Representing Architectural Design Using a Connections-based Paradigm

Ganapathy Mahalingam, Ph.D.

Abstract

Any *making*, including a work of architecture, is synthetic in nature and is made by making connections. To base the core of a computational representation of architectural design on *connections* is to base it on the very core of making. The articulation of the core of architecture, its architectonics, should be based on articulating its connections. This paper probes how connections can serve to represent architectural design. A paradigm consists of a core cluster of concepts that, for a time period, provides a framework to articulate the issues and problems facing a field and to generate solutions. This paper offers a connections-based paradigm to represent architectural design computationally. A number of connections-based strategies for the representation of architectural design have emerged. Modeling frameworks that have been identified include dendograms, bipartite graphs, adjacency graphs, plan graphs, planar graphs, Hasse diagrams, Boolean lattices and Bayesian networks. These modeling frameworks have enabled the representation of many aspects of architectural design. Is it possible to extract a uniform modeling framework from all these frameworks that enables the computation of architectural design in all its aspects? Using biological analogies, will an integration of these modeling frameworks provide the 'molecular' structure of a 'DNA' that makes up the architectural 'genome'? This paper will attempt to answer these questions.

1 Introduction

Resolving the computability of design has been a longstanding quest among researchers in computer-aided architectural design. One does not question whether design, as a cognitive activity, is possible, but one does question whether design is computable. Remarkable advances are being made in cognitive modeling using computer-based systems. The key to making design truly computable may lie in these advances in cognitive modeling. Researchers, who conclude that design is not computable, or is not computable in its creative aspect, invariably point out a computer-based system's inability to generate radically new forms. When you carefully examine what constitutes a radically new form, the answer that emerges is new connections at various topological levels.

Articulating the synthesis of forms or the generation of spatial organization has a long tradition spanning four decades. Beginning with Christopher Alexander's *Notes on the Synthesis of Form* (1964), the design profession has wrestled with the articulation of architecture. Lionel March and Philip Steadman's pioneering work on the geometry of the environment (March & Steadman 1971), followed by Lionel March's work on the architecture of form (March 1976) have set the precedent decades ago for what may now form the core for representing architectural design.

Articulating spatial organization found new energy in Hillier and Hanson's work on the social logic of space (Hillier & Hanson 1984). Hillier and his team have recently expanded their research to the broader framework of space as a machine (Hillier 1996). In doing so, they have begun to make the case for a non-discursive, analytical theory of architecture based on 'configurations.'

In his seminal work from the early 60s, Alexander attempted to get at the core process in the synthesis of forms, an issue central to architecture. In the preface to a later edition of his book on the subject (Alexander 1964), he carefully articulated his quest and its significance. He wrote:

> "In this book I presented the diagrams as the end results of a long process; I put the accent on the process, and gave the diagrams themselves only a few pages of discussion. But once the book was finished, and I began to explore the process which I had described, I found that the diagrams themselves had immense power, and that, in fact, most of the power of what I had written lay in the power of these diagrams. The idea of a diagram, or pattern, is very simple. It is an abstract pattern of physical relationships which resolves a small system of interacting and conflicting forces, and is independent of all other forces, and of all other possible diagrams. The idea that it is possible to create such abstract relationships one at a time, and to create designs which are whole by fusing these relationships—this amazingly simple idea is, for me, the most important discovery of the book."

This book led to the widespread use of diagrams or patterns as the basis for designs, especially architectural designs. The subsequent search for diagrams and patterns extended the scale attempted by Alexander in his book, the scale of a village in India. The thought that diagrams could form the basis of designs took firm root. This paper is about an attempt to extend the tradition that began with Alexander's work into a new realm made possible by advanced computing techniques.

2 Connections-based Representations

A connections-based representation is quite simply one that uses connections as an organizing framework for the representation. In this paper, the term "connections-based" is deliberately used rather than "connectionist," which has its own connotations. The connectionist model is a subset of the broader category of connections-based representations.

The term "connectionist" is widely used to describe a computational technique used to model the human brain, especially its neural network. A connectionist model is made up of a network of many simple processing units that act in parallel to produce "emergent" behavior. These simple processing units have been described as intuitive, sub conceptual and sub symbolic entities that are linked in a dynamic system that does not allow a precise conceptual level description. There is a healthy and ongoing debate about the effectiveness of the "connectionist" model to represent the working of the human brain.

A connections-based representation, on the other hand, is a diagram made up of multiple nodes that are linked in various ways. The word "diagram" is based on its Greek roots *dia* and *graphein*, which mean “through” and “write” respectively. The Greek verb *diagraphein* means to “mark out by lines” from which the noun *diagramma* is derived. Originally the word diagram referred to a geometrical figure, and for a brief period even to a written list or register, which is very curious. Based on its etymology, the word “diagram” refers to an intrinsic structure, something that is drawn “through” an entity, like a skeletal framework. When coupled with the etymology of the word “understanding,” diagrams provide intrinsic knowledge of entities. One of the goals of connections-based representations is to help acquire this intrinsic knowledge of architecture. Architecture like connections must be made and is not given (Rajchman 2000). The primary architectural act can be considered as the linking of two nodes. This is the beginning of synthesis and a plurality. Starting with this primary connection-based representation (Figure 1), a hierarchy of connections-based representations can be articulated. Connections-based representations that have emerged in research can be organized based on increasing complexity. Some of the representations that have emerged include the ones described in the following sections.

