Complexity as a result of improved design capabilities through the use of computer tools was introduced in the architectural debate since these became irreplaceable. On the other hand, not every designer is genuinely aware of the logical implications that the use of these tools may entail. Used as a simple emulation of enhanced traditional design tools—drawings and models, they do not alter the process of design significantly. However, the potential of such tools beyond their instrumentality introduces designers into the realm of digital consciousness.

This paper analyzes complexity as an inherent quality of computer aided architectural design in relation to four different digitally conscious design strategies. First, the increase of complexity involved in digital architectural designs because of their potentiality to manage enormous amounts of differentiated information. Second, the complexity inherent to an open form such as parametric or generative designs may be defined. Third, the use of the computer as a smart partner involved in the design process—i.e., form finding strategies—rather than as a simple efficient machine able to repeat our abilities faster and more effectively in certain roles of the design process. Finally, it analyzes the possibility of generating parameterized typologies as a result of the openness of form, as well as the increased complexity that randomness may introduce in algorithmic design.

The paper concludes with reflections on complexity vs. simplicity considering the fact that the simplicity characteristic of Modernism aesthetics and constructive values collide with the baroque formal complexity achieved in generative design.
1 Introduction

The advent of the digital era in architectural design has influenced architecture's own language in the past years. However, the use of computer has genuinely influenced a relatively small amount of architects, basically those that have become aware of the real repercussions in the design process that the use of C.A.A.D. may entail. Thus, Mitchell has divided architects into those who use it as a way of improving their office's efficiency and those who have understood the intrinsic machinic qualities of these tools and who have, consequently, altered their architectural language (Mitchell 2005).

Digital consciousness is a design strategy to be found in different degrees among those architects or designers that rely on the computer not only as a tool but as part of the team.

Complexity has been placed as a central issue in the debate regarding C.A.A.D. Much has been said about it, and what is more indicative—it has occupied such a dominant position right from the beginning. Mitchell has wisely addressed it and has written: "[complexity] ... is the number of design decisions relative to the scale of the project. We can measure it as the ratio of added design content to added construction content" (Mitchell 2005). The concepts of design content and construction content relate to the formal complexity of the design and to the complexity involved in the manufacturing process of a particular element, respectively. Thus, the greater number of inputs or parameters necessary to define a given geometry will increase its level of complexity. Likewise, the more operations needed to produce a constructive element, the greater the level of complexity it will have. C_Wall by MATSYS is a good example of complexity with a high ratio of added design content to added construction content despite the small scale of the construction element (Figure 1).

Complexity theory studies the dynamical behaviour of systems with hundreds or thousands of elements (Kauffman 2001). Complex systems, despite being a compound of countless parts, do behave as a whole or at least can be reduced to mainstream dynamics. Beyond a critical level of complexity the use of computers is more a real need than simply an option.

2 Complexity and Added Information

Computers may help us to improve our analytical abilities and our capacity to manage great data inputs. Any architectural design has to face at least two given starting points: the program and the site. The first refers to the architectural object itself and is related to necessity and architectural sense, the latter is entirely contextual. Thanks to computers the analysis of both sets of data can be very exhaustive. Data can be selectively itemized and exploited in georeferenced databases, as it is characteristic of G.I.S. tools, adding an increasing level of complexity in the management of site inputs. Moreover, the drawing of varied data selectively arranged by layers that may appear or disappear at a mouse click provide an enormous potential in terms of added information to the architectural design. Our understanding of the complexity characteristic of a given site may be greatly improved by the visualization of the multiple relations that coexist.

The traditional concept of architectural space conceived as a neutral static container may be substituted by that of space as a field: "This new definition of space is naturally tied to the development of 20th-century physics and its notion of 'field': space becomes a field of forces and counter forces, a field that emerges from the actions taking place within it" (Kerchove, Tursi 2009). The influence of computers tools in this new conception and formalization of architectural space is evident.
In addition, the possibility of working on editable 3D models has given architects the chance of designing spatially instead of having to cope with the limitations of projective two-dimensional drawings. This has lead to the colonization of space in unprecedented ways because of the complexity of the geometry and the impossibility to reduce it to two-dimensional views. Virtual space has lead to an increased level of geometrical description; it has merged the accuracy and sharpness of mechanical drawing with the three dimensional control over geometry characteristic of physical models. To a certain extent, Monge’s classification of geometries into those that can be rigorously described and those that cannot has been questioned with the representation of digital models; thanks to digital technologies the distance between representation and reality has been altered: at each stage of the design process, the geometry of every single part may be exactly defined (Picon 2004). Complexity is also a result of enhanced representation of geometry introduced by the accuracy of virtual models.

