From Service Innovation to Service Engineering[1]

Wil Janssen, Marc Lankhorst, Timber Haaker, and Henny de Vos

Abstract Many companies have made important steps in moving from process engineering, through process management to enterprise architecture. This allowed them to get a grip on their processes and systems, reducing cost and lowering risks. The question that arises is how to leverage such a foundation towards added value for the customer. True value comes from a robust yet flexible service infrastructure together with a professional approach to service development.

In many service organizations such an approach is largely lacking. Service development and innovation is loosely defined, responsibilities are distributed over marketing, IT and business lines, and interaction between the different stakeholders involved is poor or error prone. Service science, management and engineering, or service science for short, has been coined as the term of a new discipline that allows for an integrated approach to the development of services. In this paper we explain this new paradigm, what it encompasses and how building blocks, such as enterprises architecture, stress testing, process management and creative design fit in. We also suggest a roadmap towards services science.

1 The Importance of Services in Modern Society

Europe as well as the US are becoming service economies. Service sectors are responsible for about 70% of the GDP in Europe [20]. In the Netherlands, the complete growth of employment over the last 10 years comes from services,

[1]Part of this work was done as under the Dutch Service Innovation and ICT program (SI-I), supported by the Ministry of Economic Affairs, Agriculture & Innovation.

W. Janssen (✉) • M. Lankhorst • T. Haaker • H. de Vos
Novay/InZycht, Bamshoevelaan 71, 7523 JJ, Enschede, The Netherlands
e-mail: Wil@InZycht.nl; marc.lankhorst@novay.nl; Timber.Haaker@novay.nl; Henny.deVos@novay.nl

I. Ivanov et al. (eds.), *Cloud Computing and Services Science*, Service Science: Research and Innovations in the Service Economy, DOI 10.1007/978-1-4614-2326-3_2,

especially in healthcare [6]. As the Europe 2020 Strategy [10] makes clear, Europe's future wealth and citizens' wellbeing depend on how effectively its businesses innovate and respond to changing markets, technologies and consumer preferences. We therefore need a better understanding of how innovation is changing and how the traditional divide between manufacturing and services is blurring.

To sustain our welfare level, economic growth is needed. US productivity is higher than the European one, and has recently become even higher than leading countries in Europe [20]. And the European productivity growth is much lower than the US. The service sector is the main reason; industry has a growth comparable to the US. In the public sector, productivity growth is even smaller. In the Netherlands, only healthcare productivity has grown, whereas cost in public sector has grown drastically. All this implies that we are in need of a better approach of service innovation, both in the public sector as in the private sector.

Product-oriented companies are now adopting new service-focused business models. At the same time, service firms increasingly exploit new devices, technologies and infrastructures, such as smartphones, tablets, or interactive televisions, to improve their customers' experiences. Innovation is no longer the preserve of research and development laboratories but has become more of a distributed, cultural phenomenon, where the processes for developing new goods and services, channels to market and revenue models are evolving in response to new technological opportunities, increased customer engagement in innovation, and changing organizational structures [11, 12]. Information and communication technology is generally recognized as a driving force of innovation [10]. The impact of ICT reaches far beyond the ICT sector itself. In a recent survey among 300 innovative companies in the Netherlands, 80% of people pointed to ICT as a driving force of innovation as well as cost reduction [21].

Despite its importance, the level of professionalism in developing services cannot match the level of expertise in product development. Business cases, user studies, design alternatives and actual development are not really linked, and information and knowledge is lost en route. Especially in the case of ICT-based services, initial requirements are underspecified, leading to change requests in the process, with higher cost, longer time to market, and increased risk of even not meeting the requirements.

Whereas developing an individual service is already complex, understanding service networks adds a level of complexity to that. The interdependencies between various actors and stakeholders, the distribution of task or services in the network brings an additional dimension to the problem area. Our goal, therefore, is to work towards a rigorous, model-based, service development methodology, or service engineering approach for ICT-supported services [30]: a design methodology that is problem-oriented, encourages inventive and cognitive skills, generates systematic solutions that are transferable, and is teachable and learnable [5].

This paper is organized as follows. From the macro-economic challenges sketched above, we move to the challenges organizations face in this context. We then elaborate on the concept of service orientation in organizations before giving three main innovation areas in services.

We end by delineating an integral framework for developing services to tackle the challenges identified.

2 Innovation Challenges for Organizations

We sketched a number of societal challenges related to service innovation. The consequences of these for individual organizations are indirect at best. Companies as well as public organizations face different challenges. Confronted with a highly dynamic customer base, especially in the B2C market, being able to adapt to changing customer needs is crucial. This holds for both the channels through which customers are found and served as well as the personalization of services. Think of the role of new devices and apps for shopping, and the full digitization of tax-related services.

Agility towards the customer has consequences for operational agility. The interdependencies of products, services, systems, processes and IT can severely constrain the ability of organizational change. Every organization needs to think strategically about where it needs agility as a core competence, and develop its enterprise architecture accordingly. Only then the time to market of new or changing services can match the demands of customers. Sambamurthy et al. [27] show the connection between the IT competence of an organization, the digital options this creates, the customer, partnering and operational agility resulting from these options, and the competitive actions the organization can take. And all of these crucially depend on what they call entrepreneurial alertness: strategic and systemic foresight. Less offensive, but as important for most organizations is cost reduction. Price pressure is substantial for many companies like insurance companies and banks as well as telecom operators. This especially holds for public service organization which have been confronted with the largest budget cuts in decades. Next to reorganizing the organization, such severe cuts need rethinking the business model of the organization.