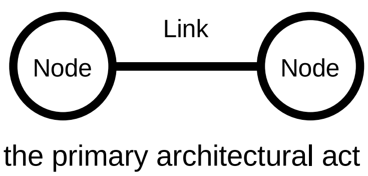

Figure 1. The architecture of connections.

2.1 Dendograms

A dendogram is a diagram that has a branch-like structure (Figure 2). Starting from a single node, branches or links lead to successive nodes. Examples of dendograms are parse trees, decision trees and binary trees. Each terminal node of a ‘tree’ representation is appropriately called a leaf. In representations such as decision trees, the leaves represent outcomes that are a result of decisions made at the nodes. Parse trees can be used to verify if a particular architectural composition has been created using a particular architectural language. Decision trees can be used to represent a design process as a hierarchical sequence of design decisions, where each design decision leads to subsequent design decisions. Because of their architecture, dendograms are useful in representing hierarchical procedures or processes. A dendogram or tree of (n) nodes has (n-1) links. Figure 3 illustrates this rule.

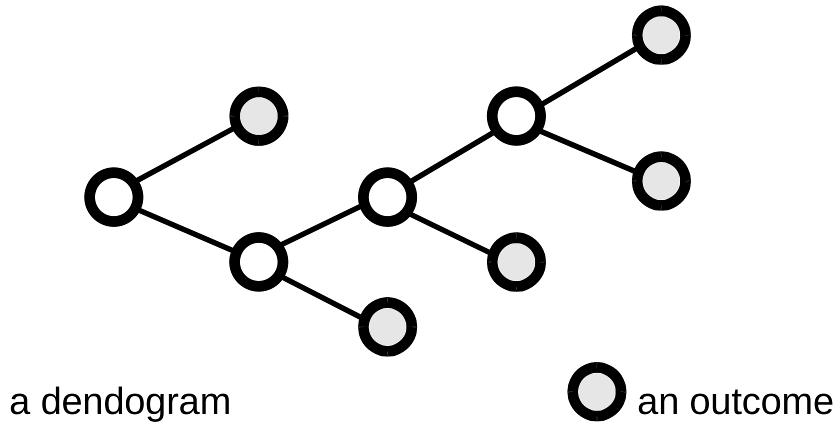

Figure 2. A dendogram.

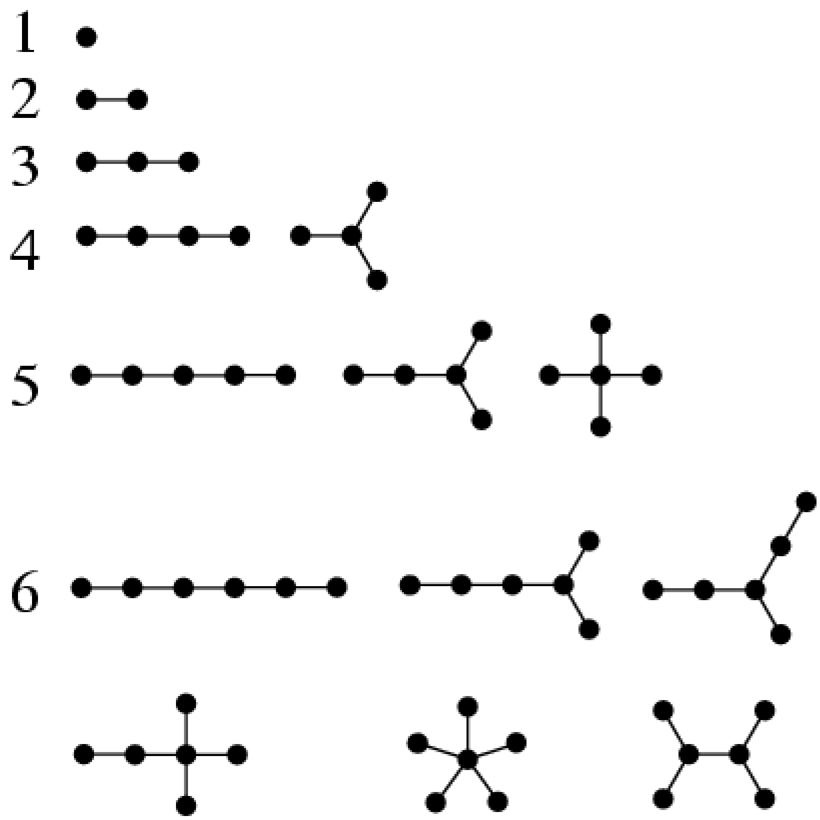

Figure 3. Dendograms of increasing order, i.e., number of nodes (image courtesy Weisstein, 1999-2003).

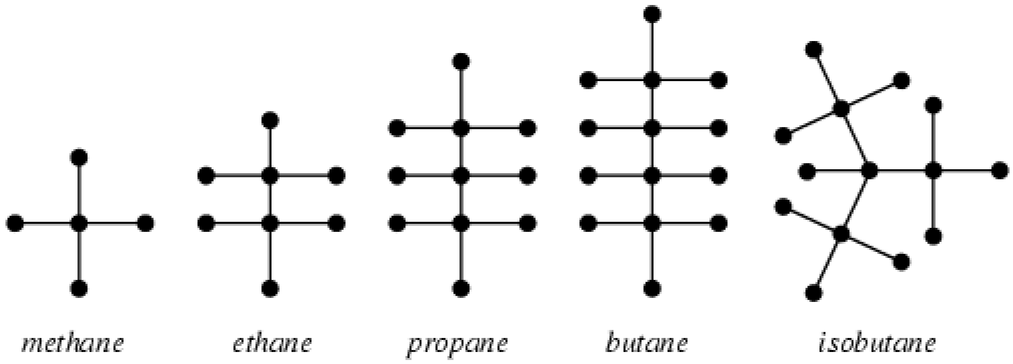

Figure 4. Molecular structures as dendograms (image courtesy Weisstein, 1999-2003).