Eisenman has written on hypercomplexity in the context of spacing strategies: “Hypercomplexity is something that is not explainable through the normal complex of logical mathematical equations. Spacing can be explained, but it requires a level of complexity not in conventional geometries. It is already another realm of description” (Eisenman 1997). It is relevant to point out that added information regarding the contextual has been thoroughly exploited among those architects eager on diagrams. Brett Steele has classified two types of digitally conscious architects: those who have centred their discourse on context and those who have condensed their efforts towards the object of the design, thus confronting two design strategies based on diagrams and process, respectively (Steele 2005). The first group (i.e., Peter Eisenman—diagrams—, UN Studio—van Berkel and Bos— or MVRDV—datascapes—) relies on the analytical capacity intrinsic to diagrams to explore and justify their architecture as a result of contemporary urban topographies. The second group (i.e., Greg Lynn, Marcos Novak, MATSYS), focuses on the autonomy of the architectural design itself, giving importance to the processual nature of design and investigating the topological relations by using a wide range of C.A.D./C.A.M. systems.
3 Complexity and Open Form

The complexity intrinsic to C.A.A.D. has other aspects; probably the most relevant of them all, from an architectural design point of view, lies on what we have termed as open form. An open form is a conceptualized formal structure rather than a closed form imposed by the architect. C_Wall by MATSYS can easily illustrate this concept; the design process can be followed in Figure 2. The original parametric massing model could vary its shape if the parameters that define such geometry were modified (i.e., number of folds). Likewise, the number of irregular honeycomb cells that articulate the wall could change—and, consequently, the scale of the openings in the wall—by varying the density of the point cloud bitmap used to produce them. A voronoi algorithm was implemented to transform the point cloud into a collection of cells. The cellular solids were then voided and unfolded into panels to be finally CNC cut and reassembled at a larger scale. This is an example of parametric and algorithmic design combining C.A.D./C.A.M. techniques only possible with digital design. All the potential geometries defined in such an open form would constitute a parameterized typology.

Very differently, a traditional designer would draw sketches or make models in order to produce a design. His action was physical and therefore the control over the geometry had to be formative, constructivist we could say, in the sense of Pareyson (Pareyson 1987). The geometry of the design would evolve from the inception phases to the finished project. Nevertheless, at any stage, the form would be frozen because it would be necessarily fossilized into matter. Independently of how suggestive the first sketches might be, their openness could only be implicit. Working with computers following traditional procedures—drawing or modelling in the virtual space—is likewise formative despite the unlimited editability of any digital model.

On the other hand, parametric and generative designs imply an explicit degree of openness: the geometry of the design is encoded and conceptualized in a non material language; in fact, the architect produces formal structures, not material shapes. Open form refers to this essential difference which is founded on the virtuality of the code. Hence, the open form generated from the script defines a parameterized typology of possible forms enclosed within the range of parameters introduced. Thus, the description of a “genetic” formal code relies on the generic definition of a geometry with the potential to generate a whole set of varied configurations (Frazer 1995).

C.A.A.D. programs running scripts that are graphically transcribed into 3D models (Rhino/Rhinoscript, Autocad/Autolisp, 3DMax/3DMaxScript, etc.) introduce a new opportunity for the development of architectural design that can be then referred to as algorithmic or parametric; such is the case of the Loophole pedestrian bridge proposal by R&Sie in collaboration with THEVERYMANY between Cieszyn in Poland and Cesky Tesin in Czech (Figure 3). That is to say, the drawing is not any longer a graphical act in the sense of the correspondence between the order given to the computer (marking of traces) and what is displayed on the screen (a support). Instead, it is the formalization of a script written in a computer language capable of generating codified designs.

This is a genuine revolution in architectural design because the scripts are as abstract as the language we use to communicate may be; at least in this sense, we may speak of parametric abstraction. Marcos Novak coined the term transarchitectures referring to architectural imagery generated in the cyberspace, a new frontier to explore within architecture: perhaps a new formal abstraction architecture-borne. Novak has written regarding the parameterization of architectural geometry: “The algorithm that produces these forms works as follows: data is interpreted as two sets of points in 3D space in bodyspace, an instance of output of the algorithm becomes a form of a material architecture [...]” (Novak 1998). This represents a further and radical step in the evolution of architectural language that is genuinely digital. In his brief but consistent text, Algorithmic Architecture, Terzidis has referred to this qualitative change: “For the first time perhaps, architectural design might be aligned with neither formalism nor rationalism but with intelligent form and traceable creativity” (Terzidis 2006).
Algorithmic architecture can hence be seen as the most advanced stage in the field of computer aided architectural inventiveness. Nevertheless, this conceptualization of form attempts to break the intuitive control of the architect over his own design. He is now getting used to scripting in order to design instead of sketching or modelling as he used to. Consequently, a defamiliarization occurs when his activity is directed to a numerically and coded design which is necessarily non material. The written numerical code produces in the designer a certain feeling of banishment in comparison to the familiarity a drawing or a model may produce.

