XXXIII. International RESER Conference: paper template for authors

Service Design Patterns for Transforming Business with Service Dominant Architecture (SDA): Insights from a Longitudinal Case Study

Peter Weiß[1], Markus Warg[2], Andreas Zolnowski[3]

[1]Pforzheim University; [2]Institut für Service Design, Hamburg; [3]Signal Iduna, Hamburg

Service Dominant Architecture (SDA) operationalizes S-D Logic and Service Science concepts and lies at the core of the digital transformation endeavor of an insurance company in Germany. A service perspective offers new strategic options and creates new opportunities to change value creation pathways through a systemic view on value creation activities. In this scenario, Service Dominant Architecture (SDA) serves as construction plan for our case company and guides collaborative creation, building and application of novel value propositions on digital service platforms. The paper describes our design theorizing based on service design principles and patterns to generalize our findings. SDA service design patterns are evaluated using a longitudinal case study approach of an insurance company in Germany.

1. Introduction

Today, digital technologies fundamentally transform whole business models, products and services and stimulates emergence of new markets (Ross et al. 2023, Vargo et al. 2023). Companies alter their existing business models by incorporating digital technologies to strive for new value propositions and new resource configurations. New capabilities need to be built around emerging digital technologies and trends such as hybrid cloud, intelligent process automation, and artificial intelligence (AI), in particular machine learning or generative AI. However, investments in IT infrastructure capabilities should be driven primarily by business initiatives. In this context, strategic agility expresses the ability of a company to readily implement respective business initiatives opening up new opportunities for value co-creation. However, this requires upfront targeted investments to build required foundation of execution. Foundations for execution are enabled by enterprise architecture and related ICT infrastructure capabilities as pivotal decision domain for organizations to transform their business (Ross et al. 2006, Henderson and Venkatraman, 1993, Weill et al. 2002).

The paper presents insights from a longitudinal case study of an insurance company in Germany which the authors accompany since 2015. Our research explores the potential and challenges of digital transformation in the insurance business. Trends such as connected cars, automated driving, smart home, smart cities, connected healthcare are just a few examples representing change and new requirements for the insurance business. In subsequent sections, presented research takes focus holistically on SDA. SDA as framework (Warg et al. 2016, Spohrer et al. 2022) was derived from the knowledge base of the concepts of Service Science, S-D logic, and institutional economics, with the aim of putting the findings, logics, and processes into practice by facilitating actors in the process of value cocreation (Spohrer et al., 2022, p. 93). In previous contributions on research on SDA has primarily presented results and knowledge concerning IT artifact building, intervention and evaluation in its organizational context (Baskerville et al. 2018).

The remainder of the paper is organized as follows. First, research approach and objectives are described. Then design principles and patterns are elucidated. Next, SDA and its main building blocks are summarized. We present on basis of a concrete example and use cases, how service design patterns are implemented and used for transforming business in our case company with SDA. Finally, we reflect on our findings and conclude with an outlook on future research activities.

2. Research Approach and Objectives

In this section, we reflect and expand on chosen research approach and objectives. Presented research on Service Design Patterns with SDA is primarily informed and guided by a Design Science Research (DSR) approach oriented towards requirements discussed in (Hevner et al. 2004, Peffers et al. 2007, Baskerville et al. 2018, Peffers et al. 2020, Lee et al. 2011), Case Study Research (Yin 2014) and Action Design Research (ADR) (Sein et al. 2011).

Accordingly, applied research approach is abductive. This paper is part of the communication activity and step which is a key part of a DSR project. Our research design incorporates a longitudinal single case study approach (Yin 2014, 49-56) allowed us to investigate and study the problem at hand (since 2015), as well as to strive for a deeper and more comprehensive understanding of the given organizational context and its properties.

SDA represents our primary unit of analysis for chosen single case study approach which is still studying the initial research propositions and questions informed by ADR (Action Design Research) building, intervening and evaluating SDA as artifact (Warg et al. 2016, Weiß et al. 2018). SDA as architecture enables responsible actors (entities) such as organizations to evolve roles and systems that by their implementation and mutual value creation become dynamic value cocreation configurations and by this service systems (Spohrer et al., 2022; Warg & Engel, 2016; Warg et al., 2015, p. 93; Warg et al., 2016). Accordingly, SDA as framework consists of five subsystems. At the core, SDA operationalizes Service-Dominant Logic (SDL) and Service Science concepts serving as kernel theory (Gregor et al. 2020) and provides justificatory knowledge for design theorizing. Design theorizing forseees to develop respective service design principles and patterns to

communicate commonalities through guiding principles as essence of realized solution designs.

SDA research is oriented towards needs and requirements of IS research by "[…] growing knowledge around building and evaluating the IT artifact" (Baskerville et al. 2018, 362). Thus, we refer as well to the methodology of technical action (Eekels and Roozenburg 1991) and consider system development and evaluation (Nunamaker et al. 1990; 2015). Subsequently, we reflect and generalize prescriptive design knowledge on basis of design principles and patterns to continue our DSR research project. We present on basis of a concrete example and use cases, how service design patterns are used for transforming business in our case company with SDA.

Design theorizing is an expected norm and prerequisite for DSR (Baskerville et al. 2018, 363, Lee et al. 2011, Gregor et al. 2020). Developed theory needs to address prescriptive statements about "[…] how artifacts can be designed, implemented and evaluated (Baskerville et al. 2018, 363). Accordingly, we are collecting as theorizers in indicated process primarily relevant descriptive information and insights from actors (such as implementors, recipient users, enactors) how SDA is changing conditions, structures, systems, service systems, processes and routines and communicate them regularly according to DSR requirements (Basker ville et al. 2018, Peffers et al. 2007, Hevner et al 2004, Warg et al. 2016, 2019a, Weiß et al. 2019b, 2022, Vial 2019). Hence, our research embraces a learning abroach based on creative insights, trial-and-error-processes and reflection of created effects based on implemented use cases. Our IT artifact is considered as vehicle for research and practice impact (Baskerville et al. 2018, 369).

In the remainder we argue that service design patterns are an adequate strategy and mean to transform business with SDA.

3. Design Principles and Patterns

In previous section we have already highlighted DSR and its key objectives to "[…] understand, explain and sometimes predict the development use, and impact of information systems and related sociotechnical artifacts in organizations and other social contexts (Gregor et al. 2020, 1037). In response to requirements of DSR our research addresses both artifact and theory. In previous publications building, intervening and evaluating the IT artifact were in focus (Warg et al. 2016, 2019, 2023, Weiß et al. 2018, 2019). Hence, in the remainder we argue that design principles and patterns contribute to design theorizing (Gregor et al. 2020, Lee et al. 2011, Baskerville et al. 2018, 367). In this way, research outcomes and respective learnings are formalized to striving for broader applicability of SDA beyond the given organizational context (Sein et al. 2011, 41).