Dendograms are especially useful in the modeling of molecular structures (Figure 4) and by analogy the relationships between spaces in a building. When the nodes of the dendogram are used as the insertion points or instantiation points of polyhedra representing spaces, dendograms can form the skeletal generating framework of the complex spatial form of a building. Dendograms can also be used to model the load-transfer action in structural assemblies. Each node represents a structural component and each link a load transfer path.

When dendograms are used to represent design processes, they become the representation of time. In this case, the dendogram is used as a state-transition graph. Each node of the dendogram represents the state of a design at a particular time. Though the state-transition graph extends in space, the spatial boundaries of the entity whose evolution is being described by the state–transition graph can be fixed. When used as a decision tree, dendograms become design decision paths in action space that are traversed in time.

A more general case of the dendogram, which is not a hierarchical tree, called a permeability map, was developed by Hillier and Hanson (1984) to represent the privacy gradient in a set of spaces.

2.2 Bipartite Graphs

A bipartite graph is a connections-based representation whose nodes can be partitioned into two sets such that no two nodes in any set are adjacent (Figure 5). In a complete bipartite graph, in addition, every node in one set is connected to every node in the other set. A tree is also a bipartite graph. Bipartite graphs are useful when the representation has two distinct set of elements that are related to each other but not amongst themselves. Bipartite graphs are also used to model a type of representation called a Petri Net, especially the channel-agency form of the Petri Net. Petri Nets are used to model hardware devices, communication protocols, parallel programs and distributed databases.

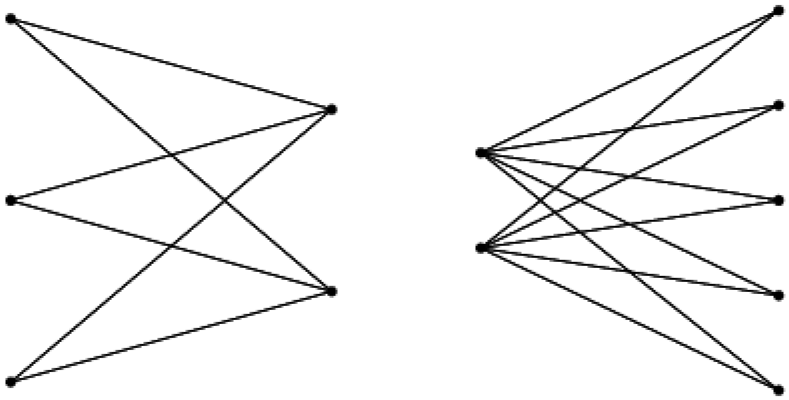

Figure 5. Bipartite graphs K $_{3,2}$ and K $_{2,5}$ (image courtesy Weisstein, 1999-2003).

The bipartite graph can be used to model a situation where a set of environmental sensors interacts with a set of architectural elements. A complete bipartite graph exhausts all the relations between the components in such as system.

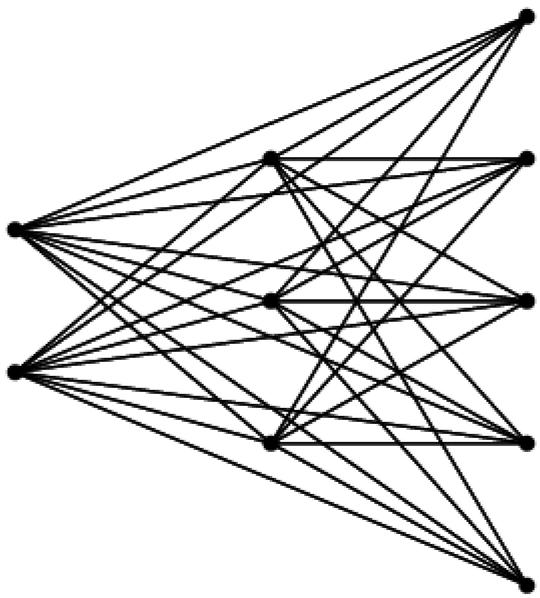

Figure 6. A 3-partite graph K $_{2,3,5}$ - an example of a k-partite graph (image courtesy Weisstein, 1999-2003).

The model of the environmental performance of an architectural space where there is a set of sources that generate environmental performance criteria, a set of receivers, that is, inhabitants who experience these environmental performance criteria, and a set of architectural elements, can be represented by a complete 3-partite graph (Figure 6) that exhausts all the relations between the various components.

Bipartite graphs can also be used as armatures for architectural designs. A notable example of the use of bipartite graphs in the arts is the use of a $K_{18,18}$ bipartite graph (Figure 7) in Umberto Eco's *Foucault's Pendulum*.