Just as language is structured on universal concepts, scripts in architectural design conceptualize geometries. The same lack of sharpness in the description of the physical that is characteristic of textual language may be found in scripts. Drawings or physical models, on the contrary, have an enormous descriptive competence regarding the realm of materiality. That is the reason why any designer/programmer will regularly run the script to visualize the geometry on the display. However, as Picon has pointed out, the dichotomy between virtuality and reality is not characteristic of architectural design, any design is always a virtual object: it anticipates a whole range of possible materializations.

Algorithmic architecture takes advantage of the computers’ analytical potential to manage casuistries impossible to deal with the inherent limitations of human beings. Obviously, algorithms are not able to produce finished designs; however, they can be used to develop and explore an extraordinary wide range of formal possibilities. Moreover, whereas mouse-based operations to manipulate 3D models cannot transcend computerization—a basic level of C.A.D.—algorithmic architectural design is truly computational.

Algorithms like parametric design imply open form, however the iteration we may find in algorithms produce generative designs. Both breed formal typologies that depend on the parameterization of the form they configure. The use of algorithms enriches the complexity even further as the structure of the algorithm itself is generative. Provided algorithms define formal structures generated from a script which are then graphed and modelled virtually; the idea of order required in all architecture is implicit in the structural design of the algorithm itself. Indeed, this is a property of algorithms that opens new perspectives for architectural research.

Algorithms may be formally driven, something which justifies how algorithmic formal investigations can be lead in certain directions. That is to say, anyone using algorithmic procedures does not lack of formal criterion; instead, the designer will drive the formal search implicit in the script trying to explore new possible geometries. Consequently, we can explore many possibilities simply modifying the starting parameters. Since the computer is capable of producing a countless number of results, the work of the architect will then shift to choosing and discarding the computer output. Accordingly, the architect’s role will involve analyzing and evaluating the obtained geometry rather than directly generating the architectural form. It is in this sense that the control over form is no longer constructivist.

Neil Leach has addressed this issue in the field of Morphogenesis: “More recently it has been appropriated within architectural circles to designate an approach to design that seeks to challenge the hegemony of top-down processes of form-making, and replace it with a bottom-up logic of form finding. The emphasis is therefore on material performance over appearance and on processes over representation.” (Leach 2009). From a literal standpoint, the emulation of certain forms in nature in the field of architecture as shape optimizations would imply a certain figuration understood as a simple mimesis. This attitude falls squarely within what has been termed Biomimetics. However, if the order of nature is what is sought to imitate, the strategy is closer to the abstraction from the pre-existent.

4 Complexity, Computerization vs. Computing and Form Finding

Algorithmic architecture takes advantage of the computers’ analytical potential to manage casuistries impossible to deal with the inherent limitations of human beings. Obviously, algorithms are not able to produce finished designs; however, they can be used to develop and explore an extraordinary wide range of formal possibilities. Moreover, whereas mouse-based operations to manipulate 3D models cannot transcend computerization—a basic level of C.A.D.—algorithmic architectural design is truly computational (Terzidis 2006).

5 Complexity, Randomness, and Parameterized Typology

Generative design in architecture introduces a degree of variability unimaginable to be achieved through conventional means. And what is even more important, it introduces randomness as a new factor into the design, something which
can only be articulated through the consideration of open form. Kostas Terzidis has associated complexity and randomness on the following terms: “[...] randomness is characterized as the maximum of complexity, and the opposite of regularity and simplicity” (Terzidis 2008). A slight variation in the code can entail a huge change in the final form. Thus, algorithms have an enormous formal potential; however all must be governed by a certain order implicit within the algorithm itself, its inner structure. All the forms generated by the algorithm will have some sense, will be formally oriented and, therefore, will have a common order. The change of parameters contained in the algorithm will result in a collection of geometries that define a parameterized typology. Again, the idea of open form appears to be necessary to produce such typology. And what is more substantial, such openness can only be possible through the generation of scripts that define generic forms. These geometries have to be defined topologically instead of geometrically (DeLanda 2005); it is the relation between the parts and the whole, and most significantly their connectivity, what makes possible the genesis of the digital or parameterized typology.

