From Environmental Structure to Service Systems Thinking: Wholeness with Centers Described with a Generative Pattern Language

DAVID ING, INTERNATIONAL SOCIETY FOR THE SYSTEMS SCIENCES AND AALTO UNIVERSITY

In the early 1990s, pattern languages originating in architecture began the foundation for development of work in object-oriented design and methods. Since that time, the work of Christopher Alexander has continued to develop, with new emphases only hinted upon in the research of the 1990s. The SSMED (Service Science, Management, Engineering and Design) vision originating circa 2003 is now being more thoroughly founded on systems thinking, leading to a new perspective labeled as Service Systems Thinking. Advances in generative pattern languages and collaborative Internet technologies are proposed to enhance the development of an emerging pattern language for service systems.

Categories and Subject Descriptors: **H.1.1 [Models and Principles]**: Systems and Information Theory—*General Systems Theory*; **H.5.3 [Information Interfaces and Presentation]**: Group and Organization Interfaces—*Organization Design;* ***J.4 [Social and Behavioral Sciences]****: Economics, Psychology, Sociology;* ***K.4.3 [Computers and Society]****: Computer-supported cooperative work*

General Terms: Service systems, systems thinking, pattern language

Additional Key Words and Phrases: Design thinking,

ACM Reference Format:

Ing, D. 2014. From Environmental Structure to Service Systems Thinking: Wholeness with Centers Described with a Generative Pattern Language. ACM Trans. Pattern Languages of Programming. n, n, Article n (October 2014), nn pages.

1. INTRODUCTION: SERVICE SYSTEMS THINKING AIMS TO BUILD ON CHRISTOPHER ALEXANDER'S APPROACH AS A FOUNDATION

Service systems thinking is proffered as a label for an emerging body of work that: (i) builds on systems thinking extending social systems science (i.e. socio-psychological, socio-technical and socio-ecological systems perspectives) into service systems science; (ii) advances a transdisciplinary appreciation of service science, management, engineering and design (SSMED); (iii) explores the practices of architectural design in Christopher Alexander's work on generative pattern languages; and (iv) collaborates through a multiple perspectives inquiring system with the new federated wiki platform. This endeavour is seen as a community activity that could take ten years to mature.

This article aspires to engage the pattern language community not only to repurpose the broad range of pattern catalogs already developed across the broad range of domains, but also to more deeply appreciate Christopher Alexander's clearer articulation of generative pattern languages in his later writings.

In brief, service systems thinking can be described both with an intentional representation and an object-process representation.

Author's address: D. Ing, 151 Booth Avenue, Toronto, Ontario, Canada M4M 2M5; email: coevolving@gmail.com

In an *intentional representation*, service systems thinking is a resource that can be applied by service scientists, managers, engineers and designers.

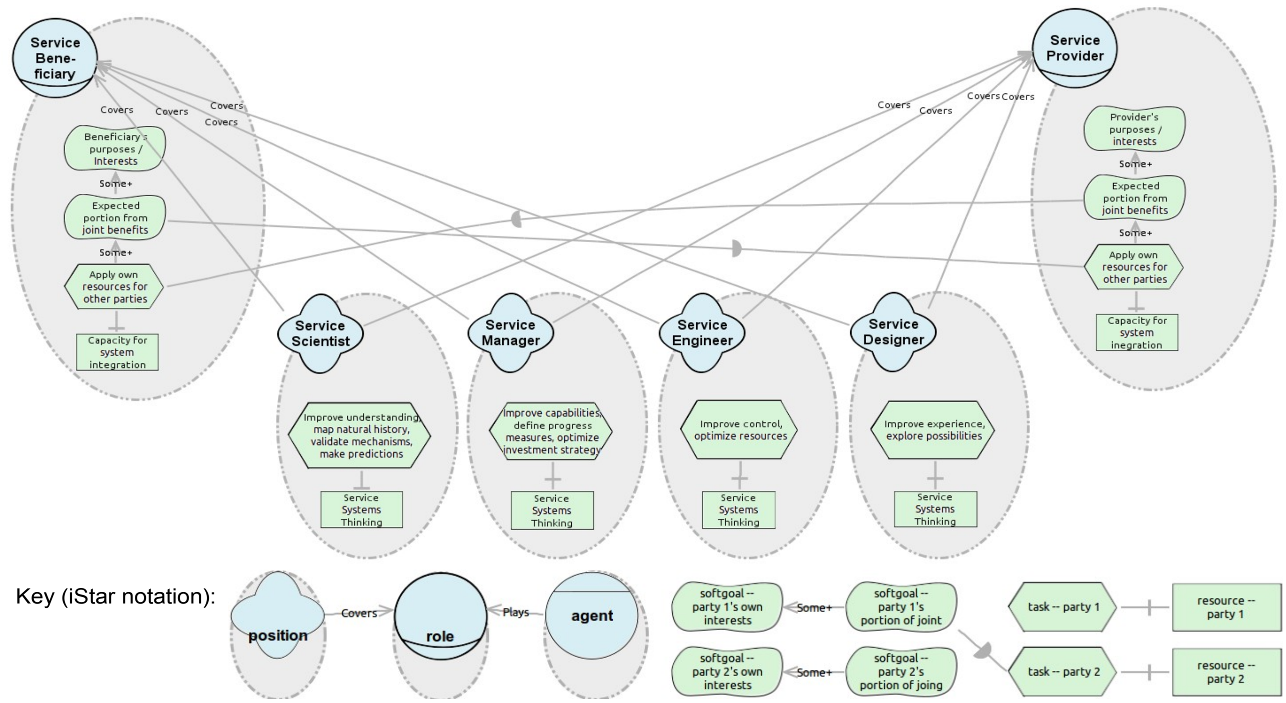

Illustration 1: Service Systems Thinking -- An intentional perspective

Illustration 1 depicts a service system with two roles: a beneficiary and a provider, using an i* (pronounced eye-Star) notation (Horkoff and Yu 2006). Each role has its own softgoals of purposes and interests. The expected portion of joint benefits from the relationship depends on the combination of resources (as hardgoals) that are applied by the other parties and itself. Among the resources at hand for each role is the capacity for system integration

Each of the service beneficiary and service provider roles may be covered by a position. A service scientist position has hardgoals to improve understanding, map natural history, validate mechanisms and make predictions; a service manager position has hardgoals to improve capabilities, define progress measures and optimize investment strategy; a service engineer position has hardgoals to improve control and optimize resources; a service designer position has hardgoals to improve experience and explore possibilities (Spohrer and Kwan 2009).

Service systems thinking could be a resource that supports the hardgoals for all of these positions, as a cross-disciplinary platform for communicating.

In an *object-process representation*, service systems thinking (as a process) is related to a service systems thinking community (as an object). Illustration 2 depicts that service systems thinking is handled by the service systems thinking community, using OPM notation (Dori 2006). Service systems thinking exhibits systems thinking (a process), SSMED (an object), generative pattern language (an object) and multiple perspectives open collaboration (a process).

The services systems thinking community handles four processes: conversations for orientation, conversations for possibilities, conversations for action, and conversations for clarification (Winograd 1986).

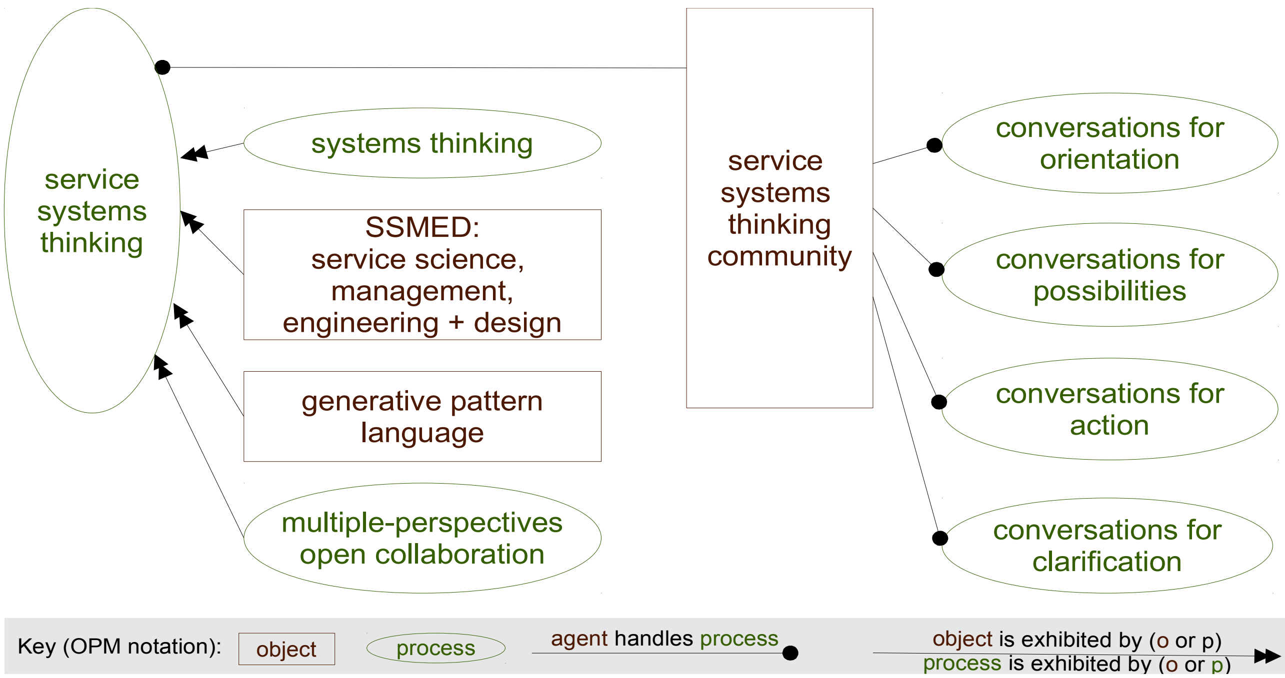

Illustration 2: Service Systems Thinking: An object-process representation

The service systems thinking community is still in a formative phase. This article provides orientations on SSMED and generative pattern language. Content on advances in systems thinking in the 21st century can be covered through alternative orientations (Ing 2013). Multiple perspectives open collaboration has been implemented in a new federated wiki technology where orientations are better presented through web video and hands-on learning (Cunningham 2012a).