Focus on customers and cost may come at a price: reduced intrinsic innovation capabilities. An organization needs to be able to innovate constantly and therefore needs skills to collaborate with external partners [8, 36] as well as combine operational excellence with new product development [34]. The latter requires a so-called *ambidextrous organization* [22]. This means that an organization should take care to invest in the right portfolio of projects, serving a combination of short term goals and longer term objectives, leading to a mixed set of competences in the organization.

3 The Service Orientated Organization

IT management encompasses different aspects, ranging from determining the strategic orientation of the IT organization to management and control of delivery and operations. Furthermore, the information systems landscape itself, especially that of large, information-intensive organizations, has become a complex field that combines all kinds of concepts, paradigms, building blocks, and instruments. Think of paradigms like process management, rule based organization, service orientation, event based, or Software as a Service (SaaS) and cloud. How can we get a grip on this multifaceted landscape?

It is impossible to manage all these different elements individually. Some of these are too fine-grained, such as business rules or events; some are too IT-centric, such as business objects or components; some are too large and serve too many purposes to manage them as a single functional element, such as complete business applications like ERP systems; and some of these, such as business processes, are too business-specific to provide a management handle on more generic IT functionality. We need a concept that is in between these other notions and captures the essence of what an organization does or means for its surroundings: *service*.

Using the notion of service as core concept in guiding the development of organizations, both for business and for IT design, has several advantages. First, services provide a clean separation of the "what" and "how". A service provides a clear interface to its functionality, without disclosing how this functionality is realized internally. As such, a service is self-contained and has a clear purpose and function from the perspective of its environment. Its internal behaviour represents what is required to realize this functionality. For the "consumers" or users of a service, the internal behaviour of a system or organization is usually irrelevant: they are only interested in the functionality and quality that will be provided.

This also points to the second advantage of the service notion: a service is independently useful and therefore has a manageable level of granularity. Since it delivers a concrete business contribution, it is the subject of service-level agreements, its performance can be monitored separately, it can be combined with other services to provide new functionality, and it can be bought from and sold to other organizations.

Finally, services provide a bridge between business and IT vocabulary. In business terms, "service" signifies what the organization does for its customers; more recently, IT has started to use "service" for concrete, independent units of business functionality delivered via a software interface. Both uses of the word are based on the concrete contribution to the environment and the relatively self-contained character of a service.

This, of course, is not really new. At the edges of organizations we have long been thinking in terms of the services provided to customers, and internal business processes were designed to provide these services. Software engineers think in terms of functional interfaces, information hiding and encapsulation. Service thinking, however, can also be applied to, for example, internal business processes and software applications, rendering them into "service networks": services become the core building block of the entire information ecosystem.

These services can be provided at a distance, over the Internet. Some years ago, subscribing to software in this way was labeled as "Application Service Provisioning" (ASP). This never made it big, at least partly because it was based on a direct link between application and customer: the ASP provider in fact merely hosted and maintained the application for each customer separately. Many of these applications were not developed for multi-tenant use, delivery across the Internet, or pay-per-use billing models. Newer delivery models, collectively called "Software as a Service" and cloud computing, have overcome these limitations.

Service orientation also stimulates new ways of thinking. Traditionally, applications are considered to support a specific business process, which in turn realizes a specific business service. Service orientation allows us also to adopt a bottom-up strategy, where the business processes are just a mechanism of instantiating and commercially exploiting the lower-level services in a collective offering to the outside world. In this view, the most valuable assets are the capabilities to execute the lower-level services, and the business processes are merely a means of exploitation.

By concentrating on agile development of business and software services, we focus on the value that organizations provide to their environment. Of course, these services are realized by all kinds of business processes, software applications and technical infrastructure. However, these are subordinate to the services they deliver. Traditionally, agile methods are strongly focused on software development; here, we take a much broader scope, applying agile principles and practices to more than just software.

4 Systematic Service Innovation

Organizations face numerous challenges, ranging from growth targets, matching rapidly changing customer demand, to cost reduction and continuous innovation. We illustrated these challenges above. The service oriented organization paradigm provides a basis to meet these challenges.

In this section we identify three major areas where organizations can build upon service orientation to tackle the challenges in a systematic way: robust service model design, agile organizing and innovation investing.

4.1 Being able to Develop Robust Business Models

The introduction of new services and service processes in the continuously changing business landscape requires careful and informed business planning that takes into account the relevant developments in the market, society and technology.

The true value of innovative concepts and technologies is largely determined by the business models in which they are embedded. Choices are complex as cooperation with others in value networks is often necessary and multiple business model options are available. Companies therefore have a need for a long-term vision on potential business models, their own position within these models and the road that may lead them to this position, including an analysis of the robustness of the business model with respect to different context influences.

For this we coined the term business model roadmapping, i.e. a description of the chain of intermediate steps and critical choices to arrive at a desired business model. This creates a longitudinal insight into the opportunities for business innovation and related business planning. To validate the robustness of business models and roadmaps, scenario analysis can be used: validating the strong and weak parts of business models and roadmaps by applying scenario analysis methodologies. As a result the "fit" of a business model with a future business environment can be determined or the "robustness" of a business model with regard to a collection of future environments. This methodology is called business model stress testing.

