Please quote as: Li, M. M. & Peters, C. (2018): Reconceptualizing Service Systems – Introducing Service System Graphs. In: International Conference on Information Systems (ICIS). San Francisco, CA, USA.

Reconceptualizing Service Systems – Introducing Service System Graphs

Short Paper

Mahei Manhai Li[1]
[1]University of Kassel
Information Systems
Pfannkuchstr. 1, 34121 Kassel, Germany
mahei.li@uni-kassel.de

Christoph Peters[1,2]
[2]University of St. Gallen
Institute of Information Management
Müller-Friedberg-Strasse 8, 9000
St.Gallen, Switzerland
christoph.peters@unisg.ch

Abstract

Although service systems are a fundamental aspect of service science, its inherent complexity and systems view makes it impossible to model using current approaches. By formalizing service systems, we introduce a conceptual modeling approach for service systems engineers, using formal graph theory, as well as a graphical notation. We term the conceptual model a 'service system graph'. The service system graph enables both a structural perspective and a processual perspective simultaneously. Thus, it enables 1) modeling value co-creation, value propositions, as well as 2) evaluating and simulating different service system configurations. To illustrate our modeling approach, we model an exemplary service system of a software implementation project. We contribute to service science literature by formalizing a mathematical model for service systems. For practice, we provide service systems engineers an approach model and analyze service businesses from a holistic service systems perspective.

Keywords: Service systems engineering, service systems modeling, service systems, conceptual modeling, service science, operational service systems, systems thinking

Introduction

Spanning at least two-thirds of Germany's, Australia's and the United States' GDP, the service industry is becoming increasingly complex and digital (Jaaron and Backhouse 2018; Mrass et al. 2017). From a research perspective, service systems are becoming increasingly important, which can cover entire market ecosystems or single value exchanges (Breidbach and Maglio 2016). Yet, without the proper tools, concepts and models to model service systems, its practical perspective is only limited (Alter 2017) and systems perspective shed only limited light on how to deal with complexity.

In order to address the complexity of systems made of systems, we draw upon hypergraph theory (Berge 1989) to mathematically formalize the concepts of service systems. A service system consists of resources, actors, service objects, and activities. Take the example of a CRM system implementation: an IT-consultant requires software packages, servers, and specific parameters to configure the system to the needs of an organization. The IT-consultant is only able to deliver its service by bringing all the above-mentioned resources together. The systems perspective of *configuring the CRMs* focuses on the **relationship of all required elements** to perform an **activity**.

We conceptualize service system graph (SSG) to model value proposition and value creation while taking both activity, actor and resources into account describe their mathematical notation. Furthermore, this approach enables us to model service systems both from a processual, as well as from a structural relationship-oriented perspective. This enables us to capture the holistic view of service systems.

Our research goal is to develop a modeling approach that describes service systems and enables its analysis. Thus, we base our research on the following research questions: **(RQ1)** *How can service systems be modeled?* **(RQ2)** *How can we analyze and evaluate service system configurations?*

Related Work: Various Service Modeling Approaches

Service engineering approaches can be traced back to both the new service development (NSD) stream of research (Scheuing and Johnson 1989) and the more engineering and software development influenced the stream of research on service engineering (Bullinger and Scheer 2003). Service blueprints (Shostack 1982) were used maintaining a strong focus on the service management and marketing perspective (Edvardsson and Olsson 1996), while service engineering focused more on how to develop new services.

In parallel, a multitude of modeling approaches emerged that address the gap between IT models and business models. Web service design and development addresses the identification of the right services and then organizes them into composite services. This enables a manageable choreography of services to support business processes at a later point. In this vein, IBM developed SOMA, a method to develop a service-oriented architecture (SOA)-based solutions (Arsanjani et al. 2008). SOMA is comprehensive and includes conceptual data models, as well as process-oriented models. SOA-based approaches often focus on the reuse and combination of e-services (Cardoso et al. 2015).

Recombinant service systems engineering is a service system engineering approach that focuses on service innovation. Due to the advent of increasingly complex service systems, (e.g., service systems for smart services), many new services are not designed via top-down approaches. Many new service innovations use already existing resources and solutions by third-party suppliers, thus innovating by recombination of resources (Beverungen et al. 2018).