Section 2 of this article describes key features in the science of services systems that may reframe the approach to a generative pattern language. Section 3 traces the development of ideas by Christopher Alexander over 50 years, and highlights writings where his worldview is clarified.

Section 4 explores possibilities for service systems thinking, as questions in which alternative paths forward warrant collaboration. This article concludes in Section 5, recounting the activities which have taken place to date.

2. ORIENTATION: DISTINCT FEATURES IN SERVICE SYSTEMS INCLUDE COPRODUCTION, OFFERINGS, VALUE AND RESOURCES

The centrality of services in human activity was recognized in the 20th century with *service management* (Normann 1984), but the call for a science of service systems did not come until the 21st century. This idea was introduced to the service science community in 2005 (Spohrer 2005).

> Over the past three decades, services have become the largest part of most industrialized nations' economies. Yet there's still no widely accepted definition of service, and service productivity, quality, compliance, and innovation all remain hard to measure. Few

researchers have studied service, and institutions have paid little attention to educating students in this area (Spohrer et al. 2007).

In a concise orientation to some key features in service systems, the content for appreciating the domain is described in section 2.1. Coproduction is outlined in section 2.2; offerings are defined in section 2.3; inquiry into value in service science is described in section 2.4; resources are analyzed as operand and operant in section 2.5; and actors and intentions in service systems are introduced in section 2.6. In section 2.7, the progress on a science of service systems is compared to the development of computer science from its origins.

2.1 Service systems dominate human activity in development countries

Our everyday lives have service systems omnipresent in technical, organizational and socio-political forms. We are immersed in service systems, so developing a greater appreciation just requires drawing attention to them. A proposed curriculum for primary and secondary schoolchildren illustrates how much of civilization we take for granted.

- Systems that *move, store, harvest and process* include transportation; water and waste management; food and global supply chains; energy and energy grids; and information and communication technology (ICT) infrastructure.
- Systems that *enable healthy, wealthy and wise people* include building and construction; banking and finance; retail and hospitality; healthcare; and education (including universities).
- Systems that *govern* include cities; regions and states; and nations (Spohrer and Maglio 2010).

Table 1: Types of service systems (adapted from Spohrer and Maglio 2010)

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

The order of these service systems ranges roughly from the more concrete to the more abstract. Kindergarten children could learn about transportation systems as they travel from home to school. Grade 1 students could visit a water treatment plant. By Grade 2, students could learn how food reaches their dinner tables. The most abstract service systems are provided by governments, better explored in later high school.

While defining “service” has been approached by a wide variety of perspectives, describing a “service system” compatible with a systems thinking worldview is rarer. A publication oriented towards innovation for education, research, business and government by the University of Cambridge proposed a concise wording:

> 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).

In the $54 trillion system of systems in our world, improvement is seen as a $4 billion challenge (IBM 2010). This challenge could be taken up by a variety of disciplinary professions. Service scientists could aim to improve that basic understanding of service systems, mapping their natural history, and validating mechanisms so that better predictions could be produced. Service managers might then have a better foundation on which to improve capabilities, define progress measures, and optimize investment strategies. Service engineers would have an applied science in which they could improve control and optimize resources. Service designers might take a lead in improving service experiences, and exploring the possibilities for better value propositions and government mechanisms (Spohrer and Kwan 2009). Service systems thinking could serve as a crosswalk to bridge disciplinary mindsets and language for more effective collaboration.

2.2 Service providers help customers create value for themselves, as coproducers

A service system, by definition, has multiple parties in interaction. Mechanistic conceptions of systems as producer-product, e.g. economic depictions of value chains, or engineering depictions of supply chains, tend to emphasize parts as independent with low-intensity interactions as handoffs. Interactive concepts of systems see parts (in nature) or roles (in human interactions) as coproducers. Coproduction is expressed as "*the* most critical concept" in purposeful systems (Ackoff and Emery 1972, 23). Richard Normann grounded much his work in systems theory.

> What is new is not co-production, but the way it now expresses itself in terms of role patterns and modes of interactivity. The characteristics of today's economy naturally reshape co-productive roles and patterns. The distinction between "producer" and "consumer", or "provider" and "customer" is ever less clear as the business landscape takes more of a "service" mode (Normann 2001, 96).

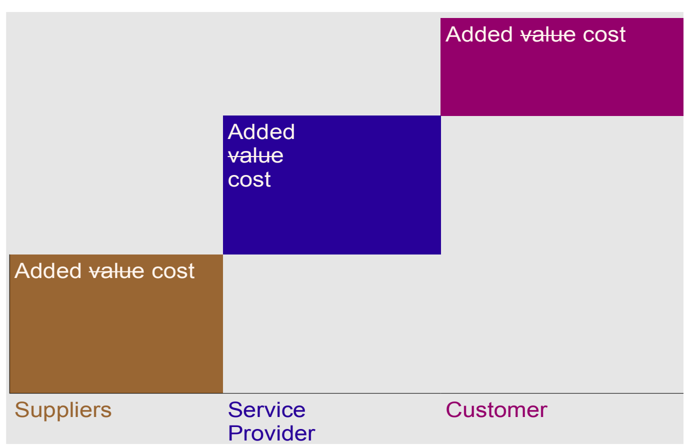

Illustration 3: Not added value; added cost

A production system can produce with only a producer. A service system presumes at least two parties, and may serve not only the customer who consummates the transaction, but potentially also additional downstream beneficiaries and upstream suppliers. Rather than analytically focusing on bilateral relations, a *value constellation* approach draws a more inclusive boundary around a larger set of involved parties.

With multiple interactions between parties taking place within a value constellation, the idea of a "value chain" with "added value" at each stage shown in Illustration 3 is dissolved into a representation of added costs accumulated sequentially in interactions.

> Our traditional about value is grounded in the assumptions and the models of an industrial economy. According to this view, every company occupies a position on the value chain. Upstream, suppliers provide inputs. The company then adds values to these inputs, before passing them downstream to then next actor in the chain [whether another business or the final consumer] (Normann and Ramirez 1993, 65).

This “assembly line” mindset is more appropriate in a world where demand exceeds supply, so that production lines are optimized for greatest efficiency, and the variety available to customers is low. In a world where supply exceeds demands, the interactions between parties can have higher variety.

Let's flesh out the Ikea example that is commonly presented as an example. A mechanistic value chain perspective “follows the money” with the provider signatory (e.g. Ikea) providing an output, and the customer signatory (e.g. the father of a family as purchaser) paying an additional profit for acquisition.

Alternatively, in an interactive value constellation perspective depicted in Illustration 4, let's recognize four parties: (i) the suppliers (e.g. foresters, furniture makers); (ii) the provider signatory (e.g. Ikea, as the prime mover orchestrating the design, manufacturing and distribution); (iii) the customer signatory (e.g. the father who foots the bill for the purchase); and (iv) the beneficiary stakeholders (e.g. other family members in the home who enjoy the furniture). All four parties can be seen as coproducers in the service system. The interactive value of primary interest should be value in use, i.e. by family members enjoying the furnishings for many years after the father has executed on the transaction of purchase. That interactive value is a distinct from the profits that the provider signatory (e.g. Ikea) gains.

> IKEA is able to keep costs and prices down because it has systematically redefined the roles, relationships and organizational practices of the furniture business. [….]
>
> IKEA wants its customers to understand that their role is not to consume value, but to create it. […] IKEA's goal is not to relieve customers of doing certain things but to mobilize them to do easily certain things they have never done before. Put another way, IKEA invents value by enabling customers' own value-creating activities. … Wealth is [the ability] to realize your own ideas (Normann and Ramirez 1993, 66–67).

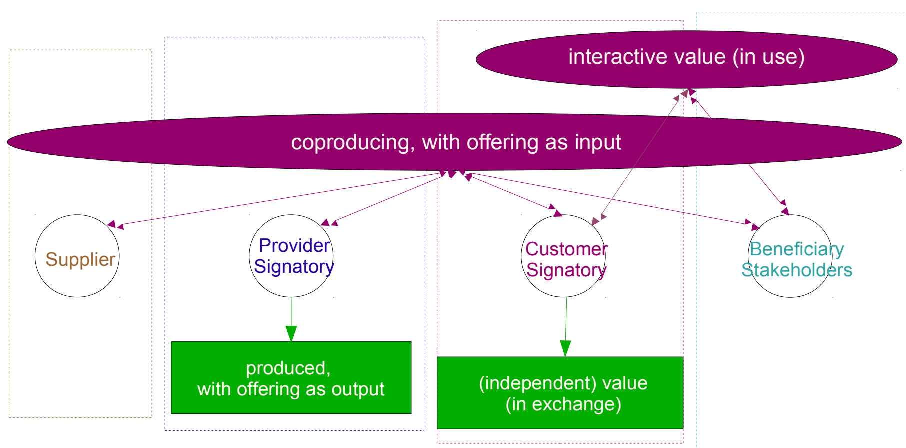

Illustration 4: Enabling interactive value creation

In the illustration, interactive value is depicted as a process where enjoyment takes place over a period of time, as compared to the value in exchange that occurs at only a point in time. In the larger service system, independent transactions are deemphasized relative to the ongoing relationship in the context of mutually changing environments

> From [the] value constellation perspective, value is co-produced by actors who interface with each other. They allocate the tasks involved in value creation among themselves and to others, in time and space, explicitly or implicitly. This opens up many opportunities for defining relationships between actors and reassigning activities. If we look at a single relationship in a co-productive system (for example, that between customer and supplier) this view implies that the customer is not only a passive orderer / buyer / user of the offering, but also participates in many other ways of consuming it, for instance in its delivery. Etymologically, consumption means value creation, not value destruction; this sense of consumption is inherent in the "value constellation" point of view. Furthermore, as actors participate in ways that vary from one offering to the next, and from one customer / supplier relationship to the next, it is not possible to take given characteristics for granted: co-producers constantly reassess each other, and reallocate tasks according to their new values of the comparative advantage each other to have (Normann and Ramirez 1994, 54).

