Service Systems Thinking:
From Environmental Structure to a New Generative Pattern Language

David Ing
International Society for the Systems Sciences, and
Aalto University

July 2015 – PURPLSOC, Danube University Krems

Agenda

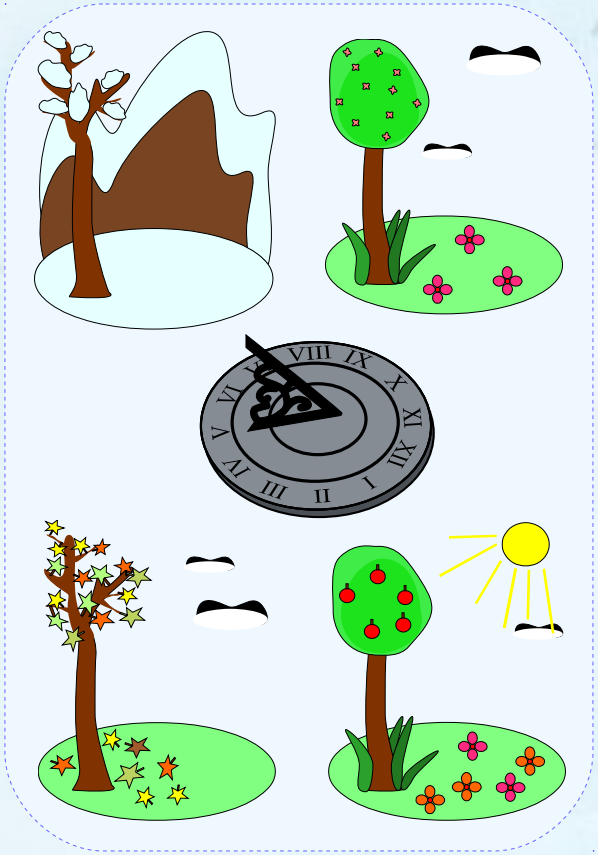
1. What could
   Service Systems Thinking be?

2. From architectural programming
   to a generative pattern language

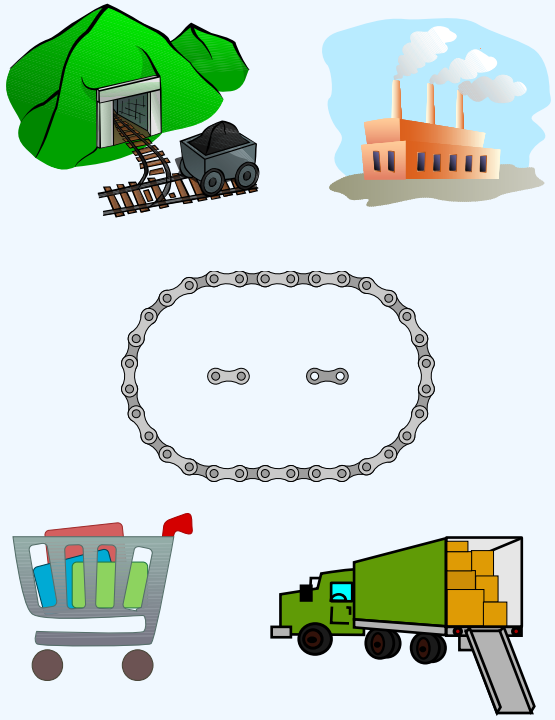
3. Federated wiki and
   a value constellation ontology?

4. An invitation

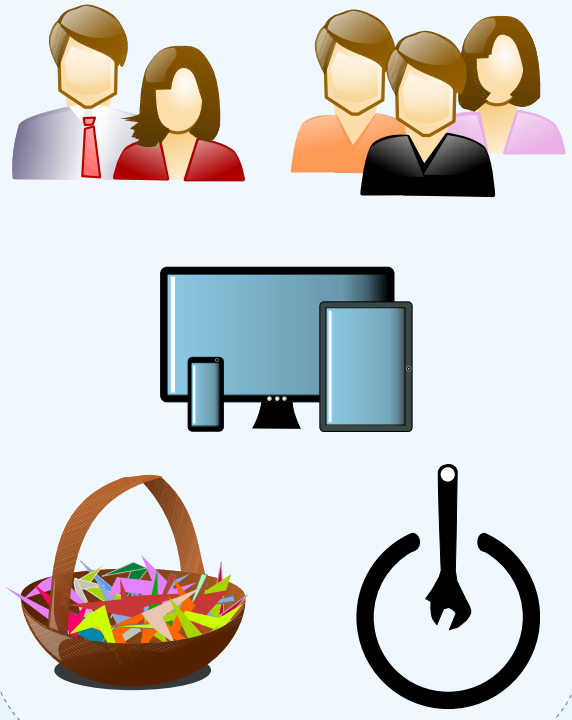
Is thinking different across agricultural systems, industrial systems, and service systems?



Agricultural Systems



Industrial Systems



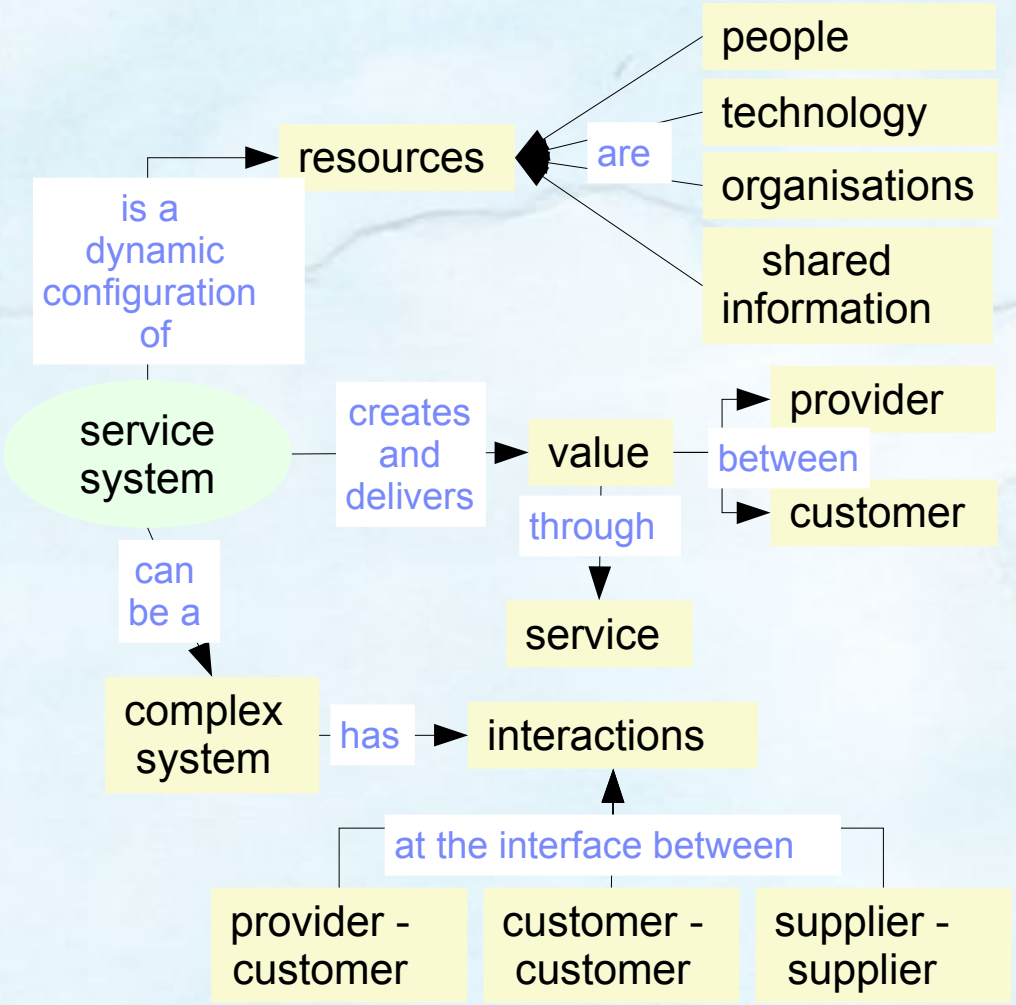
Service Systems(?)

Service systems in our society can be ranked from concrete to abstract, as subjects for schoolchildren

Systems that move, store, harvest, process	• Transportation	K
	• Water and waste management	1
	• Food and global supply chain	2
	• Energy and energy grid	3
	• Information and communications (ICT) infrastructure	4
Systems that enable healthy, wealthy and wise people	• Building and construction	5
	• Banking and finance	6
	• Retail and hospitality	7
	• Healthcare	8
	• Education (including universities)	9
Systems that govern	• Government (cities)	10
	• Government (regions / states)	11
	• Government (nations)	12

Source: Spohrer, James C., and Paul P. Maglio. 2010. “Toward a Science of Service Systems: Value and Symbols.” In Service Science: Research and Innovations in the Service Economy, edited by Paul P. Maglio, Cheryl A. Kieliszewski, and James C. Spohrer, 157–94. 10.1007/978-1-4419-1628-0_9

Service systems (Cambridge IfM and IBM, 2008)

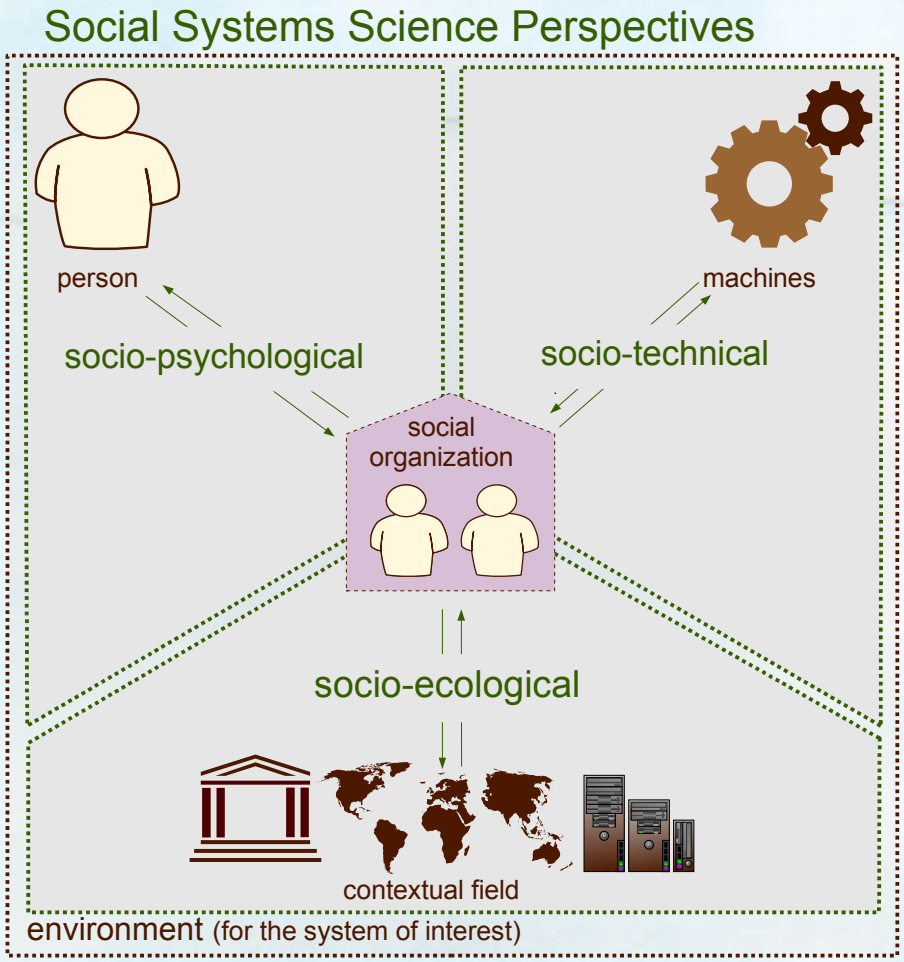


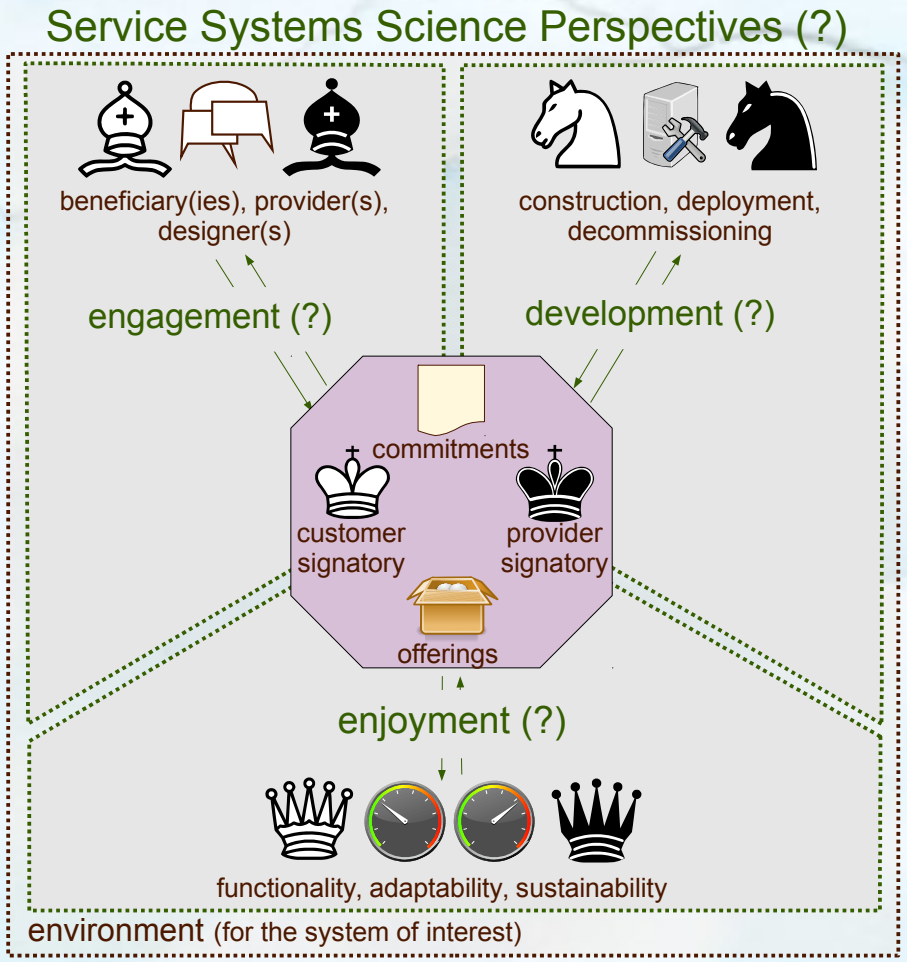
A service system can be defined as a dynamic configuration of resources (people, technology, organisations and shared information) that creates and delivers value between the provider and the customer through service.

In many cases, a service system is a complex system in that configurations of resources interact in a non-linear way. Primary interactions take place at the interface between the provider and the customer. However, with the advent of ICT, customer-to-customer and supplier-to-supplier interactions have also become prevalent. These complex interactions create a system whose behaviour is difficult to explain and predict.
(IfM and IBM, 2008, p. 6)

Source: IfM, and IBM. 2008. *Succeeding through Service Innovation: A Service Perspective for Education, Research, Business and Government.* Cambridge, UK: University of Cambridge Institute for Manufacturing. http://www.ifm.eng.cam.ac.uk/ssme/ .

Can we build on Social Systems Science towards a new Service Systems Science?





