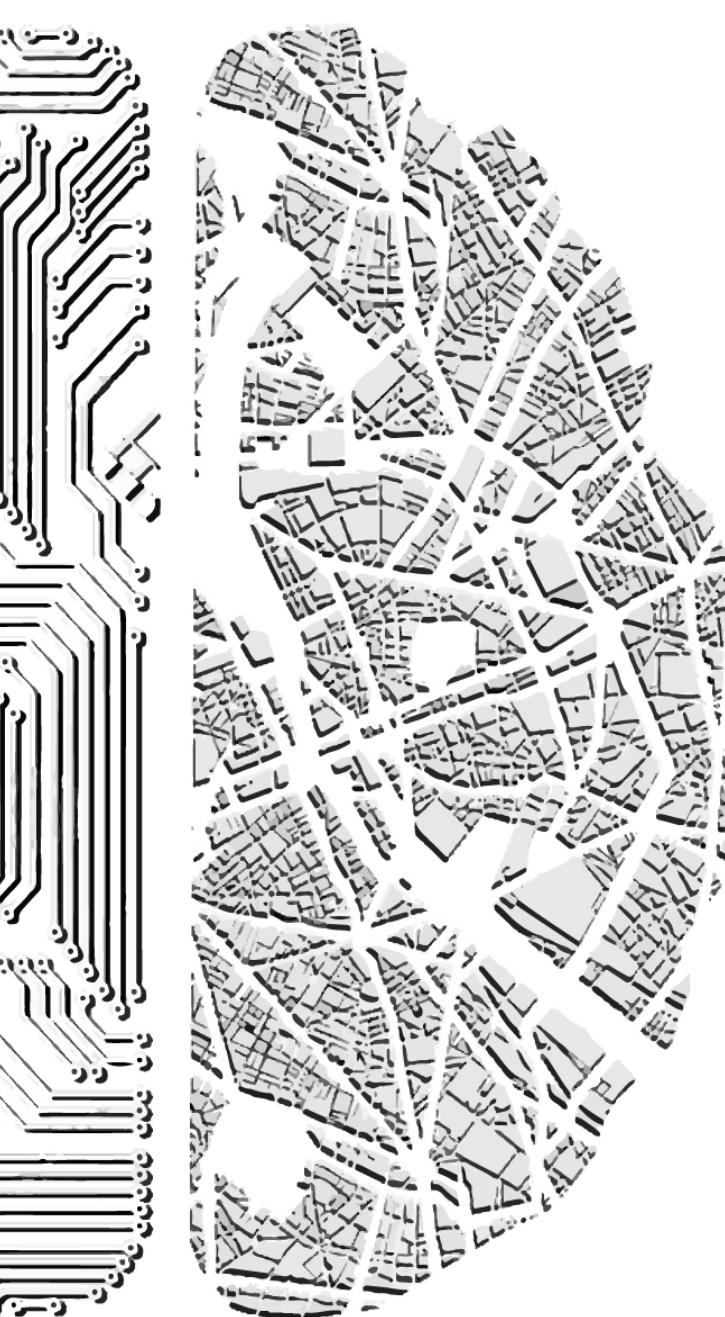


The Advent of Architectural AI

Stanislas Chaillou | Harvard Graduate School of Design



Architecture + AI

Towards a New Approach

The practice of Architecture, its methods, traditions and know-how are today at the center of passionate debates. Challenged by outsiders, arriving with new practices, and questioned from within, as practitioners doubt of its current state, Architecture is undergoing a truly profound (r)evolution.

Among the factors that will leave a lasting impact on our discipline, technology certainly is one of the main vectors at play. The inception of technological solutions at every step of the value chain has already significantly transformed Architecture. The conception of buildings has in fact already started a slow transformation: first by leveraging new construction technics, then by developing adequate softwares, and eventually today by introducing statistical computing capabilities (including Data Science & AI). Rather than a disruption, we want to see here a continuity that led Architecture through successive evolutions

until today. Modularity, Computational Design, Parametricism and finally Artificial Intelligence are to us the four intricated steps of a slow-paced transition. Beyond the historical background, we posit that this evolution is the wireframe of a radical improvement in architectural conception.

A. A Four-Period Sequence

Modularity, Computational Design, Parametricism and finally Artificial Intelligence are not air-tight steps, independent of one another: each period interpenetrates and borrows from the precedents. It is why, when looking backward at history it is critical to distinguish two levels of creation: inventions & innovations. Inventions stem from academic research, while innovations are induced by inventions. In architecture, Innovations actually shape a continuously moving practice. A practice which has been playing on the back and forth between periods, inventions & innovations. From there, our chronology aims at demonstrating the deeply interwoven evolutions of the computational and the architectural fields before introducing the age of architectural-AI, as a culminating point. It is why rebuilding the context and the highlights of the recent history of our discipline is a prerequisite to our work.

Modular Systems

Modularity could be set as the starting point of systematic architectural design. Initiated in the early 30's, the advent of modular construction brought to the conception phase both a language and an architectural grammar, contributing to simplify and rationalize building design.

Theorized for the Bauhaus by Walter Gropius, as early as 1920, the modular grid was carrying the hope of technical simplicity and affordability. Coming from different directions, modularity arose at first as a topic of investigation for academics and practitioners. Gropius initially introduced the idea of "Baukasten", a typical module to be then aggregated through strict assembly rules. This systematicity will be echoed one year later with Le Corbusier's "Modulor". By applying the modular rigor down to the human scale, Le Corbusier, as of 1946, offered a holistic implementation of the modular principles. The built environment dimensions would be aligned on key metrics and ratios derived from the human body. And indeed, from "La Tourette" to the "Unité d'habitation" in Marseille, Le Corbusier systematized the dimensions and spans to match the prescription of the "Modulor".

