The Computer-Aided Rough Patterns of Christopher Alexander

Diana Cristobal Olave

Nexus Network Journal
Architecture and Mathematics

ISSN 1590-5896
Volume 23
Number 1

Nexus Netw J (2021) 23:135-150
DOI 10.1007/s00004-020-00515-y

Springer

Nexus Network Journal (2021) 23:135–150
https://doi.org/10.1007/s00004-020-00515-y

NEXUS NETWORK JOURNAL
Architecture and Mathematics

RESEARCH

The Computer-Aided Rough Patterns of Christopher Alexander

Diana Cristobal Olave[1]

Published online: 4 August 2020

Abstract
This paper analyzes Christopher Alexander's combined use of mathematical graphs and hand-made diagrams, and argues that such affinities marked the insertion of *roughness* into architectural computational thinking. Within the techno-scientific context of American postwar architecture, the techniques of transcription and calculation used by Alexander at the Center for Environmental Structure reveal the progressive erasure of determinacy that took place within an architecture practice with empiricist, mathematic and computational preferences. Rather than establishing an optimized and quantified standard to which architecture had to conform, Alexander's rough diagrams and mathematical graphs serialized variation and provided room for indeterminacy and contingency within a clearly defined set of rules.

Keywords Patterns · Computer · Algorithm · Roughness · Graph · Diagram

Introduction

In an interview for *Architectural Design* in March 1971, architect and mathematician Christopher Alexander complained about the negative connotations that free-hand sketches had acquired within the techno-science fervor of architecture postwar debates. The profound "stiffness" with which some of his own sketches had been redrawn in a "design-methods" publication, he claimed, signaled a larger disciplinary problem that inadequately associated exactitude with rationality:

> I made the drawings and they were very rough free-hand sketches. I sent them to the editor, carefully explaining that the *roughness* and free-handness was deliberate, the reason being very simple: namely, the patterns that I was describing are extremely fluid entities and the free-hand drawing captures

✉ Diana Cristobal Olave
dc37@princeton.edu

1 Princeton University, Princeton, US

Birkhäuser

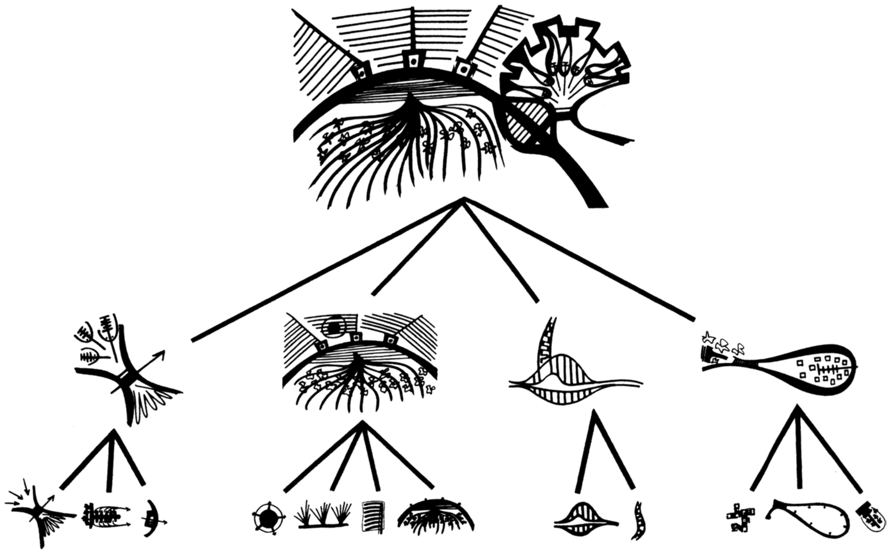

Fig. 1 Alexander's combination of graph structures and hand-drawn diagrams Source: Alexander (1964b: 153). Reproduced with permission from the publisher from *Notes of the Synthesis of Form*, by Christopher Alexander. © 1964 by the President and Fellows of Harvard College. © renewed 1992 by Christopher Alexander

> the *fluidity* much better than a precise machine-like drawing… It may sound trivial. I don't think it is. The idea that the discipline cannot tolerate the idea of a free-hand drawing is a rather serious indication of the state of mind that prevails among the people who practice it (Alexander 1971: 768).

Positioned against the rigid determinacy of the technical drawing, sketches with rough, crooked and hatched lines of different thicknesses filled the publications of the Center for Environmental Structure, a non-profit corporation founded by Alexander in 1967. Most importantly, this shift took place within a practice concerned with mathematical and computational methodologies. The Center adopted a conglomerate of mathematical techniques that entered architectural practices at the height of the Cold War, and that moved away from quantitative certitude and precise mensuration towards the search for mathematical relationships and structures—graph theory, set theory, and game theory among others. This so-called "new" and "modern" mathematics permeated a wide range of humanistic disciplines, providing the "soft sciences" with a hard mathematical core (Vardouli 2017, 2020; Steingart 2020). Alexander did not only participate in this mathematical turn, he also contributed to increase the indeterminacy of such techniques by intertwining graph theory with open-ended free-hand sketches. He termed these rough sketches *diagrams* (Fig. 1)

This paper examines Alexander's combined use of mathematical graphs and hand-made diagrams, and argues that such affinities marked the insertion of *roughness* into architectural computational thinking. Alexander's graphs have recently caught the attention of scholars, who have highlighted such mathematical structures in relation to histories of computation (Steenson 2014, 2017), or who have used graph theory to provide a graphic and mathematical representation of his popular design guide *A Pattern Language* (Dawes and Ostwald 2020). This paper explains how graph theory substituted earlier quantitative reasoning, by also paying closer attention to the fact that hand-made rough diagrams were used in combination with such mathematical graphing techniques. It further argues that Alexander's hand-made diagrams provided room for contingency, variability and adaptability within the structural determinism of the mathematical graph. It was at the intersection of the mathematical graph and the hand-made diagram that it became possible for Alexander to regard algorithmic rule-following as open-ended and not deterministic. This shift towards *algorithmic roughness* should not only be regarded as distinctive of Alexander's approach, but rather as a contribution within intellectual histories of postwar mathematics in architecture, when concerns with precise mensuration shifted towards open-ended algorithms.