Agenda

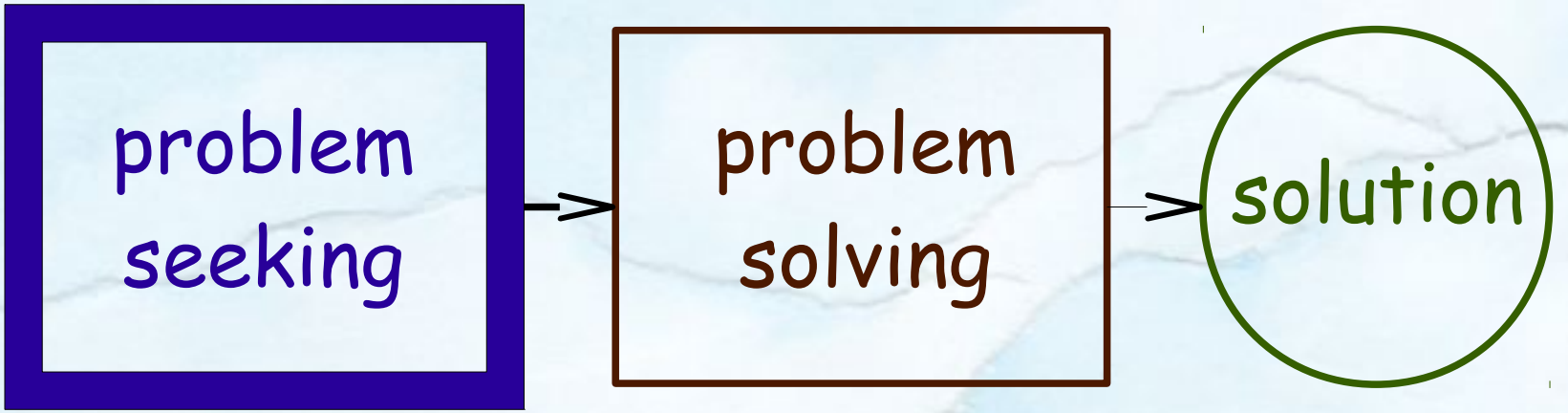
1. What could
   Service Systems Thinking be?

2. From architectural programming
   to a generative pattern language

3. Federated wiki and
   a value constellation ontology?

4. An invitation

Pattern language presumes *problem seeking* as *architectural programming*, and *problem solving* as *design*

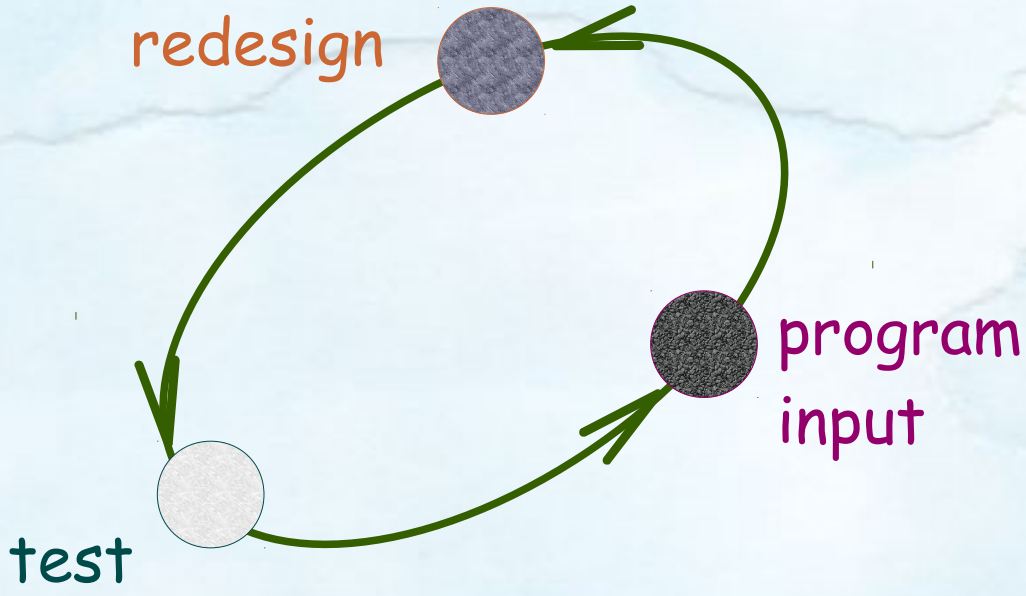


Programming is a specialized and often misunderstood term. It is "*a statement of an architectural problem* and the requirements to be met in offering a solution. While the term is used with other descriptive adjectives such as *computer* programming, *educational* programming, *functional* programming, etc., in this report, programming is used to refer only to architectural programming.

Why programming? The client has a project with many unidentified sub-problems. The architect must define the client's total problem.

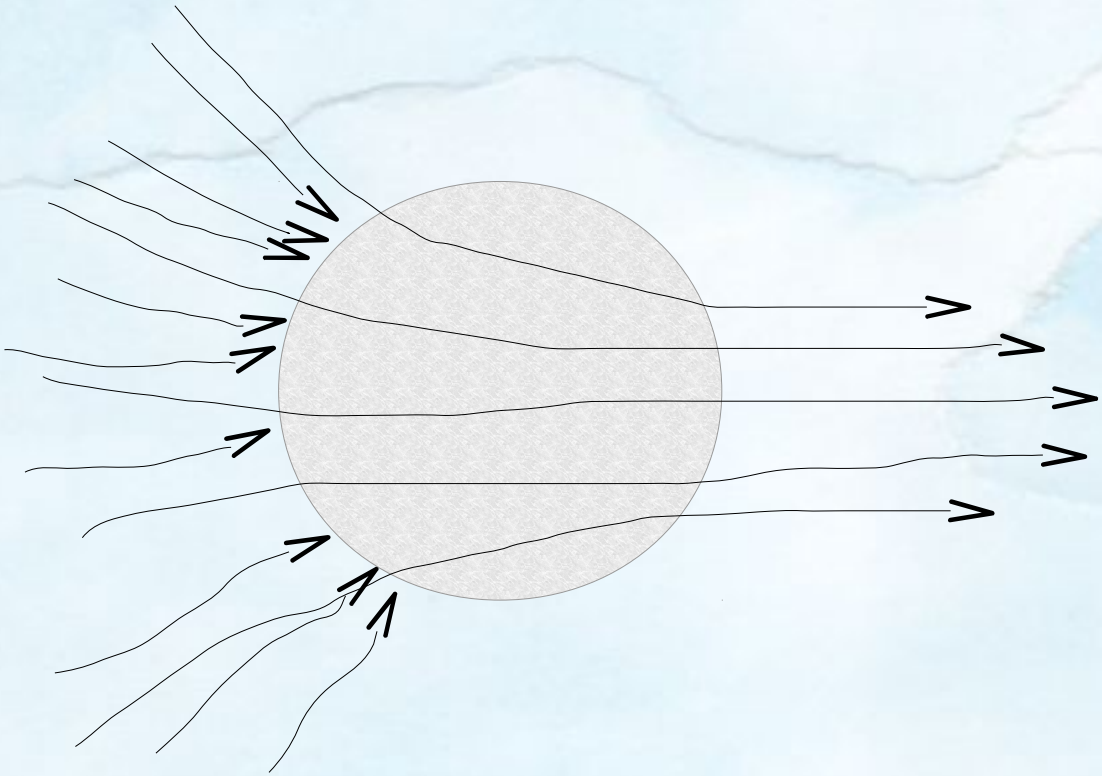
Design is problem solving; programming is problem seeking. The end of the programming process is a statement of the total problem; such a statement is the element that joins programming and design. The "total problem" then serves to point up constituent problems, in terms of four considerations, those of form, function, economy and time. The aim of the programming is to provide a sound basis for effective design. The State of the Problem represents the essense and the uniqueness of the project. Furthermore, it suggests the solution to the problem by defining the main issues and giving direction to the designer (Pena and Focke 1969, 3).

Programming through design, testing and redesign is inefficient



If a client approaches the architect with very little information, the architect may have to respond by programming through design. He could produce sketch after sketch and plan after plan trying to satisfy undefined requirements. Programming through design can involve misuse of talent and, indeed, risks of creating a “solution“ to the wrong problem.
(Pena and Focke 1969).

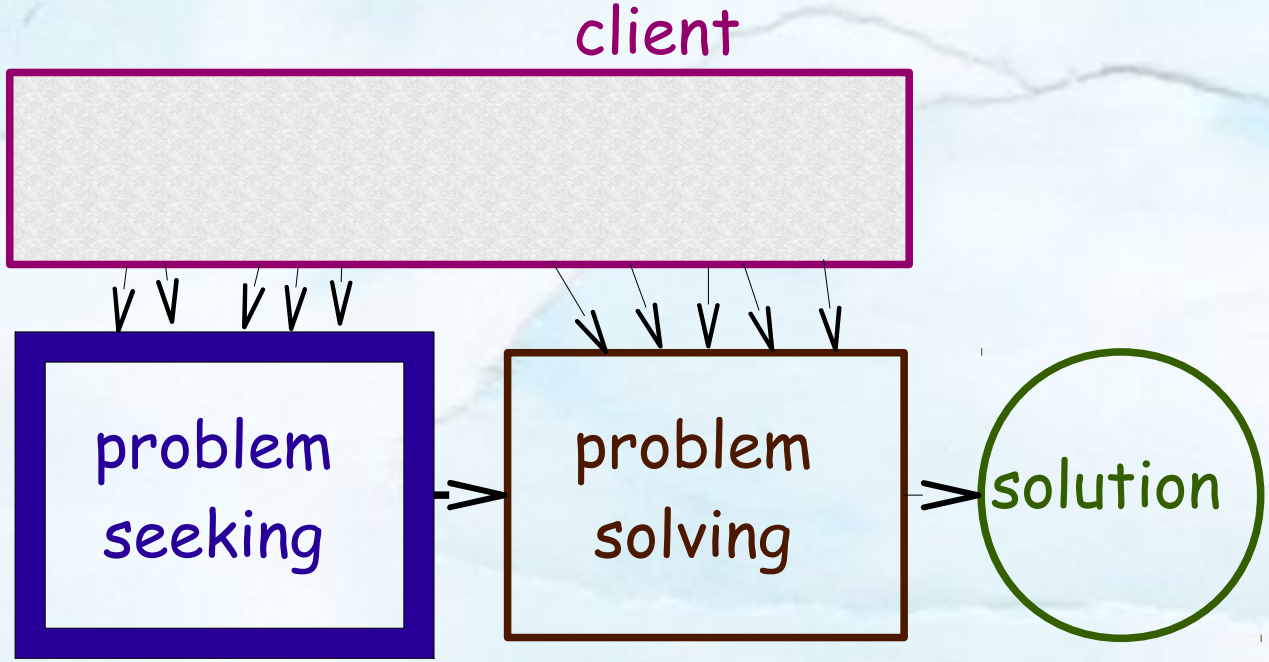
Discrimination between major ideas and details is necessary to avoid confusion in problem solving



On the other hand, a client may present the architect with too much information but involving mostly irrelevant details. The risk here is that the architect's solution will be based on details rather than major ideas. In this case, the architect must plough through an abundance of information and discriminate between major ideas and details (Pena and Focke 1969).

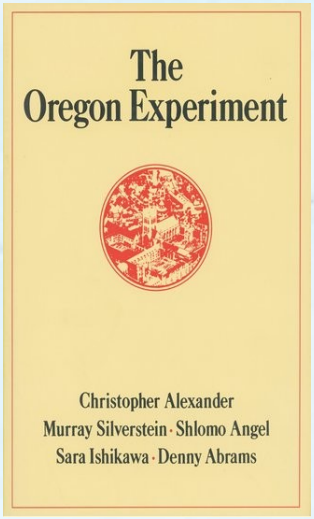
The client is involved in the process of architectural programming

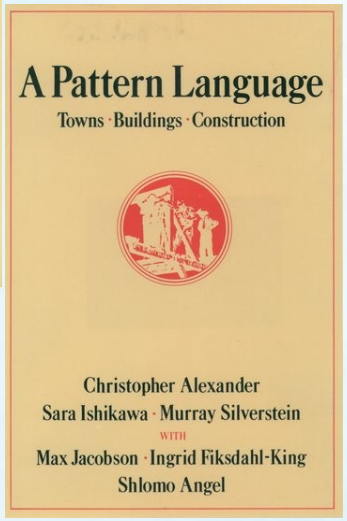
The analytical procedure used by CRS provides a framework for decision making. Within it the architect help the client identify and make decisions that need to be made prior to design. Within it, the architect can suggest alternatives and other information to bring about decisions.

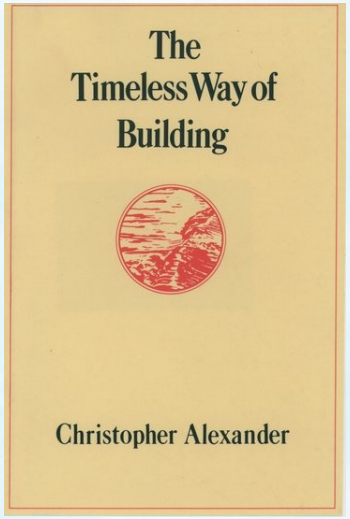


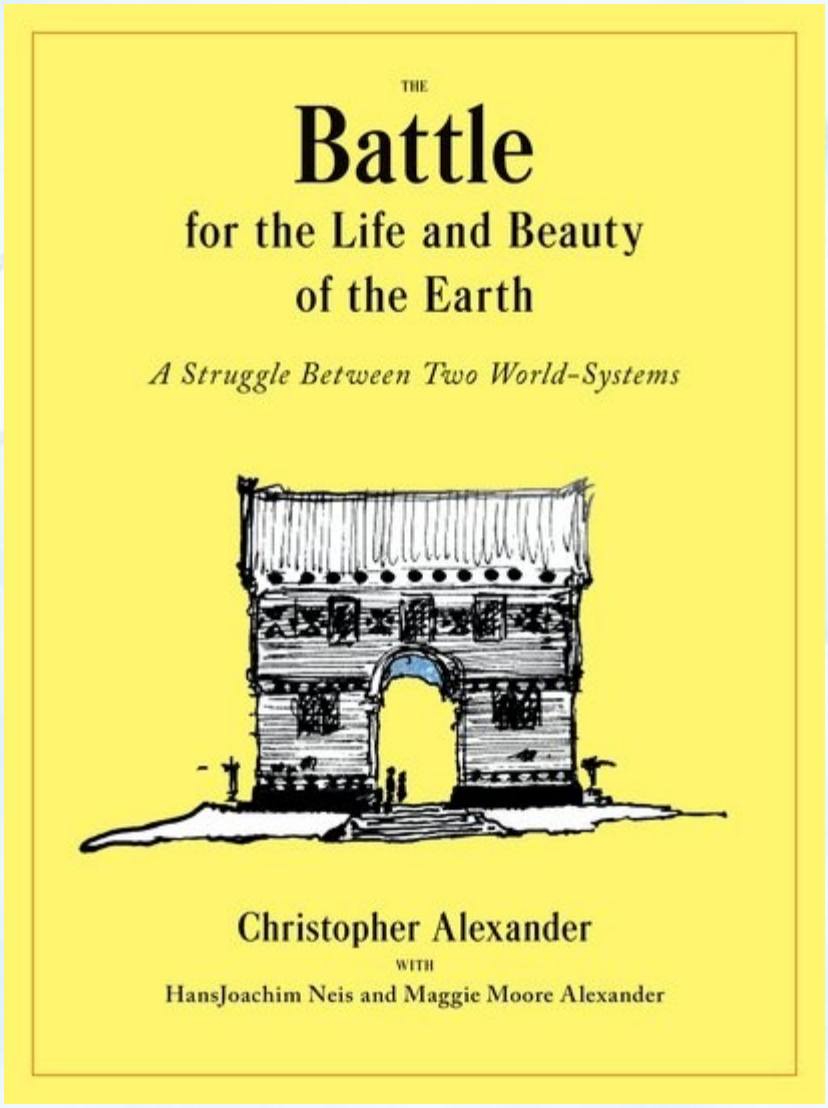
There are times when the architect must evaluate the gains and risks in order to stimulate a decision. Yet, note the emphasis on client decisions; the architect merely participates and at most, recommends.
(Pena and Focke 1969).