3.1. Design Theorizing Framework

DSR research design embraces academic rigor and practical relevance, respectively (Hevner et al. 2004). Thus, underlying research process foresees to theorize about the created IT artifact, for example by formulating design principles, as well as to evaluate its practical utility including analyzing the feasibility of deployed IT artifact in

the organizational context, e.g. through cases or laboratory trials (Baskerville et al. 2018, 365). Subsequently, we describe our design theorizing based on design principles and patterns with the aim to generalize our findings. Figure 1 illustrates the context of presented SDA research and underpins the relationship between design principles and design patterns.

Presented research approach analyses both perspectives shown, that of the theorizer and the implementer. An important element of our SDA research is that architectural elements are derived and argued on basis of justificatory knowledge, namely S-D Logic, which is continuously being evaluated concerning its practical relevance. The figure illustrates the dynamic relationship between design principles and design patterns as the produced IT artifact is evolving through organizational use and subsequent design iterations (Sein et al. 2011, 42). As architecture, SDA offers design principles and patterns for change becoming "[…] part of social and cultural practices and are reused over and over again in different social, organizational, and technological contexts" (Spohrer et al. 2022, 38). Gregor et al. (2020, 1637-38) refer to design principles for technology-based artifacts in socio-technical systems and propose guiding principles including six requirements how design principles should be formulated, presented and communicated. In general, three categories of design principle formulation can be identified: (1) design principles that encapsulate users' use of artifacts, (2) design principles that encapsulate artifact features, and (3) design principles that describe both (i.e., that are focused on both artifact features and user activity)" (Gregor et al. 2020, 1228).

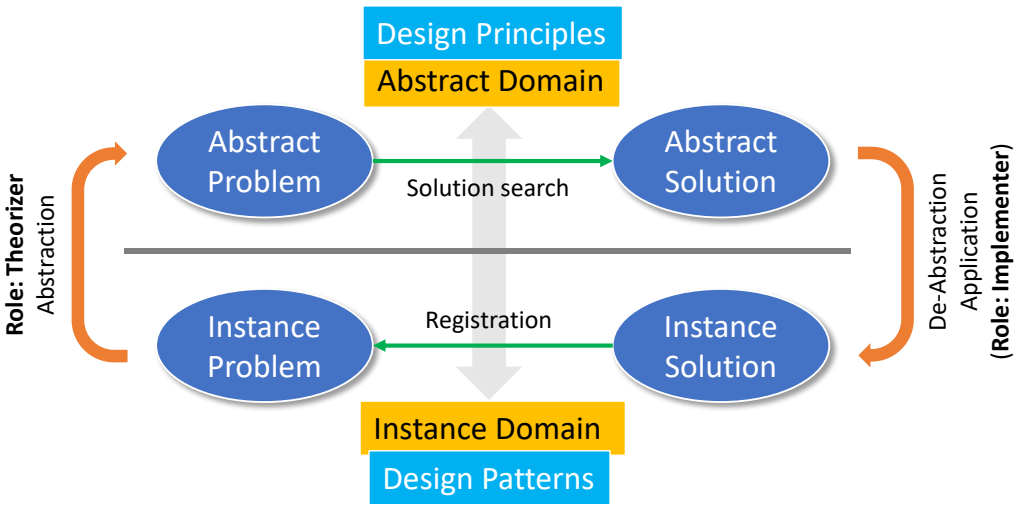

Figure 1: Relationship Design Principles and Patterns (based on Gregor et al. (2020, 1630), Lee et al. (2011, 6)) (own illustration)

SDA research addresses the latter category by facilitating the process of value cocreation (user activity) and artifact features (service systems). Research activities based on use cases indicate "[…] the need to understand users in their routines and activities and their interaction with the envisioned artifacts" (Gregor et al. 2020, 1225). SDA is understood as theory-ingrained artifact based on S-D Logic and Service Science principles from which related design principles are derived. Consequently, S-D Logic and Service Science serve as kernel theories delivering required "justificatory knowledge" as foundation and explanation for the SDA conceptual design (Gregor et al. 2020, 1226). Our design theorizing draws primarily

from S-D Logic foundational premises and axioms. Furthermore, Service Science and service systems are manifesting the theoretical elements of our designed artifact (Sein et al. 2011, 41).

3.2. Formalizing Design Principles and Patterns

Design principles are argued to be theoretical abstractions serving a purpose and possess utility (Gregor et al. 2020, 1229). Correspondingly, design principles should be "[…] understandable and useful in real-world design contexts" (Gregor et al. 2020, 1223-26). Design principles are appropriate when they (1) guide implementation and deployment activities, and (2) offer support to accomplish organizational goals (Gregor et al. 2020, 1638). Theorizing process arguably is necessary to arrive at abstractions in form of design principles which contribute to research practice and support application of design knowledge in professional practice. This is achieved through continuous reflection/abstraction and application/experimentation occurring in cycles as shown in Figure 1 until "[…] relatively stable design knowledge can be formalized" (Gregor et al. 2020, 1229).

ADR distinguishes two dominant design perspectives (Sein et al. 2011, 42), namely 1) IT-dominant and 2) organization-dominant and position potential theoretical and practical contributions of an IT artifact in a research continuum spanned by those two perspectives as respective endpoints. IT-dominant design seeks innovation from the realization of novel technological design, whereas organization-dominant design takes focus on innovations related to organizational intervention (Sein et al. 2011, 42). Emerging artefact is continuously instantiated and repeatedly tested through organizational intervention (Sein et al. 2011, 42). In this way, practitioners contribute with first-hand experience to artifact's design and participate actively in creating design knowledge (Sein et al. 2011, 42). Subsequently, we take a closer look on the two separated domains shown in Figure 1 and clarify purpose, activities and roles.

Abstract Domain and Design Principles

In the abstract domain actors elaborate on and formulate appropriate design principles. On this level, DSR research takes focus on abstraction activities (generalization) to develop a design theory to ease the solution search in targeted domain (Gregor 2006, 616). Solution search builds connections between perceptions of a problem, related changes and possible actions (Gregor et al. 2020, 1629). Design principles are evaluated on basis of convincing demonstrations that the principles work when applied in real practice (Gregor et al. 2020, 1232). During design theorizing researchers infer and explicate key concepts observed in a specific SDA instance. Chosen key concepts need to be grounded on and validated against respective justificatory knowledge (Lee et al. 2011, 7; Gregor et al. 2020, 1632). Strong design theory requires to demonstrate "[…] applicability across widely different settings, and address a broad class of design problems" (Lee et al. 2011, 7).

Starting with a preliminary, initial design ongoing organizational interventions achieve new perspectives and varying participating actors contribute through suggesting refinements based on reflections on evaluation results (Sein et al. 2011, 44). Further activities in the abstraction domain foresee to work towards more general design principles to achieve generalized outcomes (Sein et al. 2011, 44). This includes generalizations on expected organizational change along with implementation and

intervention of the IT artifact (Sein et al 2011, 44). This requires development of design-relevant explanatory/predictive theory that "[…] formally captures the translation of general theory constructs from outside IS to the design realm" (Gregor et al. 2020, 1232). Design knowledge abstracted and captured by the theorizer into design principles serving as representations in the abstract domain, however to be interpreted and used in the instance domain by the roles implementor, enactor and recipient user (Gregor et al. 2020, 1029-30). Design principles are formulized through explicating aim, context, mechanism and rationale (Gregor et al. 2020, 1633).

