

archi|DOCT

*The e-journal for the
dissemination of doctoral
research in architecture.*

Supported by the ENHSA Network | *Fueled by the* ENHSA Observatory

4

February **2015**

www.enhsa.net/archidoct

ISSN 2309-0103

MATTER

Structures vs Textures: challenging the hegemony of geometrical consistency

Manolis Paterakis // Faculty of Architecture // National Technical University of Athens

Abstract

In a context where standard methodologies of Information Technology – such as algorithmic analysis, data management and visualization – have infiltrated into common practices of Architectural Design, there is a substantial claim that these two faculties of knowledge share a field for a common theoretical approach. Along this direction, this paper attempts a consideration of systems analysis and design as a fundamental Architecture discipline through established computational concepts, such as the object-vs-process duality. Thus, in view of the transformations induced on architectonic thinking by the integration of algorithmic and computational methodologies, it is suggested that our view of Architectural systems as arrangements of entities and objects must evolve, encompassing temporal qualities such as duration and transformation. Primarily, this research starts off from the very notion of Structure as a conception of consistency, its close association to the legacy of diagrammatic reasoning and its formalization as Object-oriented modeling in the computational domain. Structural perception of systems, as the backbone principle of architectonic analysis, is an inherent aspect of geometric inference and spatial intuition. However, as Henri Bergson has suggested, it fails to grasp Time as an affective quality instead of a differential quantity, and restricts perception of dynamic, evolving systems, such genetic and biomimetic formations, to static apprehensions frozen at arbitrary states. Thus, another perception of consistency is suggested, opposite to that of Structure, founded on the notions of Time and Duration as primitive intuitions; the concept of Texture.

Keywords

Entity; Object-oriented; Structure; Geometry; Texture; Duration; Computation

1. Mario Carpo focuses the humanists' invention of intellectual authorship on the Albertian Paradigm (Carpo 2011: p. 24-26, 71-79).

2. As James Franklin suggests, for the most part of 20th century the Logical Positivism intelligentsia ruled out the image as an unscientific instrument intended for the intellectual marginals: diagrammatic reasoning for the engineers and mental images for the Freudians (Franklin 2000: p.84).

3. For the purposes of this analysis and in favor of brevity, terms like 'Entity', 'Object' and 'Thing' will be considered essentially as synonyms, although there has been a lot of philosophical debate regarding their semantic differences and qualities.

Ever since Architecture was detached from its artisanal grassroots and adopted a more authorial creative role¹, the notion of *Structure* has become one of the most fundamental topics of architectural thinking. And although structural design may be considered a synonym for engineering and load distribution analysis, the modern concept of *Structure* as the *architectural perception of consistency in a system of discrete units* has been introduced into many faculties of knowledge as a transdisciplinary principle. The most prominent example of all, the domain of Information Technology, has adopted the term *architecture* to describe the process of analysis and design of a system; object-oriented modeling principles regard systems and networks as diagrammatic aggregations of discrete objects and components; *spatial arrangements between logical entities*. This evident focus on spatiality draws attention to the dominance of Geometry, both in the Eulerian and in the Euclidean sense, as the fundamental discipline of inference under which a system is being analyzed and designed. For the domain of a strict formal science, such as Computer Science and Mathematics of Computation, this is a relatively newfound practice; formal sciences have generally preferred *a view of inference as the manipulation of symbols according to formal rules*² and have traditionally rejected geometrical inference as 'subjective' and 'unscientific'. However, for the domain of Architecture, the concept of *Structure* that describes geometrical consistency through diagrammatic reasoning has been the backbone principle of architectural design as an intellectual practice since the Age of Euclides. This is further indicated by the extensive use of sketches, diagrams and blueprints, that regardless of their medium of transmission – either analog or digital – establish this tight association of architectural design with geometrical inference. Apparently, for many domains of knowledge that deal with the design of systems, Architecture has become the archetypal *science of Structure*. However, the Age of Computation has introduced new instruments of creation. *Algorithmic* and computational technologies of design and production claim an important place in the architect's toolset. Their inherent vocabularies of uninterpreted symbols, filled with programming code and mathematical formulas, leave little place for geometrical intuition. Furthermore, visual manifestations of computationally designed systems often seem inadequate to convey their constantly evolving nature; they restrict representation to still snapshots of their state. In short, the ubiquitous presence of the Algorithm, evident in almost all contemporary creative practices that deal with design and synthesis, makes it clear that the Age of Computation requires all faculties of knowledge to reformulate even their most fundamental principles into computable processes. Concerning the field of Architecture, we are left with a question; does the infusion of algorithmic logic in design mean that the dominance of diagrammatic reasoning is being undermined in favor of a more symbolic mode of inference? Furthermore, is the architectural concept of *Structure* as the notion of geometrical consistency between entities, being threatened by more syntactical, formalistic and less diagrammatic notions of consistency?