The service systems perspective differs from the traditional service-perspective. Service systems engineering approaches still rely on a wide variety of existing (semi-) formal modeling techniques (Beverungen et al. 2018; Hoeckmayr and Roth 2017; Peters et al. 2015) such as the diverse set found in UML (Eriksson and Penker 2000), BPMN (OMG 2011), Petri-nets (Salimifard and Wright 2001) and service blueprints (Bitner et al. 2007), to name a few.

Although service systems engineering has become increasingly accepted, the methods utilize existing modeling techniques and do not have a modeling approach that addresses the holistic systems perspective(Beverungen et al. 2018; Böhmann et al. 2014; Leimeister 2015). Common modeling techniques are currently dominated by a process perspective of activities, whereas a structural perspective has not been sufficiently explored. Service systems provide value by reconfiguration of resources and actors holistically (Alter 2017; Böhmann et al. 2014). The process perspective addresses "how", while a structural perspective would address "what" elements are configured.

We follow a different path. In practice, processes always change some form of structural representation, usually data. We argue that by embracing the process and structural duality, new possibilities arise, uncovering a more detailed service innovation approach. We, therefore, differ from modeling techniques by taking a systems perspective and combining both process and data aspects into one modeling approach.

Conceptualizing Service Systems

The I/O model for services is an adaptation from Sampson's service I/O model (Sampson 2010) from manufacturing and its reintroduction by Fromm and Cardoso (2015). All input factors are called resources. Resources can be both human resources, as well as things, which are further categorized into assets and materials. Based on this understanding, service is not the output of a process, but rather the transformation of all input factors into the output factors (Fromm and Cardoso 2015), also known as value proposition (Chandler and Lusch 2015). Our modeling approach can model value propositions. Service is, therefore, a transformation process. We assume that the types of resources depend on the agreed-upon value propositions. Furthermore, we define the input resources as a set of resources, which can have a finite amount of each resource type, such as assets, materials or people etc. Lastly, the output factors are a set with limited elements as output. Naturally, both input and output are not empty.

Complimentary to our service as a transformation process perspective, research on service science regards the **service system** as its basic unit of analysis (Böhmann et al. 2014; Leimeister 2012; Maglio and Spohrer 2008), calling for the adoption of a systems perspective (Böhmann et al. 2014). Constituent factors of service systems are **actors** utilizing operand and operant **resources** (Maglio and Spohrer 2008), whose **activities** describe the "transformation process". An actor can be individuals, teams, organizations cross-organizational business units or even software systems if they mobilize the required resources. This mobilization includes conceptual actors that describe any additional restriction on the resource configuration. Recent research also revisits the importance of **value propositions** and engagement of service systems (Chandler and Lusch 2015), in which organizations seek to find the right constellation of actors ("who"), which enables actors to find the correct resources ("who" and "with what") for a specific context ("when") to co-create value (Chandler and Lusch 2015).

Developing Service Objects and Service System Graphs

Since service systems use resources as input factors, we define a set R with $r \in R$ as all required forms of resources. However, service systems also require actors (Breidbach and Maglio 2016). We define actors as a set A with $a \in A$ representing an actor required for the transformation process.

A service system for a specific value proposition requires both actors and related resources. We called a pair of actor and required resources, **service objects** and define all service objects as a set O with $o \in O$ being a single service object. Formalized, a service object is a tuple of the required resources and the required actors specific to a value proposition. Therefore, service objects are the subject matters of service systems, which are defined in a specific context as input sets of respective outputs. Let $O \neq \emptyset$ be the set of required service objects of any service-driven organization, with $o \in O$ defined as a service object. Thus, a service object is a tuple consisting of resources and actors. Formalized, service objects are defined as follows:

Definition 1:

A finite non-empty set O with tuple of (R, A) is called service object where

i. R is a finite set of resources with $R=\{r_1, r_2 \dots r_n\}$;

ii. A is a family of subset actors of R with $A=(a_i)$ in which $a_i \subset R$ and $R=\bigcup_{i=1}^{n} a_i$ for $i \in \{1,2, \dots ,n\}$.

Definition 1 shows that service object O is a hypergraph (Berge 1989) Therefore, service objects O with tuple (R, A) are hypergraphs of service objects, which represent all inherently possible value propositions of service systems. In other words, the potential of a service system can be realized by reconfiguring its resources and pairing it with a suitable actor. Additionally, Hypergraph theory has extensively focused on its sets of vertices (Bretto 2013) , whereas we put equal importance to its hyperedges. Due to the roles of actors in service science, we inscribe the semantic meaning of actors into hyperedges. A service object includes both actors and resources, both paramount for the realization of the service.

