Moving towards a Non-Dyadic View on Service Systems and its Operationalization – Applying the Hypergraph-based Service System Model

Mahei Manhai Li
University of Kassel
mahei.li@uni-kassel.de

Christoph Peters
University of St.Gallen
christoph.peters@unisg.ch

Jan Marco Leimeister
University of St.Gallen
janmarco.leimeister@unisg.ch

Abstract

In today's VUCA world, that is characterized by high volatility, uncertainty, complexity and ambiguity, service provisioning is required to realize flexible and adaptable reconfiguration of service delivery systems and its stakeholders' resources. However, services are often embedded in greater service systems and the context information of both customer and service provider form both its boundary conditions the suitable solution service. To capture the complexity and leverage the dynamic of service systems, we propose the formal service system model (SSM) method. Following general systems theory, we define boundaries for service delivery and show SSM's applicability for ad-hoc service operations. We show its usefulness for structuring a service system for service operations, specifically scheduling, planning, and pricing of service provisioning.

We contribute to service systems engineering by applying one generalizable mathematical model for both structuring and operationalizing service systems and provide insights in-to capturing the complex relationships of its components

1. Introduction

We propose a model for service operations, which allows dynamic solutions depending on the customer context. This is important because the perspective of service systems can be applied to service operations, thus operationalizing service systems [1].

Nowadays, due to the rapidly increasing numbers and complexity of service de-mands, service providers need to adapt to customer demands and contextual circum-stances even faster. They need to dynamically respond to external demands and internal conditions. To accommodate this challenge, a company's service system needs to be able to dynamically reconfigure its required resources based on the con-text of both service providers and the customer [2]. The chosen service system con-figuration is called a dynamic solution.

The different possibilities of how a service can be realized reflect the potential of any business seen as a service system. Each possibility should be part of a model that can be used dynamically to identify a suitable service configuration. This ad-hoc reconfiguration requires a service architecture that utilizes resource mobilization and incorporates the process perspective of service exchange [2, 3]. This leads to the requirement of a powerful foundational mathematical model. Our research follows the service system model as core model to define, understand and model service systems blinded [4] and follow the basic concept of service systems as it is represented in service systems engineering [2].

Our research question is therefore twofold: RQ1: How can the service system model be used to operationalize and quantify service system configurations? Specifically, we look at one exemplary characteristic of service systems in this paper and thus formulate following question: RQ2: How can we apply the service system model (SSM) for service systems that are susceptible to dynamic reconfiguration?

We demonstrate our model using a real-life citizen-based produce delivery service. This paper focuses on the application SSM for finding dynamic solutions and is structured as follows: First we briefly introduce the service system model by explaining our understanding of service, service system and its formal definition. Next, we explain what a service system configuration entails, while introducing the running case of FreshnessDeliverd (FD), an innovative service for citizen-based produce deliver. Then, we use the example to explain the dynamic characteristic of service systems and show the advantages of SSM by operationalizing a service system and finding an optimal solution, while concluding with a discussion on our contributions, future work and a short conclusion.

2. Related Work

Model-based management, planning and control from a systems perspective has long been the domain of operations management [1]. Manufacturing systems used flow-based process models for understanding how to produce a product, and component-based models for modeling what is required for the production of a product [2]. From a manufacturing system engineering, only by using both types of models for additional model-based approaches, design, plan and control of such manufacturing systems became manageable [1–4].

With the advent of service science [5–7], researchers became interested in the new unit of analysis, service systems. In parallel, service engineering developed [8], culminating in more recent research of service systems engineering [9–12]. To model service systems, a plateau of approaches are usually employed [12] and continue the approach of both structuring how something is achieved (process) and with what it is to be achieved (structural). These include formal approaches, for example petri-graph approaches [13], as well as semi-formal and more domain-specific approaches, such as business process model and notation (BPMN) [14] and service blueprinting [15]. Still, they require a structural perspective to be operationally feasible, such as the entity relationship model [16,17].

However, we propose to use SSM, a new model based on hypergraph theory, which enables to integrate both the process and data structural information that is inhering in service systems inside one single formal model [18]. This paper leverages this dual perspective of SSM as an underlying model to operationalize service systems and apply it for dynamically configuring services.

3. Service System Model

3.1 Service System Model

The I/O model for services is an adaptation from Sampson's service I/O model [19] from manufacturing and its reintroduction by 20 [20]. All input factors are called resources. Resources can be both human resources, as well as things, which are further categorized into assets and materials.

