Dichotomy between Digital Project and Real Project Execution

Diego Pinochet 1, Claudio Labarca 2, Macarena Burdiles 3
1 Magister en arquitectura UC. Facultad de Arquitectura Pontificia Universidad Católica de Chile, 2 Magister en arquitectura UC. Facultad de Arquitectura Pontificia Universidad Católica de Chile, 3 Magister en arquitectura UC. Facultad de Arquitectura Pontificia Universidad Católica de Chile.
1 dipinoch@uc.cl, 2 clabarca@uc.cl, 3 imburdil@uc.cl

Abstract. This paper relates the experience of the author in finding digital tools to generate diversity of results in the design process of one project, aiming to generate base tools developed in the form of script codes, that may operate in diverse fields of action reacting to the diverse variables characteristics of each place. The development of the script is what allows this to happen.

Keywords: Digital Design: Manufacturing; Scripting; Programming; CAD + CAM.

Introduction

The introduction of new production techniques in architecture has been incorporated from other disciplines and industries. This new reality has opened new questions about the practice of professional architects, which has become increasingly recurrent on the incorporation of digital design and production techniques in the search for new design paradigms with originally resolved production processes. Computers are no longer considered as simple means of visual representation, but have become an essential tool to solve problems inherent in the discipline of architecture. The problem arises from the apparent dichotomy between the project and the digital delivery of the actual project into reality.

It is the materialization aspect which is defining nowadays what is known as digital architecture, what in the last decade was built digitally without any restrictions (Lynn’s blob architecture or Novak’s liquid architecture) today takes completely new dimensions, opening the action field of the discipline into the problems that give to the designing process the consideration of the production methods, this has an strict relation with the processes oriented to manufacturing (CAAD + CAM.)

On Designing complex forms

The use of digital tools applied into design has given architects and designers the complete freedom for visualizing their constructive ideas in a relatively short period of time. Simulation gave to the creative mind of the architects a new boundary of freedom. However, this liberty was kept inside the computer boundaries, because this immediate design and visualization tool didn’t necessarily deal with construction problematic. It is this construction problematic that changes the boundaries of this complete freedom given by the computer, the design process is intervened by the possibilities of realization that
digital manufacturing gives to digital modeling.

Considering now the design process that is aided by the computer, we can say that, in the recent past, the adoption of digital techniques, compiled in software like ALIAS WAVEFRONT, MAYA o 3DSMAX, have allowed to generate interesting investigations. These pursued the development of a complex form. This was made through multiple and infinite polymorphic configurations, through morphing processes, such as, inverse kinematics (IK), and fluid or particle flow simulations, for example. Though these exercises did not consider the internalization of the variables and compiled codes inside the nucleus (kernel) of the software. This leads into a situation in which the architect or the designer generates what the software allows to generate, and becomes a simple executor of routine instructions, which doesn’t necessarily imply that what is simulated on digital environment is susceptible of being materialized on the physic world.

The use of morphing techniques as an investigation and analytic instance of shape, gave origin to some questioning related to the characteristics expected for the final shape. The diverse formal iterations that arose from the morphing techniques consist on the generation of ‘infinite results derived from a linear relation between cause and effect’. Referring to the initial and the final state of the animation,
somehow, this creates doubt when considering the following questions. What is the optimum shape derived from these processes? What determines an optimum state of shape in these cases? What level of control does the designer have over the deformations that the design product suffers on the iterative process? And above all, what guaranties does the designer have of being able to bring these designs to reality?

Even if the use of animation techniques has been useful to analyze and investigate topics related to object morphing, and polymorphism, these techniques are not more than a routine execution of previously organized tools in a software, which can rarely be useful for general cases, meaning that these answers usually respond to an specific project situation, more than delivering a method. The incorporation of programming in digital environments, supposes a change into a fructuous field of action for project design in architecture. This allows redefining the labor of design as a real investigative instance.

**Customizing of digital tools and the precise indetermination of form**

The incorporation of programming techniques on architecture, is allowing the designer to redefine the form forming processes through an instance in which the designer sees him/herself forced to internalize the software’s own syntax, codes, and traducing the design process into a series of ordered steps that the computer follows to generate forms. This has brought to architects the possibility of complementing the traditional creative labour with an investigative instance, putting emphasis on the formal aspects and the processes that give shape to these forms.
It is in this field of design, where the polymorphism concept of a project is relative to the use of a customized tool, which was formerly done through the use of written tools compiled by someone else. The mentioned customized tool is derived from an algorithm of generative nature, which allows the iteration of different formal states. Each iteration or formal state of a project is controlled by the designer, being him/her who controls the variables and relations that establish the formal definition of the project.