Business model stress testing builds upon ingredients in business modeling [32], business model generation [23], and scenario analysis. The main concepts and results in the approach are given in the table below (Table 1).

Over the past few years, the field of business models has developed from defining business models, via exploring business model components and classifying business models into categories, towards developing descriptive models (for an overview, see [24]).

First of all, it is important to consider what a business model is. We agree to a large extent with the definition presented by Chesbrough and Roosenbloom [7], that a business model is a blueprint for the way a business creates and captures value from new services or products. As such, a business model describes how a company or network of companies aims to make money and create consumer value for a specific service offering [3]. Central in the business model definition is that a viable business model should create both customer value and network value.

Scenario analysis or scenario thinking, has a long tradition as first studies originate from the 1960s. A famous scenario planning example was set by oil company Shell, which anticipated the 1973 oil crisis by including one alternative scenario on a shortage of the oil supply due to political tensions in the Middle East and the subsequent rise of the oil prices [17, 29]. Scenario planning typically was adopted by the military, and implemented by for instance the Rand Corporation. While this Anglo-Saxon scenario approaches focussed on improving strategic decision making and planning processes, the French *La Prospective* school developed a scenario approach for institution and companies to deal with long-term planning. Their "normative scenarios" could serve as a guiding vision to policy and decision makers. Over the years, scenario thinking has become a common approach in many industry domains, ranging from energy and telecommunications to global economics [15].

Table 1 Key concepts in the business model analysis

Processes	Results
Business model design	*Business model*
Process of describing an existing or (re)designing a new business model. Essential: the business model design should allow for a network perspective.	Description of how a company or network of companies aims to make money and create consumer value for a specific service offering [3].
Scenario analysis	*Scenario*
Process of developing one or more scenarios based on an analysis of trends, certainties and uncertainties.	Expectations regarding possible futures that provide insight into the way the future may develop based on clearly defined assumptions concerning the relationship between relevant developments.
Business model stress testing	*Business model strengths and weaknesses*
Process that critically evaluates if a business model is viable and feasible in a scenario. Essential: There should be an alignment between the scenarios analysis and business model design in order to be able to compare them.	Overview of those elements in the business model that fit with a certain scenario.
Business model roadmapping	*Business model roadmap*
Process of developing a business model roadmap as a plan with intermediate steps achieve a desired business model *B* starting from a business model *A*.	Description or a plan that describes what intermediate steps and critical decisions have to be taken to achieve a desired business model.

4.2 Making your Service Organization Agile

The agile movement in software development has received much attention over the last two decades. These light-weight, iterative methods have gradually taken over much of the software development community, because on the one hand they provide better results in many types of projects, and on the other hand they provide a more stimulating work environment for developers. Already in the 1980s, with methods like James Martin's Rapid Application Development [19], the focus in software development started to shift from linear, waterfall-like methods to iterative and interactive approaches. In the 1990s, the three most important agile methods arose: extreme Programming [1], DSDM [31] and Scrum [28]. In 2001, representatives from these and other agile methods joined forces and wrote the Manifesto for Agile Development [2] that describes the common ground of these methods in a simple set of statements and principles.

Experience has been mounting that these agile ways of working, using short iterations and close customer contact, have a higher success rate than traditional methods for software development, at least for many types of software projects. The rigour and volume of research into the effects of agile methods still needs to be improved [9], but recent studies provide theoretical and empirical evidence for the effectiveness of agile methods; see for example the extensive overview and research by Lee and Xia [18].

Agile approaches have also gained the attention of the academic community, who have investigated its foundations and effects from a scientific point of view. A useful definition of agility consistent with the above is given by Qumer and Henderson-Sellers [26]: "Agility is a persistent behaviour or ability of an entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, and uses economical, simple, and quality instruments in a dynamic environment."

We need to address agility at three different levels within enterprises:

- *Agile Enterprises*, which strategically use change to their advantage, outmaneuvering competitors with shorter time-to-market, smarter partnering strategies, lower development costs and higher customer satisfaction.
- *Agile Practices* for design and development, focused on people, rapid value delivery and responsiveness to change.
- *Agile Systems* (both organizational and technical) that are easy to reconfigure, adapt and extend when the need arises.

These different types of agility reinforce each other: if an organization's infrastructure or business processes are more flexible, an iterative and incremental development process can more quickly and easily add value, and the organization's strategy execution is enhanced. The core of this is that *uncertainty* is given an explicit and prominent place. Whereas traditional methods and architectures plan for fixed goals and situations, agile methods and systems are aware of the uncertainties of their environment and know that they are aiming at a moving and often ill-defined target. An integrated approach for the agile development of agile services to serve agile is not yet available. A new perspective on service design processes is needed, providing development teams with the means to tailor their way of working to specific circumstances and deal with multiple stakeholder perspectives, bottom-up innovation and co-evolution of different service aspects. We advocate that agile development processes are much better suited to accommodate these needs than classical linear, top-down design processes, in which individual aspects are often developed separately and sequentially. The iterative character of agile processes, with a focus on people and interactions, close contact with customers and cross-functional teams that tackle different aspects of development at the same time, is a much better fit with the complex and multidimensional nature of service development.

