TOPOLOGICAL FUTURE: GENERATIVE BIM THINKING

ALFREDO ANDIA
Florida International University

COMPUTERS IN PRACTICE

Architectural practice has been integrating digital technology pursuing two different types of productivity gains in the past 30 years: (1) Skills, (2) Business Processes. In the first period of "skills changes" the use of 2D and 3D CAD software only replaced drafting desks and just improved drafting productivity inside the architectural office – with practically very limited gains in the whole design-build process. In a second period, "business processes changes," Architectural, Engineering, and Construction (AEC) firms are using computers to improve the productivity in the coordination of drawings and materials - primarily with the implementation of Building Information Model (BIM) software and processes (Andia 2002).

BIM in History

BIM is not a new idea. It has been around since the inception of the first commercially available CAD systems in the early 1970s. A significant number of pioneering 3D parametric software such as SSHA (developed at Edinburgh for the Scottish Special Housing Association), CEDAR (Property Services Agency), HARNESS (Department of Heath and Social Security) and OXSYS (Oxford Area Health Board) were initially designed as specialized systems to serve particular organizations and building types (McCullough & Mitchell, 1990). Parallel efforts such as CAEADS, GLIDE, GLIDE-II, ARCH-model emerged in academia in the US during that period (Eastman 1999).

OXSYS was the precursor of BDS (Building Design System) and RUCAPS (Really Usable Computer-Aided Production System) which became available commercially in the UK in the 1970s and surfaced with concepts very similar to today's BIM systems. All these systems had a common vision: to construct virtually a 3D building by modeling all their building elements and assemblies. They allowed multi-users to manipulate a single parametric 3D model. Graphic reports and 2D drawings were mere derivatives created automatically from the main 3D model. By the mid 1980s a second wave of 3D parametrically based software such as SONATA (which replaced RUCAPS and is considered the precursor of Revit), Reflex, CHEOPS, GDS, CATIA, GE/CALMA, and Pro/Engineer achieved commercial presence.

Parametric vs. CAD

Most of these pioneering parametric programs in the 1980s, became standard in industries such as electronics, aerospace, and car manufacturing. In these industries there were real rewards in accurately defined models whose performance could be analyzed, simulated, and fabricated. The elevated cost of hardware & software and the inherently segmented workflow in the Architecture, Engineering and Construction (AEC) industry proved to be significant barriers for the parametric metaphor and most practices choose to automate only their 2D drafting capabilities by massively buying CAD drafting software in Personal Computers (PC) introduced in the early 1980s.

BIM Today

It took close to two decades for the 3D parametric model to make a significant comeback in the AEC industry. BIM has become one of the central themes in the computerization of Architectural practice today. BIM is typically understood as an act of purchasing particular software and training staff, but is much more complex than that. BIM is ultimately about business processes and information management in one of the most fragmented and complex industries in the world. There is no right BIM solution but only BIM narratives that specifically respond to the particular work culture in which the design and construction teams are embedded. There are clearly 6 major discourses of BIM implementation today: (1) Collision detection BIM models, (2) Cost estimation BIM model, (3) Integrated Design and Construction BIM models, (4) The coordination of Construction Sequencing (5) Facilities Management BIM with embedded sensors in construction that aid in the management of the lifecycle of the building, (6) Procurement, price engines, and bidding systems integrated to parametric BIM modelers.

Product Information Model (PIM)

The 1970s and 1980s vision of BIM considered the 3D model as a database that could include information such as parametric geometry, material selection, manufacturing production, engineering analysis, costs, assembly details, and procurement. So a common concern in these pioneering efforts was to build standards that allowed interoperability. By the Mid 1990s the aerospace and manufacturing industry was widely adopting STEP (Standard for the Exchange of Product Model Data) and ISO standard 10303 for information exchange. The standards harmonized the transaction of data. The 3D virtual model was thought as an assembly of virtual products or PIMs, each one with its own data.

