The Argument for the Argument  
Revisiting the Architecture Machine  

Brian Lonsway  
Carleton University  

Abstract  

The argument for the argument, that is, the defense of the computable quantity, has had a profound impact on the contemporary understanding of design practice. In this paper, I explore the history of the architectural-computational 'argument' to uncover a generally accepted yet poorly understood collusion between architectural and urban theory, structuralist semantics, and computation. From arguments about the machine to the mechanics of language, and from the language of architecture to the architecture machine, the argument for the argument has radically transformed contemporary design practices, but neither the history nor the theory behind these developments has been critically examined. My own argument seeks to build upon this nexus a hypothetical construct — a post-structuralist computer — as a provocation of sorts: a challenge to contemporary computational work in architecture to critically and philosophically address its current trajectories.

Introduction  

I’d like to present an argument on the numerical argument – the computational argument – within the practice of architecture. More precisely, I’d like to make an argument about the arguments for this computational argument. For there exists a particular history of such arguments that has largely gone under-theorized, but which has had a profound impact on the way we employ computation in architecture today. Specifically, my argument is on the historical collusion between architecture, structuralist semantics, and computation. Building upon this nexus, which historically brought computation to architecture and architecture to computation, I will seek to inform a critical understanding of one of architecture’s most socially transformative effects.

The critical history of the scientization of what we now know as the humanities is well known, introduced to many architects via the work of Michel Foucault. Through the efforts of early industrialization, themselves carrying on the rationalist traditions of the western enlightenment, a desire for the objective validation of manufacturing techniques became increasingly prevalent (Noble...
Brian Lonsway
The Argument for the Argument: Revisiting the Architecture Machine

Foucault traces such efforts in the histories of medicine, psychology, the social sciences, and perhaps most well known to architects, incarceration (Foucault 1973, 1965, 1970, 1977). It is in *Discipline and Punish* where Jeremy Bentham's turn of the 19th century work on the panopticon received significant attention, and drew the interest of architects, primarily because of the spatial clarity of Foucault's project. In all of these works, Foucault addresses the role of the burgeoning scientific-rational project in identifying and constructing the human subject, and has placed spatial metaphors most generally, and architectural projects most specifically, at the core of this pursuit. While, outside of *Discipline and Punish*, architecture itself is rarely a specific subject of study, Foucault's work has much to say about the way that the scientific-rational project has transformed the practice of architecture.

In *The Order of Things*, Foucault describes, in a typically ironic mode, a three-dimensional Cartesian space to represent the scientistic advance of the humanities. This space is marked with three axes: “mathematical and physical sciences”; “sciences such as those of language, life, economics”; and “philosophical reflection.” The human sciences all:

> have the more or less deferred, but constant, aim of giving themselves, or in any case of utilizing, at one level or another, a mathematical formalization; they proceed in accordance with models or concepts borrowed from biology, economics, and the sciences of language; and they address themselves to that mode of being of man which philosophy is attempting to conceive a the level of radical finitude, whereas their aim is to traverse all its empirical manifestations (Foucault 1970:347).

It is within this ironically structured model of the human sciences that I would like to position a key development of twentieth century architectural theory, one that has an intrinsic relationship to computation, and which has radically transformed the design, perception, and inhabitation of our contemporary spatial environments: the architecture machine. Rather than a specific apparatus, the architecture machine is a historically amorphous scientific construct, a kind of vague yet quantitatively defined idea around which significant theoretical discourses have unfolded. Discourses highly relevant to understanding the current state of architectural computation.

**Architecture Machines**

The machine – a material manifestation of the scientific practices of engineering – has particular importance in the history and theory of architecture, primarily due to a single statement by Le Corbusier. His *machine à habiter*, popularized in his 1923 collection of essays *Vers une Architecture*, set the metaphor of the machine in a central position in the discourse of architectural modernism (Corbusier 1923). His proposition that a house is a machine for living in polarized discourse around the idea of architecture as engineered space, even though his use of the term “machine” was but a metaphor for complexity and a reflection of the apparent progress of the engineer over that of the architect. Nonetheless, the rhetorical tone of *Vers
une Architecture grounded the machine à habiter – and Le Corbusier himself – as symbols of the cold scientific effects of the engineered modernist aesthetic. While Le Corbusier himself may have merely modeled his design work on the aesthetic productions of the engineer; *Vers une Architecture* is perhaps one of the clearest manifestos we have which presents the quantifications of the engineer in such dizzyingly high regard. Le Corbusier’s “Argument,” the title of the first chapter in the book, clearly presages arguments to come.