Instance Domain and Design Patterns

While the theorizer translates knowledge from the instance domain to the abstract domain, the implementor interprets abstracted design knowledge for the instance domain and implements it as instances of the designed high-level artifact. Innovations are expressed in design patterns addressing the same instance problem however with different means and mechanisms. Design in engineering begins with choosing a domain. A domain is any cluster of components to form solution instances on basis of collection of practices and knowledge, its rules of combination, and its associated way of thinking (Arthur 2009, 70). Design patterns express "[…] convenience and effectiveness of an assembly, what it can accomplish, how easily it can link with other assemblies, and what it will cost" (Arthur 2009, 72). Registration encompasses evaluating, modifying and registering a solution instance in relation to the problem instance (Gregor et al. 2020, 1629-30) (see Figure 1). Interventions are preferably done on basis of use cases which necessitate reflecting specific boundary conditions. Implementors use available design knowledge and augment it with own experience and knowledge of local context and technical rules to plan and design a specific intervention in the organizational context (Gregor et al. 2020, 1630).

Design principle contain often a basic idea, expected effect or specific purpose which is translated into a working technology (Arthur 2009, 119). "Sometimes a principle is borrowed – appropriated from some other purpose or domain that uses it" (Arthur 2009, 113). Hence, application of design principles necessitates to decompose them to lower system levels to make them applicable and understandable. On lower design levels de-abstraction takes place, and previously abstracted design knowledge is broken down, explained in smaller parts and mapped onto appropriate conceptualizations and functionality offered by purposed subsystems. Implementor and enactor are required to choose the adequate level of abstraction and level of granularity to ensure that certain mechanisms are expected to achieve particular aims and outcomes (Gregor et al. 2020, 1631-32). Engineers "[…] design and construct artifacts" (Arthur 2009, 90). Design patterns are a concrete configuration of technical rules, mechanisms and means used for the instantiation of design principles. Design patterns capture design knowledge in the instance domain, because instantiated IT artifacts usually regard a specific, unique situation, setting or context (Baskerville et al. 2018, 367). Design pattern refers to the concrete instance solution, whereas design principle is linked to the abstract solution. This reflects the fact that design principles can be implemented in various ways and means achieving the same aim and outcome. Design patterns are concerned with the new emerging combinations. Arthur (2009) elucidates domain's grammar as mean to determine "[…] how its elements fit together and the conditions under which they fit together. It determines what "works"" (Arthur 2009, 77). Design patterns reflect knowledge often reducible to rules of thumb from previous experience what works and what works not in a given context and which domains should be selected and combined to achieve a

goal or outcome to solve the instance problem (Arthur 2009, 77). “Designers construct from the domain they know” (Arthur 2009, 79). Arthur (2009) argues further that domain or body of knowledge […] provides a language for expression, a vocabulary of components and practices designers can draw from” (Arthur 2009, 79).

4. Service Dominant Architecture

In this section Service Dominant Architecture (SDA) is overviewed and main building blocks are described. We draw from previous publications, namely (Warg et al. 2015, 2016, 2019, 2023, Spohrer et al. 2022, Weiß et al. 2018, 2019, 2022). Service Dominant Architecture is a construction plan used by our case company for the collaborative creation, building and application of value propositions on digital service platforms. SDA operationalizes core concepts of Service Science and S-D Logic and formulates respective value statements (not truth statements) as guiding principles for technical action (Eekels and Roozenburg 1991, 198), such as design of IT artifacts (Baskerville et al. 2018, 366; Sein et al. 2011), with the aim to intervene and create transformational impact in the given organizational context.

SDA can be viewed from a conceptual and an applied perspective: (1) firstly, SDA can serve as conceptual framework in the understanding of a structure as a virtual order or design pattern of five purposed subsystems. (2) secondly, SDA is seen as tangible structure instantiated (e.g., based on platform technologies) by at least one (responsible actor) entity. The instantiated structure consists of five purposed subsystems. that by their implementation by an responsible actor (entity) become dynamic value cocreation configurations and by this service systems (Spohrer et al., 2007). SDA applied within an actor-to-actor network facilitates the process and coordination of service exchange and value co-creation. The five purposed systems are orchestrated according to underlying organizing logic are captured and formalized as SDA design patterns. In context of information systems design SDA represents “[…] [t]he abstract ‘blueprint’ or architecture that describes an IS artifact, either product or method/intervention” (Gregor et al. 2020, 1225). SDA conceptualizes offered affordances and comprises for each of its decomposable subsystems respective “[…] statement of the aim (goal, purpose) and means for achieving the goal” (Gregor et al. 2020, 1225).

4.1. Abstraction and Framework: Design Principles

In previous section we have already elucidated aims of design theorizing in DSR projects and have described purpose and intention of relevant abstraction activities towards generalization of design knowledge beyond its original organizational context.

SDA is subject of continuous design theorizing to ease the solution search in the targeted domain. Service design patterns are detailing how the five purposed subsystems need to be configured and realized in real intervention activity. SDA clarifies related high-level requirements of service design (design principles) with regard to underlying actors, roles, processes, resources, structures and mechanisms (Töytäri et al., 2018; Warg & Deetjen, 2021a, 2021b). The architecture enables responsible actors such as organizations to evolve roles and systems that by their

implementation and mutual value creation become dynamic value cocreation configurations and by this service systems (Spohrer et al., 2022). In the following the five purposed subsystems shown in Figure 2 are explained as value statements (Warg et al., 2015; 2016; 2019). The aim is to validate in later stages if respective value statements against requirements and design principle schema (Gregor et al. 2020, 1633) including aim, context, mechanism and rationale.

1. System of Operand Resources: The system of operant resources is the heart of the SDA. It represents the workbench, where the various resources and capabilities are brought together and processed. For this, this system applies certain logics or processes. In line with Service Science and S-D Logic, the focus is on intangible capabilities, previously defined as operant resources (like competence, knowledge, skills, software code), which are used and brought together to (co-) create value propositions. The emergence of value propositions (Vargo et al. 2023, 9) is dependent on the achievable level of resource density. A high resource density positively impacts the possible combinations and thus the emergence and creation of innovative value propositions.