A closed form (as found in a conventional design) is defined geometrically and necessarily belongs to a metric space because it is material. On the contrary, an open form (as a parametric or algorithmic design) is defined topologically, it is not contained in a metric space—instead, and to a certain extent, it inhabits a topological space. Because it is a conceptual design it is not formalized into matter, rather it is a logical construct defined by a code, a non material language.

6 Conclusions

The enormous enhancement that C.A.A.D. tools have introduced in our field of knowledge and professional practice is extraordinary, their potential is yet to be explored. The control of complex surfaces with double curvature allows architectural geometries that were impossible to manage before the advent of computers. In parallel with these advances, construction technologies—most significantly C.A.M.—may virtually build any of our conceptions. Never in history had the architect such resourcefulness in both conception and construction.

However, the banalization of architectural form is ready to flourish. All these capacities, if uncontrolled, may lead us to a rampant formalism with little architectural sense. Bizarre originality is sought in a globalised world just for the sake of building landmarks to operate on a global scale: cities that compete with each other to gain visibility and neon light efflorescence. Hyper-celebrated digital designs as Gehry's Guggenheim Museum in Bilbao—only half digital, to be precise—have uncritically catapulted the author's fame ending up in a proliferation of cloned architectures of the original.

Biomimetics, conceived as a figuration of nature's organisms is on the opposite pole of modernist abstractive heritage. Morphogenesis, properly applied to architecture, would have to enter into the realm of algorithmic architecture as an abstraction of nature's generative processes and strategies—actions—instead of plain verbatim copies of natural form—objects—for the sake of it or simply because of nature's consideration as a source of beauty.

The whimsical complexity we find in many recent digital designs seems to defy the miesian simplicity characteristic of Modernism. Such clarity and elegance transcended style; it was based upon principles of necessity, efficiency, simplicity, economy, and the abstraction of geometries that were then ravishingly new, just as new as algorithmic designs may be now. It is obvious that complexity is not only the result of technology, it is also part of a cultural discourse. Complex forms were also found in Gothic or Baroque architecture, but today's digital formalism has confronted Modernist simplicity. Beyond the Modernist aesthetics, a constructive logic could be found: the use of new technologies should not be an alibi to question such consistency unless they can generate improved architectural design principles.

Open form is possible thanks to abstract constructs such as parametric or algorithmic designs. Its definition is topological instead of geometrical; unlike conventional

Figures 5 and 6. MATSYS. Zero/Fold Screen (2010)
designs, it is non material. Because it is the opposite of a closed form, it may generate parameterized typologies which are the basis of form finding strategies. Open form introduces a new design paradigm in the context of architectural design.

*Complexity* as a complementary concept to complexity may well lead us in the right direction; it is obvious that it implies a better control over the form based on procedures such as filtering, reduction, selection, and abstraction ([Kajjima 2008](#)) which require a more sensible attitude and the optimization of shape in relation to parameters and given conditions. The Zero/Fold Screen designed by MATSYS (see Figures 4, 5, 6) could be an example of complexity as understood by Mitchell and simplexity referred to minimization of the waste in the CNC cut of the original wood panelling. It is a wise use of typically complex digital geometries combined with self imposed limitations in the context of manufacturing processes and economy of materials.

Programmers/designers must be critical with their models and must permanently bear in mind that form and sense in architecture are founded on necessity. Needless is to say that the effort to program simplification instead of proliferation is larger. In spite of the infatuation digital tools may generate, Milizia’s identification between beauty and necessity in architecture is as relevant as ever ([Milizia 1781](#)):

“If architecture be daughter of necessity, even its beauties should appear to result from such. […] Nothing must be introduced which has not its proper office, and is not an integral part of the fabric itself; so whatever is represented must appear of service. No arrangement must be made for which a good reason cannot be assigned.”

## References


* These images have been kindly sent by Andrew Kudless (MATSYS) for the illustration of this paper.

** This image has been kindly sent by Marc Fornes (THEVERYMANY) for the illustration of this paper.

We want to publicly acknowledge Andrew Kudless and Marc Fornes for their cooperation. Obviously, the opinions in the text are solely responsibility of the text’s author and might not be shared by the authors of the proposals or built projects illustrated here.