Arguments which came to another significant apex at The Massachusetts Institute of Technology, via another machine. Nineteen-seventy’s *The Architecture Machine: Toward a More Human Environment* represents the first major publication of the work of Nicolas Negroponte at MIT, and documents three years of work within what later became The Architecture Machine Group, the predecessor of the Media Lab (Negroponte 1970). Sadly, and somewhat surprisingly, neither in this work nor the subsequent *Soft Architecture Machines*, do Negroponte or his co-authors cite Le Corbusier, so there is no clearly tracable relationship between the machines in question (Negroponte 1975). But Negroponte does make an effort to reinvent the wheel: “The last chapter is my view of the distant future of architecture machines: they won’t help us design; instead, we will live in them” (Negroponte 1975:5). The machines which Negroponte sees us living in are nonetheless quite different than those Le Corbusier does, and this difference carries with it an important trace of the historical shift in technologies in the ensuing 50 years. Le Corbusier’s argument is simply an argument for the metaphor of the machine, an argument that the house is a complex and materially transformative system, and that, like all successfully engineered productions of the inter-war period, should be designed along the lines of the new trades. Negroponte, on the other hand, is arguing for a different kind of machine: a mathematical machine that represents the apotheosis of distributed and environmentally embedded computational power. Where Le Corbusier seeks the engineering of the house as if it were a machine, Negroponte seeks the engineering of the house as a machine. For the goal of the architecture machine is to work “toward a more human environment” explicitly through the computerization of the living environment. One would have little difficulty placing the architecture machine within Foucault’s space of the human sciences, traversing the empirical manifestations of being through mathematics, science, and a form of philosophy which is derived from the computational. Le Corbusier’s machine would be a bit more difficult to place therein, however, as, save for the absurdly pseudo-scientific Modulor, there is little in his work which is actually engineered or scientifically determined. So what changed in the discourse of the machine between the machine à habiter and the architecture machine?

**Scientism in Cambridge**

To borrow the idea of an argument from Friedrich Kittler, the “computer” went from being a job description to the name of an assembly of electronics (Kittler 1999). Until the 1940’s a computer was
the person (more specifically, woman) who performed data entry in the modern computer's predecessors. Afterwards, the person became the box. Of course, a tremendous amount happened in the 50 years between Vers une Architecture and The Architecture Machine, but within the realm of computation — specifically within the history of arguments about the computational argument — we can look to three key developments for inspiration: cybernetics, information theory, and the theory of natural language patterns. Between Norbert Weiner’s 1948 Cybernetics or Control and Communication in the Animal and the Machine, Alan Turing’s 1948 “Intelligent Machinery,” Claude Shannon and Warren Weaver’s 1949 The Mathematical Theory of Communication, and Noam Chomsky’s 1955 Logical Structure of Linguistic Theory, the theory behind modern computation was born (Wiener 1948; Turing 1948; Shannon and Weaver 1949; Chomsky 1975). The historical inter-connections of these works and their contributions to modern computing have been well documented, so I shall not repeat that history here. It is worth pointing out, however, that the nexus of these research trajectories provided, in the context of the burgeoning scientism of the mid-century, a critical platform for computationally-influenced spatial research. Cybernetics provided an objective theory for human-environment as well as human-machine interactions. Information Theory provided a semantic framework for the numerical encoding of non-numeric (spatial, graphic, etc.) information. And structuralist linguistics provided the mechanism, at least in the United States, for the symbolic interpretation of space as well as the theoretical framework for the understanding of space, architecture, and cities as communication systems. The coming together of these frameworks and theories in the body of the computer, furthermore, led spatial thinkers to move in an explicitly computational direction, whether this meant the embrace of the computer as a tool or the conceptual embodiment of the theory of computation in their work, even while perhaps avoiding it in practice.