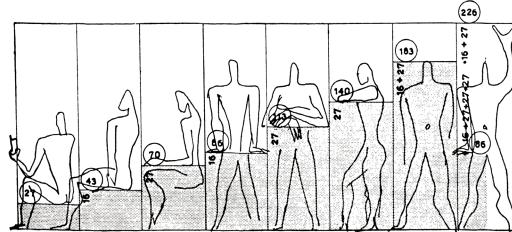


Figure 1: Corbusier's Modulor

With Buckminster Fuller, however, Modularity rapidly evolved towards a more integrated vision embedding building systems within the module as exemplified by the Dymaxion House. This attempt pushed to its extreme the possibility of modular housing, setting a vibrant precedent, and proof-of-concept for the industry.

Thereafter, following these early theorists, architects were invited to bend their design ethos to the imperative of the matrix, and by the same token, to

transfer part of the technicality of building design to the logic of the module. Less hassle, less costs, more predictability. Modularity would then swiftly extend to the industry as a whole: the Winslow Ames House, built by Professor Robert W. McLaughlin in 1933, and first large-scale modular project in the world, was perceived as a major breakthrough, as much as the very expressive Habitat 67 from Moshe Safdie. City

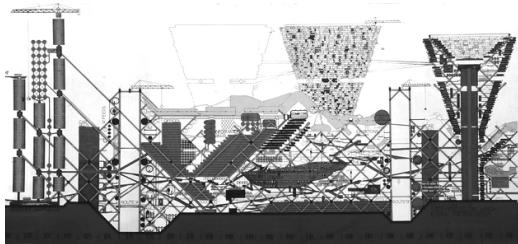


Figure 2: Plugin City by Archigram

— 4 —

planning even got influenced at the turn of the 60's, when projects like the "Plugin City" of Archigram developed the possibility of modular cities. Through the constant assemblage and dismantlement of modules, fitted on a three-dimensional structural matrix, cities could find a renewed logic, addressing both the possibility of growth and -as always- the imperative of feasibility.

However, connecting the grid, the modules, and the assembly systems through mechanistic rules eventually led to a quasi-gamification of a LEGO-like conception of Architecture. But practice cannot be just a "put together" board game aggregating a set of basic assembly rules and processes. The monotony of the resulting designs rapidly trivialized the theory, and the constructive weakness of its assembly systems finally discouraged architects. Nevertheless, through

its system of rules, Modularity is remaining today an underlying constructive principle still vivid throughout the practice.

Computational design

At the turn the 80's, as the complexity of modular systems was soaring, the advent of computation brought back feasibility and scalability to modular design. Beyond the resurrection of the module, the systematicity of rule-based design was somehow rehabilitated.

Coming from different directions, a high-level reflection about the potential of computational design started as early as the mid-50's, within an adjacent discipline: Engineering. In 1959, Professor Patrick Hanratty released PRONTO, the first prototype of CAD (Computer Assisted Drawing) software, geared towards engineered parts design. The possibility offered by such software, coupled to the potential computational power fast paced evolution, jump-started a discussion within the architectural field. Soon after, Christopher Alexander, architect and then professor at U.C. Berkeley started the discussion by laying down the key principles for Computational Design. In his "Notes on the Synthesis of Form" (1964) and later, in "A Pattern Language" (1968), Alexander theorized why and how computers should be used to address the question of shape design. His early understanding of software potential for design, was deeply contrasting with the hardware-centric focus at the time. The founding principles he defined in his book are still today the bedrock of software programming: concepts like recursions, object-oriented programming as well as their application to design have represented a radical

move forward. Following this momentum, an entire generation of computer scientists and architects will create a new field of research: Computational Design. The Architecture Machine Group (AMG) at



Figure 3: URBAN 5, AMG MIT

MIT, led by Professor Nicholas Negroponte is probably its most exemplary embodiment. Negroponte's book "The Architecture Machine" (1970) encapsulates the essence of the AMG's mission: investigating how machines can enhance the creative process, and more specifically, the architectural production as a whole. Culminating with the release of projects URBAN II and later URBAN V, this group will then demonstrate, even before industry would engage in any effort, the potential of CAD applied to space design.

Following such conclusive research, architects and the industry at large actively pushed these inventions to the state of innovations. Frank Gehry certainly was

the most vibrant advocate of the cause. For him, the application of computation could drastically relax the boundary of assembly systems and allow for new shapes & building geometries. Gehry Technologies, founded by Gehry and Jim Glymph in the 80's typically used early CAD-CAM software - such as CATIA - from Dassault Systems to tackle complex geometric problems. Setting here the precedent for 30 years of Computational Design, Gehry Technology would demonstrate the value of computation to architects, provoking a landslide in the profession. Over the next 15 years, the irresistible growth of computational power & data storage capacities, combined with increasingly affordable and more user-friendly machines, massively facilitated the adoption of 3D-design software. Architects rapidly endorsed the new system on the base of a clear rationale: Computational Design (1) allows a rigorous control of geometry, boosting design's reliability, feasibility and cost, (2) facilitates and eases collaboration among designers, (3) and finally enables more design iterations than traditional hand-sketching could afford. More tests & more options for better resulting designs.

However, along the way, as designers were engaging with Computational Design, a couple of shortcomings eventually arose. In particular, the repetitiveness of certain tasks, and the lack of control over complex geometric shapes became serious impediments. Those paved the way to a brand-new movement which was emerging within Computation Design: Parametricism.

