Introduction to Complexity and Applied Complexity, Spring 2021

Module 18 — The Timeless Way

Notes by Sav Sidorov

Readings

- Christopher Alexander — The Timeless Way of Building (chapters 1 & 2)
- Christopher Alexander — A Pattern Language (introduction)

An Introduction to Christopher Alexander

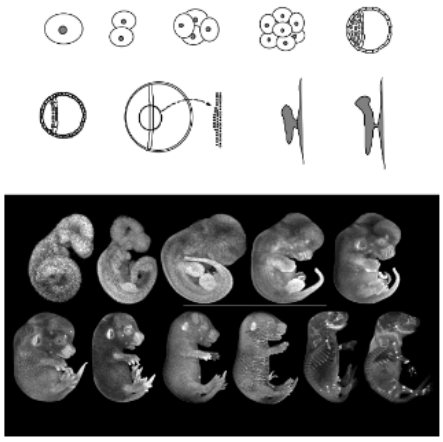
What does it mean to build something in an organic way? That's the central question we'll be exploring in this module.

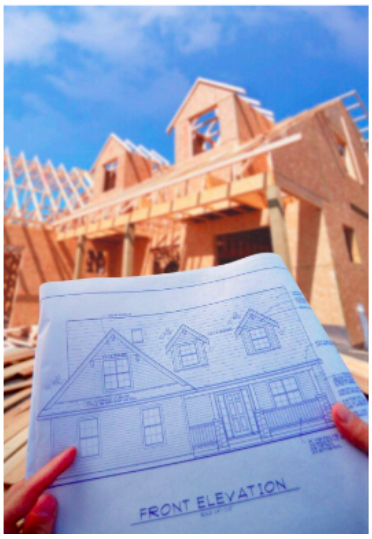
There are two readings, both from Christopher Alexander — *The Timeless Way of Building* and *A Pattern Language*. These two books form a coherent whole, even though they were released as separate volumes.

Alexander speaks in a way that can appear esoteric, but the more you dig into it, you begin to realize why he's using the language that he uses. There's a weird inversion that happens: esoteric language can map to something highly abstract, but also to something highly practical. In a practical setting, systems usually have very high dimensionality, and discussing the higher order relations that arise turns out to be very difficult. Alexander takes these implicit relations and makes them explicit by presenting things in this esoteric way.

In the introduction to A Pattern Language, Alexander lays out the fact that when using a pattern language, there's this logical progression of scales that one has to work down — from larger scales down to the smaller scales.

What's the underlying motivation for discussing these ideas? What it boils down to is: How can we reconcile building and growth? How can we reconcile self-organization and evolutionary processes with actually taking materials and forming some physical structure?





Let's go back to the juxtaposition of fetus development and the blueprint process for constructing buildings. In our modern way of thinking, we typically see construction as being almost synonymous with this blueprint planning process — plan out all the details, and then build.

The question to ask is: is there a way to build things that seems more like a growth process? It seems antithetical, but Alexander's whole body of work has centered around this notion, and proving that it's possible. And not only that it's possible, but actually superior.

The blueprint process is a very particular process that we've gotten into the habit of doing. But in fact, throughout much of human history, nothing was created by blueprint. It was created in the way that Alexander describes.

Alexander is a practicing architect, and all of his discussions are housed in architecture. But really, his project is much larger than that field and touches on many more domains — namely software development and permaculture.

We've seen that in a centrally-controlled system, the bandwidth gets so constrained that there's a very hard upper limit to the amount of complexity you can generate. If we want to go above that bandwidth constraint, we need a new approach — one that represents the system in its entirety. We need a way to generate things that are more complex than any one person can hold in their head. That's the problem that Alexander is tackling.

People as "Inter-Actors"

Here's one way to look at the reconciliation of building and growth. In the building process, humans become the "inter-actors", as we'll call them. They are the sensors and effectors that allow the forces in the space to communicate and affect one another, and eventually allow those forces to resolve such that there's no more tension or inconsistencies in the system.