Entity vs Process: an architectonic history of information systems

The reign of Cartesian rationalism established a view of the universe as a system of primitive entities, as a complex apparatus made of machines down to a level of final and discrete elements. As a direct intellectual product of newtonian mechanistic thought, the concept of the Entity³ has been the dominant paradigm of analysis ever since; a self-contained, discrete, individual unit of information described from a finite set of properties and qualities. Thus, multiple faculties of knowledge consider systems as aggregations of entities, as artic-



4. For a brief summary of object-oriented principles see (Weisfield 2009: p.5-12).

5. For an introduction see the related wikipedia article: http://en.wikipedia.org/wiki/Actor-network_theory (last accessed: Jan 17, 2015). For a more detailed view see (Latour 2005).

6. According to Henri Bergson, "the systems science works with are, in fact, in an instantaneous present that is always being renewed; such systems are never in that real, concrete duration in which the past remains bound up with the present. When the mathematician calculates the future state of a system at the end of a time t , there is nothing to prevent him from supposing that the universe vanish-

lated compositions of discrete *Objects*. In the field of Architecture, the consideration of buildings and structures as modular compositions of either primary or industrially prefabricated components has established architectural design as the archetypal example of the entity-oriented design paradigm. Under these terms, when the concept of the entity was introduced into the domain of Computer Science, during the late sixties with the emergence of object-oriented programming principles⁴, it immediately shifted the interest of software development towards diagrammatic reasoning practices (such as blue-printing, prototyping, and pattern-based design) and rendered the term 'Architecture' as the primordial concept that characterized that new perspective on considering information systems. Object structures and genealogies have been the core topics of interest ever since, in several fields of Information Technology such as software design patterns, structured data-storage systems and agent-based simulation applications. Moreover, recent approaches have attempted to introduce the concepts of object-oriented modeling in domains of sociology and psychology, with the Actor-Network Theory⁵ being the most prominent example of this adaptation. In short, developing formations of both Matter and Data along this entity-oriented tradition has always been a pursuit for a coherent and consistent Structure; a logical and efficient arrangement of the relations between entities and their components, translated into the spatial vocabulary of diagrams and graphs. In this context, architectural problems are abstracted into issues about entity *nature, classification and composition*.

On the other hand, the recent fusion between computational technologies and design practices has introduced systems and formations that remain on a constant evolutionary progress; for example, biomimetic algorithms and particle swarms require the progression of time in order to *grow*, to evolve into a state of equilibrium. Hence, there is a shift of focus in design analysis from the forms and formations being developed towards the actual transformations they are undergoing in order to evolve. The static, mechanistic nature of entity-oriented structuring however, appears to leave out not only the prospect of a qualitative change of the system, but altogether the concept of Time considered as an affective quality, rather than a differential quantity⁶. In essence, diagrammatic inference constructs a static image of the system, where the only possible representation of evolution or transformation is through the Newtonian mechanics of motion. Thus, computational design has drawn attention to the concept of the *Process*, a term that has been extensively investigated from a computational as well as a philosophical perspective.