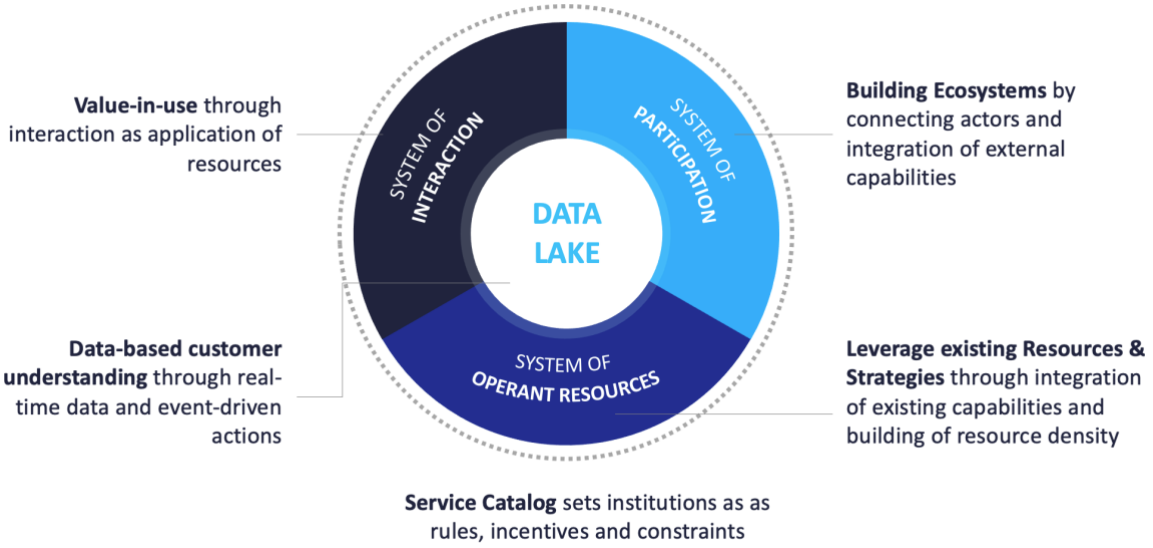

Figure 2: Overview Service Dominant Architecture (SDA) and aims of the five purposed subsystems (Warg et al. 2023, Spohrer et al. 2022).

2. System of Interaction: The system facilitates value in use and value in context by enabling the application of capabilities bundled in value propositions. Interaction enables resource integration and service exchange between actors and by these new resources with value creating potential.

3. System of Participation: The concept of co-creation includes other (external) actors as co-producers of the value proposition. In this process the system of participation enables actor-to-actor orientation and the participation of other actors by coordinating actors and facilitating the process of resource integration.

4. System of Operational Data Stores (Data Lake): From an actor's (e.g. organization) point of view, data received and generated by interacting with other actors (e.g. customer) should be systematically recorded and evaluated in real time. In this way, data and knowledge about the preferences and the context of other actors like customers can be build up continuously.

5. System of Institutional Arrangements (service catalogue): As rules, institutions enable the coordination of actors and the access to and use of resources. In conjunction with design pattern, institutions enable the coordinated creation of solution designs by connecting actors, and enabling the integration of resources.

Table 1 summarizes the SDA value statements and refers in addition to linked justificatory knowledge for design theorizing.

Table 1: SDA design principles for purposed subsystems

Component	Rationale	Justificatory knowledge
(1) System of Operant Resources	workbench mobilizing, orchestrating and processing various resources and capabilities in real-time to co-create	Value Proposition - Coproduction - Resource Density
(2) System of Interaction	facilitating interactions of actors to facilitates value in use and value in context to (co-) create value propositions	Interaction - Resource Integration - Service for Service Exchange
(3) System of Participation	enables actor-to-actor orientation and participation of other actors by coordinating actors and facilitating resource integration	Coproduction - Resource Integration - A2A Networks
(4) System of Data (Data Lake)	knowledge derived from data received and generated by interacting with other actors and recorded and evaluated in real-time	Data - Knowledge - Event-driven
(5) Service Catalog	rules for coordination of actors and access to and use of resources to create solution designs	Actor Coordination - Institutions - Trust - Social/Economic Practices

SDA as architecture can best support the required changes, constantly adapting to multiple environmental factors like competitors, regulations, technologies, customer preferences, employees, partners, shareholders, while competing for collaborators (following the paradigm and ideal of emerging architecture) (Vargo et al. 2023).

4.2. De-Abstraction and Implementation

In essence, SDA is a construction plan to create value propositions enabled by service platforms and to enact service systems. Value propositions when of required intensity connect other service systems creating complex value constellations.

SDA as architecture enables both the process and the output of planning, designing and constructing (Gamma et al., 1995; Safin et al., 2010; Warg & Deetjen, 2021a). Conceptually, SDA provides a set of design patterns, and practically, the patterns is instantiated as a structure of five systems enabling both the process and the output of mutual value creation (see Figure 3).

SDA service design patterns as shown in Figure 3 strive for an overarching perspective which enables an integrated view on the crucial elements of the instance solution. Agile development practices and “patterns” play a salient role to build and intervene artifacts and to build required design knowledge. Service Design Patterns support learning and contain design knowledge. "Patterns are a starting point not a final destination" (Fowler, 1997) for the agile process of value cocreation between organizations, technology provider, AI provider and other actor. SDA provides required domain vocabulary as "ubiquitous language" for operationalizing Service Science and S-D logic concepts through building shared domain models (Evans, 2004, Fowler 1997). Watermann (2015) denotes this as "Emergent Architecture"; an architecture in which the team makes only the minimum architecture decisions up-front, such as selecting the technology stack and the highest-level architectural styles and patterns" (Waterman et al., 2015). Supplemented by the centrality of service and the required Service-Dominant mindset, we propose to denote this approach as "Agile Emergent Service Dominant Architecture".

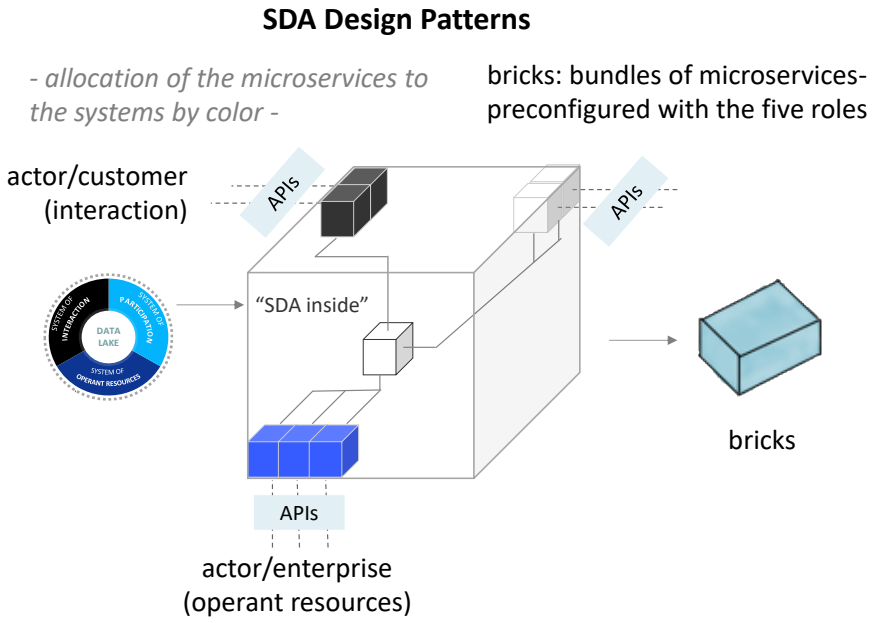

Figure 3: Nature of SDA Service Design Patterns: Construction Plan of Stacks

Emergence in this context requires standard engineering as practice to “[…] find a form, a set of architected assemblies, to fulfill a set of purposes” (Arthur 2009, 91). As already highlighted, service design patterns contain design knowledge from previous experiences and learning. Hence, building, intervening and evaluating according to ADR means “[…] matching a purpose with some concept of a structure that will meet it, and putting together a combination of assemblies that will bring this structure to reality” (Arthur 2009, 91).