Development processes should also be explicitly focused on observing changes in their environment and acting upon these. The speed of change that organizations have to deal with keeps increasing, and processes must be responsive and even predictive in character to accommodate these changes. These properties should be designed into the development process. Moreover, it should be self-aware, i.e., use mechanisms and practices to observe its own performance and if necessary, change its operation accordingly. This use of reflection is a common characteristic of agile

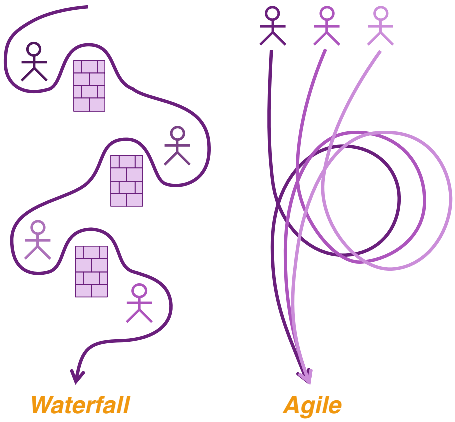

Fig. 1 Waterfall vs. agile processes

methods. Scrum, for example, uses the "sprint retrospective" meeting in which after each iteration, the way of working of the team is evaluated and adapted.

This adaptive character of development processes does not mean that change knows no bounds. The complex nature of service design necessitates the use of sound engineering principles and techniques. External dependencies, technological complexity, regulatory compliance, risk management and other factors all require an approach of bounded or controlled variation. Architecture is a core discipline to provide such managed variation. It specifies the high-level, strategic or otherwise important principles and decisions that together span the design space, like a vector space in algebra.

Another important use of architecture is to explicitly design mechanisms in the operational systems and processes that support change. Not only should development processes be agile and adaptive, but the results they create should also be flexible and amenable to change. Various kinds of design models, ranging from domain, requirements and architecture models to detailed artifacts describing the inner workings of business processes and IT systems, play an important role in both controlling complexity and fostering change. Such models make business knowledge visible across the enterprise, promoting coherence and consistency.

Moreover, a flexible infrastructure that can be configured with such models, instead of laboriously writing software code, may greatly enhance the agility of the organization and its systems. Models can be changed more easily than code, and the effects of changes may be evaluated at the model level before processes and systems are changed, thus avoiding costly errors and re-implementations.

In agile development, the role of these models is not the same as in traditional design processes, however, where specialists each work on their own aspect models and then hand them over to the next person in the design chain. Rather, different models and other artefacts need to be evolved iteratively and in parallel, while guarding their mutual coherence and consistency. The figure above illustrates this idea (Fig. 1).

4.3 Investing in Service Innovation

Organizations with a large installed base of enterprise applications often have poor insight into the relative quality, cost–benefit ratios and risks associated with their application portfolio. As a result, many information systems are maintained far beyond their original technical and business life expectancy, as replacement risks are often overrated in comparison to maintenance costs. For older systems, maintenance costs increase, relative business benefits decrease, and the risk of failure increase over time. Maintenance in fact is a misleading term: bits and bytes do not rust and do not need to be painted or oiled. Most maintenance consists of adding functionality, either to accommodate new business requirements or to integrate with other systems. This additional functionality also needs to be maintained, thereby increasing maintenance costs even further.

Moreover, the complexity of a system increases with its size; more complex systems are harder to change, so each new business requirement becomes more difficult and costly to accommodate than the previous one. This can also be shown empirically: Verhoef [37] Fig. 14, for example, shows a graph depicting that the productivity per additional function point goes down with the system size, and thus maintenance costs go up. Hence, over the entire lifecycle of a system, the initial development costs are only a fraction of the total cost of ownership, and the older the system, the more dominant maintenance costs become. In addition, failure risks of old systems increase and the last remaining people with knowledge of these systems leave, incurring additional costs for dealing with these risks and knowledge gaps.

Furthermore, organizations tend to have an increasingly large IT portfolio, since it turns out to be very difficult to really switch off a system. This may lead to a situation in which the entire IT budget is spent on maintaining old systems, and no budget is left for innovation. In such a situation, the only way out is a significant increase of the IT budget, since renovating, shutting down or replacing old systems also requires an up-front investment.

If this budget is not available, an organization has painted itself in a corner; if a new market entrant comes along that starts from a blank slate with a modern system landscape, with the associated lower cost level, it will outperform and out-compete the incumbent. Over the last years, this is what has happened in many markets, for example with online stockbrokers or new utility companies. Only some government organizations have the "luck" that they do not have to face such competition, but they too come under pressure from an increasingly unfavorable comparison to the private sector.

As a consequence, organizations needs instruments to assess the value of their IT as well as their IT projects with respect to their contribution to both strategic as well as operational targets. Such instruments comprise a portfolio dashboard that indicates the current and future value of applications, and the benefits, costs and risks associated with replacing them.