In traditional goods-dominant logic, a typical manufacturing process consists of resources that are transformed, such as raw materials, plus factors, which are needed for the transformation but are not transformed by itself, such as tools and workers. Depending on what is to be transformed, the resources that are transformed could be either physical attributes of the resource (e.g. manufacturing) or transformed by moving the resources to other locations (e.g., logistic). Furthermore, a transformation of human resources can

be through physical changes (e.g.: healthcare) and also through changes of the mind (e.g. mobile learning and entertainment services) [20]. In each example, the conditions of certain resources were transformed, while some resources were not affected by the transformation (Hill 1977). One example is mobile learning services, in which the learner, who learns by using the app, changes its mental state, and thus acquires new knowledge. The app itself was only needed to learn, yet the transformation process did not affect the app itself. We argue that operant resources are also input factors for a transformation process. We acknowledge the operand and operant resource perspective and integrate both types of resources into our understanding of service as the transformation of all input factors into the output factors [20], also known as value proposition [21].

Additionally, the value of a service is only realized during its use, also referred to as value-in-use [7]. By integrating operand and operant resources, actors transform all required resources to realize the before-promised value proposition. Since different actors are part of the service transformation and the value proposition is agreed-up by and between at least two actors, the realization of value is also referred to as (co-) creation of value [22].

3.2 Service System

Complimentary to our service as a transformation process perspective, research on service science regards the service system as its basic unit of analysis [9,23,24], calling for the adoption of a systems perspective [9]. Constituent factors of service systems are **actors** utilizing operand and operant **resources** [23], 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.

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.

Recent research also revisits the importance of **value propositions** and engagement of service systems [21], 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 [21]. Since a service system includes different types of resources and actors, who create value to a customer, we

define the term service objects that pairs corresponding resources and actors, which addresses a value proposition. Since services at its core have value propositions, which are comparable to promises made to customers, customer-side, realizing the value proposition is imperative. From the service-provider perspective, it is the constellation of resources that actors require, that is imperative.

Currently, the complexity of service systems arises, since any system can consist of several subsystems. This recursive or nested system of systems principle has its origins in general systems theory [25].

The seven constituent elements of service systems are the basis upon which this paper defines service systems: A service system is defined as a configuration of, resources, actors and service activities [5]. Additionally, to cope with increasing complexity, we have introduced a combination of resources and actors into a single unit, which we term service objects. Next chapter leads with a more detailed description of each element:

Resources include all operant and operand resources [7], which include both material things, such as screws and money and immaterial ones, such as knowledge or systems. They are either acted upon or are used to act with [26]. In other words, it represents all input that is needed that enables an activity.

Actors can be persons, business units, organizations in general or any other form of agents that acts upon or uses the resources, thus mobilizing the resources for a specific purpose. This includes software systems as actors. For our purpose, we also allow conceptual actors that have the role of describing any additional restriction on resource configurations.

Service Objects are pairs of an actor and all the resources he requires to perform an activity. These objects are inspired by objects from object-oriented programming, which were also introduced to aggregate common functionalities into conceptual objects to create more structure into their code and thus reduce complexity [27].

Service Activities are required to transform the resources using the help of actors [28]. In other words, service activities require service objects as input. The output of service activities is in the form of transformed service objects.

Value creation is the entirety of transformations and respective service objects that are needed for realizing the value to a customer. If many actors are involved in the value creation process, we refer to it as value co-creation. **Value propositions** are what companies promise their customers [28]. To realize the value propositions, the value must be created. The value proposition represents the perceived value from a customer point of view. How value is co-created beyond what the customer perceives is not part of the value proposition.

3.2 Service System Model definition

The service system model relies on the underlying mathematical service system graph (SSG) [29]. The service system graph extends hypergraph theory by introducing a mapping between different hypergraphs that allows modeling a processual perspective in and resource-driven data perspective. Applying the service system graph to the above-mentioned service system concept we create a service system model. into the following constituent factors, which we will briefly outline and have been established in previous work [29]. Our understanding of service systems takes on a set theory conform perspective, in which we abstract the resources into sets and model the relationships between different resource sets. Additionally, our perspective captures the different combinations of these sets and how this structure is applied to the concept of a service and its corresponding service system.