Parametricism

In the world of parameters, both repetitive tasks and

complex shapes could possibly be tackled, when rationalizable to simple sets of rules. The rules could be encoded in the program, to automate the time-consuming process of manually implementing them. This paradigm drove the advent of Parametricism. In few words, if a task can be explained as a set of commands given to the computer, then the designer's task would be to communicate them to the software, while isolating the key parameters impacting the result. Once encoded, the architect would be able to vary the parameters and generate different possible scenarios: different potential shapes, yielding multiple design outputs at once.

In the early 1960, the advent of parametrized architecture was announced by Professor Luigi Moretti. His project "Stadium N", although theoretical initially, is the first clear expression of Parametricism. By defining 19 driving parameters - among which the spectators' field of view and sun exposure of the tribunes -, Moretti derived the shape of the stadium directly from the variation of these parameters. The resulting shape, although surprising and quite organic, offers the first example of this new parametric aesthetic: organic in aspect, while strictly rational as a conception process. Bringing such principle to the world of computation will be the contribution of Ivan Sutherland, three years later. Sutherland is the creator of SketchPad, one of the first truly user-friendly CAD software. Embedded at the heart of the software, the notion of "Atomic Constraint" is Sutherland's translation of Moretti's idea of parameter. In a typical SketchPad drawing, each geometry was in fact translated on the machine side into a set of atomic constraints (parameters). This very notion is the first formulation of parametric design in

computer's terms. Samuel Geisberg, founder of the Parametric Technology Corporation (PTC), would later, in 1988, roll out Pro/ENGINEER, first software giving full access to geometric parameters to its users. As the software is released, Geisberg perfectly summed up perfectly the parametric ideal:

"The goal is to create a system that would be flexible enough to encourage the engineer to easily consider a variety of designs. And the cost of making design changes ought to be as close to zero as possible."

Now that the bridge between design and computation was built thanks to Sutherland and Geisberg, a new generation of "parameter-conscious" architects could thrive. As architects were becoming more and more capable of manipulate their design using the proxy of parameters, the discipline "slowly converged" to Parametricism, as explained by P. Schumacher. In his book, "Parametricism, a New Global Style for Architecture & Urban Design" Schumacher explicitly demonstrated how Parametricism was the result of a growing awareness of the notion of parameters within the architectural discipline.

From the invention of parameters, to their translation into innovations throughout the industry, we see a handful of key individuals, who have shaped the advent of Parametricism. This parametrization of architecture is best exemplified at first by Zaha Hadid Architects' work. Mrs. Hadid, an Iraqi architect trained in the UK, with a math background would found her practice, with the intent to marry math and architecture through the medium of parametric design. Her designs would typically be the result of rules, encoded in the

program, allowing for unprecedented levels of control over the buildings' geometry. Each architectural move would be translated into a given tuning of parameters, resulting in a specific building shape. Hadid's designs are the perfect examples to this day of the possible quantification of architectural design, into arrays of parameters. Her work however would have not been possible without Grasshopper, software developed by Mark Rutten in the year 2000's. Designed as a visual

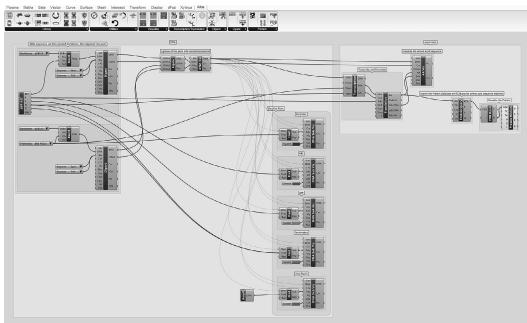


Figure 4: Grasshopper by Marc Rutten

programming interface, Grasshopper allows architects to easily isolate the driving parameters of their design, while allowing them to tune them iteratively. The simplicity of its interface coupled with the intelligence of the built-in features continues today to power most buildings' design across the world and has inspired an entire generation of "parametric" designers. Finally, beyond the short-term benefits of Grasshopper for building design, a more profound revolution, driven by parametrization and started in the early 2000's, is still underway today: BIM (Building Information Modeling). Spearheaded by Philip Bernstein, then Vice President of Autodesk, the birth and refinement of BIM has brought rationality and feasibility to a brand-new level within the construction industry. The underlying idea of the BIM is that every element in a building 3D

model is function of parameters ("properties") that drives each object's shape and document them. From Autodesk Revit -the major BIM software today- to Sutherland's SketchPad, we see a single common thread: the explicit utilization of parameters as the driving force of design.

However, the parametrization of design has proven over the past 10 years to have reached a plateau, both technically and conceptually. Parametric modeling failed to account for (1) the compounded effect of multiple variable at once, (2) the imperative of space organization and style over strict efficiency, (3) the variability of scenarios, and finally (4) the computational cost of simulations. Independently from its technical shortcomings, parametric design is flawed by its theoretical premise: Architecture could be the result of a fixed number of parameters, that the architect could simply encode, as an abstraction, away from its context, its environment and its history. In fact, Parametricism, when applied 'by the book', proved to neglect the immense complexity of space planning: countless parameters, and profound cultural & societal factors actually participate to the urban equilibrium. This deep reality, combining adjacent disciplines in a systemic way, can today finally be addressed, as our profession encounters Artificial Intelligence.

Artificial Intelligence: a Statistical Approach to Architecture

Artificial Intelligence is fundamentally a statistical approach to architecture. The premise of AI, that blends statistical principles with computation is a new approach that can improve over the drawbacks of parametric architecture.

