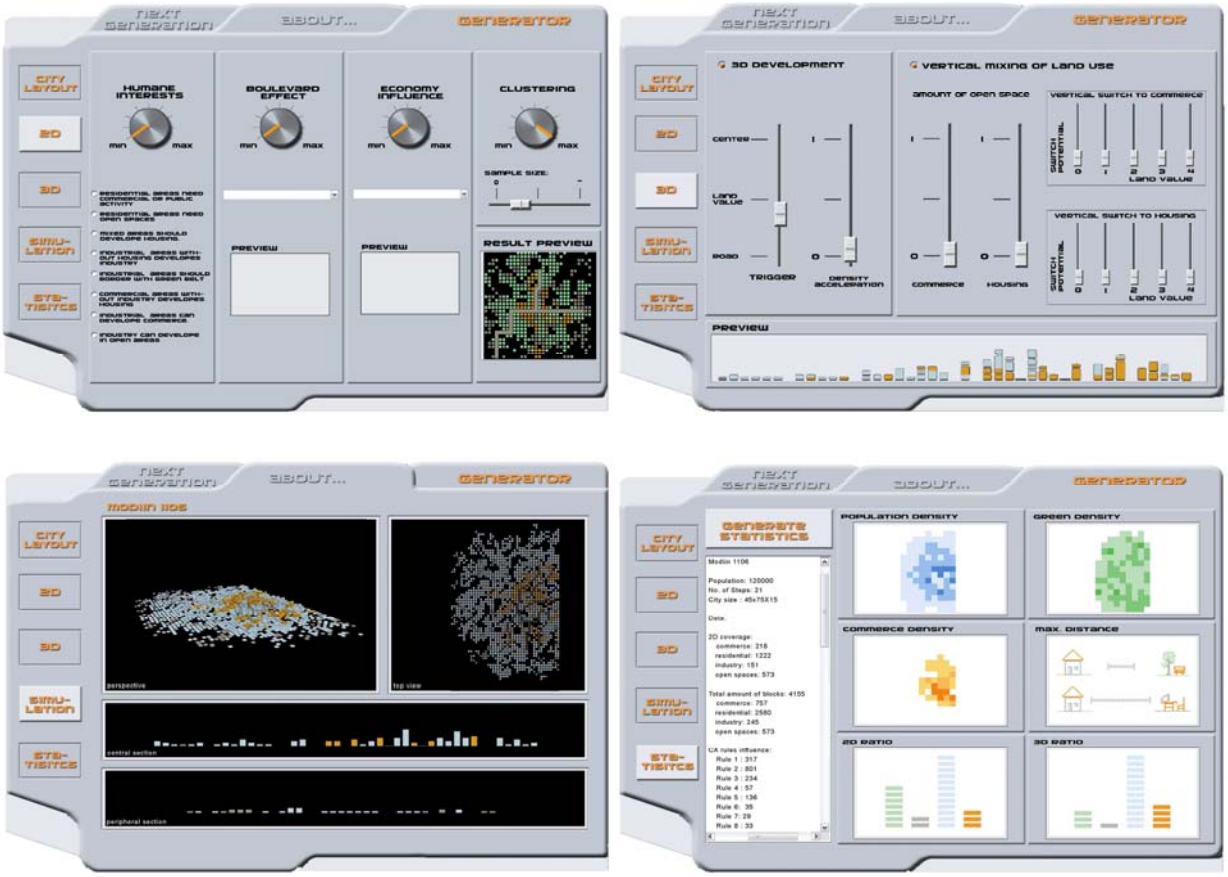

Figure 4 – Evo_city1 – new interface for parametric control over the generation process (image: Moshit Fidelman and Guy Austern).

While the initial generation result concentrated on 2.5-D spatial distribution of land use (see

Figure 5), the second stage (evo_city2) generation shifted to 3-D distribution of land use and a new suggestion for complex circulation in a future dense f high rise environment (see Figure 6).

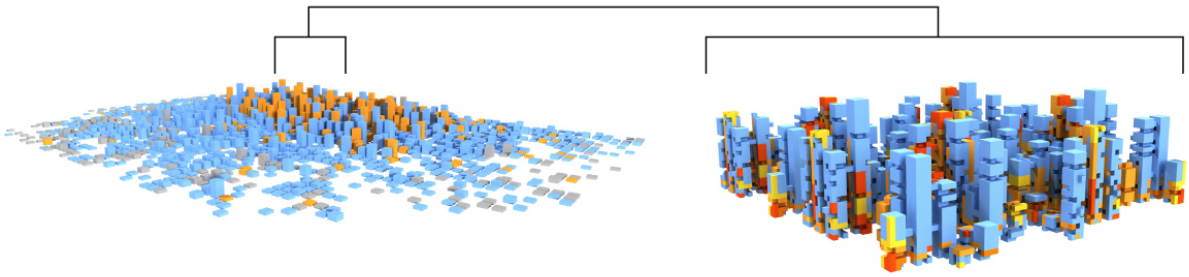

Figure 5 – Evo_city 1 - initial generation result – parametrically controlled 2.5-D spatial distribution of land use (left), Evo_city 2 secondary generation - 3-D distribution of land use based environmental and circulation parameters(right) (image: Moshit Fidelman and Guy Austern).