It is also valuable to note that the key work developed by Weiner, Shannon and Weaver, and Chomsky was all conducted at or around the Massachusetts Institute of Technology in Cambridge, Massachusetts, the same home to a radical transformation in post-modern architectural theory. Weiner obtained his PhD from Harvard in 1912, departed briefly, returned to Cambridge in 1915, and then taught at MIT in 1918 where he remained a professor (although largely in absentia) until his death in 1964. Claude Shannon obtained his PhD from MIT in 1940, and returned to teach there between 1956 and 1978. And Noam Chomsky obtained his PhD from the University of Pennsylvania in 1955, and immediately took up a position at MIT, where he had regularly conducted research for his doctoral work. He remains a professor at MIT. Cambridge Massachusetts, thus, forms an important link between the work surrounding these individuals — including the material development of the post-war generation of computer systems — and a critical body of formative research by highly influential architectural theorists. Three in particular — Kevin Lynch, Christian Norberg-Schultz, and Christopher Alexander — resided at or passed through Harvard or MIT at critical moments in their careers during
this time, and were exposed to, or took a formative role in, the developments of the quantitative revolution in architectural theory.

Alexander is perhaps the most obvious: the 1964 publication of his Harvard dissertation: *Notes on the Synthesis of Form* documents in a profoundly clear form the scientific-rational desires of a mid-century architectural theorist (Alexander 1964). While since criticized by Alexander himself for being overly rationalist, this work demonstrates an innovative fusion of the sciences of language, computation, and anthropology. Alexander’s quantitative observation of a village in India, his development of a syntactical theory for its formal development, and the mathematic model used as a validation of his theory clearly demonstrate the power of what we can usefully call the computational sciences. While Alexander's mathematical model had been toned down by the 1977 publication of *A Pattern Language*, the computational-linguistic aspects have remained in such full force that Alexander’s work has become phenomenally influential in the discipline of object-oriented software development (Alexander, Ishikawa, and Silverstein 1977; Coplien and Schmidt 1995). Far from a cross-disciplinary adoption of Alexander’s work, the software developer’s use of *A Pattern Language* is simply acknowledging the work’s own conceptual origins. *A Pattern Language* has, of course, been highly influential in architectural and urban design, and brings into these domains the same determined rationalization that it has more clearly brought to software developers. Its clarity and reductive simplicity is easily graspable and highly computable, making it quite attractive to those seeking implementation of a highly controllable and quantitatively measurable design strategy.

In a similar vein, although less explicitly tied to the historical development of computation, the work of Kevin Lynch represents a potent and highly influential body of work tied to the linguistic and computational scientism of the mid 20th century. His first publication, „The Form of Cities,” was published in a special issue of Scientific American in 1954 (Lynch 1954). Published immediately upon his return from a year of European travels taken after completing his PhD (MIT, 1947), he outlines in two sections – “Shape” and “The Internal Pattern” – syntactical analyses of form which presage his more explicitly semiotic approach in 1960’s *The Image of the City*. The four publications which appeared between these two trace the evolution from a scientifically-founded body of observational research to the formulation of a theory of urban form based on an explicitly semiotic framework (Lynch 1956, Lynch 1958, Lynch and Rodwin 1958, Lynch 1959). The linguistic operations of *The Image of the City*, both semiotic and syntactical, lead Lynch to an analysis of urban form which is highly discretizing. The fluid body of the city becomes an assemblage of Lynch’s famous elements: paths, edges, nodes, districts, and landmarks. Their assembly results in an image of the city which, while subjectively formed by each individual, becomes for Lynch a quantifiable measure of legibility. The search for a discrete structure, upon which a measurably legible image can be derived, reveals a scientific pursuit influenced by mid-century structural linguistics. Chomsky’s universal structures, at the service of AI researchers, are
tweaked, in Lynch’s work, for an application in architectural theory.