The mathematical form of an element graph is a graph of order=1, that is, $|O_i|=1$ for $i \in \{1, 2, \dots ,n\}$ (Berge 1989). It represents a service object $o_o \in O$ with tuple (a_o,R_o) where $|R_o|=1$. It is apparent that the elementary graph itself has edges. We changed the representation from a solid dot by adding a circle around it to indicate that it also has a hyperedge and hence constitutes a service object and not simply a resource. . We argue that single resources can always be considered as element graphs. However, we recommend only drawing the hyperedge, if it is either an explicit output of a service object or if the element graph itself is a single input (see: application).

If element graphs exist on the hypergraph, for which the relationship of service objects only applies to a single service object, so that the equation $|O_i|=1$ *for* $i \in \{1,2, \dots ,n\}$ holds, each element graph is illustrated as an *elementary object*. The elementary object alone can be the result, as well as intermediate results of any service endeavor. If the above-mentioned elementary object is put together with other service objects, another service system can be configured. This is a deciding characteristic for manufacturing use cases (e.g.: (Hill 1977)), in which outputs are used as inputs for other processes, thus creating a path. We will draw upon the path characteristic shortly when introducing service activities.

Definition 2

O is a finite non-empty set of service object and O is a hypergraph of service objects. A mapping $\psi(\psi^-,\psi^+)$ with $\psi: O \times O \rightarrow Boolean$ where $O \times O \subset 2^O$ is called a service activity of service objects.

Service activities for service objects are represented by the binary mapping between different service objects. One service object is seen as input, whereas the other is seen as output, while the transformation process is the service activity that makes the transition from one service object to another. The mapping ψ is a tuple of (ψ^-, ψ^+), which is a directed or counter-directed mapping of hypergraphs.

In this paper ψ^+ is used for the directed mapping, accompanied by drawing an arrow line. This is not to be confused with directed hypergraphs, which only allows relationships between elements of different hyperedges (Gallo and Scutellà 1998).

Definition 3:

We define a finite non-empty set R of resources, a finite non-empty set A of actors and set O defined as tuple (R, A) as hypergraph of a service object, Ψ set of value creation functions as service activity, then the tuple SSG(R, A, Ψ) is called the service system graph, representing the service system; The value creation function is defined as follows:(i) Ψ: $\Psi(O) \rightarrow O$ with $\bigcup_{i=1}^{n} \psi_i(o) = O_{output}$, where $o \in O_{input} \subset O$ and $O_{output} \subset O$. A service system graph is a directed graph, which models the value creation and value propositions of a chain of services. The service system is a family of subset service objects under value creation functions . Thus, strictly speaking, a single service object itself is also a service system.

Proposition 1: Service objects can be shared for different value propositions. **Proof:** According to the definition of SSG$\langle R, A, \Psi \rangle$ and function $\Psi(\Psi^-, \Psi^+)$, we said a service object is a multi-required service object if $\bigcap_{i=1}^{n} \psi_i(o) \neq \emptyset$ where $\psi \in \Psi^-$ and $\exists o \in O$, $n \geqslant 2$.