It is at this point that the designer may reconsider the process that gives origin to a specific project, by establishing which are the main concepts that give origin to a determined design idea and being able to abstract and organize them into the own language of the software. It is in this diagrammatic idealization of the genesis of a project where we can notice that the factors that the designer handles, considering polymorphic iterations of a singular design, are no longer related to a cause and effect linear relation; instead they respond to the adaptable capacity of shape, depending on the programmed code. The code establishes behaviour patterns that consider this non linearity of the generative processes, but mostly, redefine constantly the project in relation to its own condition of dynamic system. Through this, the use of algorithms for generating a code is an alternative method for generating a project, a means through which the architect is able to achieve his/her goal of building an idea. This method allows incorporating immediately in the design matters the manufacturing issue.

This is due mainly to the fact that adoption of programming techniques inside the design field requires the incorporation of algorithmic thought. In
algorithmic thought the structure and abstraction of rules constitute the base for the generation of a determined design. The structured thought, which is previous to form finding, allows the systematic extraction of logical principles, relations and restrictions inside a system, which is finally traduced into a ‘generic solution’ plan.

The introduction of algorithmic thought to the design process (essential to the programming of customized tools into the digital environment) allows introducing an interesting concept that can be catalogued as ‘precise in determination of shape’. Precise in determination of shape consists on building design as a field of possibilities, based on the non linear relation between physical components of a project and the algorithmic pattern that allows visualizing different possibilities of results, always responding to the same language and initial conceptual matters of the project. This generic solution plan is directly related to the structured construction of a script which acts as a fertile interface for form finding, with the difference that these parameters and variables are now controlled by the designer, and don’t come predetermined by the software. The design process is reoriented to the generation of an ‘information model’ that works in the measure that the architect introduces in the code the necessary values to generate the diverse states of one object.

Considering this, design is constituted as a much more efficient investigation instance, due to the incorporation of parameters to the generative process of a form. This allows breaking the linearity of the traditional design process, and allows establishing ranks of behaviour ruled by the value of variables. To this we call handling information inside the code.

Such capacity of handling information inside a structured process of design, allows the architect to incorporate the formal genesis of the project with its construction possibility, in a parallel and immediate way. This breaks the traditional timing of an architectural design process where generally the first big step is to design, and the second huge step is to build. By incorporating some real world variables into the simulation and programming process the architect using script should be able to face the real world problematic before the project is proposed to be built. The design process is oriented to build an information model, which will allow iterating different shapes
of a same project and by this being able to locate in different action fields, giving a series of different answers to different problems however always using the same ‘generic solution plan’.

Customizing of digital design tools through scripting is focused on the creation of these information models, mentioned previously, that allow to be materialized through CAM technology. They belong to what is known as ‘file to factory’ type of processes. This logic tends to the composition of a design model oriented to the fabrication of components and its later assembly. On this assembly process the designer incorporates inside the script code the constructive logic that the CAM platform offers; using for example cutter through laser or carving through CNC machinery, allowing generating the necessary information for the construction of each project component.

This can determine that the designer, through the application of available software technology, finds his/her design labour changing from creator to investigator, as he/her turns to the application of alternative and personalized models to understand and systemize the design process. This settles us in a context in which the questioning of digital techniques in architecture has gone beyond representational matters, leading the discussion to the problematic of ‘becoming real’, and how for years this was something that did not interest digital architecture.

Design is established as a constant search that allows facing and giving solution to different problems, which locate it in a favourable position for non standardized production. However there is still the doubt on the real pertinence of the application of these design processes through programming, above all because the generation of an information model, based on variability handling, for its subsequent prototyping is conditioned (each day less) to aspects as the availability of a technological support that allows the fabrication of parts.

For the methodological matters, the applied scripting techniques applied in architecture, allow the generation of a fertile interface for the architect, through which architects can reach great sophistication levels for projects, beyond the formal aspects of design, relating to the considerations related to variable handling and the multiple data that inform the project.

References