Paradoxically, most BIM implementation today in the AEC industry has been centered only in the 3D geometry of the project. There has been only modest progress in the PIM vision and true interoperability of data. However, there are initial signs of a massive move to create digital catalogs populated with intelligent objects that could be embedded into BIM models. The pioneering projects of the 1970s did not have the Internet. The metadata of PIM objects could develop enormous BIM libraries which could be searched automatically. Throughout the design and construction phase manufacturers, distributors, and even contractors could provide initial bids improving significantly tasks such as cost estimation, procurement, and order fulfillment. Also a pricing engine could make the BIM model an internet portal: a nD BIM connected to a worldwide metadata engine that is fully integrated to a bussines-to-bussines bidding system: such as the one found in other industries: Sabre, Worldspan, and Pegasus in travel, Chemconnect in the Chemical trade, PartsBase in aerospace or BuildNet in the construction industry which went bankrupt in 2001 after $140 Million investment. The model could provide real-time pricing from multiple brands which can be connected automatically to all interested parties such as constructors, sub-constructors, and distributors.

Shortcomings in Current Practice

The computerization process of industries it is more a social phenomena of technology consumption rather than pure technological invention or vision. The AEC industry has been using computing technology mostly to control cost and optimize coordination but not to truly revolutionize design-build practices. The current computerization metaphor it is controlled by a few giant software companies that are not interested in revolution but to maintain the status quo via minimal productivity improvements. Radical transformation of the industry would fundamentally alter their markets and potential revenue stream.

COMPUTING IN ACADEMIA

Although, significant numbers of administrators and educators in schools of architecture today support the digital metaphor promoted by the largest software companies in the world, many academic researchers in the field of "digital architecture" have provided a broader critique to computerization of design in the past 50 years. The most ambitious scholarly endeavors have been dedicated to computerize architectural intelligence or thinking. These projects ask a fundamental question: what do architects do? Since the Renaissance the discipline of architecture has not been directly involved in the construction of buildings but more particularly in the creation of a "heuristic experiential" knowledge of everything that occurs before construction. Architecture is based on "experiences in action" (Schön 1984), is argumentatively "wicked" (Rittel and Webber 1973), and is extremely difficult to compute – as most design activities.

With different grays and shades the discourses of computing architectural thinking in academia could fit theoretically into the following three categories:

Coding Problem Solving Processes In Architectural Academia

Central to the traditional academic conceptualizations are the intellectualization of computing that occurred in the 1960s in fields such as artificial intelligence, information processing, cybernetics, and design methods. Among the most celebrated in architecture at the time were Christopher Alexander's "misfit variables," Nicolas Negroponte's "architectural machine," M. Asimow's "design elements," Christopher Jones' "factors," Bruce Archer's "subproblems," and Nigel Cross "automated architect." Specifically influential were the ideas of Nobel Prize laureate Herbert Simon that stated that human problem solving behavior could be simulated and programmed (Newell and Simon, 1972). However, narratives that suppose that the design professions navigate only in problem-solving realms are incomplete and do not truly understand the political and poetic challenges that design disciplines confront in late capitalism. If Simon ideas on rule-based problem solving are truly computable then we would tend to solve our human-space-needs based on a very restricted framework. We would only accept the factors we consider important in a particular time and would be developing solutions to a stationary framework.

Coding Formal Expression in Architectural Academia

Using CAD and scripting to automate, reproduce, or quickly test the formal appearance of architecture has been another traditional endeavor in architectural schools. The 1970s and 1980s work on "shape grammars" by George Stiny and William Mitchell have resurfaced in a more avant-garde format in schools of architecture in the past decade using popular tools such as Rhino scripting, grasshopper, MEL, Generative Components, and others.

Designers, when using programming, are forced to make explicit their design process and the parametrical conditions to which their design respond. In these new circumstances the generative software no longer mimics the traditional environment in which the architect has to model everything. Architectural models are no longer frozen. They became parametric and manageable. Parametric techniques substitute the sculptural or figurative designer and allowing much more complex spatial formation.

Most architectural academicians using these new types of software seem to get infatuated with the shape generation possibilities. After a while one can clearly observe very precise families of forms and software tricks that bounce in blogs across the oceans and between architectural schools. The apparent aesthetic exhaustion of this contemporary generation of scripting techniques is the result of the obsession architecture has had almost exclusively with complex geometry, shapes, and form. In this theoretical paradigm, most courses and research that bear names such as "topology and performance" usually are limited only to shape or geometrical aspirations. A more advanced research have emerged in developing more comprehensive strategies in which digital form-finding techniques via parametric tool can evaluate and maximize the environmentally performative aspects of projects.