Instead of thinking about people as designers and planners, we can think of them as the medium through which the materials of the system interact. The materials themselves are not living, perceiving agents, so they can't self-organize. But if the people can serve as that medium of interaction, then there can be a kind of self-organizing process that occurs.

The Quality With No Name

In the opening chapters of *The Timeless Way of Building*, Alexander begins by discussing the thing that we're trying to achieve by building this way. He calls it **the quality with no name**.

The reason it has no name is not because it's so abstract that you cannot name it. Rather, it's so concrete — so specific — that no word can capture its meaning perfectly. Words, by their very nature, are inexact and leave room for misinterpretation. Words come with baggage that misleads.

In this design process, it ultimately depends on the ability of humans in the system to be the selection mechanism — to select the possible configurations of the system that make sense. To do that, they need to be sensitive to the quality with no name.

That being said, Alexander tries to come close to the meaning by saying a few things:

> *"It is never twice the same, because it always takes its shape from the particular place in which it occurs."*

He considers the words **alive**, **whole**, **comfortable**, **free**, **exact**, **egoless** and **eternal**. Each comes close to the quality with no name. Each fails to hit the mark.

He uses the word *alive*. All we have to do is go to a standard corporate office building, or public school, to see that some places can be *dead*. If places can be dead, then they can also be *alive*. But of course, these things aren't *literally* alive — it's simply a metaphor we use. Given that it's a metaphor, it doesn't strike at the heart of this quality.

What about *whole*? These systems are self-consistent and resolved at each scale. Everything works together and has its right place. Yet *whole* can give a sense of a self-contained entity, and these entities are constantly interacting with their surroundings.

Comfortable? This word is also filled with misconceptions. We don't mean *comfortable* in the sense of being lazy, but in the sense of lacking inner contradictions. *"It is too easy to use the word for situations which have no life in them because they are too sheltered."*

Free? Things that have the quality with no name have a freedom to act, to not be held back. But the problem here is that *free* implies an arbitrariness — "anything goes." Yet what we see is the opposite — the quality with no name is very specific. *"A building which has a "free" form — a shape without roots in the forces or materials it is made of — is like a man whose gestures have no roots in his own nature."*

Exact? *Exact* in the sense that — from small details to high-level forces and flows — these things are responding and adapting to these details. Not *exact* in the sense of conforming to some image or representation.

Egoless is another word we can use. Not making some kind of big statement. Nowadays, *ordinary* things are very rare. Not that egoless buildings can't reflect the personality of their builders. Simply that the personality shown should be there to enhance the beauty and aliveness of the place, not to enhance the careers of the builders. So there's confusion that comes with using *egoless* as well.

Eternal, then. Both in the sense of something so strong and balanced that it's imperishable, and in the sense of it being timeless. Archetypal. *"They reach into the realm of eternal truth."* Yet eternal hints at a religious, mysterious, extraordinary aspect that the *quality with no name* doesn't have. *"[The pond] is above all ordinary."*

> *"The quality with no name includes these simpler sweeter qualities. But it is so ordinary as well, that it somehow reminds us of the passing of our life. It is a slightly bitter quality."*

Look at Acorn Street in Boston — it captures the *quality with no name* very well.

Tradition

One other comment here, about tradition. Alexander is an advocate for tradition. Yet it's important to note that tradition isn't good for its own sake. Whenever you build something, there's a set of problems that you have to solve. Tradition has tinkered its way to nice solutions to those problems. There are many hidden rakes that someone attempting to innovate will inevitably step on — new things are not guaranteed to work. In fact, there is a very high probability that they will fail. By the way, we're not saying that tradition is always good, just that there is certainly a function that it serves. In other words, tradition is the solution to a bunch of problems that you don't know you have until you remove the tradition.