The writing of 1975-1979 by Alexander was prescriptive; the 2012 is reflections on practice









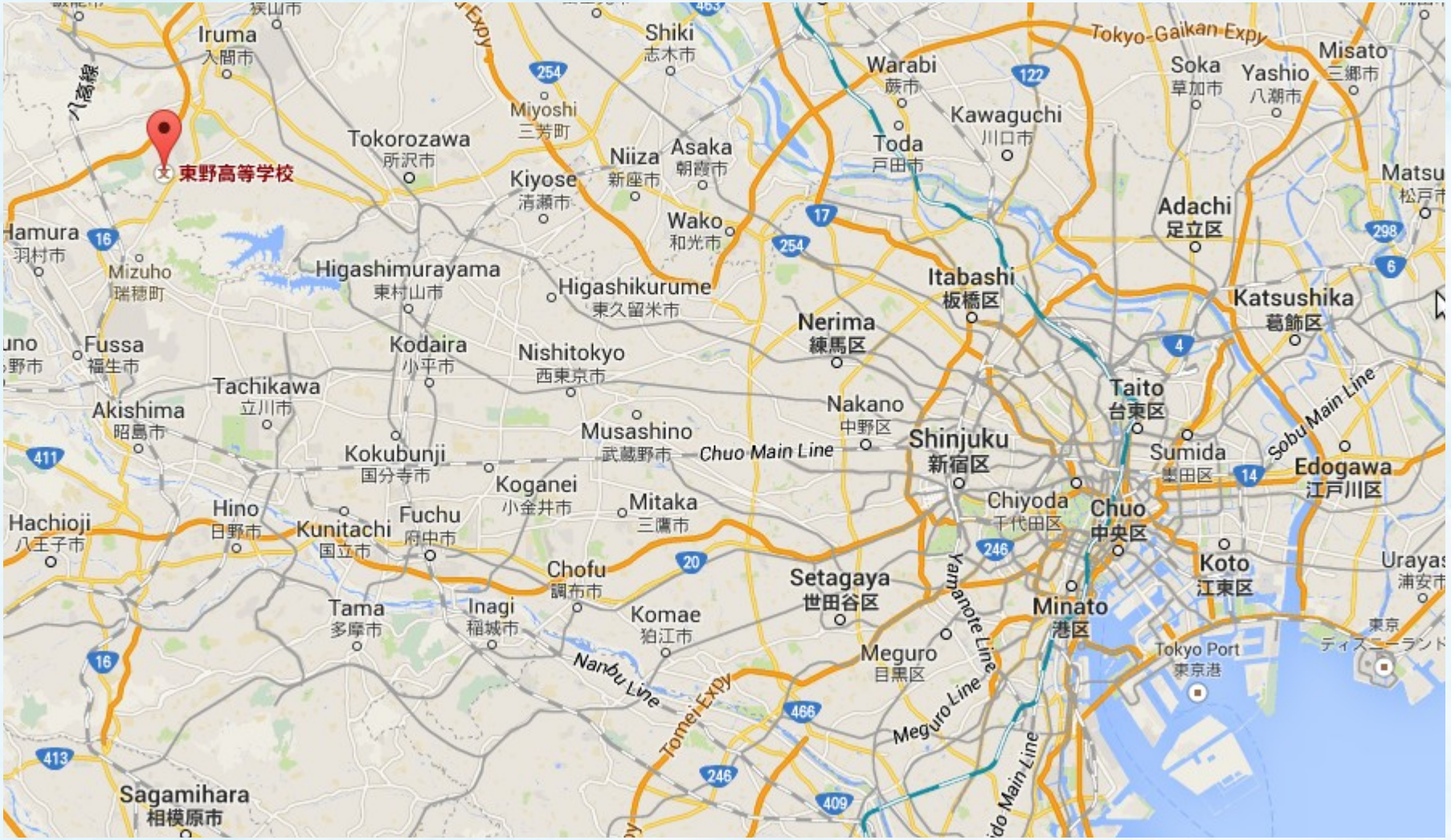
学校法人 盈進学園 東野高等学校

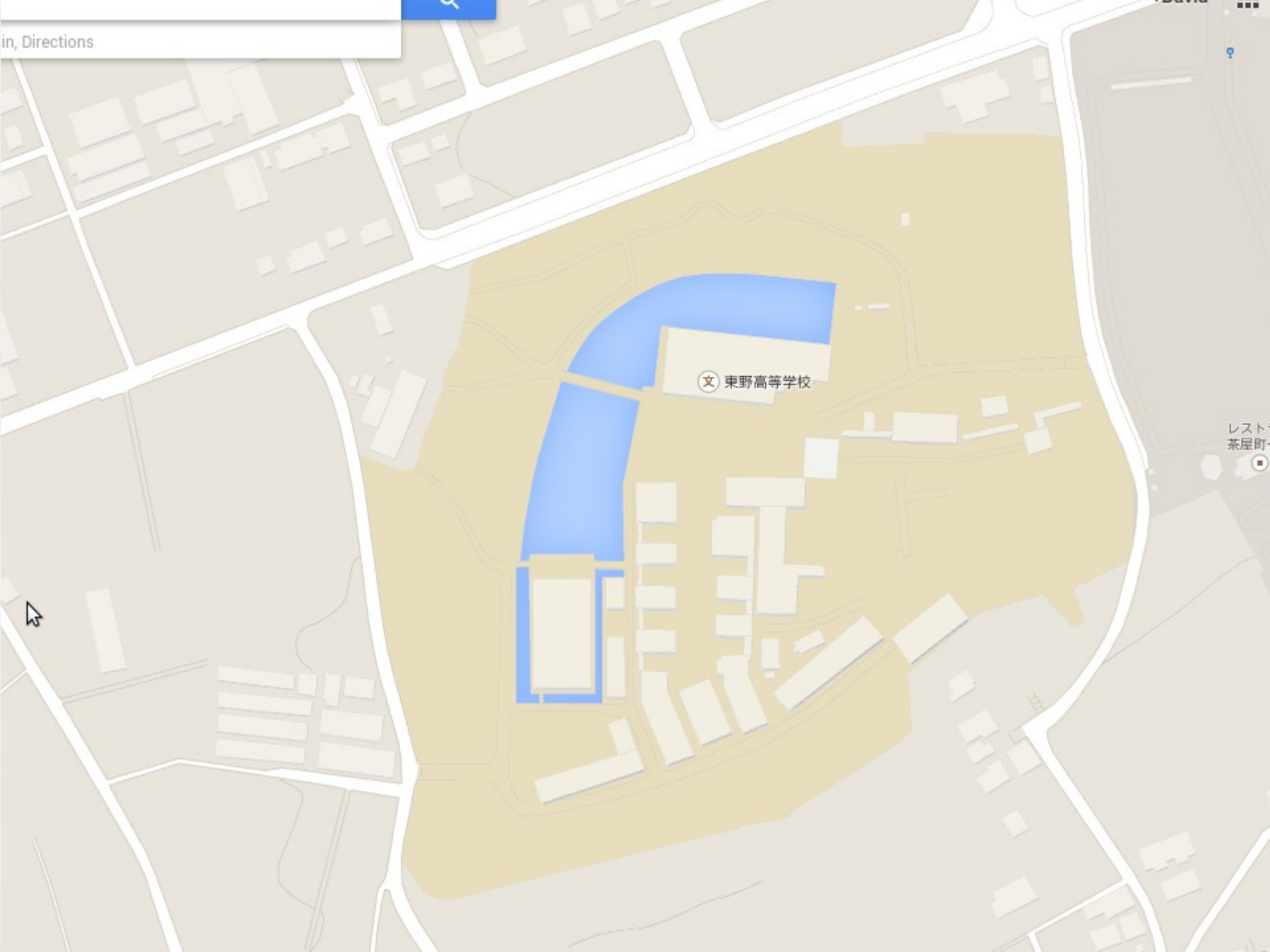
入試関連情報 | 保護者の方々へ

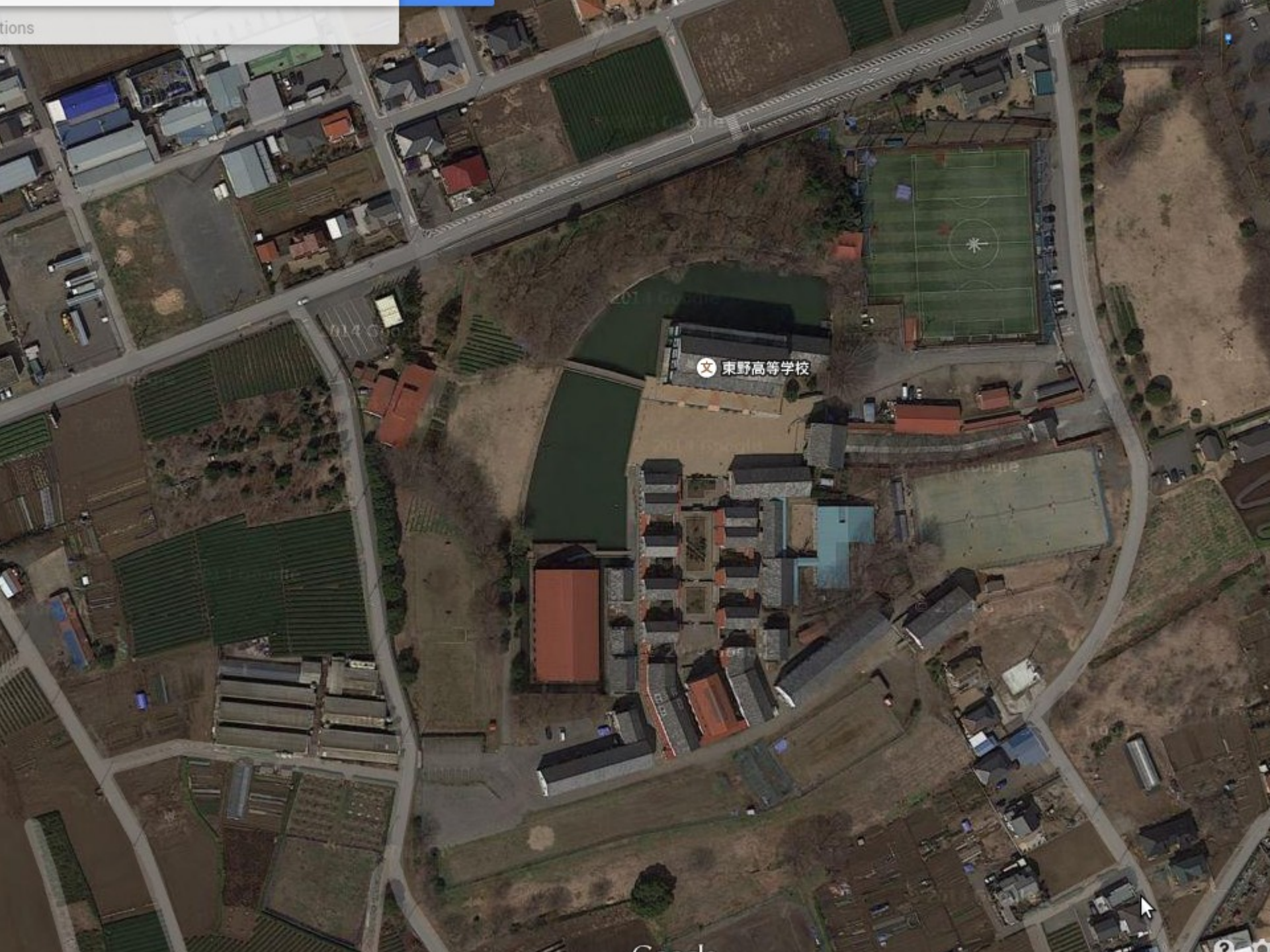
HOME | CURRENT NEWS | ABOUT HIGASHINO | ACADEMICS | GUIDANCE | ACTIVITIES & ANNUAL EVENTS | ADMISSIONS

twitter 犬猿の仲印パからノーベル平和賞受賞者。17歳のマララさんについて「テロリストが最も恐れるのは『教科書を持った少女』だ」と潘基文氏。＜ペンは剣より強し＞

The site originally was tea fields in Iruma, Saitama prefecture, northwest of Tokyo




in, Directions
東野高等学校


東野高等学校

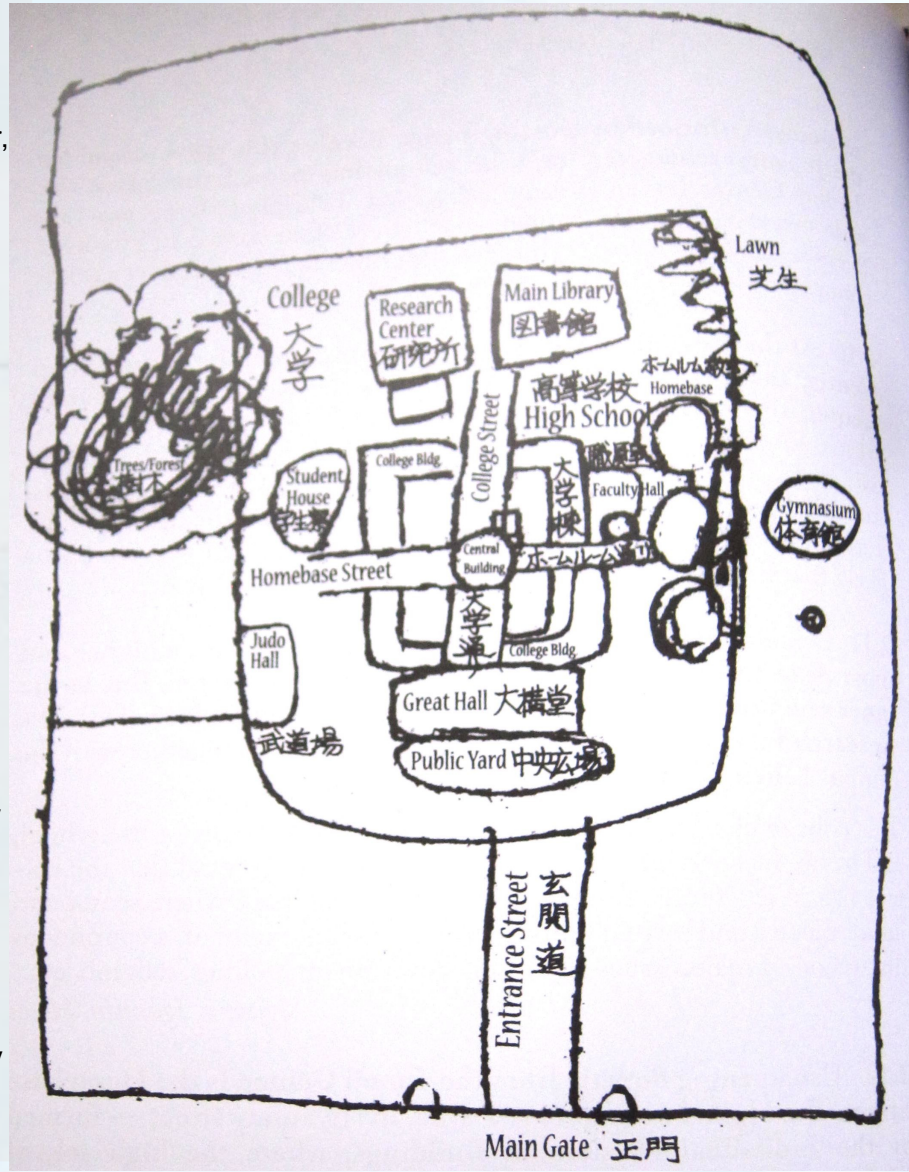
The practices employed on the 1985 Eishin project can be traced with 8 activities

1. Interview on hopes and dreams
2. Make a “poetic vision” as first sketch of a pattern language
3. Make the rudimentary pattern language physically coherent
4. Refine the language through discussions
5. Obtain approval of the pattern language
6. Renegotiate pattern language with space and money within budget
7. Find systems of centers in (i) the pattern language, and (ii) the places in the land. Combine them.
8. Adjust the site plan on the site itself (not on models)

(3) Make the rudimentary pattern language physically coherent

The Completeness of the Language: Seven Principles

- **Relationships**. Each pattern establishes certain relationships which should exist in the finished campus. The sum total of those relationships, expressed by the patterns in the language, acting together, define the possible configurations which this language generates. [p. 124]
- **Spatial**. A given pattern contains, or defines, certain spatial entities. The relationships are defined among these spatial entities.
- **Reliability**. The essence of these relationships is that they must be reliable, and true. They cannot be arbitrary relationships (as they might often be in a single person's design). They need to be sufficiently true, so that we can trust them, and would want to find these relationships present in any version of any campus that might be generated by this language.
- **Consistency**. It is not necessarily easy to define a system of patterns which is consistent. For example, if one pattern asserts a certain relationship between two entities, and another pattern asserts a further relationship between the same entities, but one which is inconsistent with the first, then that system of two patterns is inconsistent, and can only, with great difficulty, work to generate real physical configurations.
- **Inconsistency**. From time to time, two patterns which are physically inconsistent may be refreshing and life-giving. This happens because the contradiction generates vigor and opens new ideas.
- **Completeness**. A system of patterns is complete if it contains sufficient relationships to allow a well-formed configuration to be built.
- **Coherence**. A system of patterns is coherent if the relationship specified amongst the patterns tend, most of the time, to generate easily graspable mathematical configurations.