"Learning", as understood by machines, corresponds to the ability of a computer, when faced with a complicated issue, first to grasp the complexity of the options showed to him and second to build an "intuition" to solve the problem at stake. In fact, when coining down the concept of AI, John McCarthy, back in 1956, has defined it as "using the human brain as a model for machine logic". Instead of designing a deterministic model, built for a set number of variables and rules, AI lets the computer create intermediary parameters, from information either collected from the data or transmitted by the user. Once the "learning phase" achieved, the machine can generate solutions, not simply answering a set of predefined parameters, but creating results emulating the statistical distribution of the information showed to him during the learning phase. This concept is at the core of the paradigm shift brought by AI. The partial independence of the machine to build its own understanding of the problem, coupled with its ability to digest the complexity of set of examples, turns up-side down the premise of Parametricism. Since not all rules & parameters are declared upfront explicitly by the user, the machine can unexpectedly unveil underlying phenomena and even try to emulate them. It is quantum leap from the world of heuristics (rule-based decision making) to the world of statistics (stochastic-based decision making).

The penetration of Artificial Intelligence in the architectural field was already forecasted early on by a few theorists, who, before us, saw AI's potential for architectural design. Far from crafting intelligent algorithms, these precursors designed and speculated on the potential of such systems. As URBAN II was released by Negroponte and his group, the idea of a "machine assistant" was already well underway.

URBAN V, a later version, would assist the designer, by adapting rooms layout- defined as blocks - to optimize adjacencies and light condition as the user draws onto a modular grid. In fact, URBAN V distinguished two layers of information: implicit and explicit. The implicit dimension is the one handled and deduced by the machine, while the explicit one is the dimension set by the user. This duality of information in URBAN V is the direct translation of the machine-human complementarity wished by Negroponte. And it is within the set of implicit parameters, that the "intelligence" - in other words the AI- built within the machine would find its expression. Corrections

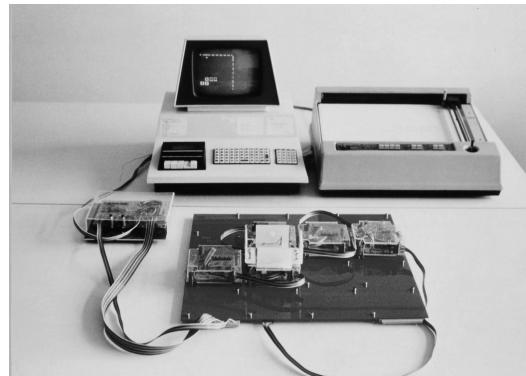


Figure 5: GENERATOR, Cedric Price

proposed by the computer, by tuning the implicit parameters, would be surfaced to the users as suggestions. To an ill-placed set of rooms, URBAN V would notify the user: "TED, MANY CONFLICTS ARE OCCURRING". A few years later, Cedric Price, then Professor at the chair of Architecture at Cambridge University, invented the GENERATOR (1976). Acknowledging Negroponte's work, Price used the AMG's work on AI and pushed it further, investigating the idea of autonomous ever-changing building, that

would “intelligently” respond and adapt to users’ behaviors. For Price, under the term “intelligent” lies the idea of encoding a behavior, that the GENERATOR would follow. However, below Negroponte’s work, or Price’s prototypes, lied an unresolved issue: the actual intelligence of the algorithm. Although the interface and protocols were in place, the actual procedural complexity of the core algorithms were still quite weak, based on simple heuristics relationships.

The design of intelligent algorithms, also called AI, actually found a renewed interest at the beginning of the 80’s. The sudden increase in computational power, and the steep increase of funding’s brought back the question of intelligence at the center of AI’s investigation. Key to this period are two main revolutions: expert systems and inference engines. The former corresponds to machines able to reason based on a set of rules, using conditional statements. An actual breakthrough at the time. The latter, best exemplified by the Cyc Project, developed by Douglas Lenat, were involving machines geared towards inference reasoning. Using a knowledge base (a set of truth statements), an inference machine would be able to deduce the truthfulness of a new statement as compared to its knowledge base. It is not until the early 90’s, and the mathematization of AI that the