G. K. Chesterton:

> *"In the matter of reforming things, as distinct from deforming them, there is one plain and simple principle; a principle which will probably be called a paradox. There exists in such a case a certain institution or law; let us say, for the sake of simplicity, a fence or gate erected across a road. The more modern type of reformer goes gaily up to it and says, 'I don't see the use of this; let us clear it away.' To which the more intelligent type of reformer will do well to answer: 'If you don't see the use of it, I certainly won't let you clear it away. Go away and think. Then, when you can come back and tell me that you do see the use of it, I may allow you to destroy it.'"*

A Pattern Language

What is *A Pattern Language* all about? Alexander presents one instance of a pattern language in his book, but it is by no means the only one.

Every pattern language is made up of **patterns** — they serve as the "atoms" of structure. What are patterns? Patterns describe the relationships between things — how things interact with one another.

How the pattern is fulfilled is always unique. No two instantiations are exactly the same, because no two situations are exactly the same — patterns are sensitive to the context.

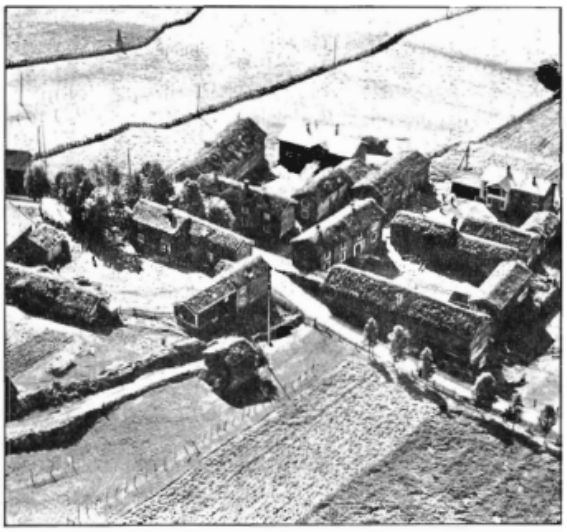
Patterns also interact with other patterns, both "vertically" (at different scales) and "horizontally" (at the same scale). Patterns are inherently a **multiscale** and **relational** concept.

Examples

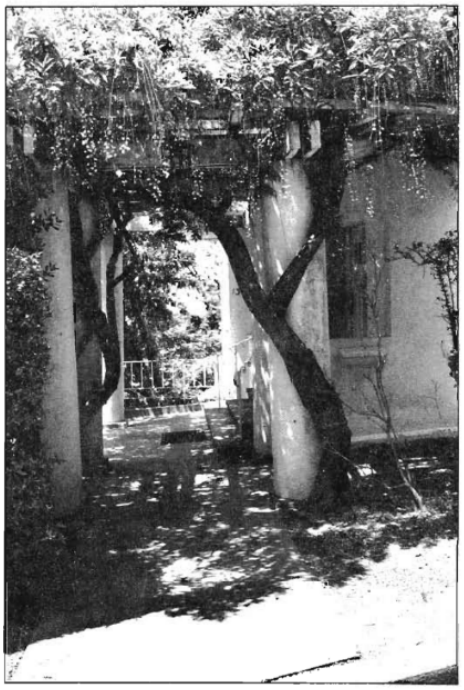
A few concrete patterns to get a sense for patterns:

105 SOUTH FACING OUTDOORS**

106 POSITIVE OUTDOOR SPACE**



112 ENTRANCE TRANSITION**



171 TREE PLACES**

Pattern Language as a "Gate"

Alexander talks about the pattern language as a "gate" — an entrance point into the timeless way of building.

The goal of a pattern language is not to innovate something "new", but to spark something very old to us. It's a way of building that is very different from the blueprint process. It may involve no drawings at all.

The timeless way of building involves no professional architects. Take a look at these examples:

Each building is unique, yet it builds on a common theme. The construction process here is very organic, far from some kind of blueprint or central planning approach.

Coarse-to-Fine

Patterns are nested within other patterns. The application of a pattern to a given space has a scaling logic that moves from coarse- to fine-grained patterns.

The larger-scaled patterns become, essentially, constraints on the smaller-scale patterns. Notice that you can't do it the other way around. Start fuzzy, then get down into the specifics.

It's also important to mention that the scales between two patterns shouldn't be so different that they don't interact. Interaction between scales is crucial to bringing about this interesting coarse-to-fine cascade.