(5) Obtain approval of the pattern language [page 1 of 8]

This pattern language is a list of key centers, each of which contributes some essential quality to the campus. The list was established long before any design started. [p. 130]

The list contains 110 essential patterns, each describing a generic kind of center, and itself made of other centers. As they are defined here, these 110 key patterns completely govern and define the life of the school. Even before we have any idea about the physical configuration of the buildings, their shape, or design, or the way these centers are made real in space, it is already obvious that the school is given its life to an enormous degree, merely by this list of patterns. [p. 151]

1.	Global Character of the Campus
2.	The Inner Precinct
3.	The Buildings of the Inner Precinct
4.	The Streets of the Inner Precinct
5.	The Outer Precinct
6.	Features of the Inner Precinct
7.	Special Outdoor Details
8.	Interior Building Character

1.1	An outer Boundary surrounds the Campus.	A white, 60 cm wall serves as the based for a wooden fence. [....]
1.2	Contained by this Outer Boundary there is an Outer Precinct. ...	A second wall, far inside the first, surrounds the school itself, and forms a second zone between the first and second wall. [....]
1.3	The Inner Precinct is a densely built area where School and College have their major buildings and activities.	It is the place where the daily life of students and faculty occurs. [....]
1.4	...	...
1.5	As a whole, the Campus is given character by stone foundation walls, natural concrete walls, wood columns, ...	In addition ...

(6) Renegotiate pattern language with space and money within budget

How can something like the cost or budget be made practical?

... we finish the pattern language phase with a serious analysis of space and money. It is done right away, so that any hidden conflicts are visible, and can immediately come into the open to get resolved.

First of all, we make a record of all of the spaces and areas which were defined by the pattern language -- adding up, pattern by pattern, the total outdoor space and indoor space. In our case, the analysis showed us that the requested numbers were too large. [….]

AREAS REQUESTED BY THE FACULTY			
A. Built Space (indoor space in square meters)	**First guess requested**	**Available 73.4%**	**Renegotiated finalized**
Public Yard Buildings	**945 m²**	**693 m²**	**750 m²**
Buildings of the Tanoji Center	**7583 m²**	**5566 m²**	**5604 m²**
Cloister (research center)	**1350 m²**	**991 m²**	**1150 m²**
Homebase Street buildings	**5680 m²**	**4169 m²**	**4300 m²**
Buildings in the Outer Precinct	**2432 m²**	**1785 m²**	**1400 m²**
Total	**17990 m²**	**13204 m²**	**13204 m²**
B. Coverage of Land (outdoor space in square meters)	**First guess requested**	**Available 79.5%**	**Renegotiated finalized**
Total	**84286 m²**	**67000 m²**	**67000 m²**

Second, as the simplest way to trim all space to our available budget, we made an average percentage reduction for all items, one figure for trimming indoor space; and then another for exterior land area. Each item was trimmed by a similar (but not identical) percentage. [....]

Third, we then asked the faculty to re-allocate the spaces, keeping the same trimmed totals, in order to conform to the available resources. The rule was simple: they could increase some, but must then decrease others, so that the total areas remained as they must remain.

(7) Find systems of centers in (i) the pattern language, and (ii) the places in the land. Combine them. [page 1 of 3]

The first system consists of **patterns** created notions or entities that exist in people's minds). These patterns exist in a loose and undeveloped form in people's minds, even if they have not explicitly built a pattern language. When the pattern language *is* explicitly defined, it is more clear and makes a more powerful system which will get better results, especially because it comes from the feelings of people themselves. [p. 169]

The Most Important Centers Given by the Pattern Language

... the patterns together, geometrically … does not indicate any one arrangement on the land.

1. The **Entrance Street**.
2. The entrance street leads to a big square element which we refer to as the **Tanoji Center**.
3. This was to be the core of the college, and the center of gravity of the **Five College Buildings**.
4. Leading out from the Tanoji Center, in some direction, is **The Homebase Street**, the core of the high school.
5. **Individual Classroom Buildings** open along the **Homebase Street**.
6. The **Great Hall** and **Main Square** next to it.
7. The **Library and Research Center**, to one side. [p. 170]

Diagram 1: Seven most important centers in the pattern language, which together give a broad conceptual picture of a possible layout that the centers can have. Not to scale.

(7) Find systems of centers in (i) the pattern language, and (ii) the places in the land. Combine them. [page 2 of 3]

The second system exists in the form of **places** on the site, discernible places that can be seen and felt on the site, if you have sufficient sympathy with the land. You can make this system explicit, by making a map of the centers, and paying attention to their structure. [p. 169]

The Most Important Centers Suggested by Land Forms

... "natural places" in the land.

1. **Natural Entrance Position**. The most important among these centers was the location of the main approach. This was in the southeast corner, partly because of a bus stop in Nihongi village, and partly because of the feeling of one's natural desire about how best to approach the site.

2. **The Ridge**, running along the south of the project site. A beautiful spot, with breeze, sunshine, view ... and a very delightful feeling. This was the high point in the site, and it was on this very point that we sat and looked and sat and talked, until we began to see what was really there to be seen.

3. **The Swamp**, where vegetables used to be grown, the low point in the terrain -- a kind of swamp -- that later became a lake.

4. **A Natural Place for Large Buildings**, a zone in the middle, running the way contours ran, from north to south.

5. **Minor Entrance Position**, the northwest corner -- a natural high spot, from which to view the site, also a natural point for a secondary entrance.

6&7. **East and West Ends of the Ridge**, the two ends of the ridge, which formed natural high points, and at each end, the feeling of a terminus, along the two ends of the ridge. [p. 171]

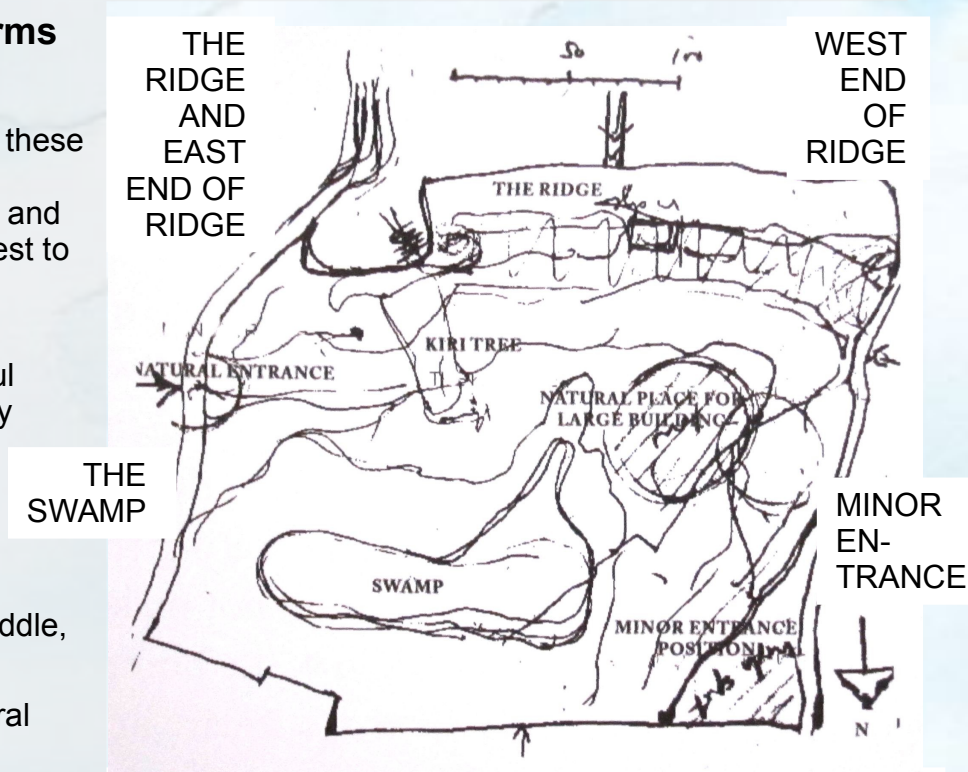


Diagram 2: The seven most NATURAL centers in the land, which, together can lead to a basic possible layout that the centers can have, in their LOCATIONS in the land.

(7) Find systems of centers in (i) the pattern language, and (ii) the places in the land. Combine them. [page 3 of 3]

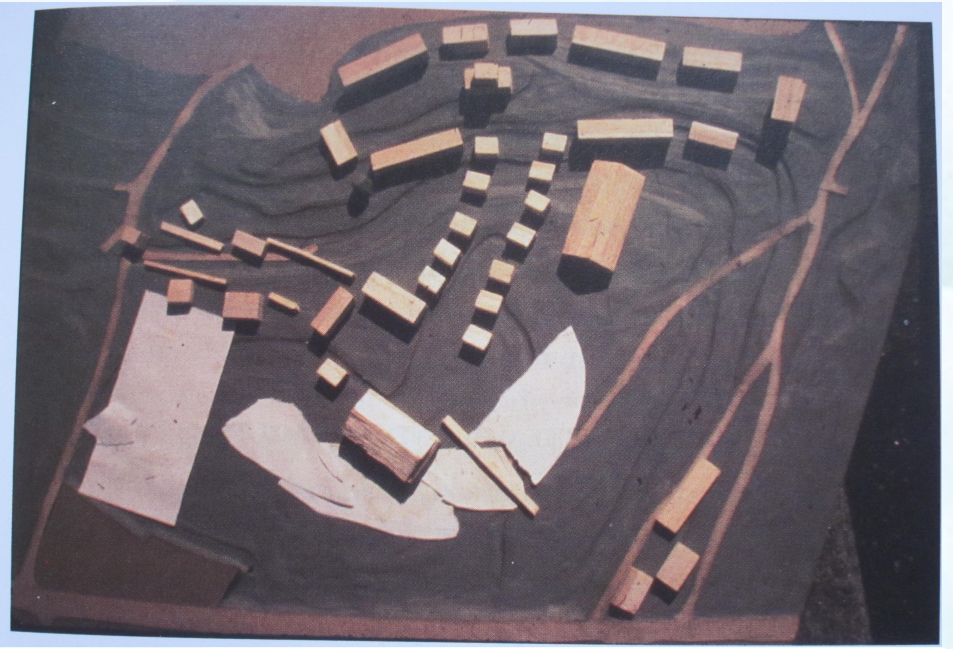
... to bring these two systems of centers together. We have to hunt for a single configuration which springs from both centers, and integrates the qualities of both. We must find a way in which the system of centers defined by the pattern language can be placed, so that it enhances, preserves, and extends, the system of centers which is already in the land. It is a kind of healing process, which uses the new centers given by the pattern language, to heal the configuration of the old centers -- those that exist in the land.

... this is the single most difficult phase of the work. ... it took ... about nine months of continuous effort, to get the site plan right. [p. 173]

... after ... months of frustration, the problem did get solved.

[...] a new point emerged. The fact that the homebase street would be more powerful as an *approach* to the Tanoji Center, than as something *hanging off* it. This was hard to see, at first, because it implied reversing the main sequence of the pattern language. But when we tried it, it was clear that the sequence almost instantaneously "jelled" with the land configuration.

Instead of this:	We now had:
1) Entrance Street	1) Entrance Street
2) Main Square	2) Main Square
3) Tanoji Center (College)	3) Home Base Street (High School)
4) Home Base Street (High School)	4) Tanoji Center (College)



The small balsa-wood model of the site, scale 1:500, on which the solution finally became apparent

(8) Adjust the site plan on the site itself (not on models)

We have already made it clear that nearly all of our work on the site plan was done on the site itself. Whatever we did on models, we used the models as if they were site itself -- and relied on feelings that we could feel in the model, imagining that it was the site itself. This was made necessary by the huge distance between California and Japan.

As one works on a site, and the plan gradually emerges, it is necessary, of course, to leave marks -- sticks, stones, markers of various kinds -- to fix the position of the different things which have been decided. On the Eishin site... the site was covered in tea bushes. [….] A marker therefore had to be about six feet high, even to be seen at all.

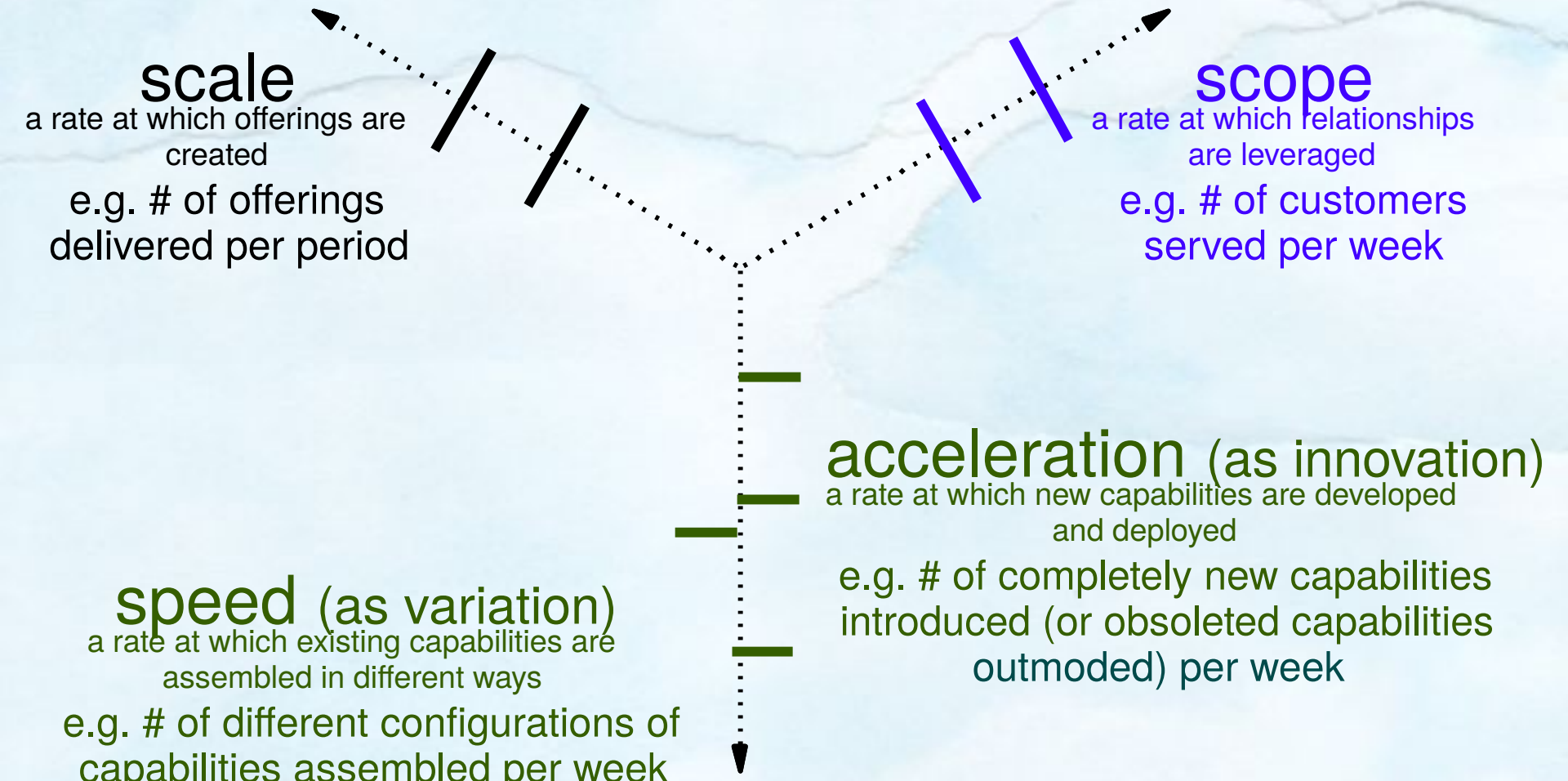
So we used six-foot-long bamboos. [….]. We ... tied different colored ribbons and cloths -- white, yellow, red, blue -- to the ends of our long bamboos. These were our markers -- our *flags*. [p. 180]

We had started making these flags quite early in the process. Even in July of 1982, as we began to get an idea established about the entrance position, we marked it with three or four of these flags. They looked beautiful. **And they made it possible to visualize the evolving site plan, truly, because they were real.**

When I came back to Japan after the breakthrough in November, we took about two hundred of these flags to the site, and began planning them in the ground, starting to make a realistic version as opposed to the very rough-and-ready diagram we had made so far. At this stage, now dealing with the real positions and dimensions on the land, we brought true feeling to the land itself. It was visible on the ground. [p. 181]

Colored flags, to identify various special purposes and areas on the land.