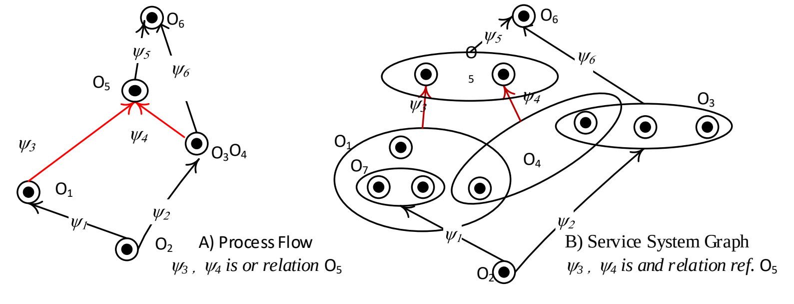
Proposition 2: Service objects can be provisioned by different service objects. **Proof:** One service object can be delivered by more than one activity. According to definition 3 of SSG(R, A, Ψ), $\exists \Psi (\Psi^-, \Psi^+)$, a service object is called a multi-delivered service object when $\bigcap_{i=1}^{n} \psi_i(o) \neq \emptyset$, where $\psi \in \Psi^+$ and $\exists o \in O$, $n \geqslant 2$. **Proposition 3:** Service objects can be provisioned by service objects and simultaneously be a service object, which provisions another service object. **Proof:** Based on service system graph SSG(R, A, Ψ) and the service object O, subset of activities $\Psi_{after} \subset \Psi$ and $\Psi_{before} \subset \Psi$ with $\Psi_{before} = \{\bigcup_{i=1}^{n} \psi i \mid \bigcap_{i=1}^{n} \psi_i(o) \neq 0$ where $\psi\psi \in \Psi^+$ and $\exists o \in O\}$, $\Psi_{after} = \{\bigcup_{i=1}^{n} \psi i \mid \bigcap_{i=1}^{n} \psi_i(o) \neq 0$ where $\psi \in \Psi^-$ and $\exists o \in O\}$, then $\forall \psi_1 \in \Psi_{after}$ follows $\forall \psi_2 \in \Psi_{before}$.

Application

Up until now, we have only conceptualized service systems as SSGs. The following addresses the wide range of SSG application value. SSG is a modeling approach developed for service businesses. In addition to meeting the specific functions of the service business, it should have other basic functions of modeling such as for modeling value proposition, co-production by service. A modeling approach often has two basic functions: building a model from the data level and the process level. Data modeling requires a clear description of the characteristics of the data and the relationship between the data. A typical representative is the BOM (Bill of Material) modelling method. The process level describes how data is processed and the direction of its logical flow under acting from functions. One wide-spread method is BPMN.

SSG also models data level structures. Referring to BOM, we know that it expresses the data relationships in tree form. Based on graph theory, trees are a type of simple graphs (Berge 1989). Therefore, hypergraphs are networks, whereas the tree is the result of a hypergraph projection. Thus, SSGs can describe both a network structure, as well as tree structure.

To model service systems, we combined both aspects and model both data and processes. To illustrate the difference between process models and SSG, we refer to Figure 1. Figure A) shows the process flow using a simple graph, in which both states and activities are depicted. Comparing to B) it is apparent that SSG can show richer information and can represent a greater level of detail. Each business process corresponds to a function that transforms input into output factors. These factors are implicit in A). In other words, in A) data structure is out of scope. Yet, as B) shows, SSG also shows data structures, something that process-oriented models do not.

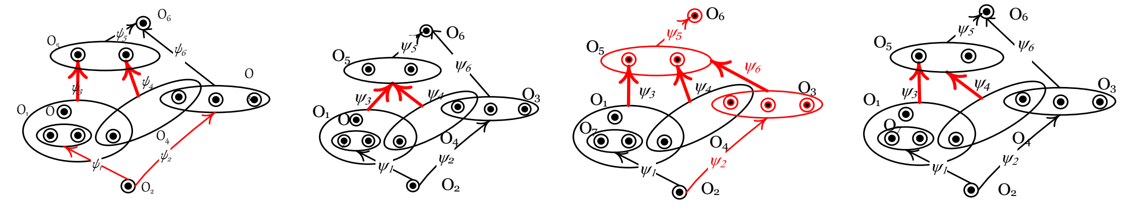


A) Process Flow
ψ_3, ψ_4 is or relation O_5

B) Service System Graph
ψ_3, ψ_4 is and relation ref. O_5

Figure 1 - Comparing Process Modeling with SSG

SSG's applicability is thus very broad and differs from process view models. Based on definition 2.3 a SSG describes combinations of any resources of a finite set, including the relationships between configurations (all functional relations). It represents both highly complex and diverse situations. Similarly, BPMN has evolved over the years. Based on the increasing complexity and changing industry demands, it adapted their "notation" and included data repositories. However, process modeling is limited by its functional one-sidedness. In contrast, SSG includes both data and process views, thus increasing its area of application and opening new potentials for a holistic and systemic modeling approach.



(i) ψ_3, ψ_4 is **and relation**

(ii) ψ_3, ψ_4 is **or relation**

(iii) ψ_3, ψ_4, ψ_6 is ***and/or relation***

(iv) ψ_3, ψ_4 is ***and/or relation***

Figure 2 Comparison to Process Operations

Figure 2 shows how some seemingly similar service systems can have very different meanings. The service systems (i) – (iv) differences lie in their activities marked in red, each representing operations usually seen in process-based modeling approaches. SSG provides its users with versatile modeling options, giving greater detail of modeling relations, as shown in (iii) and (iv).