In other words, by structuring these characteristics, we identified a suitable mathematical model to represent this set theory and combinatorial approach to services via hypergraphs. Extending hypergraph theory by simply drawing from the input-output model [26] and the notion of mapping, the interrelations of the graph elements can also be captured. This enables the modeling of both service objects and their relationship via activities and enables service specific concepts, such as value proposition and value co-creation. The following sections provide a formal definition of the above-mentioned terms.

Definition 1 – Service Object: A finite non-empty set O with tuple of (R, A) is called service object where R is a finite set of resources with $R=\{r_1, r_2 \dots r_n\}$ and A is a family of subset actors of R with A=(ai) in which $a_i \subset R$ and $R= \bigcup_{i=1}^{n} a_i$ for $i \in \{1,2, \dots ,n\}$. Also, a recursive relationship is possible because a service object can be a resource.

Definition 2 - Service Activity: 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.

Definition 3 – Service System Graph: 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) $\Psi: \Psi(O) \rightarrow O$ with $\bigcup_{i=1}^{n} \psi_i(o) = O_{output}$, where $o \in O_{input} \subset O$ and $O_{output} \subset O$ and $\exists o \in O \ \psi^-(o) \cap \psi^+(o) = \emptyset$ and Ψ^* be called associated function with:

(ii) $\Psi^* = \Psi^K \circ \Psi^L$ where $\Psi^*, \Psi^K, \Psi^L \subset \Psi$. The element function $\psi \in \Psi$ coupled with a service object is called value proposition with:

(iii) $\psi(o) = o'$, where $o \in O_{input}$ and $O_{input} \subset O$, $o' \in O_{output}$ and $O_{output} \subset O$, $\psi \in \Psi$.

Function $\Psi^-(O)$ defines which service objects are required as input factors and function $\Psi^+(O)$ defines the output service objects. ψ^* defines the association between two activities. The service system is a family of subset service objects [18,30]. Thus, strictly speaking, a single service object itself is also a service system. A more thorough definition can be found at [18].

4. Service provisioning as systems configuration

4.1 Example case: Citizen-based Produce-delivery Service

To describe service systems, we model a real-life scenario of an innovative service, which has been developed by three service engineers. They address the need for ad-hoc delivery of locally grown fresh produce.

Mid-sized German cities are often surrounded by farmers. Furthermore, many citizens commute to and from work, passing by farms. Therefore, the idea of FreshnessDelivered (FD) was born. The service they provide is both selling fresh produce and delivering it to the customer. In addition to just farms, even urban gardens and hobby gardening enthusiasts can also offer their produce using FreshnessDelivered. To accommodate the buying process and the delivery, FD has a network of potential sellers and deliverers, which is our service system.

For our example we assume that FD only supplies eggs and apples from local citizens and farmers. The simplified service system of FD includes three main actors: one local farm that produces both eggs and apples, one local apple yard owner and one local chicken egg enthusiast, an elderly citizen who lives in a rather big mansion and is reliving the past and decided to have 10 egg-laying hens. The local farm's main concern is selling its eggs, since they are convinced that the quality of their product speaks for itself. To increase sales, they include another byproduct, apples, as bundles and sells it for a special price as bundles. The resulting FD service system is illustrated in Fig. 1. Furthermore, the service request was made at 2 p.m. and has to be delivered to the pick-up spot not later than 5 p.m.,

meaning that the customer requires the service to be delivered within the next 3 hours.
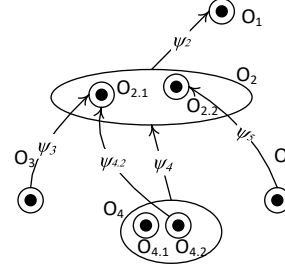

Fig. 1 Service System of FreshnessDelivered (local eggs and apples)

We start with the simulated case of a customer ordering different fresh produce: two apples and five eggs. Once the order request gets to FD, they will run it against the existing service system model. The parameters of the model are illustrated in Table 1.

As a next step, both the model and the table are required so that the service provider FD can decide which suppliers to take from. Using the service system perspective, one might also say that FD needs to decide how to configure its service system to provide a reasonable service to the customer. To further explore this, we explain that service system configurations are fundamental for understanding dynamic solutions.