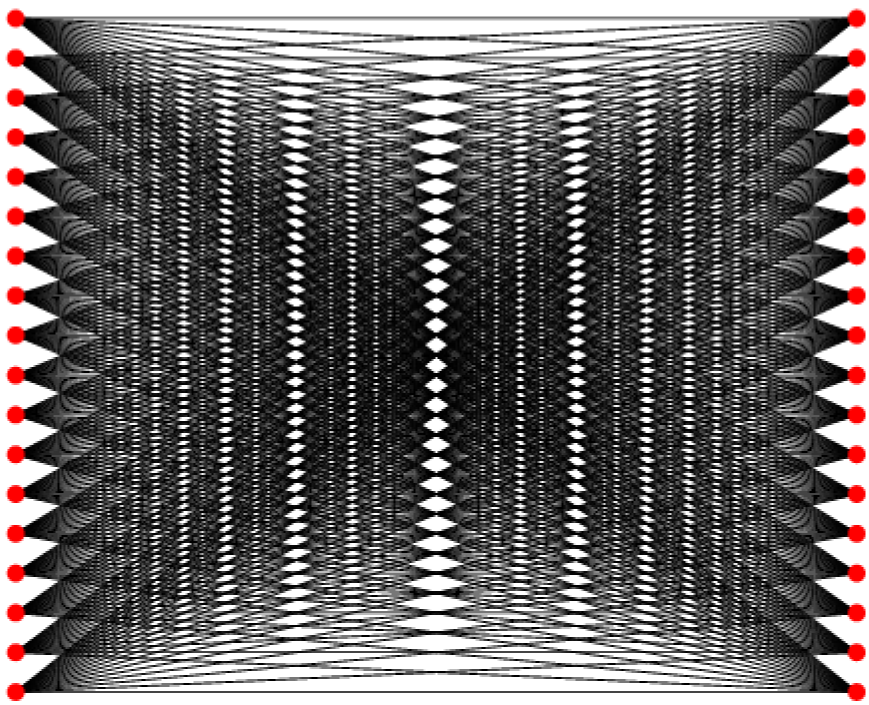

Figure 7. Umberto Eco's K $_{18,18}$ bipartite graph armature in *Foucault's Pendulum* (image courtesy Weisstein, 1999-2003).

2.3 Adjacency Graphs

In an adjacency graph, each separate space is represented as a node. Spaces that are in contact with another, that is, they are adjacent, are connected by links. In this representation, spaces that are connected only at corner points are not considered adjacent. The general exterior space is also represented as a node. All the 'interior' nodes connect to this general 'exterior' node. Adjacency graphs and their alternate form of representation, adjacency matrices, have been used in architectural design to establish proximal relations between spaces. When the duals of planar adjacency graphs are drawn, the 'wireframe' plan of a set of spaces can be generated. In a recent project (Hwang & Choi 2002), adjacency graphs were used as metadata for information retrieval in a spatial information storage system.

In an interesting analysis performed by Steadman (March & Steadman 1971), different designs by the architect Frank Lloyd Wright for different clients and sites were shown to have the same adjacency graph as the basis for the organization of spaces.

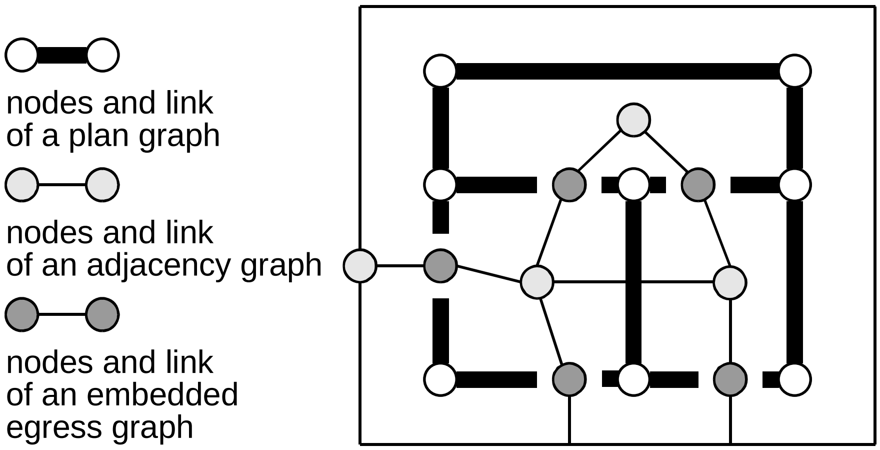

Figure 8. Plan graphs, adjacency graphs and embedded graphs.

2.4 Plan Graphs and Planar Graphs

In a plan graph (Figure 8), the junctions between walls in an architectural plan are represented as nodes and the walls themselves are represented as links. In this representation, the representation of "walls" is not restricted to physical barriers alone, but includes other divisions of space as well. A plan graph of a set of spaces is related to its adjacency graph. One is called the dual of the other.

A planar graph (Figure 8) is quite distinct from a plan graph and it is easy to be confused by the similar sounding terms. A planar graph is a graph that can be drawn on a plane without any of the links crossing each other. A completely connected planar graph (Figure 9), that is, a graph in which each node is connected to every other node, cannot have more than 4 nodes. This implies that to maintain a complete set of relations between more than four components in a connections-based representation requires three-dimensional spatial thought.

Because of the way in which a plan graph is constructed, it is always planar. Plan graphs and adjacency graphs can be integrated with other graphs, which can be embedded in them. The example shown in Figure 8 represents the modeling of an egress pattern in the floor plan of a building. Each egress element, a door or a window, is represented as a node. This node is embedded in the link between nodes that represent spaces in an adjacency graph of the plan. The egress node is also embedded in the plan graph of the floor plan. If this representation is used to simulate egress from a building during an emergency such as a fire, then traversing the graph can establish whether there is a safe egress path to the exterior of the building.