Respective design projects are typically about planning and constructing of a new version of a known technology by applying known concepts and methods to given problems (Arthur 2009, 91). SDA service design patterns “[…] offer standard solutions to problems that come up repeatedly, designs that can be modified for particular uses” (Arthur 2009, 102). Solutions arise “[…] through practitioners finding new way, a new clever combination of existing components and methods that resolves a standard problem” (Arthur 2009, 102). Technically, service design patterns define models concerning interfaces between the various system components shown

and transform them into implicit and explicit specification models which suggest how components are to be implemented (Fowler, 1997).

Service design patterns intend to offer guidance for domain experts to elaborate collaboratively on respective solution designs. They are guidelines on constructing digital service systems. Using SDA design patterns allows to overcome challenges of resulting system complexity by avoiding technical debt as what is discussed as “Big Ball of Mud” (Evans 2004). Service design patterns are reusable and implement resource integration patterns facilitated by new (digital) capabilities. Using service design patterns necessitates less strategic design decisions up-front, which enhances strategic agility and accelerates time to market. Service design patterns contain and deliver a common and shared language among customers, domain experts and solution architects to cocreate and implement service systems in targeted domains. Strategic design is vital to overcome system complexity by avoiding what is discussed as “Big Ball of Mud” (Evans, 2004; Vernon, 2013).

SDA as framework and conceptual model supports strategic design of information systems through enhanced collaborative understanding of the targeted business domain. Like a construction plan the service design patterns as structure of five systems facilitate both the process and the output of value cocreation. For each system service design patterns are used to further concretize and detail solution design and resource configurations. Consequently, service design patterns are the next logical step in evolving the SDA for practitioners and researchers alike. In conclusion, SDA supports emergence through its service design patterns which allow context-specific configurations of instance solutions through usage of standardized solution components. In next section we refer to a concrete example and use case to illustrate how service design patterns are applied.

5. Use Case Solution Design and Demonstration

DSR projects are typically longitudinal streams of research (Baskerville et al. 2018, 368-69). Use cases provide the context for instantiation and demonstration of the novel IT artifact and are seen as a research contribution that “[…] embodies design ideas and theories yet to be articulated, formalized and fully understood” (Baskerville et al. 2018, 362). Subsequently, a new solution design for AI-based decision is described, before selected SDA use cases are overviewed, and the use case “Mobility Platform” is briefly presented.

5.1. Solution Design for AI-based Decision-Making Processes

SDA allows for rapid change and adoption of new technologies, including AI (Artificial Intelligence), to accelerate digital transformation and to turn resource density into true, market-accelerating service innovations.

SDA contains and delivers a common and shared language among domain experts and solution architects to implement information systems in targeted domains. Guiding principle are Service Science and SD-Logic informed design patterns. SDA addresses exactly this particular need and provides required domain vocabulary as "ubiquitous language" and building foundations for required domain models (Evans,

2004). To model the solution design for AI-based support of decision-making processes in organizations, we have chosen a two-step process.

In the first step, key elements and their interplay are shown in generic terms using the SDA conceptual framework. SDA provides respective design patterns to ensure that solutions reflect appropriately the overall conceptual model of value cocreation (Fowler, 1997). And in a second step, specific requirements of AI-based decision processes in organizations are mapped on design patterns. On basis of identified use cases and events solutions are developed and specified. Figure 4 maps requirements to the conceptual model and demonstrates the interplay of the design patterns by implementing AI-based decision support. AI technologies are connected via the System of Participation to the other systems. The multi-mode data of the organization are integrated and harmonized applying the System of Operant Resources. The value in use of the decision support, e.g. time savings for employees, is realized via the System of Interaction. And the actor and data coordination is orchestrated via the System of Institutions.

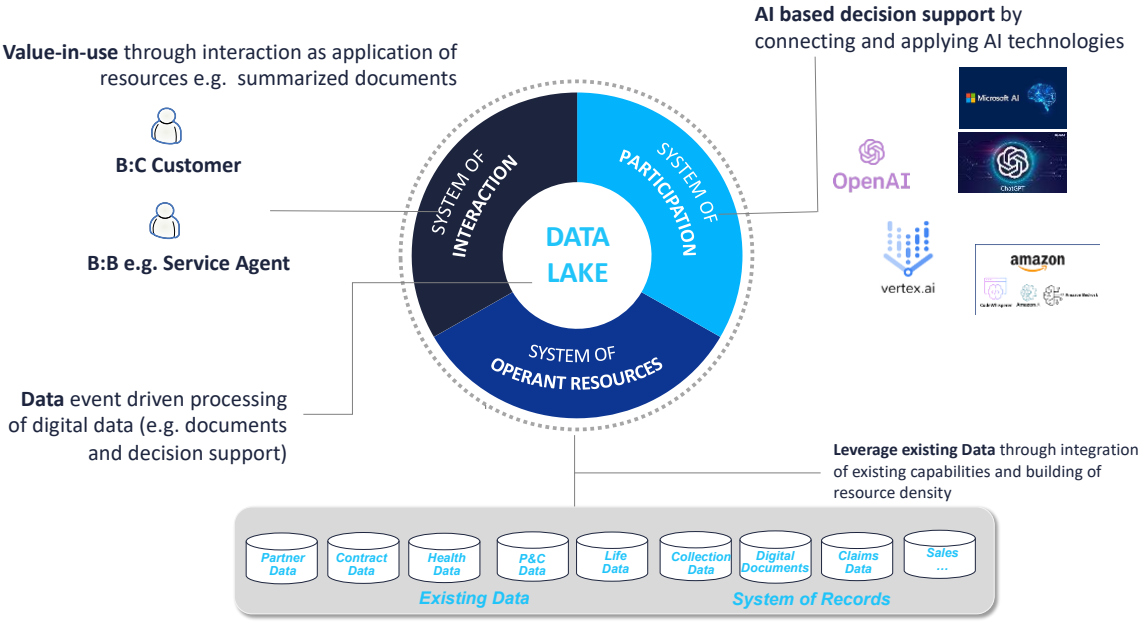

Figure 4: AI-based decision support (own illustration)

Using SDA we clarify in this first step related high level and generic requirements of AI-based decision support with regard to underlying actors, resources, processes, structures, mechanisms as well as actors' roles (Töytäri et al., 2018; Warg & Deetjen, 2021a, 2021b). Design pattern enables responsible actors such as organizations to evolve roles that by their implementation and application become dynamic value cocreation configurations and by this service systems (Spohrer et al., 2022). Co-producer (human or non-human, e.g. technologies) as actors are connected and their resources are integrated by applying the System of Participation. Organization itself as actor and responsible entity of the five systems integrates its strategic relevant resources using the System of Operant Resources. Interaction (B2B or B2C) is organized with System of Interaction. All data - whether generated by co-producers, the organization or in the course of interactions - is orchestrated in the System of Data. And institutions as rules for coordination, access or constrains of actors' rights

or resource integration arrangements are set within the System of Institutions (Service Catalog).