The paper begins by examining postwar critiques of the computer in architecture that centered around the idea of 'over-precision,' focusing on Alexander's contribution within the 1964 First Boston Conference on Architecture and the Computer. Then, it discusses Alexander's use of hand-made diagrams and mathematical graphs and it examines the shift towards roughness against the backdrop of twentieth-century mathematical and computational cultures. Finally, it looks at the consequences of such a move. Under *algorithmic roughness*, two consequences followed. First, the role of computers in the production of architectural objects shifted from methodologies that aimed at optimizing precise quantifiable parameters (such as distances), to methodologies that afforded contingency and variability through open-ended algorithms. Alexander's so-called "patterns" embodied this transition by providing a reusable and recurring methodology that was also adaptable, tentative, transformable and rough. Second, a stream of endlessly variable architectural solutions were generated from a single set of rules, providing almost personalized buildings for users with heterogeneous tastes and ways of life. Unlike artisanal customization, this kind of computer-aided variability promised an unlimited number of non-standard variations using only one set of rules. The graph dictated a topological sequence, whilst the hand-made diagram provided cushioning and variability. Rather than marking an optimal standard to which architecture had to conform, Alexander's patterns provided exhaustive variations and nuances within a single set of rules.

Computing Roughness

Prior to the foundation of the Center for Environmental Structure, Alexander's work was well-known within the "Design Methods" movement for incorporating architecture into a wider assemblage of disciplines and techniques, including set and graph theory, algorithms and probability theory. Such techniques were used as corrective mechanisms to subjective decision-making, offering a picture of rationality defined by sequential rules, rather than judgment, inferences, and other more traditional ideals of reason (Erickson 2013: 1–24; Lobsinger 2013: 652–685). At the Center, he continued to embrace such systematic approaches to design, while also rejecting the structural hardness and exactitude required by some of these approaches (Alexander 1966: 185–190). In other words, Alexander was critical of, yet expanded upon, his own previous techniques. A second methodological trend more concerned with ideas of roughness, can thus be seen superimposing itself to the first one, concerned with quantitative exactitude.

This shift towards roughness should also be seen as the result of Alexander's concerns with the computer, which at that moment was a tool that required precise quantification. At the 1964 Conference on Architecture and the Computer, Alexander—who was one of the first architects to write his own computer programs—polemically confronted major proponents of computer-aided design. Comparing the computer to "a huge army of clerks… all stupid and entirely without initiative," (Alexander 1964a: 52) he opposed the tendency that attributed artificial intelligence to the computer and to the visually-oriented computer graphics trend that was also popular in the 1960s. Yet his major criticism to the use of the computer in architecture was articulated from the point of view of precision: "if the love for precision outweighs our ability to pick significant problems, and our ability to distinguish the relevant from the irrelevant, then we must admit that this compulsion to be precise has made us bankrupt" (Alexander 1964a: 53). His criticism against what he qualified as "trivial over-precision" (Alexander 1964a: 53) did not stop there. Citing a contemporaneous study of computer-aided planning in hospital design that estimated the amount of traffic between a series of possible room types in a hospital, he complained that the use of the computer had forced the authors of this particular study—and many other authors also concerned with computer-aided systems approaches to architecture—to use only parameters that could be easily measured (such as the amount of walking done by patients), and to deal with unnecessary precise mensuration:

> Any intelligent designer could examine the various hospital plans examined by the computer and could tell roughly what relative amounts of different traffic they would generate. The key word here is "roughly." It is unnecessary to know the amounts of walking generated by a plan to the second decimal place, because it is irrelevant—and only has the appearance of accuracy. It is *insignificant accuracy*. It is like measuring the size of a cooking apple with a micrometer (Alexander 1964a: 53).

Despite these counter-arguments to widespread uses of the computer in architecture and design, Alexander left no doubt that his criticism was not to the computer itself. To Alexander the computer was a "wonderful, almost miraculous invention" (Alexander 1964a: 54) that did *not* threaten intuition and creativity. In his opinion, the computer's major potential as regards to design and architecture was at a structural level. Alexander rejected the computer as an instrument that could bring precise mensuration into architecture, and instead embraced the internal protocols of computation from a methodological standpoint, as a set of codes that could be used to formulate a rough and structure-oriented design process, via graph theory.

Alexander was not the only one to use graph theory and apply it to architecture during this period. Architects working at computer laboratories across Great Britain and the United States popularized this method to provide plan layout analyses (March and Steadman 1971; Broadbent 1973; Ostwald 2011; Vardouli 2020). Neither was he the only one who aimed to use computers while avoiding the "over-precision" of quantitative techniques. For instance, the global planning agency Doxiadis Associates pioneered methods of cluster analysis in urban planning (DA 1970), and the Harvard Laboratory for Computer Graphics applied statistical methods of analysis and prediction through computer-generated low resolution maps (McMahon 2013). Yet, Alexander's singular methodology combined the "relational" thinking of graph theory with a critique to computational exactitude. Unlike the methodologies that followed the tradition of statistics and urban planning disciplines, and therefore used massive quantities of data, Alexander stopped seeing mathematics as a science to extract and measure "large numbers," and proposed the study of "qualitative" structure of relations (Lévi-Strauss 1954).

Within the debates around the use of computers in architecture, the simultaneous use of mathematical graphs and hand-made diagrams contributed to push forward a structure-oriented view of computation, that could increase flexibility through open-ended algorithms. Borrowing Alexander Galloway's definition of a computer protocol as "an algorithm, a proscription for structure whose form of appearance may be any number of different diagrams or shapes" (Galloway 2004: 30) it could be argued that the structure of *A Pattern Language* incorporated computational protocols into a design methodology without computers. In other words, if in the early 1960s Alexander used the computer as a direct instrument for design, in *A Pattern Language* the computer was reimagined as a methodology rather than as a tool. Such a computational methodology promised to escape precise mensuration and reveal another set of qualities based on topology and graph theory.