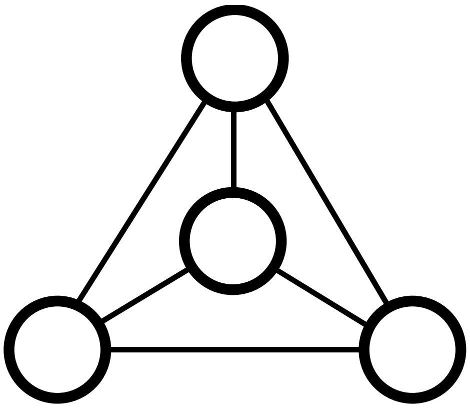

Figure 9. A completely connected planar graph.

2.5 Hasse Diagrams

According to March (1976), a Hasse diagram (Figure 10) is a diagram of connected nodes such that you can move from one node to another through a set of one or more “upward” links. As such it can be used to model “directional” synthesis of any set of entities. The Hasse diagram defines a progression from a null set to a full set of entities, where each intermediate set is a cover (a mathematical relation) of the immediately preceding set or sets. The directional buildup of an architectural design or a conceptual map that defines an architectural design as it evolves incrementally can be represented by a Hasse diagram.

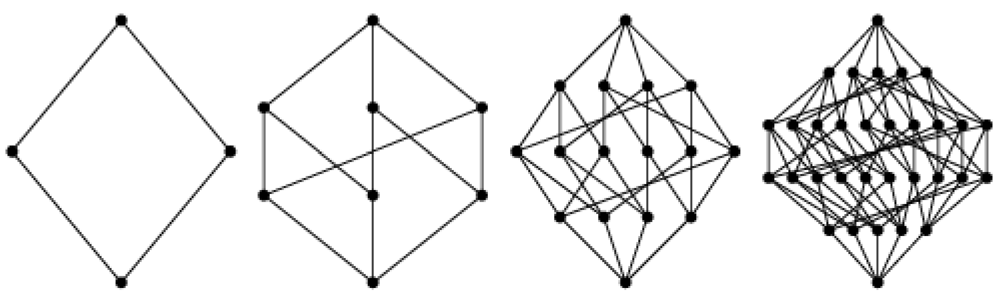

Figure 10. Hasse diagrams of sets with 2, 3, 4 and 5 entities (image courtesy Weisstein, 1999-2003).

2.6 Boolean Lattice

According to March (1976), a Boolean lattice is a representation of Boolean algebra as a complemented, distributive lattice. A Boolean algebra b(A) of a set A is the subsets of A that can be obtained by a finite number of the set theoretic operations of OR (union), AND (intersection) and NOT (complementation). Each element of b(A) is called a Boolean function. The number of Boolean functions of a set of 2 entities (say 1 and 0 as in a binary system) is 16. Computing circuitry is based on a Boolean algebra of 2 entities. The abstract structure of the Boolean algebra is isomorphic to specific algebras used in set theory, the algebra of events, symbolic logic, switching algebra and automated process control. If the sequence of design operations that generates an architectural design can be represented by Boolean lattice, then it can be automated.

2.7 Bayesian Networks

A Bayesian network (Figure 11) has been described as a “belief’ network. Each node in a Bayesian network represents the probability of a variable (a Bayesian variable) in a system. Nodes are linked to each other based on conditional dependence. The network is based on a probability model and distribution. The connections are causal connections and are directional. The direction is always from cause to effect. The dependent node is called a ‘child’ and the influencing node is called the ‘parent.’ Time is also introduced into the model as a parent is the temporal antecedent of a child. In a complex Bayesian network, many cycles of dependencies can be set up. Once a Bayesian network has been set up, a variable is given a value based on observation. Calculations are then performed to find the values of all other variables based on their probability of occurring. Once all variables are established, the network defines the probable state of the system.

A Bayesian network can be used to create a predictive model of environmental performance criteria in the design of an architectural space. Such environmental performance criteria can include temperature, illumination and sound intensity.

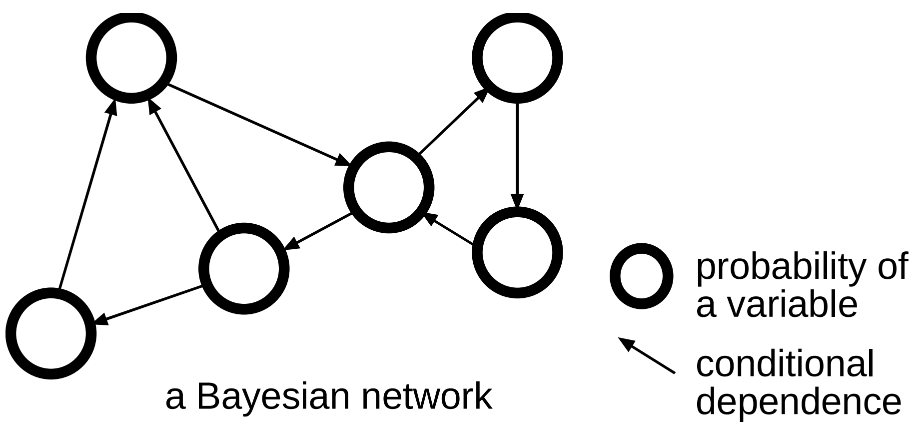

Figure 11. A Bayesian network.