There's a way in which we might call the coarse-to-fine process *top-down*. But we don't mean top-down as in centrally controlled. To avoid confusion with terminology, we won't use top-down to describe this process.

Barcelona is a great example of the coarse-to-fine approach. At a large scale bird's eye view, we see a very regular structure, imposed by a constraint from the city government.

Yet if we zoom in, we notice a lot of irregularity in each city block. Each block has a rich inner structure.

So what we see is a top-down constraint, followed by organic growth within each block.

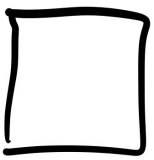
The Process

What we have so far: a quality without a name, and a pattern language that acts as the "gate", allowing this quality to be instantiated. But how do we go about that?

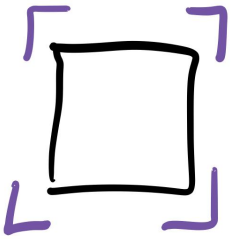
Our goal is to construct and organize a space such that all forces in the space are resolved — such that there's no inner tension, no friction. However, that's much easier said than done. The forces in the real world are myriad and multiscaled. It takes time and attention to identify and resolve them all.

Structure-Preserving Transformations

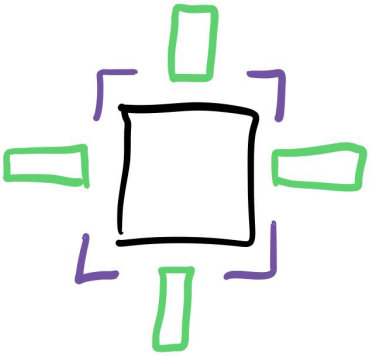
Another idea that Alexander presents — in his book *The Nature of Order* — is the notion of **structure-preserving transformations**. These are simple, iterative processes for modifying geometric shapes, intended to build "life" in these shapes.
Let's demonstrate this by starting to draw a square:

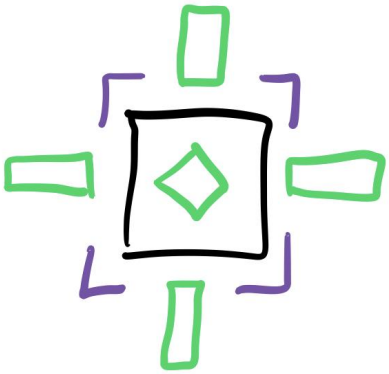


Next, we can elaborate on this structure in a way that preserves and reinforces it:



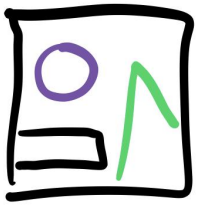
And keep doing this iteratively:





Of course, there's no one way to do this, but you have to respect the geometry inherited from the previous iteration.

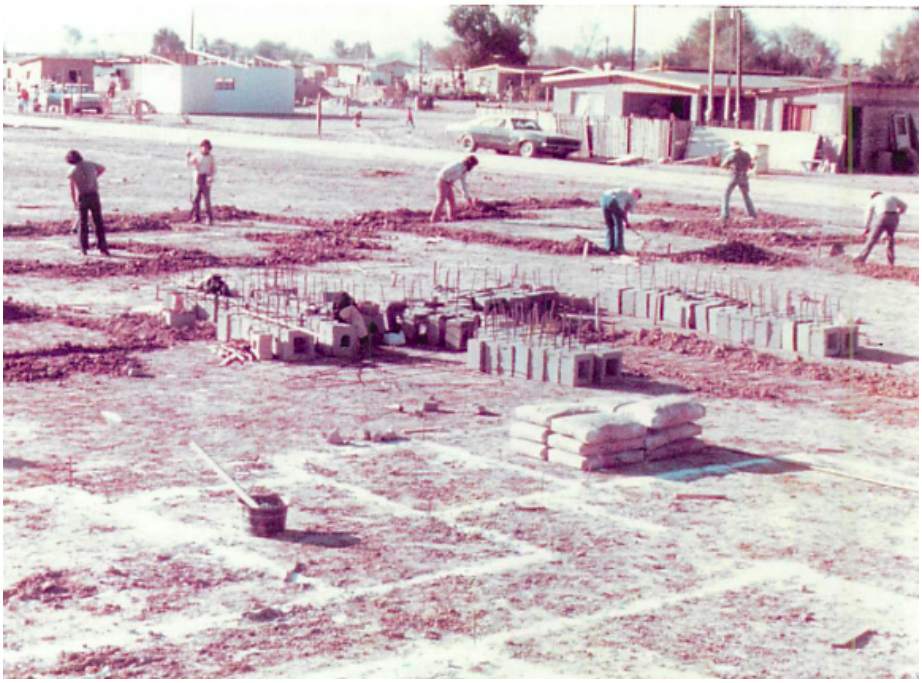
Alternatively, let's say you start randomly adding details to your square:



And now you have something that looks like modern art — something that's not sensitive to the given structure, or almost explicitly anti-sensitive to it, which is not what we want.

The process of applying structure-preserving transformations looks like a step-by-step adaptation process. Imagine you're trying to flip a set of coins and make them all land on heads. If you take an all-or-nothing approach, throwing all coins at once, and you do this over and over, it's going to be quite a while before you get them all to land on heads simultaneously. On the other hand, if you flip each individually, you can get them all to

land on heads relatively quickly. Iterative evolution is better than trying to get everything to work all at once. Using this metaphor in a practical context, look at these builders:

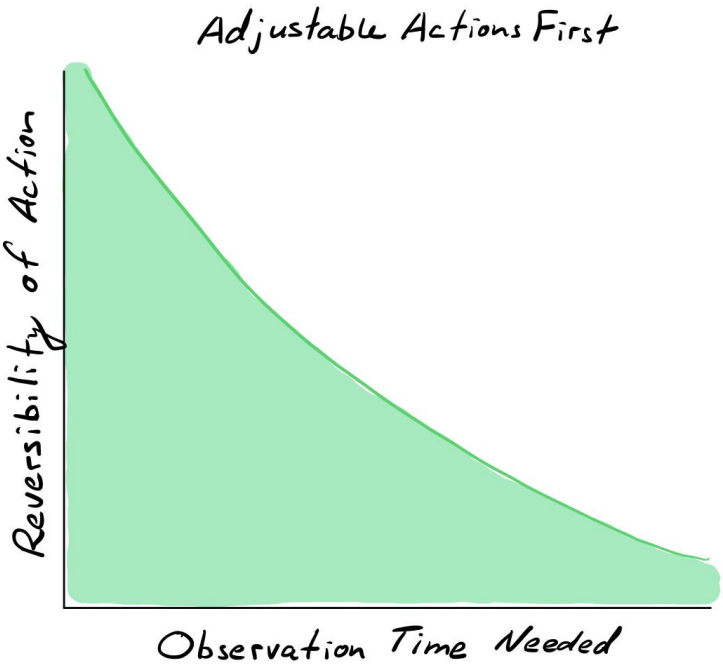


They're planning out the building boundaries in the context of the place in which they'll be building — an iterative process. *"Design and construction are integrated."*

Reversibility in Exploration

As you're implementing these ideas, one of the most important things to consider is the reversibility of the actions you're taking. How easy is it to reverse a decision? As you're tinkering with a design, you want to make sure that the decisions you're making are reversible.

A graph from Ben Falk (from his book *The Resilient Farm and Homestead*) that demonstrates this:



The less time you need to observe the outcome of your decision, the easier it is to reverse. In other words, the tighter the feedback loop between action and outcome, the easier it is to tinker and figure out what works through iteration. Chase tight feedback loops.

In addition, by making reversible decisions in an environment, you discover the way that things interact *in the actual system* instead of an artificial testing environment. Mock up the structure, not the environment. As you become more sure of the decisions you've made, you can replace the easy-to-reverse mockups with more permanent solutions.

A heuristic approach: notice those little points of friction in your physical space as you go about your day. If you take a moment and resolve those frictions, little by little, a place becomes more fluid and thereby more whole.