Rough Diagrams

The handbook *A Pattern Language* was published in full in 1977. The Center described it as "a coordinated body of design solutions" that would improve cumulatively over time through public scrutiny and criticism. Positioned against the supposedly prevailing top-down tendencies embedded in processes of standardization, these "design solutions" were proposed as alternatives to modular systems, grids and prefabricated building systems. Yet at the heart of the book lay the notion of a "true invariant." The notion of invariant, which Alexander borrowed from mathematics and biology, differed from other more explicit notions of architecture standards because they were "processes," "operations" and "relationships," rather than physical objects. In other words, the object of standardization was of an intangible nature and subject to interpretation.

Alexander and his peers argued for the presence of hidden forces, relations and structures that shaped the interaction between the subject and the environment in a social, psychological and spatial way. Diagramming was the technique through which the Center explored questions of systematization of relations.

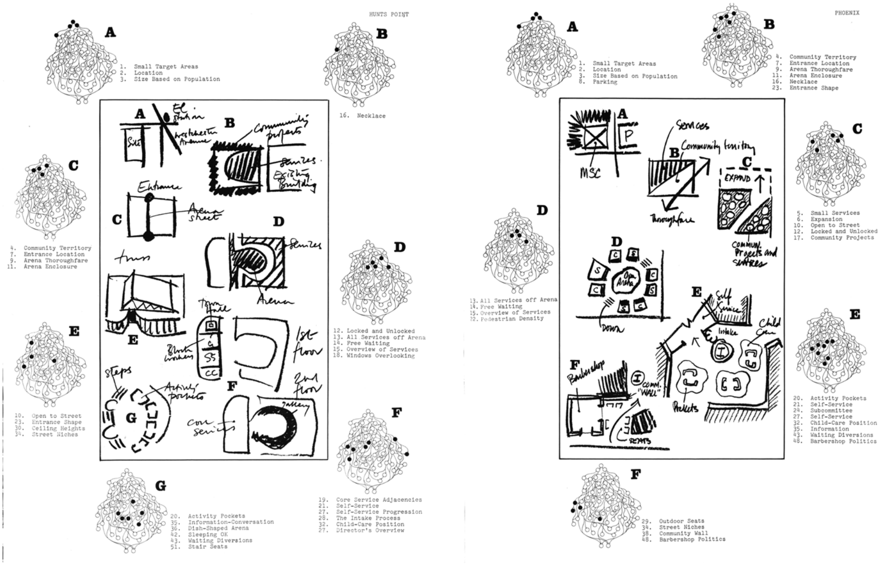

Fig. 2 Collection of pattern diagrams associated with a graph Source: Alexander et al. (1968). *A pattern language which generates multi-service centers*. Berkeley, Calif., Center for Environmental Structure: 23, 39 (Alexander 1968). Reproduced with permission from Christopher Alexander and the Center for Environmental Structure

The authors proposed the diagram as the means of transcription by which spatial relationships—formulated through terms such as *adjacency, concavity, betweenness, facing towards*, etc.—could be made explicit, visible and thus systematized.

> If you can't draw a diagram of it, it isn't a pattern… A pattern defines a field of spatial relations, and it must therefore always be possible to draw a diagram for every pattern. In the diagram, each part will appear as a labeled or colored zone, and the layout of the parts expresses the relation which the pattern specifies (Alexander 1973: 33). (Figure 2)

Diagramming was neither a new word, nor a new technique in architecture. However, with the advent of digital computing the term took on a slightly different meaning that emphasized its performative rather than representational qualities, and that highlighted aesthetics of data and of mapped information. Theorist Alexander Galloway has referred to the algorithmic nature of diagrams as scripted procedures that, together with the technology of the computer and the management style of what he called a "protocol," define "societies of control" (Galloway 2004: 29–55). Architect Georges Teyssot has turned to Gilles Deleuze to explain the algorithmic nature of architecture diagrams under digital computing. Using Deleuze's notion of the diagram as an "abstract map of relations between forces," Teyssot has proposed to look at the diagram as a graphic inscription that is "based on adaptable (customable) software, capable of producing changing modalities of a structural topology driven by performance" (Teyssot 2012: 02). That is, a form of inscription that deals with the representation of dynamic modulations of forces rather than with fixed values.

In a similar way, the diagrams in *A Pattern Language* are instructions that denote a set of indeterminate spatial relations. The framing of the diagram as a visual image that transforms the perception of the physical environment is a theme that relates, to an extent, to the previous works of Kevin Lynch and Gyorgy Kepes at the MIT-Harvard Joint Center for Urban Studies, even though these earlier examples used terms such as *imageability* and *image-making* rather than *diagramming*. But despite their similarities and common grounding in cognitive and environmental psychology, Alexander criticized Lynch's concept of imageability, arguing that Lynch's images were rooted in a visual rather than in an operational logic. This shift from an image-as-representation to an image-as-process is precisely what distinguishes the diagrams of *A Pattern Language*. Even though handmade, these diagrams are the result of a computational mental construction that organized knowledge according to a problem-solving design methodology. "In the context X, the problem Y will occur unless the field Z [the diagram] is also present. Therefore, if you are designing an environment of the type X, create this relation among parts Z" (Alexander 1973: up). Borrowing Lorrain Daston and Peter Galison's notion of the image as "coming-into-existence" (an image that functions less for re-presentation than for presentation) Alexander's diagrams have a double purpose as simultaneously images-as-evidence—concerned with demonstration—and images-as-tools—concerned with making (Daston and Galison 2007: 382-412). In other words, these are diagrams that were meant to be used as working objects rather than diagrams meant to exclusively interpret an actual environment.