Table 1: Connections-based representations and their use in the representation of

architectural design

Connections-based Representation	Architectural Representation
Dendogram	Relationship of spaces, Load-transfer in structures
Bipartite Graph	Environmental control using sensors, Modeling of environmental performance
Adjacency Graph	Relationship of spaces, Relationship of surfaces in a space
Plan/Planar Graph	Architectural floor plans, Surface distribution in spaces
Hasse Diagram	Architectural design generation using a kit of parts
Boolean Lattice	Automated architectural design generation
Bayesian Network	Prediction models for environmental performance

3 Modeling Architectural Design Using Networks

In the past, the computational representation of architecture, both as a product and as a process, has utilized a wide range of representational frameworks. Architecture has been represented as data structures, databases, procedures, algorithms and virtual computers. A connections-based representational framework now extends this range of representations to include networks (Table 1).

3.1 Elements of a Network

A network has only two basic elements, nodes and links. The complexity of networks is based on the number of nodes and their interconnections or links. When used as a representational framework, networks provide the following opportunities:

- Nodes: the representation of state (properties, variables, parameters, probability elements)
- Links: the representation of relations (constraints, semantics, physical connections, causal connections, transformations, transfer functions, dependencies)
- Networks: the representation of state, the representation of a process, the representation of probabilities of outcomes
- Network of networks: the representation of complex spatial systems

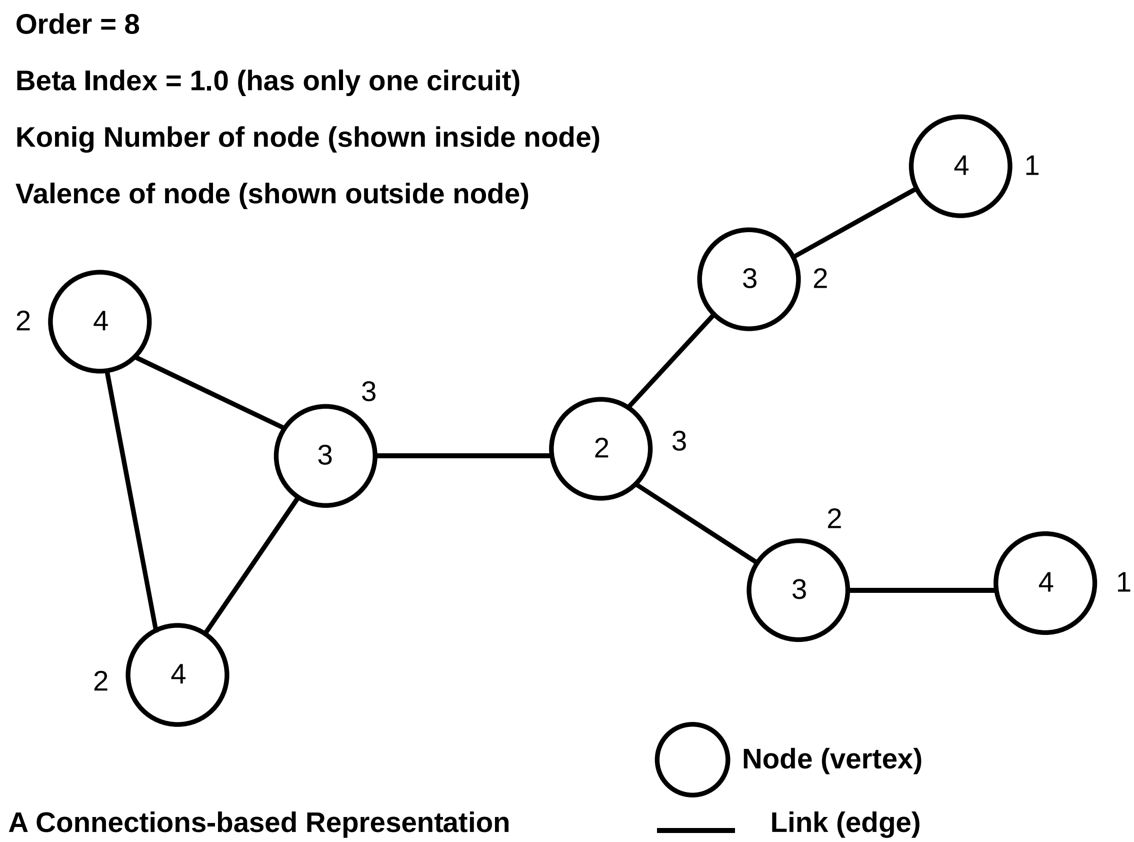

Figure 12. Properties of networks.

3.2 Properties of a Network

The various properties of networks (Figure 12) lend themselves to various representations of architecture. Some of the properties include:

Hamilton Path/Hamiltoninan Cycle or Circuit: A path in a network that starts at a starting node, goes through each node only once, is not obligated to traverse each link and ends at the starting node. It describes a circuit (Figure 13).

Euler Path/Eulerian Cycle or Circuit: A path that traverses each link of the network once, with no restrictions on the number of times it goes through a node. An Euler path is possible in a network only if the network is connected and no more than two nodes have an odd valence.

Order of the Network: The order of a network is the total number of nodes in the network.

Valence: The number of separate links to a node.

Konig Number of a Node: It is the maximum number of links in the shortest path to connect a particular node in the network to any other node in the network. The Konig number is used to establish the ‘centrality’ of node in a network.

Beta Index: It is a measure derived by dividing the number of links in a network by the number of nodes in the network. A network with a *beta* index of less than 1.0 is a tree, a network with a *beta* index of 1.0 has only one circuit and a network with a *beta* index of greater than 1.0 is a complex network.