A 'Process' describes the execution of an algorithm, a procedure that takes arbitrary data as input and subjects them under multiple transformations to produce new information. Not all algorithms signify processes though; their purpose is usually associated with some ongoing, repeating execution of instructions that receives and produces a continuous flow of information. In that sense, evolution in a system of processes can be seen as a series of convoluted parallel flows, a continuous surface of interweaving *threads of execution* that fold and unfold progressively; in essence, a system of processes can be regarded as *deleuzian machine of a continuous flux*⁷ in which the primary perspective of analysis is Time as concrete Duration. To further grasp this temporal aspect of the Process, it is necessary to consider it in a sequential manner; in fact, the essence of computation as an abstraction of evolution and transformation, lies in the concept of *Ordinality*, the principle of order and succession in which the elements of a flow, or a *generation* are laid out.

It was Georg Cantor who first, in the context of formulating his set theory, divided the nature of the natural numbers into two aspects; *cardinals and ordinals*.⁸ These concepts where later applied to other faculties of knowledge, such as linguistics and computation.

es from this moment till that, and suddenly reappears. It is the t-th moment only that counts—and that will be a mere instant. What will flow on in the interval—that is to say, real time—does not count, and cannot enter into the calculation.” (Bergson 1911, p.22).

7. See (Deleuze and Guattari 2004).

8. For a simplified version on Cantor’s distinction see (Davis 2001: p.120-121).

9. For this reason, sometimes the first formulation of Cantor’s Set Theory is called “Simple” or “Naive”. For a complete review of Russell’s paradox, see the relevant wikipedia article: http://en.wikipedia.org/wiki/Russells_paradox (last accessed: Oct. 12th 2014).

10. Douglas Hofstadter includes a detailed review on recursive structures and processes. See (Hofstadter 1999: p.127-152).

The notion of Cardinality applies to numbers that describe sizes and populations; the multitude of a set as the number of its elements. In this sense, cardinality is a pure entity-oriented concept as it induces issues about nature, classification and composition; in order to specify cardinality for a set of elements, the elements themselves must be discrete, individual units; to be included or excluded from the set, each element’s nature must be resolved according to the set precondition. On the other hand, the notion of Ordinality applies to numbers that describe succession; the position of each element in a series. Symmetrically, ordinality implies a focus on evolution; in the pursuit of deciding on the order of things in a sequence, we must reflect on the law that incites which elements precede and which elements follow, we must deduce the Rule of the sequence. And in reverse, having the initial order of things or knowing the Rules that conduct an evolutionary system, the principle of ordinality enables us to compute, to construct, to produce the rest of the sequence up to a virtual infinity.

One of the most signifying differences between entity-oriented and process-oriented modeling, is the legitimization of self-referentiality; *enabling definitions of a thing in terms of itself*. Self-references have always been a constant problematic issue in several domains of epistemology, such as mathematics, computer science, or even philosophy and architecture theory. From an entity-oriented perspective, a self-referential definition of an object undermines its discreteness and induces paradoxes and inconsistencies in an object-oriented system. In traditional predicate logic, as well as in Axiomatic systems (such as Euclidean Geometry) self-reference is a synonym to paradox, inconsistency and falsity. The most characteristic self-reference paradox example, the Russel paradox first observed on Cantor’s set theory⁹, emerges when trying to classify sets that contain themselves as elements. Simply put, a discrete object that contains itself cannot be modeled or visualized by modes of reasoning based on spatial intuition, such as Geometry.

On the other hand, self-referentiality in the process-oriented model is entirely inherent as it provides powerful mechanisms of Recursion¹⁰. The extensive application of recursive functions, that is functions that contain themselves inside the function body, is evident in all faculties of knowledge that deal with pattern recognition or its reverse counterpart; *form generation*. Most examples of form generational algorithms, currently widespread in computer graphics, visual design and architecture, are either recursive or employ some kind of self-referential iteration. Since the lambda-calculus system, formulated by Alonzo Church, and Noam Chomsky’s work on syntactic structures and generative grammars, recursion has always been linked to generative systems that produce new structures from a finite set of elements. This close association of recursion with generation can be decoded if we consider a recursive process as a transformational mechanism; *the current state of the system is expressed as a modification of its previous one*. In this sense, a system evolved through multiple iterations of a recursive process contains (or in the deleuzian sense, *enfolds*) all its predecessor generations, thus *immanently accumulating duration; in essence, a recursive system transforms qualitatively rather than quantitatively*.