Should we think of architectural programming as trade-offs between scale, scope, speed and acceleration?



Solutions to problems come not from a pattern per se, but through generativity in the pattern language

In many problem-solving strategies, we try to attack problems directly. In doing so, we often attack only symptoms, leaving the underlying problem unresolved. Alexander understood that good solutions to architectural problems go at least one level deeper. **The structures of a pattern are not themselves solutions, but they generate solutions. Patterns that work this way are called generative patterns. A generative pattern is a means of letting the problem resolve itself over time, just as a flower unfolds from its seed:**

Image: Unfolding, by mrpeaches83 at https://www.flickr.com/photos/mrpeaches83/

9. This quality in buildings and in towns cannot be made, but only generated indirectly by the ordinary actions of the people, just as a flower cannot be made, but only generated from the seed (Alexander, 1979. p.xi)

An ordinary language like English is a system which allows us to create an infinite variety of one dimensional combinations of words, called sentences…. A pattern language is a system which allows its users to create an infinite variety of those three dimensional combinations of patterns which we call buildings, gardens, towns. ….

Thus, as in the case of natural languages, the pattern language is generative. It not only tells us the rules of arrangement, but shows us how to construct arrangements as many as we want which satisfy the rules. (Alexander, 1979: pp. 185 186)

Why is generativity important? First, ... most real problems go deeper than their surface symptoms, and we need to address most interesting problems with emergent behavior. Second, a good pattern is the fruit of hard work and intense review and refinement. Simple problems can be addressed through simple rules, since the solutions are more direct or obvious than we find in generative solutions. The pattern form excels an engaging the reader in generative solutions: to understand the principles and values of lasting solutions and long-term emergent behavior. Good patterns go beyond the quick fix.

[James O. Coplien, Software Patterns 1996, pp. 33-34]

An *unfolding* is a process which gets you from one stage or moment of development to the next moment of development, in the evolution of a neighborhood or in the evolution of a building

1. An unfolding is a dynamic configuration that acts to generate form.
2. An unfolding arises from the particular whole in which it is forming. It is shaped by the whole, and acts upon the whole, and causes the rebirth of the whole.
3. An unfolding is by its nature personal, and requires human input and human feeling from the people doing the work, as an essential part of its contribution to the formation of the environment.

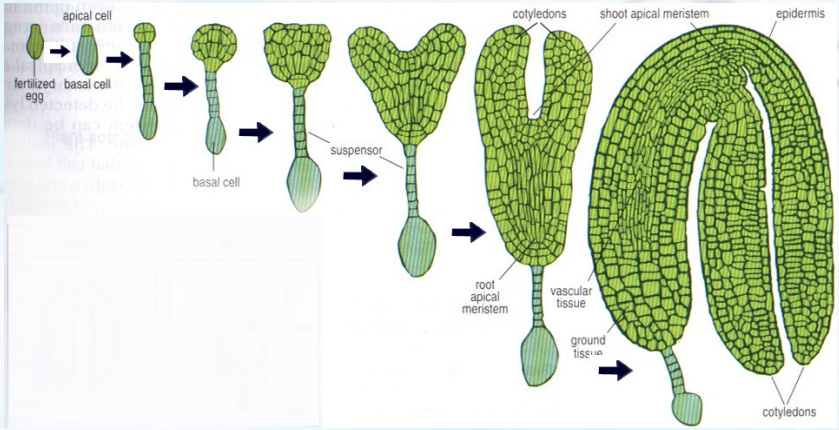
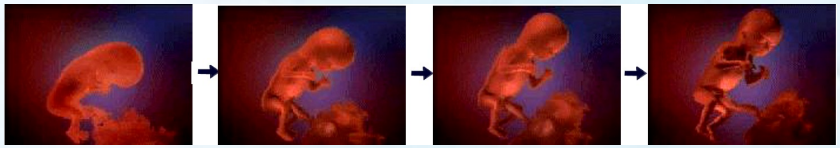
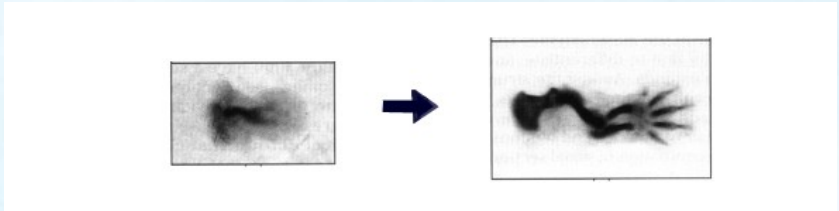


Diagram of a typical angiosperm (flowering plant) unfolding



Photographs of a human embryo unfolding

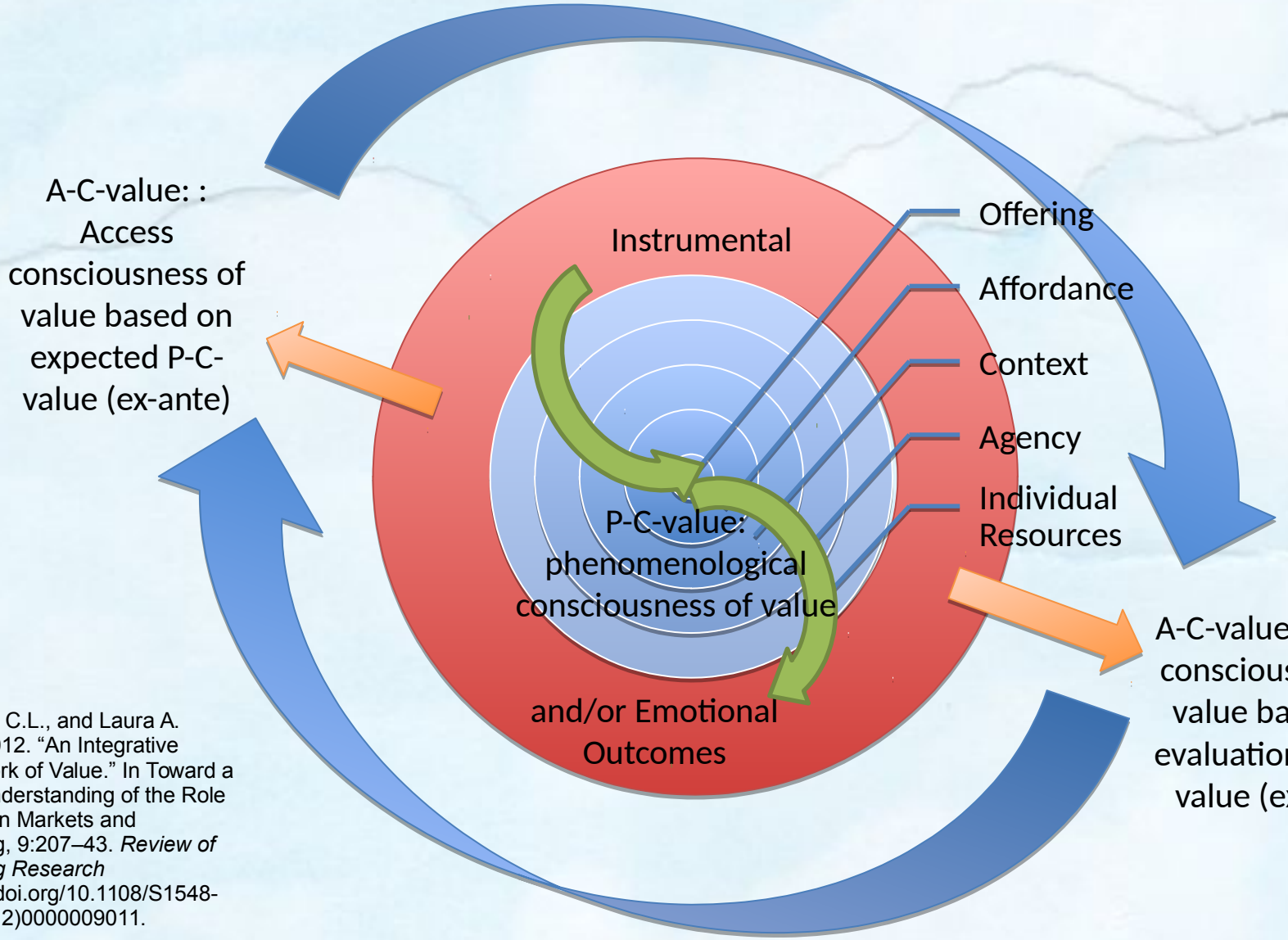


Two photographs, three days apart, of a mouse foot unfolding

It is helpful to compare such unfoldings with similar phenomena in plant morphogenesis and embryology. Both in the angiosperm shown below, and in the embryo shown beneath it, you can picture each unfolding as a limited and brief process which in the first one gradually shapes the seed, and in the second, takes the blur that is the beginning of a hand in the embryo, to the next stage of development where the hand gets its first outline fingers.

Adapted from Christopher Alexander, “What is an unfolding?” at http://www.livingneighborhoods.org/ht-0/whatisanunfolding.htm

Research in service science provides new dynamic conceptualization of value of an offering



Ng, Irene C.L., and Laura A. Smith. 2012. "An Integrative Framework of Value." In Toward a Better Understanding of the Role of Value in Markets and Marketing, 9:207–43. *Review of Marketing Research* http://dx.doi.org/10.1108/S1548-6435(2012)0000009011.

Agenda

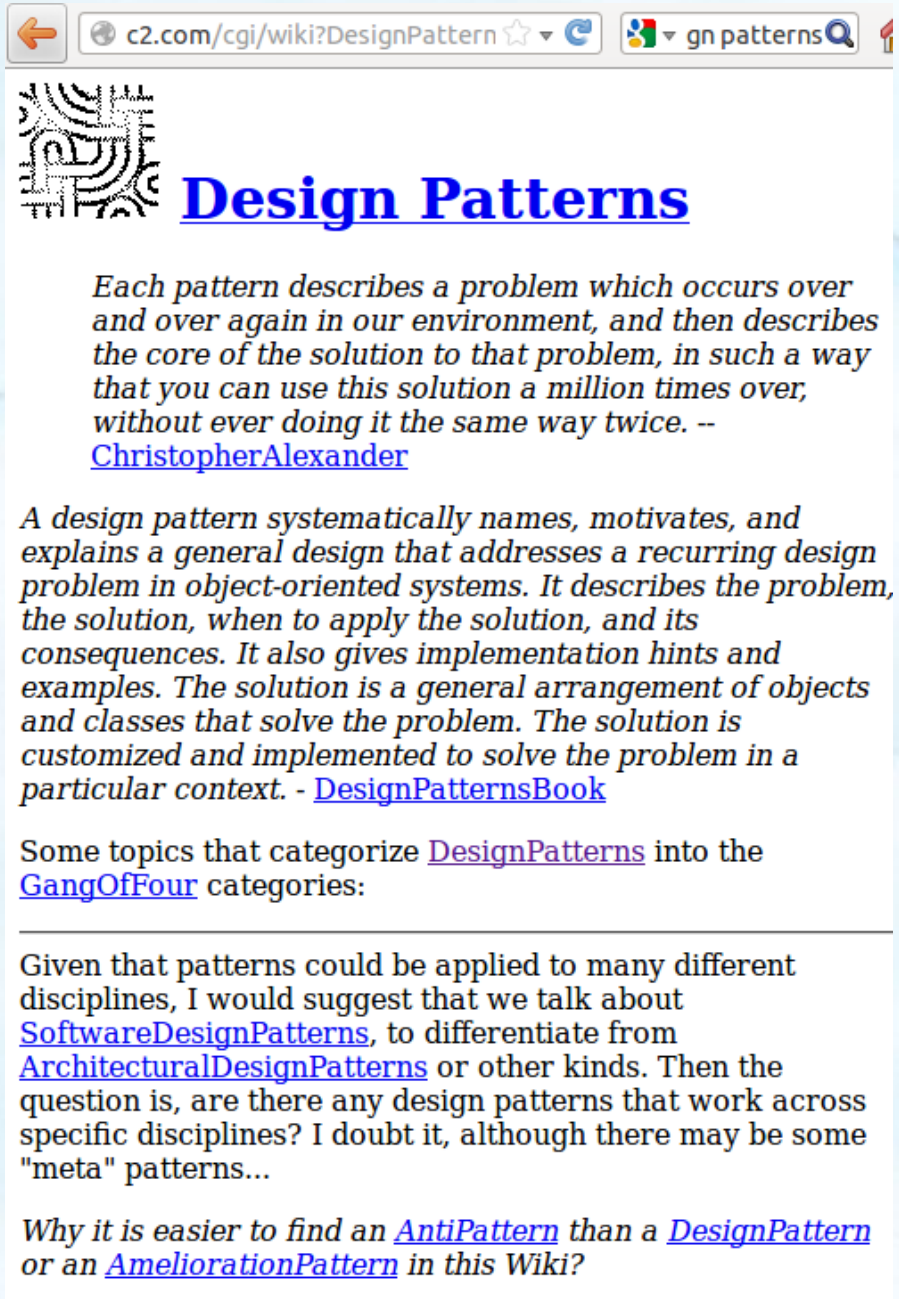
1. What could
   Service Systems Thinking be?