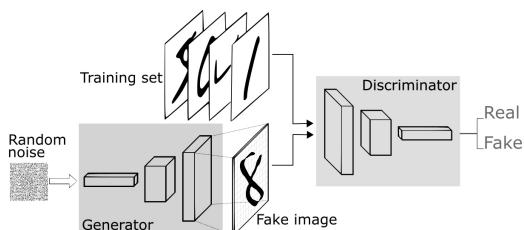
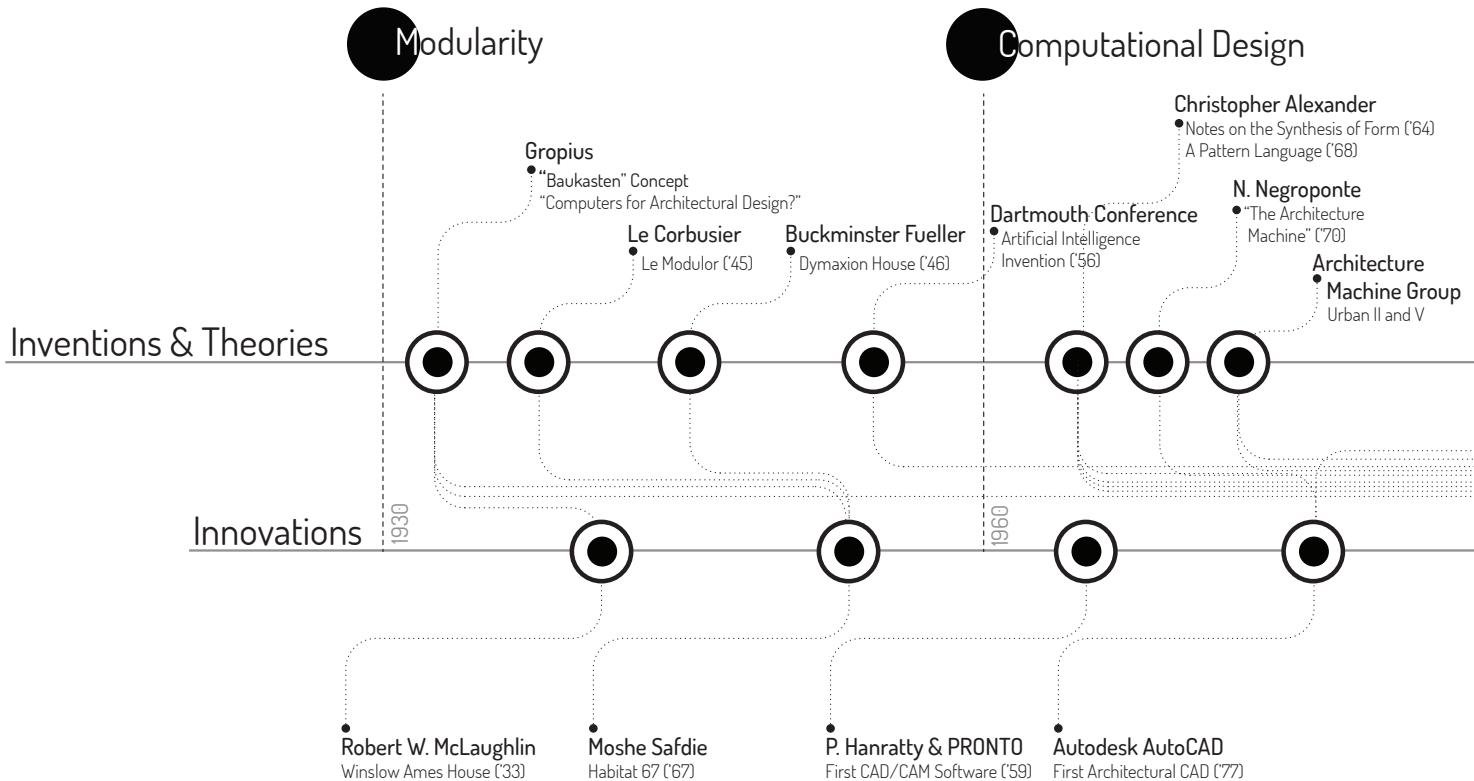
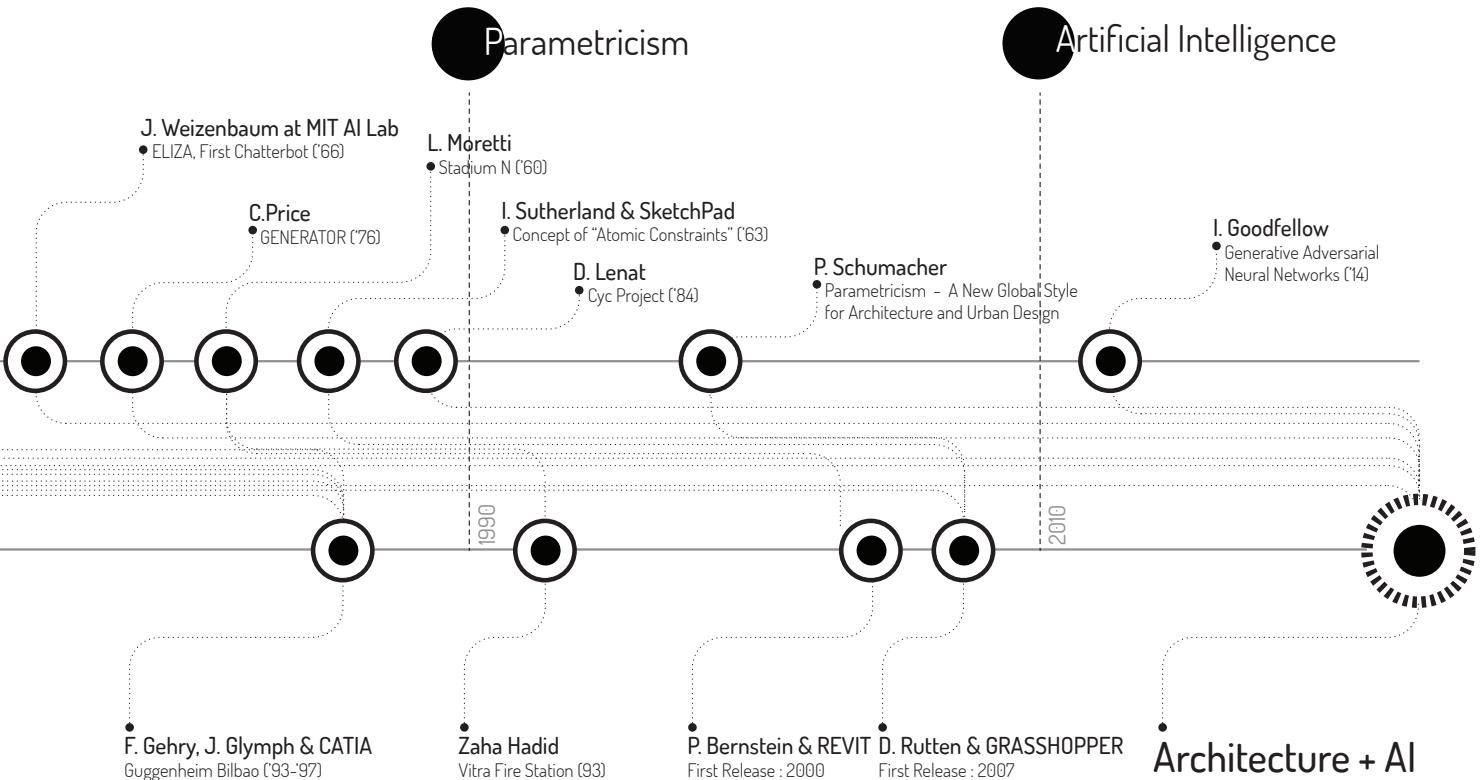