Table 1. FreshnessDelivered Service System Parameters

Servcice Object	Description	Delivery time in h	Unit Cost [1]	Unit Cost [10]	Unit Cost [100]
O_3	Apple yard	1	1	9	85
O_5	Chicken egg enthusiast	6	0,35	2,5	25
O_4	General Farm (special)	1	2,5	25	250
$O_{4.1}$	Eggs	1	0,35	2,75	17,5
$O_{4.2}$	Apples	1	1,2	9,5	87

4.2 Understanding Service System Configurations and Dynamics

Conceptually speaking, a service system configuration is a dynamic configuration and is implemented by a specific value creation path between several service objects and their activities. In our model we also call it a value (co-)creation, since it often involves different actors working together to provide one service for a customer. Since each service object also has corresponding value propositions, any service system configuration, which is a configuration of service objects and thus of actors and resources, is also a value configuration. As such, a company is a service system and has the potential to configure different service system configurations to achieve a certain goal.

In the Fig. 1 below, that goal would be represented as achieving O_1. In our case, O_1 is a service object that has the customer as an actor and two apples and five eggs as resources. In other words, with the citizen-based produce delivery service, O1 is an object consisting of the customer receiving his ordered goods.

As Fig. 1 indicates, FD's service system has the potential to realize customer service. FD now faces the challenge of configuring the right service system for the job. To pinpoint the "right" configuration, two guiding questions need to be answered: can we deliver on time and if yes, how do we the service with the greatest margin? In other words, the time constraint and the number of ordered goods create the need for a dynamic model that can adjust accordingly to that context information.

Before answering those two questions in the next chapter, it is imperative to understand that different configurations are all subsystems G_1, G_2 and G_3 of an overall service system model G. As shown in Fig. 2, there are exactly three possible configurations for achieving G_1. To sum up, there are three possible permutations of who can provide what to deliver 2 apples and five eggs.

Specifically, the red path in the bottom left graph, middle graph and right graph are each possible service system configurations. The service system model allows us to identify these three "paths". Let us call the left path configuration G_1, the middle one G_2 and the third one G_3. For G_1, FD would buy two apples from the apple yard and five eggs from the general farm shop. For G_2, FD would have to rely completely on one actor, the general farm, to provide two apples, as well as five eggs. For G_3, we would rely on the apple yard to sell two apples and the elderly chicken enthusiast to provide five eggs. Furthermore, we assume that the general farm does not sell two apples only to FD. These possible configurations can next be used to dynamically assess the proper configuration based on the given circumstances (time constraint and maximum margin).
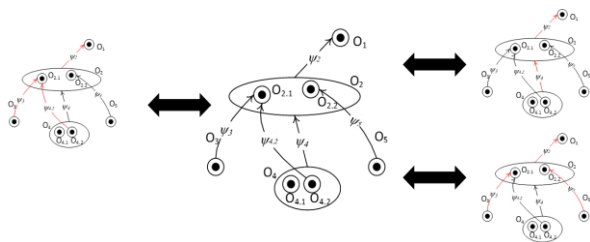

Fig. 2. Different Service System Configurations

5. Operationalizing Dynamic Solutions

First, the customer made a very clear time constraint of 3 hours. Furthermore, the service provider FD has the interest to minimize costs to maximize their profit margin, assuming that the price elasticity of the customer is almost non-existent. Relying on the service system graph, two analyses must be made: a time analysis and a cost analysis. Both are possible using our model. We start with the time analysis.

5.1 Time Analysis – optimizing delivery time

We have to calculate and compare total deliver time of each configuration $T_1(O_2)$, $T_2(O_2)$ and $T_1(O_2)$ with each other, with T being the required delivery time function. T calculates the sum of all delivery times for a certain configuration. Since the drop-off location is fixed, the delivery time can be calculated based on a 3rd party program, such as google maps. For our purposes, the maximum delivery time can be found in Table 1. Based on our time analysis it easy to identify that $T_1(O_5)$ is not suitable for this customer request, since delivery time of $T \circ \psi_5$ is 6 hours, which means that the total delivery would take 7 hours (1 hour for the apple delivery could happen in parallel to the 7 hours of the citizen-based elderly egg delivery. The final package would, however still require 1 additional hour). One simple explanation for the long delivery time from the egg enthusiast might be specific opening hours (the elderly citizen might be visiting relatives for the next 5 hours), high traffic estimation or a large distance between the mansion of the elderly egg enthusiast and the drop-off location. However, both $T_2(O_2)$ and $T_3(O_2)$ are both plausible service configurations, since both can deliver their goods within two hours.
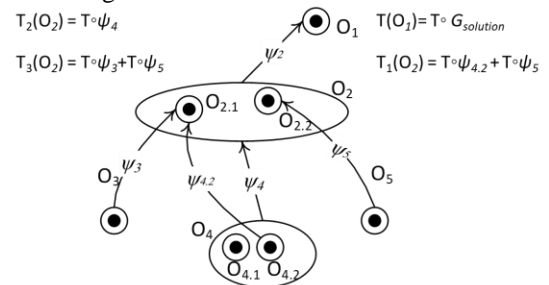