With foundations in systems theory, coproduction is a concept that can be appreciated across the disciplines of science, management, engineering and design, as a common foundation for service systems thinking.

2.3 Offerings are three-dimensional packages either as outputs to, or inputs for, customers

The rise of research into services has led to some confusion of that term. In definitions that emphasize activities or processes with ties between service provision and economic exchange, an implication could be that "everything is a service" (Vargo and Lusch 2004b). This is an unfortunate semantic overloading.

In a clearer definition of a service system, the label of offering is introduced to describe a delivery package in three dimensions, as shown in Illustration 5: physical product content, service and infrastructure content, and interpersonal relationship (people) content. Since any offering coproduced by a value constellation – that could include subcontract, supplier, customer and beneficiary roles – involves contributions by each of the parties, the shape of the delivery package could be different in every interaction.

> … it is useful to examine the offering in terms of a three-dimensional activity package [Illustration 3]. The three axes are *hardware* (or the 'physical product content' of the offering), *software* (the 'service and infrastructure content'), and '*peopleware*' (the interpersonal relationship or 'people content').
>
> - The physical content of the offering consists of elements such as the core product, the packaging, the quality and dependability of the good and its material components, the product range, etc.

- The service content includes distribution, technical support, product modifications, customer training, on-line advice, troubleshooting, warranties and other trust-supporting insurance aspects, information brochures, brand reputation, complaint handling, invoicing, integrated information systems, etc.

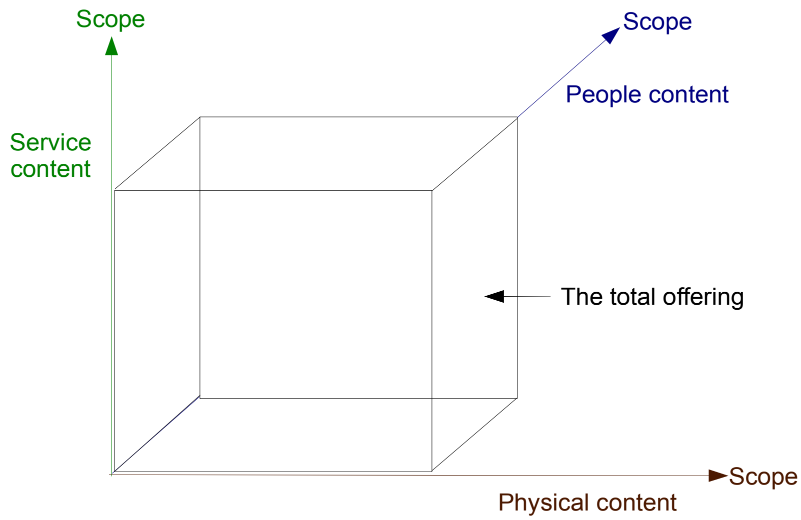

Illustration 5: The three-dimensional offering (Ramirez and Wallin 2000)

- The people content covers issues like long-term partnerships, interpersonal trust, reputation, human resource co-development, etc.

> In keeping with Levitt's view that a product only has meaning from the viewpoint of the customer, different customers will emphasize different axes of the offering.
>
> In co-production terms, the value-creating potential along each of the dimensions of the offering – physical, service or people content – depends on the value-creating system of the customer (Ramirez and Wallin 2000, 58–59).

In this definition of a service system, there are non-service parts to the offering. The way that the customer uses the offering frames its value.

> Offerings are the *output* produced by one (or several) actor(s) creating value -- the 'producer' or 'supplier' -- that becomes an *input* to another actor (or actors) creating value - the 'customer' (Ramirez and Wallin 2000, 47).

Some customers are interested in engaging with a provider for an offering more as an output that requires little or no additional processing, while others want the offering more as an input to be processed with other inputs towards a result with greater value. Customer value can either be derived through transactions or through relationship. The cross of those two dimension leads to the matrix in Illustration 6 (Ramirez and Wallin 2000, 141–145).

- In an *industrial logic* (e.g. 1920s automobile mass production), production cost reductions enable the offering as an output to create value was primarily through an more affordable transaction.

- In a *service logic* (e.g. branded automobiles with models following the customer's age), ensuring continuing customer satisfaction enables an offering as an output to create value primarily through relationship.
- In a *self-service logic* (e.g. do-it-yourself packages), independence and convenience maximization enables an offering as an input to create value through an affordable transaction.
- In a partnership logic (e.g. anticipatory personalization capabilities), value co-development enables an offering as input to create value through an enduring relationship.

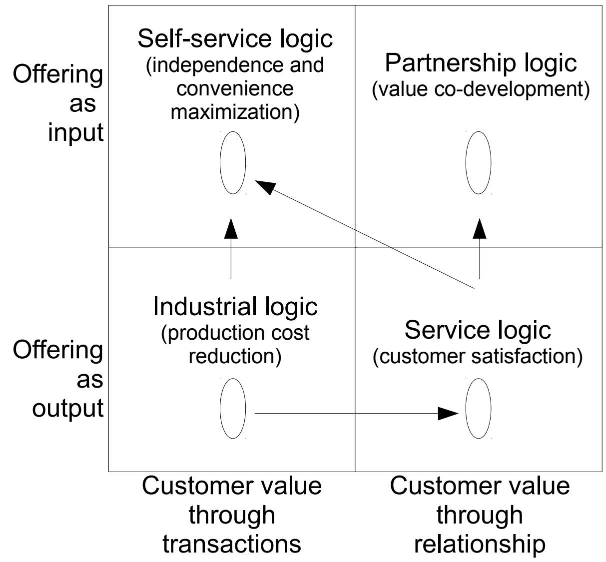

Illustration 6: Alternative views on how offerings and customer relationships interact

The party who designs the offering may be described as the orchestrator or prime mover or the service system capabilities. With an offering as an output, the orchestrator is generally the provider. With an offering as an input, any of the coproducers may rise into a role as orchestrator.

2.4 Value is appreciated interactively by each party in exchange, in use, and in context

Reviewing the academic literature on value, six themes of understanding can be appreciated and mapped into an integrative value framework (Ng and Smith 2012).

From *philosophical* foundations dating back to Plato (360 B.C.E.), value was described as intrinsic (i.e. good to have for itself) and/or extrinsic (i.e. good to have as instrumental to achieve or obtain something else that is good. By 1927, Heidegger proposed an existential philosophy where individuals give meaning to existence in terms of their actions or projects. In 1939, Husserl proposed a phenomenological concept of object conceived in the experience of it. Through Giddens (1979), Chandler and Vargo (2011) argue that individuals and their contexts are mutually constitutive, whereby a context could be simultaneously be a resource for one actor and a deterrent for another actor. All of these views can be labeled as "use-value".

From *economic* foundations with Adam Smith in 2014, "value in exchange" (i.e. as the power to purchase other goods) was presented as distinct from "value in use" (i.e. as the utility of a particular object. Endowed with invariant properties of goodness and contexts presumed to similarly perceived by all, homogeneity led to a goods-centric focus where products were manufactured in seek of target markets who would perceive value. The experience of use-value after the purchase informing future transactions led to the discipline of marketing.

From *management* foundations, the "selling value" of products circa 1957 evolved by marketers to become exchange value that was superior to competitors. Two firm-centric approaches emerged as (i) the *economic worth of the customer* (EW) in lifetime purchases; and (ii) the *perceived satisfaction of the firm's offerings* (PS) in a stream of repeat purchases. Two preferential judgements of the customer were expressed as (iii) *net benefit* (NB), i.e., the evaluation of outcomes as net difference between the benefits and costs associated with acquiring and consuming an offering, and (iv) *means-end* (ME), i.e.

the evaluation of attributes offering as means towards a goal in the customer's use situations. Evaluating v*alue at the point of choice* can be different from the evaluation at the point of use.

The *modern conceptualization* led by Holbrook (1994) sees value as residing not in an object, a product or possession, but as an “interactive, relativistic preference experience”, where the customer is an active participant in its creation. This view was extended in Service-Dominant Logic (Vargo and Lusch 2004a; Vargo and Lusch 2008), with a recapturing of value-in-use. Thus, firms cannot provide value, but only offer propositions of value, with the customer determining the value and the cocreation with the company at a given time and context. Customers are always co-creators of value-in-use contexts, but my not always be co-producers of a firm's offerings.

As a new contribution to service science, P-C-value and A-C-value are presented as a reconciliation and an integration of the preceding conceptualizations. The value being created may sit in different levels of consciousness at different times.