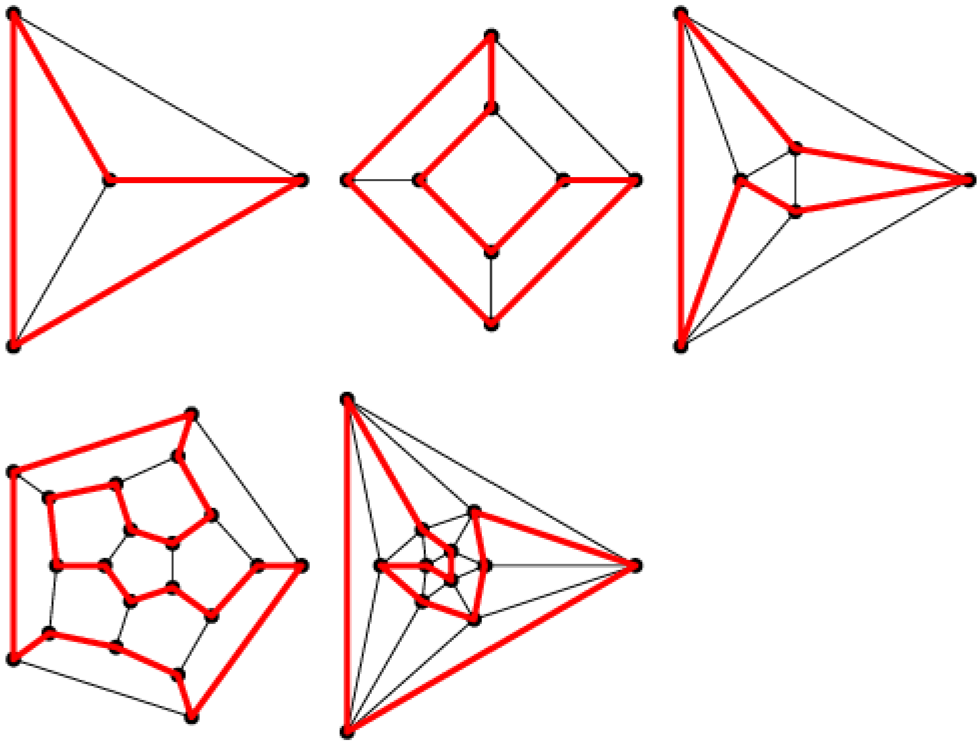

Figure 13. Hamiltonian circuits in graphs of Platonic solids (image courtesy Weisstein, 1999-2003).

4 Advantages of a Connections-based Representation of Architecture

The proposal for a non-discursive, analytical theory of architecture (Hillier 1996), may on the surface look like an erosion of a much-fought-for-and-gained political freedom in architectural expression, but in the end, may turn out to be more liberating than political freedom.

The single major advantage of using connections-based representations is the potential for distributed representation. Artificial intelligence researchers modeling the working of the brain distinguish between two kinds of representations, symbolic representations and distributed representations.

Symbolic representations use symbols such as words and numbers. These symbolic units have meanings associated with them. These units are combined into propositions in a language using the grammar(s) of that language. For example, words are combined into sentences (propositions) using the grammar of the English language. Similarly, numbers can be combined into mathematical propositions using a mathematical grammar. The main disadvantages of symbolic representations are that they are language-based and propositional; they are "brittle" and not fault-tolerant. Symbolic representations are considered "brittle" because symbolic units either exist or do not. A word is there in a sentence or it is not. Symbolic representations are not considered fault-tolerant because minor damage to a symbolic conceptual structure can cause the loss of the entire concept.

Distributed representations are ones in which meaning is not captured in a single symbolic unit, but arises from the interaction of a network of units. The common example that is given to illustrate this is that the concept "grandmother" is not stored in a single "grandmother cell" in the brain, but in a pattern or network of interacting neurons (brain cells). Distributed representations now provide the foundation for realistic computational models of human cognition related to visual, olfactory, auditory and tactile perception. Connections-based representations combine the features of symbolic representations (their structural sensitiveness) and distributed representations (their sensitiveness to statistical distributions of low-level perceptions) making them the ideal representational framework.

5 The Future of the Paradigm

The future of this paradigm lies in its ability to uncover the core of architecture, its architectonics. The architectonics of architecture may well be the architectonics of human thought. Rather than architecture being a theater of memory, through this paradigm architecture stands to be revealed as the theater of thought.

Stephen Grand OBE, a researcher from the UK and the developer of the computer game *Creatures*, has been developing an intelligent robot called Lucy. His goal is ensure that Lucy graduates from nursery school. Based on his research, Grand believes that the crucial element for intelligence is a particular circuit of neurons in the cerebral cortex of the brain that enables learning. Grand's goal is to unravel this circuit and use it to create an alternative to the digital computer that is similar to a living system. Since architecture is created by some of the most exacting neural processing known to humans, the key to this neural circuit could conceivably lie in a work of architecture. A connections-based coding, hence understanding, of this work of architecture, can lead to the discovery of this "learning" circuit.

On the other hand, Douglas Hofstadter, in his seminal book, *Gödel Escher Bach*, points out that the neural substrate of humans may pose barriers to certain processes of thought. He describes the problem encountered when someone tries to make "sense" of the Epimenides paradox thus:

> "Now my feeling is that the Tarski transformation of the Epimenides paradox teaches us to look for a substrate in the English-language version. In the arithmetical version, the upper level of meaning is supported by the lower arithmetical level. Perhaps analogously, the self-referential sentence which we perceive ("This sentence is false") is only the top level of a dual-level entity. What would be the lower level, then? Well, what is the mechanism that language rides on? The brain. Therefore one ought to look for a neural substrate to the Epimenides paradox—a lower level of physical events which clash with each other. That is, two events which by their nature cannot occur simultaneously. If this physical substrate

> exists, then the reason we cannot make heads or tails of the Epimenides sentence is that our brains are trying to do an impossible task."