2. From architectural programming
   to a generative pattern language

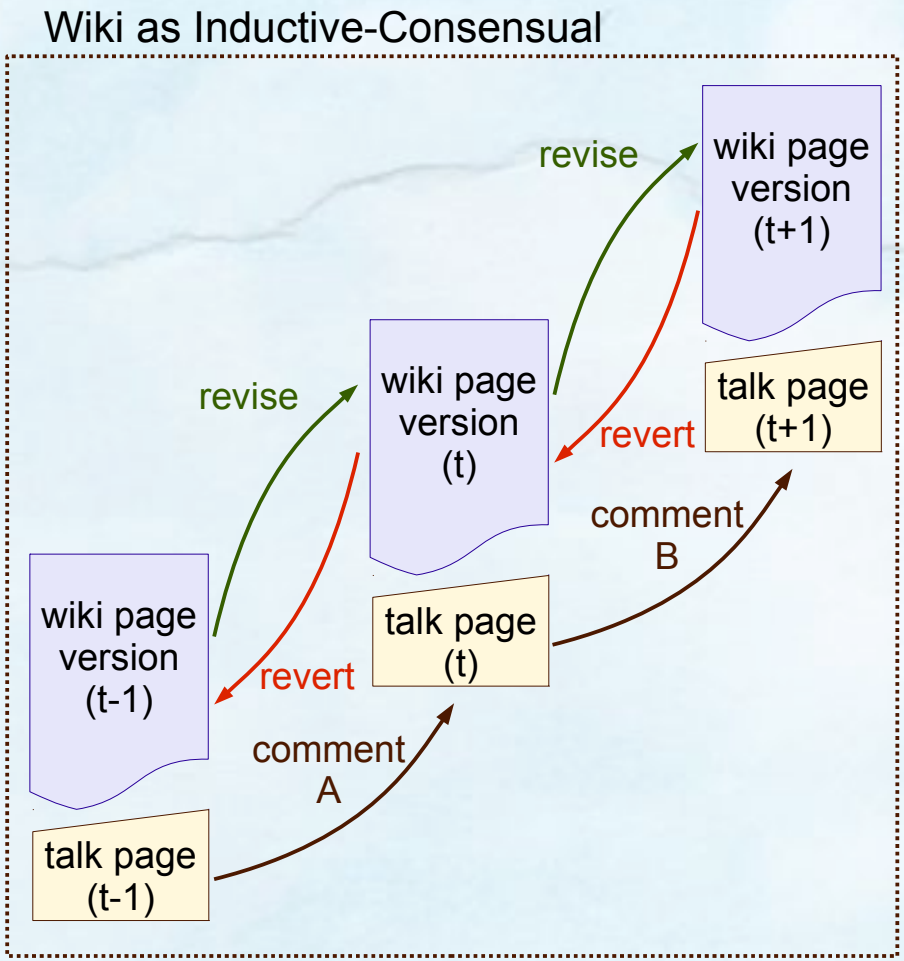
3. Federated wiki and
   a value constellation ontology?

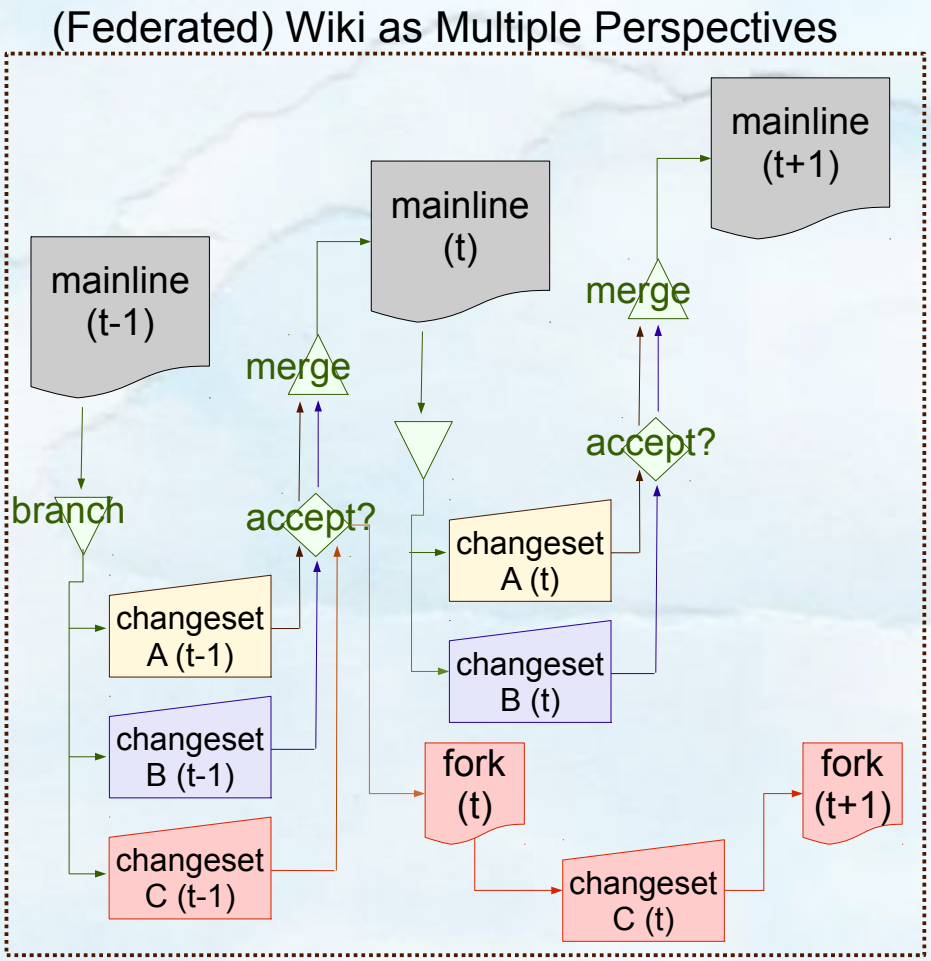
4. An invitation

Wiki was invented to support pattern language collaborations



Inductive-consensual Wiki revise-revert cycles become Federated Wiki perspectives, branch-merge or fork





Source: Mitroff, Ian I., and Richard O. Mason. 1982. "Business Policy and Metaphysics: Some Philosophical Considerations." *The Academy of Management Review* 7 (3) (July 1): 361–371. doi:10.2307/257328. http://www.jstor.org/stable/257328.

newer, older

Welcome Visitors

Welcome to the Smallest Federated Wiki. This page was first drafted Sunday, June 26th, 2011. The pages on this particular site have been edited to describe how to get things done on many of the federated sites.

Featured Sites

sites.fed.wiki.org
A catalog of federated wiki sites with domain names for page titles and brief descriptions tuned to look good in search results. Know your federation.

Topic Based Subsets
We pick topics that have been of lasting interest and subset them into their own federated wiki sites. We've built this feature into c2 wiki's Subset Wiki bridge and only use it here. github

Learn More

Read a little bit of How To Wiki. Then move on to our Sandbox and give your new knowledge a workout. Still confused? Look for answers in our Frequently Asked Questions, updates in Recent Changes.

Smallest Federated Wiki

Our new wiki innovates three ways. It shares through federation, composes by refactoring and wraps data with visualization. Follow our open development on GitHub or just watch our work in progress videos here.

We introduce the parts of a Federated Wiki page. The "story" is a collection of paragraphs and paragraph like items. The "journal" collects story edits. Should you take my page and edit it as yours, I can see what you've done and may decide to take your edits as my own.

Generative Pattern Language

While the label "pattern language" has been appropriated for a variety of contexts, the label of "generative pattern language" can be used for the "purer" thinking originating from the Center for Environmental Structure at U.C. Berkeley.

Christopher Alexander and his colleagues have a significant body of artifacts since the formation of the CES in 1967.

Pattern Manual (1967) is a charter for the CES.

A Pattern Language Which Generates Multi-Service Centers (1968) demonstrates how a pattern language could become instantiated differently for a variety of sites and circumstances.

"Systems Generating Systems (1968)" articulates the ties between a pattern language and systems thinking.

The Battle for Life and Beauty of the Earth (2012) is a history of a development project for the Eishin campus in Japan, demonstrating the CES vision from start to finish.

The variety of Current Applications of Pattern Languages often don't reflect the full vision of

A Pattern Language Which Generates Multi-Service Centers (1968)

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1968. *A Pattern Language Which Generates Multi-Service Centers*. Center for Environmental Structure. preview on Google Books

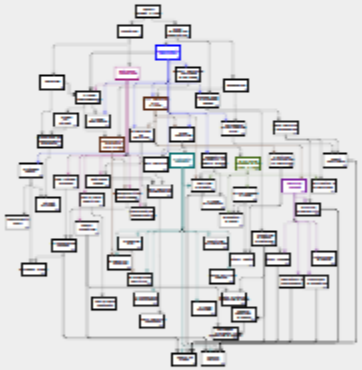
Introduction (Alexander et. al. 1968)

I. Summaries of 64 Patterns (Alexander et al. 1968)

II. The Idea of a Pattern (Alexander et al. 1968)

III. Eight Buildings Generated by the Pattern Language (Alexander et al. 1968)

IV. The Language (Alexander et. al 1968)

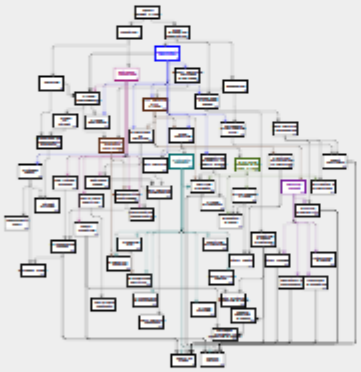


http://coevolving.com/maps/fed/Service-Centers-cascade.svg

This page is part of Historic Works on Generative Pattern Languages

A Pattern Language Which Generates Multi-Service Centers (1968)

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1968. *A Pattern Language Which Generates Multi-Service Centers*. Center for Environmental Structure. preview on Google Books



http://coevolving.com/maps/fed/Service-Centers-cascade.svg

This page is part of Historic Works on Generative Pattern Languages

The Idea of a Pattern (Alexander et al. 1968)

If we examine the patterns as they are presented in full, in the Appendix, we shall see that each pattern has two parts: the PATTERN statement itself, and a PROBLEM statement. The PATTERN statement is itself broken down into two further parts, an IF part, and a THEN part. In full the statement of each pattern reads like this:

IF:X THEN:Z / PROBLEM:Y

X defines a set of conditions. Y defines some problem which is always liable to occur under the conditions Z. Z defines some abstract spatial relation which needs to be present under the conditions X, in order to solve the problem Y.

In short, IF the conditions X occur, THEN we should Z, in order to solve the PROBLEM Y. [p. 17]

This page is part of A Pattern Language Which Generates Multi-Service Centers (1968)

Summaries of 64 Patterns (Alexander et al. 1968)

Each pattern prescribes some feature of a multi-service center building. It describes a relationship which is required to solve a problem which will occur in that building. The summary does not describe this problem; it describes only the pattern. [...] [p. 5]

1. Small Target Areas (1968): The multi-service center servces a target area with population of 34,000 ± 20%.

2. Location (1968): Service centers are located within two blocks of a major intersection.

3. Size Based on Population (1968): The total size of an MSC which services a target area of population N, is .9N square feet.

4. Community Territory (1968): The service center is divided into two zones, services and community territory; community territory includes space for community projects and a public area.

5. Small Services without Red Tape (1968): No one service has a staff size greater than 12; each service is physically cohesive and autonomous; the services are loosely organized with respect to each other.

Community Territory (1968)

PATTERN

IF: Any multi-service center,

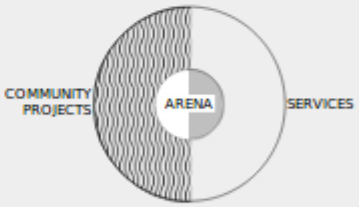
THEN:

1. The building should contain a major area which is established as community territory.

2. Community territory is distinct from the area devoted to services, but is interlocking with it.

3. Community territory contains two main components: An arena, and an area given over to community projects.

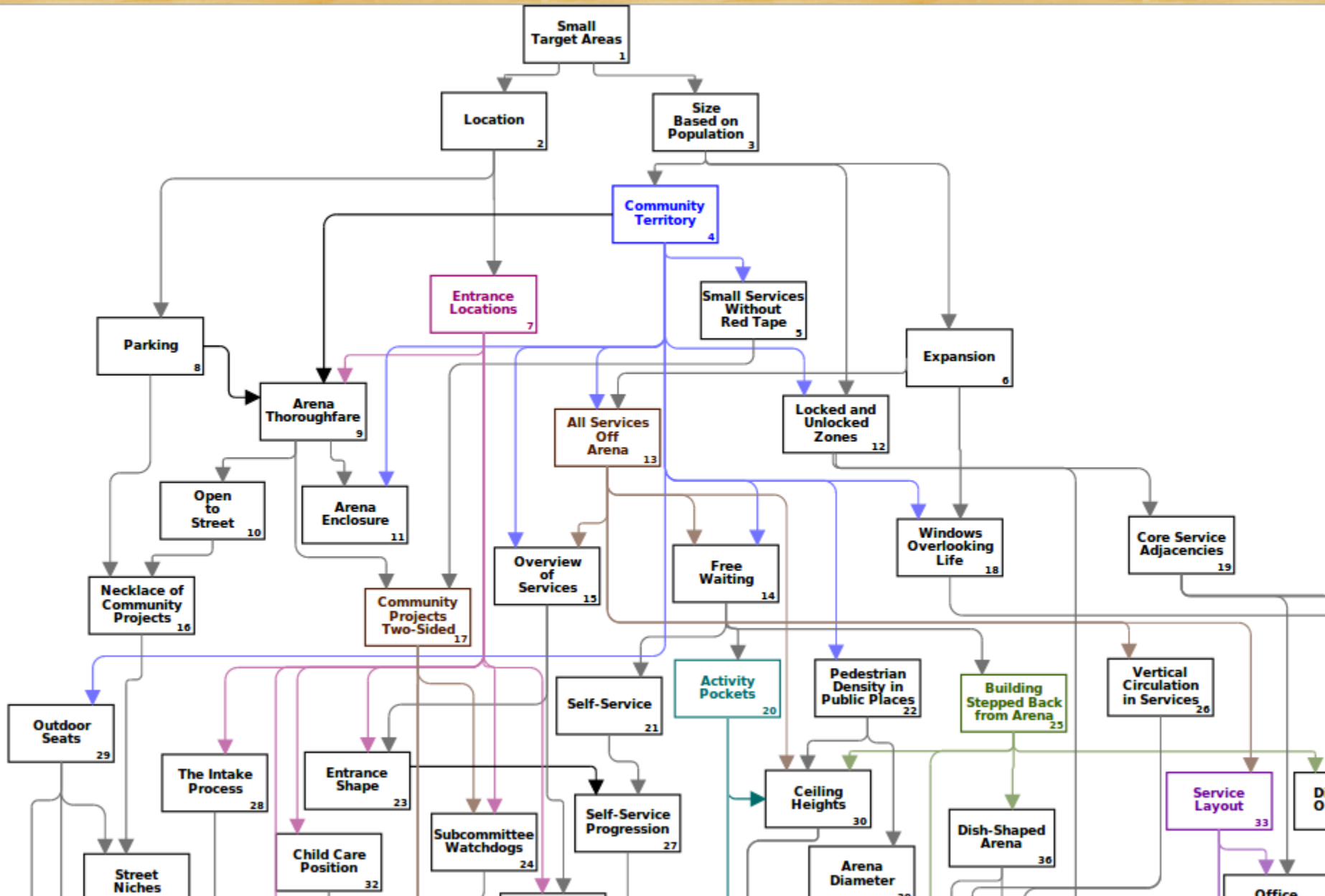
The arena is a public area, open to passers-by (whether or not they are visiting the service center), shaped in such a way as to encourage public discussions (both formal and informal), equipped with walls for day to day notices and poster, microphones, and loudspeakers. [p. 80]



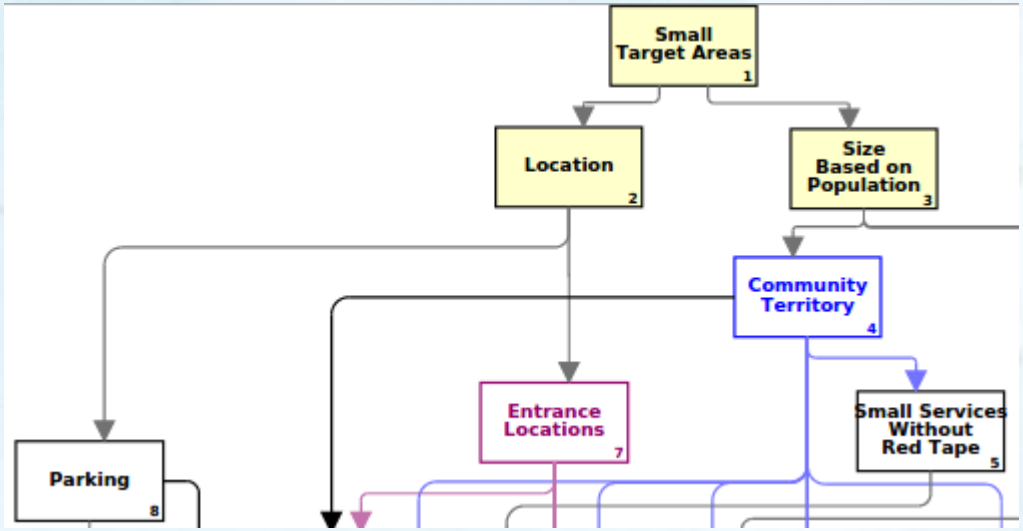
http://coevolving.com/maps/fed/Service-Centers_Community-Territory-figure.svg