Finally, Christian Norberg-Schultz passed through Cambridge Massachusetts, albeit briefly, on a Fulbright to Harvard from 1952 to 1953, likely to work with Sigfried Giedeon (who was bouncing back and forth from MIT to Harvard at the time). It is unclear if any of his scientific inclinations were formed while in the US at this time, but an attendance at Harvard or MIT at the time would certainly have led to an exposure to the American theories and practices of the emerging architectural sciences. Norberg-Schultz gains his scientism primarily from other routes, most explicitly from Hans Sedlmayr’s scientific tradition of inquiry in the arts and Rudolf Carnap’s logical positivism. In 1965’s *Intentions in Architecture*, he proposes an explicitly linguistic framework for his architectural-historical inquiry, based primarily on the Germanic structuralist semantic philosophies popular at the time (Norberg-Schulz 1965). These philosophies also, somewhat unsurprisingly, underlaid much of the work that other European expatriates already brought to the American efforts to develop the computer. In the case of Norberg-Schultz, what results from this ‘inherent computality’ is a kind of rational phenomenology of architecture, highly scientized à la Lynch. It is not explicitly computational, but proto-computational in the way it merges the science of language, mathematics, and aesthetic theory. To the extent that Norberg-Schultz’ work, especially his later *Genius Loci*, has contributed to the development of an architectural phenomenology, this phenomenology is grounded in the very discretizing, mathematical, and rationalist tendencies which other (primarily French) strains of phenomenology explicitly rejected (Norberg-Schulz 1980; Merleau-Ponty 1964). Nonetheless, the works of Norberg-Schultz have found a very comfortable berth in the annals of architectural theory, again, due in large part to the success of the scientistic project. Quantitative analyses offer profound advantages in the marketplace, where such verifiable processes are sought to best leverage commercial decisions.

Together, Alexander, Lynch, and Norberg-Schultz form a triumvirate of theorists whose arguments for the argument have been vastly influential on our built landscapes and architectural pedagogies. Elsewhere, I have examined these impacts in greater detail (Lonsway 2007). What I want to examine in this essay is the way that these scientistic theories of architecture and urban development have played themselves out in the realm of architectural computation, and what their limits are for advancing its intellectual dimensions. Typically, only Christopher Alexander is mentioned in architectural computing circles, primarily because of his own affiliations with the sub-discipline and the impact of his work in software design. However, we can trace the work of all three of these theorists in contemporary attitudes toward architectural computing, likely as a result of their pedagogical impacts on architecture scholars. Importantly, the appearance of the three key texts produced by these theorists (*Notes on the Synthesis of Form*, *The Image of the City*, and *Intentions in Architecture*) within five years of each other coincides with the first significant publications at the core
of architectural computing. The work of Lionel March, Chuck Eastman, Chris Yessios, and Bill Mitchell, among numerous others, began appearing in the late 1960’s and early 1970’s, attesting to a time when computing instruction and computing availability became practically accessible to those outside of computer science. The dominant linguistic strains in the work of Alexander (syntactics), Lynch (semiotics) and Norberg-Schultz (semantics) are, in particular, highly evident in this first wave of architectural computing research. All of these researchers were able to opportune themselves of the techniques and theories of computation to varying degrees, pursuing variant trajectories, but with a common conceptual frame of reference. It was here that architecture came to computation and computation to architecture.

**Smeared Processors, Symbologies, and Syntactaclysms**

So where has this left architectural computing research today? A common thread of much of the research pursued in the middle of the last century is its theoretical grounding in structural linguistics. Computational researchers on their own became interested in structural theories of language because of its logical cleanliness, which was a nice match for the mathematical logics of the then-current computing machines. And it served to provide tidy models of thought-processing and language skills that seemed adaptable to human-computer interaction, including the then-unproven concept of computer languages, and artificial intelligence. But in maintaining its cleanliness, the obsession with the search for the underlying structures of things neglected to acknowledge that the world is quite messy. When messiness began to enter the picture, it was either swept away through the fiat power of empirical methodology, or dealt with through the rational invention of rule systems which would attempt to account for a statistic majority of messy things. The early work of Alexander and Lynch for example, reveals, respectively, these two approaches. But there were other approaches. In France, around the same time as this work was being first published, there began to appear challenges to this structuralist position in the work of Roland Barthes, Jean-François Lyotard, Michel Foucault, Jacques Derrida, and Gilles Deleuze. Later to be lumped together under the moniker of post-structuralism, this body of work undertook the challenge of uprooting the search for truths, essences, and universal structures. Interestingly, much of it was influenced by the very scientific and linguistic theories and propositions that the structuralists were. But, in contradistinction, it read from these a set of critiques suggesting that differences between things are as important as things themselves, that systems are composed as much by relationships as identities, that quantities are perhaps nothing more than qualities, that centers are defined by margins as much as margins by centers, etc. In other words, that things are messy and aren’t completely identifiable by the quantitative rigor of scientific research.