Roughness was instrumental in enabling this shift from visual to operational inscriptions. Alexander used crooked lines and endless hatching, and described the diagrams of *A Pattern Language* as "cloudy volumes" with "imprecise sizes and edges" (Alexander 1973: 2-3). Making use of biological analogies, his diagrams blur the edges of buildings to indicate a process of design that is capable to dynamic transformations through a process of increasing differentiation.[1] His strategic use of roughness, blurriness and indeterminacy enabled a tentative and indeterminate approach that intended to avoid modular repetitions, bring variation and progressive individualization into a computational process of design. If architectural structures were to enjoy the same degree of variability as biological ones, then the diagram was the necessary indeterminate medium that could enable variability, accommodation, and adaptation within an otherwise excessively 'rigid' and 'precise' computational methodology.

[1] Alexander's biological analogies are of biological processes, not of forms. On Alexander's biological analogies used in *Notes,* see Steadman (1979: 163–179).

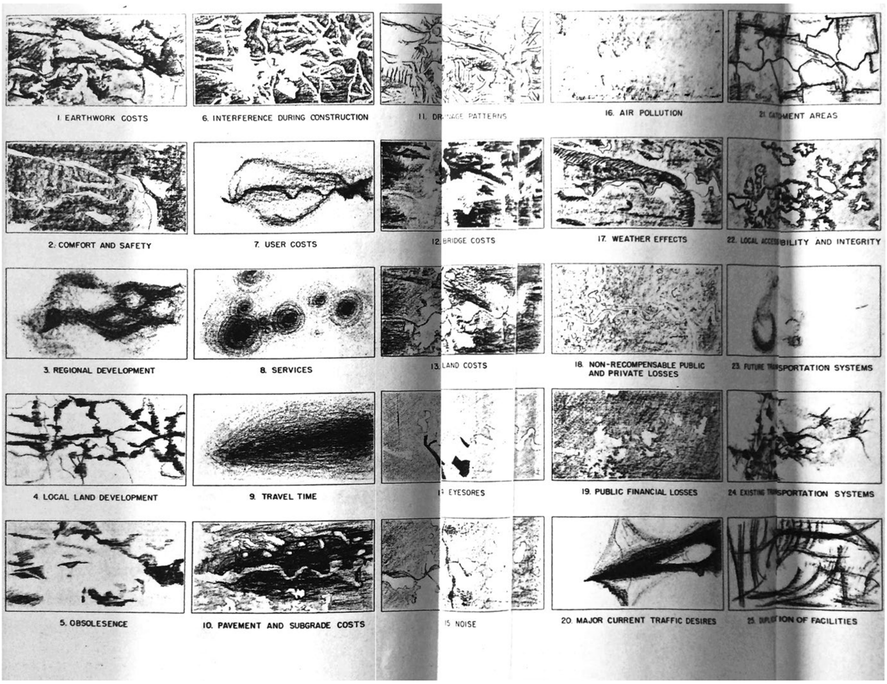

Fig. 3 Pattern diagrams of highway route location Alexander and Manheim (1962: 4). Reproduced with permission from Christopher Alexander and the Massachusetts Institute of Technology

Calculated with the Computer, Solved by Eye

Alexander's interest in the use of diagrams may be traced back to projects he developed with Marvin L. Manheim in the Civil Engineering Systems Laboratory at MIT in the early 1960s, with the computer available in the MIT Computation Center. The 1962 project, "The Use of Diagrams in Highway Route Location: An Experiment", which was described as "an experiment in the use of diagrams in design," (Alexander 1962: 30) was one of a series of exercises that aimed to combine the use of hand-made diagrams with Alexander's computer program for decomposition of mathematical graphs (the HIDECS program). To follow the computer protocols embedded in the HIDECS program—a set of rules organized by a hierarchical, inverted tree-structure—Manheim and Alexander outlined a problem-solving methodology and described the physical environment through endless lists of problems and requirements. They began by translating a set of self-imposed design requirements (such as "obsolescence," "user costs," "travel time," etc.) into hand-made charcoal drawings that indicated roughly the preferred geographical locations for each individual requirement. They labeled these drawings as diagrams and described them as "patterns of greys whose density varied over the complete range from white to black," (Alexander 1962: 33) where black represented a very good potential location for the highway and white a bad one, so far as the individual requirement was concerned (Fig. 3).

In order to combine the information contained in these diagrams Alexander and Manheim superimposed them photographically, following the organizational scheme provided by the HIDECS program. In other words, the computer guided the combinatorial sequence, but Alexander and Manheim superimposed analogically each photographic negative. They modified and adjusted the relative exposure times of the negatives in order to bring out the characteristics of each diagram as strongly as possible. Finally, they projected the combined composite photograph onto a drawing board and redrew the image by hand "in such a way as to bring out its essential organizational features" (Alexander 1962: 115). Manheim and Alexander claimed that the human eye was able to detect underlying common patterns better than a computer. In their own words, the eye became a "special-purpose computer" (Alexander and Manheim 1963: 90). In the face of a process with so many possible variations, they considered digital computers to be "too little advanced to be of much use" (Alexander and Manheim 1963:117) whereas they considered the eye and the brain to be "flexible enough not to need rigid relations between utilities" (Alexander and Manheim 1963: 117) (Fig. 4).

Diagramming, as defined by this exercise, was a practice that made possible the comparison between things that were unlike each other. Land costs, noise, safety, etc. were translated into degrees of black and white that expressed a condition of potentiality on a point basis. But unlike the Geographic Information Systems that were being developed at this period at Harvard University's Laboratory for Computer Graphics and Spatial Analysis, which also replaced paths by terrain points and compared different informational systems by utility maps, the technique put forward by Alexander and Manheim relied on a medium with a very low resolution. The *low resolution* of the diagramming technique was essential to compensate, mitigate and accommodate the conflicts that appeared between the different steps of the mathematical graph.

