Narrating Wholeness: Pattern Language Generating Semi-Lattice(s), System(s), and/or Holon(s)

David Ing, Trito Innovation Colab and Aalto University, coevolving@gmail.com

Abstract:

Does a pattern language generate into (a) whole(s)? This workshop will discuss the meaning of architecting a system, complemented with recent research from the systems sciences.

In 1967, at the formation for Center for Environmental Structure, *Pattern Manual* specified that (sub)systems are fewer in number (and implicitly larger) than patterns:

> The environmental pattern language will contain hundreds of subsystems and tens of thousands of individual patterns. Every conceivable kind of building, every part of every kind of building, and every piece of the larger environment will be specified by one or more subsystems of the environmental pattern language.
>
> In summary: An environmental pattern language is a coordinated body of design solutions capable of generating the complete physical structure of a city. The language is designed to grow and improve continuously as a result of criticism and feedback from the field (Alexander, Ishikawa, & Silverstein, 1967, p. foreword 3).

Does (and/or should) the pattern language community therefore be architecting and/or designing systems? To be clear, a subsystem is a system, with the additional property that it is contained within a larger whole.

The workshop will be conducted as a participatory session, with an intent to summarize findings for the proceedings. The workshop is organized as three steps:

- 1. Communicative Framing
- 2. Dialectical Sensemaking
- 3. Narrative Synthesizing

The extended abstract (below) outlines the workshop. Subsequent publications (i.e. a blog post) may be released after the event, to summarize some of the discussions and findings.

Keywords: *pattern language; wholeness; generative system; systems theory.*

1. Communicative Framing

The first step of appreciating the issues can be ignited by the trigger question:

- For whom, and in which situations, are (a) (sub-)system generated through pattern language?

Some contextual frames that we can share include:

- 1.1 Form and synthesis
- 1.2 Organization as semi-lattice
- 1.3 Systems generating systems
- 1.4 Generative patterns and non-generative patterns in software development
- 1.5 System-A and system-B, as two ways of shaping and building living environments
- 1.6 Holons (from systems ecology)

This non-exhaustive list of contextual frames is described in more detail, below. Additional frames may be added.

1.1. Form and synthesis

In 1964, Alexander was focused specifically on (physical) form, and on synthesis (i.e. putting parts together).

> D'Arcy Thompson has even called form the "diagram of forces" for the irregularities. More usually we speak of these irregularities as the functional origins of the form.
>
> The following argument is based on the assumption that physical clarity cannot be achieved in a form until there is first some programmatic clarity in the designer's mind and actions ; and that for this to be possible, in turn, the designer must first trace his design problem to its earliest functional origins and be able to find some sort of pattern in them. I shall try to outline a general way of stating design problems which draws attention to these functional origins, and makes their pattern reasonably easy to see.
>
> It is based on the idea that every design problem begins with an effort to achieve fitness between two entities: the form in question and its context. The form is the solution to the problem: the context defines the problem. In other words, when we speak of design, the real object of discussion is not the form alone, but the ensemble comprising the form and its context. Good fit is a desired property of this ensemble which relates to some particular division of the ensemble into form and context (Alexander, 1964, pp. 15–16).

The challenge was presented as a "design problem", and thus "problem solving", in comparison to the "problem seeking" (Peña & Focke, 1969) distinction that would come a few years later with architectural programming.

1.2. Organization as semi-lattices

In 1966, Alexander advocated organizing the city not as a tree (i.e. strict hierarchy), but as a semi-lattice (with overlaps).