A Pattern Language Which Generates Multi-Service Centers (1968)

coevolving.com/maps/fed/1968_Multi-Service-Centers-cascade.svg

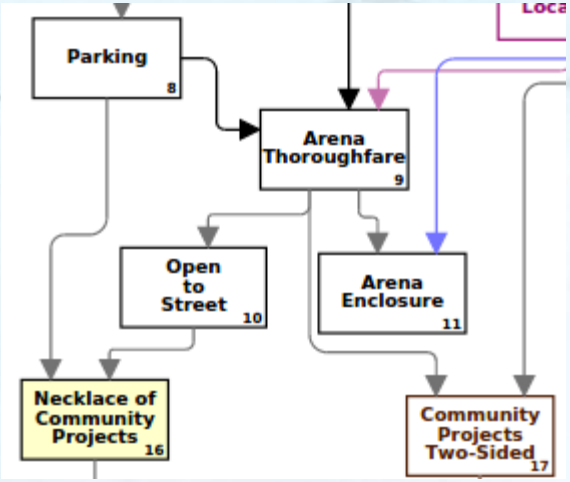


Hunts Point [page 1 of 4]: 40,000 people -- Strong community corporation -- Large block worker program -- 9 to 12 services -- Site open to three sides -- Near major intersection and transit station



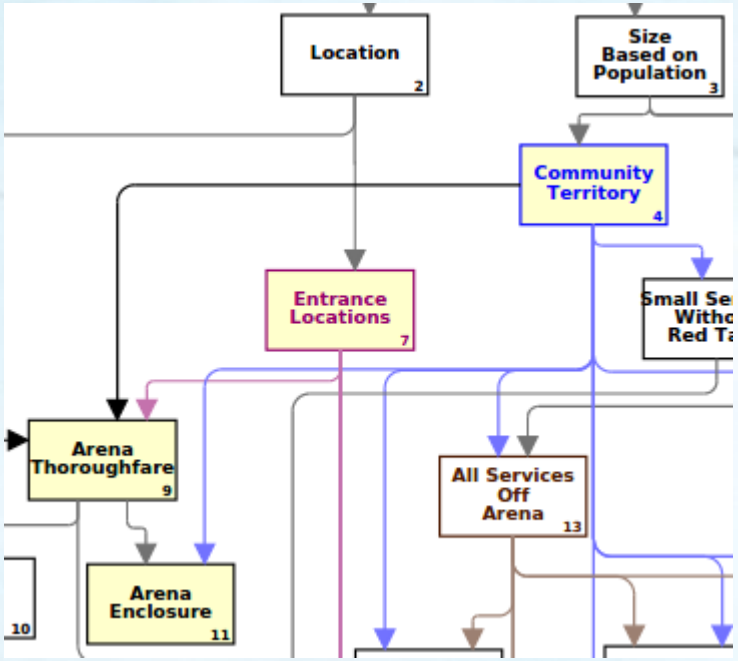
A
This multi-service center is to service 40,000 people. According to Pattern 1 Small Target Areas, this population is too large, but for political reasons, the decision stands and is irrevocable. First a triangle site was selected, right on a major intersection (Pattern 2: Location (1968)). However, other requirements made it clear that this site was too small (Pattern 3 Size Based on Population (1968)), and a larger, rectangular site was chosen, one-half block from the original site (thus still conforming to Pattern 2 Location (1968)).
On this site there was room only for emergency parking, and so Pattern 8 (Parking) does not play a major role. Nor does 5 Small Services without Red Tape, which had not been formulated prior to the Hunts Point Design.

Hunts Point [page 2 of 4]: 40,000 people -- Strong community corporation -- Large block worker program -- 9 to 12 services -- Site open to three sides -- Near major intersection and transit station



B
Pattern 16 (Necklace) calls for provisions for community projects around the "live" edge of the building; hence we confine services to the "dead" edge of this building, against other buildings.

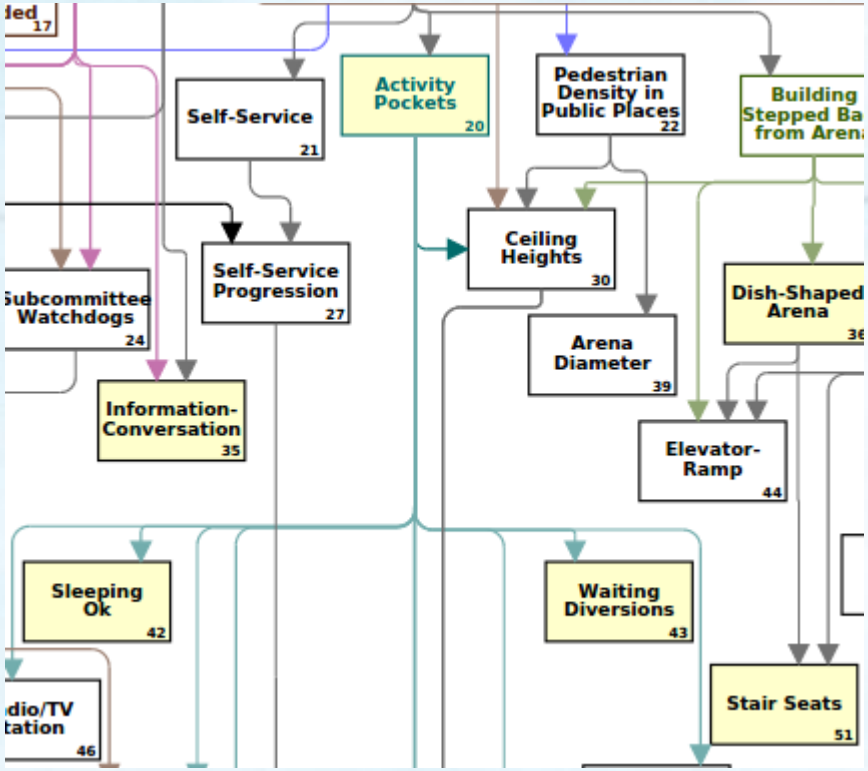
Hunts Point [page 3 of 4]: 40,000 people -- Strong community corporation -- Large block worker program -- 9 to 12 services -- Site open to three sides -- Near major intersection and transit station



C
Climate considerations made it clear that the arena could not be open (11: Arena Enclosure), and so it was developed as an interior street. Orientation of this "street" is given by local conditions in accordance with Pattern 7 (Entrance Locations).

D
The size of the arena and its relationship to waiting and services is established by Patterns 13 (All Services Off Arena), 14 (Free Waiting) and 15 (Overview of Services); and the arena is shaped accordingly.

Hunts Point [page 4 of 4]: 40,000 people -- Strong community corporation -- Large block worker program -- 9 to 12 services -- Site open to three sides -- Near major intersection and transit station



G
Finally, "pockets" in the arena are shaped and filled according to Patterns 29 (Activity Pockets), 35 (Information-Conversation), 43 (Waiting Diversions), and 42 (Sleeping Ok) (Alexander, Ishikawa, and Silverstein 1968, 22).

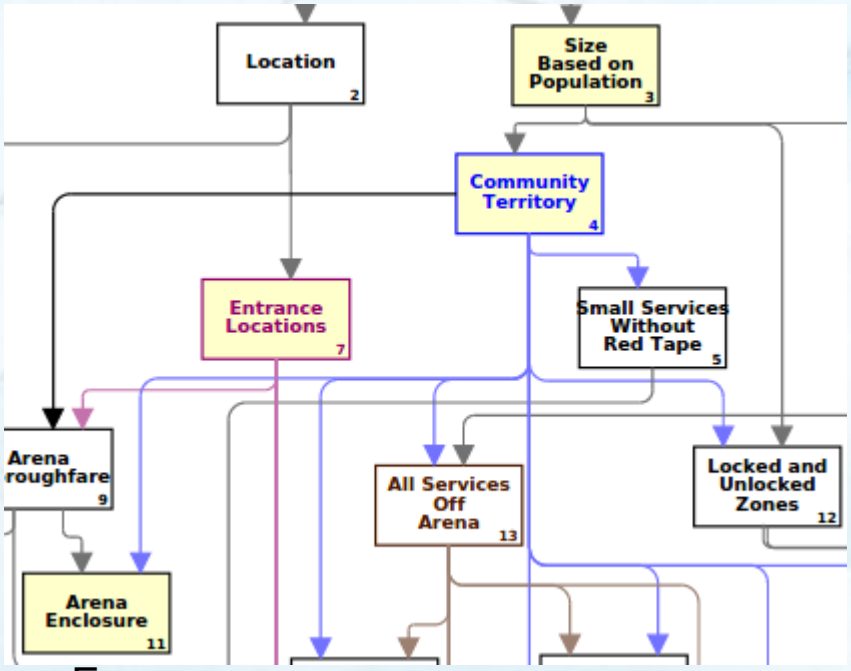
San Francisco: "Combination service and recreation center -- Mild climate -- Outdoor arena -- Strong community organization -- Corner site -- Off site parking provided"

A

To make the recreation part of the building highly accessible, the whole ground floor is devoted to recreation activity -- this area will be open late, according to Pattern 12 (Locked and Unlocked Zones); also it is highly visible from the street (10 Open to Street), and provides a thoroughfare (Pattern 9 Arena Thoroughfare). In this climate, the arena, which can be open to the sky (11 Arena Enclosure) takes on an unusual character -- it becomes a park. The whole ground floor becomes community territory (4 Community Territory).

B

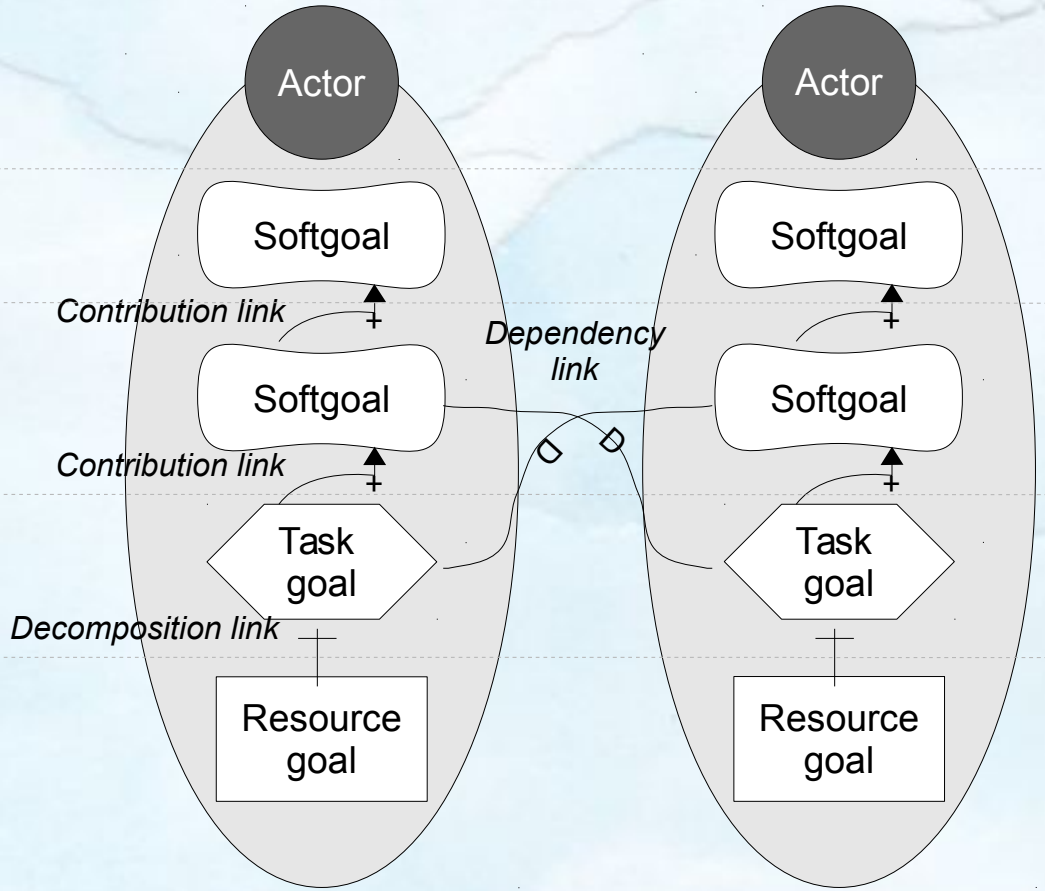
The recreation ware, which will become the hang-out for many members of the community, gives the building a natural base for community organization. It is therefore essential to put information, and community organizers and community projects at ground level. Patterns 17 (Community Projects Two-Sided), 28 The Intake Process), 35 (Information-Conversation) and 16 (Necklace of Community Projects) put them into the positions shown. [….]



E

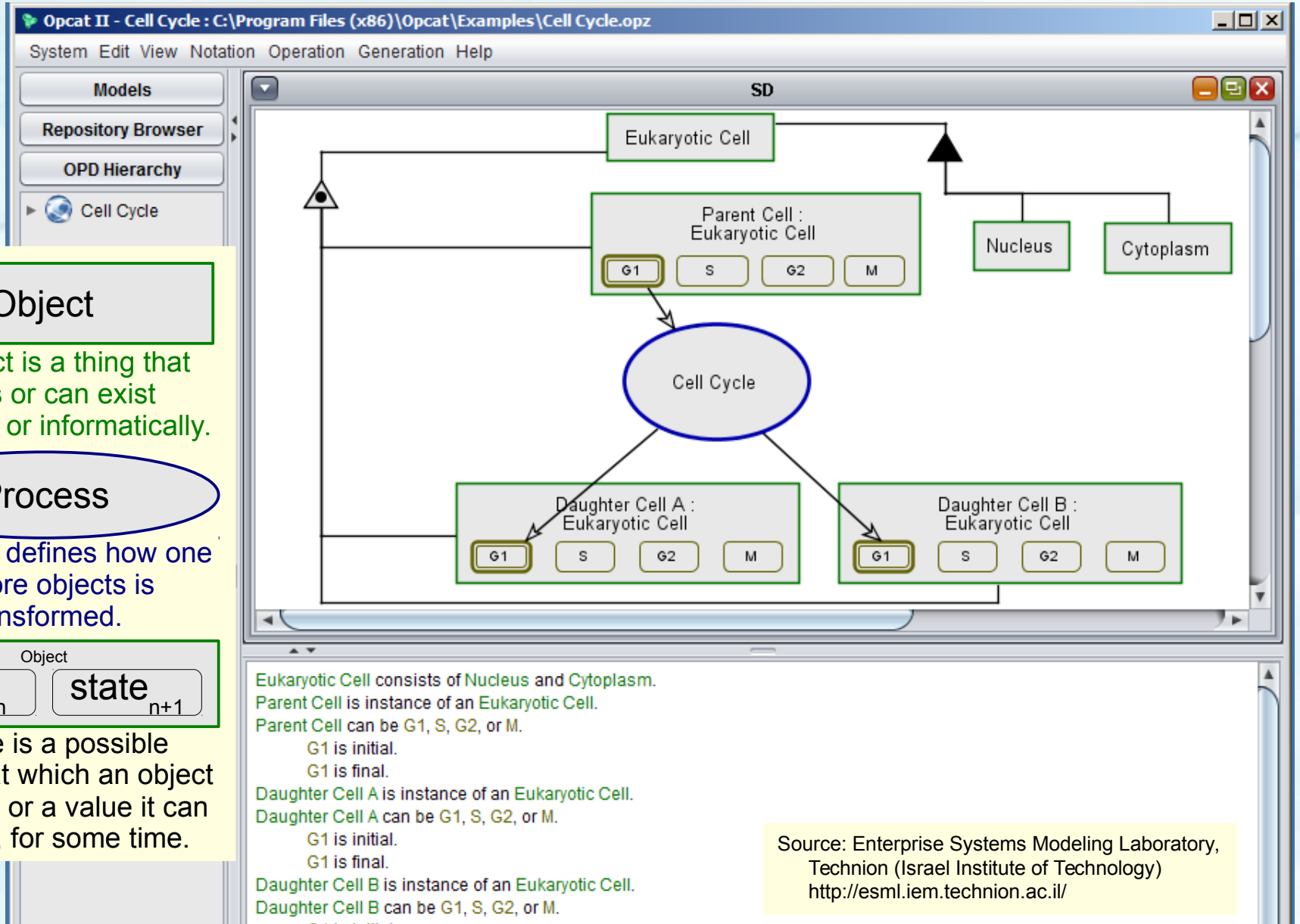
To get windows overlooking life (18 Windows Overlooking Life), there are holes from the second and third story, looking down into the recreation floor (Alexander, Ishikawa, and Silverstein 1968, 26).