5.2. Use Case Development: Mobility Platform

Stroke prevention is a prominent AI-driven use case implemented by SDA. Further details concerning this use case can be found in (Weiß et al. 2022) (see Table 2). Previously, we have already described the use case “stroke prevention” (see Weiß et al. 2022). This use case realizes collaborative, AI-driven services with a startup company named AI4medecine. Aim is to facilitate co-creation of novel value propositions through the network of stakeholders (end customer, insurance company, start-up) integrating resources such as digital technologies (e.g. wearables), business services, data services to enable personalized health services in the context of stroke prevention. Solutions are as well relying on previously described AI/ML (Machine Learning) competences and resources of the start-up ai4medicine (founded out of Berlin Charité), combined with competences, resources and services of other players.

Table 2: Selection of Example SDA Use Cases

No.	Use Case / Service	Description
1	Edith care	- personal care assistant - support for administrative activities - application process: five minutes instead of six days
2	Stroke prevention (ai4medicine)	- customer receives individual risk scoring and personalized recommendations and action list - personal health advisor app
3	Medicproof	- assistance and information services for customers to manage medicamental treatments and drugs
4	ODS - Operational Data Stores	- algorithms and partner capabilities while maintaining the highest security and data protection requirements - solution stacks (ODS), which contain all relevant data (contract and correspondence information) operating in real-time
5	Health prevention (recommendation service)	- use historical health data, rate information and integration of AI and other actors, - co-creation through individual recommendations for health prevention
6	Mobility platform	- B2B2C mobility platform to design and employ digital mobility services for car insurance and added value services - artificial intelligence to provide customers with tailor-made offers.

Another promising use case is an emerging service ecosystem offering mobility services. Currently a newly emerging SDA-enabled mobility platform is being implemented. The platform allows orchestrating and mobilizing resources from the various engaged stakeholders linked to service ecosystem. Created mobility platform represents a new emerging use case where SDA contributes through design patterns and standardized service stacks to implement value propositions and offerings.

Hence, SDA acts as enabler building service ecosystems and platforms for novel digital mobility services offered by a consortium of stakeholders from the insurance business. Remarkably, this use case and application domain is beyond the case company's boundary and can be seen as an important step towards generalizability of the SDA framework and related design principles. SDA forms the basis for the B2B2C mobility platform of stakeholders. Aim and purpose of the platform is to deliver customized on-demand business and technical solutions (stacks) which are triggered and ordered in the B2B context via the SDA service catalog (purposed subsystem). Technical components (stacks) (see Figure 3) can be installed effortless in given technical environment (instance) within minutes.

Service design patterns for AI-based decision support are as well offered and will be deployed in current and future use cases by the SDA-powered mobility platform. Consequently, those use cases allow us to further evaluate created IT artifact and validate derived design principles and patterns against user requirements and realistic application scenario. As part of the theorizing process, those insights are fed back to adapt, expand or complement our design principles. Hence, mobility platform constitutes another situated implementation which we consider as knowledge contribution from our DSR project (Baskerville et al. 2018, 362). Adhering to the goals of DSR we have invented SDA as new artifact in the new environment of the involved companies. This offers opportunities to further improve and evaluate SDA's initial design, productiveness and effectiveness through intervention activities. This is in line with what Nunamaker et al. (2015, 20-21) discuss as "proof-of-value" research. Accordingly, proof-of-value research creates bodies of explicit knowledge about the problem and solution space and strives for deeper understanding of the causal mechanism that underpin the investigated phenomena; in our case the gathered design knowledge and achieved outcomes of designs as well as implemented instances.

5.3. Findings

Previous sections presented details concerning background and context of why and how SDA as framework emerged and evolved. In this section major findings and results are summarized.

New emerging digital technologies, such as SMACIT (social, mobile, analytics, cloud, internet of things (IoT)), have already shown their potential to disrupt long established companies' business models. Digital technology trends are emerging continuously, such as artificial intelligence (AI), blockchain, process mining, etc. and keep decision makers busy to find strategic responses and right transformation strategies (Vial 2019, Ross et al. 2019). Named technology trends create "vacuums" offering new opportunities to reorganize value creation and in this way disrupting incumbents in their long-established markets (Normann 2001, 50). As well in the insurance business, companies need to find strategic responses to respond to current and

future changes in their incumbent markets. In summary, digital technologies offer new opportunities to reallocate physical activities in time and space (Normann 2001, 52). Hence, insurance companies are forced to rethink their strategies and to embark on a long journey to reconfigure or build new foundations for execution (Burden et al. 2018). One of the key challenges is to incorporate digital technologies to strive business through augmenting value propositions with new capabilities or "[…] creatively design framebreaking systemic solutions", either to maintain their existing customer base or reaching out for new, unprecedented customer segments (Normann 2001, 50).

In essence, digital technologies deliver new capabilities, three prominent capabilities discussed are ubiquitous data, unlimited connectivity, and massive processing power (Ross et al. 2021, 15). Those capabilities caused significant changes to company's organizing logic of people, processes and technology (Spohrer et al. 2022). Those changes are best described as new dominant logics for value creation demanding structural changes, changes in value creation paths and new organizing logic for shaping the foundation of execution (enterprise architecture) (Normann 2001, Spohrer et al. 2022, Ross et al. 2006, 2019, Vial 2019,5).

Transforming Business with Service Dominant Architecture (SDA)

The latter, new organizing logic, lies at the core of presented research on SDA. This new organizing logic is from our perspective arguably best described through the "service" perspective on value creation and respective imperatives (Normann 2001, Spohrer et al. 2022, Vargo et al. 2023)). Lusch and Nambisan (2015,156) argue service as transcending mental model for all types and forms of innovation (tangible and intangible). Hence, value- and experience-centric focus turns into an imperative (Lusch and Nambisan, 2015, 155). Most relevant and influential publications to understand respective strategic imperatives have been (Normann 2001, Vargo and Lusch, 2004, 2008, 2016, Lusch and Nambisan 2015, Spohrer et al. 2007, 2022). For transforming business with SDA, service innovation plays a salient role and its conceptualization as tripartite framework brings service ecosystem, service platform and value co-creation to the fore. Further, four metatheoretical foundations of S-D Logic are considered, namely: (1) actor-to-actor networks, (2) resource liquefaction, (3) resource density, and (4) resource integration. In addition, SDA as framework grounds on systemic principles of service systems and foundations of Service Science and operationalizes besides S-D Logic respective core concepts (Spohrer et al. 2007, 2022).