Coding Self-Generative Systems in Academia

A parallel theoretical approach aims at the generation of a self-generative computerized design process. The major analogy proposed here is that the design of spaces could be generated in an evolutionary and self-organization process as founded in natural formations. One of the major theoretical references is the work of philosopher Gilles Deleuze whose reflective work on space move loosely from the formation of molecular populations to flora and fauna milieus to demonstrate that form in natural structures depends of autonomous codes. According to Deleuze forms in natural structures do not pre-exist its population, forms are more like statistical results (Delezue & Guattari 1987).

Critical to Deluze thinking about the self-generative design processes of natural structures is the topological diagram. Topological transformation of the diagram allows natural forms to adapt, progress, and respond to their environments. For Deleuze natural space is always in a process of becoming, thus, it is always emergent. But underneath this turbulent process of transformation there are constant topologies that maintain populations' identity. Deleuzian space is not about the form of the smooth, the striated, the fold, or a blendshape command in Maya software. It is not an aesthetic space at all. It is evolutionary space. The radical contribution of Deleuze is that it is the final point of departure from Cartesian space. An exit from what was considered human space until the 1960s.

The ideas of Delezue have a long tradition in computer sciences and mathematics. The early pioneering computer work on cellular automaton by Stan Ulan and John Von Neumann evolved into fields such as genetic algorithms, evolutionary algorithm, emergence computation (Mitchell 1996), and more recently evolutionary fabrication. The topological, evolutionary, and genetic character of spatial incidents has been explored in many academic settings since the 1960s to today including the theoretical work of John Frazer and Paul Coates (Frazer 1995, Coates 2010).

These approaches search for "self-organizing systems" and basic "genetic codes" in architecture. Design becomes more an unexpected, unsupervised, and experiential process. These systems suspend the expertise of the author and instead develop an interaction with the architect in manipulative processes of experiential interaction. One of the biggest aims of these systems is that computing could code data into spatial relationships, therefore, transforming the contemporary heuristic world of architecture. The idea is that "generative architecture" could provide a family or realm of possible designs. One of the biggest limits remains on the difficulties of coding the large number of data, communication, and cultural practices that are involved in the highly segmented design and construction industry.

TWO INNOVATIVE CASES

The three theoretical discourses described above are interrelated. All of them develop a more in depth conversation about the heuristic methods of the earlier stages of building design. The maturing parametric digital computing methods and emergent software in other fields such as biology has established a new age for testing and implementation of some of the theoretical ideas expressed in academia.

The two prototypes that are presented bellow are significant advancements that are significant test to old research theories in architecture. The first case is the results of a 2nd year research master studio at the Berlage Institute in 2007 and lead by Professor Peter Trummer. The second case is part of the works of Aedas R&D, a research unit inside one of the largest Architectural firm in the world whose work is somewhat related to Centre for Evolutionary Computing in Architecture, at the University of East London and the University of Central London.

Case 1: A New Parametric And Topological Model For Chinese Housing

The "associative design" studio led by Professor Trummer at the Berlage Institute investigated traditional housing typologies in China in 2007. You have probably heard in the news how large numbers of traditional neighborhoods in major cities in China are rapidly disappearing and replaced by a mixture of Russian style low-rise apartment for workers, and mid, and large residential towers for the professional class and managers. Low-rise apartment houses and residential towers have become the de facto solution for housing. It is an efficient mass housing solution because it has been very difficult to conceptualize another framework to approach another level of complexity to the problem of housing in China.

If we were to apply problem-solving thinking, in the Herbert Simon tradition, we will probably end up with

Figure 1. Videos of the results of the Associative Design Studio at the Berlage Institute led by Professor Peter Trummer. Scan QR-code with a Smart Phone or use this web link: http://x.co/bSsn

an extraordinary parametric BIM model that would quickly generate designs for those generic row houses and towers with efficient coordination, accurate cost estimating, and procurement. But that problem solving approach could not provide us of other more unexpected creative observations that have been traditionally exhibited by architects in history.