5 A Framework Supporting Service Engineering

Many different approaches to a staged design methodology exist. Most practitioners agree that waterfall models are inappropriate to handle the complexity in many designs, and iterative or prototyping approaches are needed. Here, we use the methodology to be able to relate different design issues and supporting tools. In service design, different approaches are known, with a clear common denominator. Hansen and Birkinshaw [13] use the concept of the innovation value chain as a simplified representation of the steps in a design process, from creation, through design into diffusion. Osterwalder and Pigneur [23] use a similar line of reasoning. In design thinking [4] the process is stated to be less of importance, but different stages or "spaces" as he calls them, are still distinguished: the inspiration space, ideation space and implementation space. The first space is complementary to the other two approaches. Based on these insights, we have chosen the following stages in design (see Fig. 2).

In different design stages, different aspects of the service network are discussed or refined. At a high-level of abstraction, these are:

- *Creation*: needs, value and opportunities.
- *Analysis*: function, restrictions, robustness, and sustainability.
- *Design*: usability, adaptability, architecture and distribution.
- *Realization*: compliance, reliability and performance.
- *Diffusion*: acceptance, evolution and growth.

In networked service innovation, different actors can be in different stages in the same service design trajectory. For example, when introducing remote care in hospitals, the ICT solution can be in the diffusion stage for the ICT service provider, whereas for the hospital or care-taker, the service is still part of an early

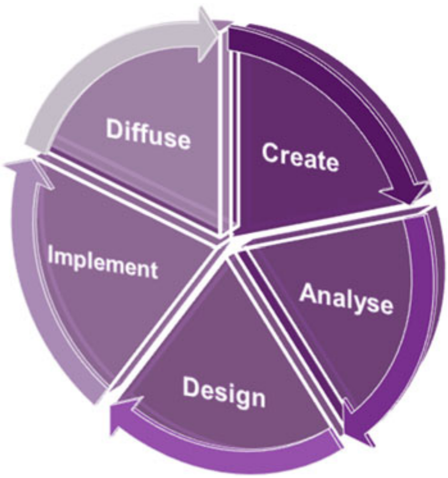

Fig. 2 Stages in service engineering

experimental phase. This can lead to totally different issues that are top of mind to the stakeholders, leading to confusion and misinterpretation.

In different stages of service engineering, we thus have different perspectives, often with different stakeholders. This implies that the techniques used in different stages will different, both in way of working as well as way of thinking. In a creation stage informal, diverging and visual techniques are often employed, whereas in design convergence and formalism is more important. In order to be able to bridge the gaps between the different perspectives, we take an integrated conceptual model as a basis for the techniques.

5.1 Supporting Service Engineering

An integral conceptual model is in itself not very helpful. Only when supported with the right techniques and tools it will fulfill the needs of the engineer. We analyzed the tools and techniques used in a small group of companies with respect to the service engineering phase they applied and the range of concepts they supported.

Without going into detail, we found out that there are a number of clusters of related techniques in different phases, which, albeit related, where hardly integrated. The steps from one cluster to another were not supported. Table 2 summarizes this method and tool scan, where the number in the table indicates how broad the coverage of service aspects is in the tools or method.

It appears that only a limited number of tools and techniques was used in the diffusion phase, and that only few techniques in the later stages could handle financial aspects. Note that this is not a complete overview of techniques and tools available, but a representative set used in practice. The numbers is the table indicate coverage of different aspects of the service to be designed, based on the STOF method (service, technology, organization, and finance) [3, 32].

The next step in designing a service engineering method is to see which typical questions need to be answered in the different service engineering stages and link these to an underlying conceptual and techniques that allow them to be answered correctly. We aim at a joint conceptual framework that covers all aspects of the service network that is being designed. On the basis of this shared language, communication between stakeholders is improved and analysis can become more rigorous. This framework is the topic of the next paragraph.

5.2 Modeling Service Networks or Networked Enterprises

There is no consensus in literature on an integrated conceptual model for networked enterprises. We propose a conceptual model, integrating a number of well-accepted techniques for different perspectives. On the top-level, we follow the terminology of the Cambridge Whitepaper on service networks and service systems [35] where

Table 2 Overview of some tools and methods supporting engineering stages

Tool/method	Creation	Analysis	Design	Realization	Diffusion
STOF	4	4			
Survey	4	4			
Mockup	3	3	3		
E3 value	3	3			
Brainstorming	3				
Scenario-analysis	2	2			
Conjoint-analysis	1	1			
BC-analysis	1				1
Exchange-design		4	4		
COPAFIJTH		4			
ArchiMate		3	3		
Portfolio mgt		3			
Requirements mgt		3			
BiZZdesigner		2	2		
DEMO/Pronto		2	2		
OOAD		2	2		
BPMN		1	1	1	
OTAP			3	3	3
BPEL			2	2	
Design tooling			2	2	
RUP/UML			2	2	
Scrum			1	1	
eMaxx-suite				3	3
Database tooling				2	
Programming tooling				2	
Testen (Tmap)				2	
Best practices maintenance					3
Prince2					3
Grip manager					2
ITIL					1
Total number	8	15	12	10	8
Percentage that stage	25%	47%	38%	31%	25%

a service network consists of related service systems, which can have one or more value propositions associated with them.

For services many different definitions exist: economic, interaction oriented, co-creation oriented. We take a system oriented view and define a service as unit of functionality that (autonomously) delivers value to its environment. A service system is defined as a dynamic configuration of resources that can create and deliver services while balancing risk-taking and value co-creation (Fig. 3).