Figure 6: GAN Typical Architecture

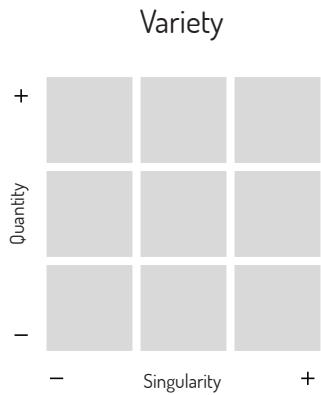
field would bring truly promising results. The advent of a new type of models would definitely reveal AI’s potential: networks and machine learning. Through the utilization of a layered pipeline, also called network, a machine is now able to grasp higher complexities than previously developed models. Such models can be “trained”, or in other words, tuned for specific tasks. More interesting even, is the idea embarked in one specific type of such models: Generative Adversarial Neural Networks (GANs). Theorized at first by Ian Goodfellow, researcher at Google Brain, in 2014, this model offers to use networks to generate images, while insuring accuracy through a self-correcting feedback loop. Goodfellow’s research turns upside down the definition of AI, from an analytical tool, to a generative agent. By the same token, he brings AI one step closer to architectural concerns: drawing and image production. All in all, from simple networks to GANs, a new generation of tools coupled to increasingly cheaper and accessible computational power is today positioning AI as an affordable and powerful medium. If Negroponte’s or Price’s work were almost empty of true machine intelligence, nowadays architectural software can finally leverage such possibility.

Although the potential AI represents for Architecture is quite promising, it still remains contingent on designers’ ability to communicate their intent to the machine. And as the machine has to be trained to become a reliable “assistant”, architects are faced with two main challenges: (1) they have to pick up an adequate taxonomy i.e. the right set of adjectives that can translate into quantifiable metrics for the machine and (2) they must select, in the vast field of AI, the proper tools and train them. Those two preconditions will eventually determine the success or the failure of AI-enabled architecture.

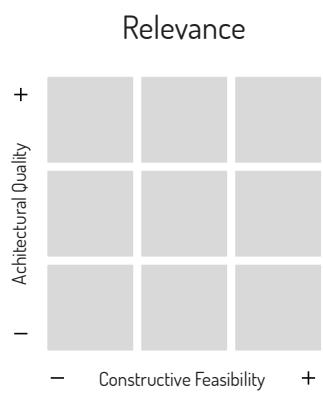




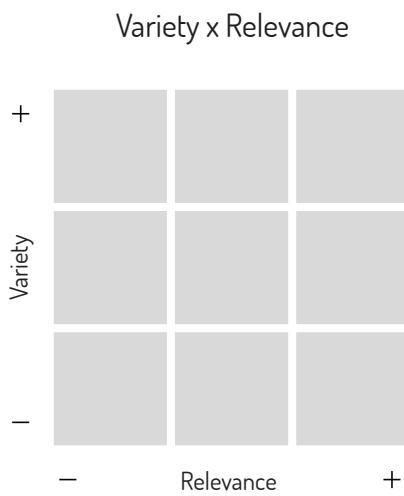
— 12



X



||



B. A Continuous Progress

Modularity, Computational Design, Parametricism and Artificial Intelligence: this four-period sequence reflects the chronology of the progress which, step-by-step, has been shaping and refining the architectural means & methods. We want to see in such momentum a form of “continuous progress” as the one experienced in the industry at large, rather than a series of unrelated disruptions. From there, an appropriate set of matrices has helped us mapping this dynamic.

First, to evidence our claim, we posit here that Architecture can be understood as a process of generating designs one can describe through two dimensions: on one side the diversity of the output produced or “Variety” and on the other side the applicability of the designs or “Relevance”.

“**Variety**” is contingent upon two underlying metrics: the “quantity of designs” sizing the volume of options created and the “singularity of designs” measuring their respective disparity.

“**Relevance**” compounds the “constructive feasibility”, i.e. the workability of the designs and their “architectural quality” including optimal program organization, space layout and contextual fit.

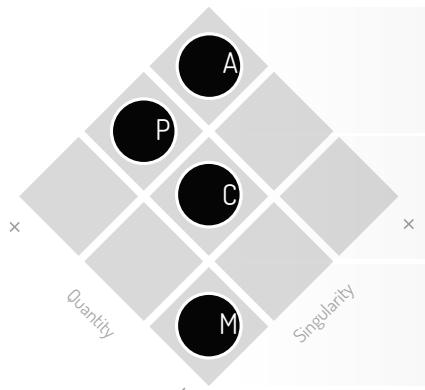
Ultimately, the combination of Variety (Quantity X Singularity) and Relevance (Constructive Feasibility X Architectural Quality) creates a framework which (1) maps out and contrasts the respective positioning of our four periods -Modularity, Computational Design, Parametricism and Artificial Intelligence and (2)



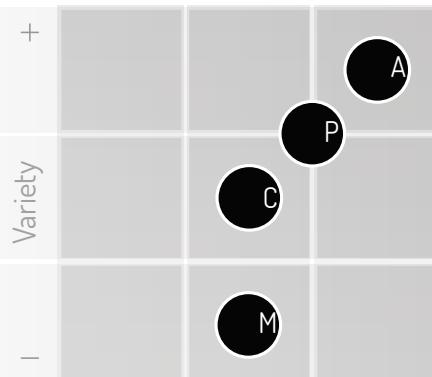
clearly evidences the culminating point of progress AI represents for our discipline.