I’d like to explore some of these critiques, in particular as they relate to the trajectory of arguments for the computational argument, through three examples drawn from contemporary architectural-computational research: the
PlaceLab at MIT’s Media Lab, parametric CAD symbolization, and the Spatial Syntax project headed by Bill Hillier, based in London, UK. My critiques, I want to stress, are not intended to repudiate or reject the many strengths of these research initiatives. It is in fact because of them that my research in this direction is even possible. Rather, my critiques are meant to provocatively explore some of the potential limits of computation as we know it, and in particular, as it has been deployed in architectural design practice. My critical analysis of these will then lead, in conclusion, to a theoretical exploration of what I call a post-structuralist computer for architecture: a provocation to consider the means by which we might expand our inquiry into the applications of computing in architecture beyond the engagements limited by conventional computing techniques.

It was a comment of a former colleague, Ted Krueger, which motivated my study of MIT’s House_n project: a comment that the Media Lab’s approach to the so-called intelligent environment was one of smearing computers around the architecture. The PlaceLab, part of the House_n project led by Kent Larson, has undertaken Negroponte’s original vision of the Architecture Machine Group – to create an architecture machine in which we live – along the lines of Myron Krueger’s (no relationship to Ted) responsive environments (Krueger 1977). There are a number of diverse research projects within this working group, but they appear to be tied together by the structural belief that it is only through the computerization of all things in the world that productive interaction between individual and environment can occur (House_n 2006). Recent imagery from the project’s website shows silverware drawers and flour and sugar canisters wired for interaction, with data collected on object usage by small sensing devices called MITes. The apparent perception of computation as a phenomenon grounded in the processor, albeit a distributed one, establishes a structural association between object and information which delimits the spatial potential for computation. Perhaps unwittingly, the project’s setup recalls the experimental test kitchens of the early 20th century Domestic Engineering movement, established to study and make efficient the workaday routines of the stay-at-home housekeeper. But even these earlier scientists understood the spatial implications of their science better: movements through free space, transitional gestures, apparent wastes of time, were tracked and documented. (Of course, these were seen as inefficient, the ire of the researchers’ efforts.) There appears to be no such interest articulated by the PlaceLab group. The stated goal of the House_n group, rather, that all projects “begin with the idea that the design of places of living and work – and the associated technologies and services – must respond directly to the unique needs, values, and activities of the individual,” itself places a premium on the idea of direct response – arguing for a one-to-one relationship between desire and effect which can be conveniently mapped onto a processor (House_n 2006). This semantic association follows the lineage of the scientistic phenomenology of Norberg-Schultz, where the objective understanding of experience leads to the development of positivist theories of architecture:
totalizing, rational, and universal. Once the experience (of modern architecture or of the flour bin) is mapped to a semantic theory, then the problem of understanding space, and the intentions of the architecture (machine), is solved (or so it is believed).

A second development of contemporary architectural computing which finds its origins in the scientized theoretical aspirations of architecture is the semiotic symbolization of architectural components in professionally-oriented Computer Aided Design software. In order to streamline the design representation process, which often calls on standardized components, commercial CAD tools as early as the 1980’s began to incorporate symbol libraries. (Early conceptual work in this area was conducted at Harvard’s Laboratory of Computer Graphics and Spatial Analysis, founded in 1968 (TheLab 006).) These symbol libraries worked much like the physical templates which predated them, allowing the user to place multiple identical symbols – of windows, doors, appliances, etc. – with ease. With the onset of single-building model efforts in the mid-1990’s, recently refigured as Building Information Management (BIM) tools, the implementation of symbols was transformed into a data-based mode of representing entire buildings. Early tools in this area – the Triforma tools now owned by Bentley are a good example – were conceived around the idea that an entire building’s constructional information could be embedded in architectural models. Each material or building component was represented by 2D and 3D graphic information, as well as specifications, visual material properties, and other data. Depending on the form of output desired, the tool would display the appropriate form of representation. While conceptually elegant, the tools required the symbolization of every aspect of a building intended to partake in the system. The drive toward advancing and streamlining this process, from the pattern-based recognition of forms for data retrieval to the parametricization of the representations of standard commercial products, has not stopped, nor has the necessary support of the theoretical foundations of this line of computational development. The strict semiotic associations that are currently established between design element and symbolic representation, based on a need to form an identity at some level of representation for a data element (the pixel, the line, the symbol, etc.), have fallen into the structuralist frame of software designers. This amounts to nothing more than Alexander’s patterns, implemented as software objects, for a model by which a continuous drawing is parsed into architectural graphemes for parametric processing. The drawing is no longer allowed to be a continuous drawing, where floor might be wall, or a window might be a door. Even the quickest of surveys of work being done in the field of semantic representations (from semantic webs to the semantic symbolizations which underlie the technology in question) reveals that there is much confusion about how to handle the ‘problems’ of ambiguity, multiple identification, indirect association, the subjectivity of data, informational alterity, etc. Of course, it is the structuralist theoretical position itself that constructs these as problems – rather than opportunities – in the first place.
Finally, the structuralism of computation can be seen to underlie the theoretical and applied work of Bill Hillier and his team at the Spatial Syntax group based in the University College London in the UK. Borrowing explicitly from the work of Alexander in understanding spatial problems as syntactical, and from Lynch in advancing the use of semiotic notational systems for spatially signifying elements, the Spatial Syntax group has developed under Hillier’s guidance a set of techniques and computational tools for representing the syntactical assembly of spaces, from cities to individual buildings. His 1984 book, *The Social Logic of Space* written with Julienne Hanson, first proposed a system of documenting discretely signifying socio-spatial elements to better understand these rules of assembly. Hillier and Hanson sought patterns which emerged from the collective documentation of these individual moments as the groundwork for a theory of space based on social behaviors. Since this first publication, the work of this group has advanced the notion of the ‘configuration’ as the most salient concept in understanding social spaces. It is the essentialized structure of these configurations which the work of the Spatial Syntax group seeks to reveal.