One-shot vs Piecemeal Growth

One other useful mental model is **one-shot vs piecemeal growth**. If you're building a house, for example, that's a very one-shot thing — you can't build half of a house. On the other hand, if you're building a street or a neighborhood, that's more of a piecemeal activity — you can build it in parts.

Take this set of buildings, for example. Each building was built in a one-shot manner, but the street is piecemeal. And yet, they all cohere and share the same aesthetic. They come from the same tradition and were built using the same pattern language.

Whole to Parts

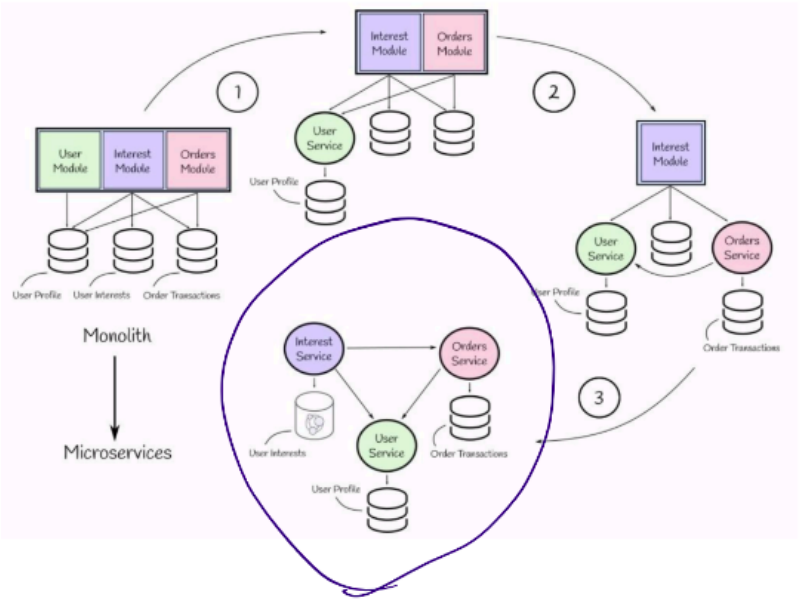
This is another design pattern common to biology, and one that's been successfully applied in engineering. Let's take an example from software development.

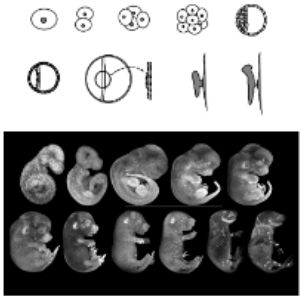
In software, there's a difference between microservices and a monolith. Microservices are, as the name suggests, small, isolated services that do different things (store data, perform queries, etc). If you change something in a microservice, because it's isolated, it doesn't have ripple effects across the entire system. A monolith is a structure where

all the pieces of the system are sharing data stores, are all together and depend on each other.

What you generally want to do is start with a monolith — a tightly integrated whole — that slowly differentiates into microservices. The point being: plan out the general structure first, kind of like a scaffold, and delve into the details later. You can't plan it all at once. Therefore, use a coarse-to-fine approach.

Here we see the monolith → microservices model contrasted with embryonic development:





15 Properties

Later in Alexander's work, in *The Nature of Order*, he breaks down these patterns even further. What are the properties that make up a pattern? To come back to the atoms analogy, these properties would be the quarks. He's identified 15.

1. Levels of scale

Something we've talked about quite frequently.

2. Strong centers

   Centers that are easily identifiable; unambiguous.

3. Boundaries

   Very essential for showing you where a thing is and isn't — a way to differentiate the inside from the outside. The modern mind tends to think of boundaries as tantamount to isolation, but it's actually the other way around. You need appropriate boundaries that allow that unit to have interactions with other things in a constructive and harmonious manner. The lack of boundaries doesn't make a system more integrated — it just makes it more of a disorganised mess.

4. Alternating repetition

   Not a simple mechanical repetition, but something more subtle.

5. Positive space

   We saw that space should generally be positive — as much convexity as possible.