The studio at the Berlage Institute, composed by a large number of Chinese students, developed a computing strategy which was far more ground-breaking and developed a more unexpected response. The studio was divided into research teams that studied issues such FAR, internal room organization, and sun trajectories, and the traditional vernacular Chinese housing. The research concluded that historically Chinese housing had clear associative design patterns that related to elements such as courtyard, climate response, social formation, and layering the levels of intimacy in housing. These elements had clear morphogenetic systems that were quickly dismantled during the 20th century by the homogenization strategies brought by western and Russian style planning. The research also embarked into studying elements such as land value strategies and Chinese national code.

The Berlage team began to search for low-rise high-density housing that followed the research of the synthetic principles of courtyards and traditional street design in Chinese cities. Several routines were coded in parametric software. These routines were populated by the previous research in the morphogenetic nature of traditional Chinese housing. The codes in the parametric software automatically negotiated the relationships of program layout with courtyard typologies, climate considerations, and national regulations that require that each house receives a minimum of 2hr. of direct sunlight during winter.

Moreover the parametric model also considers more sophisticated issues and automatically calculates the insulation and solar gain for each wall in the project. With this information you can began to define automatically the material configuration for each wall. The morphological digital model automatically generates parking, public spaces, water systems, resolves street corners, and create land value maps to insure that the neighborhoods are not segregated by income. The process is self-organized and performed so the relationships of courtyards and street are maintained but making sure that we can never encounter an exact repetition.

Case 2: Metaheuristic Tools and Processes at Aedas R&D

The R&D group inside Aedas Architecture is a team inside one of the largest firms in the world. The group is project-driven, it works aiding the firms architects explore generative and analytical computational processes in design. The group has worked in a variety of projects: from façade systems, performance analysis, digital layout, to large urban design proposals. The objective is to develop methods for design that explore the spatial and performative conditions of design more than just specific geometrical solutions for a project. Their work is closely related to academic ambitions explored in schools such as at the University of East London, University of Central London. Although their work is not as holistic as the Berlage example presented above, they offer a first glimpse how a large practice is entering into a higher discourse of computerization.

In developing their tools and processes the group has developed metaheuristic techniques to augment the traditional analog heuristic methods or practical rules of thumbs used by project teams. In computer science, metaheuristic is a computational method that searches for a large number of candidate solutions. Metaheuristic is an iterative process that can search quickly a large number of candidate solution but that can not assure that an optimal solution can ever be found. Among the many computational methods that they have developed are the following:

Adjacencies and layout: The Computational Design team has developed several 3D tools to help designers understand adjacencies diagrams and program layouts. These tools are semi-automatic, not fixed, and the user can move bubbles and/or volumetric rooms while the adjacencies among functions are maintained. As the user moves the volumetric rooms they behave like 3D Jell-O boxes that attract or repel different configurations based on their topological configuration. These tools are intended to intensify the reflective period design teams have with the program layouts rather than provide fully optimized solutions (Derix 2010).

Digital master planning tools: The Computational Design Team at Aedas also has developed methods such as massing, accessibility and movement, strategic planning, investment appraisals, and others that have been implemented at the urban scale. Two critical issues have emerged in the creation of these digitally assisted methods. The first is that users always continue to ask for more features to be added to the computer model. This creates a major visualization problem because these systems can became overwhelmed with information and the clarity and simplicity of the information can easily be lost. The second theme emerges with the potential temptation to develop optimization procedures. These systems are developed using a multi-criteria development and often there is no clear way to offer a family of optimized solution. So an option is for the methodology not to provide any solutions and focus in usability and engagement criteria with the user. The observations in usability became important and it usually critical to understand the type of supervision these tool requires. Sometimes computers run too fast, and it is better for the user to see how it struggles for a solution. At that moment the users can see potential candidate solutions and by accident help move along different scenarios in the discussions that accompany a typical planning process.

Figure 2. Videos and Images of the program layouts and digital master planning methods created at Aedas R&D. Scan QR-code with a Smart Phone or use this web link: http://x.co/bSsn

NEW COMPUTATIONAL METAPHORS

The digital methods developed by the Aedas R&D group do not have the holistic approach of the academic project at the Berlage Institute. The computerization projects at Aedas are small and more specific to the particular tasks that support the design teams inside the company. However, one can observe that both cases have the ability to challenge the traditional heuristic processes of designers. The Berlage and the Aedas digital techniques are build incrementally. They are more than augmenting tools for designers. They are based on observations from designers about the design cases and are techniques to allow the designers to reflect in practice.