Manheim and Alexander used computational protocols organized by mathematical structures to demonstrate objectivity of method and also varied the degree of the resolution in hand-made blurry diagrams, which enabled variability and adaptability. It is precisely this combination of a mathematical graph's structural determinism—which breaks down the complexity of a design problem into a series of smaller units and dictates an ordered sequence of steps—with the indeterminism of the diagram that characterizes the logic of the latter research on patterns. Just as rough diagrams were in this exercise used to provide room for change within the structural determinism of a mathematical graph, so too were rough diagrams used in *A Pattern Language* to provide cushioning and variation within an otherwise rigid set of prescriptive rules.

Malleable Mathematical Graphs

In *A Pattern Language* the sequence of different patterns is visualized by a connected mathematical graph ordered in the form of a network (Figs. 5, 6). This type of mathematical structures proliferated in the American social and information sciences during the Cold War (Harary and Norman 1976). Graphs

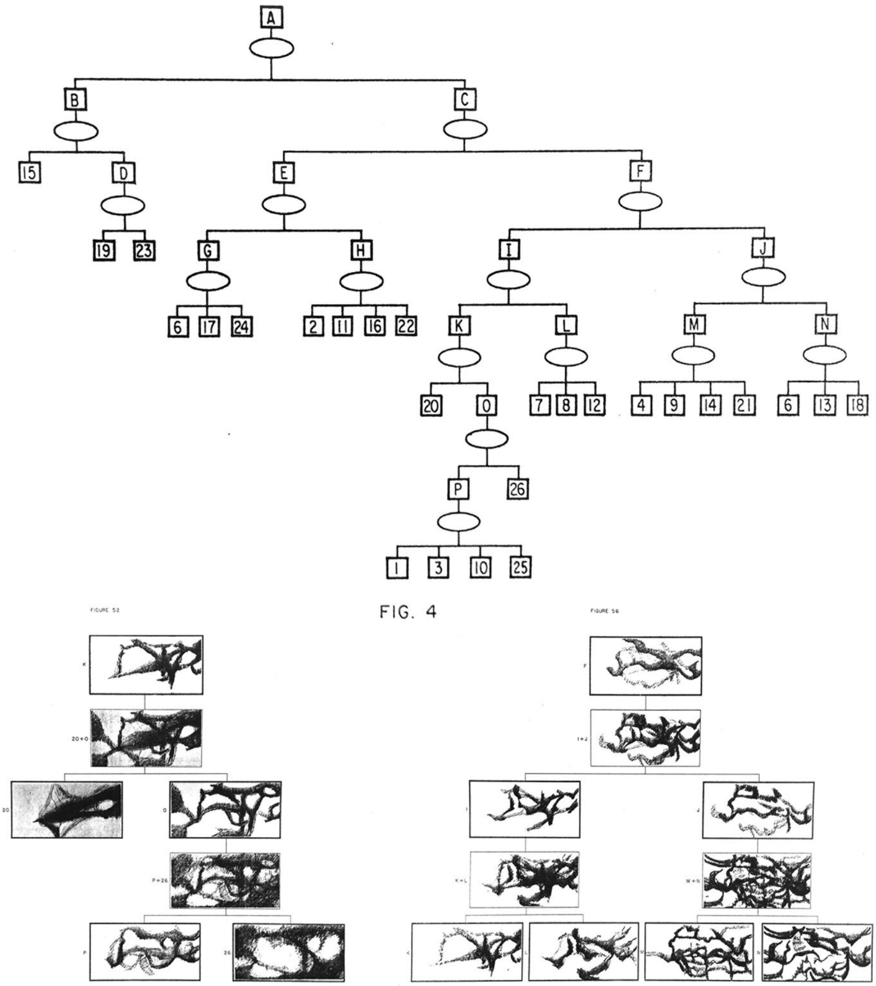

Fig. 4 Tree used to guide the combination of diagrams Source: Alexander and Manheim (1962: 6, 12, 16). Reproduced with permission from Christopher Alexander and the Massachusetts Institute of Technology

and graph-theoretic methods were borrowed from mathematics as a means to calculate in a non-numerical manner qualitative fields of research, applying order to complex systems of information that otherwise were considered arbitrary or random. By associating attributes (i.e. letters, words, names, etc.) to the nodes of these mathematical structures, information theorists, like Claude E. Shannon, and institutes for social research, like the Research Center for Group Dynamics in Michigan, used these abstract mathematical tools to represent linguistic sequences and interpersonal relations in social groups.

In the exercises developed at MIT, Alexander and Manheim used graphs to rationalize architectural problems that could not be measured quantitatively

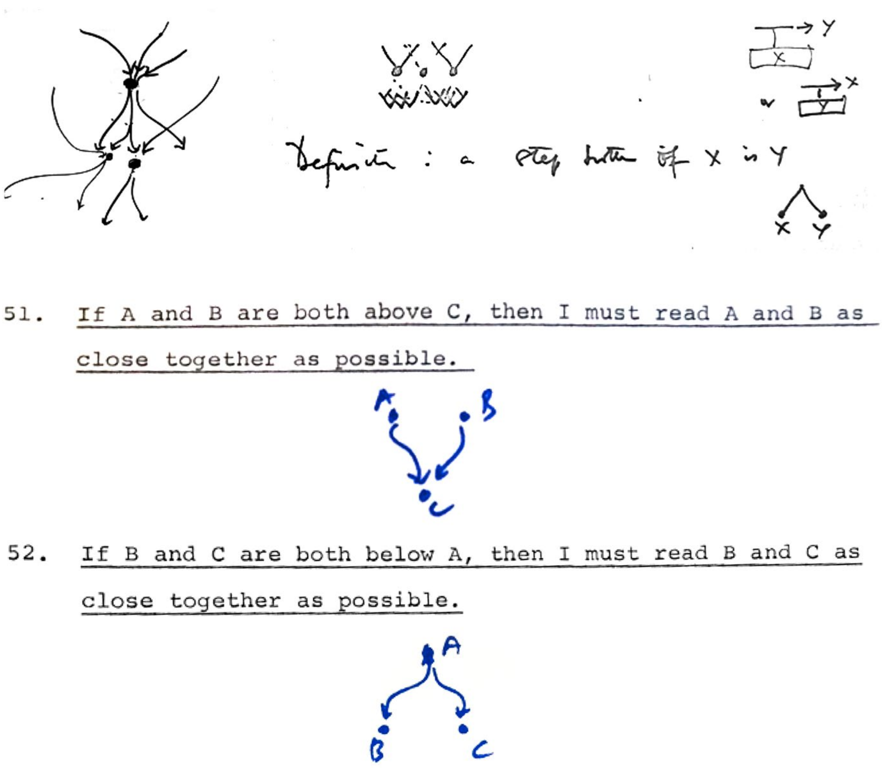

Fig. 5 Graphs drawn in the draft versions of *A Pattern Language* Source: Alexander (1969: up; 1973: 2-3-30). Reproduced with permission from Christopher Alexander and the Center for Environmental Structure