6. Good shape

   "The good shape is an attribute of the whole configuration, not of the parts; but it comes about when the whole is made of parts that are themselves whole in this rather simple geometric sense."

7. Local symmetries

The opposite of large-scale symmetries. The local structures should have symmetry, but the total structure doesn't necessarily have it. In fact, it feels more organic when it doesn't.

8. Deep interlock and ambiguity

   A yin yang type of idea. There are boundaries, but also deep integration between them — they're reaching into one another.

9. Contrast

   Self explanatory. Recall Turing Patterns, that's a great example of contrast.

10. Gradients

    When talking about contrast, gradients are usually also present.

11. Roughness

    This isn't the roughness of fractals, but the roughness that comes with the structure of the system being extremely sensitive to the local forces on it, such that things that you might think of as imperfections are not that — they're actually adapting to local situations, and break away from an abstract, platonic notion of perfection.

12. Echoes

    Whatever the repeating themes are in the system, they repeat in multiple places, at different scales.

13. The Void

    The idea that you need some empty space, otherwise everything looks busy.

14. Simplicity and Inner Calm

    "The quality comes about when everything unnecessary is removed. All centers that are not actively supporting other centers are stripped out, cut out, excised. What is left, when boiled away, is the structure in a state of inner calm. It is essential that the great beauty and intricacy of ornament go only just far enough to bring this calm into being, and not so far that it destroys it."

15. Not-separateness

    Probably the most important property. It's the idea that things respond to one another. They're not built in a vacuum, with a blindness to what's around them.

Complete Reorientation

This whole project requires a complete reorientation about how we think about objectivity, and what we take to be objective.

It calls to trust our intuition, instincts and feelings. To take things holistically, and only then begin to abstract and explain the "reasons" for the feeling.

Our perception is not centralized, and we have subtle reactions to unresolved forces that we can't articulate.

Connections to the Course

Let's go through the arc of the course up to this point, and see how it connects to what we're talking about now:

- Scope and resolution are tied to particular patterns, emergent properties are tied to particular scopes.

- The timeless way is very GENERAL. The blueprint processes are more special.

- Iterated application of simple rules can yield complexity.

- We're considering design as enabling self-organizing pattern-forming processes.

- Multiple nested scales of patterns (applied from large to small).

- Organization is defined via connectivity — horizontal and vertical relations. We can think of it as a network of patterns that interact in direct or indirect ways.

- We can't harness the *ensemble* but we can harness a "pseudo-ensemble". That being said, this whole pattern language process is primarily for managing a trajectory.

- Models can't capture the nested context of reality.

- Much like in organisms, parts-from-whole processes are useful for differentiating spaces. When forces are resolved (by designing according to a parts-from-whole process), the system becomes more self-maintaining.

- Forces that demand resolution at multiple scales touch on the notion of multiscale requisite variety.

- The "function" of any pattern is found in how it serves other patterns.

- In complex settings, an ultimate reliance on “gut feeling” is preferable. There’s no metric or model that’ll provide you with the answer.

- Subtle functional relationships, which are initially hidden, must be respected and adapted to. Bounded subsystems generate distinct locales and make an internal vs external distinction, putting us at ease.

- We talked about selection of configurations as evolutionary processes. Additionally, structure-preserving transformations enable nonergodic adjacent possible exploration.

- Fitness can only be tested in the FULL, REAL context.

- Related to antifragility, reversible early action enables the capping of downside, and the opening of upside.

- As a way around bandwidth constraints, we must generate systems *more complex* than we can imagine and plan for.

- We need a sensitivity to the hyperlocal situation and context in order to resolve forces at that scale.

- Local social coherence of a pattern language enables integration with distributed action. It allows multiple players in the system doing their independent work to integrate. In software, for example, if you have a clean, clear pattern language, it’ll save you a lot of headaches when it comes to integrating things — whether it be standards for interfaces, etc.

- Top-down / central design processes *destroy* the possibility of such context-sensitive design.

- If parts don't interact appropriately, then subtle functional details and adaptations are destroyed before they arise.