Furthermore, process models that based on simple graphs are a special form of our SSG. When a service system only includes service objects that have no more than one resource, the result is a hypergraph that represents the structure of a simple graph. In other words, hypergraphs can be reduced to process models if for all $o \in O$, $|o| = 1$ is valid.

SSG also enables novel ways to **analyze service systems holistically**. Our vision is that a user can not only use SSG to model the as-is situation of service businesses, restructure it and innovate the business. There are times when you need to select a **"path"** in an existing business context based on the required resources. Analyses can consider time or other types of cost factors. This enables the users to make decisions based on all required resources and unlock new forms of analyses, as well as existing analyses that are reliant on simple graph structures. Figure 2 (iii) shows how a service system can include different service systems representing paths. One such path has been marked read and includes ψ_2, ψ_6, $\psi_{5\ \text{path}}$. The decision maker needs to decide which path to choose.

This decision needs to be based on **analyses and evaluating alternative service system configurations**. Based on our definitions, we know that Ψ represents activities. Yet it also implies that it can be seen as a connection from one hypergraph to another hypergraph. In other words, one can see it as two sets being transformed by functions. So, it can not only represent the existence of relationships, show the direction, sequences, and paths, but more importantly, it can be used for computational operations. If an SSG is used for computing sets of input factors being transformed into a set of output factors, it can be used for simulations and provide quantifiable analyses. For this purpose, definition 3 introduced Ψ as ***value creation function***. This can be used for various purposes, such as evaluating existing service systems, comparing similar ones, reconfiguring service systems and service innovations in general. For SSG modeling, it is therefore not only advantageous to combine data and process view but more importantly, it also enables mapping activities with functions. The significance of this lies in bringing the service system model from a purely descriptive planning application into the realm of service operations.

SSG can **model value (co-)creation**, which is a characteristic specific of services. SSG has been specifically conceptualized for service systems. Therefore, typical concepts of service systems need to be addressed as well. Value (co-)creation differs by including activities since the value is only realized after actors act upon something to co-create value (Vargo and Lusch 2008). Before that activity, the non-realized value is sometimes called the potential or value proposition. In our approach, we use the combination of both service objects (sets of resources and actors) and activities to describe value (co-)creation, using its value creation function Ψ. We argue that this form of representation is both rigorous and intuitive since value creation can only happen with both the required service objects and the required activity. Co-creation is illustrated as all service objects and required activities for a given service object output. Value proposition is thus only the "last" activity and service object that leads to the result. Both value (co-)creation and proposition are the reason for SSG being a **service system specific** modeling approach.

Demonstration Case: CRM Implementation

For the purposes of demonstrating how SSG modeling is useful, this chapter introduces a simple "CRM implementation" project as an example case. Based on our research in software implementation, we assume five key services as necessary aspects for accomplishing a successful implementation: (1) someone needs to setup the CRM software system on any form of hardware; (2) users require tailored training; (3) conceptual work and configuration, including testing is essential; (4) user support has to be available to users; (5) expert business unit members need to contribute their critical domain-knowledge. Using the simplified CRM case, we will illustrate how innovation mechanisms are essential for SSG modeling and how combining process with data structure modeling views uncovers new analysis possibilities.

ITCon is a total solution provider for CRM systems, who is currently working for PowCo, a mid-sized service provider. PowCo is still struggling to decide whether they should outsource the entire project to ITCo or whether some parts can be done by their other long-term key partners (Li et al. 2017). They tasked a service system engineer to assess possible options and to reduce complexity and analyze the situation and apply SSG modeling:

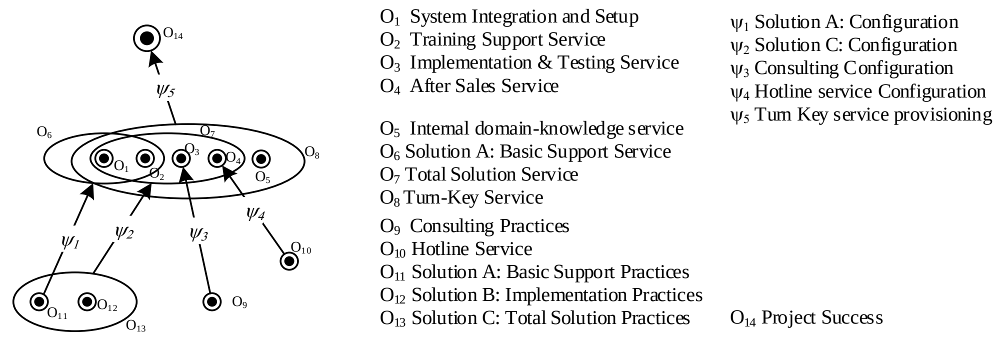


Figure 3 - CRM Project overview (Object-View)

We start by modeling the five key service objects (1) – (5) in Figure 3. Service Objects O_1-O_5 are all required for a successful CRM project. We call the final service turn-key service, metaphorically giving the keys to a CRM solution to the board. However, PowCo can decide how it wants the project to be done. The CRM project has been modeled as a service system. After careful inspection, the model shows two different configurations to realize the project. ITCon can provide a total solution to their customer by configuring O_{13}, coupled with a team placed by their customer (O_5). Alternatively, ITCon can rely on basic support practices, such as system integration and setup, as well as training services and include other parties to provide conceptual consulting practices (O_9) and helpdesk hotline services (O_{10}). Both service system graphs are feasible and reasonable.

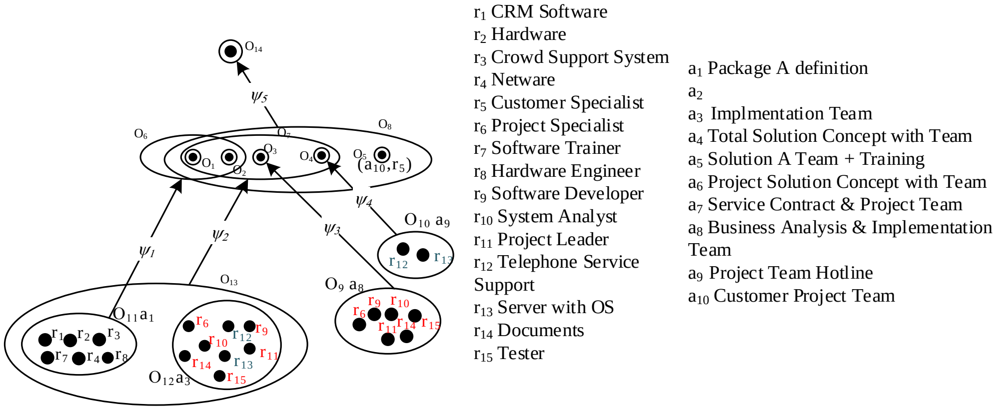


Figure 4 – From Object to Resource View

Next, the SSG zooms from an object level into a resource-based view, as illustrated in Figure 4. By comparing the resources of O_{12} with the resources of both O_9 and O_{10}, we see that they share resources. We can thus subsume O_9 and O_{10} into O_{12}. Graph (iii) of Figure 5 shows the resulting SSG, relying on the recombination mechanism. This optimized system provides PowCo with a blueprint if they decide to insource the software implementation project completely and show them what resources they require.

Furthermore, we apply both the system of system and dissociation mechanisms to split the previously mentioned two alternative ITCon paths from one service system into two different service systems, as depicted in (i) and (ii) in Figure 5. This helps in comparing and analyzing both options and simultaneously represents a service (system) portfolio for ITCon.

For simulation-based analysis, different functions can be attached to Ψ. For example: to assess simple costs of (i) and (ii), one can add total costs of all activities for (i) and compare it to the total of (ii). The individual cost structure relies on each activity. The activity uses a function that depends on both the actor and the required resources. The actor is essential, since depending on the actor, they might require fewer resources (e.g.: learning curve).