SDA Service Systems, Architecture and Service Design Patterns

"Architecture" translates and operationalizes previous theoretical concepts into applicable and implementable organizing logics and structures guiding design activities for the foundation of execution by reproducing systems of interaction through interpretative schemes facilitating resource integration and service exchange (Giddens 1984, 28-29, Spohrer et al. 2022, Warg et al. 2023, Warg 2023). Interaction, communication and emergence are salient concepts currently being discussed to explain novel outcomes produced by ad-hoc resource integration and service exchange (Vargo et al. 2023, 7). In case actors intend to reproduce respective interaction and behavior they need to generate durable patterns and interpretative schemes to be institutionalized by becoming norms, rules and beliefs) (Vargo et al. 2023, 10-11). Institutions serve as coordination mechanisms of value

co-creation. Institutional arrangements provide social structure that is both, medium and outcome of human action. This means, that new behavioral patterns are generated by actors through their behaviors (emergence), that are captured by institutional arrangements (rules, norms, beliefs) facilitating ongoing formation and reformation of multiple, increasingly complex systems or assemblage (Akaka et al. 2019, Giddens 1984, 28-29).

Agile Emergent Service Dominant Architecture

The goal is to make businesses better—more agile, more sense-and-respond, better able to keep up with and drive meaningful human-centred change in a fast-paced world, while also maintaining privacy, security, and regulatory compliance (Spohrer et al., 2022). Starting with SDA service design patters means starting with a Service-Dominant mindset (Vargo and Lusch 2008) and with only the minimum architecture decisions up-front, e.g. the technology stack and related mindset framing the solution space (Waterman et al., 2015). In the community of scrum this minimum approach is also called "Agile Emergent Architecture" (Bradley, 2018). In this context, SDA generates Service Design Patterns which capture respective interpretative schemes (Giddens 1984, 28-29) to reproduce patterns of resource integration and service exchange as organizing logic and structural elements of service systems. In this way, SDA facilitates through its five distinct subsystems a systemic view on value creation activities. SDA generates Service Design Patterns that reproduce underlying systems of interaction in organizations to strive relevant business initiatives and new value constellations to achieve novel outcomes.

6. Summary and Conclusion

The paper discussed service design principles and patterns in context of evolving SDA as design theory and framework. Previous sections presented SDA approach and selected results from our DSR project (Warg et al. 2015, Warg and Engel 2016, Warg et al. 2016, 2019, 2023, Weiß et al. 2018, 2019, 2022). As main contribution, broader applicability beyond the scope of the given organizational context and case company have been discussed. An important prerequisite constitutes the design theorizing framework elaborated and developed for our DSR project. As previously mentioned, SDA is grounded in Service Science and S-D Logic and provides an organizing logic and construction plan for shaping companies, service platforms, and service ecosystems through design patterns aimed at making it possible to build and orchestrate capabilities in a systematic way. We have highlighted, SDA can serve as conceptual framework in the understanding of a structure as a virtual order or design pattern of five purposed subsystems. (2) secondly, SDA is seen as tangible structure instantiated (e.g., based on platform technologies) by at least one (responsible actor) entity.

Firstly, results of this longitudinal case study intend to deliver interesting and relevant insights concerning impacts of digitalization and discusses related organizational aspects of servitization in the concrete context of a real digital transformation endeavor at hand. Secondly, our research explores potential and challenges of digital services innovations referring to real-life examples. The paper elucidates how our case company embraces rapid change and adoption of new technologies, including AI, to accelerate digital transformation and to turn resource density into true, market-

accelerating service innovations. SDA allows building, intervening and evaluation created IT artifacts, and we argue primarily on basis of real life examples of digital service innovations taken from our longitudinal case study research (use cases), such as personal care assistant, claim notification service, personal health advisor app, stroke prevention, information service Medicproof, and others. Intervention activities are vital for building and evaluating effective service systems designs in context as well as reflecting and generating best practices on design principles. Suitably, presented research contributes to expand the knowledge of information systems design by technical action and making.

Reusable service design patterns were introduced as best practices to build real practice solution designs based on SDA organizing logic. This stands in analogy with Lego bricks building for reusable building blocks which constitute design principles and conceptual service design patterns. Those service design patterns when mapped onto the five purposed SDA subsystems guide configuration of systems, structures, processes, actors and resources. Implementing service design patters transforms business by introducing a Service-Dominant mindset. In general, architecture decisions are done up-front incrementally and iteratively, through customizable and reusable design patterns linked to technical building blocks from the SDA technology stack (Spohrer et al. 2022). As result, SDA enables the foundation of execution to run real-life experiments stimulating transformation processes through introducing new practices and routines for organizational learning.

7. References

Bradley, C. (2018). An Introduction to Agile Emergent Architecture - Always Intentional. Retrieved 2018, September, 28th. from https://scrumcrazy.wordpress.com/2018/09/28/an-introduction-to-agile-emergent-architecture-always-intentional/

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: Finding a balance between artifact and theory. Journal of the Association for Information Systems, 19(5), 3.

Burden A, van der Ouderaa E, Venkataraman R, Nyström T, Shukla PP. Technical debt might be hindering your digital transformation. MIT Sloan Management Review. Cambridge, MA,USA: The MIT Press; 2018;60:1-7.

Evans, E. J. (2004). Domain-driven design: Tackling Complexity in the Heart of Software. Addison-Wesley.

Eekels J, Roozenburg N.: A methodological comparison of the structures of scientific research and engineering design: Their similarities and differences. Design Studies. 1991;12:197-203.

Fowler, M.: Analysis patterns: reusable object models. Addison-Wesley Professional. 1997.

Gregor, S., Kruse, L., Seidel, S. (2020). Research Perspectives: The Anatomy of a Design Principle. Journal of the Association for Information Systems, 21, 1622–1652. doi:10.17705/1jais.00649

Gregor S, Hevner AR.: Positioning and presenting design science research for maximum impact. MISQ.2013;37:337-355.

Gregor, S. (2006). The nature of theory in information systems. MIS quarterly, 611-642.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns - Elements of reusable object-oriented software. Reading: Addison-Wesley.

Giddens, A. (1984). The constitution of society: Outline of the theory of structuration. Univ of California Press.

Henderson JC, Venkatraman H. Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal.1993;32:472-484.

Hevner AR, March ST, Park J, Ram S. Design science in information systems research. MIS Quarterly. 2004;28:75-105.