> I believe that that a natural city has the organization of a semi-lattice; but that when we organise a city artificially, we organise it as a tree.
>
> *Trees and semi-lattices*
>
> Both the tree and the semi-lattice are ways of thinking about how a large collection of many small systems goes to make up a larger and complex system. More generally, they are both names for structures of sets.
>
> In order to define such structures, let me first define the concept of a set. A set is a collection of elements which for some reason we think of as belonging together. [....]
>
> When the elements of a set belong together because they co-operate or work together somehow, we call the set of elements a system. [....]
>
> You are no doubt wondering by now what a city looks like which is a semilattice, but not a tree. I must confess that I cannot yet show you plans or sketches. It is not enough merely to make a demonstration of overlap -- the overlap must be the right overlap (Alexander, 1966, pp. 47–48, 55).

Mathematically, this view is cities is based on sets. It orients towards more towards structure (as arrangement in space) than process (as arrangement in time).

1.3. Systems generating systems

In 1968, Alexander explicated an additional feature of a generative system.

> 1. There are two ideas hidden in the word system: the idea of a system as a whole and the idea of a generating system.
>
> 2. A system as a whole is not an object but a way of looking at an object. It focuses on some holistic property which can only be understood as a product of interaction among parts.
>
> 3. A generating system is not a view of a single thing. It is a kit of parts, with rules about the way these parts may be combined.
>
> 4. Almost every 'system as a whole' is generated by a 'generating system'. If we wish to make things which function as 'wholes' we shall have to invent generating systems to create them (Alexander, 1968, p. 605).
>
> [….]
>
> In a properly functioning building, the building and the people in it together form a whole: a social, human whole. The building systems which have so far been created do not in this sense generate wholes at all (Alexander, 1968, p. 605).

Beyond Christopher Alexander's direct supervision, a pattern language may or may not be be a generating system that produces systemic wholes.

1.4. Generative patterns and non-generative patterns in software development

In 1998, as pattern language was cross-appropriated into software development a distinction was made between generative patterns and non-generative (Gamma) patterns.

> What is the difference between a non-generative pattern and a generative pattern?

We observe patterns everywhere as we interact with the world around us. There are patterns in building architecture, patterns in nature, and patterns in the software people write. Recent work in software visualization is striving to bring out the patterns in software. [1] Not all these patterns are "good": for example, such research has found that people copy code from one place in a system to another, instead of generalizing the original code and reusing it in-place.

Patterns that we observe in a system that has already been built are non-generative, and are sometimes called *Gamma patterns*, after the pioneering work of Erich Gamma. [2] Such patterns are descriptive and passive.

That we can find these patterns doesn't imply any rationale behind them, and not every pattern leads to desirable results. What we want to do is capture the patterns that are ''good,'' and codify them, so people can use them when building systems. These patterns *generate* systems, or parts of systems. We can observe the patterns (in other words, see their structure or their effects) in the systems they generate. Patterns designed to shape system architectures are called *generative patterns*. They are prescriptive, and active (Coplien, 1998, p. 312).

[1] Church, Kenneth Ward, and Jonathan Isaac Helfman. "Dotplot: A Program for Exploring Self-Similarity in Millions of Lines for Text and Code." *Journal of Computational and Graphical Statistics* (1993), Vol. 2, No. 2, pp. 153-174.

[2] Gamma, Erich. ''Design Patterns -- Abstraction and Reuse of Object-Oriented Designs.'' in *Proceedings of the European Conference on Object-Oriented Programming*, Oscar Nierstrasz, ed. Berlin: Springer-Verlag, 1993.

More generally, this might be described as a distinction between "architectural patterns" and "design patterns". This mixes with the distinctions presented as experiential intentions occurring first in the imagination, and secondly in the physical built environment (Pallasmaa, 2015, pp. 11–12), and between autopoiesis (reproduction of the self) and allopoiesis (reproduction of the other) (Schumacher, 2011).

1.5. System-A and system-B, as two ways of shaping and building living environments

In his last book, Alexander completed the history on the process of constructing the Eishin school between 1981 and 1989.

Imagine a town of type "A" -- a neighbourhood if you like .... Because of the depth and scope of its structure, this world is almost infinite in its richness.