Hoftstadter proposes that when confronted with a situation such as making "sense" of the Epimenides paradox, the brain encodes the paradox in the neural substrate using "symbols" and processes it using "symbolic processing." He writes:

> "Now what would be the nature of the conflicting physical events? Presumably when you hear the Epimenides sentence, your brain sets up some "coding" of the sentence—an internal configuration of interacting symbols. Then it tries to classify the sentence as "true" or "false". This classifying act must involve an attempt to force several symbols to interact in a particular way. (Presumably this happens when any sentence is processed.) Now if it happens that the act of classification would physically disrupt the coding of the sentence—something which would ordinarily never happen—then one is in trouble, for it is tantamount to trying to force a record player to play its self-breaking record. We have described the conflict in physical terms, but not in neural terms. If this analysis is right so far, then presumably the rest of the discussion could be carried on when we know something about the constitution of the "symbols" in the brain out of neurons and their firings, as well as about the way that sentences become converted into "codings."

Such limits in neural processing may be why architecture is intrinsically homeostatic, that is, it does not change in its inherent structure. It may also be why all works of architecture can be created with a simple programming language. This paradigm will reveal such limitations in architecture, if they exist. Though this sounds pessimistic, there is hope for this paradigm as revealed by Alexander. In describing the creative potential of abstract diagrams, Alexander points out that these diagrams can evolve:

> "I have discovered, since, that these abstract diagrams not only allow you to create a single whole from them, by fusion, but also have other even more important powers. Because the diagrams are independent of one another, you can study them and improve them one at a time, so that their evolution can be gradual and cumulative. More important still, because they are abstract and independent, you can use them to create not just one design, but an infinite variety of designs, all of them free combinations of the same set of patterns."

Rather than use traditional genetic algorithms (Holland 1975) for the evolution of forms, which are used to transform one population of genetic characteristics (chromosomes) into another through processes of crossover (recombination), mutation and inversion, connections-based representation lend themselves to modeling based on biological analogies such as protein synthesis and morphogenesis (from developmental biology). Just as the molecular structure of DNA "instructs" protein synthesis, connections-based representations "instruct" the creation of works of architecture. A collection of connections-based representations (architectural DNA molecules – see analogy in Figure 4) then defines the "genome" of the field of architecture. Classifying this collection of connections-based representations, thereby defining the architectural genome, is a work on the scale of Durand's *Précis.* Unlike the process followed by Durand who focused on lines and delineation, mapping the architectural genome will focus on the underlying structure or architectonics of architecture. These underlying connections-based representations are distinguished by the fact that they do not belong to the measurement-based "metric" space of architectural modulation and variation but the "invariant" space of relationships.

6 Conclusions

One can conclude with a quote from Alexander that captures the spirit of a connections-based representation of design:

> "The shapes of mathematics are abstract, of course, and the shapes of architecture concrete and human. But that difference is inessential. The crucial quality of shape, no matter of what kind, lies in its organization, and when we think of it this way we call it form. Man's feeling for mathematical form was able to develop only from his feeling for the processes of proof. I believe that our feeling for architectural form can never reach a comparable order of development, until we too have first learned a comparable feeling for the process of design."

This paper offers *making connections* as a process of design which will allow that comparable order of development. The computational representation of architecture started with the representation of architectural entities as data structures and architectural design processes as procedures. It evolved into the representation of architectural entities and processes as virtual computers. The next stage in the evolution of the computational representation of architecture is the representation of architectural entities and processes as networks of virtual computers or computational entities. The examples of connections-based representations presented in this paper suggest that various aspects of architectural design from spatial synthesis to environmental performance control can be successfully represented using the techniques. Recent work by Wolfram (Wolfram, 2002) suggests that it is possible to go even further and model the evolution of networks, which can then be used as an overarching representational framework.

References

Alexander, C. (1964). *Notes on the Synthesis of Form*. Harvard University Press.

Durand, J. N. L. (1802) *Précis des leçons d'architecture*. Paris: Ecole Polytechnique.

Hillier, B. and J. Hanson. (1984). *The Social Logic of Space*. Cambridge University Press.

Hillier, B. (1996). *Space is the machine: A configurational theory of architecture*. Cambridge University Press.

Hofstadter, D. (1980). *Gödel Escher Bach.* Vintage Books Edition.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Hwang, J. and J. Choi. (2002). SpaceCore: Metadata for Retrieving Spatial Information in Architecture. In *Proceedings of the ACADIA 2002 Conference,* pp. 199 - 217.

March, L., Editor. (1976). *The architecture of form*. Cambridge University Press.

March, L. and P. Steadman. (1971). *The geometry of environment: An introduction to spatial organization in design*. RIBA Publications Ltd.

Rajchman, J. (2000). *The Deleuze Connections*. MIT Press.

Skiena, S. (1990). *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Reading, Massachusetts: Addison Wesley.

Weisstein, E. W. (1999-2003). *Eric Weisstein's World of Mathematics* (*MathWorld*™) http://mathworld.wolfram.com/

Wolfram, S. (2002). *A New Kind of Science*. Champaign, Illinois: Wolfram Media, Inc.