This display, although directional and qualitative, is a powerful grid to represent the concept which lies at the heart of our thesis. In summary, the dynamic of this continuous progress has been triggered by the limit of each movement at a certain point in time, exacerbated by the competition of the new one

Variety

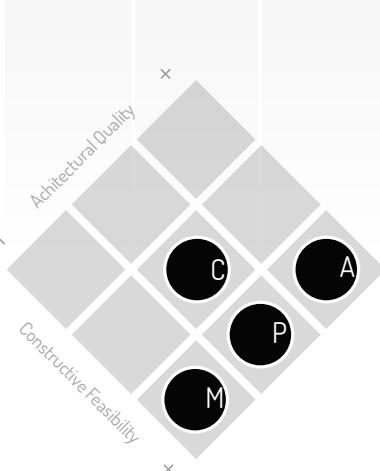


Variety x Relevance



Architectural Quality
Constructive Feasibility

Relevance



Modularity

Since its inception, modular conception has proved to be a highly constraining system, yielding low variety of potential design options. Although such options were easily meeting construction feasibility criteria, their architectural quality got questioned early on.

Computational Design

With computation design, designer could finally afford to escape the rigidity of the grid, to design feasible buildings, with actual singularity. Conception software, allowing to solve complex shapes, would help drive down conception costs and generate more design iterations.

Parametricism

Parametricism brought even more control over organic shapes, increasing constructive feasibility. By systematizing their geometry, entire buildings could be discretized into buildable elements, and resolved assembly systems. The parametric style, however, got rapidly trivialized, generating generic and repetitive patterns spaces leading to poor architectural quality.

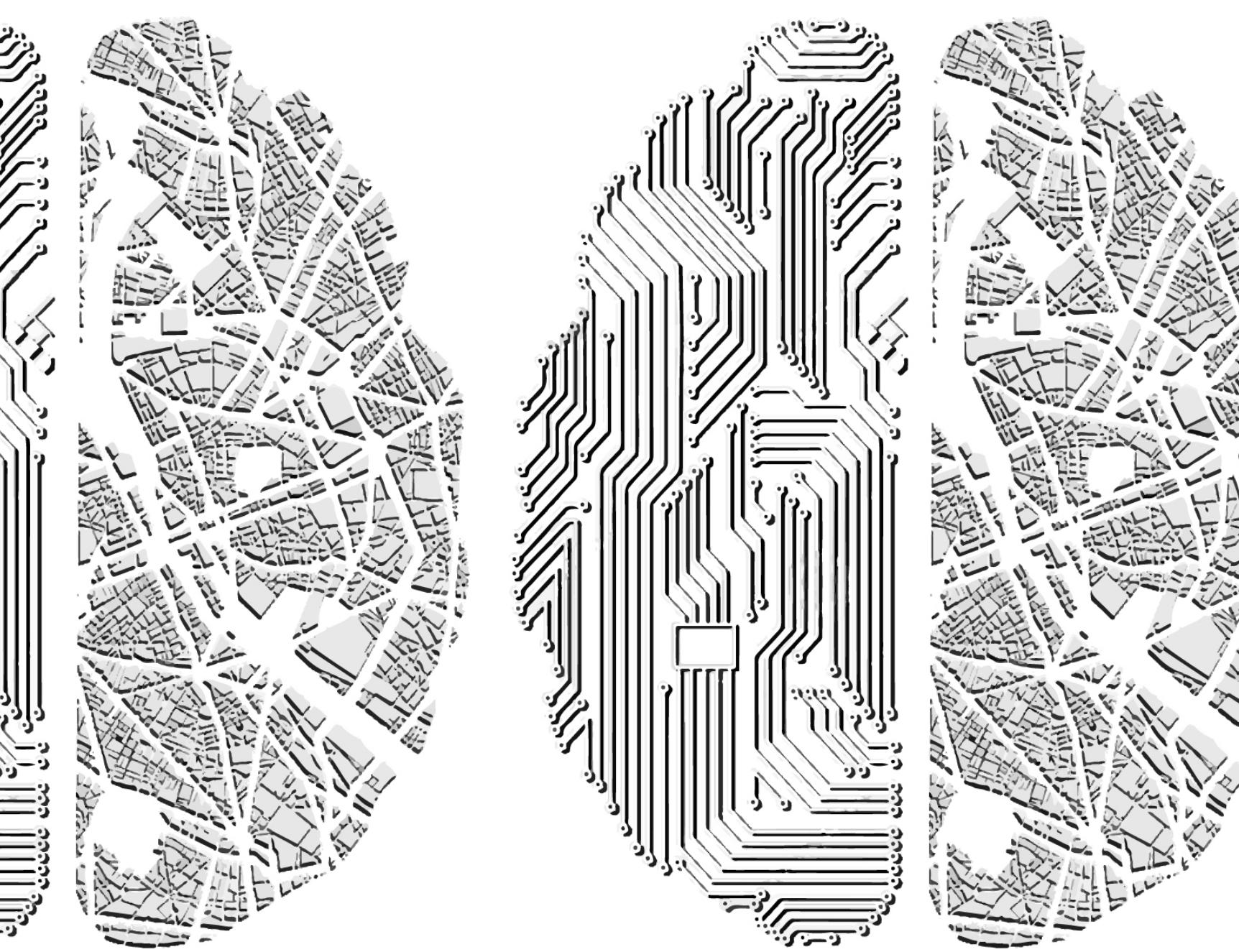
Artificial Intelligence

Artificial intelligence finally carries the promise of embarking all the benefits of previous generations, while escaping the legibility and generic style of Parametricism. It is the ultimate push towards greater architectural quality and, to us the springboard of a brand-new era.