using numerical standards. They associated graph nodes to what they termed as *misfits* (lists of architectural requirements that should be satisfied) and used the mathematical function proposed by Shannon in *The Mathematical Theory of Communication* to organize these lists of misfits into an ordered sequence of steps. Graphs provided an organizational structure that could be easily translated into the IBM machines and that therefore could ensure the computation of a wide range of different combinations in very little time. Nevertheless, in *A Pattern Language* the graph was neither dictated by a mathematical formula nor programmed by a computer. The architect Sara Ishikawa, co-author of the handbook, explained how, in the works developed by the Center, graphs were produced manually by shuffling hand-made diagrams of individual patterns on big pieces of paper and reorganizing them according to a flexible scalar logic[2] (Fig. 7). The graph was no longer an

[2] This description referred to first research project developed by the Center, published in a book entitled *A Pattern Language Which Generates Multi-Service Centers*. Interview with Sara Ishikawa by Diana Cristobal. August 25th, 2017.

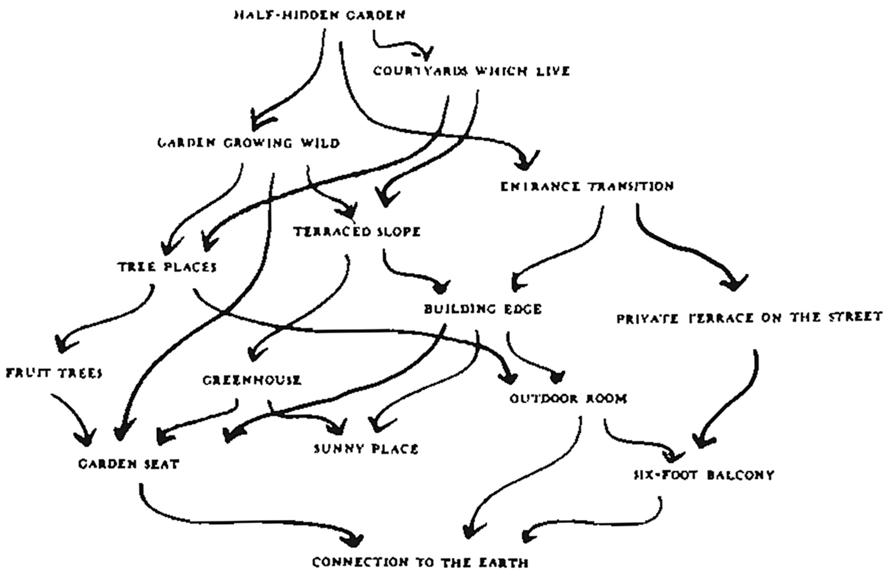

Fig. 6 Graph drawn for the published version of *A Timeless Way of Building* Source: Alexander (1979: 314). Reproduced with permission from Christopher Alexander and Oxford University Press

exact calculating apparatus, but a notational technique used to make explicit relationships between patterns at different scales. In other words, initially Alexander borrowed mathematical graphs as tools to decompose complex informational systems into smaller subsystems, guided through Shannon's mathematical theory of communication and thus measured by an algorithm that minimized the amount of information into an algebraic minimum. Eventually, the graph evolved into a notational system that was produced and ordered manually.

In *A Pattern Language*, Alexander associated the nodes of the mathematical structure to individual patterns at different scales, and used the links to organize the patterns in a nested scalar system that developed in a progressive and adjustable manner from large regional scales to construction details of buildings. Furthermore, each node within the graph structure was represented by a hand-made rough diagram that was open-ended, and thus potentially adaptable by the user of the handbook. This shift from calculation to notation should be explained as a strategy to avoid the over-determination of a rigid mathematical structure. The transition from numerical formulas to malleable notational systems were progressive steps towards roughness.

At the heart of this preference lay the idea that Alexander considered roughness and variation to be intimately linked. In an *Architectural Design* article in July 1968, Alexander opened up with the following question: "In the world today newly constructed houses and apartments are more and more standardized; yet people are very different… how can we make dwellings in such a way that 100,000 dwellings are as different from one another, and as articulately personal, as 100,000 people?" (Alexander 1968: 324). Architecture, he continued, had to be able to adapt to personal idiosyncratic characteristics. He portrayed architecture as a problem of