Structure vs Texture: a computational approach on architectural systems

Thus, a process is a thread of continuous progression. Images of branching vegetative growths, or interweaving patterns of textile threading used more frequently to visualize systems of processes are not mere metaphors employed incidentally; it is well established



11. From translator's notes in Menabrea, L.F., Sketch of the Analytical Engine Invented by Charles Babbage, Translated by Ada Augusta, Countess of Lovelace, in *Scientific Memoirs*, Vol 3 (1842). Available from <http://www.fourmilab.ch/babbage/sketch.html> (last accessed: Jan 17, 2015).

12. For an in-depth view of Ocularcentrism and the hegemony of vision in western tradition, see (Pallasmaa 2005: p.15-19).

that a significant impact on the evolution of modern computing was generated by the weaving machines of the industrial revolution. During the early 19th century, Joseph Marie Jacquard, a weaver and a merchant, invented a mechanical weaving machine that simplified the process of producing textiles, using a set of punched cards to specify the textile patterns. It was Jacquard's programmable machine that later influenced mathematician and philosopher Charles Babbage, the grandfather of computing machines, to design the Analytical Engine, the first programmable mechanical calculator. At a later time, Ada Lovelace, Lord Byron's daughter and Babbage's most beloved student, proposed that the Analytical Engine would *weave algebraic patterns, just as the Jacquard loom weaves flowers and leaves*¹¹. However, both Babbage and Lovelace died long before they could see their designs implemented.

The conceptual archetype of modern computing, Alan Turing's *Tape Machine*, originally conceived to address Hilbert's *Entscheidungsproblem* (problem of decision), can be regarded in a similar manner in view of Ada Lovelace's analogy; it manipulates a thread of information symbols. Instead of folding and twisting transformations on a flow of fibrous matter, it performs replacement and expansion transformations on a stream of arbitrary atomic symbols: digits, letters and images; *a loom of data*. Under these terms, contrary to the object-oriented perception of digital culture as an aggregation of networks, process-oriented thinking shifts this perception towards an assemblage of interwoven informational threads, *a continuous Deleuzian body textile*. Instead of the Cartesian analytic paradigm of a continuous dissection into primitive entities, nodes or particles, another paradigm is suggested: that of *ordinality, serialization and Duration*. Transitioning from one paradigm of analysis to the other, induces fundamental conversions; objects and entities become procedures and processes, questions about status and being, turn into questions about *transformation and becoming; the recursive system is no longer in pursuit for a consistent Structure; it seeks to create an emergent Texture*.

Biologists use the word "hypha" from the greek word υφή ("texture"), to describe the branching structure of fungi and bacteria; a visual description of vegetative growth. It is extensively used in several creative fields where digital computational methods are applied to produce forms influenced by nature and biology. In greek, the word υφή is used both in the occasions of vegetative growth, equivalently to the word "hypha", as well as in expressing the feeling of a surface, the taste of a food or drink, or the ambient sensation produced while listening to a musical piece, occurrences where the word "texture" is applied in English. The word υφή is typically examined in contrast to the word δομή ("structure"), since the two concepts occasionally have overlapping interpretations in the Greek language. While the latter is usually applied to a logical or spatial perception of relations between entities, tightly associated with the sense of vision and consistent with the ocularcentric tradition¹² of western culture, the former, originating from the verb υφαίνω which actually means to weave, is usually linked to experiences generated by other senses, such as touch, taste and smell, and denotes the result of blending different elements into a single emergent result. The English equivalent word Texture, originating from the same semantic root of textile manufacturing and weaving, is used in similar multi-sensory contexts to describe *a complex, yet indivisible experience* accumulated progressively through the primitive temporal intuition of continuity and sequentiality.