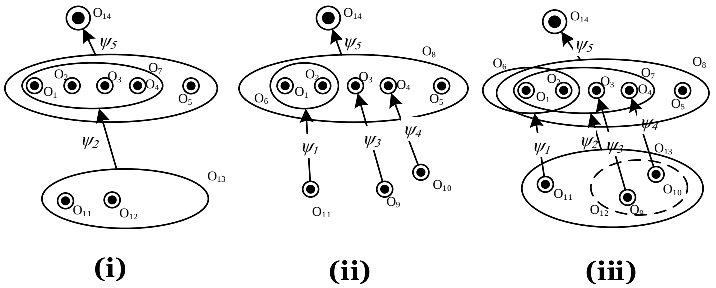


Figure 5 - Service System Reconfiguration

Discussion and Future Work

We proposed SSG as a novel approach to model and analyze service systems. From a practical perspective, our approach enables businesses to both model traditional manufacturing I/O models, as well as a service transformation process with one modeling technique. This coincides with SDL and servitization approaches (Vargo and Lusch 2008).

SSG is the foundation for understanding and analyzing service-oriented businesses. If combined with ERP systems or other enterprise systems, SSG can further research on how to systematically model service businesses. Based on our model, it is possible to run a variety of business-specific applications. In this paper, we have already shown the potential for analyzing time- and cost-based service systems, yet technically any performance indicator is implementable. These applications can, therefore, rely on the mathematical model and resulting data and functional structure to run various simulations, using a set of value creation functions. Furthermore, SSG can be used both for planning service systems and service operations and the introduction of service objects helps to structure the inherent complexity of service systems. SSG is therefore both a model to define service systems ("as-is"), as well as a modeling tool to work with service systems and its composites ("to-be"). Additionally, value propositions are represented by the value creation function, whereas its output is its realized value.

SSG is a formal approach based on a mathematical model to model service systems. Validation of its mathematical model's completeness and correctness is therefore inherent in its mathematical underpinnings. However, validation of its service innovation application has been shown, whereas the other presented application possibilities are still subject to future research.

Furthermore, it comes from a similar motivation as service system axioms, with some commonalities (Alter 2017). However, we limit ourselves on service systems for service businesses and do not encompass all forms of service systems, while following a formal approach to structure service systems. This also determines our research limitation, as we do not focus on social aspects, such as power plays and power struggles in our SSG model. Still, we believe that we contribute by formalizing a specific service system understanding, which is valuable to both practitioners and service researchers alike.

Conclusion

SSG is a novel approach to both understanding service systems and modeling service system. We first define a concept of service systems and apply hypergraph theory to formalize our service system concept. Based on this mathematical model, also known as service system model and its definitions and proven propositions, we present its application possibilities. This includes being able to value (co-)creation and how to apply functions to service system configurations to analyze and evaluate different SSG configurations. This paper concludes with a demonstration case of an IS-typical service example: a software implementation project.

Additionally, the increased modeling information can both be applied to optimization and service innovation analyses. It assists in coping with reducing the complexity of service systems using service objects and establishes an intuitive representation of service business and how they "work together" to form a service system. Furthermore, we can model service-specific concepts, such as value co-creation, value propositions. Finally, it is possible to simulate service business scenarios, which we call configuration or paths, to provide a basis for decision makings regarding service innovation and reorganizing organizations. The functional perspective on service systems enables different forms of operations, which are needed for the sought-after simulations using value creation function. All the above are subject to future research. Additionally, future endeavors should encompass the development of software-aided modeling tools, which could demonstrate its potential much more explicitly.

We argue that all service businesses can apply SSG as an approach to redefine their organization in terms of service systems. This also opens a multitude of entirely new areas of research, including resource planning, service operations or modularization, to name a few.

In conclusion, our paper introduces both a mathematical model for service systems and a new modeling approach contributing to the body of knowledge on service science in general and modeling of service systems specifically. In terms of practical contribution, we demonstrate a few potentials on how the service systems perspective can support a systematic view on service businesses and even be used for analyzing existing businesses as service systems.

References

Alter, S. 2017. "Service System Axioms that Accept Positive and Negative Outcomes and Impacts of Service Systems," *International Conference on Information Systems* (38), pp. 1–21.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., and Holley, K. 2008. "SOMA: A method for developing service-oriented solutions," *IBM Systems Journal* (47:3), pp. 377–396.

Berge, C. (ed.) 1989. *Hypergraph*: *Combinatorics of Finite Sets:* Elsevier.

Beverungen, D., Lüttenberg, H., and Wolf, V. 2018. "Recombinant Service Systems Engineering," *Business & Information Systems Engineering* (21:1), p. 50.