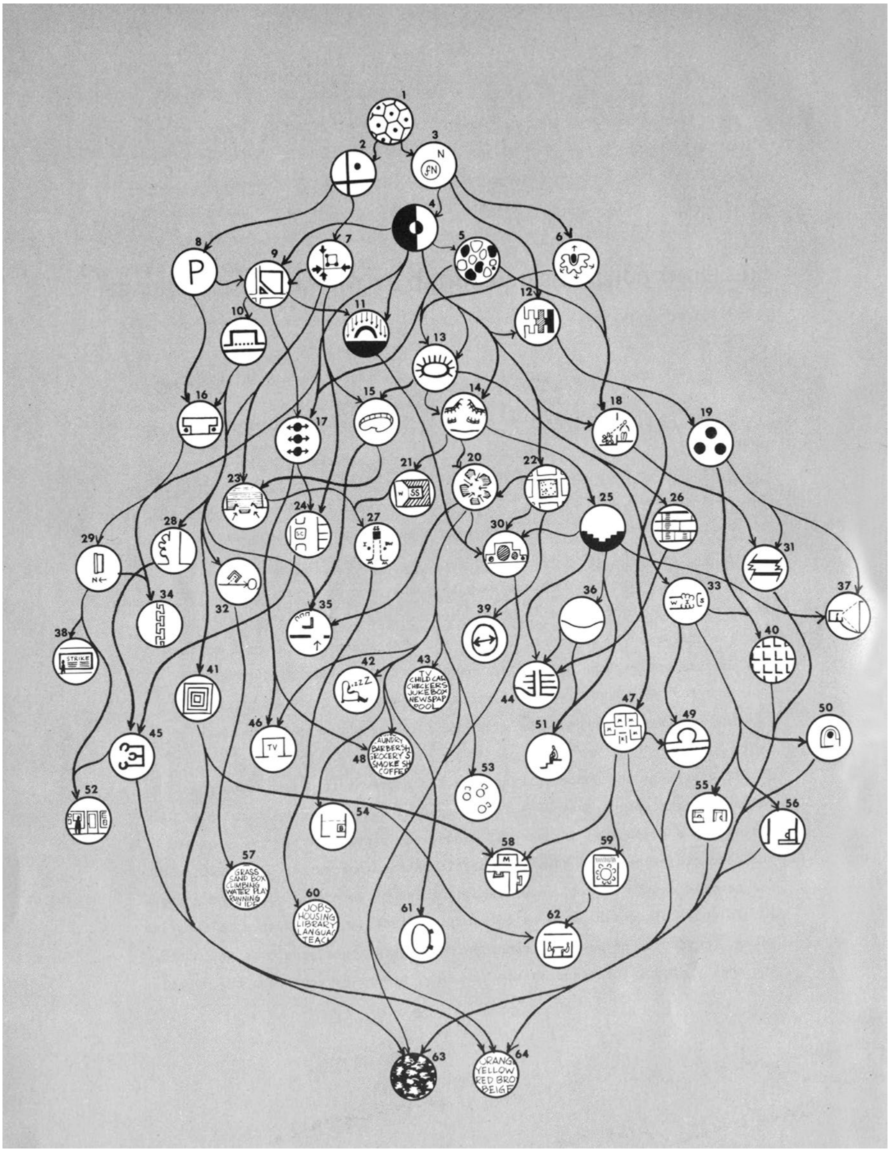

Fig. 7 The graph as a hand-drawn notational technique Source: Alexander (1968: 18). Reproduced with permission from Christopher Alexander and the Center for Environmental Structure

assisting individualization and referred to users as quasi-singular individuals with distinctive personal tastes, behaviors, beliefs and opinions. He carefully described the stories, tastes, personalities and ways of life of users and took these descriptions as evidence for the need to conceive a stream of endlessly variable architecture solutions. Variation, he claimed, "must be in large scale, and it must therefore include some kind of uniformity of process" (Alexander 1985: 75). *A Pattern*

Language hinges around this idea. Alexander used the mathematical structure of the graph to demonstrate rationality and uniformity of method, but transformed this structure into a malleable notational system that interlinked rough diagrams, thereby providing room for “subtle” and “individual” variations. What resulted was a handbook that promised to adapt to idiosyncratic preferences, such as subtle variations of room sizes, window shapes, ceiling heights and wall thickness. This aesthetic of variation, achieved through rough diagrams and malleable graphs, provided an appearance of customization and perpetual modification. The handbook demanded variety for its own sake and claimed that roughness was instrumental to its attainment.

Conclusion

In *A Pattern Language,* variability was governed by a profound regularity, a single set of rules. This set of rules was conceived to generate almost personalized buildings for users with heterogeneous tastes and ways of life. Rather than establishing an optimized standard to which architecture had to conform, Alexander’s diagrams and mathematical graphs serialized variation. Unlike artisanal customization, Alexander’s variability was simultaneously mass-produced and hyper-individuated. It promised an endless array of varying architecture solutions at no extra cost, via algorithmic roughness. Algorithmic roughness, in the form of hand-made diagrams and malleable mathematical graphs, was used to avoid the over-precision and over-determination of computer techniques, such as plant layout analysis or linear programming. In other words, Alexander brought roughness into computational thinking.

This association between the algorithmic rule following of the mathematical graph and the roughness of the hand-made diagram marked the insertion of indeterminism in the use of computers into design. The term *pattern* embodied such a transition. Derived from the Latin *patronus* (“protector, master”), the word *pattern* (“something shaped to serve as a model”) has been used in a wide set of different disciplines to denote a reusable, repetitive or recurring element, from textiles to ornaments, mathematics, geometry or linguistics (OED 2020). In the early 1960s, Alexander first adopted the term *pattern* from perceptual principles in Gestalt psychology, but gradually moved away from concerns with visual perception to embrace a broader definition in *A Pattern Language*: “a solution to that problem that occurs over and over in the environment, in such a way that you can use this solution a million times over, without ever doing it the same way twice” (Alexander et al. 1977: 10). This definition gained popularity in computer science and was later used by object-oriented software engineers to denote a general reusable solution to a commonly occurring problem within a given context in software design. Computer scientists, who were an important audience for *A Pattern Language*, describe Alexander’s definition of a pattern as a concept that instigated a change from linear programming to a less centralized and “more flexible and reusable” way of writing code (Gamma 1995).

Alexander's usage of the term *pattern* differs from earlier uses of the term precisely in the dialectic placed between repetition and variation, thus portraying architecture as a medium to satisfy varying personal choices within a set of algorithmic rules. In *A Pattern Language,* the introduction of rough modes of inscription permitted the writing of design instructions as open-ended algorithms that could generate endless design variations. This design method replaced concerns with fixed standards and modularity, and identical copies with what Mario Carpo has called "non-standard serial variations" (Carpo 2011). That is, a set in which each item is different, but also share something in common with all the others. Alexander's patterns expressed endless individualities and variations that older technologies could not support, and they did so by incorporating roughness into computational thinking. It was at the intersection of these two seemingly opposing paradigms that Alexander's contribution to computer-aided design should be identified, that is, to escaping from precision and to reveal another set of dynamic and fuzzy qualities based on open-ended algorithms and non-standard variations.