Key concepts of value cocreation can be expressed through intentional (iStar) modeling constructs

Key service system concepts	*i* constructs*
Service system entity	
High-level interests	
Expected benefits	
Value propositions	
Resources	

Source: Lysanne Lessard and Eric Yu. 2013. “Service Systems Design: An Intentional Agent Perspective.” *Human Factors and Ergonomics in Manufacturing & Service Industries* 23 (1): 68–75. doi:10.1002/hfm.20513.

Object Process Methodology came from systems engineering, with new examples in biology



Object

An object is a thing that exists or can exist physically or informatically.

Process

A process defines how one or more objects is transformed.

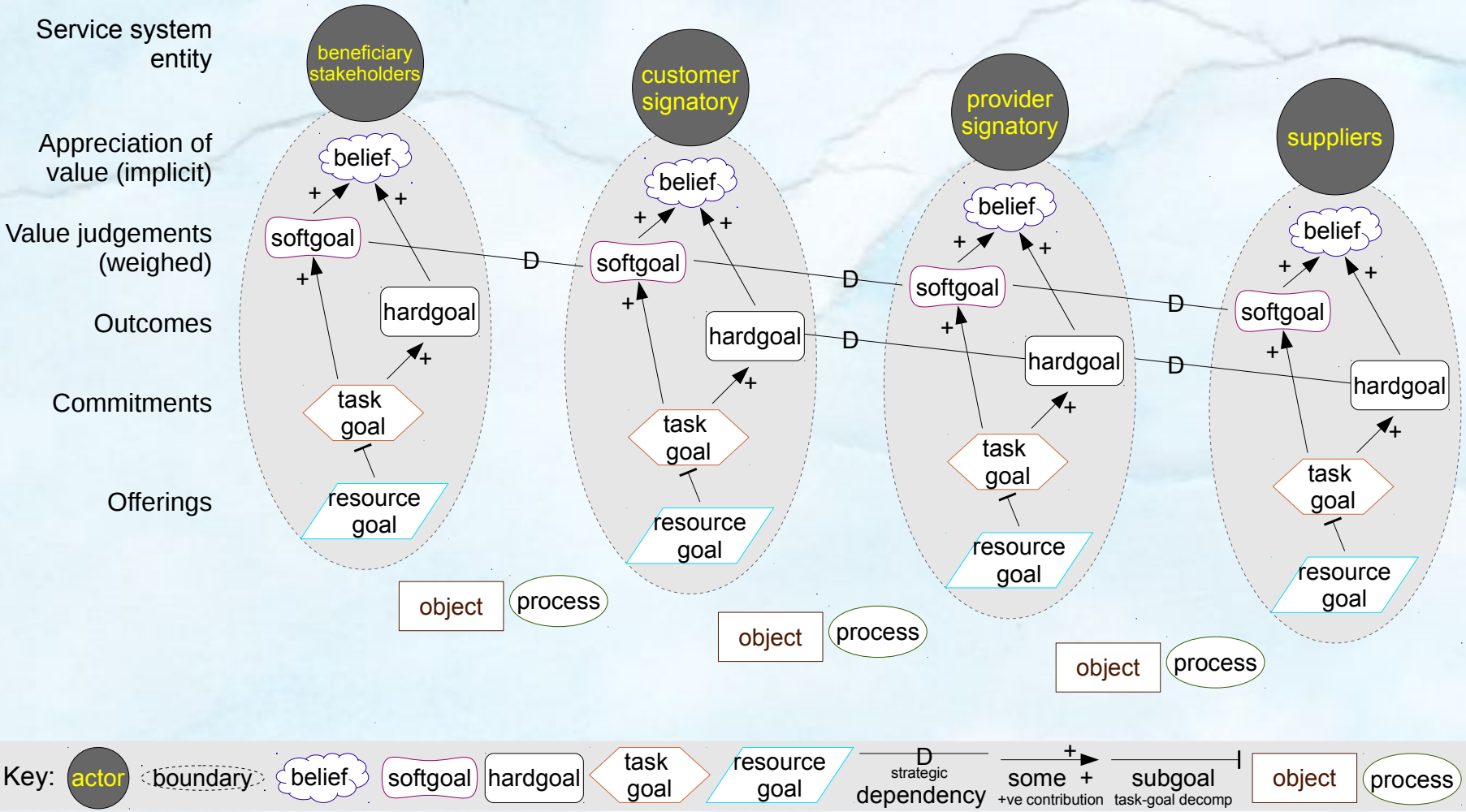
Object

$state_n$ $state_{n+1}$

A state is a possible situation at which an object can be in, or a value it can assume, for some time.

Source: Enterprise Systems Modeling Laboratory, Technion (Israel Institute of Technology) http://esml.iem.technion.ac.il/

Could we model value constellation ontology synthesizing iStar and OPM representations?



Agenda

1. What could
   Service Systems Thinking be?

2. From architectural programming
   to a generative pattern language

3. Federated wiki and
   a value constellation ontology?

4. An invitation

Seeking concurrence

- International Workshop, Jan 2014, Los Angeles
- International Symposium, June 2014, Las Vegas

- Human Side of Service Engineering, July 2014, Krakow

- ISSS 58th Annual Meeting, July 2014, Washington, DC

- Pattern Languages of Programming Conference, September 2014, Allerton, IL

- Relating Systems Thinking and Design Symposium, October 2014, Oslo

- PURPLSOC Pursuit of Pattern Languages for Societal Change Conference, July 2015, Krems

Here is a short and necessarily incomplete definition of a pattern:

A recurring structural configuration that solves a problem in a context, contributing to the wholeness of some whole, or system, that reflects some aesthetic or cultural value.[1]

Pattern Name: A name by which this problem/solution pairing can be referenced

Problem
The specific problem that needs to be solved.

Context
The circumstances in which the problem is being solved imposes constraints on the solution. The context is often described via a "situation" rather than stated explicitly.

Forces
The often contradictory considerations that must be taken into account when choosing a solution to a problem.

Solution
The most appropriate solution to a problem is the one that best resolves the highest priority forces as determined by the particular context.

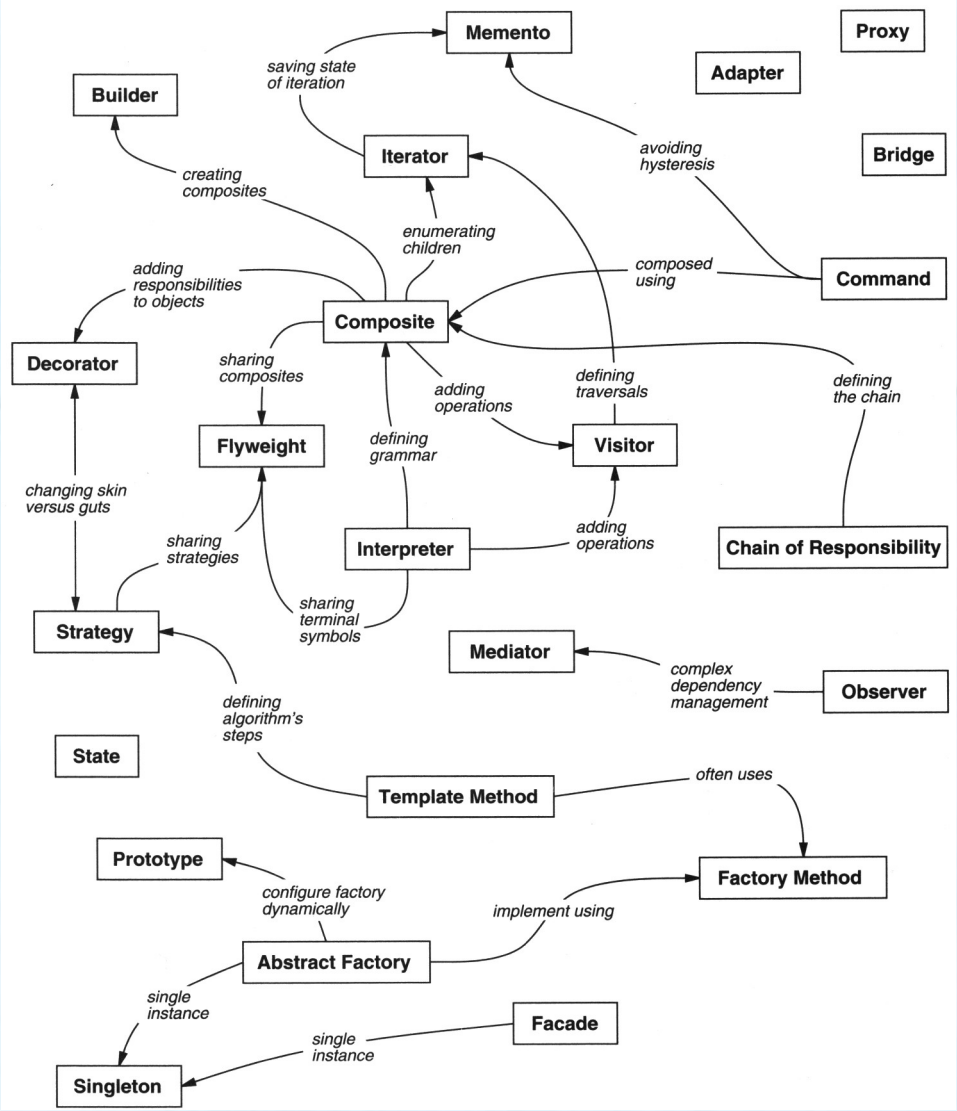
Resulting Context
The context that we find ourselves in after the pattern has been applied. It can include one or more new problems to solve

Rationale
An explanation of why this solution is most appropriate for the stated problem within this context.

Related Patterns
The kinds of patterns include:
- Other solutions to the same problem,
- More general or (possibly domain) specific variations of the pattern,
- Patterns that solve some of the problems in the resulting context (set up by this pattern)

Source: [1] Coplien, James O., and Neil B. Harrison. 2004. Organizational Patterns of Agile Software Development. Prentice-Hall, Inc. http://books.google.ca/books?id=6K5QAAAAMAAJ . [2] Gerard Meszaros and Jim Doble, “A Pattern Language for Pattern Writing”, Pattern Languages of Program Design (1997), http://hillside.net/index.php/a-pattern-language-for-pattern-writing

Design Patterns (Catalog)



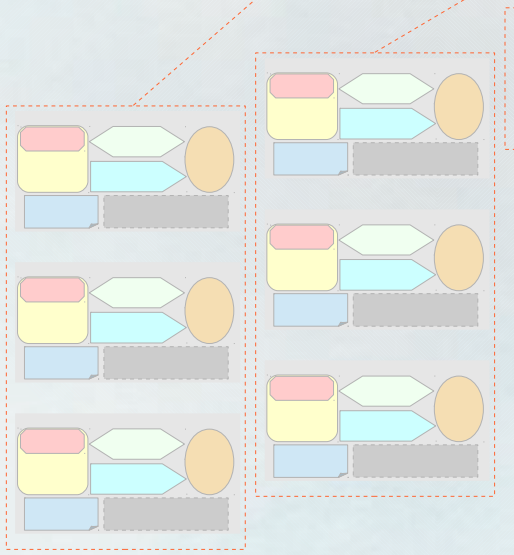
Purpose	Design Pattern	Aspects That Can Vary
Creational	Abstract Factory	families of product objects
	Builder	how a composite object gets created
	Factory Method	subclass of object that is instantiated
	Prototype	class of object that is instantiated
	Singleton	the sole instance of a class
Structural	Adapter	interface to an object
	Bridge	implementation of an object
	Composite	structure and composition of an object
	Decorator	responsibilities of an object without subclassing
	Facade	interface to a subsystem
	Flyweight	storage cost of objects
	Proxy	how an object is accessed; its location
Behavioral	Chain of Responsibility	object that can fulfill a request
	Command	when and how a request is fulfilled
	Interpreter	grammar and interpretation of a language
	Iterator	how an aggregate's elements are accessed, traversed
	Mediator	how and which objects interact with each other
	Memento	what private information is stored outside an object, and when
	Observer	number of objects that depend on another object; how the dependent objects stay up to date
	State	states of an object
	Strategy	an algorithm
	Template Method	steps of an algorithm
	Visitor	operations that can be applied to object(s) without changing their class(es)

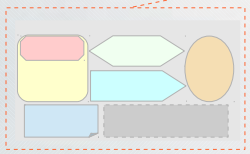
Source: Erich Gamma,Richard Helm, Ralph Johnson, and John Vlissides. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. http://books.google.ca/books?id=6oHuKQe3TjQC .

To appreciate service systems, can we aspire beyond a (Design) Pattern Catalog to a Generative Pattern Language?

(Design) Pattern Catalog

SCOPE	PURPOSE		
	Creational	Structural	Behavioral
Class	Factory Method	Adapter	Interpreter Template Method
Object	Abstract Factory Builder Prototype Singleton	Factory Method Bridge Composite ...	Chain of Responsibility Command Iterator ...





Pattern Name

Problem

Context

Forces

Solution

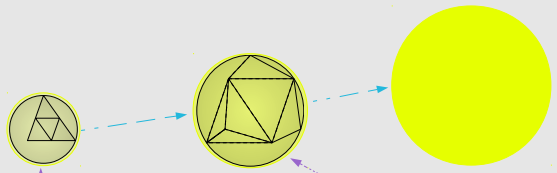
Resulting Context

Rationale

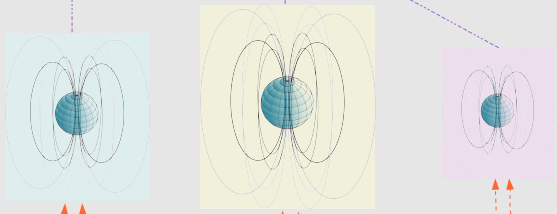
Related Patterns

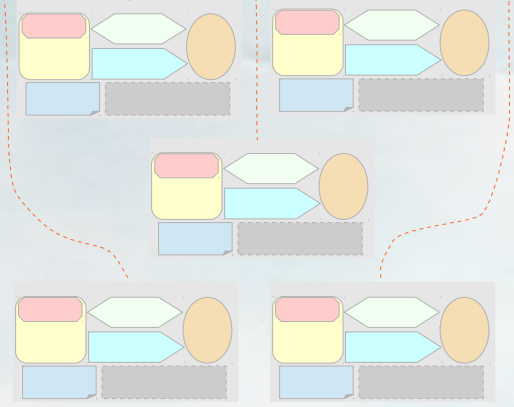
Generative Pattern Language

Unfolding wholeness (+ interactive value?)



Centres and spaces, in layers and paces





Agenda

1. What could
   Service Systems Thinking be?

2. From architectural programming
   to a generative pattern language

3. Federated wiki and
   a value constellation ontology?

4. An invitation