Fig. 3. Time analysis using SSM(FD)

5.2 Cost analysis – minimizing cost structure

During the time analysis, FD was able to eliminate one possible actor, the elderly egg supplier, who

possibly lives in a faraway neighborhood. Therefore, we removed O3 from the service system, since it is no longer useful for our purposes.

Additionally, it is in the best interest of FD to find the most cost-effective configuration among its entire service system. Due to reasons of simplicity, we limit ourselves to the costs of buying eggs and apples from different vendors. However, we also consider that the cost function can be a discreet function, as shown in Table 1. The most cost-effective configuration Gsolution is what we call a dynamic solution. FD therefore needs to know its bottom-line. Hence, $C_2(O_2)$ and $C_3(O_2)$ have to be compared.

Both cost functions are shown in Fig. 4. As mentioned in the case description, the farm has a special offer that costs 2,5$ and includes 2 apples and 10 eggs. The costs of buying 5 eggs from the big farm and 2 apples from the apple yard would cost 2,45$. Therefore, the most cost-effective configuration would be to buy 2 apples from the apple yard and buy 5 eggs from the general farm and send it to the pick-up station.
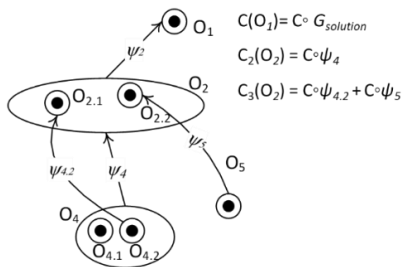

$$C(O_1) = C \circ G_{solution}$$
$$C_2(O_2) = C \circ \psi_4$$
$$C_3(O_2) = C \circ \psi_{4.2} + C \circ \psi_5$$

Fig. 4. Cost analysis using SSM(FD)

The dynamically chosen service system configuration is a chosen "path". There are two reasons, as to why we also call it a dynamic solution: First, since this path describes value-creation of actors and respective resources to solve a customer request, we call the chosen path a "solution". It is a solution towards a customer-oriented problem, based on context information, which is "what" the customer wants, how "much/many" the customer wants and where he wants the basket to be put. The dynamic part lies in finding an appropriate configuration from the service provider side. Thus, the FD was able to accommodate the dynamics of user requests and the changing cost structures and time constraints from the supplier side by leveraging the service system model and identify a dynamic solution.

6. Discussion and Future Work

We apply service system model (SSM) as an underlying modeling structure for operationalizing the service system of FreshnessDelivered, a business that connects and provides an adaptive transportation service for fresh produce.

Dynamically finding a balanced configuration between cost and time, depending on what is more important. In our simplified case, time was prioritized as a binary requirement, whereas cost was the secondary factor, which was responsible for further prioritization the remaining configurations (considering boundary conditions). In short, we were able to use the model to find the optimal configuration, while being able to dynamically take both times and cost under consideration.

In comparison to conventional application-centric modeling approaches, such as BPMN or service-specific versions of SBP, SSM includes additional information that is important for dynamically structuring services. For example: in process models, one would model the possible sequences for the delivery, while adding data objects as extensions to each activity. However, the contextual data of how the service is structured would need to be modeled separately, for example in an ERM or BOM. BPMN [16] and SBP variations [17, 33] do not observe the relationship between the required data and process' activities. We are able to model the entire structure of the ser-vice system using SSM [4, 20].

Similarly, recent modeling approaches rely on formal methods, such as petri-nets [34], also take on service-perspective, most notably service oriented computation [15]. However, they too rely only on a process perspective on what to do or communicate, yet don't include the composition structure of a service, let alone the inherent dynamics of context-sensitive service system. By relying on the multi-dimensionality of the extended hypergraph approach [4] and the concept of service system, SSM leverages both process and structural data perspective to model a more encompassing and information-rich service system.