In order for an experience to emerge as a Texture, a concrete temporal duration is required. In contrast to the instantaneous, concurrent and uniform visual stimuli of a Structure, the tactile stimuli of touch or the chemical stimuli of taste and smell are transmitted sequentially; the hand must perform a calm, uninterrupted sliding motion on a

12. The temporal aspects of the human gustatory system are a domain of constant research. For a brief summary with extended references, see (Katz 2005).

surface in order to transmit as much sequential information to the brain for the feeling of the surface texture to emerge. The receptor neurons on the tongue and mouth send continuous, consecutive stimuli to produce a variadic and evolving sensation of taste that transforms perpetually from the first foretaste into a lingering (and sometimes totally different) feeling of aftertaste¹². *A texture-experience is therefore a process, not an event*; in contrast to structures being perceived as spatial arrangements of discrete elements, textures rely primarily on the temporal intuition of duration and are perceived as indivisible emergent experiences generated by interweaving threads of matter, sounds, chemical stimuli or digital data subjected to continuous transformations. The role of the main processing unit (e.g. the human brain) is to weave these flows into a single experience; it becomes a *loom of sensory threads*.

Along these lines, object formations in the domain of Textural perception are volatile and mutable; the static structures needed to support the computationally evolving form diminish into small ephemeral elements, symbolic atoms. In Architecture, large building structures reduce their actual structural designs into atomic modules, junctures, elementary cells subjected to the total control of the computational form; a series of *semantically arbitrary monads* upholding the continuity of the emergent Texture. Along the same process-oriented perspective, Big-Data structures in Information Technology are implemented essentially as vast, flat data pools of *unstructured information atoms*, ready to be fused into domain-specific transformational processing and ad-hoc object-ontologies. Under these circumstances, process-oriented modeling undermines the dominance of spatial intuition or sense of vision as the primary guides of analysis, and focuses on the temporal aspect of the systems designed, their rules of evolution and their mode of becoming. Visual representations of such systems, such as diagrams and images, capture only still snapshots; transient depictions of their state in arbitrary points of their evolution history.

Epilogue: challenging the hegemony of geometrical consistency

This ubiquitous presence of the process-oriented modeling paradigm highlights another characteristic of the Computation Age; the inherent dynamics of the Process cannot be drafted, illustrated or sketched. Despite Bergson's critique on early 20th century mathematics about their inadequacy to grasp time and duration as concrete qualities instead of differential quantities, it is suggested that up to a certain extent, modern computational mathematics, theory of recursion and functional programming offer the potential of formalizing the notion of time as quality, both conceptually and notationally. The importance and power of this notational formalism however, as far as architects and designers are concerned, lies exactly where diagrammatic reasoning falls short; in expressing infinity, iteration and self-referentiality without compromising the consistency of the representation. Those who have worked with software packages such as Grasshopper, which involve some kind of visual programming, a replacement for writing actual code, realize that the inability of the diagrammatic representation of a process to express iteration or recursion is not because of some limitation or inadequacy of the software implementation, but, as described earlier, due to the nature of object-oriented, diagrammatic reasoning to reject self-referentiality as a foreign concept.

Naturally, this fact poses a notable competitor opposite the dominance of diagrammatic reasoning and geometric inference as the prime architectural instruments. Not in the sense that their representational function is being undermined or that they are being replaced by programming code; the tight bonds between architectural thinking and spatial



intuition are far too primitive to be threatened. What in this case is being challenged is the *hegemony of Geometry in the justification of the architectural form*. The main semantic operation of the diagram is identificatory, it associates a visual geometric object with an (abstract) discrete entity or concept; a distinct shape installed precisely because of its logical identity to the idea it represents. Hence, the consistency of the diagrammatic installation is evaluated through the geometric associations of identity between the entities that compose it. On the contrary, the existence of spatial relations in a computationally produced formation is just a secondary derivative layer of interpretation; there is no point trying to distinguish a strict underlying Structure there. The formation is not justified geometrically but syllogistically; *we can conceive the logic and consistency of a recursive system because we can distinguish the Rule that produced it*. We can conceptualize the generative process that produces a variadic geometry through the rules that conduct its evolution, without the need to justify it diagrammatically through its ephemeral manifestations. In the end, we can perceive it as an *indivisible emergent Texture*.