The techniques of ‘configurational analysis’ – of which the various ‘space syntax’ techniques are exemplars – that have been built from this idea have made it possible to bring the elusive ‘pattern aspect’ of things in architecture and urban design into the light of day, and to give quantitative expression to the age-old idea that it is ‘how things are put together’ that matters. (Hillier 1996:1)

Which brings us back to the machine. Hillier’s *Space is the Machine* (1996) gets its title from a comment of one of the group’s programmers: “You haven’t understood. The building isn’t the machine. Space is the machine” (Hillier 1996:vii). Thankfully, Hillier describes the relationship of his concept to Le Corbusier’s *machine à habiter* quite succinctly. He categorizes Le Corbusier’s engagement of the idea of the machine as a metaphor rather than a paradigm, and frees Le Corbusier from the responsibility of the latter, which he believes gave modernism its bad name. The paradigm of the machine, according to Hillier, gained its credence through the “social and political theorizing that increasingly became the intellectual context of the practice of architecture as architecture moved towards a social engineering process” (Hillier 1996:378). And this, he dismisses as problematic “because it had no concept of spatial configuration” (Hillier 1996:379). While Hillier’s critique potentially opens up a discursive space similar to that I am seeking to open up here, he quickly closes that potential by returning to a defense of the deterministic machine paradigm. As his programmer states, however, it is space that is so determining – in particular, the configurational aspects thereof – not the building. Hillier’s quantitative analysis of these configurations, via machines (perhaps architecture machines), is an attempt to reveal the determinist space machine such that an objective structural theory of architecture can be advanced.

So we remain haunted by the argument for the structuralist machine – the machine that we are to live in as metaphor, as computational object, or as configuration. Each of these examples, the
semantics of the PlaceLab’s architectural interface, the parametric semiotics of the CAD symbol, and the syntactical notations of the Space Syntax group, represents the paradigmatic nature of the argument for the argument. For it is the argument – the computational variable – that defines these ‘machines’: the argument representing the amount of times a flour container was opened, and what this might mean; the argument representing the catalog item number and parametric modifications of the pre-conceived architectural element; and the argument representing the ‘connectivity index’ of an urban configuration. Their authors appear to believe in a quantifiable objective reality of their subjects, and are working to produce their own computational machines to more effectively or accurately understand what this reality is.

**différance engines and nomadological routers**

But what if objective truth is unattainable, that the notion of machinic determinism is thrown out the window, or that semantic, semiotic, or syntactical identities are far from universal? What if the post-structuralists are to be believed? My own argument on the argument is that it is valuable, if not entirely necessary, to believe them if we are to make intellectual progress in architectural computation. For the empirical quest for absolute and universal structures will get us only so far. I would hardly want to throw out such pursuits, but they must be complemented by radically variant approaches which seek to advance our computational efforts well outside of their scientistic boundaries. I would like to introduce two concepts, critical to the post-structuralist debates that will frame hypothetical explorations into what kind of machine might help us with this intellectual challenge: différance and nomadology. I owe much of the inspiration for seeking such a practical absurdity – a post-structuralist computer – from a book by Paul Cilliers: *Complexity and Postmodernism* (Cilliers, 1998). In it, Cilliers attempts to find the most deconstructive computational structure he can. His summons of Derrida within the more scientific side of the complexity-theory discourse is quite commendable, as is his indictment of the speech-act theory of John Searle (echoing Derrida himself in Limited, Inc.), for its unfortunate contributions to AI research. And while I may disagree in the end with his earnest defense of the neural network as différance-in-action, Cilliers has set out an important inquiry valuable for its contributions to both computational and complexity theories. It was this book that set off the following provocations.