This paper presents one service operations example dealing with the dynamics of service systems holistically. By modeling the entire service system, the service provider can better identify their bottom-line for ad-hoc service system configuration by having all information in one single model. For instance, if the customer requests buying 6 apples, there might be two apple yards in the vicinity. Each apple yard might have different apple prices or special offers. Taking the entire grocery basket into account, the service provider can calculate a bottom-line and adjust its price for the entire shopping basket accordingly. The pre-requisite for being able to calculate the total costs were made possible by a previously clearly modeled service system model.

Furthermore, this paper has shown how the service system model can be used to operationalize and quantify service system configurations. Additionally, resulting from mapping appropriate functions, each configuration can be analyzed as a means of decision support. Based on the comparative analysis, an optimal service system configuration can be made based on total time or total costs. This is also the prerequisite step for planning and scheduling, with both being core tasks for operations [5]. Future work should therefore apply our approach to find dynamic solutions and then start planning and scheduling their operations. Only then can the actual value proposition, for example the promise of having the ordered fresh produce de-livered to their target destination, be realized.

Furthermore, we have demonstrated how the dynamics of service systems can be useful for the service provider. However, there are cases in which the dynamics can be useful for the customer and enable new interaction possibilities. Let us continue with the fresh produce delivery service example: If the customer does not have any fixed pick-up location for their grocery bag and is more interested in finding the lowest price, then, by leveraging the service system model, the service provider could find several pick-up locations that are cheaper to deliver to. This leverages the service-provider knowing the bottom line and adjusting their delivery station based on the profit margin. In other words, a dynamic solution could be used to provide additional customer-centric options. Future research could apply our service system model to both identify and operationalize innovative services.

In addition, new possibilities for providing possible better offers to the customers are possible, similar to upselling. In our example FD did not choose to use the specials offer of 2 apples and 10 eggs, although the average cost of an apple and egg would be lower than the average cost of the chosen configuration. One might speculate that the general farm has an interest in selling eggs at a faster pace and give apples out for free, as a special bundle. The price difference was only 5 cents, yet the customer would have gotten 5 more eggs for it. Therefore, it would have been reasonable to offer that special to the customer as a means for upselling. The customer might even appreciate such a good offer more than "just" solving his request, increasing the perceived value. However, these avenues of research are now

possible all based on a single service system model and require additional research.

Additionally, the service systems configuration perspective can potentially help with innovation that is based on reconfiguration [14]. There are different views on how resource or value reconfigurations are forms of service innovation [35].

7. Conclusion

Since service systems are the basic unit of analysis of service science, we propose one approach for modeling and hence analyzing the service system. Hence, this paper introduces how SSM can be used as an underlying model for service systems to systematically structure it holistically and infer its operationalization. As discussed, this differs from conventional service system modeling approaches (both semi-formal and formal), which only capture parts of it.

This enables us to use the foundational model as a starting point for both constructing and analyzing service systems, as well as use it for different cases of operationalization. We have shown this by one example operationalization of a produce delivery service, which has to adapt to different situations and is thus always in a state of flux [3], although always within the bounds of the initial service system. Based on the company's service system, an optimal dynamic solution was able to be realized given a specific situation, which was all derived and based its one underlying service system model [20].

Additionally, we provide researchers a model for future research and future research could focus on possible graph transformation, paving the way for a model-based SSE approach. Additionally, one of the strengths of the formal approach lies in its direct relation and implementability of databases. Using our mathematical model, the database design becomes trivial. The service system model and its dynamic characteristics is therefore a foundational structure that includes both data and processes, leading to the operationalization of service systems. Future research could use the model as a blueprint for developing model-based service delivery systems.

12. References

[1] S. Gershwin, 2.854(F16) Introduction To Manufacturing Systems: Manufacturing Systems Overview.

[2] S.B. Gershwin, Manufacturing systems engineering, Prentice Hall, Englewood Cliffs, 1994.

[3] R.G. Askin, C.R. Standridge, Modeling and analysis of manufacturing systems, Wiley, New York, 1993.

[4] H.T. Papadopoulos, C. Heavey, J. Browne, Queueing theory in manufacturing systems analysis and design, 1st ed., Chapman & Hall, London, 1993.

[5] P.P. Maglio, J. Spohrer, A service science perspective on business model innovation, Industrial Marketing Management 42 (2013) 665–670. https://doi.org/10.1016/j.indmarman.2013.05.007.