> Block (1977) describes consciousness as being of two types – phenomenal consciousness (P-consciousness) and access consciousness (A-consciousness). P-consciousness is the raw experience of movement, forms, sounds, sensations, emotions and feelings, while A-consciousness is perception, introspection, reflection, in a sense, a more heightened awareness of a phenomenon. This suggests that if we understand value creation as creating something ‘good’ as an outcome, the consciousness of that goodness during the phenomenological experience may be different from the consciousness of that goodness imagined before, or evaluated after, the phenomenon. One can even argue that within the phenomenon, the actor is merely ‘in practice’ of resource integrating, with a lower level consciousness of what is ‘good’, or what is of ‘value’, from the resources being integrated within the value-creating phenomenon. In other words, even if value is uniquely created within a phenomenon, there could possibly be two levels of consciousness of that value that could exist at different times: P-consciousness of value (P-C-value) or A-consciousness of value (A-C-value) (Ng and Smith 2012, 227–228).

This integration sees that value is not necessarily static, but dynamic according to time (i.e. before, during and/or after the experience).

P-C-value is the creation of value in context that is phenomenal, integrating (i) the existence of the offering, (ii) the affordance of the offering; (iii) the context of the offering in use situations, (iv) agency as the capacity of an actor or entity to act in the world;

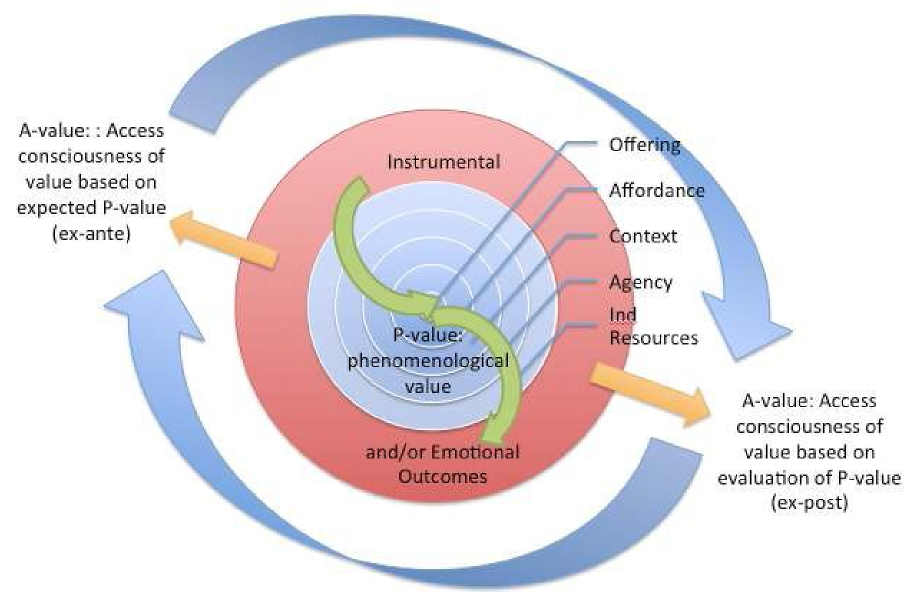

Illustration 7: The Integrated Value Framework (from Ng and Smith 2012)

and (v) actor resources of skills and competencies required to create the P-value of the offering in context.

A-C-value is argued as the perception of goodness that drives choice ex ante and valuation ex post. It is an awareness of goodness at the point of exchange.

The degree of A-C-value ex ante may be related to the P-C-value, which is related to the A-C-value ex post. These relationships have been left for future research.

2.5 While resources were previously considered only operand, service science sees operant resources

The contemporary view on service systems is that they operate in a world where resource not only include "natural resources" that are tangible, but also human ingenuity that is not tangible. This is marked by a shift from Goods Dominant (G-D) Logic to Service Dominant (S-D) Logic.

> In his analysis of world resources, Thomas Malthus (1798) concluded that with continued geometric population growth, society would soon run out of resources. In a Malthusian world, "resources" means natural resources that humans draw on for support. Resources are essentially "stuff" that is static and to be captured for advantage. In Malthus's time, much of the political and economic activity involved individual people, organizations, and nations working toward and struggling and fighting over acquiring this stuff. [….] As we discuss, this change in perspective on resources helps provide a framework for viewing the new dominant logic of marketing.
>
> Constantin and Lusch (1994) define *operand resources* as resources on which an operation or act is performed to produce an effect, and they compare operand resources with operant resources, which are employed to act on operand resources (and other operant recourses). During most of civilization, human activity has been concerned largely with acting on the land, animal life, plant life, minerals, and other natural resources. Because these resources are finite, nations, clans, tribes, or other groups that possessed natural resources were considered wealthy. A goods-centered dominant logic developed in which the operand resources were considered primary. A firm (or nation) had factors of production (largely operand resources) and a technology (an operant resource), which had value to the extent that the firm could convert its operand resources into outputs at a low cost. Customers, like resources, became something to be captured or acted on, as English vocabulary would eventually suggest; we "segment" the market, "penetrate" the market, and "promote to" the market all in hope of attracting customers. Share of operand resources and share of (an operand) market was the key to success.
>
> Operant resources are resources that produce effects (Constantin and Lusch 1994). The relative role of operant resources began to shift in the late twentieth century as humans began to realize that skills and knowledge were the most important types of resources. [….]
>
> Operant resources are often invisible and intangible; often they are core competences or organizational processes. They are likely to be dynamic and infinite and not static and finite, as is usually the case with operand resources. Because operant resources produce effects, they enable humans both to multiply the value of natural resources and to create additional operant resources. A well-known illustration of operant resources is the microprocessor: Human ingenuity and skills took one of the most plentiful natural resources on Earth (silica) and embedded it with knowledge. [….] The service-centered

dominant logic perceives operant resources as primary, because they are the producers of effects. This shift in the primacy of resources has implications for how exchange processes, markets, and customers are perceived and approached (Vargo and Lusch 2004a, 2–3).

This rethinking about focus on resources changes the perspective on how service systems should be considered.

S-D logic implies that "producing" should be transformed into "resourcing." Resourcing allows value creation through collaborative value cocreation, not only involving the provider and the beneficiary but all parties in a value-creation network. Goods remain important in S-D logic, but they are seen as vehicles for resource transmission (what some call appliances or tools), rather than containers of value. [….]

This resourcing conceptualization of service connects well with the concept of service systems as market-facing complex systems [….]

Conceptual Foundations for Service Science

S-D logic, with its process and resourcing orientation, offers a perspective for a conceptual foundation of service science, management, and engineering (SSME), as illustrated in [Table 2]. A critical element of S-D logic involves rethinking the meaning and role of resources. The key distinction is between operand and operant resources (Lusch, Vargo, and Wessels 2008, 7).

Table 2: G-D logic versus S-D logic: A change in perspective (Lusch, Vargo, Wessels 2008)

From G-D Logic	**To S-D Logic**
Operand resources	Operant resources
Resource acquisition	Resourcing (creating and integrating resources and removing resistances)
Goods and services	Servicing and experiencing
Price	Value proposing
Promotion	Dialog
Supply chain	Value-creation network
Maximizing behavior	Learning via exchange
"Marketing to"	Collaborative marketing ("marketing with")

The surfacing of S-D logic perspective, originally developed by Vargo and Lusch, has led to many practitioners reflecting on their preconceptions based on G-D logic, as well as a series of refinements by service researchers (Lusch and Vargo 2006; Vargo and Lusch 2008). For the purposes of service systems thinking, compatibility of S-D logic with systems theory was not as high as with the original concept of offerings by Normann and Ramirez, but academic inquiry continues to work out details.

2.6 Including actors and intentions in service systems models can complement objects and processes

When the word "systems" gets appended to "services", many are predisposed to think about processes. However, services also involve social relationships, where parties coordinate to provide outcomes.

Recent research into service systems has proposed that service system entities – people, organizations and/or partnerships – be represented as intentional agents, to account for intentional and strategic dimensions.

Our notion of intentional agent is drawn from agent-oriented modeling, where agents are viewed as social entities that depend on one another to reach their goals; they thus intentionally enter in relationships with one another to improve their well-being (Yu, 2009).

> *i** (short for distributed intentionality) is an agent-oriented modeling approach that has been developed to support the analysis and design of sociotechnical systems where multiple actors create networks of interdependencies; *i** enables the representation of such a system, as well as the evaluation of different alternatives that could best satisfy actors' goals (Yu, 2002). The use of *i** enables us to represent and analyze service systems at different levels of granularity. It also enables us to design and analyze service system interactions in terms of each entity's motivations. This can complement current process-based design approaches ..., whose focus on sequence of activities and information flows can help to understand *how* value is cocreated in time but do not account for *why* it is so (Lessard and Yu 2013, 69).

The i* modelling framework has been used in requirements engineering, business process design, organization modelling, software development methodologies and evolution. With the Seventh International i* Workshop being held in 2014, the body of knowledge and community has become well-developed. The basic i* notation represents actors and their associations, elements (of resources, tasks, (hard)goals, softgoals and beliefs); and links of dependencies (e.g. strategic, goal, task, resource) (Horkoff and Yu 2006).