Service networks, or service system networks are networks of connected service systems that have one or more associated value propositions: specific packages of benefits and solutions that a service system intends to offer and deliver to others (see Fig. 4). Service networks are similar to value systems as used by Porter [25]).

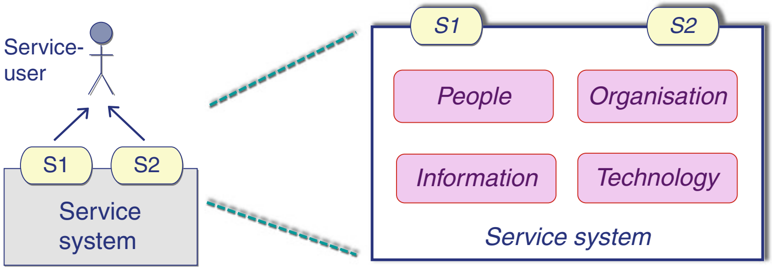

Fig. 3 Service system delivering services to users

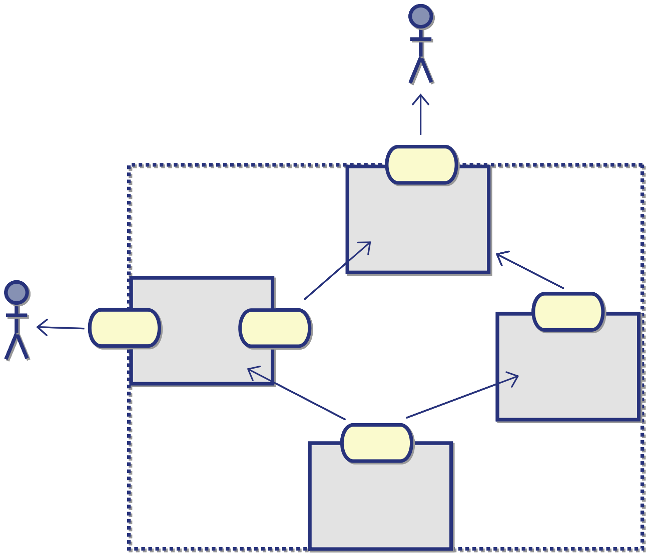

Fig. 4 Service network

The concept is closely related to that of a business model, defined as a blueprint of the way a (network of) organization(s) creates and delivers value for itself and its users through services or products. The business model therefore also refers to the internal structure and behavior of the service system or network.

To be able to analyse and design services and service systems, we need a more refined view. To do so, we employ concepts used in business modeling, such as the STOF method ([3, 32] and Business Model Generation [23]. For the step towards realization, enterprise architecture is the right level of granularity, with ArchiMate as an international standard [14, 33]. Network aspects are the focus in [16].

For the diffusion stage as well as the project aspects of service engineering, we need more than the conceptual description of the service network. We also require better understanding of the way the service network or system changes in time. For

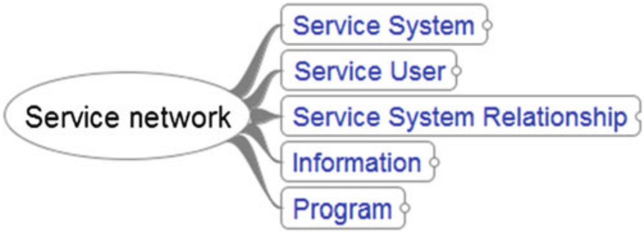

Fig. 5 Top level concepts in service networks

this purpose, ArchiMate 2.0 extensions have been proposed and are in progress of standardization. We briefly sketch the networked enterprise model in more detail.

We already introduced services, service systems and service networks. Besides this, two important aspects are required for service engineering: the underlying information model as well as the program to change or develop the service system from one plateau to another. Figure 5 presents the 5 top level concepts.

Refining the service system, we need concepts like actors, representing organizational entities, and the roles they are responsible for. Related to actors are stakeholders, representing roles of actors with a specific concern with respect to the service system. Service can be grouped into products, associated with an accompanying contract.

The value proposition delivers a certain value, based on value elements, such as speed, accuracy or beauty. To assess the effect of a service design, not only the value proposition is important, but especially the value perceived by the customer.

A service is implemented or realized though business functions, aggregating business processes (Fig. 6). Business functions and processes are supported by applications delivering technical services, consisting of different application components. For example, a claim handling service is supported by a claim handling system, supporting the underlying workflow, with components to assess the correctness of a claim and to make payments to the customer.

Another important aspect in the analysis and design of service networks are the relationships, both between organizations as well as between organizations and their customers (service users). Service systems are connected through channels, where information, goods or revenues can flow. The interaction over those channels is governed by both organizational arrangements as well as financial arrangements. Viewed from a specific direction, a service system/actor can be a partner, a customer or even a competitor.

The full list of terms used to described service networks is given at the end of this section in Fig. 7.