Jacques Derrida elucidates the idea of différance most succinctly in a lecture of the same name originally delivered in 1968, as a playful exploration of the classic Saussurean signifier/signified dyad (Derrida 1973). While semiotics is at the core of his argument, in many ways it is merely the foil, as différance is in fact at the core of Derrida’s philosophy of life itself. The concept of a nomadology was introduced by Gilles Deleuze in *A Thousand Plateaus: Capitalism and Schizophrenia* as a philosophical framework to understand the inter-relationship between language, structure, and law on one hand, and thought, self-organization, and custom on the other (Deleuze, Guattari 1987). Together, these set out to achieve similar
ends: to advance a philosophy which contemplates head-on with complexity, multiplicity, ill-definition, ambiguity, alterity, and other such traditionally unphilosophized areas of life.

Derrida’s diffusion contains three radical concepts which I would like to play out in the image of a diffusion engine: a computationally radical redux of Charles Babbage’s Difference Engine. These are:

1. that the stability of the relationship between a word and a concept is challenged by the fact that the two things are separated by both difference and deferral – a spatio-temporal condition of separation;
2. that the perception of this instability varies across modes of linguistic representation (in Derrida’s original case, speech and writing); and
3. that this instability is a process, always ongoing, rather than a state or condition of a mode of representation.

Babbage’s Difference Engine, conceptually outlined in 1822 and partially built in fits and starts during his lifetime, is considered one of the first proto-computers, a machine designed to automatically compute and print out mathematical tables using the mathematical elegance of Newton’s law of differences. As a tool based on the computation of differences, it requires the storage and manipulation of discrete values to be subtracted from one another. Its strength lies in the way the engine is ‘programmed,’ and how the results are interpreted – the machine itself is pretty stupid. What if we were to consider the design, rather, of a diffusion engine? Let’s say that this, too, would be a computational machine (perhaps even mechanical), whose manipulation would result in the outputting of numerical tables. But rather than operate by storing values and computing differences, it would have to operate by engaging diffusion. The question first has to be asked: how can a machine engage diffusion? Or, more accurately, can a machine engage diffusion? Realizable or not, it is a provocative thought – rather than look for a specific mathematical relationship between stored thought – rather than look for a specific mathematical relationship between stored values, this machine would seek, rather, the instability of the relationship between values. Results would be as meaningful as a result of differences, but they would not necessarily subscribe to any apparent logic or rule.

And it would produce these results continuously, with variation, and perhaps even differently in different modes of communication (spoken word, printed text, sound, and image). It would not perform regular manipulations, but irregular, playful ones. Serendipity would become as important to its user as verification. Sure, this engine could not likely be used to generate a reproducible quantitative result like we can with a contemporary computer, but it would introduce into the realm of computation a discursive practice, where our engagement with the machine is one of constructive play and mutually-complementary learning. A practice of de-contextualization from the scientific framework that dulls down, even deadens, computation as we currently know it.

A nomadology was first introduced by Gilles Deleuze in collaboration with Félix Guattari in 1980’s *Capitalism and Schizophrenia*. In *A Thousand Plateaux*, translated to English in 1987 as *A Thousand Plateaux*, as a philosophical framework for yet another machine: their so-called war machine. As
with différance, a short paragraph can hardly do the idea justice, but a brief schematic will do in the context of this essay. Denotatively, a nomadology – a logos of the nomos, or the laws, logics, or structures of customs or practices – is itself a bit of a provocation. A critique of the structures of logocentrism, in favor of the more fluid space of customs, practices, tactics, it nonetheless presents itself as a structure. I like to pair it with what I consider to be its complement – logonomadism, or the practice of structure – to better understand its quasi-dialectical significance. Between the space of law and the space of custom, the space of the resident and the space of the nomad, the smoothness of the desert and the striations of the map, lie the complex power structures of the contemporary nation-state: the war machine. Less directed a critique than différance, the articulation of nomadology is meant as a philosophy of the state, of the systems of contemporary power structures. Inserting the nomadic in the logocentric is, nonetheless, a significant political statement. It opens up a space for critique – a space for différance to find itself among the most logocentric sciences, the most dictatorial tyrants, the most opaque democracies. And the most reductive machines. Including architectural ones. It argues for a different kind of argument: the actual variability of the variable, the indeterminism of a mathesis. So let us consider a Deleuzian machine relevant to a contemporary practice of design: a nomadological router. Let us, in fact, return to a latent argument for the architecture machine.