Compare this imaged town with the more usual neighbourhood of type "B", typical of modern property development, where this is a stale and ugly air of repetition. [.... What] we feel is something flat, without excitement, without the urgent joy of life.

These two kinds of places, then, A and B, are typically generated in two different ways. We may therefore call these two different generating systems A and B. The first system, A, whether large or small, is fresh in imagination, generated by an infinite horn of plenty. [....] The second system, B, is oppressive. [....] The system that generates this tiredness-inducing structure, we call system-B.

[....] Describing an ongoing dispute between two fundamentally different ways of shaping and building our living of environment, each of the two "ways" may be regarded in turn as a production system, a system of thinking, a system of how to plan and build, how to organize labor and craft, how to take care of land, and money, and how the people who live and work in the environment may nourish their relationship of belonging to the land itself. [....]

> System-A is concerned with the well-being of the land, its integrity, the well-being of the people and plants and animals who inhabit the land. As we shall see throughout this book, this has very much to do with the integral nature of plants, animals, water resources, and the tailoring of each part of every part to its immediate context, with the result that the larger wholes, also, become harmonious and integral in their nature.
>
> System-B is concerned with efficiency, with money, with power and control. Although these qualities are less attractive, and less noble than the concerns of system-A, they are nonetheless important. They cannot be ignored. If we are travelling in an airplane, or a high-speed train, we shall often be very glad that this system is constructed under the guidance of some version of system-B (Alexander, 2012, pp. 10–11).

The generation of an environment (i.e. which can be conceived as containing a system) is a fine distinction in the wordings.

1.6. Holons (from systems ecology)

In 21st century rise of systems ecology (e.g. resilience science, panarchy), there has been a refinement of systems theory -- across hierarchy theory and network theory -- on holons.

> Holons are central to both hierarchy and network theory with their models and narratives. We will show a unity between the two theories by invoking holons as we compare and contrast. [....]
>
> The holon is a fundamental concept in hierarchies. It captures the dualities that appear in hierarchical complexity by emphasizing that the holon is at once an autonomous whole, while also being subsumed as a part in some upper level structure. The holon itself is the skin that partitions inside from outside. It integrates signal from the outside environment, as signal moves into the holon. In the other direction, the holon integrates the parts, so the organs cannot be seen, except as they contribute to the whole organism. Organs and organisms are only examples here, because wholeness and partness are general principles (Allen & Giampietro, 2014, pp. 33–34).

The "upper level structure", or containing whole is consistent with a portrayal of systems thinking where synthesis precedes analysis (Ackoff, 1981, pp. 16–17).

Christopher Alexander saw the pattern language of the 1970s with three essential features: (i) it has a moral component; (ii) it aims for morphological coherence in the things made with it; and (iii) it is generative, with a process that allows people to create coherence and morally sound objects (Alexander, 1999, p. 74). At that time, he was unsure if the these features had yet been translated into the discipline of software development.

2. Dialectical Sensemaking

The second step of elevating or lowering expected benefit of an option can be discussed with a trigger question:

- In systems architectures and systems designs, why would (i.e. which values would be afforded) by the development of pattern language(s)?

Three dialectics are proposed for dialectical sensemaking:

- 2.1 Types of systems and models
- 2.2 Autopoiesis and allopoiesis
- 2.3 Economies as agricultural, industrial and services (coproduction)

These dialectics could be discussed separately, or in interplay.

2.1. Types of systems and models

Ackoff's original research was in purposeful (i.e. ideal-seeking) systems, so there should be no surprise to see them presented that way.