> We focus here on mechanisms that emphasize the intentional dimension of service engagements in this domain. Core to such engagements are the benefits that each participating entity expects to gain, in exchange for which it is willing to offer something of value to another entity. Since the other entity will only accept the value proposition if it is beneficial from its own perspective, service system interactions are established in the context of perceived mutual benefits (Vargo, 2009). We have also observed that entities come into relationships with high-level interests, to which the specific benefit that can be obtained from a service engagement contributes. [….] The benefit(s) expected by each entity may then become realized values if the results of the service engagement are evaluated positively, but different determinations of value by each system can lead one system to experience higher value than other systems. At any level of granularity, a service system can thus be understood in terms of the following concepts:
>
> - *High-level interests.* General interests or objectives pursued by a service system.
> - *Expected benefits.* Specific benefits that a service system expects to gain from its *collaboration with another service system.*

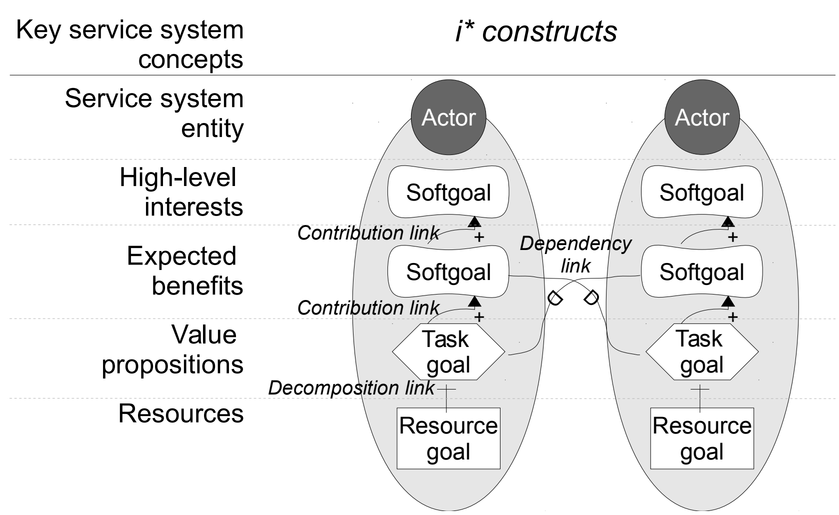

Illustration 8: Express of key concepts of value cocreation through i* modeling constructs (Lessard and Yu 2012)

- *Value proposition.* A service system's proposition to apply its knowledge, skills, and other required resources to produce something of potential benefit to another service system (Lusch et al., 2008).
- *Resources.* Operant and operand resources that can be integrated by a service system to form a value proposition (Vargo & Lusch, 2008).
- *Perceived value*. Positively evaluated outputs and outcomes of a service engagement.

> This understanding of service system value cocreation is in line with current literature but emphasizes a key dimension that has not received attention up to now: the *intentionality* of service systems. Indeed, service system entities are not only composed of resources but also of interests, desires, and needs (Lessard and Yu 2013, 71).

This intentional view represented through i* can complement more traditional modeling of entities and processes. The modelling of software systems conventionally uses UML; the modeling of hardware systems has moved towards SysML.

For conceptual modeling, a simpler alternative may be found in approach consistent with the basic concepts in systems thinking: Object-Process Methodology. OPM takes a strong stance on the fundamentals of systems.

> **Function, Structure, and Behavior: The Three Major System Aspects**
>
> All systems are characterized by three major aspects: function, structure, and behavior. The **function** of an artificial system is its value-providing process, as perceived by the beneficiary, i.e., the person or group of people who gain value from using the system. For example, the function of the organization called hospital is patients' *health level improving*. Each patient is a beneficiary of this system, the customer may be a government or a private entity, and the medical staff constitutes the group of users.
>
> Function, structure, and behavior are the three main aspects that systems exhibit. Function is the top-level utility that the system provides its beneficiaries who use it or are affected by it, either directly or indirectly. The system's function is enabled by its architecture -- the combination of structure and behavior. The system's architecture is what enables it to function so as to benefit its users.
>
> Most interesting, useful, and challenging systems are those in which structure and behavior are highly intertwined and hard to separate. For example, in a manufacturing system, the manufacturing process cannot be contemplated in isolation from its inputs -- the raw materials, the model, machines, and operators -- and its output -- the resulting product. The inputs and the output are objects, some of which are transformed by the manufacturing process, while others just enable it. Due to the intimate relation between structure and behavior, it only makes sense to model them concurrently rather than try to construct separate models for structure and behavior, which is the common practice of current modeling languages like UML and SysML. The observation that there is great benefit in concurrently modeling the systems structure and behavior in a single model is a major principle of OPM.

> **Structure** of a system is its form -- the assembly of its physical and logical components along with the persistent, long-lasting relations among them. Structure is the static, time-independent aspect of the system. The **behavior** of a system is its varying, time-dependent aspect, its dynamics -- the way the system changes over time by transforming objects. In this context, transforming means creating (generating, yielding) a new object, consuming (destructing, eliminating) an existing object, or changing the state of an existing object.
>
> With the understanding of what structure and behavior are, we can define a system's architecture.
>
>> **Architecture** of a system is the combination of the system's structure and behavior which enables it to perform its **function**.
>
> Following this definition, it becomes clear why codesign of the system's structure and behavior is imperative: they go hand in hand, as a certain structure provides for a corresponding set of system behaviors, and this, in turn, is what enables the system to function and provide value. Therefore, any attempt to separate the design of a system, and hence its conceptual modeling, into distinct structure and behavior models is bound to hamper the effort to get close to an optimal design. One cannot design the system to behave in a certain way and execute its anticipated function unless the ensemble of its interacting parts of the system -- its structure -- is such that the expected behavior is made possible and deliver the desired value to the beneficiary (Dori 2011, 216–217).

The entities in OPM include two things, (i) *objects* and (ii) *processes*, which are modeled as first class citizens in an *object-process equality principle*. The third entity in OPM is a *state*, defined as a situation in which an object can be at some point in time. Links are used to connect the three entities in Object Process Diagrams.

In formal definitions:

> An **object** is a thing that exists or can exist physically or informatically (Dori 2011, 223).

This is a structural, timeless view of the world at moment of time. This definition is more general that that normally used for object-oriented development of information systems.

For the temporal perspective, a definition of *transformation* is invoked so that time-dependent relationships amongst things are representable.

> **Transformation** is the generation (construction, creation) or consumption (destruction, elimination) or change (effect, state transition) of an object (Dori 2011, 224).

The existence of an object could be changed through a transformation, or some of its attributes could be changed over time. Thus,

> A **process** is a transformation that an object undergoes (Dori 2011, 225).

This definition of a process requires the existence of at least one object. An object can have states; a process can have subprocesses.

In the English language, a noun can sometimes mean either an object or a process. While the default is to assume a noun is an object, the object-process distinction says to classify a given noun as a process if an only four process criteria are met: (i) object involvement; (ii) object transformation; (iii) association with time; and (iv) association with verb (Dori 2011, 227).

OPM employs both graphical and text to reduce the cognitive load of interpreting a model. Software tools can map from the graphical Object-Process Diagram (OPD) to the textual Object-Process Language (OPL). Illustration 9 show an example constructed in the Opcat tool.

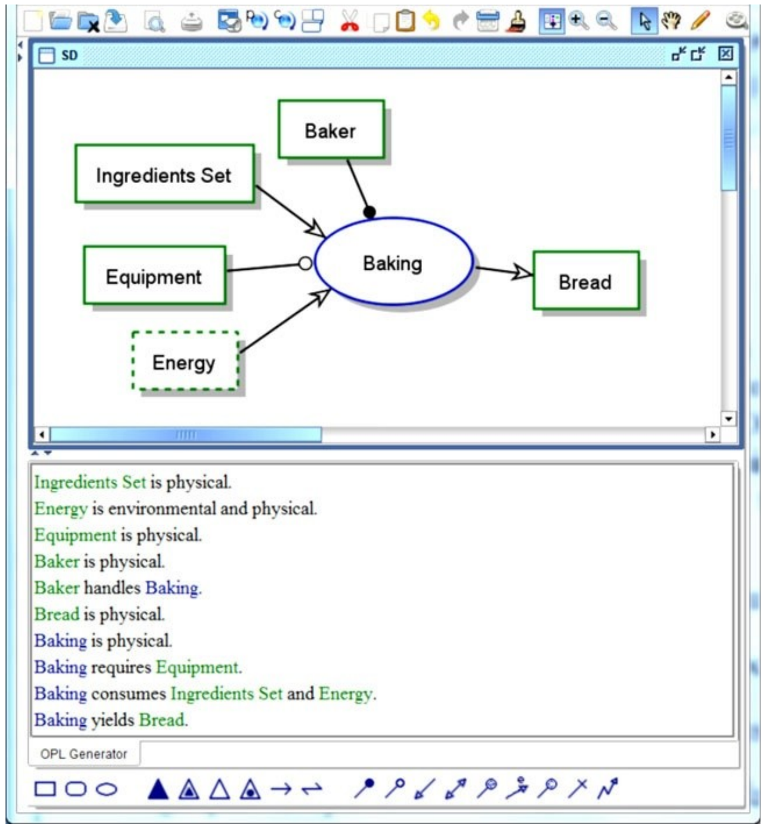

Illustration 9: A baking system, with the Object-Process Diagram (OPD) above and Object-Process Language (OPL) below (from Dori 2011, 212)

> For example, **Baking**, the central system's process, is the ellipse in [Illustration 9]. The remaining five things are objects (the rectangles) that enable or are transformed by **Baking**. **Baker** and **Equipment** are the *enablers* of **Baking**, while **Ingredients Set**, **Energy**, and Bread are its *transformees* -- the **objects** that are transformed by Baking. As the direction of the arrows indicates, **Ingredients Set** and **Energy** are the *consumees* -- they are consumed by **Baking**, while **Bread** is the resultee -- the object created as a result of Baking. As soon as the modeler starts depicting and joining things on the graphics screen, OPL sentences start being created in response to these inputs. They accumulate in the OPL pane at the bottom of [Illustration 9], creating the corresponding OPL paragraph, which tells in text the exact same story that the OPD does graphically.
>
> As the example shows, the OPL syntax is designed to generate sentences in plain natural, albeit restricted, English, with sentences like "**Baking** yields **Bread**." This sentence is the bottom line in Fig. 7.1. An English subset, OPL is accessible to nontechnical stakeholders, and other languages can serve as the target OPL. Unlike programming languages, OPL names can be phrases like **Ingredients Set** (Dori 2011, 212–213).