Lusch, R. F., & Vargo, S. L. (2008). The Service-Dominant Mindset. In B. Hefley & W. Murphy (Eds.), Service Science, Management and Engineering Education for the 21st Century (pp. 89-96). Springer US. https://doi.org/10.1007/978-0-387-76578-5_15

Lee, J.S., Pries-Heje, J., Baskerville, R.L. (2011). Theorizing in Design Science Research. International Conference on Design Science Research in Information Systems and Technology.

Lusch RF, Nambisan S. Service innovation: A service-dominant logic perspective. MISQ. 2015;39:155-175.

Lusch RF, Vargo SL. Service-Dominant Logic: Premises, Perspectives, Possibilities. Cambridge: Cambridge University Press; 2014

Maglio, P. P., & Spohrer, J. (2008). Fundamentals of service science. Journal of the Academy of Marketing Science, 36(1), 18-20.

Maglio, P. P., Vargo, S. L., Caswell, N., & Spohrer, J. (2009). The service system is the basic abstraction of service science [10.1007/s10257-008-0105-1]. Information Systems and e-Business Management, 7(4), 395-406.

Normann R. Reframing Business: When the Map Changes the Landscape. Chichester: Wiley; 2001

Nunamaker JF, Chen M, Purdin TD. Systems development in information systems research. Journal of Management Information Systems. 1990;7:89-106

Nunamaker Jr, J. F., Briggs, R. O., Derrick, D. C., & Schwabe, G. (2015). The last research mile: Achieving both rigor and relevance in information systems research. Journal of management information systems, 32(3), 10-47.

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. Journal of Management Information Systems. 2007;24:45-77.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2020). Design science research process: a model for producing and presenting information systems research. arXiv preprint arXiv:2006.02763.

Ross JW, Beath CM, Mocker M. Designed for digital: How to architect your business for sustained success. Cambridge Massachusetts, London England: The MIT Press; 2019

Ross JW, Weill P, Robertson D. Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Boston: Harvard Business Review Press; 2006

Safin, S., Delfosse, V., & Leclercq, P. (2010). Mixed-reality prototypes to support early creative design. The Engineering of Mixed Reality Systems, 419-445.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. MIS quarterly, 37-56.

Spohrer, J., Maglio, P. P., Vargo, S. L., & Warg, M. Service in the AI Era: Science, Logic, and Architecture Perspectives. Business Expert Press. 2022.

Töytäri, P., Turunen, T., Klein, M., Eloranta, V., Biehl, S., & Rajala, R. (2018). Aligning the mindset and capabilities within a business network for successful adoption of smart services. Journal of product innovation management, 35(5), 763-779.

Vial G. Understanding digital transformation: A review and a research agenda. The Journal of Strategic Information Systems. 2019;28:118-144.

Yin RK. Case study research: Design and methods. 5th ed. SAGE: Los Angeles, London, New Delhi, Singapore, Washington, DC; 2014

Yin, R. K. (2018). Case study research and applications (Vol. 6). Thousand Oaks, CA: Sage.

Vargo SL, Lusch RF. Evolving toa new dominant logic for marketing. Journal of Marketing. 2004;68:1-17.

Vargo SL, Lusch RF. Service-dominant logic: Continuing the evolution. Journal of the Academy of Marketing Science. 2008;36:1-10.

Vargo SL, Lusch RF. Institutions and axioms: An extension and update of service-dominant logic. Journal of the Academy of Marketing Science. 2016

Vargo, S. L., Peters, L., Kjellberg, H., Koskela-Huotari, K., Nenonen, S., Polese, F., & Vaughan, C. (2023). Emergence in marketing: an institutional and ecosystem framework. Journal of the Academy of Marketing Science, 51(1), 2-22.

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. The Journal of Strategic Information Systems, 28(2), 118-144.

Vernon, V. Implementing domain-driven design. Addison-Wesley, 2013.

Weill P, Subramani M, Broadbent M. Building IT Infrastructure for Strategic Agility. MIT Sloan Management Review. 2002;44:57-66

Warg, M., & Deetjen, U. (2021a). Human Centered Service Design (HCSD): Why HCSD Needs a Multi-level Architectural View. In International Conference on Applied Human Factors and Ergonomics (pp. 249-256). Springer.

Warg, M., & Deetjen, U. (2021b, 2021). Why co-creation in service ecosystems needs an architectural view 30.th RESER International Congress and Coval Conference, Alcala, Spain.

Warg, M., Frosch, M., Weiss, P., Zolnowski. (2022). Service innovation roadmaps as benchmarks for organizational learning IESS,

Warg, M., Frosch, M., Weiss, P., Zolnowski, A. (2023): Service Innovation Roadmaps as Benchmarks for Organizational Learning, International Conference of Exploring Service Science, IESS 2.3, ITM Web of Conferences, DOI: 10.1051/itmconf/20235104001

Warg, M., Weiß, P., & Engel, R. (2015). Service Dominant Architecture. http://www.fhwedel.de/fileadmin/mitarbeiter/mwa/SDA_Whitepaper_30.11.2015.pdf

Warg, M., Weiß, P., Engel, R., & Zolnowski, A. (2016). Service Dominant Architecture based on S-D logic for Mastering Digital Transformation: The Case of an Insurance Company 26th Annual RESER Conference, Naples, Italy.

Warg, M., Zolnowski, A., Frosch, M., & Weiß, P. (2019). From Product Organization to Platform Organization - Observations of Organizational Development in the Insurance Industry [Conference paper]. Naples Forum on Service, Ischia/Naples. https://www.ifsd.hamburg/OPEN-SCIENCE/

Warg, M., & Engel, R. (2016). Service-Dominierte Architektur (SDA): Kernkomponente digitaler Transformation. Zeitschrift für Versicherungswesen, 2016(12), 391-395.

Warg, M., Weiß, P., Engel, R., & Zolnowski, A. (2016). Service Dominant Architecture based on S-D logic for Mastering Digital Transformation: The Case of an Insurance Company 26th Annual RESER Conference, Naples, Italy.

Warg, M. (2023): Science, Logic and Architecture Perspectives on Organizational Learning: observations based on the onpier Mobility Platform, The Naples Forum on Service, Ravello, June 2023

Waterman, M., Noble, J., & Allan, G. (2015). How much up-front? A grounded theory of agile architecture. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering,

Weiß, P., Zolnowski, A., Warg, M., & Schuster, T. (2018). Service Dominant Architecture: Conceptualizing the Foundation for Execution of Digital Strategies based on S-D logic 51st Hawaii International Conference on System Sciences (HICSS-51), Big Island, HI.

8. Author Address

Peter Weiß, Prof. Dr.
Pforzheim University
Business Information Systems
Tiefenbronner Str. 65, 75175 Pforzheim, DE
peter.weiss@hs-pforzheim.de

Markus Warg, Prof. Dr.
Institut für Service Design, Hamburg
Blankeneser Hauptstr. 11, 22587 Hamburg, DE
markus.warg@ifsd.hamburg

Andreas Zolnowski, Dr.
University of Hamburg
Business Information Systems
Mittelweg 177, 20148 Hamburg, DE
andreas.zolnowski@signal-iduna.de