> There are three basic types of systems and models of them, and a meta-system: one that contains all three types as parts of it (see Table 1):

Table 1: Types of systems and models

Systems and models	**Parts**	**Whole**
Deterministic	Not purposeful	Not purposeful
Animated	Not purposeful	Purposeful
Social	Purposeful	Purposeful
Ecological	Purposeful	Not purposeful

> (1) Deterministic: systems and models in which neither the parts nor the whole are purposeful.
>
> (2) Animated: systems and models in which the whole is purposeful but the parts are not.
>
> (3) Social: systems and models in which both the parts and the whole are purposeful.
>
> These three types of systems form a hierarchy in the following sense: animated systems have deterministic systems as their parts. In addition, some of them can create and use deterministic systems, but not vice-versa. Social systems have animated systems as their parts.
>
> All three types of system are contained in ecological systems, some of whose parts are purposeful, but not the whole. For example, Earth is an ecological sysetm that has no purpose of its own but contains social and animate systems that do, and deterministic systems that don't (Ackoff & Gharajedaghi, 1996, p. 14).

As these models of systems are mixed with each other, the typology gets more complicated.

2.2. Autopoiesis and allopoiesis

The definition of autopoiesis originated from biology, and has been cross-appropriated (controversially) to knowledge management. Many who use that term are not familiar with an associated term, allopoiesis.

Allopoiesis is ...

> "the production by a network of interrelated component-producing processes of a system, which does not however become able to thereafter reproduce its components or processes. ... [If] the allopoietic system is really to be a system, it must at the same time be autopoietic in order to maintain its identity and coherence. This would be possible if we admit that the boundaries or other subsystems transform inputs into internally fitting elements ... while producing outputs by an inverse transformation" (François, 1997, p. 24).

Autopoiesis has also been cross-appropriated to built environments.

> An autopoietic system of architecture cross disciplinary lines. The concept of order proposed here – encompassing both social and architectural order – denotes the result of the combined effort of organization and articulation. Architectural order – symbiotic with social order – requires both

> spatial organization and spatio-morphological articulation. While organization establishes objective spatial relations by means of distancing (proximity relations) as well as by means of physically separating and connecting areas of space, articulation operates via the involvement of the user's/participant's perception and comprehension of their designed/built environment. Articulation reflects the phenomenological and the semiological dimensions of architecture. Thus, to the extent to which architecture operates through articulation (rather than mere organization), it also relies on engendering an effective semiosis within the designed/built environment. It is one of the fundamental claims of the theory of architectural autopoiesis that the semiological dimension of architecture is of central importance with respect to architecture's capacity to successfully discharge its unique societal function" (Schumacher, 2011, pp. 371–372).

These ideas may open up the appreciation of the generative and non-generative.

2.3. Economies as agricultural, industrial and services (coproduction)

In the 21st century, the world first dominated by the agricultural economies (following solar and lunar cycles) and industrial economies (driven by supply chain machinery) has seen the rise of value-creating systems (i.e. coproduction).

> With the coming of industrialization, the centre of gravity of the critical assets shifted from natural resources, negotiated privilege, and geography to mastery of production technology and capital to back it up. Newton's theories had become translated into mechanical (and later electromechanical) technology. We were a step closer to the perfect, engineerable world. The invention of the engine liberated production from the site of the source of the energy. Craft production could turn into mass production with standardized, specialized tasks. [....]
>
> The last few years again have brought a shift into the new era leading to a new strategy paradigm. As in the Industrial Revolution, the driving force again is new technology, notably -- so far -- information technology. [...]
>
> Out of these opportunities emerges a new archetype of the organization: The business company as an organizer of value creation. The critical competence of business companies today is exactly that: the competence to organize value creation. This does not mean that production competences or relationship competence are unimportant, but that such competences of organizing value creation far beyond their formal boundaries. [...]
>
> The new paradigm -- which for the moment I will call reconfiguration of value-creating systems -- also implies a dramatic conceptual change and a real shift in how we view customers. The customer is no longer just a receiver, no longer just a source of business, but now actually a co-producer and a co-designer of value creation (Normann, 2001, p. 18,24).

As most of the attention in the world has focused on industrialization, there is some complementaries with Alexander's system-A and system-B distinctions.