Perhaps my earlier critique of the reductive scientism of Negroponte’s architecture machine was unfair. Soft Architecture Machines from 1975 is peppered with images of naked women wearing mobile computers, bouncy inflatables, and gerbils running around robotically-assembling mazes. There was no doubt more than a smattering of play in the Architecture Machine Group before the media lab became an American Institution, and this playfulness is perhaps the group’s most redeeming quality. For it is in their ludic productions that there is a real potential to see an escape from the deterministic desires of their general body of research. But these two early books, as well as the architectural legacy of the group (House-n, the PlaceLab, etc.) read like objective treatises on the computability of the essential structures of space, space-making, and spatial perception. Even the naked women and the bouncy spaces – the “soft” stuff of the soft architecture machine (well deserving of a serious gender critique) – serve as examples of direct interaction between human and machine. The building reacts when I push on it, my skin tingles when I approach another naked device-wearing person. The semantic unambiguity of these definitions of “responsive” and “soft” suggest to me that even their apparent play had been scientized. But then there is the computational habitrail (Figure 1). Called SEEK, the purpose of this machine was to “show how a machine handled a mismatch between its model of the world and the real world -- in this case five hundred two-inch metal-plated cubes. The mismatch was created by a colony of gerbils whose activity “constantly disturbed the strictly rectilinear arrangement called for by the machine’s model” (Negroponte 1975:47). Until Negroponte claims that “one
substantially dislocated [block] would be placed (straight, of course) in the new position, on the assumption that the gerbils wanted it there,” there seemed to be some nomadological potential here. Nonetheless, full of cybernetic whimsy, and actually quite profound as an alternative model of computation, SEEK could be productively seen as a prototype for our nomadological router. De-marginalize the indeterminacy (the rodents, in SEEK’s case) so that it isn’t exterior to the computational model, de-rationalize the objective function of the machine such that its model is not excluded from the “real world,” and celebrate rather than correct the aleatory “mismatch,” and we have it: a post-structural architecture machine. Add a router bit, and our machine interprets a code to form a
model of its cutting path, but it also models and traces the dynamics of what is being cut, finds resistances (mismatches) and plays with them, proposes alternatives in real-time, and perhaps most importantly leaves the job unfinished, at least where it finds itself unproductive. The cybernetics of SEEK then might become critically nomadological, suggesting a model for computation which once again breaks the conventional engagements with the human-computer interface.

What, really, can my provocations accomplish? In this essay, I propose nothing more than to have them register a voice of criticality. Fundamentally frustrated by the cybernetic relationship between myself and the computer, I continually seek models of computation which can escape from its current mathesis. This is in part a reaction to the dominant models of computation deployed in architectural practices today as well as to the unfortunate scientism of design practices which computation both informs and lends itself to all too well. This essay is an attempt to ground a historical and philosophical position which challenges these dominant models, upon which alternative models may be constructed. I’m not sure how a post-structuralist computer might actually be built; perhaps it is a software problem, perhaps a hardware one, perhaps both or neither. Perhaps it is simply a perceptual problem. There are, among the computer sciences, arts and experimental architectural practices, a number of experiments which in one way or another seek to address this; my desires are hardly unique. The most critical question may not, therefore, be how to achieve this model, but how to make its often apparently undermining tactics compete with the prevailing strategies of architecture computing. One thing we might be certain about: it won’t be released by AutoDesk.

References


Merleau-Ponty, Maurice. 1964. The primacy of perception, and other essays on phenomenological psychology, the philosophy of art, history, and politics, Northwestern University studies in phenomenology & existential philosophy. Evanston, Ill.: Northwestern University Press.