References

Alexander, Christopher. 1964a. A Much Asked Question About Computers and Design. In *Architecture and the Computer; Proceedings of the First Boston Architectural Center Conference*, 52-54. Boston: Boston Architectural Center.

Alexander, Christopher. 1964b. *Notes on the Synthesis of Form*. Cambridge: Harvard University Press.

Alexander, Christopher. 1966. Relational Complexes in Architecture. *Architectural Record*. 140(3):185-190.

Alexander, Christopher. 1968. Thick Wall Pattern. *Architectural Design*. 38: 324-326.

Alexander, Christopher. 1971. A Refutation of Design Methodology. *Architectural Design* (42): 768-770.

Alexander, Christopher. 1969. Draft for *A Pattern Language*. Alexander's archive.

Alexander, Christopher. 1973. Draft for *A Pattern Language*. Alexander's archive.

Alexander, Christopher. 1979. *The Timeless Way of Building*. New York: Oxford University Press.

Alexander, Christopher and Marvin Manheim. 1962. The Use of Diagrams in Highway Route Location: An Experiment. *Civil Engineering Systems Laboratory Publication* 161. MIT.

Alexander, Christopher and Marvin Manheim. 1963. Technology: Methods and Machines to Shape the Future. *Architectural Forum.* 88-93.

Alexander, Christopher; Ishikawa, Sara and Murray Silverstein. 1968. *A pattern language which generates multi-service centers*. Berkeley: CES.

Alexander, Christopher; Ishikawa, Sara and Murray Silverstein, with Jacobson Max; Fiksdahl-King Ingrid and Shlomo Angel.1977. *A Pattern Language*. New York: Oxford University Press.

Alexander, Christopher with Howard, Davis; Martinez, Julio and Don Corner. 1985. *The Production of Houses*. New York: Oxford University Press.

Broadbent, Geoffrey. 1973. *Design in Architecture*. London, New York: John Wiley & Sons: 224-251.

Carpo, Mario. 2011. *The alphabet and the algorithm*. Cambridge: MIT Press.

Daston, Lorrain and Peter Galison. 2007. *Objectivity*. New York: MIT Press.

Dawes, Michael and Michael J Ostwald. 2020. The mathematical structure of Alexander's A Pattern Language. *Environment and Planning B: Urban Analytics and City Science*. 47(1).

Doxiadis Associates (DA). 1970. DA and the computer. *DA Review*, 6(66): 3-7.

Erickson, Paul; Klein, Judy L.; Daston, Lorraine; Lemov, Rebecca; Sturm, Thomas; and Michael D. Gordin. 2013. *How Reason Almost Lost Its Mind*. Chicago: The University of Chicago Press.

Galloway, Alexander. 2004. *Protocol: How Control Exists after Decentralization.* Cambridge: MIT Press.

Gamma, Erich. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Mass.: Addison-Wesley.

Harary, Frank and Z. Norman, Robert. 1976. *Graph Theory as a Mathematical Model in Social Science*. Ann Arbor: Xerox University Microfilms.

Lévi-Strauss, Claude. 1954. The Mathematics of Man. *International Social Science Bulletin.* 6(4): 581-590.

Lobsinger, Mary Louise. 2013. Two Cambridges. In *A Second Modernism*, ed. Arindam Dutta, 652-685, Cambridge: MIT Press.

March, Lionel and Philip Steadman. 1971. *The Geometry of Environment.* London: RIBA Publications.

McMahon, Catherine F. 2013. Predictive Machines. In *A Second Modernism,* ed. Arindam Dutta, 436-473, Cambridge: MIT Press.

OED Online, 2020. Oxford University Press, June 2020. Web. Accessed 17 June 2020.

Ostwald, Michael J. 2011. The Mathematics of Spatial Configuration: Revisiting, Revising and Critiquing Justified Plan Graph Theory. *Nexus Network Journal*, 13: 445–470.

Steadman, Philip. 1979. *The Evolution of Designs.* Cambridge: Cambridge University Press.

Steenson, Molly Wright. 2014. *Architectures of Information*. PhD diss. Princeton University, Princeton.

Steenson, Molly Wright. 2017. *Architectural Intelligence*. London: MIT Press.

Steingart, Alma. 2020. The Axiom of High Modernism. Center for Comparative Media seminar. 2 March.

Teyssot, Georges, 2012. The Diagram as Abstract Machine. *V!RUS*. 7. <http://www.nomads.usp.br/virus/virus07/?sec=3&item=1&lang=en>. [Accessed: 20 June 2020].

Vardouli, Theodora. 2017. *Graphing Theory*. PhD diss. MIT, Boston.

Vardouli, Theodora. 2020. Skeletons, Shapes, and the Shift from Surface to Structure in Architectural Geometry. *Nexus Network Journal,* 22: 487–505.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Diana Cristobal Olave is an architect and scholar, currently pursuing a Ph.D. in the Department of Architecture at Princeton University. Her research interests are situated at the intersection of architecture, media, politics and technical infrastructures in the modern period, with a special focus on techniques of calculability and visualization. In her dissertation, she theorizes the rise of algorithms during the fascist regimes that ruled Spain and Portugal in the 1960s and 1970s, and traces its applications and impacts in modern architecture and urban design. Diana was trained as an architect at ETSABarcelona. As a Fulbright Fellow she graduated from the MSc in Advanced Architectural Design at Columbia Graduate School of Architecture, Planning and Preservation. She is a founding partner of the architecture collective KnitKnot Architecture; has practiced as an architectural designer in New York, Spain and the Netherlands; and taught in a number of schools of architecture.