To progress communications in service systems thinking, making a distinction between (I) the intention-oriented perspective through i*, and (ii) the function-structure-behavior perspective in OPM is worth consideration. Although interests, benefits, value propositions and resources could be represented in OPM as well as in i*, their primacy of these elements in a service system calls for ways to increase their salience.

2.7 Service systems science has a promise to synthesizes disciplines, as did computer science

Service systems thinking, as a new field, will draw heavily on a foundational service science that has its origins only as recently as 2005. The prior experiences of IBM in the emergence of a new science of computing are a parallel. In the 1970s, the IBM Research organization was composed of physicists, chemists, electrical engineers and mathematicians. To respond to business changes requiring software systems research, new Ph.D.s joined the organization in large numbers.

> Some colleagues in IBM and in academia advocated a bold approach– creating a new academic discipline called service science (Chesbrough 2004, 2005; Horn 2005), which aims theories and methods from many different disciplines at problems that are unique to the service sector. At the start, the particular disciplines (including some engineering, social science, and management disciplines) and the particular problems (e.g., improving service innovation and service productivity) were not clear. However, this idea of an integrated service science was particularly appealing to us, as we found that the number of separate PhDs required to form a suitable services research organization had grown to nearly a dozen! We had recruited PhDs in anthropology, cognitive psychology, computer science, cognitive science, education, human factors, industrial engineering, and organizational psychology, among others. The communication challenge alone of getting such a diverse population of scientists to speak a common language around "service innovation" required training everyone in each others' disciplines to some extent, as well as injecting new, practical concepts fresh from the front lines of our own services business (Spohrer and Maglio 2008, 239).

The feature of coproduction, offerings, values and resources described above in Section 2 have cross business strategy, marketing, psychology, economics, computer science and philosophy. Improving communications across and amongst the disciplines into a new field is a challenge that may require a generation to new scholars to become fully institutionalized.

To the disciplines described above, Section 3 explores contributions from the architecting and design of built environments that had previously been cross-appropriated to computer science. Wisdom from decades of practice in those fields can inform the development of service systems thinking.

3. ORIENTATION: THE HISTORY OF ARTICULATIONS BY CHRISTOPHER ALEXANDER ARE SALIENT TO SERVICE SYSTEMS

Christopher Alexander is best known for this 1977 book, *A Pattern Language*. That work, however, was part of a larger endeavor that has extended over the following 25 years. Homeowner may be most interested in *A Pattern Language* to features of their dwellings, but should not a broader context in the subtitle of the book of "Towns, Building, Construction". The Design Patterns and collateral work in software development may not find subsequent developments by the Center for Environmental Structure relevant. However, as we look towards developing a pattern language for service systems, our perspective on the advances should be expanded.

The development of Alexander's work can be traced through his work as a builder, which has shaped his architectural theory. When the stakeholders of a system expand beyond an individual owner and his or her family, the variety of interests expands. Significant advances have been made in the development of larger scale projects, in:

- generative pattern language in a semi-lattice organization;
- participation, piecemeal growth and funding coordination;
- centering and wholeness; and
- unfolding wholeness through local adaptation.

An outline of the trail of development follows.

3.1 Pattern languages are premised on the approaches for architectural programming

The context for a generative pattern language has its roots in architectural programming. While Christopher Alexander was appointed as a research professor at U.C. Berkeley in 1965, and the Centre for Environmental Structure was formed in 1967, the idea of architectural programming was documented by practitioners from Caudill Rowlett Scott, Architects, Planners, Engineers in Houston Texas.

> Programming is a specialized and often misunderstood term. It is "*a statement of an architectural problem* and the requirements to be met in offering a solution. While the term is used with other descriptive adjectives such as *computer* programming, *educational* programming, *functional* programming, etc., in this report, programming is used to refer only to architectural programming.
>
> Why programming? The client has a project with many unidentified sub-problems. The architect must define the client's total problem.
>
> Design is problem solving; programming is problem-seeking. The end of the programming process is a statement of the total problem; such a statement is the element that joins programming and design. The "total problem" then serves to point up constituent problems, in terms of four considerations, those of form, function, economy and time. The aim of the programming is to provide a sound basis for effective design. The State of the Problem represents the essense and the uniqueness of the project. Furthermore, it suggests the solution to the problem by defining the main issues and giving direction to the designer (Pena and Focke 1969, 3).

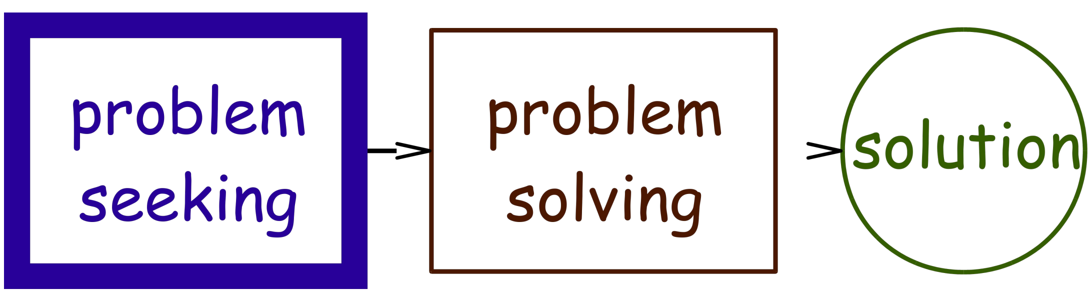

Illustration 10: Programming is problem seeking, design is problem solving (Pena and Focke 1969)

Some key traps for the architectural programmer working with the client were outlined.

How Much Information is Enough?

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

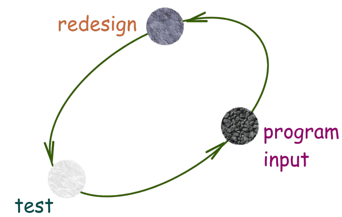

Illustration 11: Programming through design, testing and

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

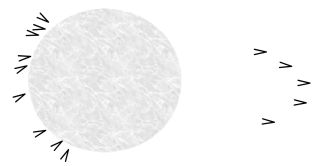

Illustration 12: Discrimination between major ideas and details is necessary to avoid confusion in problem solving

The analytical procedure used by CRS provides a framework for decision making. Within it the architect help the client identify and make decisions that need to be made prior to design. Within it, the architect can suggest alternatives and other information to bring about decisions. There are times when the architect must evaluate the gains and risks in order to stimulate a decision. Yet, note the emphasis on client decisions; the architect merely participates and at most, recommends.

The new sophisticated client wants to know how his project will be processed and when he will be involved. He wants to remove the mystique associated with the programming and design of his project (Pena and Focke 1969, 4–6).

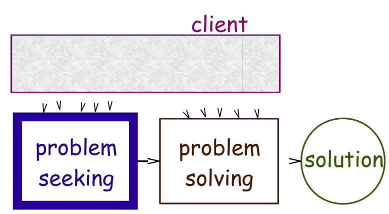

Illustration 13: The client is involved in the process

The separation of programming from design should be clear. In this architectural practice, the roles of the programmer and the designer are distinct.

Two terms need to be understood and added to the glossary of architectural practice: “Programmatic concepts” and “design concepts.” Programmatic concepts refer to the ideas intended mainly as solutions to the client's own management problems so far as they concern function and organization. Design concepts, on the other hand, refer to ideas intended as physical solutions to architectural problems.

Programmatic concepts and design concepts are so closely related that one is mistaken for the other. Design concepts are the physical response to programmatic concepts. For example, *open planning* is the physical response to *integration* of activities. In practice the confusion is compounded because most architects and some clients tend to think more easily in physical terms.

Programmatic concepts must be stated abstractly so as not to inhibit design alternatives unnecessarily. For example, the programmatic concept of *decentralization* may find a design response in either *compactness* (vertical or horizontal) or *dispersion* (varying degrees) (Pena and Focke 1969, 6–7).

Architectural programming is balance of function, form, economy and time. The considerations will impact the client and users of the built environment.

The Four Basic Considerations

If design of the facility is to solve problems of function, form, economy and time, then programming must treat these as basic considerations by which to classify information.

The first of these, *function*, deals with the functional implication of the client's aims, methods to be used to meet them, and numbers and types of people. it deals with social and functional organization. Contributions to the client could by by management consultants, behavioral scientists, and architects with intuitive insights into social values.

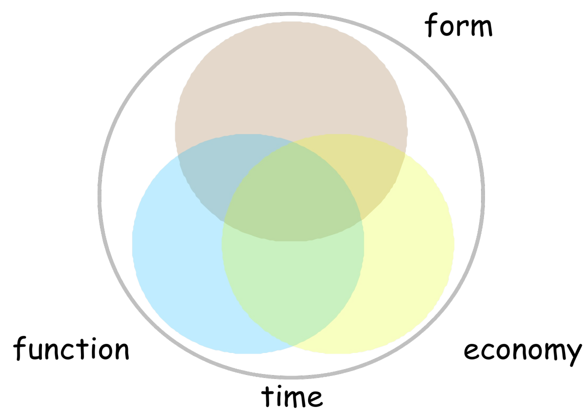

Illustration 14: The whole problem consists of the consideration for form, function, economy and time (Pena and Focke 1969)

Form, the second consideration, is used by CRS to evoke questions regarding the physical and psychological environment to be provided, the quality of construction and the conditions of the site. The physical environment involves physical needs such as illumination, heating, ventilating, air-conditioning and acoustics. The psychological environment raises values which might affect user behavior; the architect must inject these intuitively until such time as analytical means are developed.

The third consideration, economy, emphasizes the need for early cost control and brings up for consideration by the programming team the initial budget, the operating cost and long term cost which may be affected by initial quality of construction.