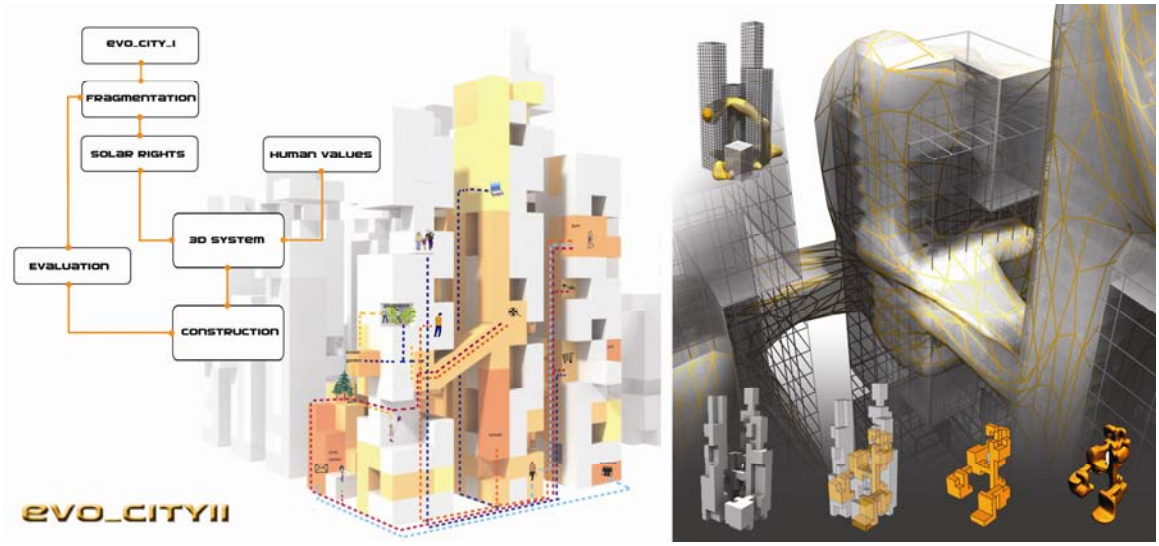

Figure 6 – Evo_city 2 suggestion for a complex circulation in a new type of high rise environment (image: Moshit Fidelman and Guy Austern)

Conclusions

The selection of the annual themes for the laboratory becomes a methodological question as it does not perceive the digital tool merely as an extension of our bodies and minds, or a means of better expressing our ideas. Instead, the themes are drawn from architectural history and theory, and the digital tools are developed to address them in a specified way. In this fashion, the question and the tools are interconnected inseparably, and the end result is an outcome of an intellectual pursuit and a corresponding tool. The advantage of using the digital tool lies not only in the ability to specify a tool for each and every

project, but also in the new dimensions that the tool generates. These new dimensions are, in fact, new realities.

Thus, the methods examined in the laboratory correspond to these realities and at the same time create them. Allowing to integrate a complex and large database, the digital tool does not reduce the architectural project into a question of typology, but rather allows specificities that, on the one hand, question the typology, asking to articulate the complexity of a project as a well-defined question with a corresponding method. And, on the other hand, the digital tool does not refute the old architectural notions, marking them as irrelevant for contemporary realities. The ability to articulate and execute the architectural program in a more complex way, through the examination of new formal expressions and architectural performance, enables rethinking architecture as an intricate question.

References:

Archer, B. 1984. Whatever Became of Design Methodology. In Development in Design Methodology ed. Cross, N. Willey & Sons. Chichester. UK. Pp 347-349. Originally published in Design Studies, 1. 1979. Pp. 17-19.

Chu, K. 2005. Metaphysics of genetic architecture and computation. In genetic architecture II: digital tools & organic forms. Site books. Pp 158-180

Cross, N. 1989. Engineering Design Methods, John Willey & Sons. Chichester. UK. p33.

Deleuze, G. and Guattari, F. 1987. A Thousand Plateaus: Capitalism and Schizophrenia, translation: Brian Massumi, University of Minnesota Press.

Eisenman, P. 1992. Visions Unfolding: Architecture in the age of electronic media. Domus, no. 734. Pp. 17-21.

Goldschmidt, G. 1997. Capturing Indeterminism: Representation in the Design Problem Space. Design Studies 18. pp 441-445.

Goldschmidt, G. 2001 . Visual Analogy - A Strategy for Design Reasoning and Learning. In Design Knowing and Learning: Cognition in Design Education. Ed, C. Eastman, W. Newstetter, M. McCracken. ELSEVIER. Pp.199-215

Jones, C. J. 1984. A Method of Systematic Design. In Development in Design Methodology ed. Cross, N. Willey & Sons. Chichester. UK. p9.

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

Kroes, p. 2002. Design Methodology and the Nature of Technical Artifacts. Design Studies 23. Pp. 87-302.

McLuhan, M. 1994. Understanding Media: the Extension of Man, Cambridge, Mass. MIT Press

Negroponte, N. 1970. The architecture machine. MIT Press. P-7,119, Preface.

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

Steele, B. 1998. Data (E)scape. - Design as Research. Daidalos 69/70. 1998/99. Pp. 54-59.

Terzidis, K. 2006. Algorithmic Architecture. Architectural Press. Pp. 14-35.