Closing Remark

We are today faced with a fantastic challenge: bringing AI to the world of architectural design. There is no doubt that AI will never—could never—automate the architect's intuition and sensibility. The human uses the machines as its tool. Not the other way around. However, benefiting from an intelligent assistant is within our reach and should be carefully studied, tested and experienced.

Moreover, AI is simply not the mere result of a sudden disruption. It is the culminating point of 70 years of inventions & innovations. As AI can balance efficiency and organicity while providing tremendous varieties of relevant design options, we see in it the possibility for rich results, that will complement our practice and address some blind spots of our discipline. Far from thinking about AI as a new dogma in Architecture, we conceive this field



Theoretical References

Digital Architecture Beyond Computers, Roberto Botazzi, Bloomsbury

Data-Driven Design & Construction, Randy Deutsch, Wiley

Architectural Intelligence, How Designers and Architects Created the Digital Landscape, Molly Wright Steenson, MIT Press

Architectural Google, Beyond the Grid—Architecture & Information Technology pp. 226–229, Ludger Hovestadt, Birkhauser

Algorithmic Complexity: Out of Nowhere, Complexity, Design Strategy & World View pp. 75–86, Andrea Gleiniger & Georg Vrachliotis, Birkhauser

Code & Machine, Code, Between Operation & Narration pp. 41–53, Andrea Gleiniger & Georg Vrachliotis, Birkhauser

Gropius' Question or On Revealing And Concealing Code in Architecture And Art, Code, Between Operation & Narration pp. 75–89, Andrea Gleiniger & Georg Vrachliotis, Birkhauser

Soft Architecture Machines, Nicholas Negroponte, MIT Press

The Architecture Machine, Nicholas Negroponte, MIT Press

A Pattern Language, 1968, Christopher Alexander

Notes on the Synthesis of Form, 1964, Christopher Alexander

Cartographic Metamorphologies; or Enter the Rowe-Bot, Andrew Witt, Log #36

Grey Boxing, Andrew Witt, Log #43

Suggestive Drawing Among Human and Artificial Intelligences, Nono Martinez, Harvard GSD Thesis, 2016

Enabling Alternative Architectures : Collaborative Frameworks for Participatory Design, Nathan Peters, Harvard GSD Thesis, 2017

Technical References

DANIEL: A Deep Architecture for Automatic Analysis and Retrieval of Building Floor Plans, Divya Sharma,Nitin Gupta,Chiranjay Chattopadhyay,Sameep Mehta, 2017, IBM Research, IIT Jodhpur

Automatic Room Detection and Room Labeling from Architectural Floor Plans, Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel, 2012, University of Kaiserslautern

Generative Adversarial Nets, Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, June 2014 ,Universite de Montreal

Pix2Pix, Image-to-Image Translation with Conditional Adversarial Networks, Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros

Automatic Interpretation of Floorplans Using Spatial Indexing, Hanan Samet, Aya Soffer, 1994, University of Maryland

Parsing Floor Plan Images, Samuel Dodge,Jiu Xu ,Bjorn Stenger, 2016, Arizona State University, Rakuten Institute of Technology

Project Discover: An Application of Generative Design for Architectural Space Planning, Danil Nagy, Damon Lau, John Locke, Jim Stoddart, Lorenzo Villaggi, Ray Wang, Dale Zhao and David Benjamin, 2016, The Living, Autodesk Studio

Raster-to-Vector: Revisiting Floorplan Transformation, Chen Liu,Jiajun Wu,Pushmeet Kohli,Yasutaka Furukawa, 2017, Washington University, Deep Mind, MIT

Relational Models for Visual Understanding of Graphical Documents. Application to Architectural Drawings, Lluís-Pere de las Heras, 2014, Universitat Autònoma de Barcelona

Shape matching and modeling using skeletal context, Jun Xie,Pheng-Ann Heng,Mubarak Shah, 2007, University of Central Florida,Chinese University of Hong Kong

Statistical segmentation and structural recognition for floor plan interpretation, Lluís-Pere de las Heras, Sheraz Ahmed, Marcus Liwicki, Ernest Valveny, Gemma Sánchez, 2013, Computer Vision Center, Barcelona, Spain

Unsupervised and Notation-Independent Wall Segmentation in Floor Plans Using a Combination of Statistical and Structural Strategies, Lluís-Pere de las Heras, Ernest Valveny, and Gemma Sanchez, 2014, Computer Vision Center, Barcelona, Spain

Path Planning in Support of Smart Mobility Applications using Generative Adversarial Networks, Mohammadi, Mehdi, Ala Al-Fuqaha, and Jun-Seok Oh . , 2018

Automatic Real-Time Generation of Floor Plans Based on Squarified Treemaps Algorithm, Fernando Marson and Soraia Raupp Musse, 2010, PUCRS

Procedural Modeling of Buildings, Pascal Muller,Peter Wonka,Simon Haegler,Andreas Ulmer,Luc Van Gool, 2015, ETH Zurich, Arizona State University

Generative Design for Architectural Space Planning, Lorenzo Villaggi and Danil Nagy, 2017, Autodesk Research

The Future of Artificial Intelligence

Semiotics Collection | Harvard Graduate School of Design