Consideration four, time, brings out the factors of change and growth, which affect function, form and economy (Pena and Focke 1969, 14–16).

The third consider

When programming is done properly, the wants and needs of the client are appreciated not as static functional specifications, but instead as goals that may evolve. A systemic approach could be evident in "negotiable programming".

Building Systems and "Negotiable Programming"

> The expanding trend to system building affects the entire building project delivery process. In programming terms, a resolve to use building systems is a goals-oriented decision which is tested at the first (goals formulation) step in programming, and, if verified, will affect program content.
>
> The use of system building makes possible a more general, flexible form of programming conveniently referred to as “negotiable programming”. Negotiable programming presupposes that the building has been developed from user requirements and performance criteria, and that it will produce the kind of flexibility that will make net space requirements “negotiable” within a fixed gross area. The aim is to make the end product a building with the flexibility to change as user requirements change.
>
> Through recourse to system building every program requirement remains negotiable throughout the design and building process, and because of inherent flexibility the functional organization of the interior remains always negotiable (Pena and Focke 1969, 36–37).

<< The challenge of problem seeking >>
In addition, Alexander was critical that design problems did not come in a tree-like structure, saying that “A City is Not A Tree” and was instead a semi-lattice organization.

> Too many designers today seem to be yearning for the physical and plastic characteristics of the past, instead of searching for the abstract ordering principle which the towns of the past happened to have, and which our modern conceptions of the city have not yet found. These designers fail to put new life into the city, because they merely imitate the appearance of the old, its concrete substance: they fail to unearth its inner nature.
>
> What is the inner nature, the ordering principle, which distinguishes the artificial city from the natural city? You will have guessed from the first paragraph what I believe this ordering principle to be. I believe that a natural city has the organisation of a semi-lattice; but that when we organise a city artificially, we organise it as a tree (Alexander 1966).

Alexander cited the Jane Jacobs' 1961 The Death and Life of Great American Cities as similarly pursuing the life in cities.

3.2 Circa 1967, an institution focused on pattern languages for the built environment was formed

<< Pattern manual >>

3.3 Circa 1968, generative pattern language was expressed as an extension of systems thinking

As architectural theory, “systems generating systems” was presented as four points:

> 1. There are two ideas hidden in the word system: the idea of a *system as a whole* and the idea of a *generating* system.
>
> 2. A *system as a whole* is not an object but a way of looking at an object. It focuses on some holistic property which can only be understood as a product of interaction among parts.
>
> 3. A *generating* system is not a view of a single thing. It is a kit of parts, with rules about the way these parts may be combined.

> 4. Almost every ‘system as a whole’ is generated by a ‘generating system’. If we wish to make things which function as ‘wholes’ we shall have to invent generating systems to create them. [Alexander 2011, p. 59; Alexander 1968, p. 605]
>
> In a properly functioning building, the building and the people in it together form a whole: a social, human whole. The building systems which have so far been created do not in this sense generate wholes at all (Alexander 1968).

3.4 Circa 1968, the pattern language on multi-service centers prescribed a generic scheme for organization

Shortly after the founding in 1967 of the Center for Environmental Structure with the specification of a pattern manual (Alexander, Ishikawa, and Silverstein 1967), the architects constructed a report that demonstrated how a pattern language would work.

> In this report, we present a prototype for multi-service center buildings.
>
> A multi-service center is a community facility, which provides a variety of special services to citizens. It is intended especially to help solve some of the problems of low income communities. Experimental multi-service centers have been started in many cities throughout the United States. however, there is not yet any agreement about the form which multi-service centers should take – either in their human organization, or in their special organization.
>
> Our report deals chiefly with the spatial organization; but since human and spatial organization cannot property be separated, many of the specifications given in this report, go deeply into question of human organization as well (Alexander, Ishikawa, and Silverstein 1968, 1).

In this early work, the focus was on the pattern language, oriented specifically towards architecting built environments. While Alexander acknowledged that human organization and spatial organization could not be separated, his focus was on the built environment.

3.5 Circa 1971-1974, processes of participation, piecemeal growth and funding coordination rose

Published in 1975, The Oregon Experiment x

> … this book … is the master plan for the University of Oregon, and describes a practical way of implementing these ideas in a community. However, we must emphasize at once that we are dealing here with a very special king of community. Unlike most communities, it has a single owner (The State of Oregon), and single, centralized budget. This situation is not only unusual, it is even opposite to the ideas which are actually needed to make the way of building which we call the timeless way, appear in a society. However, we believe that a modified version of this way of building is possible, even under these restrictions, and this book, beyond its function as a master plan for the University of Oregon, is our attempt to define this process.
>
> This process will apply in full, to any other community where there is a single owner, and a single centralized budget. This means that it will apply, for example, to a kibbutz, a

> hospital, a corporate industrial plant, a farm, a cooperative factory, any settlement where the concept of private property has been abolished, and any benevolent institution run by a government for the welfare of its citizens.
>
> We repeat, that we do not consider these types of institutions ideal. In a future book, we shall describe the process of implementation that is needed in a more ideal neighborhood or community, where people own their houses, common land and workshops, and there is not centralized budget. In this book, we nevertheless propose a process which can allow people under the half-ideal conditions of the centralized budget, to take care of the environment for themselves, and have some measure of control over their own lives (Alexander et al. 1975, 3–4).

Six principles were proposed:

> 1. The principle of organic order.
> Planning and construction will be guided by a process which allows the whole to emerge gradually from local acts.
>
> 2. The principle of participation.
> All decisions about what to build, and how to build it, will be in the hands of the users.
>
> 3. The principle of piecemeal growth.
> The construction undertaken in each budgetary period will be weighed overwhelmingly towards small projects.
>
> 4. The principle of patterns.
> All design and construction will guided by a collection of communally adopted planning principles called patterns.
>
> 5. The principle of diagnosis.
> The well being of the whole will be protected by an annual diagnostic which explains, in detail, which spaces are alive and which ones dead, at any given moment in the history of the community.
>
> 6. The principle of coordination.
> Finally, the slow emergence of organic order in the whole will be assured by a funding process which regulates the stream of individual projects put forward by users. (Alexander et al. 1975, 5–6)

The Oregon Experiment was the third volume in a series. The first volume was 1977 *The Timeless Way of Building* that "describes a theory of planning and building which is, essentially a modern post-industrial version of the age-old pre-industrial and traditional processes which shaped the world's most beautiful towns and buildings for thousands of years" (Alexander 1979). The second volume was the 1977 *A Pattern Language* "explicit set of instructions for designing and building, which defines patterns at every scale, from the structure of a region to the nailing of a window; set out in such a way that laymen can use it to design a satisfying and ecologically appropriate environment for themselves and their activities"(Alexander, Ishikawa, and Silverstein 1977).

3.6 Circa 1976-1978, a centering process was seen as a way in which wholeness would develop

Published in 1987, A New Theory of Urban Design

> During the period of 1976-1978 one the authors (CA), had become aware a deeper level of structure lying "behind" the patterns. At this level of structure, it was possible to define a small number of geometric properties which seemed to be responsible for wholeness in space. Even more remarkable, it was possible to define a single process, loosely then called "the centering process," which was capable of producing this wholeness (with its fifteen or so geometric properties) at any scale at all, irrespective of the particular functional order required by the particularities of a given scale. [….]
>
> … we began to imagine a process of urban growth, or urban design, that would create wholeness in the city, almost spontaneously, from the actions of the members of the community … provided that every decision, at every instant, was guided by the centering process (Alexander et al. 1987, 4–5).

Alexander proposed seven detailed rules for growth.

1. Piecemeal growth: Growth should occur incrementally.
2. The growth of larger wholes: Each increment of growth should help form larger centers.
3. Visions: Proposed growth must be experienced and expressed as a vision.
4. Positive urban space: Buildings must create coherent adjacent public space.
5. Layout of large buildings: The layout of a building should be coherent with with building's overall position.
6. Construction: The structure of every building must generate smaller wholes within itself.
7. Formation of centers: Every whole must be a center in itself and must also provide a system of centers around it.

x

3.7 Circa nnnn – Nature of Order

3.8 Circa 2012, the potential for unfolding wholeness through local adaptation was presented as an alternative to the dominant systems of efficiency and control

Published in 2012, The Battle for Live and Beauty of the Earth described the development of Eishin campus.

> From the very beginning of the building project described in this book, we intended to show that architecture can bring life to a community – indeed, that it is necessary in order to help the community come to life. Thus, we mean to show how the physical fabric of the buildings plays a necessary and unavoidable role in the success of a community. [….]
>
> The purpose of all architecture, the purpose of all spatial-geometric organization, is to provide opportunity for live-giving situations. The central issue of architecture, and its central purpose, is to create those configurations and social situations, which provide encouragement and support for life-giving comfort and profound satisfaction – and sometimes excitement – so that one experiences life as worth living. When this purpose is forgotten or abandoned, there is no architecture to speak of (Alexander 2012, 2–3).

This publication coincided with four-volumes of The Nature of Order.