Both cases avoid producing optimal results and allow all participants in the process to think about "what if" scenarios. The digital methods try offer extra interpretations. These methods do not necessarily speed the process of design. Instead they are helpful to provide contradictory observations. For example the Aedas "layout and adjacency" tools are a medium in which a designer can play to see where they were not able to see before.

Finally, both cases are perhaps more radical in one particular issue. They move away from the current metaphor of computerization in architecture in which few large monopolistic software vendors control the discourse of computerization of the whole industry. They move parametric thinking into undirected processes and into the every day tribulations that face practice of architecture. According to Christian Derix the head of the computational design group at Aedas R&D says that these tools "requires constant learning of dedicated teams that can construct frameworks of computational heuristics within live design contexts (hence industry has an edge over academia in this field of research)" (Derix 2010).

CONCLUSION: CODING DESIGN REFLECTION IN ACTION

Both cases divert from the contemporary geometric BIM paradigm and the highly theoretical field of architectural computing presented here. They are not incredible optimistic about optimization and automation of the design process. They are not interested in generating problem solving tools but are deeply interested into entering into the politics of the topological nature of how we have constructed our heuristic thoughts through history.

Paradoxically, both cases presented here will probably be attacked for promoting an automated vision of design and narrowing the design possibilities, eliminating poetry, and giving restricting solutions - killing by default the highly treasured democratic right for the genius artist-architect to exist. However, the state of the contemporary generic city is the result of the failure of the profession to respond in a meaningful way to the extremely successful typologies such as malls, retail boxes, and suburbs that topologically reproduce almost everywhere.

Both cases presented here show a new exit for the profession. An intellect, a level of brain power that architecture did not had previously. Architects that deal with housing in China today can only write books about the vernacular. With the current methods they can not elucidate a workable morphology that could compete with the mid-rise and high-rise housing products the market copies and pastes ubiquitously. The Deleuzian topological model and population thinking suggested in these prototypes points to a new speciation of inhabitation.

Within this framework, we can also imagine the potential connection that these cases could have with highly integrated BIM networks and PIM databases. They may aid design teams to obtain very accurate costs scenario evaluations. Architecture, with these new powers, can aim at more complex issues in urbanity today.

Since the early 1990s Architects have used computing merely to generate more and more complex forms. This addictive practice kidnapped the avant-garde into a worldwide competition for the formally spectacular. High-end architecture became the race for the new geometry. Promoted by a network of cultural pimps, star-architecture turned into the ultimate trophy of the cultural, Olympic dramas, petroleum royalties, and CEO elites. The formal twists, bends, and splits are a hit in the world of media but it has barely touched our every-day life. The prime-time influence of star-architecture is limited. It allows for only one Frank Gehry per city. Form-finding in generative systems that search for higher intelligence of human settlements, beyond just geometry, can expose our way of working and reform our urban DNA.

REFERENCES

Andia, Alfredo. “Reconstructing the Effects of Computers on Practice and Education during the past Three Decades.” *Journal of Architectural Education* 56.2 (2002): 7-13.

Coates, Paul. *Programming.architecture*. London: Routledge, 2010.

Deleuze, Gilles, and Félix Guattari. *A Thousand Plateaus: Capitalism and Schizophrenia*. Minneapolis: University of Minnesota, 1987.

Derix, Christian. “Mediating Spatial Phenomena through Computational Heuristics.” *ACADIA 10: LIFE In:formation* (2010).

Eastman, Charles M. *Building Product Models: Computer Environments Supporting Design and Construction*. Boca Raton, FL: CRC, 1999.

Frazer, John. *An Evolutionary Architecture*. London: Architectural Association, 1995.

McCullough, Malcolm, William J. Mitchell, and Patrick Purcell. *The Electronic Design Studio: Architectural Education in the Computer Era*. Cambridge, MA: MIT, 1990. Print.

Mitchell, Melanie. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT, 1996.

Rittel, Horst W. J., and Melvin M. Webber. “Dilemmas in a General Theory of Planning.”*Policy Sciences* 4.2 (1973): 155-69.

Schön, Donald A. *The Reflective Practitioner How Professionals Think in Action*. New York: Basic, 1984.