Bitner, M. J., Ostrom, A. L., and Morgan, F. N. 2007. "Service Blueprinting: A Practical Technique for Service Innovation," *California Management Review* .

Böhmann, T., Leimeister, J. M., and Möslein, K. 2014. "Service Systems Engineering," *Business & Information Systems Engineering* (6:2), pp. 73–79.

Breidbach, C. F., and Maglio, P. P. 2016. "Technology-enabled value co-creation: An empirical analysis of actors, resources, and practices," *Industrial Marketing Management* (56), pp. 73–85.

Bretto, A. 2013. *Hypergraph Theory*, Heidelberg: Springer International Publishing.

Bullinger, H.-J., and Scheer, A.-W. (eds.) 2003. *Service Engineering*: *Entwicklung und Gestaltung innovativer Dienstleistungen*, Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg.

Cardoso, J., Fromm, H., Nickel, S., Satzger, G., Studer, R., and Weinhardt, C. (eds.) 2015. *Fundamentals of Service Systems*, Cham, s.l.: Springer International Publishing.

Chandler, J. D., and Lusch, R. F. 2015. "Service Systems: A Broadened Framework and Research Agenda on Value Propositions, Engagement, and Service Experience," *Journal of Service Research* (18:1), pp. 6–22.

Edvardsson, B., and Olsson, J. 1996. "Key Concepts for New Service Development," *The Service Industries Journal* (16:2), pp. 140–164.

Eriksson, H.-E., and Penker, M. 2000. *Business modeling with UML*: *Business patterns at work*, New York: Wiley.

Fromm, H., and Cardoso, J. 2015. "Foundations," in *Fundamentals of Service Systems*, J. Cardoso, H. Fromm, S. Nickel, G. Satzger, R. Studer and C. Weinhardt (eds.), Cham, s.l.: Springer International Publishing, pp. 1–32.

Gallo, G., and Scutellà, M. G. 1998. "Directed hypergraphs as a modelling paradigm," *Decisions in Economics and Finance* (21:1-2), pp. 97–123.

Hill, T. P. 1977. "On Goods and Services," *Review of Income and Wealth* (23:4), pp. 315–338.

Hoeckmayr, B. S., and Roth, A. 2017. "Design of a Method for Service Systems Engineering in the Digital Age," *International Conference on Information Systems* (38), pp. 1–23.

Jaaron, A. A. M., and Backhouse, C. J. 2018. "Operationalisation of service innovation: a systems thinking approach," *The Service Industries Journal* (38:9-10), pp. 561–583.

Leimeister, J. M. 2012. *Dienstleistungsengineering und -management*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Leimeister, J. M. 2015. *Einführung in die Wirtschaftsinformatik*, Berlin, Heidelberg: Springer Gabler.

Li, M. M., Peters, C., and Leimeister, J. M. 2017. "Designing a peer-based support system to support shakedown.," *International Conference on Information Systems* , pp. 1–13.

Maglio, P. P., and Spohrer, J. 2008. "Fundamentals of service science," *Journal of the Academy of Marketing Science* (36:1), pp. 18–20.

Mrass, V., Li, M. M., and Peters, C. 2017. "Towards a taxonomy of digital work," *European Conference on Information Systems (ECIS)* (25), pp. 2515–2524.

OMG 2011. "Business Process Model And Notation 2.0,"

Peters, C., Blohm, I., and Leimeister, J. M. 2015. "Anatomy of successful business models for complex services: Insights from the telemedicine field," *Journal of management information systems : JMIS* (32:3), pp. 75–104.

Salimifard, K., and Wright, M. 2001. "Petri net-based modelling of workflow systems: An overview," *European Journal of Operational Research* (134:3), pp. 664–676.

Sampson, S. E. 2010. "The Unified Service Theory," in *Handbook of Service Science*, P. P. Maglio, C. A. Kieliszewski and J. C. Spohrer (eds.), Boston, MA: Springer US, pp. 107–131.

Scheuing, E. E., and Johnson, E. M. 1989. "A Proposed Model for New Service Development," *Journal of Services Marketing* (3:2), pp. 25–34.

Shostack, L. G. 1982. "How to Design a Service," *European Journal of Marketing* (16:1), pp. 49–63.

Vargo, S. L., and Lusch, R. F. 2008. "Service-dominant logic: Continuing the evolution," *Journal of the Academy of Marketing Science* (36:1), pp. 1–10.