A conceptual framework becomes effective with proper communication and attractive and appealing means. What is perceived as appealing highly depends on the stakeholders involved. The representation used in Business Model Generation [23] has proven to be appealing to a business or management audience. To

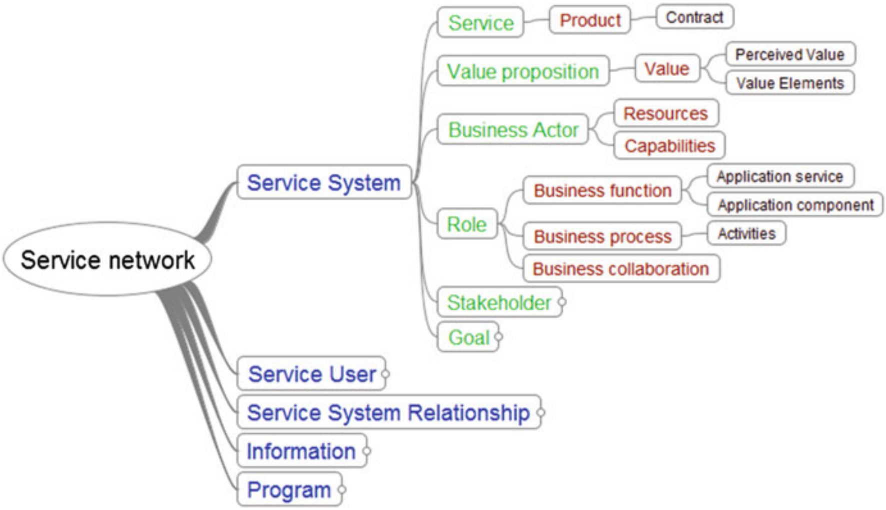

Fig. 6 Refinement of service systems

communicate with enterprise architects, ArchiMate and ArchiMate viewpoint [33] are very useful. System engineers are used to notations as UML to communicate.

All these different perspective must be combined and translated between in a service engineering exercise. Tool support to that end is lacking, still.

6 Conclusions

In this paper we have shown the importance of the service sector to the (European) economy and the challenges organizations face nowadays. These challenges call for a more professional, rigorous approach to developing new services, similar to engineering practices in industry. We identified three major areas of improvement for service engineering: building robust service models, becoming agile, and smart portfolio management, and indicated some of the instruments under development there.

In order for all different elements in service engineering to be tackled equally well, a service engineering process was delineated and we analyzed different tools and methods with respect to their contribution to the different stages in that process. A shared language covering all engineering stages and all aspects of service networks was introduced as a basis for a systematic approach.

To establish a rigorous service engineering approach generating an overview of current tools and approaches is an important step. Our conclusion is that these current tools and approaches do not match the level of engineering we aim for. As a next step, different tools and methods will be mapped onto the service network

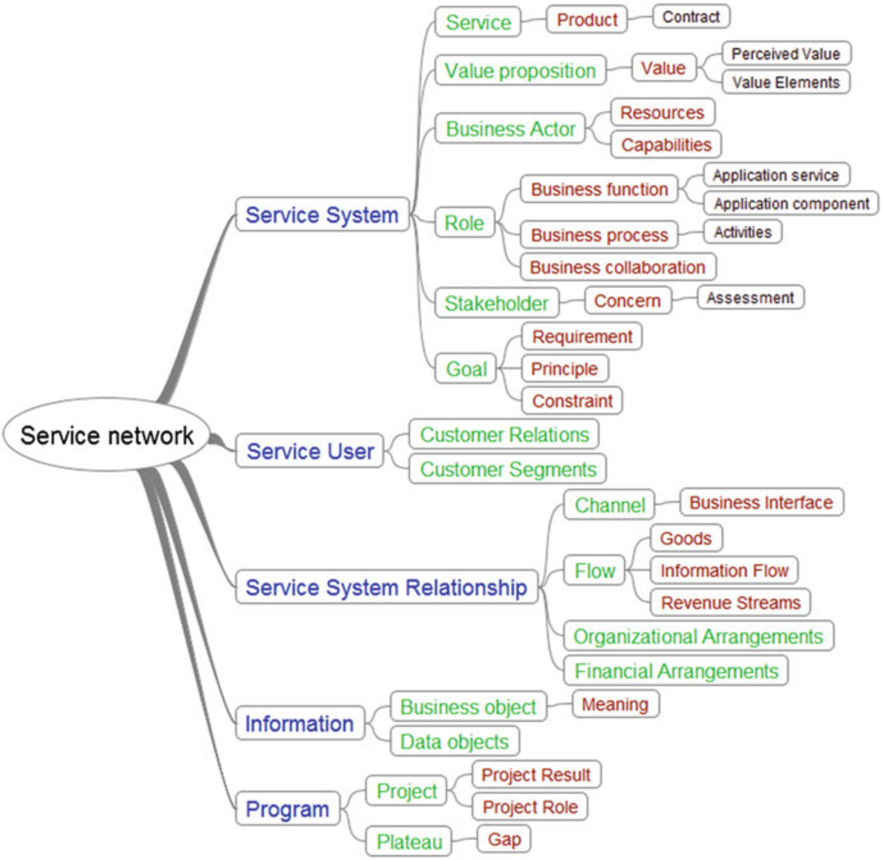

Fig. 7 The service network framework

framework in order to link them in a better way, allowing for information in one design stage to flow to a next stage of engineering. Also, feedback in later stages can then be fed back into earlier design stages in order to refine the model, which is crucial in an iterative approach.

Once different tools and methods have been linked, a coherent and complementary sets of tools and techniques can be chosen, tailored to the application domain or the competences of the organization. At that moment, we finally make the step from service innovation to service engineering.