3. Narrative Synthesizing

The third step is explores balancing action across:

- (i) producing outcomes;
- (ii) designing for learning to deal with the variety of (phenotypic) responses within the current appreciation of the environment ; and

- (iii) architecting for learning to deal with the variety of (genotypic) environnments that have not been anticipated.

Being aligned and efficient today, while being adaptive enough for changes in the environment is the principal idea of the ambidextrous organization (March, 1991). This may be further refined to differentiate structural ambidexterity (managing tradeoffs by putting dual structures into place, some focused on alignment while others focus on adaptation), and contextual ambidexterity (the behavioural capacity to simultaneously coherence working of activities working towards the same goals, and adaptability as a capacity to reconfigure activities to meet changing demands) (Gibson & Birkinshaw, 2004).

4. References

Ackoff, R. L. (1981). *Creating the Corporate Future: Plan or Be Planned For*. New York: John Wiley and Sons.

Ackoff, R. L., & Gharajedaghi, J. (1996). Reflections on Systems and their Models. *Systems Research*, *13*(1), 13–23. https://doi.org/10.1002/(SICI)1099-1735(199603)13:1<13::AID-SRES66>3.0.CO;2-O

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

Alexander, C. (1966). A City is Not a Tree. *Design*, *206*, 46–55.

Alexander, C. (1968). Systems Generating Systems. *Architectural Digest*, *38*. Republished in *Computational Design Thinking*, edited by Achim Mengers and Sean Ahlquist (2011), pp. 58-67. Originally published in *Systemat*, a journal of the Inland Steel Products Company.

Alexander, C. (1999). The origins of pattern theory: The future of the theory, and the generation of a living world. *IEEE Software*, *16*(5), 71–82. https://doi.org/10.1109/52.795104

Alexander, C. (2012). *The Battle for the Life and Beauty of the Earth: A Struggle Between Two World-Systems*. Oxford University Press.

Alexander, C., Ishikawa, S., & Silverstein, M. (1967). *Pattern Manual*. Berkeley, California: Center for Environmental Structure.

Allen, T., & Giampietro, M. (2014). Holons, creaons, genons, environs, in hierarchy theory: Where we have gone. *Ecological Modelling*, *293*, 31–41. https://doi.org/10.1016/j.ecolmodel.2014.06.017

Coplien, J. O. (1998). Software design patterns: Common questions and answers. *The Patterns Handbook: Techniques, Strategies, and Applications*, 311–320.

François, C. (1997). *International encyclopedia of systems and cybernetics*. Munich: K. G. Saur.

Gibson, C. B., & Birkinshaw, J. (2004). The antecedents, consequences, and mediating role of organizational ambidexterity. *Academy of Management Journal*, *47*(2), 209–226. https://doi.org/doi.org/10.5465/20159573

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, *2*(1), 71–87. https://doi.org/10.1287/orsc.2.1.71

Normann, R. (2001). *Reframing business*. New York: Wiley.

Pallasmaa, J. (2015). Empathic and Embodied Imagination: Intuiting Experience and Life in Architecture. In J. Tidwell (Ed.), *Architecture and Empathy* (Vol. 3, pp. 4–19). Espoo, Finland: Tapio-Wirkkala - Rut Brut Foundation.

Peña, W. M., & Focke, J. W. (1969). *Problem Seeking: New directions in architectural programming* (1st ed.). Houston, TX: Caudill, Rowlett and Scott.
Schumacher, P. (2011). *The Autopoiesis of Architecture: A New Framework for Architecture*. Wiley.

5. About the author:

David Ing is a cofounder and systems change researcher at Trito Innovation Colab. He is completing his Ph.D. in Industrial Management at Aalto University. In 2017/2018, doctoral research from those studies was published as Open Innovation Learning: Theory building on open sourcing while private sourcing. In 2012, he graduated from IBM after 28 years of service. A resident of Toronto, Canada, he is frequent traveller and most readily found on the Internet.