“No doubt, for greater strictness, all considerations of motion may be eliminated from mathematical processes; but the introduction of motion into the genesis of figures is nevertheless the origin of modern mathematics. We believe that if biology could ever get as close to its object as mathematics does to its own, it would become, to the physics and chemistry of organized bodies, what the mathematics of the moderns has proved to be in relation to ancient geometry. The wholly superficial displacements of masses and molecules studied in physics and chemistry would become, by relation to that inner vital movement (which is transformation and not translation) what the position of a moving object is to the movement of that object in space. [...] Such a science would be a mechanics of transformation, of which our mechanics of translation would become a particular case, a simplification, a projection on the plane of pure quantity. [...] But such an integration can be no more than dreamed of; we do not pretend that the dream will ever be realized. We are only trying, by carrying a certain comparison as far as possible, to show up to what point our theory goes along with pure mechanism, and where they part company.”

(Bergson, 1911: p.32-33)

Acknowledgment

This text is a revised edition of an original paper titled “Entity-oriented vs process-oriented modeling in Architectural thinking: the rise of the functional algorithm” presented in the “What’s the matter: Materiality and Materialism at the Age of Computation” International Conference in Barcelona, Spain organized by the ENHSA on September 2014. Both texts are part of a work-in-progress doctoral dissertation under the supervision of Prof. Dimitris Papalexopoulos at the School of Architecture of N.T.U. of Athens.

References

- Bergson H. (1911) *Creative Evolution*, Translated by A. Mitchell, New York: Henry Hold & Co. Full text available online from Project Gutenberg, <http://www.gutenberg.org> (last accessed Jan 17, 2015).
- Bitsakis E. (2003) [Being and Becoming]. Athens: Daedalos 6th ed. (In Greek)
- Carmo M. (2011) *The alphabet and the algorithm*. MIT Press, Cambridge MA.
- Davis M. (2001) *Engines of Logic: Mathematicians and the Origin of the Computer*.



New York: W.W. Norton & Co.

Deleuze G., Guattari F. (2004) *Capitalism and Schizophrenia Vol. I: Anti-Oedipus*. Trans. Robert Hurley, Mark Seem and Helen R. Lane. New York: Continuum.

Eden, A. H. (2002) A theory of object-oriented design. *Information Systems Frontier* 4:4, p. 379-391.

Franlin J. (2000) Diagrammatic reasoning and modelling in the imagination: the secret weapons of the Scientific Revolution. In: Freeland G., Coronos A. (eds.) *1543 and All That: Image and Word, Change and Continuity in the Proto-Scientific Revolution*, Dordrecht: Kluwer, p.53–115.

Hofstadter D. R. (1999) *Godel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books Inc.

Katz D. B. (2005) The Many Flavors of Temporal Coding in Gustatory Cortex, *Chemical Sciences* vol.30, No. suppl. 1, Oxford University Press. Available online from: http://chemse.oxfordjournals.org/content/30/suppl_1.toc (last accessed: Jan 18, 2015).

Latour B. (2007) *Reassembling the Social: An Introduction to Actor-Network-Theory*. OUP Oxford.

Leibniz G. W. (2006) [The Monadology]. Ed. Anapolitanos D., Trans. Lazarides S., Athens: Ekkremes. (In Greek)

Pallasmaa J. (2005) *The Eyes of the Skin: Architecture and the Senses*. Pref. S. Holl, New York: John Wiley & Sons Ltd.

Weisfield M. (2009) *The object-oriented thought process*. Boston: Addison-Wesley, 3rd ed.