References

1. Beck, K. (1999). Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston, MA
2. Beck, K. et al. (2001). Manifesto for Agile Software Development. http://www.agilemanifesto.org/
3. Bouwman, H., Haaker, T. and De Vos, H. (2008). Mobile service innovation and business models, Berlin Heidelberg: Springer Verlag
4. Brown, T. (2009). Change by Design. HarperCollins, NY
5. H-J. Bullinger, K-P. Fähnric and T. Meiren (2003). Service Engineering—methodological development of new service products. Int. J. Production Economics 85, pp. 275–287
6. Central Bureau of Statistics (2011). Employment healthcare sector show spectacular growth. http://www.cbs.nl/nl-NL/menu/themas/arbeid-sociale-zekerheid/publicaties/artikelen/archief/2011/2011-3440-wm.htm. Accessed Aug. 1st 2011
7. Chesbrough, H. and R. S. Rosenbloom, (2002). The role of the business model in capturing value from innovation: evidence from Xerox corporation's technology spinoff companies Industrial and Corporate Change (11), pp. 529–555
8. Chesbrough, H. (2003). Open Innovation: the New Imperative for Creating and Profiting from Technology, Harvard Business School Press, ISBN: 9781422102831
9. Dybå, T. and Dingsøyr T. (2008). Empirical studies of agile software development: A systematic review. Information and Software Technology (50:9–10), pp. 833–859
10. European Commission (2010). A Digital Agenda for Europe. COM(2010)245 final/2
11. European Commission, (2010). Europe 2020—A strategy for smart, sustainable and inclusive growth. COM(2010)2020
12. Europe Innova, Innovation in service expert panel (2011). Meeting the challenge of Europe 2020—the transformational power of service innovation
13. Hansen, M. and J. Birkinshaw (2007). The innovation value chain, Harvard Business Review 85 (6), pp. 121–130
14. Iacob, M. E., Jonkers, H., Lankhorst, M. and Proper, E. (2009). ArchiMate 1.0 Specification. Van Haren Publishing
15. Iacocca Institute (1991). 21st Century Manufacturing Enterprise Strategy. An Industry-Led View. Volumes 1 & 2. Iacocca Institute, Bethlehem, PA
16. Janssen, W., and Steen, M. (2001). Rapid Service Development: an integral approach to e-business engineering. In: Web Engineering. Murugesan, Deshpande (eds.). Springer LNCS 2016, pp. 119–132
17. Kleiner, A. (1996). The Age of Heretics. Heroes, Outlaws and the Forerunners of Corporate Change, Doubleday, New York, NY
18. Lee, G. and Xia, W. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. MISQ, 34(1):87–114
19. Martin, J. (1991). Rapid Application Development. Macmillan, New York
20. McKinsey Global Institute (2010). Beyond austerity: A path to economic growth and renewal in Europe
21. Newcom, Survey Innovation Vision. On behalf of Novay. 2009
22. O Reilly, C. A. (2004). The ambidextrous organization. Harvard Business Review, volume 82, pp. 74–81
23. Osterwalder, A. and Y. Pigneur (2010). Business model generation. John Wiley & Sons. Hoboken New Jersey
24. Pateli, A. G. and G. M. Giaglis (2003). A framework for Understanding and Analysing eBusiness Models, 16th Bled Electronic Commerce Conference eTransformation, Bled, Slovenia
25. Porter, M. E. (1985). Competitive Advantage, Ch. 1, pp 11–15. The Free Press, New York
26. Qumer, A. and Henderson-Sellers, B. (2008). An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering. Information and Software Technology 50(4):280–295

27. Sambhamurthy, V., Bharadwadj, A. and Grover, V. (2003). Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms. MIS Quarterly, 27(2):237–263
28. Schwaber, K. and Beedle, M. (2002). Agile Software Development with Scrum. Prentice Hall, New Jersey
29. Schwartz, P. (1991). The Art of the Long View, John Wiley & Sons, New York, NY
30. Spohrer, J., Maglio, P. P., Bailey, J. and Gruhl, D. (2007). Steps Toward a Science of Service Systems. IEEE Computer 40(1):71–77
31. Stapleton, J. (1997). DSDM, Dynamic Systems Development Method: The Method in Practice. Addison-Wesley, Boston, MA
32. STOF, www.stofmethod.com
33. TOG, www.theopengroup.org
34. Treacy, M. and F. Wiersema (1993). Customer intimacy and other value disciplines, Harvard Business Review, January-February
35. University of Cambridge Institute for Manufacturing (IfM) & IBM (2008). Succeeding Through Service Innovation: A Service Perspective for Education, Research, Business and Government, White Paper
36. Vanhaverbeke, W. (2007). The Interorganizational Context of Open Innovation, In: Chesbrough, H., Vanhaverbeke, W. and West, J. (Ed.), Open Innovation Research a New paradigm, Oxford University Press, ISBN: 9780199226467
37. Verhoef, C. (2002). Quantitative IT Portfolio Management. Science of Computer Programming 45(1):12–96

http://www.springer.com/978-1-4614-2325-6

Cloud Computing and Services Science
Ivanov, I.; van Sinderen, M.; Shishkov, B. (Eds.)
2012, XV, 390 p. 127 illus., Hardcover
ISBN: 978-1-4614-2325-6