A Proposal for

Collaboration on a Pattern Language for Service Systems

(Science, Management, Engineering and Design)

David Ing
International Society for the Systems Sciences,
Aalto University, and Healthcare EQ Inc.

January 2014

Agenda

A. Service systems
(science, management, engineering and design)

B. Pattern language
(c.f. pattern catalog)

C. A starter set?
- 7 conditions from service systems science

D. Collaboration
- Inquiring system
- Technologies

E. Next steps

Agenda

- A. Service systems (science, management, engineering and design)
- B. Pattern language (c.f. pattern catalog)
- C. A starter set?
  - 7 conditions from service systems science
- D. Collaboration
  - Inquiring system
  - Technologies
- E. Next steps

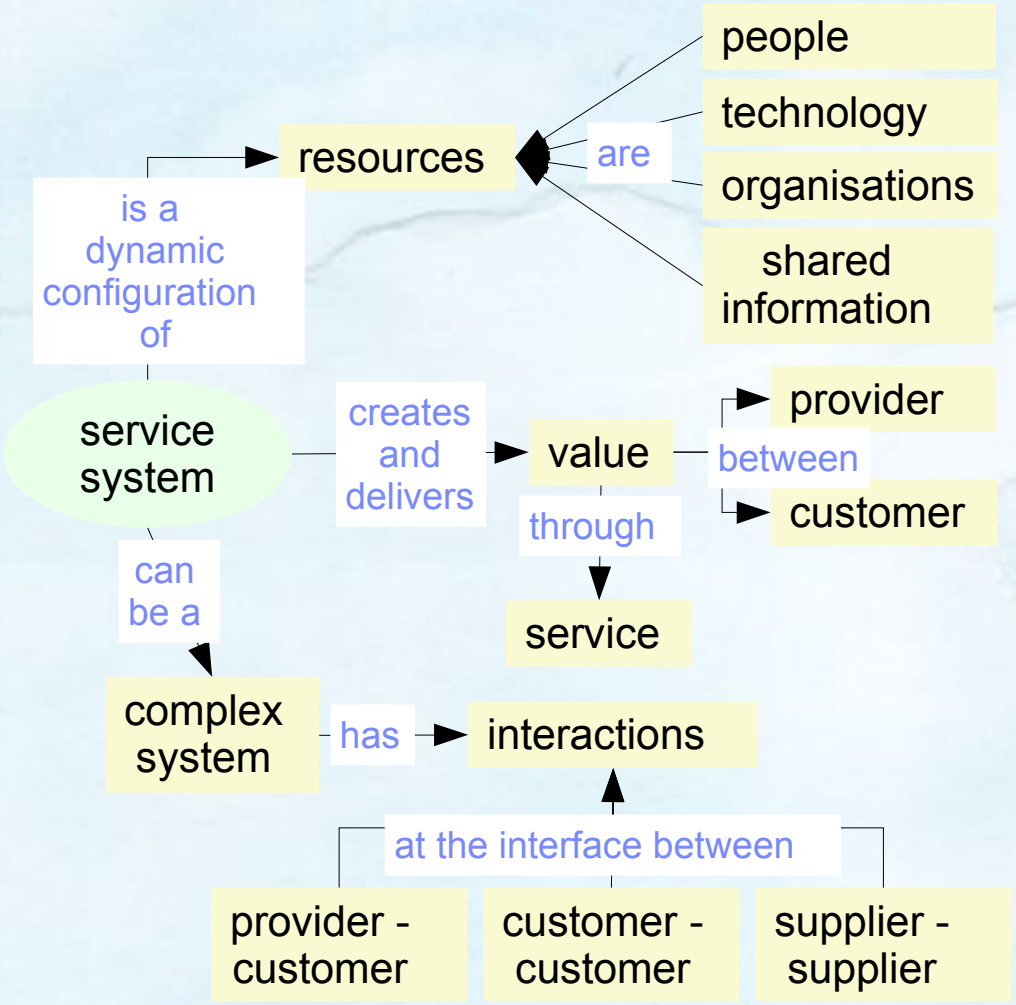
1. How is systems science approaching service systems?
2. What does science mean to a systems thinker?
3. What is systems thinking?

Human civilization is served by systems in technical, organizational and socio-political form

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

[Spohrer and Maglio 2010]

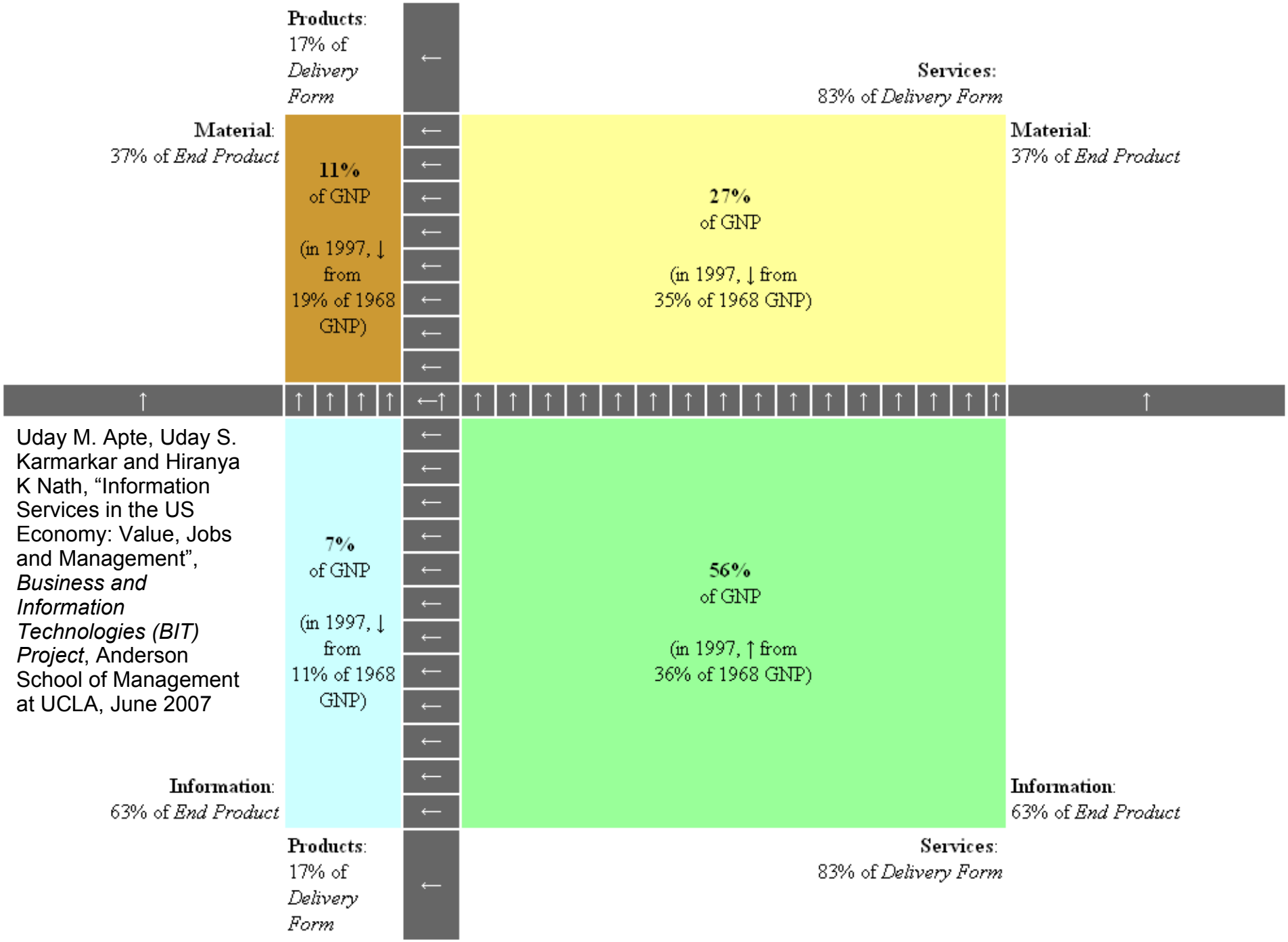
Service systems (Cambridge IfM and IBM, 2008)



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.

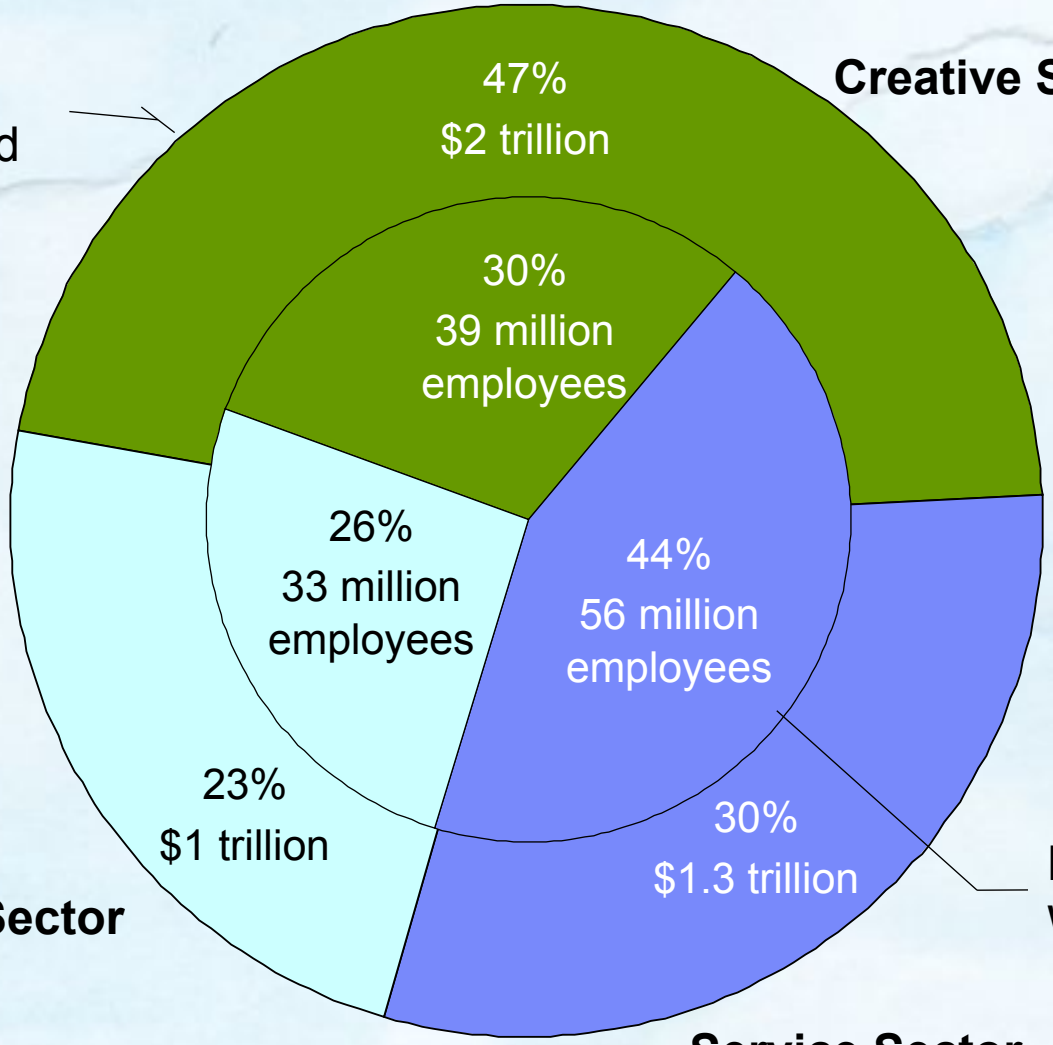
In many cases, a service system is a complex system in that configurations of resources interact in a non-linear way. Primary interactions take place at the interface between the provider and the customer. However, with the advent of ICT, customer-to-customer and supplier-to-supplier interactions have 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/ .



Uday M. Apte, Uday S. Karmarkar and Hiranya K Nath, "Information Services in the US Economy: Value, Jobs and Management", *Business and Information Technologies (BIT) Project*, Anderson School of Management at UCLA, June 2007

Creative class generates greater wealth per employee



Source:
Richard L. Florida (2004) *The flight of the creative class : The new global competition for talent.*

Basic Concepts. If we are to understand human history as the evolution and design of value-cocreation mechanisms between entities, then where should we begin?

Let's start by understanding the following ten basic concepts:

1.	Resources	Businesses may own physical resources or contract for physical resources, but as a type of resource they are themselves not physical, but instead a conceptual-legal construct. So in the end, all resources fall into one of four types: *physical-with-rights, not-physical-with-rights, physical-with-no-rights, and not-physical-with-rights.*
2.	Service system entities	The most common types of service system entities are people and organizations. New types of service system entities are constantly emerging and disappearing. Recently, open-source and on-line communities have emerged as service systems entities.
3.	Access rights	"By what authority, do you use that resource?" Service system entities have four main types of access rights to the resources within their configuration: *owned outright*, *leased/contracted*, *shared access*, and *privileged access*. Shared access resources include resources such as air, roads, natural language, and internet web sites. Privileged access resources include resources such as thoughts, individual histories, and family relationships.
4.	Value-proposition-based interactions	"I'll do this, if you'll do that." [….] Interactions via value propositions are intended to cocreate-value for both interacting entities. Both interacting entities must agree, explicitly or tacitly, to the value proposition.
5.	Governance mechanisms	"Here's what will happen if things go wrong." [….] If value is not realized as expected, this may result in a dispute between the entities. Governance mechanisms reduce the uncertainty in these situations by prescribing a mutually agreed to process for resolving the dispute.
6.	Service system networks	"Here's how we can all link up." [….] Over time, for a population of entities, the patterns of interaction can be viewed as networks with direct and indirect connectivity strengths. A service system network is an abstraction that only emerges when one assumes a particular analysis overlay on the history of interactions amongst service system entities.
7.	Service system ecology	"Populations of entities, changing the ways they interact." Different types of service systems entities exist in populations, and the universe of all service system entities forms the service system ecology or service world ….
8.	Stakeholders	"When it comes to value, perspective really matters." The four primary types of stakeholders are *customer*, *provider*, *authority*, and *competitor*. In addition … other stakeholder perspectives include employee, partner, entrepreneur, criminal, victim, underserved, citizen, manager, children, aged, and many others.
9.	Measures	"Without standardized measures, it is hard to agree and harder to trust." The four primary types of measures are *quality*, *productivity*, *compliance*, and *sustainable innovation*.
10.	Outcomes	"How did we do? Can this become a new routine or long-term relationship?" […] Beyond a standard two player game, with a customer player and a provider player, ISPAR assumes there exists both an authority player as well as a competitor-criminal player.

Source: Jim Spohrer and Stephen K. Kwan. 2009. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References." *International Journal of Information Systems in the Service Sector* 1 (3): 1–31. doi:10.4018/jisss.2009070101 .

Service systems worldview. These ten basic concepts underlie the *service systems worldview* ...

1. Resources
2. Service system entities
3. Access rights
4. Value-proposition-based interactions
5. Governance mechanisms
6. Service system networks
7. Service system ecology
8. Stakeholders
9. Measures
10. Outcomes

… the world is made up of
populations of service system entities that
interact (normatively) via
value propositions to cocreate-value, but often
disputes arise and so
governance mechanisms are invoked to resolve disputes.

Formal service system entities are types of legal entities with rights and responsibilities, that can own property, and with named identities that can create contracts with other legal entities. [….] Formal service systems exist within a legal and economic framework of contracts and expectations.

Informal service system entities include families ..., open source communities ..., and many other societal or social systems that are governed typically by unwritten cultural and behavioral norms (social systems with rudimentary political systems).

Natural history of service system entities. Service science seeks to create an understanding of the formal and informal nature of service in terms of entities, interactions, and outcomes, and how these evolve (or are designed) over time. An initial premise is that the entities, which are sophisticated enough to engage in rationally designed service interactions that can consistently lead to win-win value cocreation outcomes, must be able to build models of the past (reputation, trust), present, and future (options, risk-reward, opportunities, hopes and aspirations) possible worlds, including models of themselves and others, and reason about knowledge value ….

Source: Jim Spohrer and Stephen K. Kwan. 2009. “Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References.” *International Journal of Information Systems in the Service Sector* 1 (3): 1–31. doi:10.4018/jisss.2009070101 .

Basic questions. A general theory of service system entities and networks formed through value-proposition-based interactions has four parts

… which directly lead to the four basic types of questions that SSMED seeks to answer.

Science

(improve understanding, map natural history, validate mechanisms, make predictions).

What are service system entities, how have they naturally evolved to present, and how might they evolve in the future? What can we know about their interactions, how the interactions are shaped (value propositions, governance mechanisms), and the possible outcomes of those interactions both short-term and long-term?

Management

(improve capabilities, define progress measures, optimize investment strategy).

How should one invest to create, improve, and scale service system networks? How do the four measures of quality, productivity, compliance, and sustainable innovation relate to numerous key performance indicators (KPIs) of business and societal systems? Is there a "Moore's Law" of service system investment? Can doubling information lead to a doubling of capabilities (performance) on a predictable basis?

Engineering

(improve control, optimize resources).

How can the performance of service system entities and scaling of service system networks be improved by the invention of new technologies (and environmental infrastructures) or the reconfiguration of existing ones? What is required to develop a CAD (Computer-Aided Design) tool for service system entity and service system network design?

Design

(improve experience, explore possibilities).

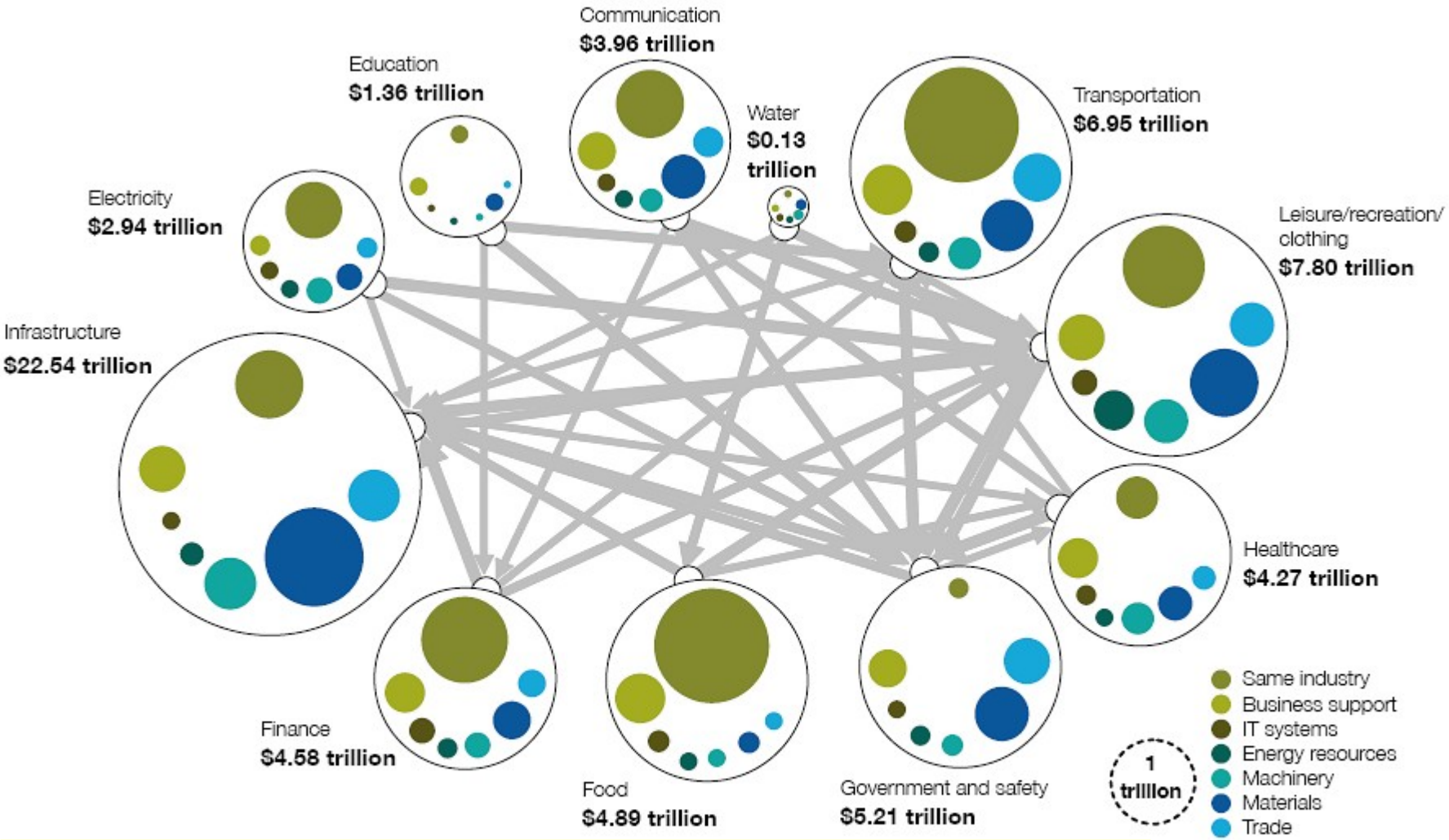
How can one best improve the experience of people in service system entities and networks? How can the experience of service system creation, improvement, and scaling be enhanced by better design? Can the space of possible value propositions and governance mechanisms be explored systematically?

Sciences of the artificial. Sciences of the artificial are different from natural sciences, and so it becomes especially important to consider these four parts – science, management, engineering, and design – as important knowledge components. In "The Sciences of the Artificial" (Simon 1996), Simon reflects "The world we live in today is much more man-made, or artificial, world than it is a natural world....

Service Science, Management, Engineering, and Design (SSMED) is emerging as one of the sciences of the artificial. Service science is knowledge about service system entities, value-proposition-based interactions (or value-cocreation mechanisms), governance mechanisms, and the other seven basic concepts. Following Simon even further, one could argue that service system entities are physical symbol systems, dealing with symbols that are named resources, and grounded in physical routines for carrying out the symbolic manipulations related to named resources.

Source: Jim Spohrer and Stephen K. Kwan. 2009. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References." *International Journal of Information Systems in the Service Sector* 1 (3): 1–31. doi:10.4018/jisss.2009070101 .

US$54 trillion system of systems -- IBM



http://www-935.ibm.com/services/us/gbs/bus/html/ibv-smarter-planet-system-of-systems.html.

Note: Size of bubbles represents systems' economic values. Arrows represent the strength of systems' interaction.
Source: IBM Institute for Business Value analysis of Organisation for Economic Co-operation and Development (OECD) data.

Figure 1: We live and work within a complex, dynamic and interconnected US$54 trillion system of systems.

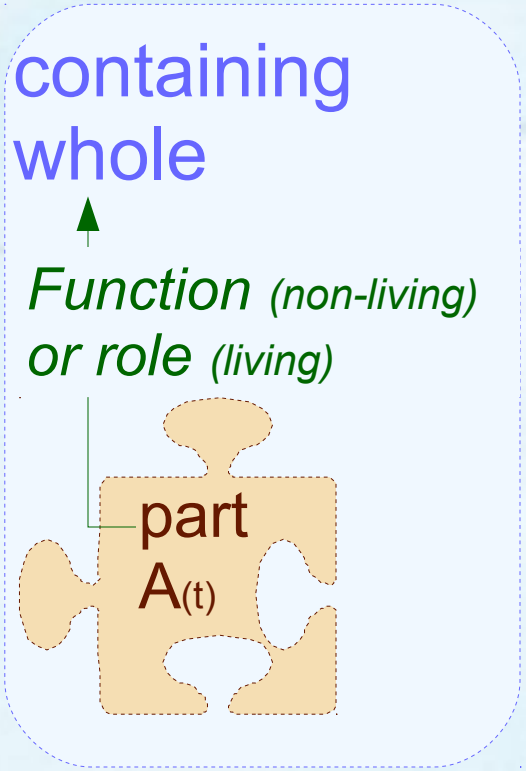
Defining systems science(s) → science?

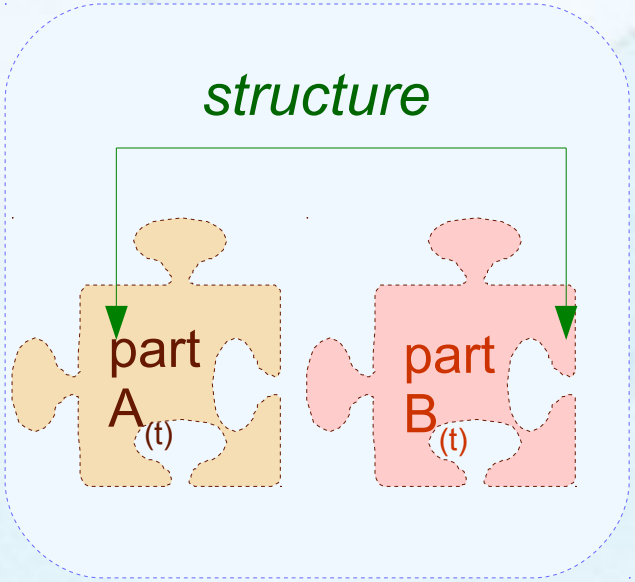
Primary intellectual virtue:	**Episteme**	**Techne**	**Phronesis**
Translation / interpretation:	Science (viz. epistemology)	Craft (viz. technique)	Prudence, common sense
Type of virtue:	Analytic scientific knowledge	Technical knowledge	Practical ethics
Orientation:	Research	Production	Action
Pursuits:	Uncovering universal truths	Instrumental rationality towards a conscious goal	Values in practice based on judgement and experience
Colloquial description:	**Know why**	**Know how**	**Know when, know where, know whom**

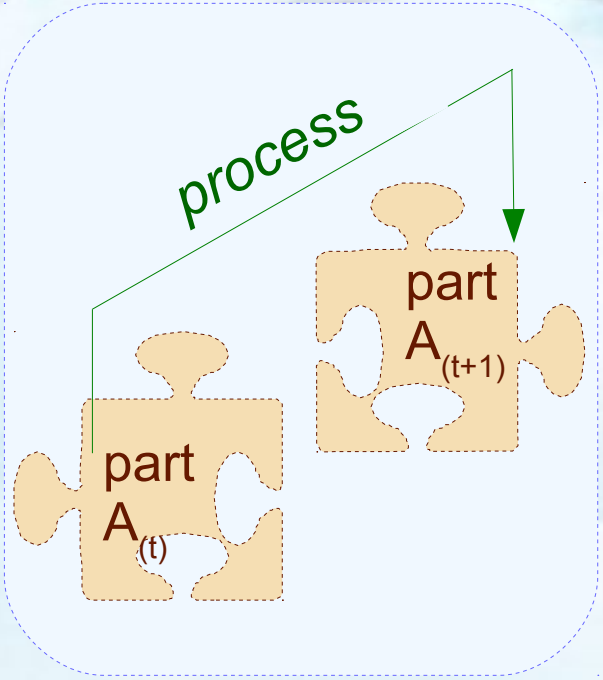
Defining systems science(s) → science?

Primary intellectual virtue:	**Episteme**	**Techne**	**Phronesis**
Translation / interpretation:	Science (viz. epistemology)	Craft (viz. technique)	Prudence, common sense
Type of virtue:	Analytic scientific knowledge	Technical knowledge	Practical ethics
Orientation:	Research	Production	Action
Nature:	Universal	Pragmatic	Pragmatic
	Invariable (in time and space)	Variable (in time and space)	Variable (in time and space)
	Context-independent	Context-dependent	Context-dependent
Pursuits:	Uncovering universal truths	Instrumental rationality towards a conscious goal	Values in practice based on judgement and experience
Colloquial description:	**Know why**	**Know how**	**Know when, know where, know whom**

Systems thinking is a perspective on wholes, parts and their relations





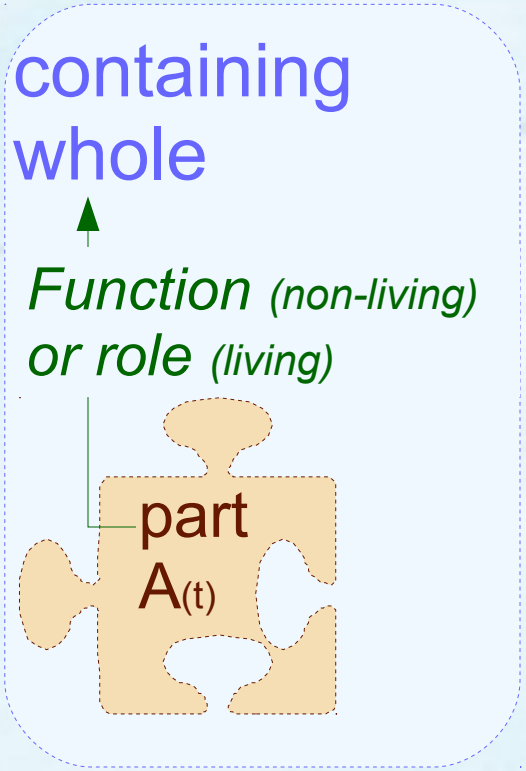


Function
"contribution of the part to the whole"

Structure
"arrangement in space"

Process
"arrangement in time"

Systems thinking: synthesis precedes analysis
(Ackoff 1981)



Synthesis precedes analysis
1. Identify a containing whole (system) of which the thing to be explained is a part.
2. Explain the behavior or properties of the containing whole
3. Then explain the behavior or properties of the thing to the explained in terms of its role(s) or function(s) within its containing whole.

Domains of systems thinking

Categories of systems thinking:	**Systems theory**	**Systems methods**	**Systems practice**
Primary intellectual virtue:	Episteme	Techne	Phronesis
Colloquial description:	Know why	Know how	Know when, know where, know whom
Systems thinking domains:	•Living systems theory •Hierarchy theory •Open Systems Theory •Viable System Model •Inquiring Systems •Critical Systems Theory •Panachy and ecological resilience	•System dynamics •Soft Systems Methodology •Interactive Planning •Action Research •Structured Dialogic Design •Strategic Assumption Surfacing and Testing •Search Conference •Deep Dialog	•Language Action Perspective •Appreciative Systems •Evolutionary Development •Systems Intelligence

Paths to develop systems thinking

Episteme (e.g. theoretical science, codified principles)	**Techne** (e.g. methods and techniques, collaboration)	**Phronesis** (e.g. hands-on experience, values in practice)	*Proposed path for learning and coevolving*	*Case domains*
☐ (weak)	✔ (strong)	✔ (strong)	Induction: **Why** are the natures or behaviours of systems similar or dissimilar?	Service systems?
✔ (strong)	☐ (weak)	✔ (strong)	Abduction: **How** are future systems to be developed or improved over current systems?	Ecosystems?
✔ (strong)	✔ (strong)	☐ (weak)	Deduction: **When, where** and for **whom** are systems material and/or salient?	Governing / policy systems?

Agenda

A. Service systems
(science, management, engineering and design)

B. Pattern language
(c.f. pattern catalog)

C. A starter set?
- 7 conditions from service systems science

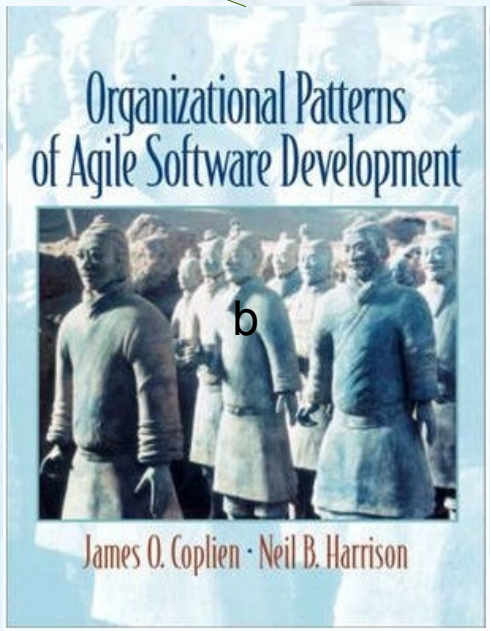
D. Collaboration
- Inquiring system
- Technologies

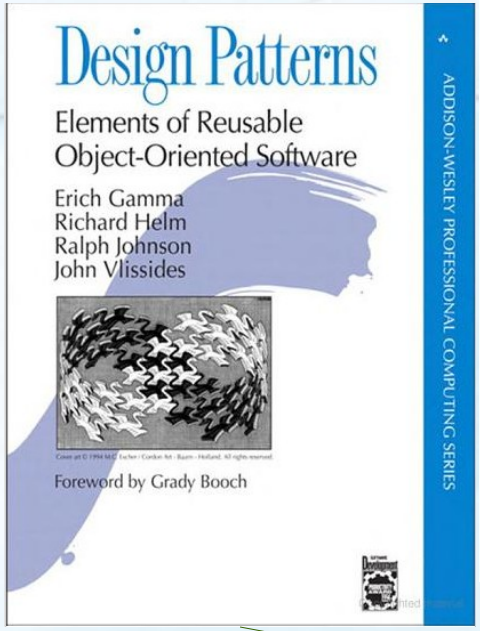
E. Next steps

1. An example: Intimacy Gradient
2. The Hillside Group (history)
3. Pattern language form
4. Pattern language and systems thinking?
   - The quality without a name
5. Domains with recent pattern language activity

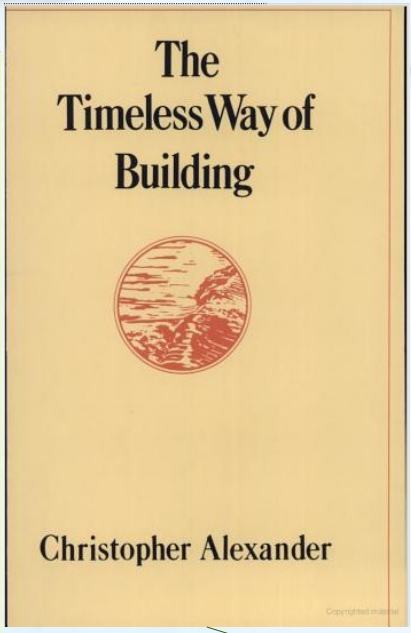
An evolution of pattern languages across domains

2005 http://books.google.com/books?id=6K5QAAAAMAAJ ; http://orgpatterns.wikispaces.com/

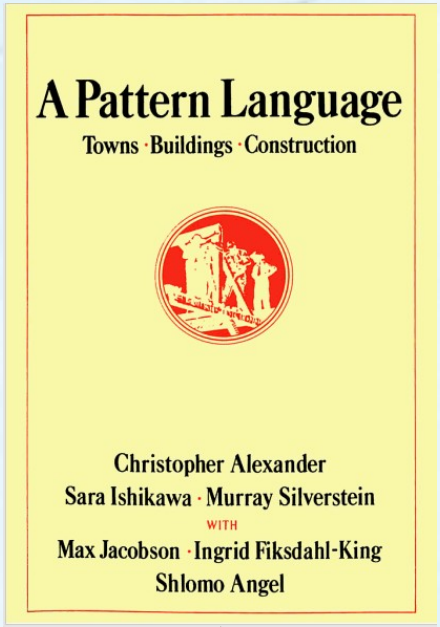




1994 http://books.google.com/books?id=6oHuKQe3TjQC



1979 http://books.google.com/books?id=H6CE9hlbO8sC



1977 http://books.google.com/books?id=hwAHmktpk5IC ; http://www.patternlanguage.com/

127 INTIMACY GRADIENT**

. . . if you know roughly where you intend to place the building wings -- WINGS OF LIGHT (107), and how many stories they will have -- NUMBER OF STORIES (96), and where the MAIN ENTRANCE (110) is, it is time to work out the rough disposition of the major areas on every floor. In every building the relationship between the public areas and private areas is most important.

* * *

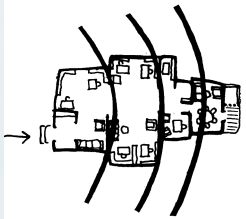
Unless the spaces in a building are arranged in a sequence which corresponds to their degrees of privateness, the visits made by strangers, friends, guests, clients, family, will always be a little awkward.

In any building -- house, office, public building, summer cottage - people need a gradient of settings, which have different degrees of intimacy. A bedroom or boudoir is most intimate; a back sitting room. or study less so; a common area or kitchen more public still; a front porch or entrance room most public of all. When there is a gradient of this kind, people can give each encounter different shades of meaning, by choosing its position on the gradient very carefully. In a building which has its rooms so interlaced that there is no clearly defined gradient of intimacy, it is not possible to choose the spot for any particular encounter so carefully; and it is therefore impossible to give the encounter this dimension of added meaning by the choice of space. This homogeneity of space, where every room has a similar degree of intimacy, rubs out all possible subtlety of social interaction in the building.

We illustrate this general fact by giving an example from Peru - a case which we have studied in detail. [….]

The intimacy gradient is unusually crucial in a Peruvian house. But in some form the pattern seems to exist in almost all cultures. We see it in widely different cultures -- compare the plan of an African compound, a traditional Japanese house, and early American colonial homes -- and it also applies to almost every building type -- compare a house, a small shop, a large office building, and even a church. It is almost an archetypal ordering principle for all man's buildings. All buildings, and all parts of buildings which house well defined human groups, need a definite gradient from "front" to "back," from the most formal spaces at the front to the most intimate spaces at the back.

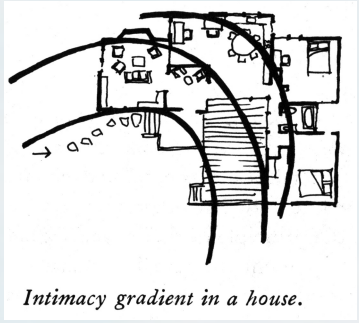
In an office the sequence might be: entry lobby, coffee and reception areas, offices and workspaces, private lounge.



Office intimacy gradient.

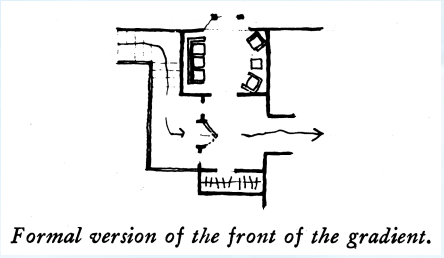
In a small shop the sequence might be: shop entrance, customer milling space, browsing area, sales counter, behind the counter, private place for workers.

In a house: gate, outdoor porch, entrance, sitting wall, common space and kitchen, private garden, bed alcoves.



Intimacy gradient in a house.

And in a more formal house, the sequence might begin with something like the Peruvian sala -- a parlor or sitting room for guests.



Formal version of the front of the gradient.

Source: Christopher Alexander et. al. 1997, *A Pattern Language: Towns, Building, Construction*, Oxford Press.-

127 INTIMACY GRADIENT**

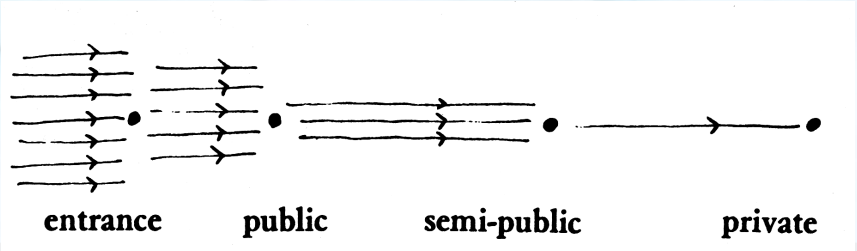
. . . if you know roughly where you intend to place the building wings -- WINGS OF LIGHT (107), and how many stories they will have -- NUMBER OF STORIES (96), and where the MAIN ENTRANCE (110) is, it is time to work out the rough disposition of the major areas on every floor. In every building the relationship between the public areas and private areas is most important.

* * *

Unless the spaces in a building are arranged in a sequence which corresponds to their degrees of privateness, the visits made by strangers, friends, guests, clients, family, will always be a little awkward.

Therefore:

Lay out the spaces of a building so that they create a sequence which begins with the entrance and the most public parts of the building, then leads into the slightly more private areas, and finally to the most private domains.

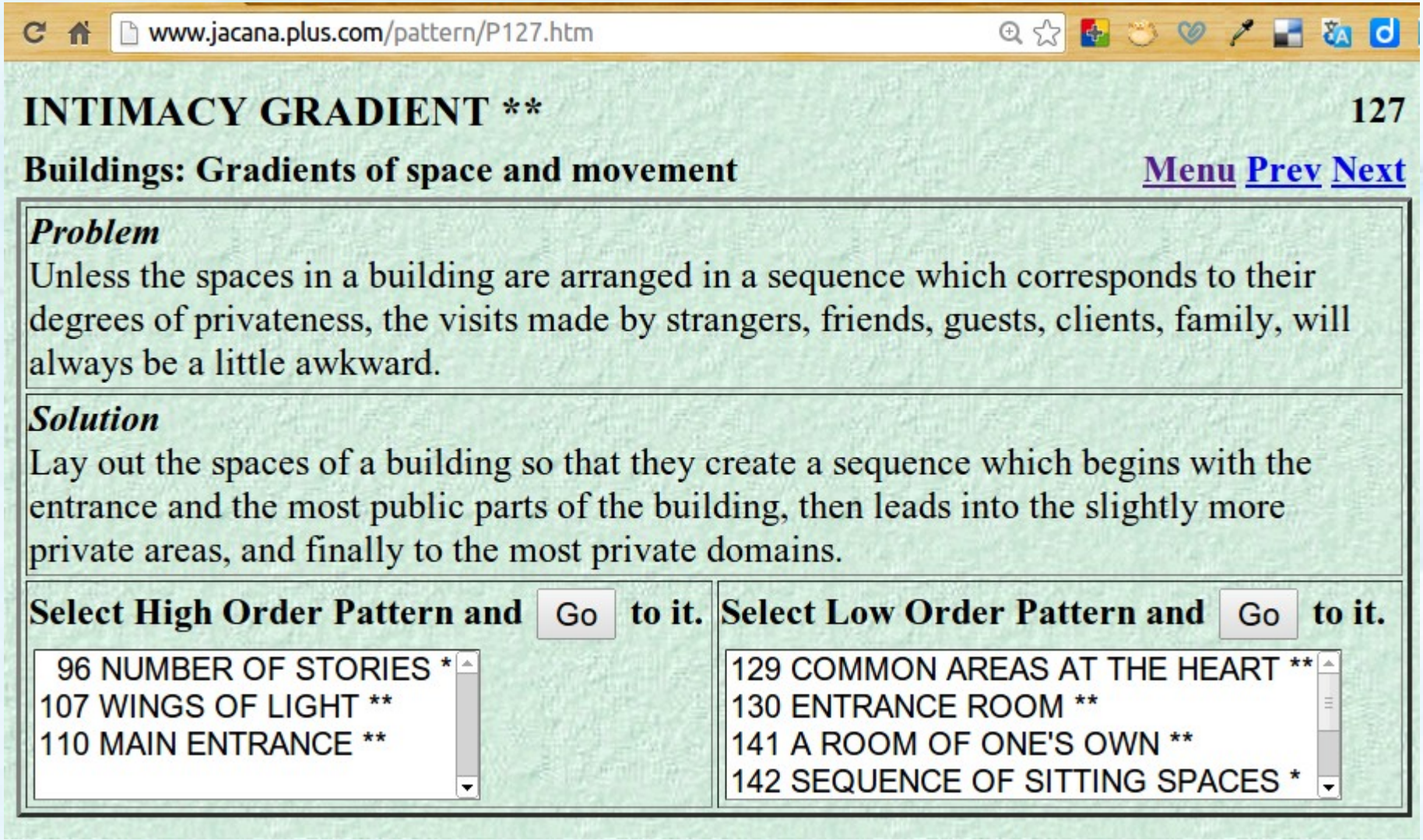


* * *

At the same time that common areas are to the front, make sure that they are also at the heart and soul of the activity, and that all paths between more private rooms pass tangent to the common ones -- COMMON AREAS AT THE HEART (129). In private houses make the ENTRANCE ROOM (130) the most formal and public place and arrange the most private areas so that each person has a room of his own, where he can retire to be alone A ROOM OF ONE'S OWN (141). Place bathing rooms and toilets half-way between the common areas and the private ones, so that people can reach them comfortably from both BATHING ROOM (144); and place sitting areas at all the different degrees of intimacy, and shape them according to their position in the gradient - SEQUENCE OF SITTING SPACES (142). In offices put RECEPTION WELCOMES YOU (149) at the front of the gradient and HALF-PRIVATE OFFICE (152) at the back. . . .

Source: Christopher Alexander et. al. 1997, *A Pattern Language: Towns, Building, Construction*, Oxford Press.-

127 INTIMACY GRADIENT**



Source: Christopher Alexander et. al. 1997, *A Pattern Language: Towns, Building, Construction*, Oxford Press.-

August 26, 2004

Intimacy Gradient and Other Lessons from Architecture

A number of my posts have been about integrating different domai
understand how human behavior should be incorporated in the des
Dunbar Number in sociology, and both Four Kinds of Privacy and F
work in the cryptography field. The topic of this post comes from th

In order to provide for Progressive Trust, you need to establish wh
Gradient".

The concept of Intimacy Gradient comes from architect Christophe
Language: Towns, Buildings, Construction. (Oxford University Pres

> *Pattern #127 - Intimacy Gradient:*
>
> ***Conflict:*** *Unless the spaces in a building are arranged in
> corresponds to their degrees of privateness, the visits ma
> guests, clients, family, will always be a little awkward.*
>
> ***Resolution:*** *Lay out the spaces of a building so that they
> begins with the entrance and the most public parts of the
> slightly more private areas, and finally to the most private*

In architecture there are always some areas of the house or buildir
entry, the living room, the atrium, etc., and areas that are more pri
bedrooms, and offices. In a good design there is some marker of c
areas -- it might be a difference in ceiling height, a stairway leading
entrance. As an example, in the classical Japanese tea house, you

Failure to respect the Intimacy Gradient results in uncomfortable b
about a Frank Gehry building at Case Western Reserve University:

> *I asked many of the graduate students how they felt abou
> "Horrible," said one. "Like living in a refrigerator" said ano
> comfortable offices and gathering places, and had the mo
> Now everything is so sterile, and the acoustics so bad, tha
> together. I have to go outside if I want any privacy."*

The Intimacy Gradient is also used in other media. As I noted in my
Hand Circus:

> *When we arrived, we were led down the side of the theatr
> noticed that it looked like we were all being led backstage
> a sudden see an entrance -- maybe 5 foot tall requiring m
> through and to our surprise, we are have walked through*

The Intimacy Gradient is also used in other media. As I noted in my review of Seven Fingers of the Hand Circus:

> *When we arrived, we were led down the side of the theatre and all of a sudden I noticed that it looked like we were all being led backstage. We curve around and all of a sudden see an entrance -- maybe 5 foot tall requiring most of us to duck. We duck through and to our surprise, we are have walked through a fridgerator, and we are on the stage!*
>
> *One of the 7 players welcomes us, and another offers random people a glass of tea as we walk across the stage to our seats. The stage is set like a city loft, with a tv, some couches, a bed, a bathtub and shower, a kitchen, and of course the fridgerator we entered through. On the stage, and chatting to members of the audience are the 7 cast members, all wearing comfortable looking white shorts or athletic and white t-shirts.*
>
> *The audience arrives over 30 minutes and the 7 players act as if we are guests of their loft, serving some of us tea, chatting, sweeping the floor, etc.*

Entering through the refrigerator door raised the intimacy of the experience for the audience of that circus. Thus in spite of it being produced in a large auditorium it felt as up-close and personal as did the much smaller Circus Contraption.

The Intimacy Gradient exists in movies as well -- anywhere you see a scene taking place in a public space that transitions down through smaller and tighter shots ultimately to a closeup of a face it is much more intimate then just cutting to the closeup.

In social software design, there also needs to be an Intimacy Gradient. One of the problems with Wikis is that there is often very little transition between public and intimate, and doing so can be quite jarring. SocialText, a Wiki service vendor, is aware of this problem and is "seeking to add more layers to the 'intimacy gradient', without recreating the highly structured collaboration tools that exist today". Ross Mayfield outlines this possible future Intimacy Gradient for SocialText:

- *The broadest tier is a guest space, available to all*
- *The second tier is a knowledgebase, accessible to all employees and contractors*
- *The third tier is product development, for employees and contractors bound by a confidentiality agreement*
- *The fourth tier is for the core management team to share confidential financial and HR information.*

The Hillside Group – Design Patterns

hillside.net/patterns

THE HILLSIDE GROUP

Site Search

Search...

Hilltop | Books | Contact | Conferences | Patterns | Vision | Wiki

MAIN MENU

- Hilltop
- Books
- Contact
- Conferences
- Patterns
  - Books
  - Patterns Catalog
  - About Patterns
  - TPLoP
  - Education
  - Mailing Lists
  - Writing
  - Tools
  - Links

You are here: Home > Patterns

DESIGN PATTERNS LIBRARY

Welcome to the patterns home page. It is a source for information about all aspects of software patterns and pattern languages. If you are new to patterns, James Coplien and Richard Gabriel have created a succinct pattern definition.

Patterns and Pattern Languages are ways to describe best practices, good designs, and capture experience in a way that it is possible for others to reuse this experience. The Hillside Group takes pleasure in sponsoring many different PLoP conferences that are provided for the betterment of the pattern community.

Fundamental to any science or engineering discipline is a common vocabulary for expressing its concepts, and a language for relating them together. The goal of patterns within the software community is to create a body of literature to help software developers resolve recurring problems encountered throughout all of software development. Patterns help create a shared language for communicating insight and experience about these problems and their solutions. Formally codifying these solutions and their relationships lets us successfully capture the body of knowledge which defines our understanding of good architectures that meet the needs of their users. Forming a common pattern language for conveying the structures and mechanisms of our architectures allows us to intelligibly reason about them. The primary focus is not so much on technology as it is on creating a culture to document and support sound engineering architecture and design.

The Hillside Group – Software (Design) Pattern (Definition)

hillside.net/patterns/50-patterns-library/patterns/222-design-pattern-definition

- Hilltop
- Books
- Contact
- Conferences
- Patterns
  - Books
  - Patterns Catalog
  - About Patterns
  - TPLoP
  - Education
  - Mailing Lists
  - Writing
  - Tools
  - Links
- Vision
- Wiki

tune in to se-radio.net

DESIGN PATTERNS LIBRARY

Software Patterns

by James O. Coplien
Bell Laboratories
Naperville, Illinois

Patterns are a recent software engineering problem-solving discipline that emerged from the object-oriented community. Patterns have roots in many disciplines, including literate programming, and most notably in Alexander's work on urban planning and building architecture. (Alexander, 1977).

The goal of the pattern community is to build a body of literature to support design and development in general. There is less focus on technology than on a culture to document and support sound design. Software patterns first became popular with the object-oriented Design Patterns book (Gamma et. al., 1995). But patterns have been used for domains as diverse as development organization and process, exposition and teaching, and software architecture. At this writing, the software community is using patterns largely for software architecture and design.

Here is a pattern used in Lucent telecommunication products such as the Switching System® (extracted informally from *Adams, 1996*).:

Description

Name: *Try All Hardware Combos*

Problem: *The control complex of a fault-tolerant system can arrange its subsystems in many different configurations. There are many possible paths through the subsystems. How do you select a workable configuration when there is a faulty subsystem?*

Context: *The processing complex has several duplicated subsystems including a CPU, static and dynamic memory, and several busses. Standby units increase system reliability. 16 possible*

Example pattern – Lucent Telecommunications product

Description

Name: Try All Hardware Combos

Problem: The control complex of a fault-tolerant system can arrange its subsystems in many different configurations. There are many possible paths through the subsystems. How do you select a workable configuration when there is a faulty subsystem?

Context: The processing complex has several duplicated subsystems including a CPU, static and dynamic memory, and several busses. Standby units increase system reliability. 16 possible configurations (64 in the 4 ESS) of these subsystems give fully duplicated sparing in the 5ESS. Each such configuration is called a configuration state.

Forces: You want to catch and remedy single, isolated errors. You also want to catch errors that aren't easily detected in isolation but result from interaction between modules. You sometimes must catch multiple concurrent errors. The CPU can't sequence subsystems through configurations since it may itself be faulty. The machine should recover by itself without human intervention, many high-availability system failures come from operator errors, not primary system errors. We want to reserve human expertise for problems requiring only the deepest insights.

Solution: Maintain a 16-state counter in hardware called the configuration counter. There is a table that maps that counter onto a configuration state. The 5ESS switch tries all side 0 units (a complete failure group), then all side 1 units (the other failure group), seeking an isolated failure. When a reboot fails, the state increments and the system tries to reboot again. The subsequent counting states look for multiple concurrent failures caused by interactions between system modules.

Resulting Context: Sometimes the fault isn't detected during the reboot because latent diagnostic tasks elicit the errors. The pattern Fool Me Once solves this problem. And sometimes going through all the counter states isn't enough; see the patterns Don't Trust Anyone and Analog Timer.

Rationale: The design is based on hardware module design failure rates (in Failures in a trillion (FITs)) of the hardware modules. The pattern recalls the extreme caution of first-generation developers of stored program control switching systems.

This is a good pattern because:

- *It solves a problem*: Patterns capture solutions, not just abstract principles or strategies.
- *It is a proven concept*: Patterns capture solutions with a track record, not theories or speculation.
- *The solution isn't obvious*: Many problem-solving techniques (such as software design paradigms or methods) try to derive solutions from first principles. The best patterns generate a solution to a problem indirectly--a necessary approach for the most difficult problems of design.
- *It describes a relationship*: Patterns don't just describe modules, but describe deeper system structures and mechanisms.
- *The pattern has a significant human component (minimize human intervention).* All software serves human comfort or quality of life; the best patterns explicitly appeal to aesthetics and utility.

A *pattern language* defines a collection of patterns and the rules to combine them into an architectural style. Pattern languages describe software frameworks or families of related systems.

Source: http://hillside.net/patterns/50-patterns-library/patterns/222-design-pattern-definition

Patterns and Pattern Languages are ways to describe best practices, good designs, and capture experience in a way that it is possible for others to reuse this experience[1]

Pattern Name:

(Use italics for pattern names per Meszaros).

Aliases:

(Aliases, or none)

Problem

Give a statement of the problem that this pattern resolves. The problem may be stated as a question.

Context

Describe the context of the problem.

Forces

Describe the forces influencing the problem and solution. This can be represented as a list for clarity.

- Force one
- Force two

Solution

Give a statement of the solution to the problem.

Resulting Context

Describe the context of the solution.

Rationale

Explain the rationale behind the solution.

Known Uses

List or describe places where the pattern is used.

Related Patterns

List or describe any related patterns.

Source: [1] “Patterns”, The Hillside Group, http://hillside.net/patterns ; [2] “Writing Patterns”, AG's HTML template at http://hillside.net/index.php/ag-template ; “Canonical Form” (for writing patterns) at http://c2.com/cgi/wiki?CanonicalForm

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

www.martinfowler.com/articles/writingPatterns.html

Writing Software Patterns

I've spent a lot of my writing energy writing patterns. From time to tim
questions about why I do that and what makes a good pattern. This is
how I look at patterns with my suggestions for people who are interes
patterns themselves.

01 August 2006

Martin Fowler

Find similar writing

Contents

Christopher Alexander

- (1) Picture with archetypal example
- (2) Paragraph sets context with how it helps to complete larger patterns
- (3) Three diamonds (start of problem)
- (4) Headline essence of problem (bold type)
- (5) Body of problem, empirical background
- (6) Solution instructions (bold type) describing field of physical and social relations
- (7) Diagram
- (8) Three diamonds (main body finished)
- (9) Paragraph that ties pattern to smaller patterns

Gang-of-Four

(Gamma, Helm, Johnson, Vlissides 1994, Design Patterns)

- Intent
- Motivation
- Applicability
- Structure
- Participants
- Collaborations
- Consequences
- Implementation
- Sample Code
- Known Uses
- Related Patterns

Portland

(C2 wiki, short)

- Problem
- … therefore ...
- Solution

Pattern-Oriented Software Architecture

- Summary
- Example
- Context
- Problem
- Solution
- Structure
- Dynamics
- Implementation
- Example resolved
- Variants
- Known uses
- Consequences
- See also

Patterns of Enterprise Application Architecture

- How it works
- When to use it
- Examples

Pattern Language and Systems Thinking?

A pattern doesn't exist apart from a pattern language; its first purpose is to establish connections to other patterns in the language ([Alexander1977], p. xii). But to understand pattern languages, you must first understand what a pattern is. We know this is recursive, and to understand recursion, you must first understand recursion. We must start somewhere, and we start here: with patterns.

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.

Some of these aspects of pattern don't come out in the popular literature, and you may not find them all in the same place in Alexander's definitions. But they are the key elements of what makes a pattern a pattern, and what makes it different from a simple rule. **A pattern is a rule**: the word *configuration* should be read as "a rule to configure." But it is more than just a rule; it is a special kind of rule **that contributes to the overall structure of a system, that works together with other patterns to create emergent structure and behavior.** [p. 14]

Alexander believes that order in any system fundamentally depends on the process used to build the system. This is why the fundamental process is important (see the section PIECEMEAL GROWTH (6.2)). **It is important that each step preserves structure and gradually adds local symmetries, and the organization unfolds over time. It is step-by-step adaptation with feedback. Simply following the pattern language doesn't give you a clue about how to handle the feedback. So that's why the fundamental process exists: to give complete freedom to the design process to attack the weakest part of the system, wherever it may be.**

However, the fundamental process cannot work on a human scale without some kind of cognitive guide that is built on experience and which can foresee some of the centers that must be built. That's what patterns are: essential centers.

If unfolding is important, how do you know what order to unfold things? The sequence is crucial. You want a smooth, structure-preserving unfolding. It shouldn't feel like "organizational design."

So, what a sequence does is:

- Preserves structure;
- Keeps you doing one thing at a time;
- Takes the whole organization into account at each step;
- May be repeated tens of thousands of times.

Sequences take you into unpredictability, and into circumstances you handle with feedback, always in the context of the whole organization. Sequences are where generativity comes from. [p. 37]

Source: Jim Coplien and Neil Harrison 2004. *Organizational Patterns of Agile Software Development*

JRP
Journal of Research Practice
Innovations and Challenges in Multiple Domains

Journal of Research Practice
Volume 2, Issue 1, Article R1, 2006

JRP
ISSN: 1712-851X

Review:

The Art of Observation: Understanding Pattern Languages

Werner Ulrich
University of Fribourg, SWITZERLAND;
The Open University, UNITED KINGDOM
wulrich@gmx.ch

The Timeless Way of Building. Book by Christopher Alexander. Published by Oxford University Press, New York, 1979, xv+552 pp., ISBN 0-19-502402-8, USD 34.65.

Suggested Citation: Ulrich, W. (2006). The art of observation: Understanding pattern languages--A review of Christopher Alexander's *The Timeless Way of Building* (New York, Oxford University Press, 1979). *Journal of Research Practice, 2*(1), Article R1. Retrieved [date of access], from http://jrp.icaap.org/index.php/jrp/article/view/26/46

Christopher Alexander's book, *The Timeless Way of Building,* is probably the most beautiful book on the notion of quality in observation and design that I have been reading since Robert Pirsig's (1974) *Zen and the Art of Motorcycle Maintenance*. It was published in 1979, when Alexander was a professor of architecture at the University of California, Berkeley, where I was at that time studying. Although I was aware of some of Alexander's famous articles such as "A city is not a tree" (Alexander, 1965), the book (Alexander, 1979) never quite made it to the top of my reading list. This remained so until recently, when I met a software developer who enthusiastically talked to me on a book he was currently reading, about the importance of understanding design patterns. He was talking about the very book I had failed to read during my Berkeley years and which, as I now discovered, has since become a cult book among computer programmers and information scientists, as well as in other fields of research. I decided it was time

Email the author (Login required)
Post a Comment (Login required)

1. The Quality without a Name

… the essence of the Quality without a Name consists in the idea of design patterns that are alive and which, if identified in sufficient number, can be used to make up a whole pattern language for quality design.

2. Patterns that are Alive

As a rule, a room that does not have a window place lacks quality; its windows are just holes in the wall.

3. The Idea of a "Pattern Language"

… patterns are not arbitrary design ideas but can and need to be identified and verified through careful observation. Furthermore, patterns become meaningful only within a hierarchy of interdependent patterns, in which each pattern helps to complete larger (more generic) patterns within which it is contained, and in turn is further completed by smaller (more specific) patterns that it contains.

4. Against Modular Architecture

The way a pattern language works is not through a process of addition or combination of preformed parts of a design, but through a sequential process of unfolding, in which each pattern is developed in the context of the whole that is given by previously unfolded patterns ...

Design thus resembles more the evolution of an embryo than the drawing of an architectural plan. It is a process of growth--of increasing differentiation--with the pattern language operating as its genetic code. No application of a pattern will ever generate exactly the same result, for the result depends on the context generated by the previous stages of growth. This is different from conventional architectural design, in which the details of a building are made from identical, modular parts (e.g., prefabricated windows).

The Quality Without a Name

Alexander's search, culminating in pattern languages, was to find **an objective (rather than a subjective) meaning for beauty, for the aliveness that certain buildings, places, and human activities have**. The objective meaning is the quality without a name, and I believe we cannot come to grips with Alexander in the software community unless we come to grips with this concept. [….]

The quality is an objective quality that things like buildings and places can possess that makes them good places or beautiful places. Buildings and towns with this quality are habitable and alive. The key point to this — and the point that really sets Alexander apart from his contemporaries and stirs philosophical debate—is that the quality is objective.

It started in 1964 when he was doing a study for the Bay Area Rapid Transit (BART) system …. One of the key ideas in this book was that in a good design there must be an underlying correspondence between the structure of the problem and the structure of the solution — good design proceeds by writing down the requirements, analyzing their interactions on the basis of potential misfits, producing a hierarchical decomposition of the parts, and piecing together a structure whose structural hierarchy is the exact counterpart of the functional hierarchy established during the analysis of the program. (Alexander 1964)

Alexander was studying the system of forces surrounding a ticket booth, and he and his group had written down 390 requirements for what ought to be happening near it. Some of them pertained to such things as being there to get tickets, being able to get change, being able to move past people waiting in line to get tickets, and not having to wait too long for tickets. What he noticed, though, was that **certain parts of the system were not subject to these requirements and that the system itself could become bogged down because these other forces — forces not subject to control by requirements—acted to come to their own balance within the system**. For example, if one person stopped and another also stopped to talk with the first, congestion could build up that would defeat the mechanisms designed to keep traffic flow smooth. Of course there was a requirement that there not be congestion, but there was nothing the designers could do to prevent this by means of a designed mechanism.

Alexander proposes some words to describe the quality without a name, but even though he feels they point the reader in a direction that helps comprehension, **these words ultimately confuse. The words are *alive*, *whole*, *comfortable*, *free*, *exact*, *egoless*, and *eternal***. I'll go through all of them to try to explain the quality without a name.

Source: Richard P. Gabriel 1996, *Patterns of Software, http://dreamsongs.net/*

SEBoK

Guide to the Systems Engineering Body of Knowledge

Patterns of Systems Thinking

Patterns of Systems Thinking

This topic forms part of the Systems Thinking knowledge area (KA). It identifies systems **patterns** as part of the basic ideas of **systems thinking**. The general idea of patterns and a number of examples are described. A brief conclusion discusses the maturity of **systems science** from the perspective of **principles** and patterns.

Contents [hide]

Systems Patterns

This section first discusses definitions, types, and pervasiveness of patterns. Next, samples of basic patterns in the form of hierarchy and **network** patterns, metapatterns, and **systems engineering** (SE) patterns are discussed. Then samples of patterns of failure (or "**antipatterns**") are presented in the form of system archetypes, along with antipatterns in **software** engineering and other fields. Finally, a brief discussion of patterns as maturity indicators is given.

Pattern Definitions and Types

The most general definition of pattern is that it is an expression of an observed **regularity**. Patterns exist in both natural and artificial systems and are used in both **systems science** and **systems engineering** (SE). Theories in science are patterns. Building **architecture** styles are patterns. **Engineering** uses patterns extensively.

Patterns are a representation of similarities in a set or class of **problems**, **solutions**, or systems. In addition, some patterns can also represent uniqueness or differences, e.g., uniqueness pattern or unique identifier, such as automobile vehicle identification number (VIN), serial number on a consumer product, human fingerprints, DNA. The pattern is that a unique identifier, common to all instances in a class (such as fingerprint), distinguishes between all instances in that class.

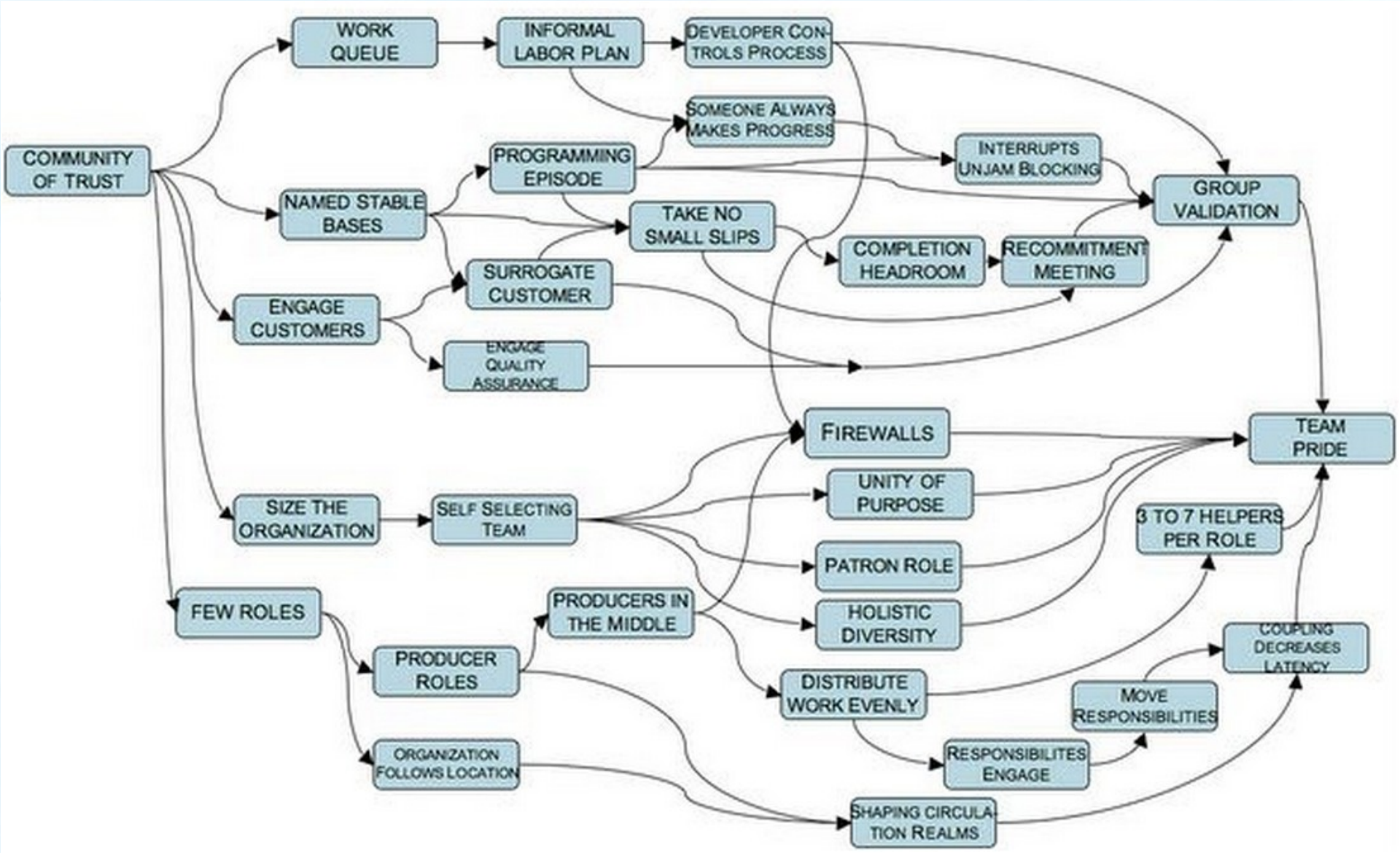
The term pattern has been used primarily in building architecture and urban planning by Alexander (Alexander et al. 1977, Alexander 1979) and in

Quicklinks
- Main Page
- Editors' Note
- Meet the Editors
- Acknowledgements
- How to Read the SEBoK
- Download SEBoK PDF
- Cite the SEBoK
- Release History
- Copyright Information
- About the SEBoK
- Go to Sandbox

Outline
- Table of Contents
- Part 1: SEBoK Introduction
- Part 2: Systems
  - Systems Fundamentals
  - Systems Science
  - Systems Thinking
    - What is Systems Thinking?
    - Concepts of Systems Thinking
    - Principles of Systems Thinking
    - **Patterns of Systems Thinking**
  - Representing Systems with Models
  - Systems Approach Applied to Engineered S
- Part 3: SE and Management
- Part 4: Applications of Systems Engineering
- Part 5: Enabling Systems Engineering
- Part 6: Related Disciplines
- Part 7: SE Implementation Examples

Navigation
- Knowledge Areas
- Topics
- Use Cases
- Case Studies
- Vignettes
- Glossary of Terms

Scrum Patterns Summary



Source: https://sites.google.com/a/scrumorgpatterns.com/www/scrumpatternssummary

www.javiergarzas.com/en/2013/07/24/interview-to-jim-coplien-1/

Jul 24, 2013 *Posted by Javier Garzás in General | 1 comment*

Interview with Jim Coplien (1/2)

Jim Coplien (also known as Cope) is one of the key founders of software patterns movement and the agile development, a co-author of "Organizational Patterns of Agile Software Development" and "Lean Software Architecture for Agile Software Development" among others key books.

Now he works with the fathers of Scrum to facilitate its evolution as formalization.

Personally, it is a great honor for me to interview Jim, because part of my dissertation was about design patterns. And Jim is one of the fathers of the movement.

Jim, thank you very much for the interview.

1 – Most knows your work with patterns, but... Who is Jim? Where are you from? Background? Where

I'm Product Owner for the Scrum Patterns effort, Scrum PLoP® (http://www.scrumplop.org). It is, in some ways, an outgrowth of the organizational patterns work that started 20 years ago (http://orgpatterns.wikispaces.com). I'm proud of this work because it is the only body I know of that is chartered as a non-partisan group to evolve a rationalized definition of Scrum. [….]

There are three factors that make the Scrum Patterns special.

1. They adopt a systems thinking view of organizational transformation, rather than a rulebook approach. This means that we can get beyond technique to organizational structure and to principles and values, and really address the human issues that make complex development so hard. Patterns help us think in systems ways that are more or less the opposite of "root cause analysis."

2. The Scrum Patterns are shaped by the thinking at the foundations of Scrum and written first-hand by those great thinkers: Jeff Sutherland, Michael Beedle, Gabrielle Benefield, Jens Østergaard, and more.

3. They reflect input from all the major certifying entities, and where we lack engagement with key constituencies today we are always seeking to be inclusive with more folks.

I think it's important to understand that **what we're building isn't just a pattern catalog. You shop by paging through a catalog and choose one or two things to take home. We are building much more formal constructs called pattern languages. Pattern languages include sets of rules that constrain meaningful combinations of patterns according to a generative grammar, that can be used by the designer to generate a myriad of wholes.** What this means in layman's terms is that we are building a roadmap that can inspire organizations by showing them the many paths to building great Scrum teams. A pattern language requires judgment, insight, and adaptation on the part of its users. Very few of the publications currently called "patterns" have this generative ability. However, the inventor of patterns, Christopher Alexander, insists that this is an essential property of patterns. They compose with each other to create "morphological wholes." These Wholes are teams, value streams, relationships, cycles in time, and other structures in the development organization. [….]

Most engineering students think in terms of short time frames; a good mature engineer thinks ahead to how use and nature will cause a structure to weaken or become obsolete. Patterns attack that kind of entropy. Engineers building Japanese temples today plant trees that will be used to build their successors 200 years from now; patterns in construction lay a foundation for a good future by understanding the past.

Scrum Patterns Summary

The patterns without which Scrum is unlikely to work

These patterns are Scrum. In June 2008, founders of Scrum and Organizational Patterns met together with a small number of other experts from both camps to map out patterns from the book *Organizational Patterns of Agile Software Development* as they apply to projects under the Scrum framework. These patterns represent elements of the framework, as well as fundamental practices which dissect the framework into its fundamental underlying components. Because patterns work at the level of structure rather than cause-and-effect, we call these patterns Second-Level Scrum patterns, which can be compared to the First- and Third-level Scrum patterns described below.

These patterns are independent of software development per se. The group also agreed to patterns without which Scrum is unlikely to succeed, and about which Jeff Sutherland said he has incorporated into every Scrum he has started. These can be found on the page Software Scrum Patterns.

The third and final part of the picture is the pattern language by Beedle et al. that describes the cause-and-effect components of Scrum. You can see a summary of those on the page First-Level Scrum Patterns.

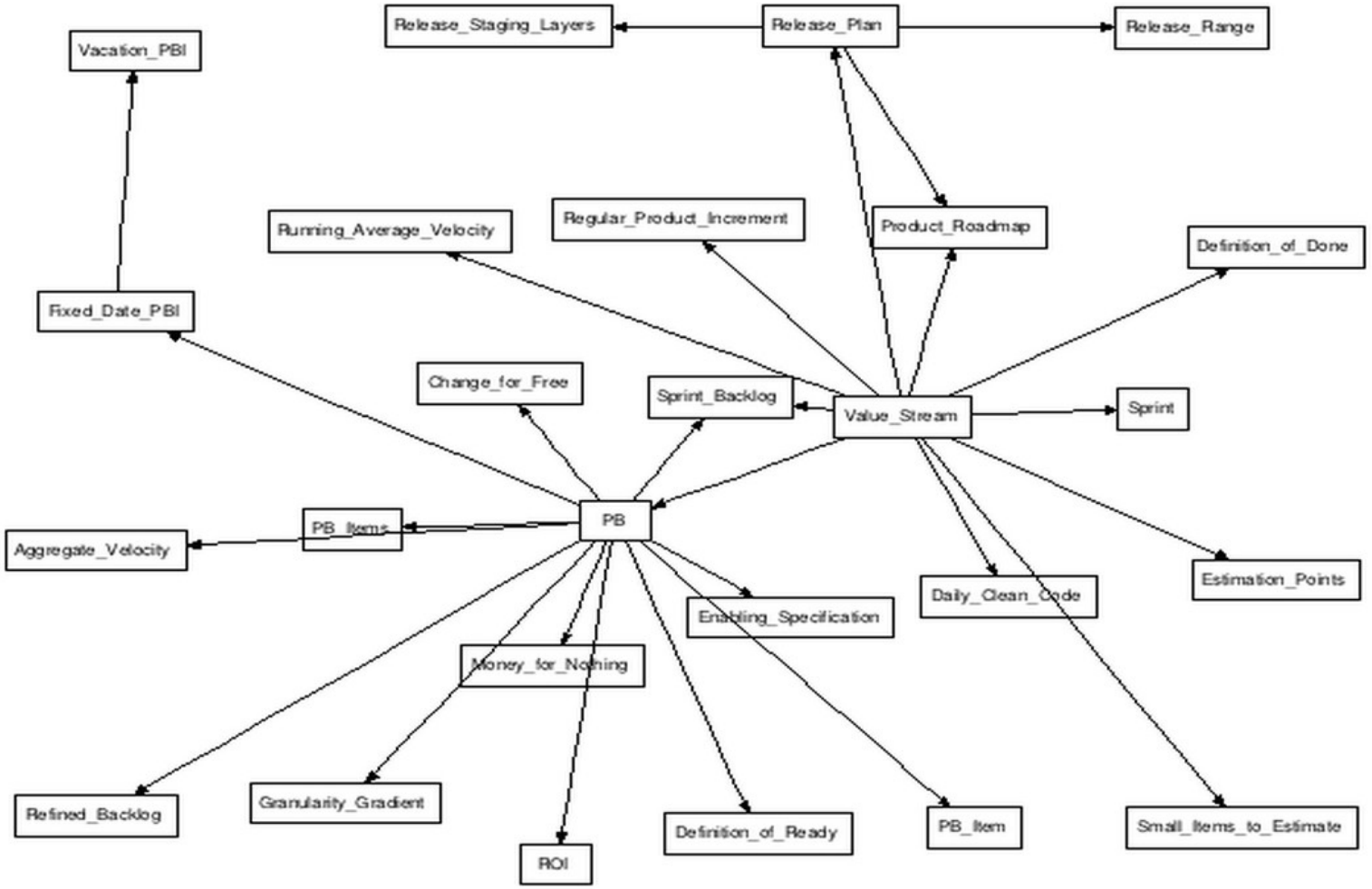
See the pattern language picture at the bottom of this page.

Pattern Name	Description	Scrum Notes
COMMUNITY OF TRUST	People are open enough to surface uncomfortable truths	Scrum is based on honesty, openness, and visibility
WORK QUEUE		You have a product backlog and a sprint backlog
NAMED STABLE BASES	Don't use continuous integration, because that makes it difficult for the team to have a shared vision of where the base stands. Integrate public integrations regularly	While continuous integration is the norm, you also have a delineated, agreed target where everything must converge in a coordinated way at the end of a sprint. Also, some architectural changes cannot be made piecemeal but are best handled by offline work on a branch administered by the source management system.
ENGAGE CUSTOMERS	Customers should be tightly coupled to the organization	Scrum has no customer role! However, no Scrum would be complete without one. Note that the Product Owner is not the customer: the Product Owner stands to optimize ROI for the enterprise, and though this role takes input from the customer, it is not the locus of caring for customer interests! However, also see SURROGATE CUSTOMER below.
SIZE THE ORGANIZATION	Start small and grow gradually	Small teams work best. Scrum talks about teams of seven plus or minus two people; over time, the smaller teams have been winning out over the larger ones.
FEW ROLES	The total number of roles is small	Of course, original Scrum has only three roles. However, because DOMAIN EXPERTISEIN ROLES is also important we admit a small number of specialized roles. If there are too many roles (areas of specialization) the specialization works against the process of finding out what you need to
FEW ROLES	The total number of roles is sm	
INFORMAL LABOR PLAN	If developers need to do the mo important thing now, then let developers negotiate among themselves or "just figure out th right thing to do" as regards sho term plans, instead of master planning.	
PROGRAMMING EPISODE	You have fixed-length sprints	
STAND-UP MEETING	Get the team together for a regular short meeting to exchange status information.	
SURROGATE CUSTOMER	Sometimes, if it is inconvenient impossible to talk to a real customer in real time, you need local stand-in.	
ENGAGE QUALITY ASSURANCE	Quality Assurance is not an afterthought	
SELF-SELECTING TEAM	Appointed teams don't work. Th people should decide what peo should be on which teams.	
PRODUCER ROLES	If your organization has too ma roles, but does not know which eliminate, then identify roles as Producers, Supporters, or Deadbeats; eliminate the Deadbeats and combine some the Supporters	
ORGANIZATION FOLLOWS LOCATION	Units of a feather flock together the organization is colocated	
DEVELOPER CONTROLS PROCESS	The developer calls the shots	

https://sites.google.com/a/scrumplop.org/published-patterns/home/pattern-map

115
days until
ScrumPLoP!

Pattern Map



groupworksdeck.org/what-we-mean-by-pattern

group works

A Pattern Language for Bringing Life to Meetings and Other Gatherings

Username Password Log in

Home | The Patterns | Deck | Our Community | News | Contact

- Buy printed deck from 100fires.com
- Download free deck
- iPhone version

Donate

Latest News

Successful Portland Workshop!

Last weekend we hosted another set of very successful workshop offerings around the Group Works...

Productive Eugene Work Session

Can you believe that after this packed weekend the four of us-- Daniel Lindenberger, Dave...

Portland Workshops - November 15-17, 2013

Hereby announcing a Group Works team offering . . .

What we mean by "pattern"

For the purposes of this project, a pattern is a feature we believe shows up repeatedly in group processes that result in deepening, connection, and a fulfillment of purpose.

Scope: Our scope is the realm of deliberative/dialogic group processes aimed at goals such as decision-making, input, feedback, strategizing, visioning, and conflict resolution, that take place in the context of meetings, conferences, and other convenings that have these goals. Thus there are many times when people gather together that are beyond our zone of exploration (weddings, soccer matches, choir practice, etc.), although some of the patterns we come up with might also sometimes apply beyond our chosen domain.

While recognizing that there is a lot of overlap between patterns of face-to-face and online interaction, we are focused on face-to-face settings.

While we believe that many of these patterns show up cross-culturally, we do not make the claim that our pattern language is universal. This language is being written by a cross-section of North Americans from a variety of group backgrounds (political activism, communal living, and other alternative cultures; higher education; corporate experience in finance, software, and other fields; religious organizations; nonprofit management; indigenous tribes; public agency work; etc.). That diversity lends strength but falls far from addressing all places and groups. Any session takes place within a specific cultural context, and our assumption is that users of this language will take what inspires them and adapt appropriately.

Range: If you think about everything that goes into making a group conversation fulfilling, there is a vast range of things to pay attention to, from the most general to the most specific. In crafting this language, we ask people to focus on a particular middle section within this range, and hold your thinking at that level. We want to avoid being too general: For example, values such as democracy and cooperation, or principles such as Schwarz's "valid information" and "free and informed choice," are assumed or embedded rather than explicitly considered in the body of this work. We also want to avoid being too specific: There is a level of detail that is already well-represented in the existing literature and that we do not seek to replicate, namely:

- listing and explanations of methods (e.g. The Change Handbook);
- catalogues of tools and techniques.

Search

Mailing Lists

Join "Pattern-Hi" discussion list:
email address Subscribe
Pattern-Hi archives

Join "Pattern-Lo" announcement list:
email address Subscribe
Pattern-Lo archives

Source: http://groupworksdeck.org/what-we-mean-by-pattern

Group Works: A Pattern Language for Bringing Life to Meetings and Other Gatherings

1. Intention

Serving and attending to the larger purpose for the gathering and how it is manifested, including addressing its longer term meaning and consequence. Why are we here, what's our shared passion, and what are we aiming to accomplish.

Commitment	Priority Focus	Setting Intention
Invitation	**Purpose**	

2. Context

Understanding and working with the broader context and circumstances both in place and in culture.

Aesthetics of Space	Group Culture	Power of Place
Circle	**History and Context**	Whole System in the Room
Gaia	Nooks in Space and Time	

3. Relationship

Creating and maintaining quality connection with each other, honouring our full selves, and recognizing power relations. Includes being authentic and sometimes foregrounding emotional needs in the moment rather than task.

Appreciation	Honour Each Person	Shared Airtime
Breaking Bread Together	Hosting	**Tend Relationships**
Celebrate	Power Shift	Transparency
Good Faith Assumptions		

4. Flow

Covers rhythm, energy, and pacing. When we do what and for how long. Things to pay attention to both in anticipating the event and in responding to circumstances in the moment, to support movement along the intended trajectory toward the desired outcome.

Balance Process and Content	Follow the Energy	Right Size Bite
Balance Structure and Flexibility	Iteration	Ritual
Closing	Opening and Welcome	Seasoned Timing
Divergence and Convergence Rhythm	Preparedness	Subgroup and Whole Group
	Reflection/Action Cycle	Trajectory
	Rest	

5. Creativity

Using multiple intelligences and a variety of modes to open up creative possibilities.

Challenge	Improvise	Playfulness
Expressive Arts	Mode Choice	Power of Constraints
Generate Possibilities		

6. Perspective

Noticing and helping the group more openly and thoughtfully explore different ways of seeing an issue. Watching, understanding, and appreciating divergent viewpoints, ideas, values and opinions. The key is in how you look at something.

Common Ground	Go Meta	Translation
Embrace Dissonance and Difference	Seeing the Forest, Seeing the Trees	Unity and Diversity
Fractal	Time Shift	Value the Margins
		Viewpoint Shift

7. Modelling

The essential skills and responsibilities for both facilitator and participants, to demonstrate good group practice and ensure the process goes well. Includes monitoring, nurturing and mentoring the group, enabling their effective personal and collective self-management.

Appropriate Boundaries	**Holding Space**	Shared Leadership and Roles
Courageous Modelling	Listening	Simplify
Discharging	Mirroring	Taking Responsibility
Dwell with Emotions	Not About You	Witness with Compassion
Guerrilla Facilitation	Self-Awareness	

8. Inquiry & Synthesis

Discovering coherence and moving toward convergence. From gathering information to exploring knowledge to arriving at understanding, shared meaning, consensus, or clear outcomes.

Deliberate	Harvesting	Moving Toward Alignment
Distilling	**Inform the Group Mind**	Naming
Experts on Tap	Inquiry	Story
Feedback	Mapping and Measurement	Yes, and
Go Deeper		

9. Faith

Trusting and accepting what happens in a spirit of letting go and letting come. The mystery, synergy, and ineffable, complex magic of emergence. You can invite it, but you can't control it. Felt as a deep sense of connection not only to those assembled and to the work's purpose but to the larger universe as well.

All Grist for the Mill	Letting Go	Silence
Dive In	Magic	Spirit
Emergence	Presence	Trust the Wisdom of the Group

Source: http://groupworksdeck.org/patterns_by_category

Liberating Voices!
A Pattern Language for Communication Revolution

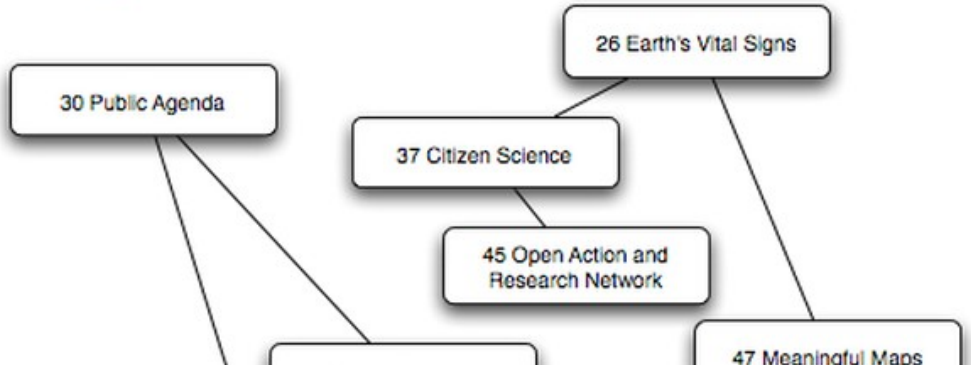
We are now in the 7th year of a 12-year project to help understand, motivate and inform the worldwide movement to establish full access to information and communication — including the design and management of information and communication systems.

We're working together to develop one or more "pattern languages" which can help people think about, design, develop, manage and use information and communication systems that more fully meet human needs now — and in the future.

Our "pattern language" is a holistic collection of "patterns" that can be used together to address an information or communication problem. Each "pattern" in this pattern language, when complete, will represent an important insight that will help contribute to a communication revolution.

The book that contains the first version of the Liberating Voices pattern language is now available from MIT Press. There are several "context" chapters in addition to the 136 patterns. The patterns are also available on this site, some in slightly different form.

Using a Network of Patterns



ACCESS TO PATTERNS

Contributors
Liberating Voices (book) Patterns
Pattern Pool (all submissions)
References in book patterns
I feel lucky! Look at a random pattern
Pattern Language Mailing List Join the Liberating Voices! discussion!

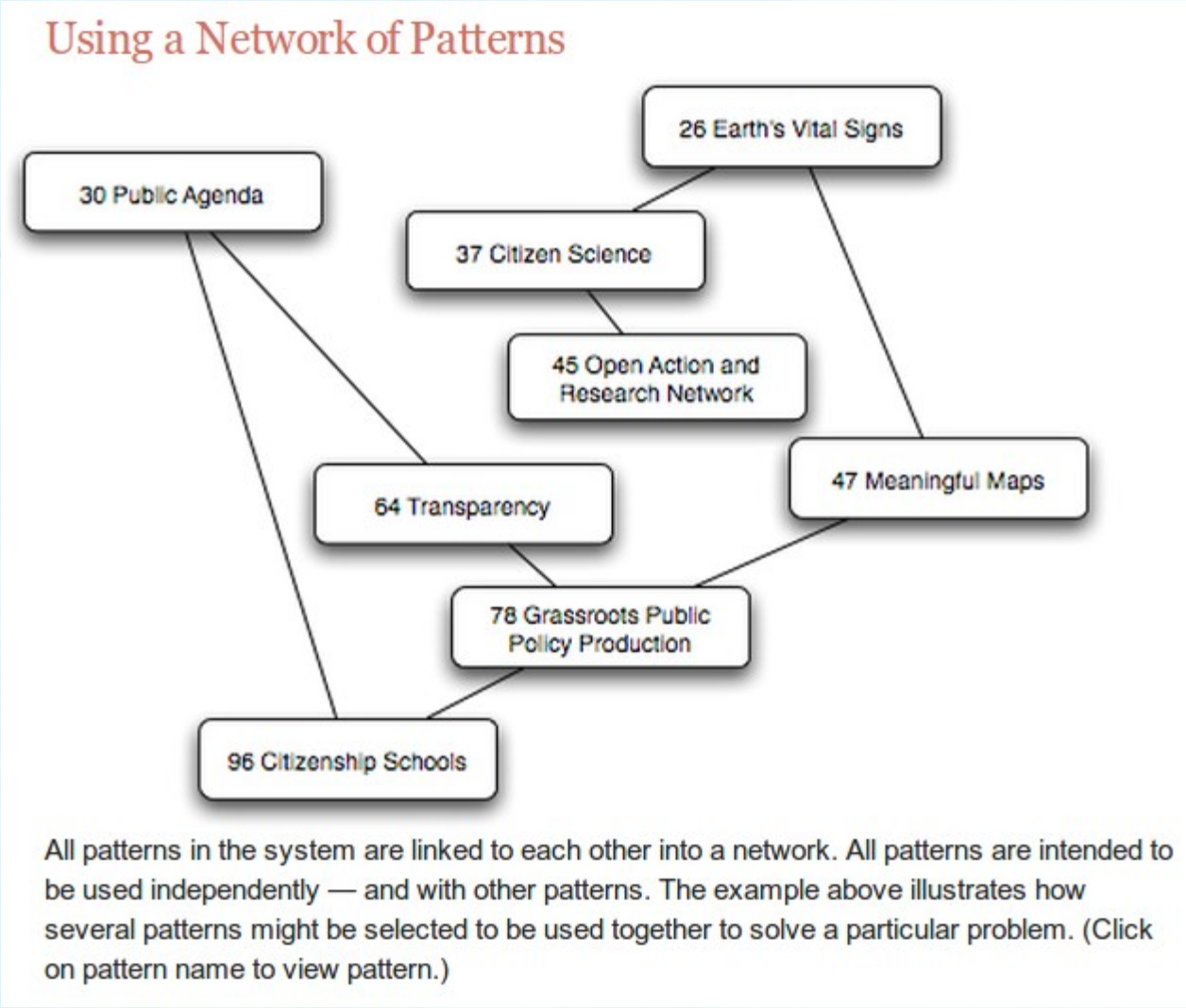
SEVEN RANDOM PATTERNS

Transforming Institutions
Public Agenda
Digital Emancipation
Transparency
Community Building Journalism
Engaged Tourism
Whistle Blowing

PATTERN LANGUAGE DOCUMENTS

Pattern Cards for workshops (under development)
Statement of Purpose

All patterns in the system are linked to each other into a network. All patterns are intended to be used independently — and with other patterns.



Source: http://www.publicsphereproject.org/patterns/

MEANINGFUL MAPS

in community action education engagement Orientation product social critique

Pattern number within this pattern set: 47

Andy Dearden
Sheffield Hallam University
Scot Fletcher
Handspring design, Sheffield, UK

Problem:

People are often unaware of the state of the world around them — especially "invisible", second-orde relationships. Many of the important issues for the community, the environment and for humanity are improve the world, we must understand the current situation, highlight the important factors, and help understand the issues. How can we collect up to date information and present it in a way that people understand?

Context:

This pattern is useful for community groups, advocacy groups and campaigns that are working to im around them. This might be about local environmental quality, promoting international respect for hun needs such as clean- water or nutrition, or access to opportunities, in a neighbourhood, city or countr target their resources carefully to achieve the maximum impact. They also want to communicate the encourage others to support their work. To be effective they need to reveal hidden relationships.

Discussion:

To act effectively to improve a situation, we need to understand that situation. Using a map can act monitoring the current situation in an area and showing how this is changing; and as a way of presen other people to raise their awareness and encourage them to support the work. The image above is b Apple Map of New York City http://www.greenapplemap.org

EARTH'S VITAL SIGNS

in case study collaboration education engagement globalism localism Orientation policy research for action servi Theory

Pattern number within this pattern set: 26

Jenny Frankel-Reed

Problem:

Societys great scientific capacity to measure and interpret the world and the role of humans in nature has failed translate into improved environmental stewardship. Modern environmental challenges are often difficult to see, time and space from their sources, and threaten global consequences. The increasing complexity and chronic r acute nature of today's environmental problems requires a revolution of decisionmaking the systematic integrat earths vital signs.

Context:

Signals detected by scientists about earth's natural patterns and processes and the impacts of humans on thes processes are earth's signs - indicators of what can be seen as either ecological health or the capacity of the ea accommodate human demands. The conditions of earth's systems tend to be worsening on a global scale, but dramatically from place to place. Human decisions about how to live on earth drive these trends and can potent reverse their negative directions.

Policymakers, public interest organizations, universities, and governments can utilize earth's signs to better ma human and environmental well-being. Policymakers' decisions about sustainable practices in land- and resource dependent sectors can be backed by scientific understanding about the effects of policies on resources. Citizen demand better environmental stewardship from their leaders at local to global scales with improved access to a translation of relevant earth information at the proper scale. Governments and enforcement bodies can strengthe monitoring capabilities and base development decisions on the latest information about trends in human impact

Agenda

A. Service systems
(science, management, engineering and design)

B. Pattern language
(c.f. pattern catalog)

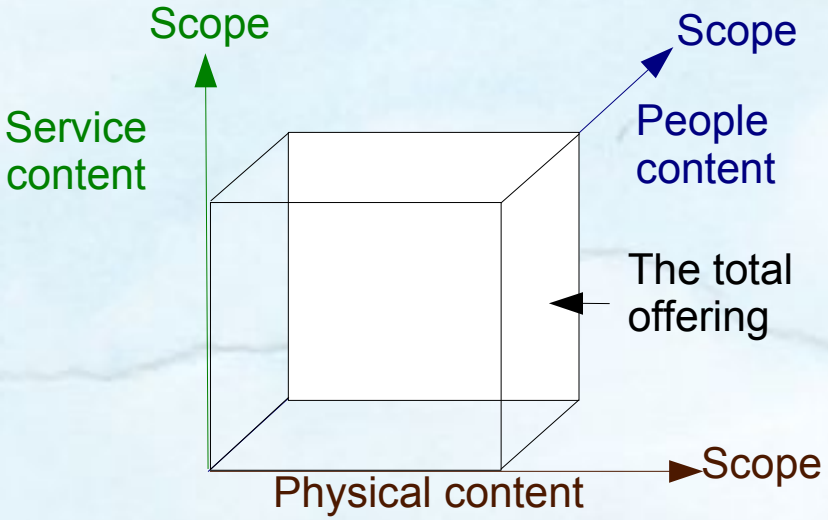
C. A starter set?
- 7 conditions from service systems science

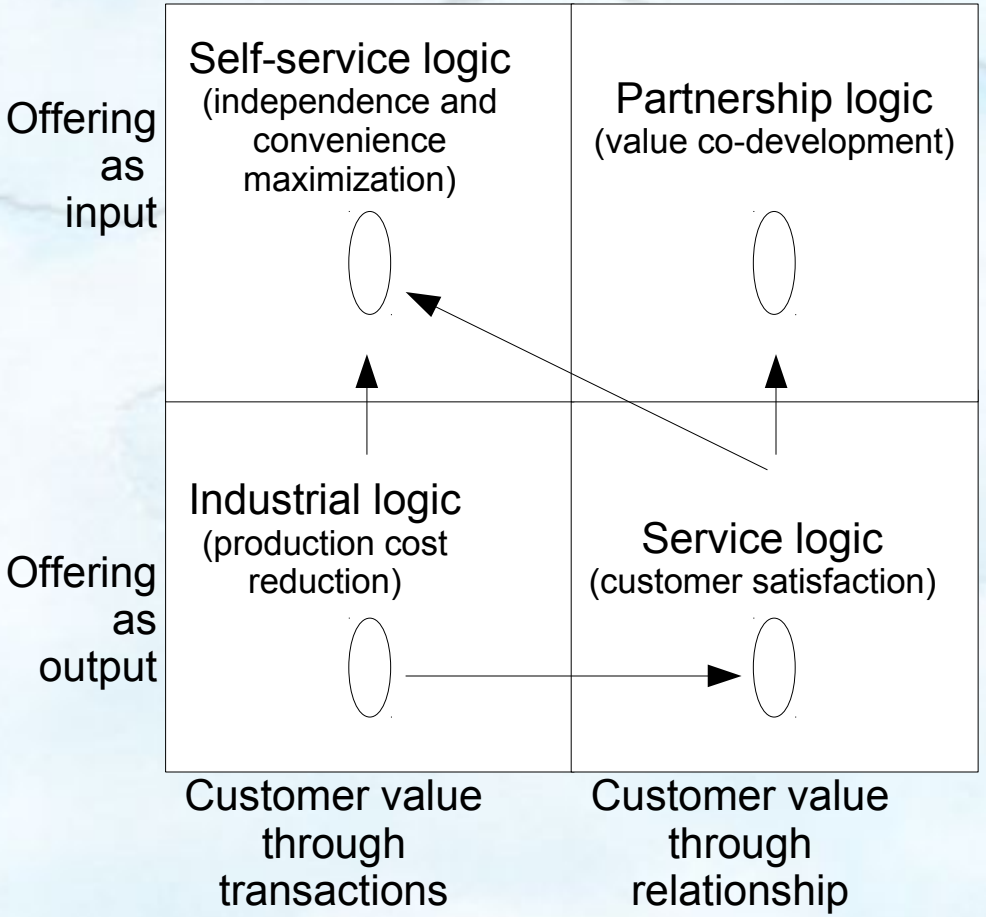
D. Collaboration
- Inquiring system
- Technologies

E. Next steps

1. Activity package mismatch:
   Theory of the offering
   (Normann and Ramirez)
2. Coordination fumble:
   Language action perspective
   (Winograd and Flores)
3. Change target discord:
   Reactivism, inactivism, preactivism, interactivism (Ackoff)
4. Resource scaling collapse:
   Supply side sustainability
   (Allen, Hoekstra, Tainter)
5. Environmental context shift:
   Causal texture theory (Emery and Trist)
6. Pacing layers trap:
   Coevolution and learning (Brand, Bateson)
7. Regeneration failure:
   Panarchy (Holling and Gunderson)

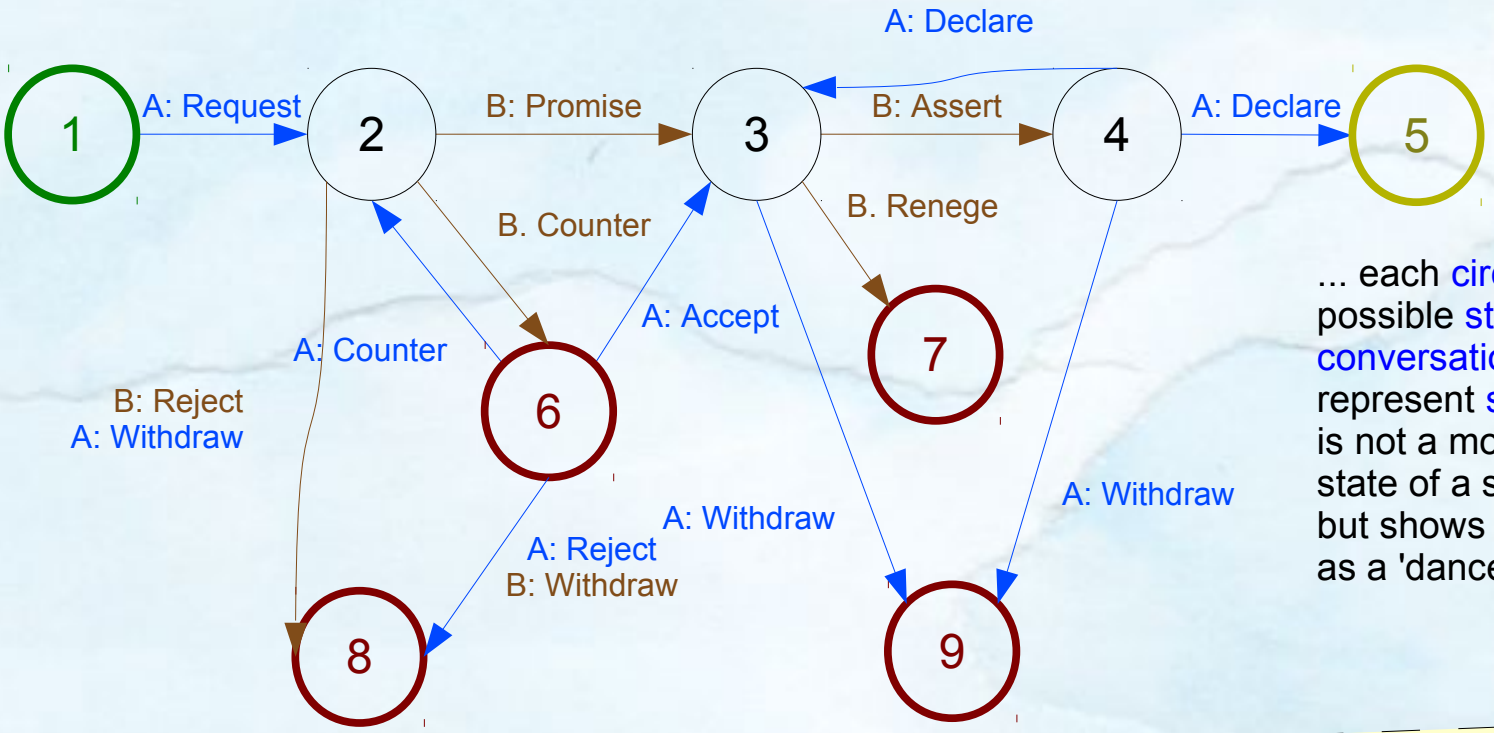
1. Activity package mismatch: Theory of the offering





Source: Rafael Ramírez and Johan Wallin. 2000. *Prime Movers: Define Your Business or Have Someone Define It Against You*. Chichester, England: Wiley.

2. Coordination fumble: Language action perspective

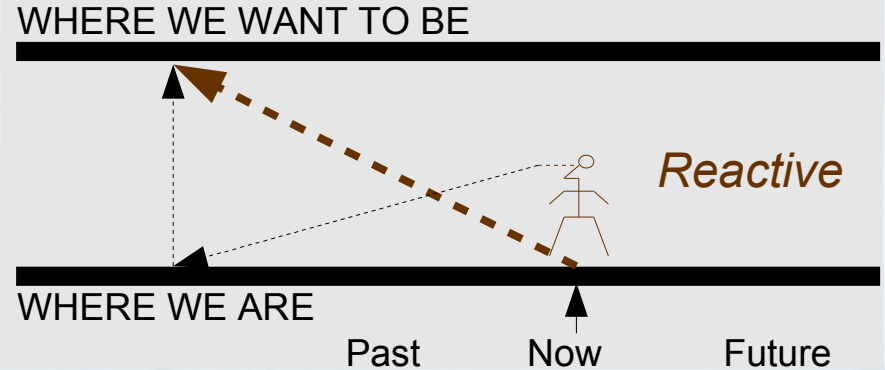


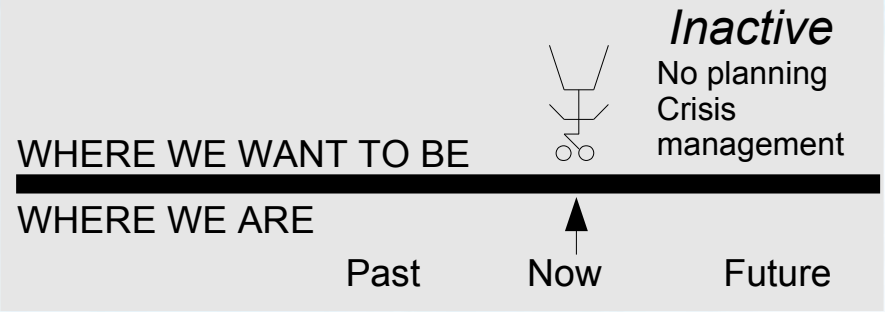
... each circle represents a possible state of the conversation and the lines represent speech acts. This is not a model of the mental state of a speaker or hearer, but shows the conversation as a 'dance.'

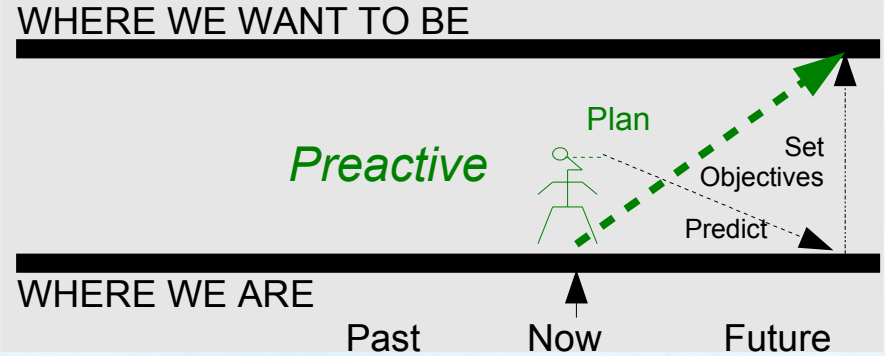
- Commitment to a deliverable
  - produce
- Commitment to a process
  - follow
- Commitment to a capability
  - provide
- Commitment to a relationship
  - contribute

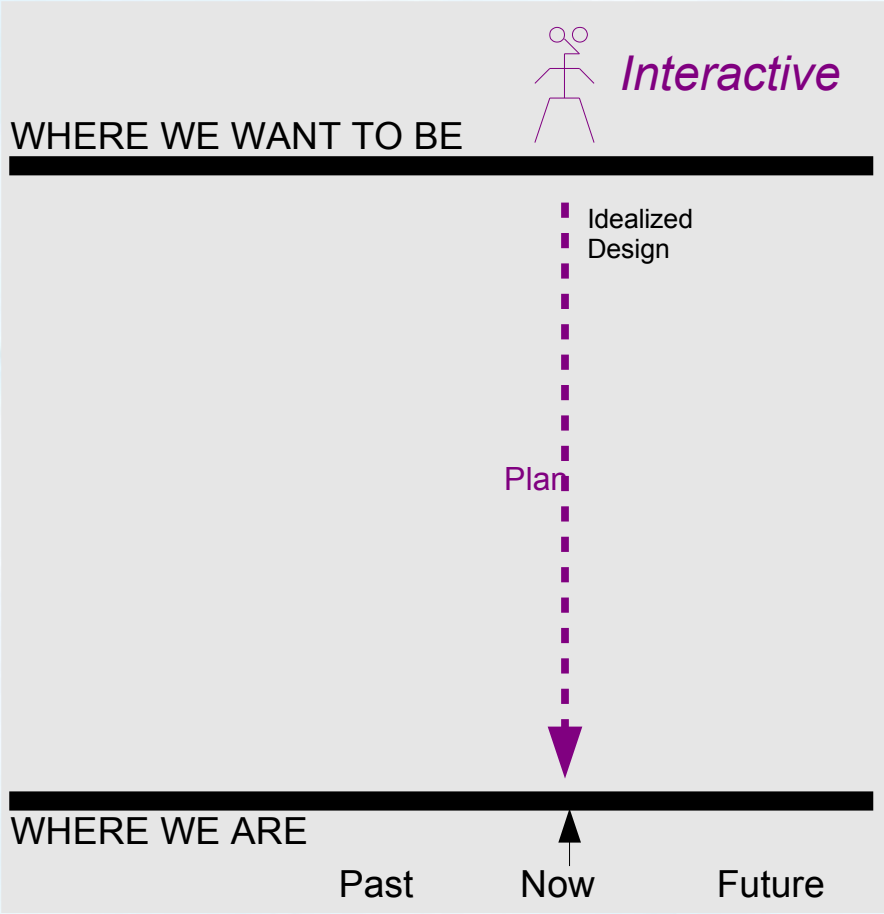
Source: Terry Winograd, and Fernando Flores. 1986. *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, NJ: Ablex; David Ing. 2008. "Offerings as Commitments and Context: Service Systems from a Language Action Perspective." In *Proceedings of the 12th International Conference of the UK System Society*. Oxford, UK.

3. Change target discord: Reactivism, inactivism, preactivism, interactivism



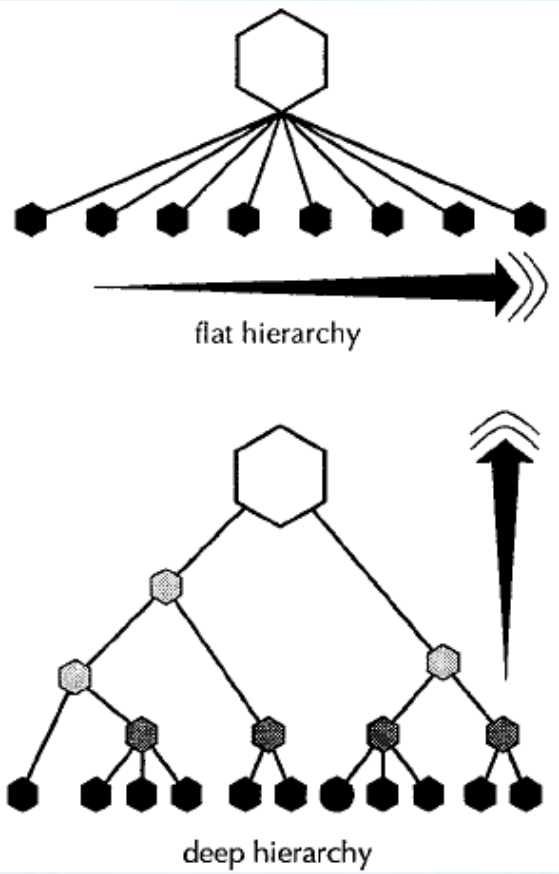






Source: Russell L. Ackoff. 1999. *Re-creating the Corporation: a Design of Organizations for the 21st Century*. Oxford University Press. http://books.google.ca/books?id=xylRdiAbpr8C .

4. Resource scaling collapse: Supply side sustainability



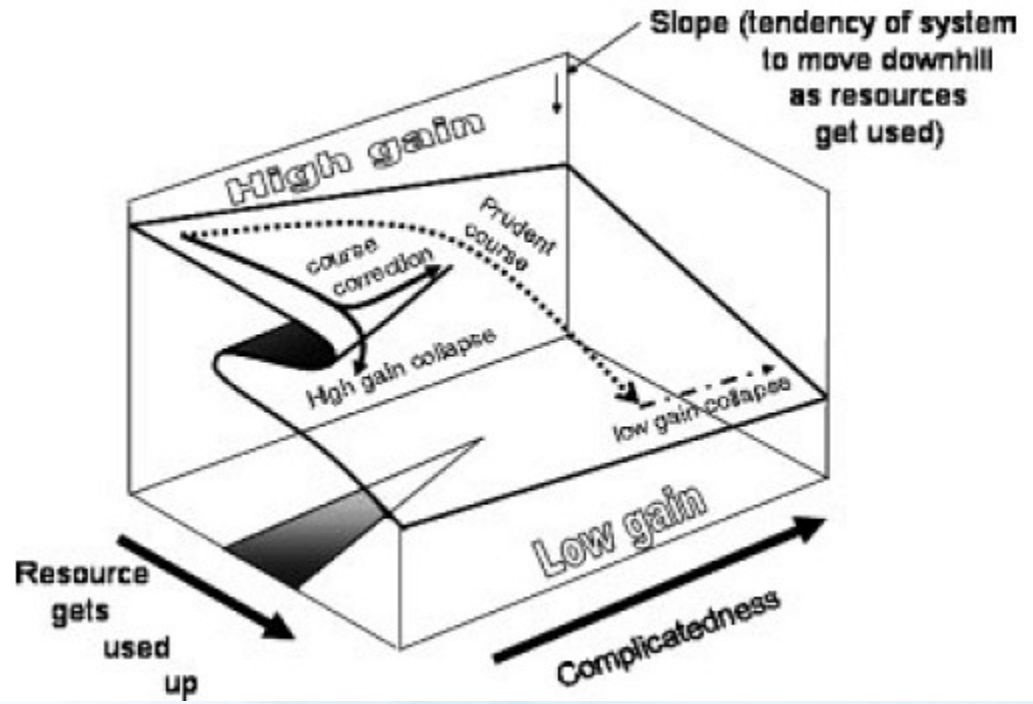
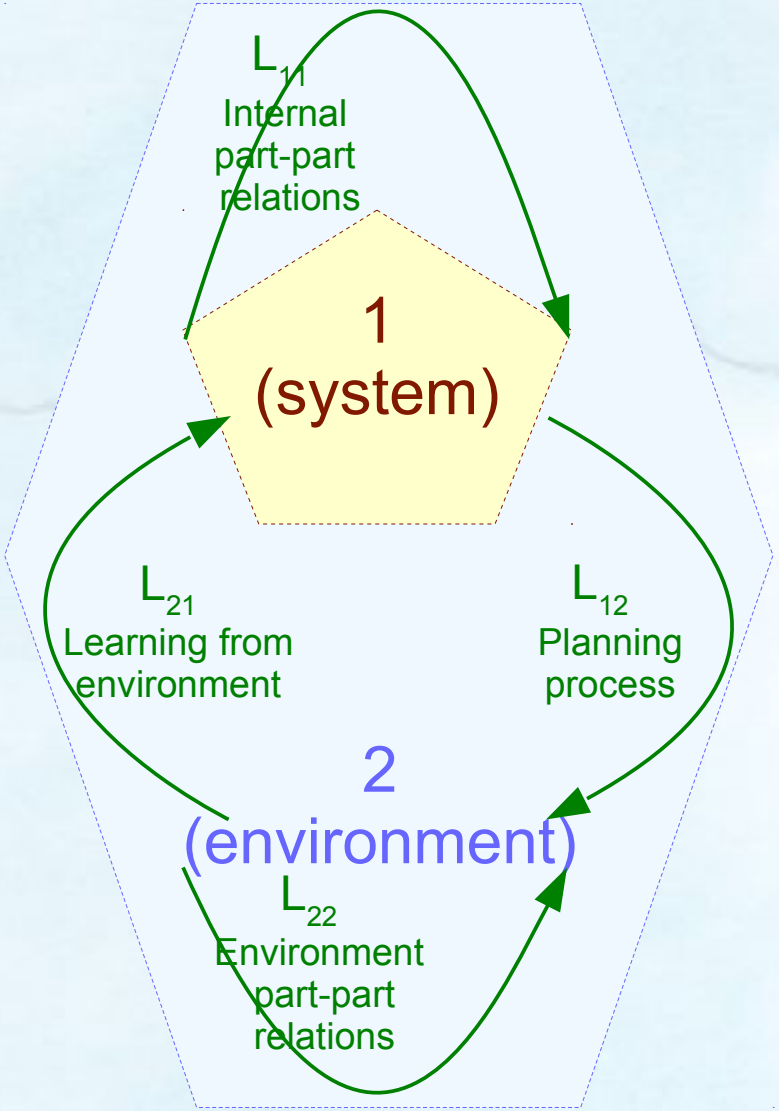


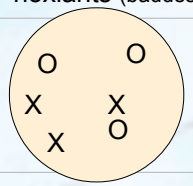
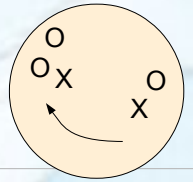
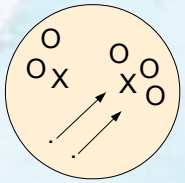
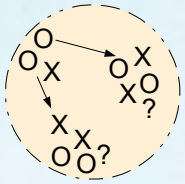
Figure 7. A representation of the tracks that lead from high to low to super low gain patterns. [Allen, Allen, Malek 2006]

Figure 3. The top hierarchy shows increases in complicatedness by increasing the structural elaboration. Structural elaboration is portrayed as widening the span in horizontal differentiation.
The bottom hierarchy shows increasing complexity, by an elaboration of organization. New levels appear as new constraints emerge as limits to the positive feedbacks of the emergent process. Elaboration or organization increases hierarchical depth. [Allen, Tainter, Hoekstra 1999]

Source: Timothy F. H. Allen, Joseph A. Tainter, and Thomas W. Hoekstra. 1999. "Supply-side Sustainability." *Systems Research and Behavioral Science* 16 (5): 403–427; Timothy F. H. Allen, Peter C. Allen, Amy Malek, John Flynn, and Michael Flynn. 2009. "Confronting Economic Profit with Hierarchy Theory: The Concept of Gain in Ecology." *Systems Research and Behavioral Science* 26 (5): 583–599.

5. Environmental context shift: Causal texture theory



	Where O = goals (goodies), X = noxiants (baddes)	
Type I. Random Placid		Goals and noxiants randomly distributed. Strategy is tactic. "Grab it if it's there". Largely theoretical of micro, design, e.g. concentration camps, conditioning experiments. Nature is not random.
Type 2. Clustered Placid		Goals and noxiants are lawfully distributed – meaningful learning. Simple strategy – maximize goals, e.g. use fire to produce new grass. Most of human span spent in this form. Hunting, gathering, small village. What people mean by the "good old days".
Type 3. Disturbed Reactive		Type 2 with two or more systems of one kind *competing* for the same resources. Operational planning emerges to out-manoeuvre the competition. Requires extra knowledge of both Ss and E. E is stable so start with a set of givens and concentrate on problem solving for win-lose games. Need to create insturments that are variety-reducing (foolproof) – elements must be standardized and interchangeable. Birth of bureacractic structures where people are redundant parts. Concentrate power at the top – strrategy becomes a power game.
Type 4. Turbulent		Dynamic, not placid/stable. Planned change in type 3 triggers off unexpected social processes. Dynamism arises from the field itself, creating unpredictability and increasing *relevant uncertainty* and *its continuities*. Linear planning impossible, e.g. whaling disrupted reproduciton, people react to being treated as parts of machine. Birth of open systems thinking, ecology, and catastrophe theory.

Source: Fred E. Emery, and Eric L. Trist. 1965. "The Causal Texture of Organizational Environments." *Human Relations* 18 (1) (February): 21–32. doi:10.1177/001872676501800103. http://dx.doi.org/10.1177/001872676501800103.

6. Pacing layers trap: Coevolution and learning

SITE

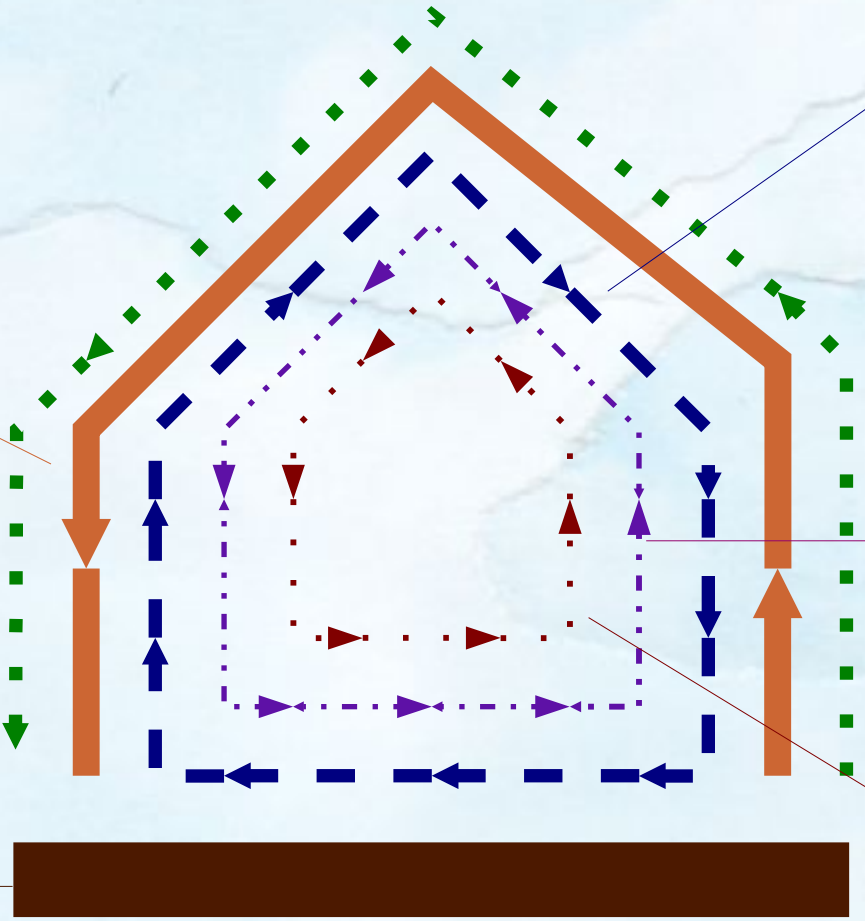
This is the geographical setting, the urban location, and the legally defined lot, whose boundaries outlast generations of ephemeral buildings. "Site is eternal", Duffy agrees.