[6] P.P. Maglio, C.A. Kieliszewski, J.C. Spohrer (Eds.), Handbook of Service Science, Springer US, Boston, MA, 2010.

[7] S.L. Vargo, R.F. Lusch, Service-dominant logic: Continuing the evolution, J. of the Acad. Mark. Sci. 36 (2008) 1–10. https://doi.org/10.1007/s11747-007-0069-6.

[8] H.-J. Bullinger, A.-W. Scheer, Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, in: H.-J. Bullinger, A.-W. Scheer (Eds.), Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, Springer Berlin Heidelberg, Berlin, Heidelberg, s.l., 2003, pp. 3–17.

[9] T. Böhmann, J.M. Leimeister, K. Möslein, Service Systems Engineering, Bus Inf Syst Eng 6 (2014) 73–79. https://doi.org/10.1007/s12599-014-0314-8.

[10] T. Böhmann, J.M. Leimeister, K. Möslein, The New Fontiers of Service Systems Engineering, Bus Inf Syst Eng (2018). https://doi.org/10.1007/s12599-018-0553-1.

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

[12] D. Beverungen, H. Lüttenberg, V. Wolf, Recombinant Service Systems Engineering, Bus Inf Syst Eng 21 (2018) 50. https://doi.org/10.1007/s12599-018-0526-4.

[13] W. Reisig, Towards a conceptual foundation of service composition, Comput Sci Res Dev 33 (2018) 281–289. https://doi.org/10.1007/s00450-017-0383-2.

[14] P. Fettke, Business Process Modeling Notation, Wirtsch. Inform. 50 (2008) 504–507. https://doi.org/10.1007/s11576-008-0096-z.

[15] M.J. Bitner, A.L. Ostrom, F.N. Morgan, Service Blueprinting: A Practical Technique for Service Innovation, California Management Review (2007).

[16] A.-W. Scheer, Business Process Engineering Study Edition: Reference Models for Industrial Enterprises, Springer Berlin Heidelberg, Berlin, Heidelberg, s.l., 1998.

[17] H.-J. Bullinger, A.-W. Scheer, K. Schneider (Eds.), Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen mit 24 Tabellen, 2nd ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[18] M.M. Li, C. Peters, Reconceptualizing Service Systems – Introducing Service System Graphs, International Conference on Information Systems (ICIS) (2018).

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

[20] H. Fromm, J. Cardoso, Foundations, in: J. Cardoso, H. Fromm, S. Nickel, G. Satzger, R. Studer, C. Weinhardt (Eds.), Fundamentals of Service Systems, 1st ed., Springer International Publishing, Cham, s.l., 2015, pp. 1–32.

[21] J.D. Chandler, R.F. Lusch, Service Systems: A Broadened Framework and Research Agenda on Value Propositions, Engagement, and Service Experience, Journal of Service Research 18 (2015) 6–22. https://doi.org/10.1177/1094670514537709.

[22] S.L. Vargo, P.P. Maglio, M.A. Akaka, On value and value co-creation: A service systems and service logic perspective, European Management Journal 26 (2008) 145–152. https://doi.org/10.1016/j.emj.2008.04.003.

[23] P.P. Maglio, J. Spohrer, Fundamentals of service science, Journal of the Academy of Marketing Science 36 (2008) 18–20. https://doi.org/10.1007/s11747-007-0058-9.

[24] J.M. Leimeister, Dienstleistungsengineering und -management, 2nd ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2019.

[25] L. von Bertalanffy, General System Theory, 4th ed., George Braziller, New York, 1968.

[26] T.P. Hill, On Goods and Services, Rev Income Wealth 23 (1977) 315–338. https://doi.org/10.1111/j.1475-4991.1977.tb00021.x.

[27] I. Jacobson, Object-oriented software engineering: A use case driven approach, Addison-Wesley, Harlow, 1998.

[28] P.P. Maglio, C.A. Kieliszewski, J.C. Spohrer (Eds.), Handbook of Service Science, Springer US, Boston, MA, 2010.

[29] M.M. Li, C. Peters, J.M. Leimeister, A Hypergraph-based Modeling Approach for Service Systems, Informs Conference on Service Science (2018).

[30] C. Berge (Ed.), Hypergraph: Combinatorics of Finite Sets, Elsevier, 1989.