3.9 Challenge of teleological versus nonteleological

<< Introna >>

4. POSSIBILITIES: THE EMERGING SERVICE SYSTEMS THINKING COMMUNITY HAS SOME QUESTIONS TO CONSIDER

The

4.1 How should the community evaluate its progress on producing a generative pattern language?

<< Unfolding wholeness >>

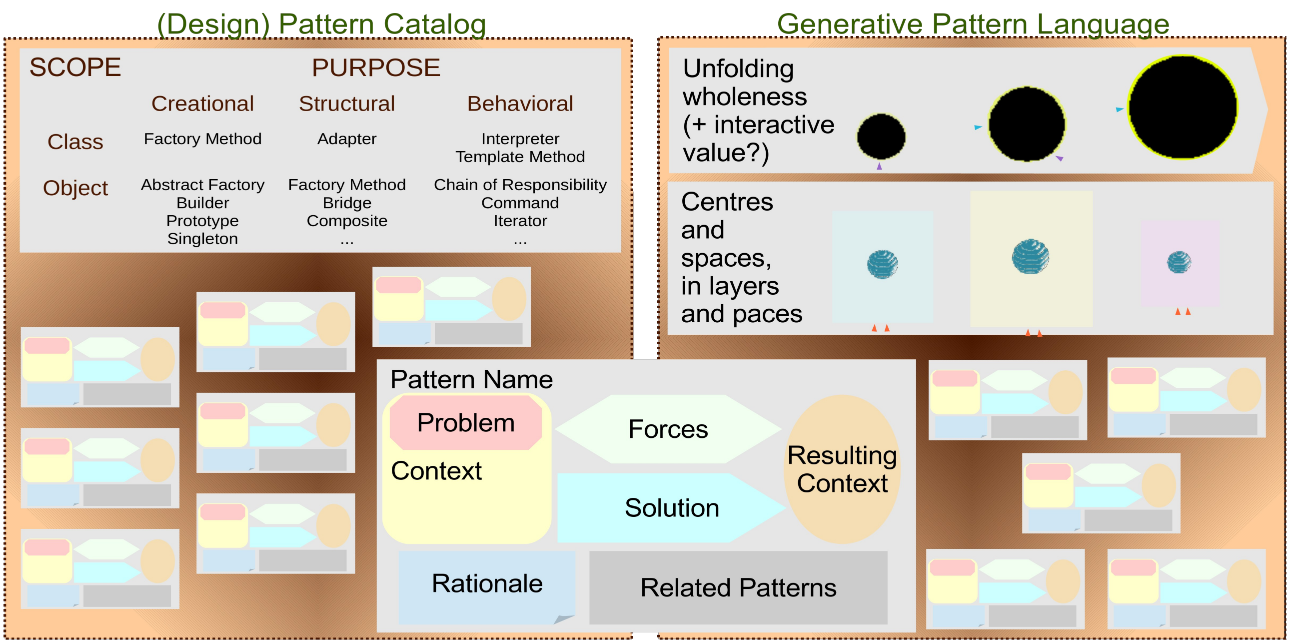

Illustration 15: From a Design Pattern Catalog to a Generative Pattern Language

<< Centers and spaces in layers and in paces (Brand 1994) >>

4.2 Does servicizing existing pattern languages make sense?

<< list >>

<< Federated wiki (Cunningham 2012b) could help >>

4.3 Will diagrammatic representations make the pattern languages easier?

Compare to Gang of Four (Gamma et al. 1995)

In the 1971 preface to the paperback edition of Notes on the Synthesis of Form, Alexander made the explicit tie between patterns and diagrams.

> Today, almost ten years after I wrote this book, one idea stands out clearly for me as the most important in the book: *the idea of the diagrams*.
>
> These diagrams, which, in my more recent work, I have been calling patterns, are the key to process of creating form .

4.4 Should actors be explicitly incuded in the generative pattern language?

<< Service systems have to serve >>
<< Try iStar >>

4.5 Can structural and processual viewpoints be simultaneously addressed in interactivity?

<< Separation of structural patterns from process patterns>>

4.6 Is there a contribution that Open Systems Theory make towards architectural programming?

What is systems thinking?

> Systems thinking is a perspective on parts, wholes and their relations (Ing 2013).

<< DP 1 … DP 3 >>
Service Systems Thinking

- Engagement
- Development
- Enjoyment

5. FUTURE ACTION: CONTINUING THE CONVERSATIONS

At the beginning of the journey, Service Systems Thinking is endorsed the some leading professional organizations
Preliminary outlines have been endorsed by:

- the International Society for the Systems Sciences (ISSS);
- the International Council on Systems Engineering (INCOSE); and
- the International Society for Service Innovation Professionals (ISSIP).

Meetings in which participation will be encouraged include:

- June 2014, Las Vegas – the International Symposium of the International Council on Systems Engineering (INCOSE);
- July 2014, Krakow, Poland, at the Human Side of Service Engineering Meeting; and
- July 2014, at the annual meeting of the International Society for the Systems Sciences (ISSS);

PloP 2013 is a place where a group focused on Service Systems Thinking can convene.

REFERENCES

Ackoff, Russell L., and Fred E. Emery. 1972. On Purposeful Systems. Aldine-Atherton.
Alexander, Christopher. 1966. “A City Is Not a Tree.” Design, Council of Industrial Design, , no. 206.

———. 1968. “Systems Generating Systems.” Architectural Digest 38. http://books.google.ca/books?id=Ib4yEJErR5EC&pg=PA58.

———. 1979. The Timeless Way of Building. Oxford University Press. http://books.google.ca/books?id=H6CE9hlbO8sC.

———. 2012. The Battle for the Life and Beauty of the Earth: A Struggle Between Two World-Systems. Oxford University Press. http://books.google.com/books?id=xAMEywAACAAJ.

Alexander, Christopher, Sara Ishikawa, and Murray Silverstein. 1967. Pattern Manual. Berkeley, California: Center for Environmental Structure. http://books.google.com/books?id=_VYpGwAACAAJ.

———. 1968. A Pattern Language Which Generates Multi-Service Centers. Center for Environmental Structure. http://books.google.ca/books?id=FGdPAAAAMAAJ.

———. 1977. A Pattern Language: Towns, Building, Construction. New York: Oxford University Press US. http://books.google.ca/books?id=hwAHmktpk5IC.

Alexander, Christopher, Hajo Neis, Artemis Anninou, and Ingrid King. 1987. *A New Theory of Urban Design*. Oxford University Press. http://books.google.com/books?id=STCoRQzgDZcC.

Alexander, Christopher, Murray Silverstein, Schlomo Angel, Sara Ishikawa, and Denny Abrams. 1975. *The Oregon Experiment*. Oxford University Press. http://books.google.ca/books?id=u2NSI4vSu_IC.

Cunningham, Ward. 2012a. “Smallest Federated Wiki.” *Federated Wiki*. http://ward.fed.wiki.org/view/welcome-visitors/view/smallest-federated-wiki.

———. 2012b. *Federation*. MOV. The Realtime Conference. Portland, OR. http://vimeo.com/52637141.

Dori, Dov. 2006. “Modeling Knowledge with Graphics and Text Using Object-Process Methodology.” In *Encyclopedia of Knowledge Management*, 683–93. Hershey, PA: Idea Group. http://esml.iem.technion.ac.il/wp-content/uploads/2011/08/Object-Process-Methodology.pdf.

Horkoff, Jennifer, and Eric Yu. 2006. “Basic I* Notation.” *iStar Quick Guide*. http://istarwiki.org/tiki-index.php?page=iStarQuickGuide#Basic_i_Notation.

IBM. 2010. *The World’s 4 Trillion Dollar Challenge: Using a System-of-Systems Approach to Build a Smarter Planet*. GBE03278-USEN-02. Institute for Business Value. http://www-935.ibm.com/services/us/gbs/bus/html/ibv-smarter-planet-system-of-systems.html.

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/.

Ing, David. 2013. “Rethinking Systems Thinking: Learning and Coevolving with the World.” *Systems Research and Behavioral Science* 30 (5): 527–47. doi:10.1002/sres.2229.

Normann, Richard. 1984. *Service Management: Strategy and Leadership in Service Businesses*. Wiley. http://books.google.ca/books/about/Service_management.html?id=gT4UAQAAMAAJ.

———. 2001. *Reframing Business*. New York: Wiley.

Normann, Richard, and Rafael Ramirez. 1993. “From Value Chain to Value Constellation: Designing Interactive Strategy.” *Harvard Business Review* 71: 65–65. http://hbr.org/1993/07/designing-interactive-strategy.

———. 1994. From Value Chain to Value Constellation: Designing Interactive Strategy. Chichester, England: Wiley.

Ramirez, Rafael, and Johan Wallin. 2000. Prime Movers: Define Your Business or Have Someone Define It against You. Chichester, England: Wiley.

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

Spohrer, Jim. 2005. “Why the World Needs More Systems Thinkers Focused on Service Systems”. Conference presentatation presented at the 49th Annual Meeting of the International Society for the Systems Sciences, Cancun, Mexico, July 4. http://isss.org/world/cancun-2005-retrospective#Monday.

Spohrer, Jim, 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.

Spohrer, Jim, Paul P. Maglio, John Bailey, and Daniel Gruhl. 2007. "Steps Toward a Science of Service Systems." *Computer* 40 (1): 71–77. doi:10.1109/MC.2007.33.

Trist, Eric L., and Hugh Murray. 1997. "Historical Overview: The Foundation and Development of the Tavistock Institute to 1989." In *The Social Engagement of Social Science: The Socio-Ecological Perspective*, edited by Eric L. Trist, Frederick Edmund Emery, and Hugh Murray, 3:1–35. Philadelphia: University of Pennsylvania Press.

Vargo, Stephen L., and Robert F. Lusch. 2004. "The Four Service Marketing Myths Remnants of a Goods-Based, Manufacturing Model." *Journal of Service Research* 6 (4): 324–35. doi:10.1177/1094670503262946.

Winograd, Terry. 1986. "A Language/action Perspective on the Design of Cooperative Work." In *Proceedings of the 1986 ACM Conference on Computer-Supported Cooperative Work*, 203–20. Austin, Texas: ACM. doi:10.1145/637069.637096.

Received September 2014; revised xxxx 2014; accepted xxxx 2014