STRUCTURE

The foundation and load-bearing elements are perilous and expensive to change, so people don't. These are the building. Structural life ranges from 30 to 300 years (but few buildings make it past 60, for other reasons).

SKIN

Exterior surfaces now change every 20 years or so, to keep up with fashion or technology, or for wholesale repair. Recent focus on energy costs has led to re-engineered Skins that are air-tight and better-insulated.



SERVICES

These are the working guts of a building: communications wiring, electrical wiring, plumbing, sprinkler system, HVAC (heating, ventilation, and air conditioning), and moving parts like elevators and escalators. They wear out or obsolesce every 7 to 15 years. Many buildings are demolished early if their outdated systems are too deeply embedded to replace easily.

SPACE PLAN

The interior layout, where walls, ceilings, floors, and doors go. Turbulent commercial space can change every 3 years; exceptionally quiet homes might wait 30 years.

STUFF

Chairs, desks, phones, pictures; kitchen appliances, lamps, hair brushes; all the things that twitch around daily to monthly. Furniture is called mobilia in Italian for good reason.

Source: Stewart Brand. 1994. *How Buildings Learn: What Happens after They're Built*. New York: Viking.

7. Regeneration failure: Panarchy

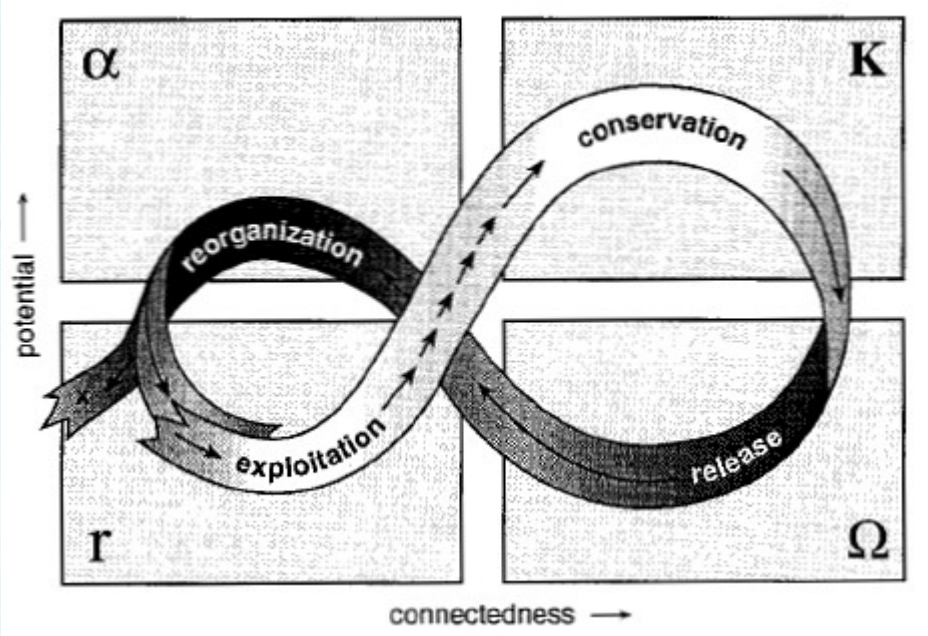


Figure 4. A stylized representation of the four ecosystem functions (r, K, Ω, α) and the flow of events among them.

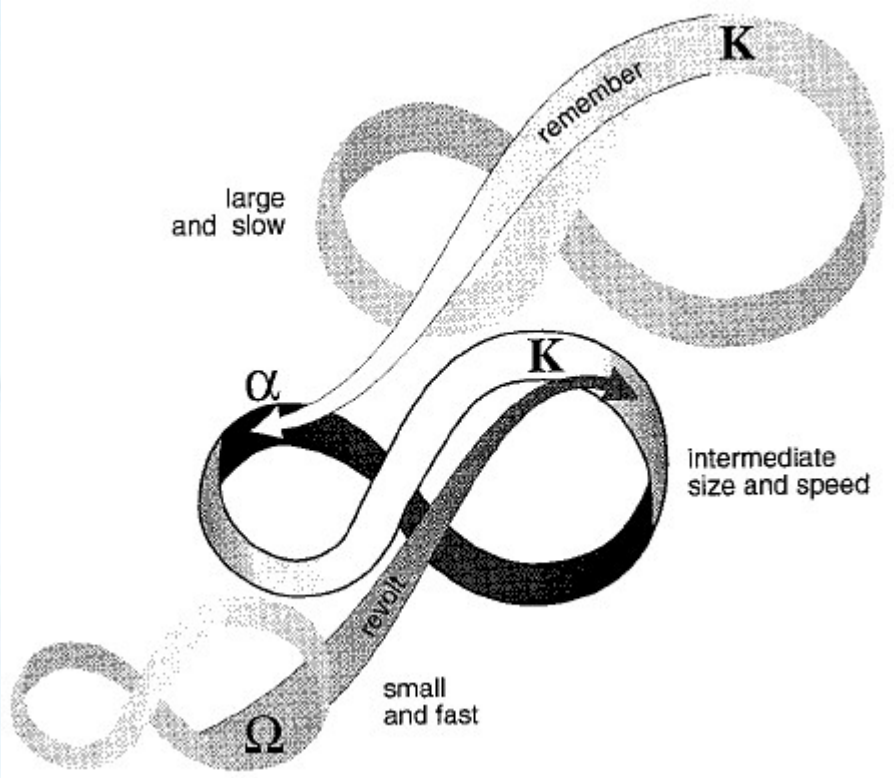


Figure 7. Panarchical connections. [....] the "revolt" connection ...can cause a critical change in one cycle to cascade up to a vulnerable stage in a larger and slower one. The ... "remember" connection ... facilitates renewal by drawing on the potential that has been accumulated and stored in a larger, slower cycle.

Source: C. S. Holling 2001. "Understanding the Complexity of Economic, Ecological, and Social Systems." *Ecosystems* 4 (5): 390–405. doi:10.1007/s10021-001-0101-5. http://dx.doi.org/10.1007/s10021-001-0101-5.

Agenda

A. Service systems
(science, management, engineering and design)

B. Pattern language
(c.f. pattern catalog)

C. A starter set?
- 7 conditions from service systems science

D. Collaboration
- Inquiring system
- Technologies

E. Next steps

1. The design of inquiring systems
(West Churchman via Ian Mitroff)

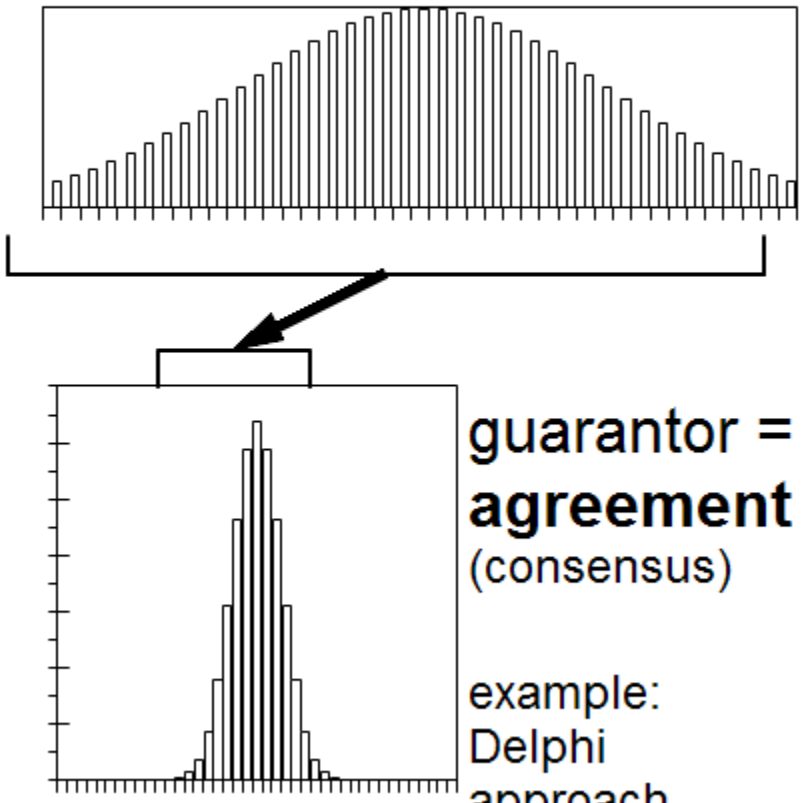
2. Design thinking and
Jungian psychological types

3. New technologies for collaboration:
- Git and Github
- Federated wiki
- Kune (Apache Wave ← Google Wave)
- Etherpad Lite

Design of inquiring systems: Ways of knowing (1, 2)

The first way of knowing

Inductive-Consensual IS

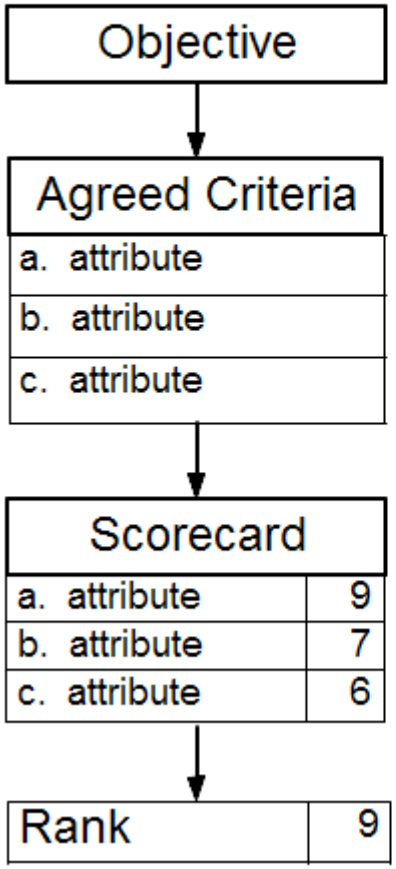


guarantor = **agreement** (consensus)

example: Delphi approach

The second way of knowing

Analytic-Deductive IS



guarantor = **logical consistency** (fact nets)

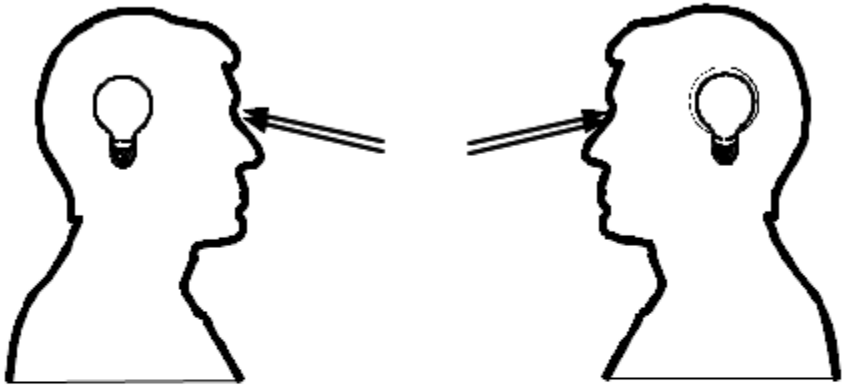
example: finding the "best man" for the job

Source: Ian I. Mitroff, and Harold A. Linstone. 1993. *The Unbounded Mind: Breaking the Chains of Traditional Business Thinking*. Oxford U Press.

Design of inquiring systems: Ways of knowing (3, 4)

The third way of knowing

Multiple Realities IS



model + data as inseparable whole

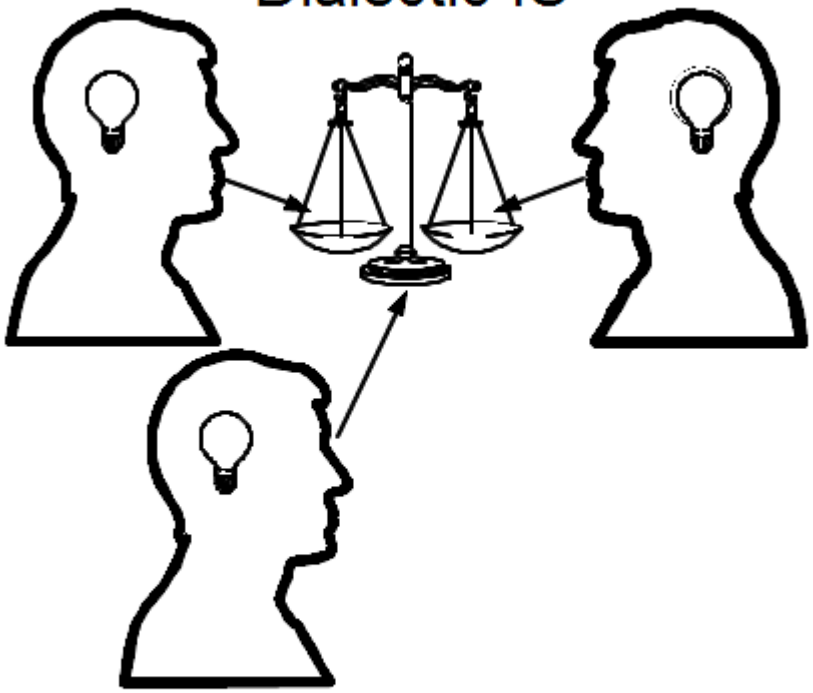
For human beings to have experience or gain knowledge about the external world, something must be built into the internal structure of their minds ...

guarantor = (ability to see) range of views (representations)

example: disciplinary views of the causes of the drug problem

The fourth way of knowing

Dialectic IS

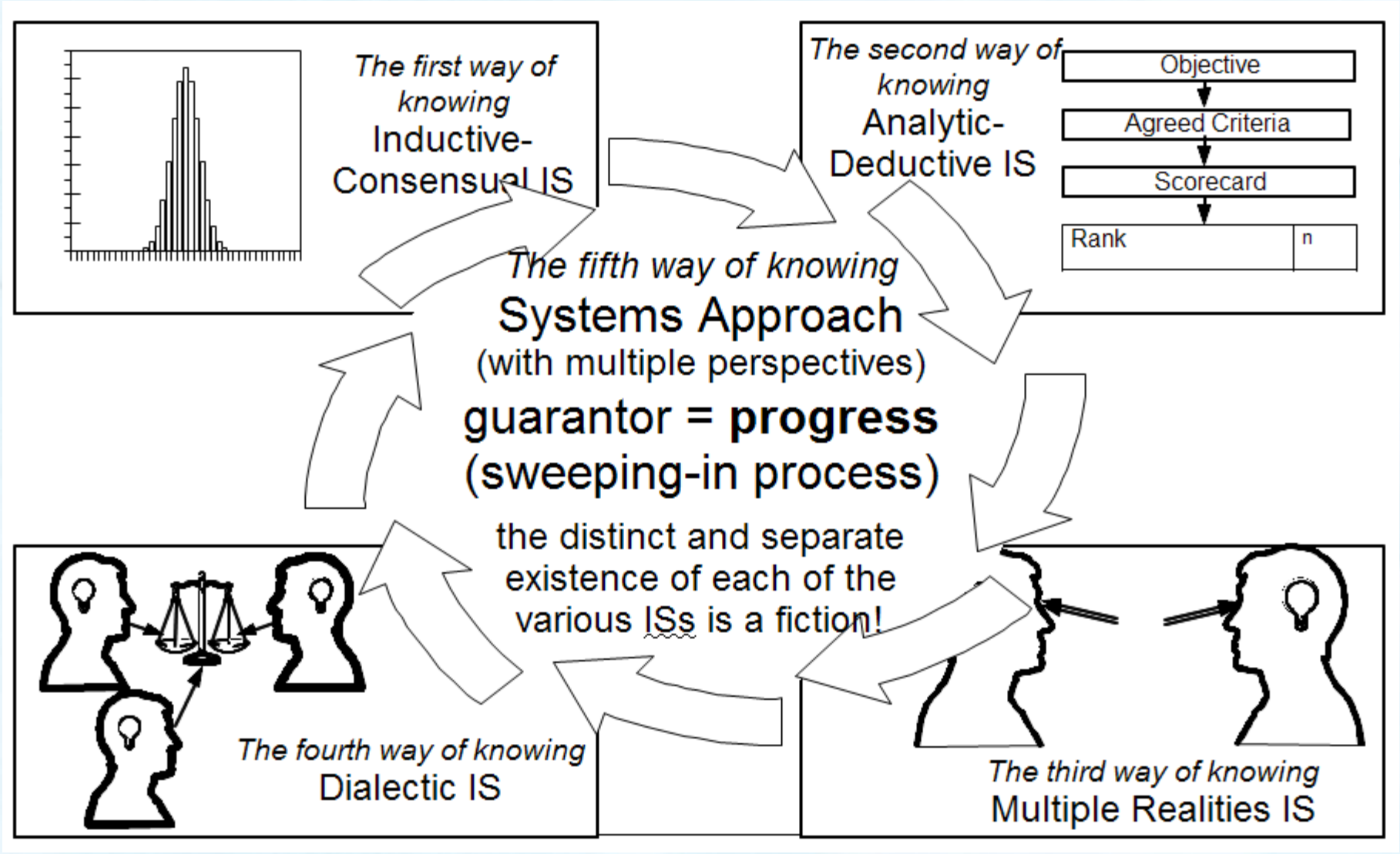


guarantor = conflict

example: challenging assumptions of what skid row housing should be

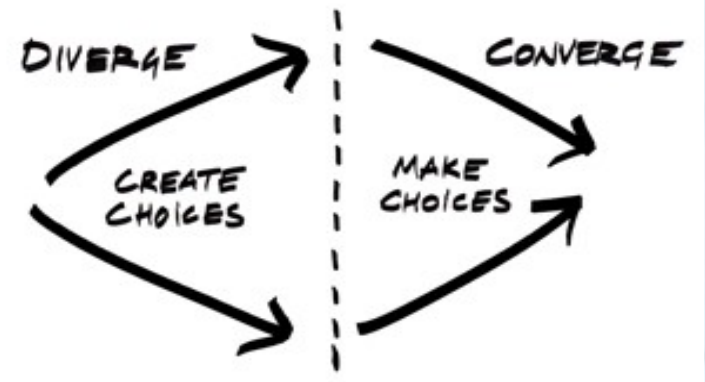
Source: Ian I. Mitroff, and Harold A. Linstone. 1993. *The Unbounded Mind: Breaking the Chains of Traditional Business Thinking*. Oxford U Press.

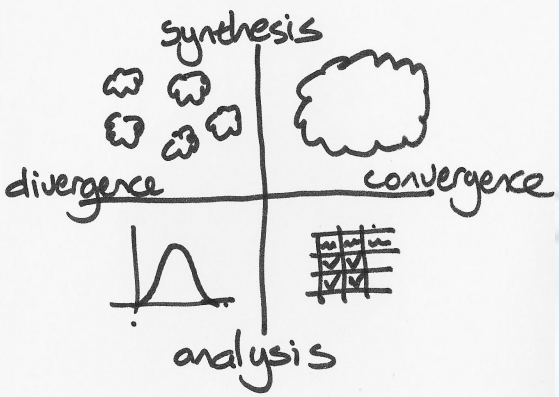
Design of inquiring systems: Ways of knowing (5)



Source: Ian I. Mitroff, and Harold A. Linstone. 1993. *The Unbounded Mind: Breaking the Chains of Traditional Business Thinking*. Oxford U Press.

Design Thinking: Divergent-Convergent, Synthesis-Analysis





Design thinking is different and therefore it feels different.

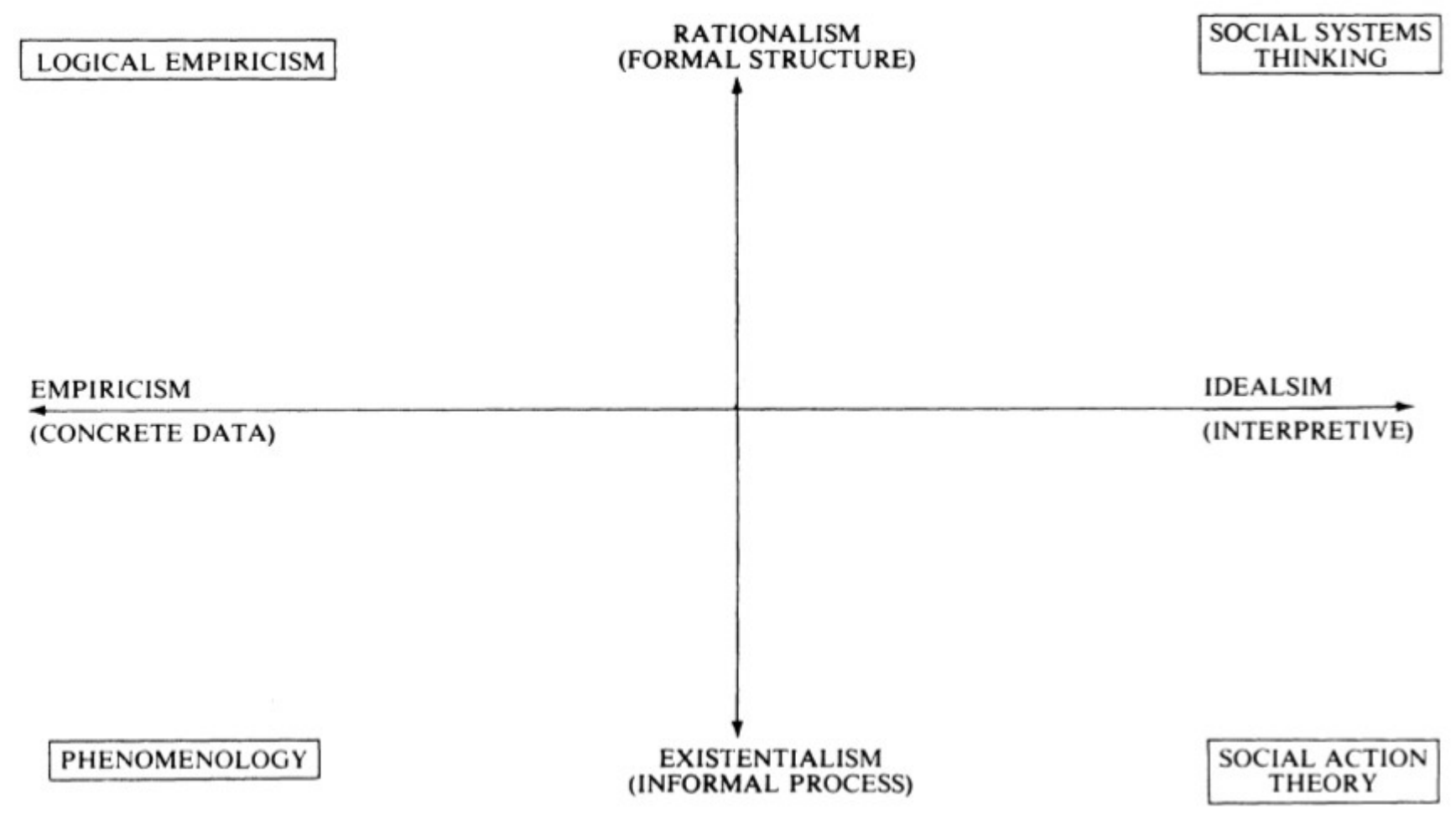
Firstly it is not only convergent. It is a series of divergent and convergent steps. **During divergence we are creating choices and during convergence we are making choices**. For people who are looking to have a good sense of the answer, or at least a previous example of one, before they start divergence is frustrating. It almost feels like you are going backwards and getting further away from the answer but this is the essence of creativity. Divergence needs to feel optimistic, exploratory and experimental but it often feels foggy to people who are more used to operating on a plan. Divergence has to be supported by the culture.

The second difference is that design thinking relies on **an interplay between analysis and synthesis, breaking problems apart and putting ideas together**. Synthesis is hard because we are trying to put things together which are often in tension. Less expensive, higher quality for instance. [….]

Designers have evolved visual ways to synthesize ideas and this is another one of the obstacles for those new to design thinking; a discomfort with visual thinking. A sketch of a new product is a piece of synthesis. So is a scenario that tells a story about an experience. A framework is a tool for synthesis and design thinkers create visual frameworks that in themselves describe spaces for further creative thinking.

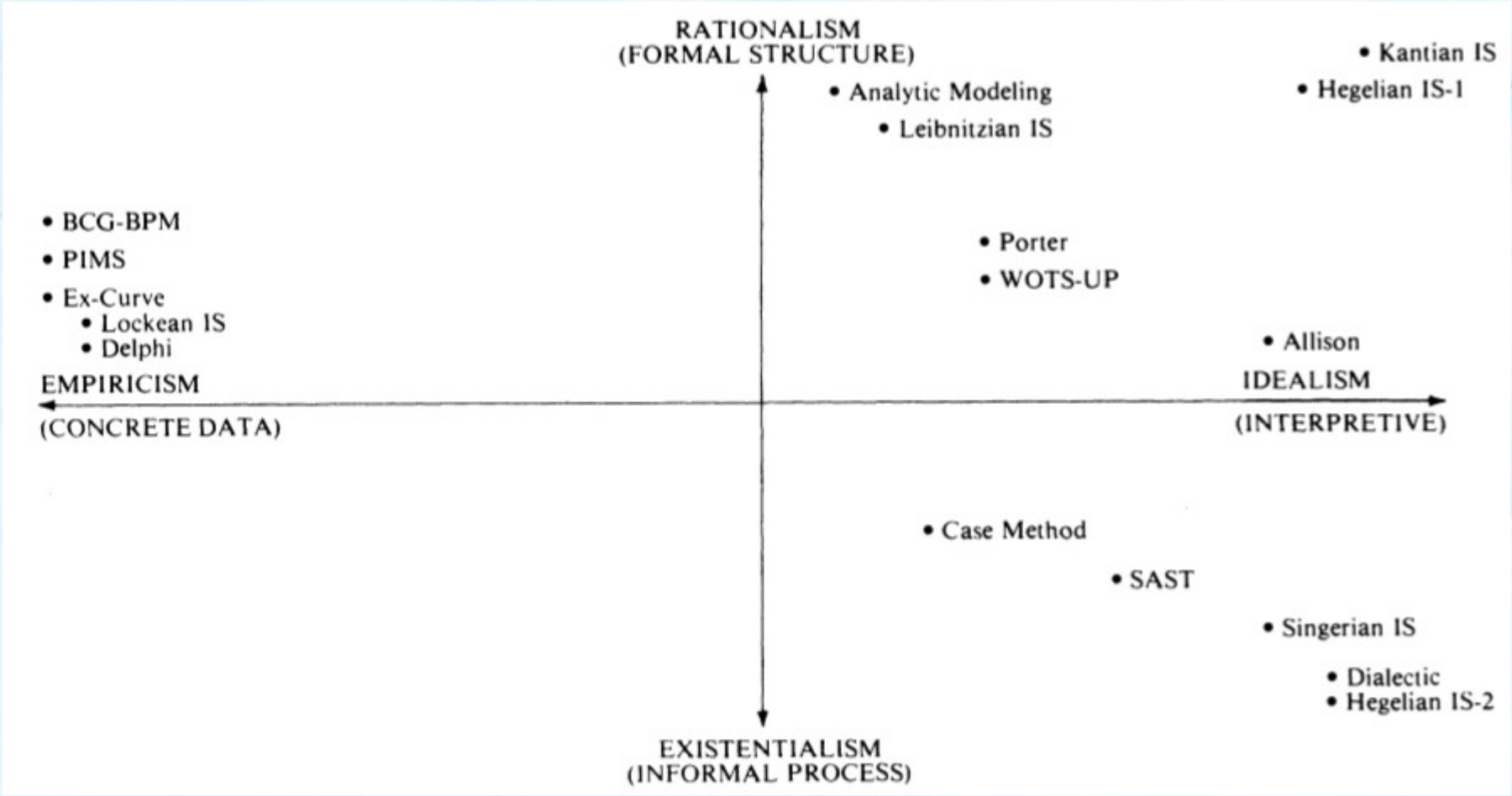
Source: Tim Brown "What does design thinking feel like?" *Design Thinking* (blog), Sept. 7, 2008 at http://designthinking.ideo.com/?p=51 ; "Why Social Innovators Need Design Thinking", *Stanford Social Innovation Review*, Nov. 15, 2011 at http://www.ssireview.org/blog/entry/why_social_innovators_need_design_thinking .

Business Policy Metaphysics (Mitroff and Mason 1982)
Figure 1: Basic Philosophical Stances



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.

Business Policy Metaphysics (Mitroff and Mason 1982)
Figure 2: Some Approaches to Policy as Applied Metaphysics



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.

	Technical (T)	Organizational (O)	Personal (P)
Worldview	Science-technology	Unique group or institutional view	Individual, the self
Objective	Problem solving, product	Action, process, stability	Power, influence, prestige
System focus	Artificial construct	Social	Genetic, psychological
Mode of inquiry	Observation, analysis: data and models	Consensual, adversary bargaining and compromise	Intuition, learning, experience
Ethical basis	Logic, rationality	Justice, fairness	Morality
Planning horizon	Far (low discounting)	Intermediate (moderate discounting)	Short for most (high discounting for most)
Other descriptors	Cause and effect Optimization, cost-benefit analysis Quantification, trade-offs Use of probabilities, averages, statistical analysis, expected value Problem simplified, idealized Need for validation, replicability Conceptualization, theories Uncertainties noted	Agenda (problem of the moment) Satisficing Incremental change Reliance on experts, internal training of practitioners Problem delegated and factored, issues and crisis management Need for standard operating procedures, routinization Reasonableness Uncertainty used for organizational self-preservation	Challenge and response, leaders and followers Ability to cope with only a few alternatives Fear of change Need for beliefs, illusions, misperception of probabilities Hierarchy of individual needs (survival to self-fulfillment) Need to filter out inconsistent images Creativity and vision by the few, improvisation Need for certainty
Criteria for "acceptable risk"	Logical soundness, openness to evaluation	Institutional compatibility, political acceptability, practicality	Risk aversion
Scenario typology	Probable	Preferable	Possible
•Criterion	analysis (reproducible)	value (explicative)	image (plausible)
•Orientation	exploratory (extrapolative)	normative (prescriptive)	visionary
•Mode	structural	participative	perceptual
•Creator	think-tank teams	stakeholders	individuals
Communications	Technical report, briefing	Insider language, outsiders' assumptions often misperceived	Personality, charisma desirable

Source: Harold A. Linstone, and Murray Turoff. 2011. "Delphi: A Brief Look Backward and Forward." *Technological Forecasting and Social Change* 78 (9) (November): 1712–1719. doi:10.1016/j.techfore.2010.09.011. http://dx.doi.org/10.1016/j.techfore.2010.09.011.

Management Information Systems (with a broader view of knowledge, effectiveness, action, and purpose)

Psychological Type
- (a) Thinking-Sensation
- (b) Thinking-Intuition
- (c) Feeling-Sensation
- (d) Feeling-Intuition

Class of Problems
- (a) Structured
  - (1) Decisions under certainty
  - (2) Decisions under risk
  - (3) Decisions under uncertainty
- (b) Unstructural-"Wicked" Decision

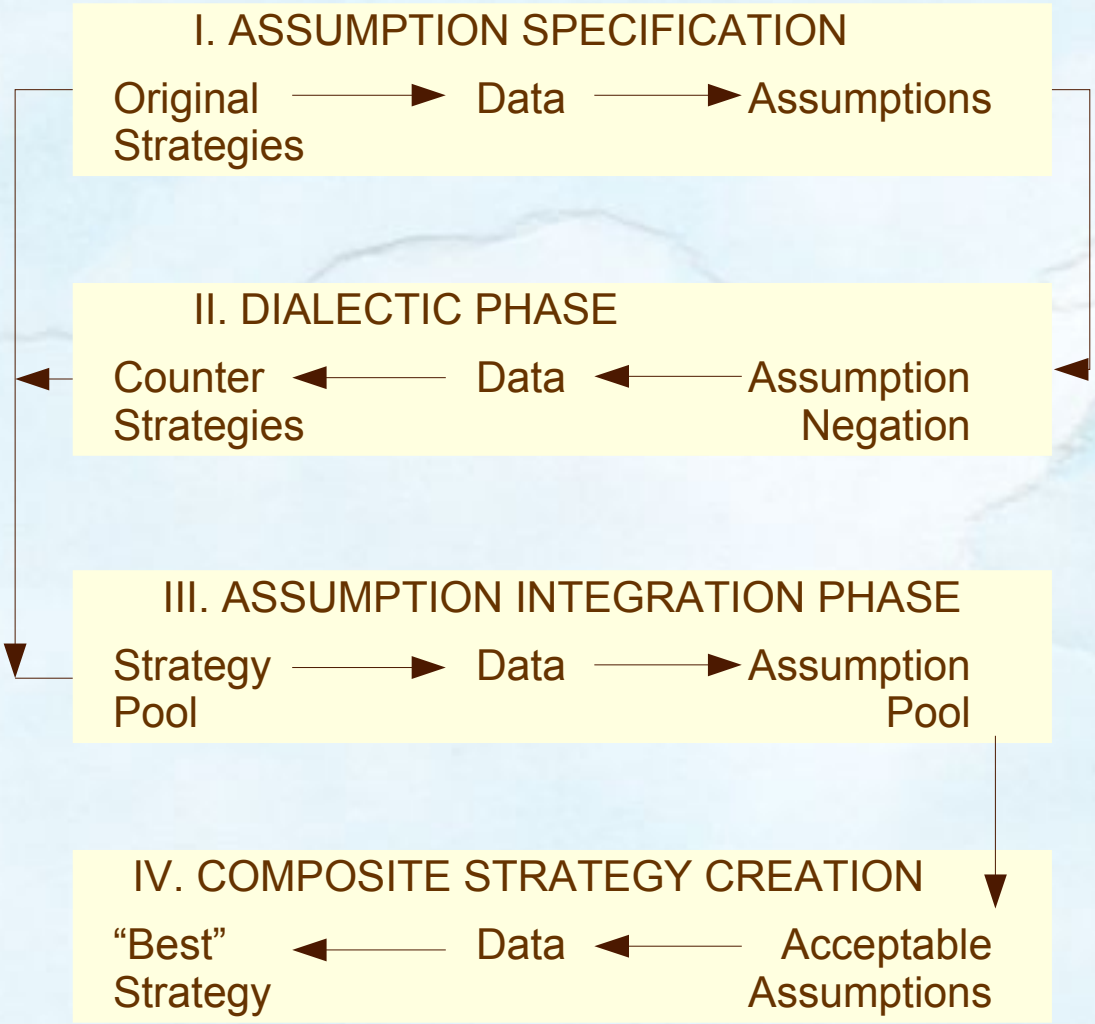
Organizational Context or Organizational Class of Problem
- (a) Strategic planning
- (b) Management control
- (c) Operational control

Method of Evidence Generation and Guarantor of Evidence-Inquiring Systems (IS)
- (a) Lockean IS (Data Based)
- (b) Leibnitzian IS (Model Based)
- (c) Kantian IS (MIultiple Models)
- (d) Hegelian IS (Deadly Enemy-Conflicting Models)
- (e) Singerian-Churchmanian IS (Learning Systems)

Modes of Presentation
- (a) Personalistic
  - (1) Drama – Role Plays
  - (2) Art – Graphics
  - (3) One-to-One contact group information
- (b) Impersonalistic
  - (1) Company reports
  - (2) Abstract models – computerized information systems

Source: Richard O. Mason and Ian I. Mitroff. 1973. "A Program for Research on Management Information Systems." Management Science 19 (5) (January 1): 475–487. doi:10.2307/2629445. http://www.jstor.org/stable/2629445.

Strategic Assumption Surfacing and Testing



By working backwards to underlying assumptions, the proposed process ... requires that each strategy contain in addition to supporting data a list of assumptions (i.e., given conditions, events, or attributes that are or must be taken as true) which implicitly underlie the strategy.

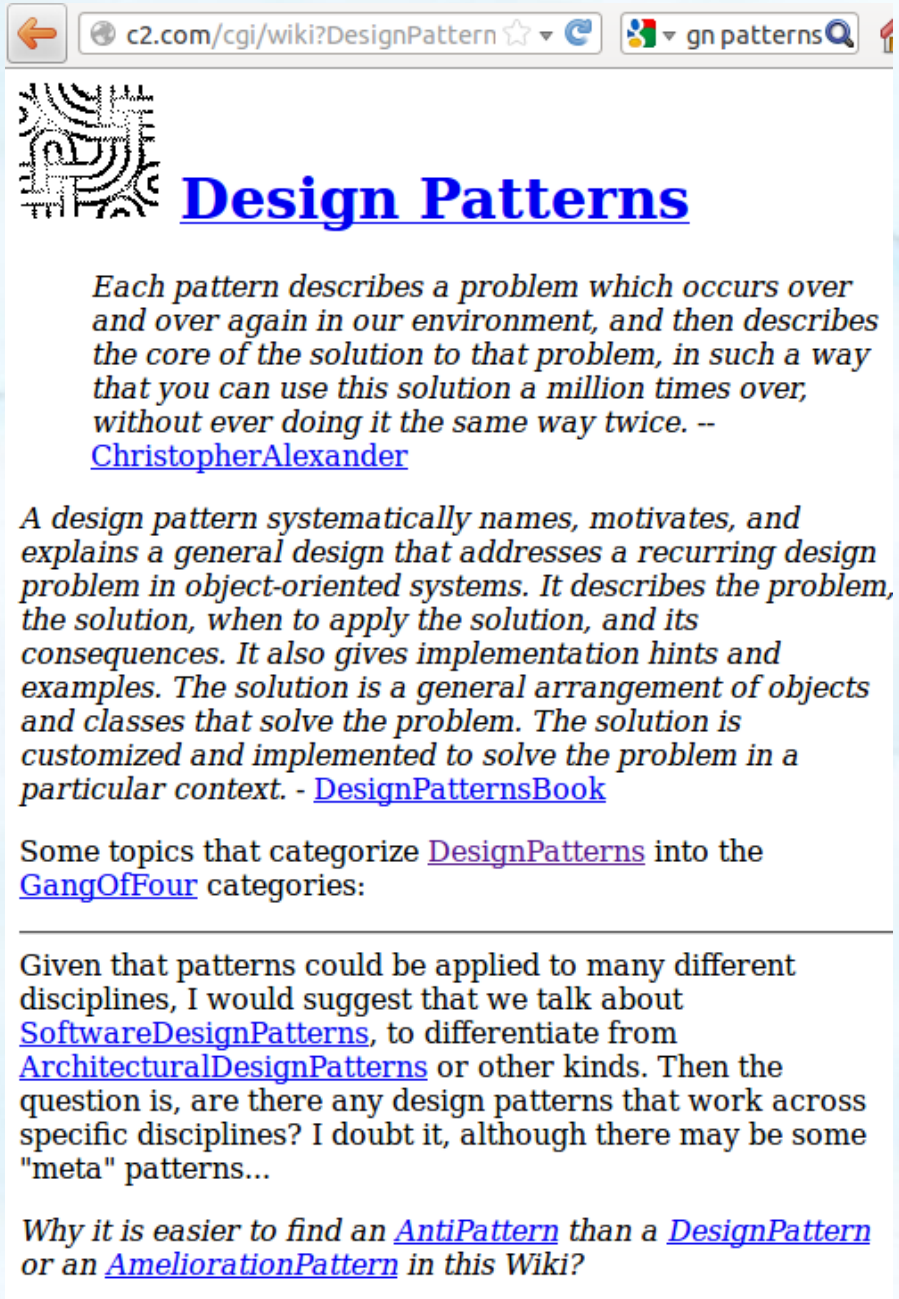
... each assumption previously identified is negated and reformulated as a counter-assumption that negates the spirit of the original statement. If the counter-assumption is implausible, it is dropped. Those counter assumptions which one can conceive of as being true or plausible in some circumstances are then examined individually and collectively to see if they can be used as a basis both for defining and deducing an entirely new strategy.

Instead of trying to resolve differences in strategies directly at the resultant level of strategy, the process concentrates on negotiating an acceptable set of assumptions that the decision makers are prepared to take as given conditions for the formulation of the problem.

… development operates on a more rational basis as defined in traditional problem solving and decision theory terms. The composite set of acceptable assumptions can be used as an explicit foundation upon which the problem can be defined.

Source: Ian I.Mitroff and James R. Emshoff. 1979. "On Strategic Assumption-Making: A Dialectical Approach to Policy and Planning." *The Academy of Management Review* 4 (1) (January): 1. doi:10.2307/257398. http://dx.doi.org/10.2307/257398.

Wiki was invented to support pattern language collaborations



C2 Portland Pattern Repository → Hillside Group

c2.com/ppr/

Portland Pattern Repository

We're writing about computer programs in a new stylistic form called pattern languages. The form has many internal references which map well to hypertext links. We've added links to published (or soon to be published) documents. Short summaries appear in the...

- Pattern Language Catalog

We've also created a space for exploring the not-quite-yet patterns we all carry around in our heads...

- People, Projects & Patterns

The Hillside Group's Patterns Home Page lists other pattern resources including papers, books, conferences.

New survey results are in. This form tallies survey responses as they are made. Have a look to see what people like about

hillside.net/patterns/

THE HILLSIDE GROUP

Hilltop | Books | Contact | Conferences | Patterns | Vision | Wiki

MAIN MENU

- Hilltop
- Books
- Contact
- Conferences
- **Patterns**
  - Books
  - Patterns Catalog
  - About Patterns
  - TPLoP
  - Education
  - Mailing Lists
  - Writing
  - Tools
  - Links
- Vision

You are here: Home > Patterns

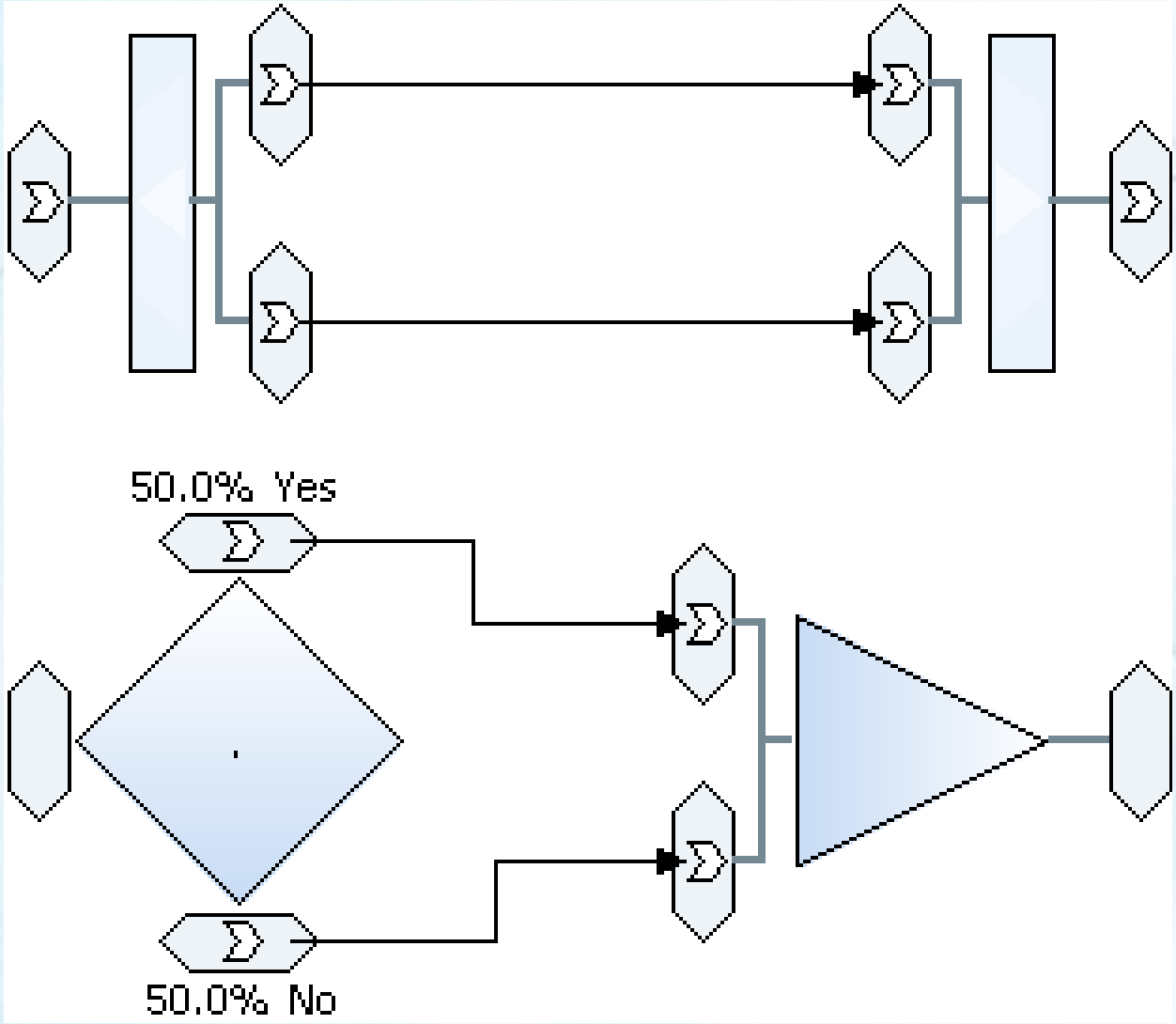
DESIGN PATTERNS LIBRARY

Welcome to the patterns home page. It is a source for information languages. If you are new to patterns, James Coplien and Richard

Patterns and Pattern Languages are ways to describe best practic designs, and capture experience in a way that it is possible for ot this experience. The Hillside Group takes pleasure in sponsoring r PLoP conferences that are provided for the betterment of the pat community.

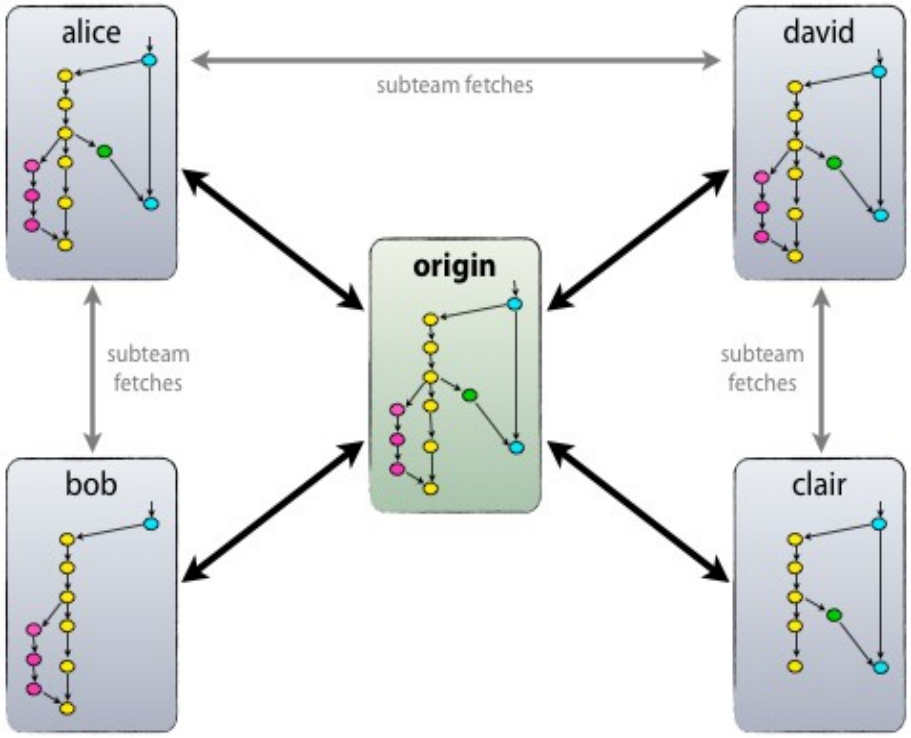
Fundamental to any science or engineering discipli common vocabulary for expressing its concepts, a The goal of patterns within the software communit software developers resolve recurring problems en development. Patterns help create a shared langua experience about these problems and their solutio their relationships lets us successfully capture the understanding of good architectures that meet the pattern language for conveying the structures and us to intelligibly reason about them. The primary f on creating a culture to document and support sou

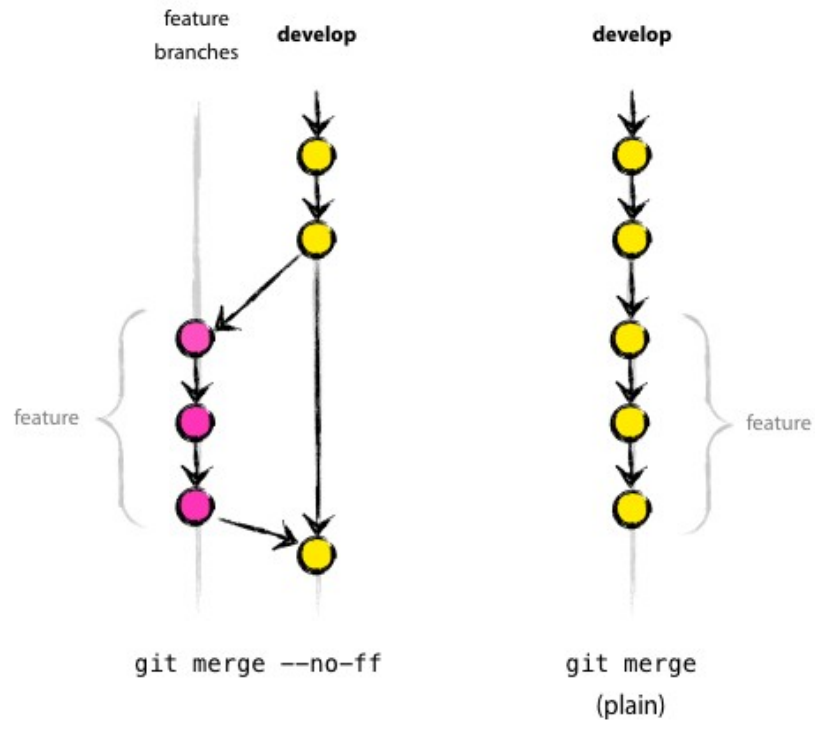
Collaboration evolution as Fork-Join, and Branch-Merge



Source: David Ing, *Open source with private source: coevolving architectures, styles and subworlds in business* (forthcoming)

Git and Github: Work Organization





Git is architected as decentralized, with an origin from where individuals may push to and pull from (as well as amongst each other).

This organization of work enables individuals to first work independently, and then subsequently discuss merging their changes together.

Source: Vincent Driessen, “A successful Git branching model” January 05, 2010 at http://nvie.com/posts/a-successful-git-branching-model/

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.

arstechnica.com/gaming/2009/10/google-wave-we-can

google wave

ars technica

Software Engineer Object Oriented Programming
JQuery Lead Developer Relational Databases J2EE
CSS3 SQL Java GitHub C# .Net AJAX Visual Studio
Software Engineer Object Oriented Programming
JQuery Lead Developer Relational Databases J2EE

MAIN MENU ▾ MY STORIES: 25 ▾ FORUMS SUBSCRIBE JOBS

OPPOSABLE THUMBS / GAMING & ENTERTAINM

Google Wave: we came, we saw, we played D&D

Give a bunch of computer geeks a new medium to

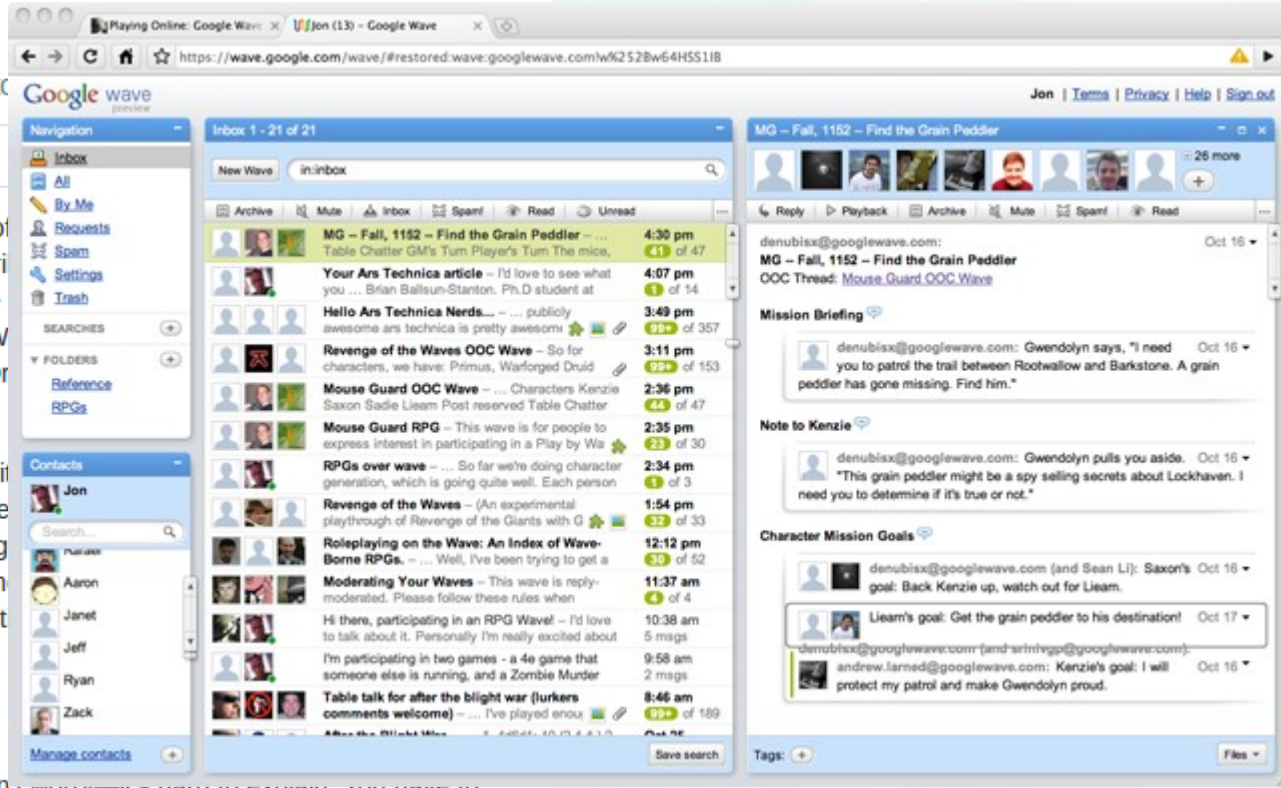
by **Jon Stokes** - Oct 26 2009, 9:00pm EDT

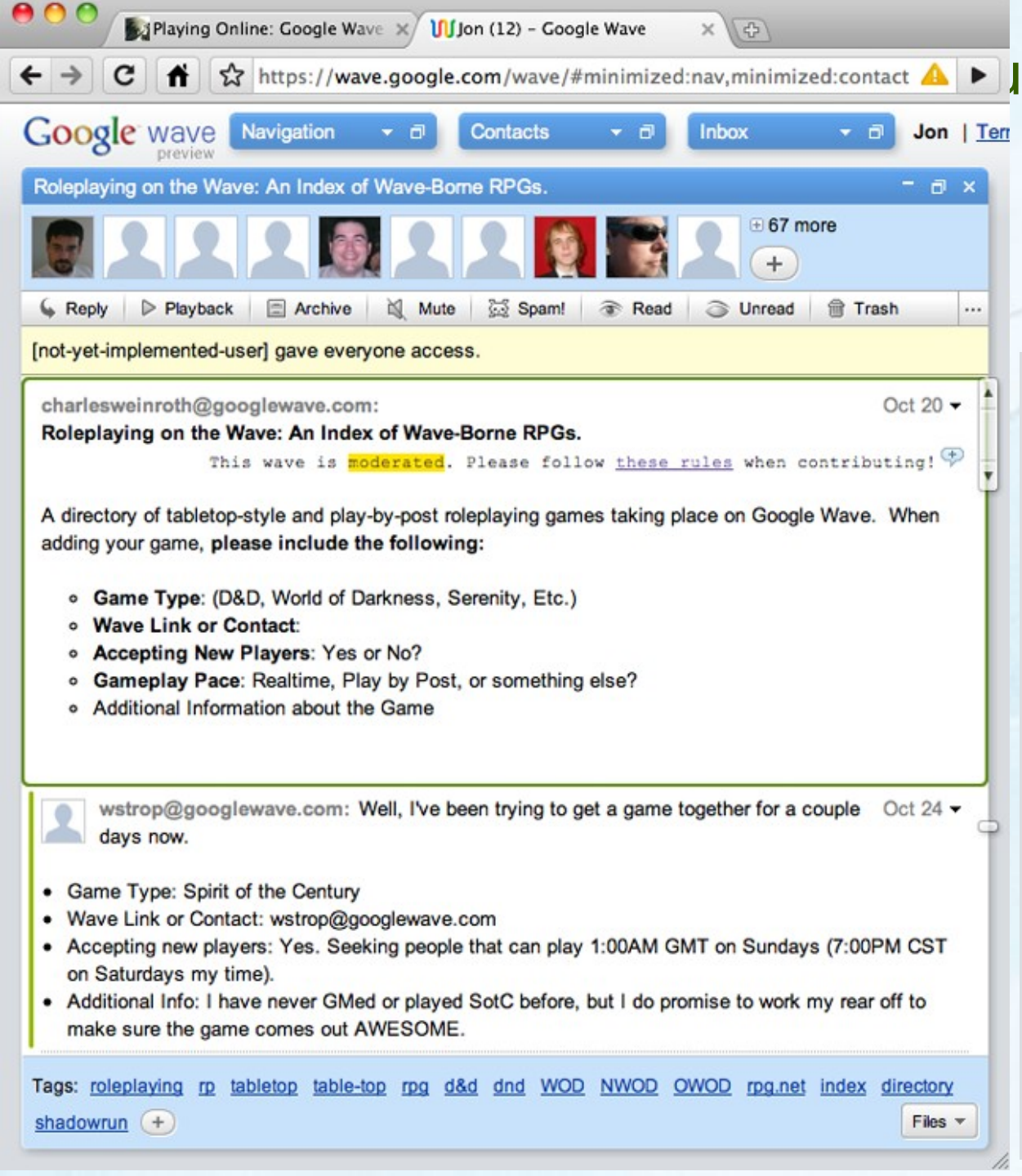
From the early days of the printing press to the dawn of
thing humans do upon inventing a new medium is distri
may hold true for most humans and most media, there
geek. From Nethack to play-by-post forums on the WW
upon inventing a new medium is play Dungeons and Dr
role-playing game itch is scratched.

Thus it was that when I finally got my Google Wave invit
least bit surprised to quickly discover a handful of Wave
progress, and many more in various stages of planning
from the sideline and talked to some Game Masters an
consensus that Google Wave has as much RPG potent
proverbial tabletop.

Getting oriented

If you're not familiar with the concept behind Wave, don't worry—it's hard to explain. You have to attempt to use it for something before you get any kind of handle on what it is, what it isn't, and what it probably should be. If I had to describe, the best I could do is this: Wave is a mash-up of email, IM,





ures

Ok, well then Korvald grips his staff
his eyes while whispering his prayer
tzenes@googlewave.com (and randomleetwe

focuses on this wounded beetle and brings his R
beetle.

1d20+5: 22 vs AC

Quinn Conklin: the brand flashes in to fi
beetel's back and it shreakes in pain try to
it's inctile glare. How much damage did you do?

so its **1d8+1: 9**, but I also give Eken
his next attack roll, so he'll get +9 in
tzenes@googlewave.com (and randomleetwe

battered fire beetle staggers back fro

cloud of burning vapor catching all three of you i

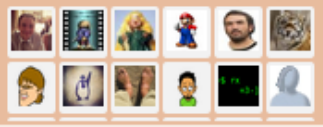
LI **1d20+4: 19** vs reflex
Ekemon **1d20+4: 19** vs reflex
Korvald **1d20+4: 14**

tzenes@googlewave.com: that's a

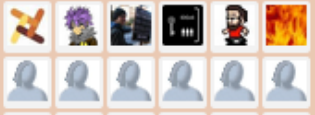
vs reflex

Group members

administrators (22)



collaborators (116)



Options

documents

- wiki
- events
- tasks
- lists
- chatrooms
- trash

Give us feedback!

kune

New doc | Add Gadget | Participate | Share | More

About kune

nov 02, 2013

*[**Castellano** debajo, ниже по **Русски**, παρακάτω για **Ελληνικά**]*
*Frequently, we see the scope of action of our **group/collective/organization** on Internet **limited** by several aspects, such as distance, **economic** means or **technical knowledge**, which make us heavily dependent on others. What would happen if we were able to overcome those problems ourselves, without any technical and economic effort?*
Kune allows the collaborative management of a collective with just an Internet connection.

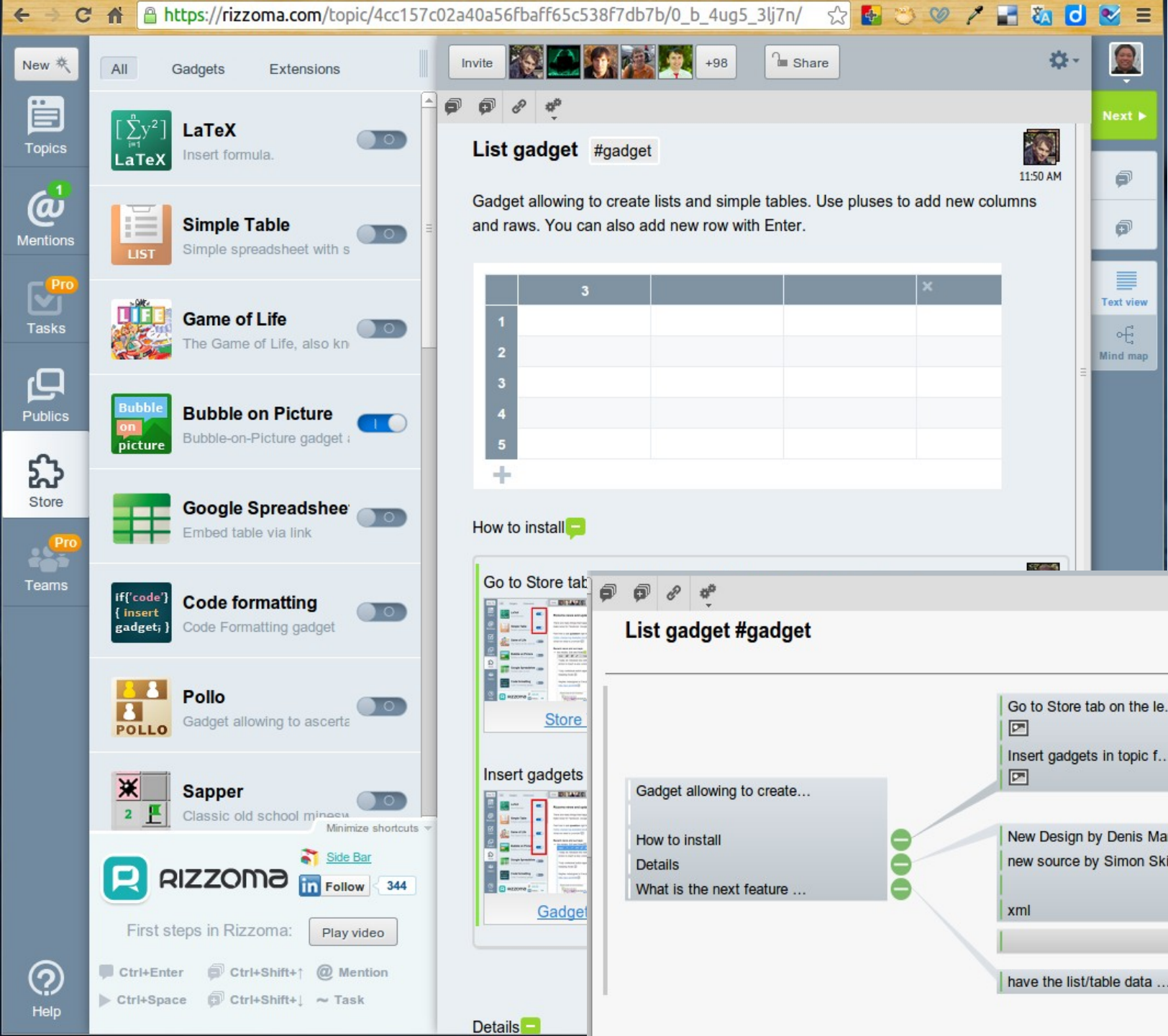
What would you be able to do with Kune?
You could create an online group space for the creation of collaborative documents, build your group website, plan and hold meetings in real time, interact or share your contents with other users of your group or others in your social network.

What is Kune, then?

- *Is it a communication tool? Yes, in several ways: chat (compatible with gmail and similar ones) and other more advanced ways to stay in touch...*
- *Is it a social network? Yes, although not centralized (as one big library) but distributed (as many interconnected libraries).*
- *Is it a real-time creation system for documents, like Google Drive? Yes.*
- *Is it an advanced new generation email system? Yes, showing your ongoing conversations and draft documents in an Inbox.*
- *Is it an online collaborative space for my collective/group? Yes, where you can share/build together.*
- *Is it a multimedia space? Yes, you can integrate and share videos, photos, maps, twitter...*
- *Is it free? Yes, fully free/libre open source software and Creative Commons contents.*
- *Could it help us to create our website? Yes (currently in development)...*
- *Is it extensible? Yes, the same way as Firefox is, with many add-ons to do many other things.*
- *Besides, it protects your privacy and encourages the sharing of knowledge and common goods.*

Without Dependencies

documents


https://rizzoma.com/topic/4cc157c02a40a56fbaff65c538f7db7b/0_b_4ug5_3lj7n/
New
Topics
Mentions
Tasks
Publics
Store
Teams
Help
All
Gadgets
Extensions
LaTeX
Insert formula.
Simple Table
Simple spreadsheet with s
Game of Life
The Game of Life, also kn
Bubble on Picture
Bubble-on-Picture gadget
Google Spreadshee
Embed table via link
Code formatting
Code Formatting gadget
Pollo
Gadget allowing to ascerta
Sapper
Classic old school mines
Minimize shortcuts
RIZZOMA
Side Bar
Follow
344
First steps in Rizzoma:
Play video
Ctrl+Enter
Ctrl+Shift+↑
Mention
Ctrl+Space
Ctrl+Shift+↓
Task
Invite
+98
Share
List gadget
#gadget
11:50 AM
Gadget allowing to create lists and simple tables. Use pluses to add new columns and raws. You can also add new row with Enter.
How to install
Go to Store tab
Store
Insert gadgets
Gadget
Details
Next
Text view
Mind map
List gadget #gadget
Gadget allowing to create…
How to install
Details
What is the next feature …
Go to Store tab on the le…
Insert gadgets in topic f…
New Design by Denis Mart
new source by Simon Skile…
xml
have the list/table data …
short
expanded

board.net/p/community

Collaborate seamlessly on documents! This pad text is synchronized as you type, so that everyone viewing this page sees the same text. No need to press a **save** button.
Create your own board and a (secret) name for it here: http://board.net . Pass it on to your colleagues with the share button.

board.net features

- simultaneous writing
- share the URL with others (nobody else knows it and no search engine crawls on it)
- timeslider - view any previous board status (be patient, may require reloading the board)
- bookmark versions
- get e-mail notifications when someone starts or finishes editing (currently off)
- invite others by e-mail (currently off)
- offline viewing (currently off)
- embed via iframe into your webpage
- optimized for printing (currently off)
- text chat on each board
- talk to another author via audio and video (enter your names top right, click on icon. Based on WebRTC, needs advanced browser like latest Firefox >22 or Chromium, no further plugin required, feature is in beta testing) (currently off)
- based on open source Etherpad-Lite technology, more features to come
- free and ads-free (we might ask for a donation in future)

FAQs
Place questions and exchange experience on using board.net here

Can I protect a board with a password?
Currently no, and honestly: you won't need it. Simply choose a (long) name of your board which can not be easily guessed. Pass it on to your colleagues. No search engine is able to find out the board names.

Editing does not work or I got an error message
Use a Firefox or Chromium browser, they are the most compatible, IE may not work at all, not all Safari browsers may work.
If the board "hangs" and you have no clue, simply wait until the next day. The board server is restarted every night at 6 a.m. CET to collect all "dead" boards.
If you happen to get a message "An error occured" you don't need to send it to us, we'll have it in the log files and will try to analyse it at each software update cycle.

GitHub, Inc. [US] https://github.com/ether/etherpad-lite/wiki/S

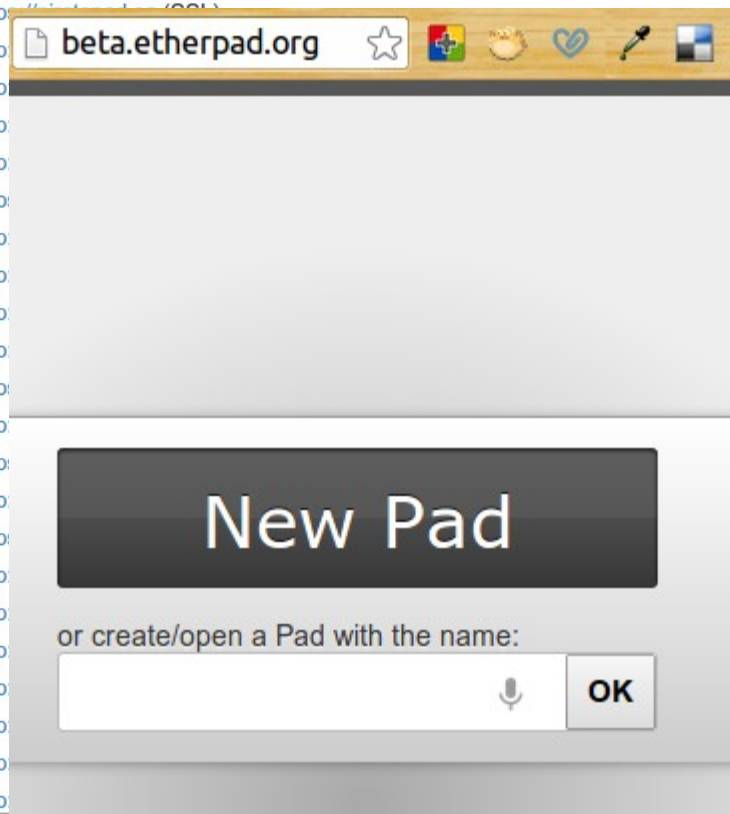
This repository Search or type a command Explore Gis

ether / **etherpad-lite**

Home Pages History

Sites That Run Etherpad Lite

- http://beta.etherpad.org (pads are destroyed after 30 days, used as a testing e
- http://notas.dados.gov.br
- http://board.net (plugins installed: speech input (works with Chromium browse checkboxes for tasklists, video and audio chat (meet on a board, enter name, http://board.net/p/community)
- http://notes.occupy.net/



Agenda

A. Service systems (science, management, engineering and design)	**1.** Domain scope?
B. Pattern language (c.f. pattern catalog)	**2.** Pattern form?
C. A starter set? •7 conditions from service systems science	**3.** Content (with repurposing)?
D. Collaboration •Inquiring system •Technologies	**4.** Dialectic with authentic subjectivity?
→ E. Next steps	**5.** Who and when?

