Generative Pattern Language

By Michal Telem and Karen Fingas

Karen Fingas and Michal Telem 2018
Agenda:

1. What is Generative Pattern Language?
2. Background: Christopher Alexander
3. Main Concepts and Qualities
4. Examples for implementations
5. Recommended Processes
6. Pros and Cons
Christopher Alexander
Architect

Intimacy Gradient
“The Pattern Language”, 1977

Alexander was looking for a way to approach design at all levels: from cities and towns to houses and rooms and even constructions techniques.

In the 1970s, Alexander observed certain things that were seen repeatedly in the form of architectures and found that they are “patterns”.

In his book “A Pattern Language - Towns, Buildings, Construction” Alexander released 253 patterns extracted from his and his colleagues’ knowledge.

http://www.patternlanguage.com/apl/aplsample/aplsample.htm
Main Concepts

At first:

Context - Problem - Solution

Whenever a certain CONTEXT exists, a certain PROBLEM will arise; the stated PATTERN will solve the PROBLEM and there should be provided in the CONTEXT.
Each **Pattern** is a rule which establishes a relationship between a **context**, a **system of forces** which arises in that context, and a **configuration** which allows those forces to resolve themselves in the context.
Currently: Process oriented towards service systems

(i) voices on issues (who + what)

(ii) affording value(s) (how + why), and

(iii) spatio-temporal frames (where + when)
Currently: Process oriented towards service systems

<table>
<thead>
<tr>
<th>1 Pattern Label</th>
<th>Signing in for services</th>
</tr>
</thead>
</table>
| 2 Voices on issues(who/what) | a) For a client, what services are available to me, now and on appointment?  
(b) For a parent, what do I do with my kids while I’m busy?  
c) For a child, what can I do while my parent is at the Multi Service Centre? |
| 3 Affording values(how/why) | Matching client needs with MSC resources, so that holistic treatments are received.  
Triaging and scheduling so that urgent cases are prioritized, and wait times are tolerable |
Currently: Process oriented towards service systems

<table>
<thead>
<tr>
<th><strong>4 Spatio-temporal frames(where/when)</strong></th>
<th>On demand lookups of trending and prior MSC busy and slow periods transparently visible onsite and on the Internet, enabling clients to adjust and/or rebook</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5 Containing Systems(slower/larger)</strong></td>
<td>For municipal, regional and national agencies, are community health and social services in their jurisdictions well provided?</td>
</tr>
<tr>
<td><strong>6 Contained Systems(faster/smaller)</strong></td>
<td>For friends or assistants speaking on behalf or interpreting for a client, is the situation understood?</td>
</tr>
</tbody>
</table>
Generative

It not only tells us the rules of arrangement, but shows us how to construct as many arrangements we want which satisfy the rules.

http://www.patternlanguage.com/apl/aplsample/aplsample.htm
Many Patterns form a language
Unfolding - The key element of any generative code
<table>
<thead>
<tr>
<th>Attributes of the design process</th>
<th>Development philosophies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultimate purpose</td>
<td>Goal / purpose</td>
</tr>
<tr>
<td>Intermediate goals</td>
<td>Effectiveness / efficiency</td>
</tr>
<tr>
<td>Design focus</td>
<td>Ends / results</td>
</tr>
<tr>
<td>Design scope</td>
<td>Part</td>
</tr>
<tr>
<td>Design process</td>
<td>Creative problems solving</td>
</tr>
<tr>
<td>Design problems</td>
<td>Complexity and conflict</td>
</tr>
<tr>
<td>Design management</td>
<td>Centralized</td>
</tr>
<tr>
<td>Design control</td>
<td>Direct intervention in line with a master plan</td>
</tr>
<tr>
<td></td>
<td>Ateleological development</td>
</tr>
<tr>
<td></td>
<td>Wholeness / harmony</td>
</tr>
<tr>
<td></td>
<td>Equilibrium / homeostasis</td>
</tr>
<tr>
<td></td>
<td>Means / process</td>
</tr>
<tr>
<td></td>
<td>Whole</td>
</tr>
<tr>
<td></td>
<td>Local adaptation, reflection and learning</td>
</tr>
<tr>
<td></td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>Decentralized</td>
</tr>
<tr>
<td></td>
<td>Indirect via rules and regulators</td>
</tr>
</tbody>
</table>

David Ing “From Environmental Structure to Service Systems Thinking: Wholeness with Centers Described with a Generative Pattern Language.”.
Proposed procedures for making a pattern Language

1. Pattern Mining
2. Pattern Prototyping
3. Pattern Writing
4. Pattern Organizing
5. Catalog Editing

Karen Fingas and Michal Telem 2018
The quality without a name: Objectivity

To seek the timeless way we must first know the quality without a name. There is a central quality which is the root criterion of life and spirit in a man, a town, a building, or a wilderness. This quality is objective and precise, but it cannot be named.

(Christopher Alexander)
Examples:

Build environment - http://www.livingneighborhoods.org/ht-0/archive.htm

Software development - Agile


Service Systems - An Alexandrian format and pattern for Multi Service Centers

A Pattern Language for Systemic Transformation (PLAST) - “You cannot understand a system until you try to change it” said Kurt Lewin

https://www.academia.edu/9138366/A_Pattern_Language_for_Systemic_Transformation_PLAST_-_Re_Generative_of_Commons
Generative Pattern Language - Pros:

Flexible and versatile Methodology

Great tool to combine hard system thinking and soft systems thinking

Gives the opportunity to pay full respect to unique features of a system

Ease of use - Provides sequence of actions - order and Tells us which patterns “go together”
Generative Pattern Language - Cons

Relies on collaboration - not everyone can/will participate

Patterns take labour and time to develop

Need experts
“A generative code is a system of unfolding steps that enable people in a community to create...”

- Christopher Alexander
References


https://www.youtube.com/watch?v=j3a4j9xN6lI.


http://dreamsongs.net/Files/Pat ternsOfSoftware.pdf.

Iba, Takashi, “Peer Learning via Dialogue with a Pattern Language”, in the 7th International Conference on Collaborative Innovation Networks (COINs17), Detroit, USA, Sep., 2017.
https://www.slideshare.net/takashiiba/peer-learning-via-dialogue-with-a-pattern-language

https://www.slideshare.net/ChihyangLi/design-patterns-in-swift-ch0-introduction
https://blog.p2pfoundation.net/what-is-an-unfolding/2015/04/12
https://www.slideshare.net/ChihyangLi?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview