Received: 30 May 2017 | Revised: 11 April 2019 | Accepted: 4 September 2019

DOI: 10.1111/isj.12269

WILEY

SPECIAL ISSUE PAPER

Capturing smart service systems: Development of a domain-specific modelling language

Rocco Xaver Richard Huber[1] | Louis Christian Püschel[2] | Maximilian Röglinger[2]

[1]FIM Research Center, University of Augsburg Project Group Business & Information Systems Engineering of the Fraunhofer FIT, Augsburg, 86135, Germany

[2]FIM Research Center, University of Bayreuth Project Group Business & Information Systems Engineering of the Fraunhofer FIT, Bayreuth, 95444, Germany

Correspondence
Maximilian Röglinger, FIM Research Center, University of Bayreuth Project Group Business & Information Systems Engineering of the Fraunhofer FIT, Bayreuth 95444, Germany.
Email: maximilian.roeglinger@fim-rc.de

Abstract

Over the last years, the nature of service has changed owing to conceptual advances and developments in information technology. These developments have given rise to novel types of service and smart service systems (SSS), ie, resource configurations capable of learning, dynamic adaptation, and decision making. Currently, the internet of things (IoT) is turning physical objects into active smart things, bridging the gap between the physical and the digital world. Smart things advance SSS as they observe the physical environment, access local data, immerse into individuals' everyday lives and organizational routines. In line with the emergent nature of both phenomena, the impact of the IoT on SSS yet needs to be explored. Building the basis for explanatory and design-led research and for the analysis and design of SSS, a means for the conceptual modelling of SSS that accounts for novel IoT-enabled concepts is in high need. Hence, we designed, demonstrated, and evaluated a domain-specific modelling language (DSML) for SSS. We evaluated the DSML by using it in the modelling of real-world scenarios from all functional IoT domains, by submitting it to the scrutiny of industry experts, by discussing it against generic DSML requirements, and by analysing to what extent it meets domain-specific design objectives compared with competing artefacts. To demonstrate the DSML, we

included a complex real-world scenario centred around the Nest Learning Thermostat.

KEYWORDS
design science research, domain-specific modelling language, internet of things, service science, smart service systems

1 | INTRODUCTION

Services play a central role in our economies and society. By 2014, the service sector already accounted for about 78% of the Unites States' and 74% of the European Union's gross domestic product (World Bank, 2017). With products getting commoditized very fast in turbulent environments, many organizations have embraced the ideas of servitization, exploring new value propositions by offering "value-added" services alongside their products (Velamuri, 2013). While such services were at first simple and could be easily separated from products (eg, financial or maintenance services), since they have become more complex and tightly integrated. Taking a fundamentally new and inclusive view, where products only serve as transmitters, Vargo and Lusch (2004) conceptualize service as the general process of value cocreation through the application and integration of resources for the mutual benefit of service providers and consumers.

Besides these conceptual advances, service has changed thanks to transformational developments in information technology (IT) (Barrett, Davidson, Prabhu, & Vargo, 2015). These developments have given rise to new types of service and smart service systems (SSS), ie, resource configurations capable of learning, dynamic adaptation, and decision making (Lim & Maglio, 2018). One of the most recent advances with an impact on SSS is the internet of things (IoT) (Porter & Heppelmann, 2014, 2015; Yoo, Boland, Lyytinen, & Majchrzak, 2012). In the IoT, physical objects (ie products) are equipped with sensors, actuators, computing logic, and communication technology. This turns objects from passive tools into active smart things, which not only bridge the gap between the digital and the physical world but also are self-dependent actors in ever more situations (Oberländer, Röglinger, Rosemann, & Kees, 2018; Porter & Heppelmann, 2014; Yoo et al., 2012). Smart things advance the capabilities and scope of SSS as they monitor the physical environment, access local data, and immerse into individuals' everyday lives and into organizational routines (Beverungen, Müller, Matzner, Mendling, & Vom Brocke, 2017; Yoo, 2010).

As SSS and the IoT are emergent phenomena, the impact of the IoT on SSS yet needs to be explored. Only recently, Lim and Maglio (2018) and Beverungen et al. (Beverungen, Müller, et al., 2017) proposed definitions of SSS that account for the IoT, listing smart things as constitutive components. However, the academic discourse on technology-enabled service is fragmented, which is why theories and methods are often only understood by researchers from one discipline (Bardhan, Demirkan, Kannan, Kauffman, & Sougstad, 2010; Larson, 2016). Similar problems occur in industry, where service engineering and product development projects are insufficiently understood owing to their interdisciplinary nature (Bardhan et al., 2010; Slama, Puhlmann, Morrish, & Bhatnagar, 2015). Furthermore, the IoT has been discussed primarily with respect to technological challenges (Atzori, Iera, & Morabito, 2010; Kortuem, Kawsar, Sundramoorthy, & Fitton, 2010), from a business-to-business (B2B) and a business-to-consumer (B2C) perspective (Caputo, Marzi, & Pellegrini, 2016; Geerts & O'Leary, 2014; Oberländer et al., 2018), and with a focus on individual smart things (Püschel, Röglinger, & Schlott, 2016; Beverungen et al., 2017b). However, the role of smart things in broader contexts such as SSS is underresearched. Building the foundation for explanatory and design-led research to analyse and design SSS, a means for the conceptual modelling of SSS including novel IoT-enabled concepts is in high need.

The literature includes numerous service modelling approaches, ranging from technical languages, which focus on machine-to-machine communication, to conceptual ones that facilitate the communication among modellers and

model users (Alter, 2012b; Becker, Beverungen, & Knackstedt, 2010; Cardoso, 2013; Cardoso, Pedrinaci, & de Leenheer, 2013; OMG, 2017; Razo-Zapata, de Leenheer, Gordijn, & Akkermans, 2012). As has become evident during our study and as we will justify in the evaluation section based on selected conceptual modelling approaches, existing works neither fully account for the characteristics of SSS nor for concepts enabled by the IoT. The same holds for modelling approaches from the IoT domain, most of which focus on technical details, are restricted to distinct domains, or can only handle small-scale scenarios (Christoulakis & Thramboulidis, 2016; De, Barnaghi, Bauer, & Meissner, 2011; Meyer, Ruppen, & Magerkurth, 2013; Oberländer et al., 2018; Xu, Xu, Li, Lv, & Liu, 2012). Although the value of these approaches is indisputable, we do not know of any approach located at the intersection of the service and the IoT domains. Hence, our research aims to address this gap by focusing on the conceptual modelling of SSS. Our research question is as follows: *How should a conceptual modelling language for SSS look like?*

To answer this question, we developed a conceptual domain-specific modelling language (DSML) that reconstructs concepts and relationships from the SSS domain and provides an abstract and a concrete syntax. With modelling languages being valid design artefacts, we adopted the design science research (DSR) paradigm (Gregor & Hevner, 2013; Hevner, March, Park, & Ram, 2004) and followed Peffers, Tuunanen, Rothenberger, and Chatterjee's (2007) DSR methodology (DSRM). In the design and development phase, we adopted selected steps of Frank's (2013) DSML engineering method (DSMLEM). Our DSML incorporates knowledge from the SSS and IoT domains, experience gained through the modelling of simple and complex real-world scenarios from all functional IoT domains (Borgia, 2014), and the feedback of experts from different organizations.

Our study is organized as follows: first, we provide background on SSS and the IoT, before outlining our research method. After that, we introduce the DSML, including design objectives (DOs), abstract and concrete syntax, and semantics. Next, we demonstrate the DSML by presenting a complex real-world scenario centred around the Nest Learning Thermostat. We also report on our evaluation activities. We conclude by summarizing findings, implications, limitations, and by pointing to areas for further research.

2 | BACKGROUND

2.1 | Service, service systems, and SSS

Service is an interdisciplinary concept appearing in diverse contexts. Hence, no single definition has yet been accepted (Alter, 2012a; Rai & Sambamurthy, 2006; Spohrer & Maglio, 2008). Most definitions hold that service involves at least two entities with different roles (eg, service provider, customer, or participant), applying and integrating resources in an interactive and collaborative process to cocreate value for mutual benefit (Peters et al., 2016; Vargo & Lusch, 2016). In service research, the singular term "service" reflects the process of doing something beneficial for and in conjunction with some entity, rather than units of output as implied by the plural term "services" (Vargo & Lusch, 2008a). When using service in this paper, be it in singular or plural, we refer to distinct sets of interacting service systems and related resources involved in value cocreation. This complies with the inclusive meaning of the singular term "service" and goes beyond the reductionist use of the plural term "services."

Service research distinguishes between operand and operant resources. While operand resources are "passive" resources on which an act needs to be performed to produce an effect, operant resources are "active" resources employed to act on or in concert with other resources to cocreate value (Vargo & Lusch, 2004). Moreover, resources can be social, material, or a mixture of both (Leonardi, 2013). As we will discuss in the next section, the IoT enables a new perspective on resources as smart things can be seen as self-dependent actors in ever more scenarios (Oberländer et al., 2018). Such smart things are operant resources. Equipped with sensors, actors, computing logic, and communication technology, smart things extend traditional ideas of material agency, which so far only accounted for the way objects act when humans provoke it (Leonardi, 2013). In addition, individuals are sociomaterially entangled with IT-enabled resources (eg, smart phones), which serve as intermediaries between humans and other material resources (Oberländer et al., 2018).

Service systems are dynamic resource configurations, including people, organizations, information, and technology, connected with other service systems through value propositions (Maglio, Vargo, Caswell, & Spohrer, 2009). The boundaries of service systems are determined by the resources they can bring to bear. Value propositions are invitations to engage in service (Chandler & Lusch, 2015). If value propositions are accepted, service systems apply and integrate each other's resources to cocreate value (Beverungen, Lüttenberg, & Wolf, 2018). Hence, service is enacted through interacting service systems (Hamilton, 2004; Razo-Zapata et al., 2012; Vargo & Lusch, 2008a). As opposed to general systems, service systems are defined not only by relationships among resources but also by the fact that their properties and behaviours are more than the sum of the included resources (Maglio et al., 2009; von Bertalanffy, 1976). This is why service systems must include at least one operant resource to act upon other resources (Maglio et al., 2009). As service systems are resources configurations, they are resources themselves, a property that requires distinguishing between composed and atomic service systems. Composed service systems include other service systems (ie, at least two operant resources across all aggregation levels). The smallest service system is an individual offering knowledge and skills or a self-dependent smart thing that offers data and functions (Maglio et al., 2009; Oberländer et al., 2018).

Over the last decade, service has changed thanks to advances in IT, which have given rise to SSS (Barrett et al., 2015). Two major developments are the emergence of nested service system structures and smartness. As for the former, increased connectivity not only facilitates the interaction among service systems but also enables service systems to include other service systems (Yoo et al., 2012). Nested structures of service systems are also referred to as composed service systems in service research (Maglio et al., 2009) and as systems of systems in the IoT domain (Porter & Heppelmann, 2014). They enable novel value propositions by pooling resources and leveraging synergies (Nielsen, Larsen, Fitzgerald, Woodcock, & Peleska, 2015; Porter & Heppelmann, 2014). At the same time, the diversity of resources and potentially competing priorities make it hard to manage nested service systems (Becker, Beverungen, Knackstedt, Matzner, Müller, & Pöppelbuß, 2013; Nielsen et al., 2015).

Smartness, the second major development, often remains unspecified (Gretzel, Sigala, Xiang, & Koo, 2015). In the literature, different perspectives can be found: Ouyang et al. (2017), for example, state that smartness depends on digital technologies. For Fleisch and Thiesse (2007), smart means that humans delegate control tasks that they have performed themselves so far. When it comes to SSS, two literature streams can be distinguished. In the first stream, which has been discussed for years, smartness refers to dynamic adaptation, learning, and decision making, all of which is enabled by data analysis and self-x capabilities (Barile & Polese, 2010; National Science Foundation, 2014). Data can be used for transactional purposes (eg, collection, exchange, and storage) and for analytical purposes. Commonly, the latter is further distinguished into basic (ie, descriptive) and extended analytical (ie, diagnostic, predictive, and prescriptive) purposes (Allmendinger & Lombreglia, 2005; Porter & Heppelmann, 2014, 2015; Want, Schilit, & Jenson, 2015). Analytical data usage, in turn, enables basic self-x (eg, self-monitoring and self-diagnosis) and extended self-x capabilities (eg, self-optimization and self-configuration), which are needed for learning and adaptation (National Science Foundation, 2014). In the second literature stream, which has emerged only recently, smartness implies that service systems include smart things that may serve as boundary objects among service systems and bridge the gap between the digital and the physical world (Beverungen, Matzner, & Janiesch, 2017; Beverungen, Müller, et al., 2017; Ouyang et al., 2017). In that sense, SSS extend digital service systems that solely exist in the digital world (Beverungen, Matzner, & Janiesch, 2017).

As both streams view smartness from different angles, they entail disjoint definitions of SSS. Only recently, Lim and Maglio (2018) reconciled both streams through an integrated definition. Accordingly, we define SSS as service systems capable of learning, dynamic adaptation, and decision making (Medina-Borja, 2015), which include smart things and interact with other service systems (Allmendinger & Lombreglia, 2005; Beverungen, Müller, et al., 2017; Lim & Maglio, 2018; Wuenderlich et al., 2015). To realize learning, dynamic adaptation, and decision making, SSS leverage extended data analysis and self-x capabilities (Beverungen, Müller, et al., 2017; Kephart & Chess, 2003; National Science Foundation, 2014).

2.2 | Internet of things

Driven by the convergence of multiple technologies (eg, sensor technologies, miniaturization, mobile connectivity, artificial intelligence, and embedded systems), the IoT reflects the transformation of physical objects into smart things (Yoo et al., 2012). The IoT can be defined as the connectivity of physical objects equipped with sensors and actuators to the internet via data communication technology (Oberländer et al., 2018). It is commonly assumed that smart things can exist independently from IT, a view excluding devices such as personal computers or smart phones (Gartner, 2013; Mattern & Floerkemeier, 2010; Uckelmann, Harrison, & Michahelles, 2011). Two key features of the IoT are that smart things bridge between the physical and the digital world and that they are evolving into self-dependent actors (Oberländer et al., 2018).

To appropriately account for smart things in our DSML, we drew on Püschel et al.'s (2016) multilayer taxonomy (Figure 1), which groups the characteristics of smart things according to 10 dimensions and four layers. Building on established IoT technology stacks (eg, Porter & Heppelmann, 2014 ; Yoo, Henfridsson, & Lyytinen, 2010), these layers are thing layer, interaction layer, data layer, and service layer.

The thing layer, located at the bottom of the taxonomy, covers a smart thing's sensing and acting capabilities. Sensing capabilities enable smart things to access local data, eg, by observing the environment or by monitoring their internal status. Acting capabilities cover a smart thing's ability to communicate with or influence its environment directly or via intermediaries. The interaction layer marks the intersection of a smart thing's local (physical) representation and the digital world. Püschel et al. (2016) characterize interactions in terms of their direction, multiplicity, and partners. The interaction layer also accounts for the compatibility with other providers' offerings. The data layer covers relevant data sources and ways of data usage. Data sources include a smart thing's internal status (eg, battery level), context (eg, temperature or humidity), and usage (ie, frequency of use) as well as external data (eg, from the cloud). Data usage covers a smart thing's ability to use data for transactional or analytical purposes (Porter & Heppelmann, 2014). The service layer covers smart things' offline functionality and their main purpose. Offline functionality reflects to what extent a smart thing's functionality depends on a working internet connection. A smart thing's main purpose indicates whether the value it creates primarily relies on its representation in the physical world (eg, smart vacuum cleaners), on additional services (eg, fitness trackers), or on the integration into ecosystems (eg, the Nest Learning Thermostat). Details and examples can be found in Püschel et al. (2016).

While early smart things had simple sensors, actuators, and computing logic (eg, smart toothbrushes or -tagged workpieces), recent advances led to smart things with extended data analysis and self-x capabilities (Leonardi, 2013; Porter & Heppelmann, 2014). Oberländer et al. (2018) pose that ever more smart things are self-dependent actors.

	Dimension	Characteristics			
Service	Main Purpose	Thing-centric	Additional services	Ecosystem integration	
	Offline Functionality	Limited	None		
Data	Data Usage	Transactional	Analytical (basic)	Analytical (extended)	
	Data Source	Thing state	Thing context	Thing usage	Cloud
Interaction	Thing Compatibility	Proprietary	Open		
	Partner	User(s)	Business(es)	Thing(s)	
	Multiplicity	One-to-one	One-to-many	Many-to-many	
	Direction	Unidirectional	Bidirectional		
Thing	Acting Capabilities	Own	Intermediary		
	Sensing Capabilities	Lean	Rich		

FIGURE 1 Multilayer taxonomy of smart things (Püschel et al., 2016)

In line with the characteristics compiled by Püschel et al. (2016) and the insights shared by Oberländer et al. (2018), we distinguish between dependent and self-dependent smart things. In general, self-dependency reflects a system's ability to organize itself without any external force (Ashby, 1991; Batool & Niazi, 2017). Self-dependent smart things leverage extended data analysis and self-x capabilities to act and decide in line with goals and without external intervention (sometimes even without external triggers) and to learn by refining models of themselves. Moreover, self-dependent smart things take over decision-making tasks from humans and can be parameterized with goals (Encarnação & Kirste, 2005; Porter & Heppelmann, 2014; Rijsdijk & Hultink, 2009). In contrast, dependent smart things need triggers for every action and can neither learn nor be parameterized with goals (Rijsdijk & Hultink, 2009). However, based on their sensing capabilities, dependent smart things can use basic self-x functions explicitly (eg, self-monitoring of battery level) or implicitly as part of functions related to a smart thing's purpose in the physical world (eg, self-monitoring of speed and direction as parts of a smart vacuum cleaner's cleaning function). When introducing the abstract syntax of the DSML, we argue that the distinction between dependent and self-dependent also applies to software components, which we call digital hubs (Lea & Blackstock, 2014).

3 | RESEARCH METHOD

3.1 | DSR methodology

To facilitate the conceptual modelling of SSS, we propose a DSML. We decided in favour of designing a DSML compared with extending a general-purpose modelling language (GPML), since DSMLs enable increased modelling productivity, better coverage of the target domain, and they can be understood more easily by modellers and model users (Frank, 2013). An extended GPML would include many concepts that do not fit the target domain. However, when specifying our DSML, we adopted ideas from existing service- and IoT-related DSMLs as well as from GPMLs such as the Unified Modelling Language (UML) family (OMG, 2017) or the Fundamental Modelling Concepts (FMC) standard (FMC Modeling, 2017) wherever reasonable.

Our DSML consists of three components: abstract syntax, concrete syntax, and semantics (Nordstrom, Sztipanovits, Karsai, & Ledeczi, 1999; Wand & Weber, 2002). The abstract syntax covers concepts and relationships from the target domain in terms of a metamodel. Metamodels are (semi-) formal descriptions of how to build conceptual models (Eriksson, Henderson-Sellers, & Ågerfalk, 2013). The concrete syntax defines graphical and textual notational elements that enable representing models as diagrams (Mannadiar & Vangheluwe, 2010). The semantics specify how to interpret the concepts and relationships included in the abstract syntax (Bouhdadi, Balouki, & Chabbar, 2007). The key challenge being the reconstruction of concepts and relationships from the SSS domain, we primarily focused on the development and evaluation of the abstract syntax. We also proposed a concrete syntax to support the modelling of SSS. This, however, was of secondary importance, as most notational elements could be taken from the literature.

As DSMLs are models (ie, statements about relationships among constructs), they are valid design artefacts (March & Smith, 1995). Thus, our study follows the DSR paradigm (Gregor & Hevner, 2013), and we adopted Peffers et al.'s (2007) DSRM to structure our research process. The DSRM includes six phases: problem identification, definition of DOs, design and development, demonstration, evaluation, and communication. As the DSRM is a generic research process, it should be complemented through a research method that fits the artefact type. Therefore, we adopted Frank's (2013) DSMLEM, which has proven useful in other DSML projects (eg, Heise, Strecker, & Frank, 2014 ; Heß, Kaczmarek, Frank, Podleska, & Taeger, 2015 ; Strecker, Frank, Heise, & Kattenstroth, 2012). Being a stand-alone research method, the DSMLEM partially overlaps with the DSRM. Thus, we only used some steps. Below, we outline how we conducted the DSRM (except for the communication phase) and which steps we selected from Frank's (2013) DSMLEM. When developing our DSML, we did not perform all steps strictly sequentially, but iteratively to reflect the idea of design science being a search process and of the iterative refinement of DSMLs (Frank, 2013; Hevner et al., 2004). Figure 2 provides an end-to-end overview of our research design.

The first DSRM phase requires defining the research problem and justifying the value of the proposed artefact. As outlined in the introduction, SSS are an emergent phenomenon not only with enormous economic potential but also affected by the IoT. In academia and industry, a means for the conceptual modelling of SSS that accounts for concepts enabled by the IoT is in high need. Our DSML addresses this gap and serves as basis for future explanatory and design-led research as well as for the analysis and design of SSS.

The second DSRM phase requires specifying domain-specific DOs that support the assessment of the developed artefactand the comparison with competing artefacts. DOs should be derived from the research problem and backed by the literature (Peffers et al., 2007). With SSS being our target domain, we derived three DOs covering the variety of resources involved in SSS (DO1), nested structures of service systems (DO2), and smartness as a central idea of SSS (DO3). All DOs are backed by the literature and have been confirmed by industry experts.

In the design and development phase, the artefact is built (Peffers et al., 2007). As the DSRM does not offer guidance on how to design DSMLs, we adopted selected steps of Frank's (2013) DSMLEM. The DSMLEM includes seven steps: (1) clarification of scope and purpose, (2) analysis of general requirements (GRs), (3) derivation of specific requirements (ie, DOs), (4) specification of the modelling language (ie, abstract syntax), (5) provision of a modelling notation (ie, concrete syntax), (6) development of a modelling tool, and (7) evaluation and refinement. As steps (1), (3), and (7) are covered by the DSRM, we focused on steps (2), (4), and (5). The primary focus of our research being the reconstruction of concepts and relationships related to SSS, we refrained from developing a modelling tool (6). This step is also marked as optional by Frank (2013).

After developing the DSML, we conducted a demonstration and evaluation. In line with our primary research focus, we focused on a content-oriented evaluation. The realization of our DSML as a formal metamodel that can be implemented in modelling tools is subject to future research. First, we modelled numerous simple and complex real-world scenarios, which are intermediate evaluations (Appendix B). The complex scenarios included the Nest Learning Thermostat (NEST, 2017), Rolls-Royce's Engine Health Management (Rolls-Royce, 2018a, 2018b), and a publicly funded research project on IoT-enabled intralogistics in hospitals. Furthermore, we conducted semistructured interviews with experts from various organizations (Myers & Newman, 2007). The interviews included a review of the DOs, the abstract and concrete syntax, and the modelling of a simple and a complex real-world scenario. We also reflected and incorporated the experts' feedback, which can be found in the evaluation section and in Appendix E. Based on the data collected in the interviews and on general considerations, we discussed the DSML against Frank's (2013) GRs for DSMLs (Appendix F) and against the DOs (Siau & Rossi, 1998). In the evaluation, we also compared the DSML with selected conceptual modelling approaches to assess whether it outperforms these competing artefacts regarding the DOs. For demonstration purposes, we included the complex scenario related to the Nest Learning Thermostat, which was also discussed with the industry experts.

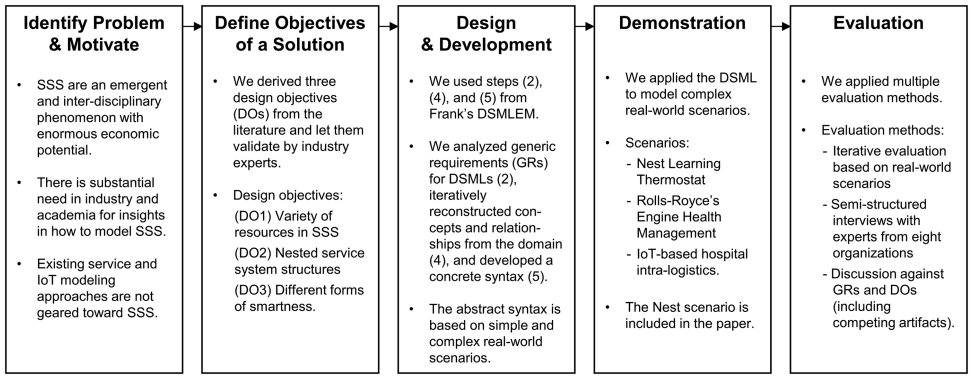


FIGURE 2 End-to-end overview of the research design based on the design science research methodology (DSRM) and the domain-specific modelling language engineering method (DSMLEM)

3.2 | DSML engineering method

We selected steps (2), (4), and (5) of Frank's (2013) DSMLEM to structure the design and development phase of the DSRM. Step (2) requires defining GRs independent from the target domain, which should be met by any DSML. As Frank (2013) recommends reusing GRs, we adopted his catalogue as it has been successfully used in other DSML projects (Heise et al., 2014; Heß et al., 2015; Strecker et al., 2012) and—to the best of our knowledge—no other catalogue is as comprehensive. Frank's (2013) GRs cover familiarity (GR1), invariability (GR2), level of detail and abstraction (GR3), and mapping of language concepts (GR4). As these GRs partially overlap with general DSR evaluation criteria, we used them as part of the evaluation.

Step (4) of Frank's (2013) DSMLEM covers the reconstruction of concepts and relationships from the target domain and the compilation of an abstract syntax in the form of a metamodel. As a first step, Frank (2013) recommends iteratively modelling current and future scenarios. This should initially involve simple scenarios, later moving on to complex and potentially fictitious scenarios. We first reviewed the service- and IoT-related literature. This led to an initial set of concepts and relationships (ie, service, resource, relationship, and service system). Next, we further developed the DSML based on real-world scenarios, uncovering novel concepts and relationships. We selected our scenarios in line with Borgia's (2014) functional IoT domains (ie, industrial, smart city, and health well-being) so as to ensure that—as far as possible—the identified concepts were comprehensive. For example, after deriving initial concepts from the literature, we mapped them to the Nest Learning Thermostat scenario. This led to further resources (eg, physical environment) and relationships (ie, observation and parameterization). After each application of our DSML, we abstracted from case-specific peculiarities to keep the metamodel parsimonious. In addition, we reflected on new concepts, first, by studying the literature and, second, by subjecting all elements to the scrutiny of industry experts. We investigated further scenarios until all authors agreed that the abstract syntax had reached conceptual saturation, ie, no new concepts and relationships were being identified (Briggs & Schwabe, 2011). The scenarios that influenced our DSML and corresponding changes to our DSML are listed in Appendix B.

At the same time, we developed the concrete syntax with notational elements. This activity corresponds to step (5) of Frank's (2013) DSMLEM. As concrete syntax should be based on commonly accepted and easily understandable notational elements—for instance, taken from existing DSMLs and GPMLs—we reused existing notational elements wherever reasonable (Frank, 2013). For example, we looked into UML's class and use-case diagrams as well as into the concrete syntax of service- and IoT-related DSMLs (eg, Becker et al., 2010; Oberländer et al., 2018; Razo-Zapata et al., 2012). In the few cases where a notational element could not be taken from the literature, we devised new elements and validated them during the expert interviews. As the concrete syntax was of secondary importance for our research, we made pragmatic design decisions. That is why our concrete syntax needs to be refined and further evaluated in the future.

4 | SPECIFICATION OF THE MODELLING LANGUAGE

In this section, we introduce our DSML for SSS. Below, we first outline the DSML's target audience and intended purpose of use. We then outline and justify the DOs. Finally, we present the abstract and concrete syntax. Our textual elaborations reflect the semantics. The DSML, as presented below, does not only reflect the results of our design process, but also reflects the expert feedback received during the evaluation (Appendix E).

4.1 | Target audience and intended purpose of use

Our DSML is targeted at researchers and practitioners involved in the conceptual modelling of SSS. In keeping with the interdisciplinary nature of SSS, modellers and model users typically come from various disciplinary backgrounds and diverse corporate functions. Our DSML bridges the gap between these disciplines, offering a unified nomenclature for concepts and relationships from the SSS domain. Accordingly, the DSML is intended to be used for the

conceptual modelling of SSS with a focus on services, service systems, resources, and relationships. Conceptual modelling is conducted early in product development, systems design, or service engineering projects to ensure that stakeholders share the same understanding. Models created with our DSML make SSS tangible and facilitate the communication among modellers and model users (Wand & Weber, 2002). They can also serve as input for technical specifications required for the implementation of SSS. Owing to its focus on conceptual modelling, the DSML abstracts from technical features. Nevertheless, it reuses concepts, relationships, and notations wherever possible to maximize the compatibility with other conceptual and technical service- and IoT-related modelling approaches.

4.2 | Design objectives

To guide the construction and evaluation of our DSML, we formulated DOs that a DSML for SSS should meet (Peffers et al., 2007). As opposed to Frank's (2013) GRs, DOs are domain specific. We derived the DOs from our SSS definition and discussed them with the experts involved in the evaluation. Our SSS definition, which we introduced in the background section, engenders multiple requirements. First, with SSS including resources ranging from individuals to smart things, a DSML should cope with this diversity. Second, with SSS involving interactions among service systems, a DSML must be able to handle nested service system structures—including the dynamic adaptation of structure and behaviour through the integration of new resources and the withdrawal of existing ones over time. Third, as smartness is a vital property of SSS, a DSML should enable capturing different facets of smartness related to learning and decision making beyond the inclusion of smart things. These key requirements led us to specify the following DOs:

4.2.1 | DO1: A DSML for SSS should capture the diversity of the resources and relationships involved

Service research investigates how service systems, ie, dynamic resource configurations, cocreate value (Böhmann, Leimeister, & Möslein, 2014; Maglio et al., 2009; Vargo, Maglio, & Akaka, 2008). Resources are typically tied to the roles of individuals and organizations (Vargo & Lusch, 2016). However, advances in IT increase the diversity of resources involved in SSS. For example, the IoT leads to self-dependent smart things, which can be considered as equal to individuals in ever more scenarios (Leonardi, 2013; Porter & Heppelmann, 2014). The IoT also enables new relationships (Oberländer et al., 2018; Perera, Zaslavsky, Christen, & Georgakopoulos, 2014b). For example, smart things can use their sensing capabilities to observe the environment and, in response, trigger interactions with individuals or other smart things (Borgia, 2014). In addition, self-dependent smart things may be parameterized through goals instead of requiring triggers for every single action (Encarnação & Kirste, 2005).

4.2.2 | DO2: A DSML for SSS should capture nested service system structures

Owing to advances in IT, service systems are strongly connected. In many cases, value cocreation occurs in nested service systems (Porter & Heppelmann, 2014). For example, a seemingly straightforward service, such as the intelligent management of parking lots in smart cities via a cell phone app with cashless payment, involves many service systems. Moreover, the potential of the IoT can only be tapped if smart things are not considered in isolation but embedded in broader contexts (Beverungen, Müller, et al., 2017; Porter & Heppelmann, 2014; Püschel et al., 2016). However, nested service systems increase coordination and management complexity (Becker et al., 2013; Matook & Brown, 2017). One reason is that the set of resources involved is not static over time, meaning that resources can enter or leave (Bennett & Lemoine, 2014). Other reasons are competing priorities, limited capacities, or the lack of a central coordination authority in distributed value cocreation (Nielsen et al., 2015).

4.2.3 | DO3: A DSML for SSS should capture different facets of smartness. Smartness is vital for SSS

While one literature stream argues that smartness results from the inclusion of smart things (Beverungen, Matzner, & Janiesch, 2017; Beverungen, Müller, et al., 2017), the other links smartness with dynamic adaptation, learning, and decision making (National Science Foundation, 2014). As the inclusion of smart things is already covered by DO1, this DO focuses on the second view, which requires data analysis and self-x capabilities to be realized. As mentioned, data can be used for transactional and analytical purposes (Porter & Heppelmann, 2014; Want et al., 2015). Extended data analysis capabilities enable self-x capabilities, which in turn are the basis for learning and decision making as well as for self-dependent smart things and digital hubs (National Science Foundation, 2014). Self-dependent smart things and digital hubs, in turn, are vital for realizing learning, dynamic adaptation, and decision making on the level of SSS.

4.3 | ABSTRACT SYNTAX

We specified the abstract syntax in terms of a semiformal metamodel (Eriksson et al., 2013). To facilitate the understanding of the abstract syntax, the metamodel has a modular structure. We first introduce the core metamodel, including service, service system, resource, and relationship as concepts. We then explain these concepts by unfolding subconcepts. To that end, we follow the practice of specifying metamodels through simplified UML class diagrams (Eriksson et al., 2013). Next, we present an integrated version of the metamodel. As the connections between resources and relationships as well as between the subconcepts of service systems are complex, we present details in Appendices C and D. Furthermore, Appendix A includes detailed design rationales that justify why and how concepts from the literature were included in the final metamodel. Having introduced the abstract syntax, we define a structural and a behavioural. Both views are backed by General Systems Theory (Boulding, 1956; Bertalanffy, 1976; OMG, 2017) and facilitate the modelling of SSS through different models.

4.3.1 | Core metamodel

The core metamodel includes four concepts: service, service system, resource, and relationship (Figure 3). In our DSML, service reflects a distinct set of service systems and related resources involved in value cocreation. Each service has a purpose and involves at least two service systems characterized by value propositions (Maglio et al., 2009). In keeping with their property of being invitations to engage in service (Chandler & Lusch, 2015), value propositions depend on the related service system, not on its engagement in service. If accepted, value propositions relate to one or more services, reflecting which and how resources are involved. Therefore, we modelled value propositions as attributes of service systems and as association class attached to the aggregation metarelationship between service system and service. As resource configurations, service systems consist of resources. Value cocreation becomes evident not only in terms of the value propositions of involved service systems but also—as a collaborative and interactive process—in the relationships among the resources associated with these service systems (Spohrer & Maglio, 2008). Furthermore, relationships can connect resources from one and different service systems (Edvardsson, Tronvoll, & Gruber, 2011). Likewise, resources can be part of many service systems, eg, an individual can be involved in a smart home and a smart shopping service system. Moreover, smart things can serve as boundary objects between service systems (Beverungen, Müller, et al., 2017). As relationships play different roles in different services, the metamodel includes the "relationship sequence" association class attached to the aggregation metarelationship between service and relationship. This association class allows for a simple temporal or logical ordering of relationships involved in distinct services.

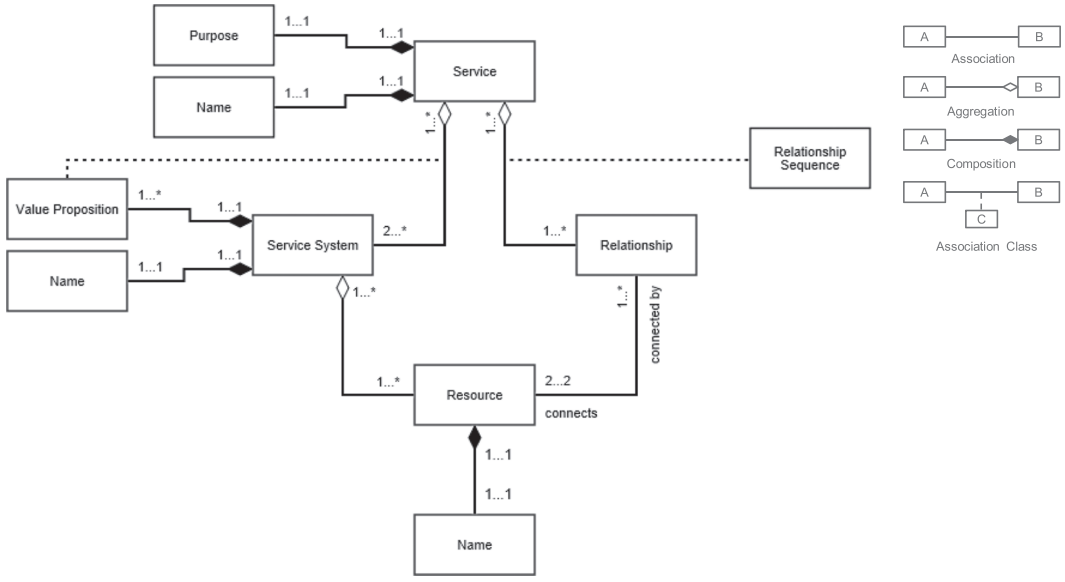


FIGURE 3 Core metamodel of the domain-specific modelling language (DSML)

4.3.2 | Resources

The DSML covers the sociotechnical spectrum of resources, ranging from entirely social to entirely technical resources (Oberländer et al., 2018; Peters et al., 2016). Accordingly, we split resources into individuals, smart things, digital hubs, and the physical environment (Figure 4).

Individuals are humans involved in value cocreation, during which they share and apply knowledge and skills (Böhmann et al., 2014; Maglio & Spohrer, 2008; Peters et al., 2016). Individuals can also have properties (eg, blood pressure) that can be observed by smart things. Individuals can take the roles of service customers and participants. Service customers directly benefit from their engagement in service (Alter, 2008, 2012b), whereas service participants benefit indirectly. Furthermore, individuals are operant resources and, as such, can constitute service systems (Maglio et al., 2009).

Smart things are physical things equipped with sensors, actuators, computing logic, and connectivity (Oberländer et al., 2018). They exist in the physical world and have a representation in the digital world (Beverungen, Müller, et al., 2017; Medina-Borja, 2015). Smart things may serve as boundary objects between service systems (Beverungen, Müller, et al., 2017; Star, 2010). Digital hubs are software components that only exist in the digital world (Lea & Blackstock, 2014). Smart things and digital hubs can perform various tasks, including the collection, storage, exchange, and analysis of data. Tasks may include the coordination and parameterization of other resources or refer to self-x capabilities (Lea & Blackstock, 2014; Porter & Heppelmann, 2014; Yoo, 2010). Smart things can also perform tasks in line with their purpose in the physical world (eg, smart cars can be driven) (Püschel et al., 2016). Just like individuals, smart things and digital hubs can have properties. Properties of smart things relate to their representation either in the physical (eg, colour) or in the digital world (eg, battery status). Accordingly, they need to be captured via observations or interactions. By contrast, the properties of digital hubs only relate to the digital world (eg, power on/off) and must be accessed via interactions.

Smart things and digital hubs dispose of data and functions (Atzori et al., 2010; Kortuem et al., 2010). While data are collected, exchanged, stored, or analysed, functions actively process or analyse data. Functions are used to model tasks that smart things perform in the physical or the digital world (eg, the cleaning function of a smart vacuum cleaner) or tasks that digital hubs perform in the digital world (eg, forecasting of stock prices or recommending books).

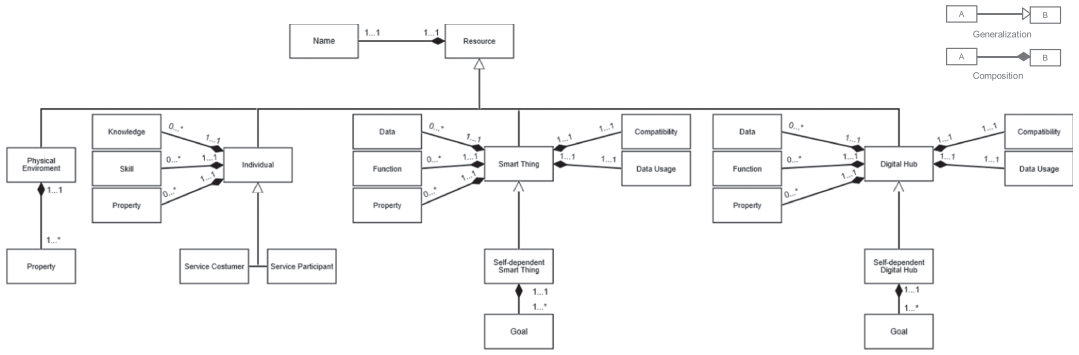


FIGURE 4 Detailed metamodel for "Resources"

Functions can refer to self-x capabilities implicitly (eg, self-monitoring as part of a smart vacuum cleaner's cleaning function) or explicitly (eg, a function dedicated to self-optimization). In self-dependent smart things and digital hubs, extended self-x functions are also running in the background, which is why they only need to be modelled if deemed appropriate by the modeller. In our DSML, data collection and exchange are modelled via observations and interactions (eg, the temperature outside must be observed and sent to a digital hub via an interaction), not via functions.

In line with their nature, we distinguish between dependent and self-dependent smart things and digital hubs (Batool & Niazi, 2017; Beverungen, Müller, et al., 2017; Kephart & Chess, 2003; National Science Foundation, 2014). Self-dependent smart things and digital hubs can act and decide in line with goals, without external intervention (sometimes even without external triggers), and they can learn by refining models of themselves, both enabled by extended data analysis and self-x functions. "Without external intervention" does not exclude interactions with other resources, which are necessary to perform a function or achieve goals. In the SSS domain, external intervention or trigger refers to interactions with all kinds of resources. As we will discuss below, self-dependent smart things and digital hubs can be parameterized by goals, and they can parameterize other self-dependent smart things and digital hubs. Although dependent smart things and digital hubs require external triggers for any action, cannot learn, and cannot be parameterized with goals, their functions may include basic self-x components. Self-dependent smart things and digital hubs are operant resources and can constitute service systems. An example of a self-dependent smart thing is an autonomous car that drives without human intervention, learns from interactions with other cars, and may pick up an individual after work without an external trigger. In the metamodel, we omit the characteristic "dependent" for smart things and digital hubs.

Apart from the distinction into dependent and self-dependent smart things and digital hubs, compatibility and data usage are important attributes of smart things and digital hubs (Püschel et al., 2016). Compatibility, which can be proprietary or open, refers to the ability of smart things and digital hubs to communicate and collaborate (Böhmann, Leimeister, & Möslein, 2018; Henfridsson & Lindgren, 2010; Püschel et al., 2016; Yoo et al., 2010). Proprietary means that a proprietary or private platform provides limited or no access to the data and functions of a smart thing or digital hub. Moreover, proprietary platforms only support smart things or digital hubs from one provider or trusted partners. Open means full, or only slightly limited, compatibility with third-party components.

Data usage reflects whether a smart thing or digital hub uses data for transactional or analytical purposes. Transactional usage covers the collection, storage, and exchange of data; the processing of simple interactions; and the execution of simple tasks that may refer to a smart thing's purpose in the physical world (Püschel et al., 2016). For example, the smart shower Eva Drop uses collected data (ie, current water temperature) to adjust the water flow. Analytical data usage augments transactional usage through basic and extended data analysis capabilities (Borgia, 2014; Porter & Heppelmann, 2014; Püschel et al., 2016). Analytical data usage can relate to the ordinary functions of smart things (eg, comparison of data from multiple sensors) and digital hubs (eg, forecasting of stock prices) or to

self-x functions (eg, self-diagnosis, self-optimization, or self-configuration). Self-dependent smart things and digital hubs always use data analytically.

The physical environment is passively involved in value cocreation. Smart things and individuals can observe properties of the physical environment (Borgia, 2014). As shown in Appendix C, the physical environment must relate to at least one smart thing or one individual via an observation and must have at least one property (eg, temperature or humidity). When used in concrete models, the environment can refer to the environment at large or to a specific object (eg, a nonsmart product).

4.3.3 | Service systems

The metamodel includes four subconcepts for service systems: atomic service systems, atomic SSS, composed service systems, and composed SSS (Figure 5). A description of how service systems and resources are related is shown in the integrated metamodel (Figure 7). Further details on how the subconcepts of service systems are connected are included in Appendix D.

Atomic service systems do not include other service systems. An individual offering knowledge and skills or a self-dependent digital hub offering functions and data are the smallest forms of atomic service systems. Individuals or self-dependent digital hubs that interact with other dependent smart things or digital hubs are atomic service systems, too. Self-dependent smart things are atomic SSS (Oberländer et al., 2018). Atomic SSS contain one self-dependent smart thing, may include arbitrarily many dependent smart things and digital hubs, but must not include individuals or further self-dependent smart things or digital hubs. An example is a smart thermostat that reconfigures temperature settings and heating times without external trigger when noticing that individuals repeatedly turn up the heating in the evening.

Composed service systems include at least one further service system (Ackoff, 1971; Maglio et al., 2009; Nielsen et al., 2015), leading to nested service system structures. With composed service systems being complex, the metamodel features a modular architecture, meaning that composed service systems can be decomposed (Baldwin & Clark, 2000; Shaw & Holland, 2010). We distinguish between composed service systems and composed SSS. Composed service systems include at least one further service system. One example is a group of people that cooks and prepares dinner together (Maglio et al., 2009). By contrast, composed SSS include at least one further service system or SSS. An example is a household where a smart washing machine (self-dependent smart thing) interacts with an individual and with the manufacturer's e-commerce platform (dependent digital hub). The washing machine monitors the level of detergent and continuously adjusts the minimum threshold by monitoring both washing behaviour and the delivery time for detergent. When the threshold is surpassed, the washing machine asks the individual whether they would like to buy detergent or whether the machine should order it from the e-commerce platform. Here, the smart washing machine serves as boundary object between the household (composed SSS) and manufacturer (atomic SSS).

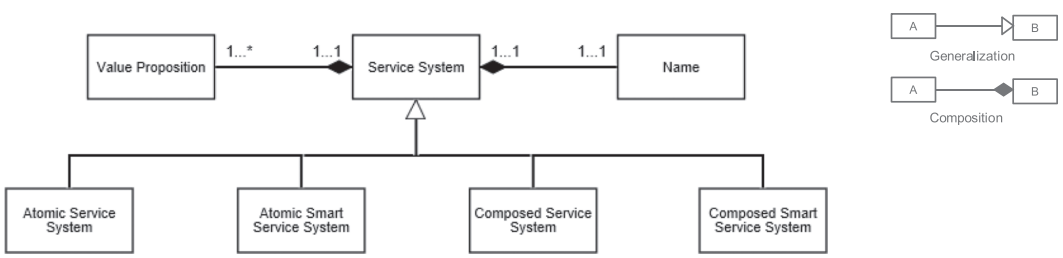


FIGURE 5 Detailed metamodel for "Service Systems"

4.3.4 | Relationships

Resources are connected through relationships. To model SSS, we distinguish between interaction, observation, and parameterization (Figure 6). As the example of the physical environment has shown, resources cannot be connected arbitrarily. Admissible connections depend on the involved subconcepts of resources and relationships. We included a detailed metamodel that comprises all connection rules in Appendix C.

Interactions are the most common relationship. Generally, interactions involve the coming together of resources that engage in some kind of exchange (Oberländer et al., 2018; Suchman, 2009). Interactions occur when resources exchange data, trigger functions, or notify about events. Interactions can have data and function attributes, which, if used, should comply with the data and functions of the interacting resources. Interactions can also have properties when accessing the properties of digital hubs or smart things. Interactions take place at distinct points in time. In cases where an interaction is a request, we assume the corresponding reply needs not be modelled separately (Kees, Oberländer, Röglinger, & Rosemann, 2015). That is, interactions can be unidirectional or bidirectional (Püschel et al., 2016). When modeling real-world examples, we noticed that repeated and nonrepeated interactions occur. For example, smart things can update digital hubs about their status in predefined intervals, while they may notify service customers about events only once (eg, the achievement of a fitness goal). The metamodel covers this property by means of the "repetitiveness" attribute. Furthermore, hierarchical and nonhierarchical interactions need to be distinguished (Cilliers, 2001; Maglio et al., 2009; Powell, 1990). The metamodel covers this property via the "coordination mode" attribute. Nonhierarchical interactions build on the assumption that resources are peers that negotiate how to coordinate their activities and agree on interaction routines (Becker et al., 2013). Cilliers (2001) emphasizes that many service systems feature hierarchical structures, implying hierarchical interactions where resources control other resources, with or without granting degrees of freedom. Hierarchical interactions that grant degrees of freedom are referred to as parameterizations (Encarnação & Kirste, 2005). For example, smart homes usually have a central digital hub that controls connected smart things in a hierarchical manner. With SSS involving multiple smart things, an uptake of decentral structures can be expected, for example, in smart grids (Kolokotsa, 2016).

Observations involve the acquisition of data from primary sources via sensing capabilities (Perera, Zaslavsky, Christen, & Georgakopoulos, 2014a; Streitz et al., 2005). In the SSS context, smart things and individuals can observe the properties of other resources, ie, individuals, other smart things, or the physical environment. Since digital hubs have no physical representation, they can neither observe nor be observed. Their properties need to be accessed through interactions. Self-observation (eg, the detection of a low detergent level by a smart washing machine)—meaning the ability of smart things or digital hubs to monitor their internal status—is not modelled via observations since relationships connect different resources. Such self-x functions are covered via the data and function attributes

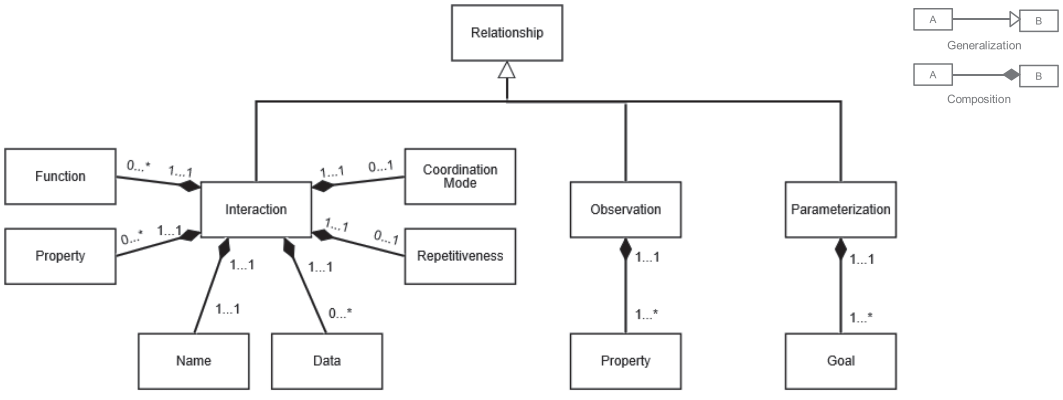


FIGURE 6 Detailed metamodel for "Relationships"

of smart things and digital hubs. In the case of self-dependent smart things and digital hubs, this can also be covered via implicitly modelled self-x functions running in the background.

Parameterization means that resources agree on goals such that one of the two resources commits to achieve these goals. Accordingly, parameterizations have goal attributes that should comply with the goals of the committed resource. Goals parameterize how the committed resource acts and decides during a specific period of time (Encarnação & Kirste, 2005). Parameterizations are closely related to the management by objectives paradigm (Drucker, 2009). In contrast to interactions that occur at distinct points in time and have a transactional and functions-oriented character (eg, turn on/off, notification, update, or request), parameterization refers to a period and focuses on coordination and control, granting degrees of freedom for own decisions. Thereby, only self-dependent smart things and digital hubs can be parameterized. In addition, individuals and self-dependent smart things as well as digital hubs can parameterize other resources. For example, the Nest Learning Thermostat features different modes users can choose. To realize these modes, the Nest Thermostat does not only interact with connected smart home devices but also parameterizes them as far as possible. Possible forms of parameterizations are shown in Appendix C.

4.3.5 | Integrated metamodel

Figure 7 shows the integrated metamodel that includes all concepts and relationships. For reasons of clarity, it leaves out the detailed connection between resources and relationships (Appendix C), the detailed connection among the subconcepts of service systems (Appendix D), and attributes. Most importantly, the integrated metamodel specifies the general connection between service systems and resources.

4.4 | Structural and behavioural views on the metamodel

In line with General Systems Theory, our DSML differentiates between a structural and a behavioural view of the metamodel (Bertalanffy, 1976; Kurpjuweit & Winter, 2007). Each view is geared toward a specific modelling goal, includes partially different concepts and relationships, and is represented by different model types. Below, we introduce both views conceptually. Their application is detailed in the demonstration section.

The structural view focuses on the interplay of service systems, resources, and relationships as well as on their grouping into services. It shows which service systems and relationships contribute to which service, which resources

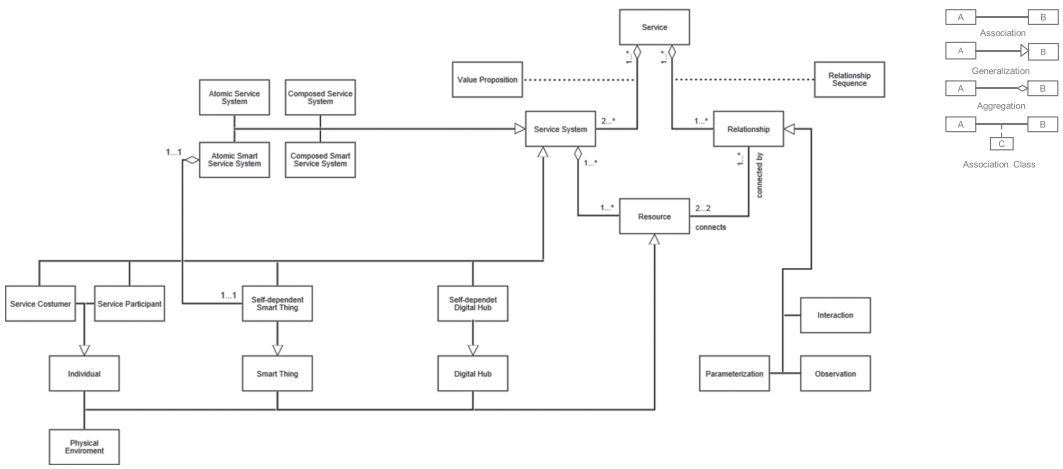


FIGURE 7 Integrated metamodel

are involved in which service systems, which smart things serve as boundary objects, how resources and service systems lead to nested structures, and how service systems and resources are connected via relationships. When applied in practice, the structural view leads to a single *Integrated Service System Model*, which provides an integrated picture of all services, service systems, resources, and relationships involved.

The behavioural view focuses on the process of value cocreation. As opposed to the structural view, the behavioural view does not lead to a single integrated model, but to a *Service Description Model* per service. Service Description Models are list-like textual descriptions indicating how the value propositions of the involved service systems become manifest in the form of temporally or logically ordered relationships. The ordering of relationships is enabled through the association class "relationship sequence" (Figure 7). To reveal the process of value cocreation, Service Description Models emphasize the attributes of concepts and relationships more than is customary in the Integrated Service System Model. The detailed inclusion of attributes would unnecessarily inflate the Integrated Service System Model, and so Service Description Models not only complement the structural view but also increase the level of detail.

4.5 | Concrete syntax

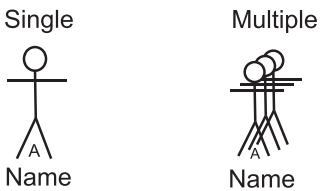
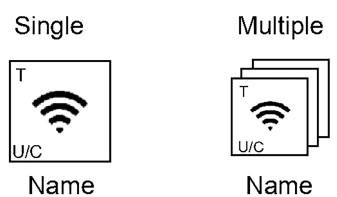
Table 1 shows the concrete syntax of our DSML, including graphical and textual notational elements. It also includes references to justify from which sources notational elements were taken. For most elements, we could draw from existing DSMLs and GPMLs. This, we hope, will support intuitive usage and increase familiarity among modellers and model users. For example, we adopted the symbols used to represent individuals, resources, and service systems from UML class and use-case diagrams (OMG, 2017) and from the FMC family (FMC Modeling, 2017). We also adopted the symbol for smart things from Oberländer et al. (2018) and Slama et al. (2015) and the notation for service from service modelling approaches (Becker et al., 2010; Razo-Zapata et al., 2012). Regarding the few concepts for which no notational elements were available (ie, observation and parameterization), we created new ones and validated them during the evaluation. For some elements (eg, smart thing, individual, and digital hub), we included group symbols, so-called multiobjects. This is useful for complex scenarios, as multiobjects can serve as placeholders. This complies with the nature of SSS, where resources and service systems may not be known at design time but enter or leave at runtime.

5 | DEMONSTRATION

To demonstrate the DSML, we illustrate how we used it to model smart home services enabled by the Nest Learning Thermostat. We chose this scenario as it requires the use of almost all concepts of our DSML. In addition, the Nest Learning Thermostat—which is one of the key resources of this scenario—is very popular in the IoT domain, and many sources are publicly available (NEST, 2017; Oberländer et al., 2018). Thanks to its growing popularity, many companies are seeking to collaborate with Nest.

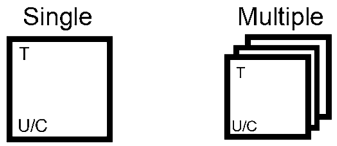
Figure 8 shows the Integrated Service System Model of the Nest scenario. It includes the following services: Energy Saving Service, Coming Home Service, and Rush Hour Reward Service. On the highest level of aggregation, it includes seven service systems (ie, Connected Driving, If This Then That [IFTTT], Smart Home, Nest, Bank, Energy Provider, and Meteorological Office) and related resources and relationships. The most central resource, the Nest Learning Thermostat, is a self-dependent smart thing that connects individuals, smart things, and digital hubs such that they optimize the temperature and other settings of private homes and businesses (NEST, 2017). In our example, the Nest Thermostat has relationships with all other resources of the smart home service system (eg, Smart Home Devices, House Owner, and Living Space). It also serves as a boundary object between the smart home and the Nest service system. Hence, it is modelled on the boundaries of both service systems and linked to the Nest Cloud. Via the Nest Cloud, the thermostat is connected with resources of other

TABLE 1 Concrete syntax

	Name	Notational Element	Definition	Selected References (Concept)	Selected References (Notational Elements)
Resource	**Individual**	 $A \in \{C;P\}$	Service customers (C) are individuals involved in value cocreation with direct benefits. Service participants (P) are individuals involved in value cocreation without direct benefits.	- Alter (2008) - Alter (2010) - Alter (2012b) - Böhmann et al. (2014) - Maglio et al. (2009)	- Fundamental Modelling Concepts (FMC Modeling, 2017) - Unified Modelling Language (OMG, 2017) - Technical Architecture Modelling (FMC Modeling, 2017)
	Smart Thing	 $T \in \{D;S\}$ $U \in \{T;A\}$ $C \in \{P;O\}$	Smart things are physical objects equipped with sensors, actuators, computing logic, and connectivity. Smart things can serve as boundary objects among service systems. *Dependent (D)*: Need for external triggers for any action, no learning capabilities, cannot be parameterized, and may use basic self-x functions. *Self-dependent (S)*: Ability to act and to decide in line with goals, without external intervention (sometimes even without triggers), *and* to learn by refining internal models, both enabled by extended data analysis and self-x functions.	Beverungen, Müller, et al. (2017) - Oberländer et al. (2018) - National Science Foundation (2014) - Porter and Heppelmann (2014) - Star (2010) - Yoo et al. (2010)	- Fundamental Modelling Concepts (FMC Modeling, 2017) - Oberländer et al. (2018) - Technical Architecture Modelling (FMC Modeling, 2017)

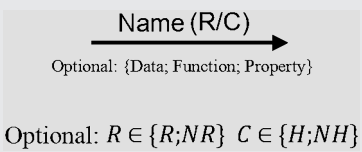
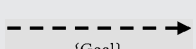
(Continues)

TABLE 1 (Continued)

Name	Notational Element	Definition	Selected References (Concept)	Selected References (Notational Elements)
		Proprietary (P): No, or severely constrained, compatibility with components from other providers. *Open (O)*: Full, or only slightly limited, compatibility with components from other providers. *Transactional (T)*: Operational collection, storage, and exchange of data as well as processing of simple interactions and execution of simple tasks. *Analytical (A)*: Capabilities for analyzing data in a descriptive, diagnostic, predictive, or prescriptive manner. Self-dependent smart things process data analytically. *Notational comment: Smart things should modeled on the boundaries of service systems if they serve as boundary objects.*		
Digital hub	 $T \in \{D;S\}$ $U \in \{T;A\}$ $C \in \{P;O\}$	Digital hubs are software components without a physical representation. *Dependent (D) vs Self-dependent (S)*: see Smart Thing. *Proprietary (P) vs Open (O)*: see Smart Thing. *Transactional (T) vs Analytical (A)*: see Smart Thing.	- Lea and Blackstock (2014) - Porter and Heppelmann (2014) - Yoo et al. (2010)	- Fundamental Modelling Concepts (FMC Modeling, 2017) - Unified Modelling Language (OMG, 2017) - Technical Architecture Modelling (FMC Modeling, 2017)

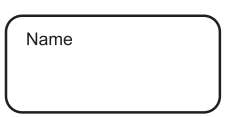
(Continues)

TABLE 1 (Continued)

	Name	Notational Element	Definition	Selected References (Concept)	Selected References (Notational Elements)
	Physical environment		The physical environment represents the entirety of environmental objects. In line with the used name, it can either refer to the environment at large or to a distinct environmental object that does not possess own sensing and acting capabilities.	- Borgia (2014) - Perera et al. (2014a) - Streitz et al. (2005)	- Fundamental Modelling Concepts (FMC Modeling, 2017)
Relationship	**Interaction**		Interactions take place when two resources come together to engage in exchange. They occur at distinct points in time and are transactional in nature. Interactions are equipped with a name. They can be annotated with data exchanged, functions used, and/or properties accessed. *Repeated (R)*: Interaction occurs repeatedly. *Nonrepeated (NR)*: Interaction occurs once or without defined intervals. *Hierarchical (H)*: One resource can control the other. *Nonhierarchical (NH)*: The involved resources are peers.	- Kees et al. (2015) - Suchman (2007)	- Fundamental Modelling Concepts (FMC Modeling, 2017) - Unified Modelling Language (OMG, 2017) - Technical Architecture Modelling (FMC Modeling, 2017)
	Parameterization		Parameterizations indicate that one resource steers another resource with reference to mutually agreed goals. Parameterizations refer to a distinct timespan and have a	- Encarnação and Kirste (2005)	- No source available

(Continues)

TABLE 1 (Continued)

	Name	Notational Element	Definition	Selected References (Concept)	Selected References (Notational Elements)
			management focus, granting the steered resource degrees of freedom in action and decision making. Parameterizations are annotated with goals.		
	Observation		Observations refer to the active acquisition of data from a primary source through sensing capabilities. Observations are annotated with properties.	- Borgia (2014) - Perera et al. (2014a) - Streitz et al. (2005)	- No source available
Service and service system	**Service**		Service involves at least two service systems and related resources in a collaborative and interactive process wherein they cocreate value for the benefit of one another. Services are equipped with a name.	- Peters et al. (2016) - Spohrer and Maglio (2008) - Vargo and Lusch (2016)	- Service Networks Modelling (Danylevych, Karastoyanova, & Leymann, 2010) - Service Network Notation (Bitsaki et al., 2009)
	Service system		Service systems are dynamic resource configurations that involve resources and are connected via interrelated resources. Service systems are equipped with a name. *Notational comment: The boundary of a service system needs not be modelled if the service system is atomic and does not involve further resources.*	- Ackoff (1971) - Maglio et al. (2009) - Nielsen et al. (2015) - Porter and Heppelmann (2014)	- Fundamental Modelling Concepts (FMC Modeling, 2017) - Technical Architecture Modelling (FMC Modeling, 2017) - Unified Modelling Language (OMG, 2017)

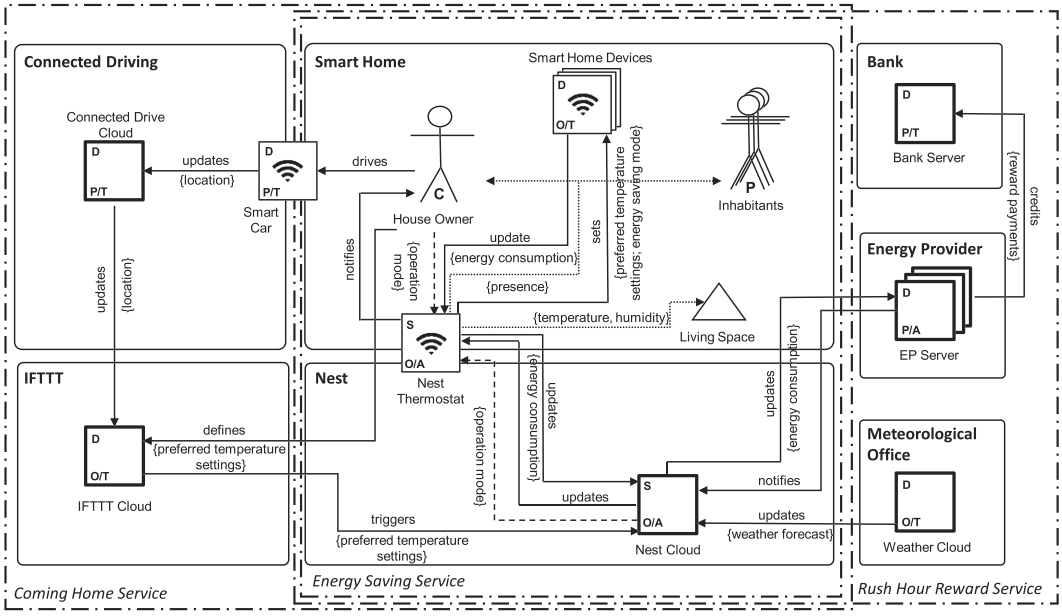


FIGURE 8 Integrated Service System Model of the Nest scenario

service systems (eg, Weather Cloud or IFTTT Cloud). These connections enable the Nest Thermostat to harness the data and functions of other resources.

Below, we present the Coming Home Service and the Rush Hour Reward Service, as all service systems are involved in at least one of these services. Furthermore, the functioning of both services relies on the Energy Saving Service, which manages the energy use of smart homes and buildings. The Energy Saving Service leverages the thermostat's ability to observe inhabitants and properties of rooms (eg, the temperature of the lounge). As the house owner parameterizes the Nest Thermostat (eg, energy saving mode), the thermostat updates the Nest Cloud with observed data (eg, energy use of smart home devices and the inhabitants' behaviour). By adding external data from other service systems (eg, weather forecast), the Energy Saving Service learns patterns related to energy use and human behaviour and can optimize the overall energy use of the smart home in line with current and predicted weather conditions.

Coming Home Service

Involved Service Systems (top level): Connected Driving, IFTTT, Smart Home, Nest

Purpose: Smart home gets set to preferred temperature settings when house owner is leaving a predefined area with his car.

Service Description:

(1) House Owner *defines* recipe with preferred temperature settings on the IFTTT Cloud

(2) House Owner *drives* Smart Car

(3) Smart Car uses its self-observation function to track its location and *updates* Connected Drive Cloud regarding location property

(4) Connected Drive Cloud *updates* IFTTT Cloud regarding location property

(5) IFTTT Cloud *triggers* Nest Cloud based on predefined recipe with preferred temperature settings when predefined location limitation is reached

(6) Nest Cloud *parameterizes* Nest Thermostat to new operation mode {preferred temperature settings,….}

(7) Nest Thermostat dynamically adapts its operations settings and *sets* Smart Home Devices to reach the preferred temperature

FIGURE 9 Service Description Model for the Coming Home Service

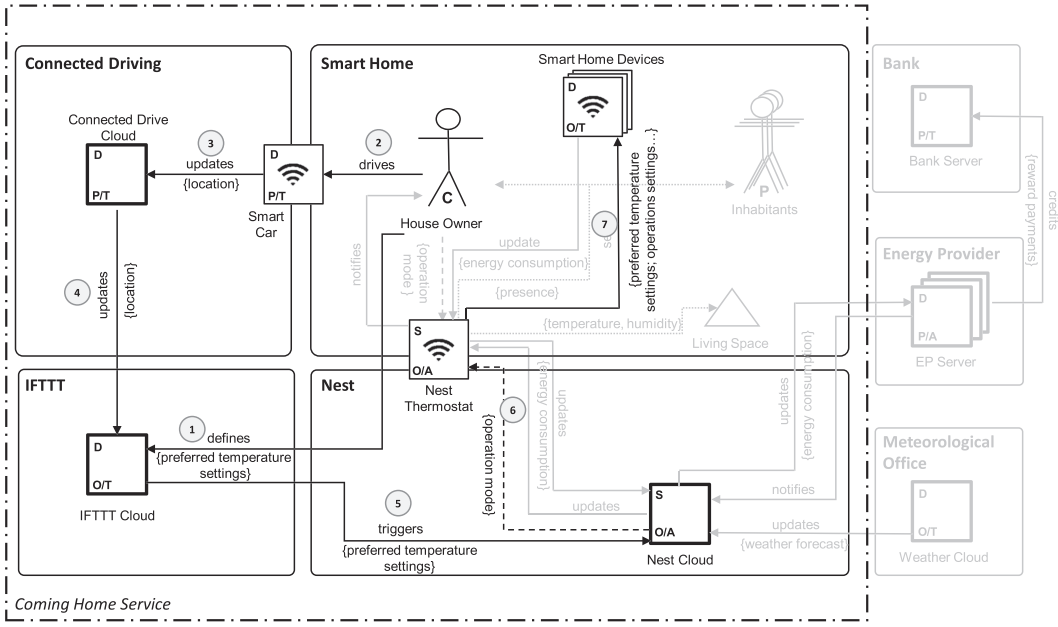


FIGURE 10 Integrated Service System Model with highlights for the Coming Home Service

The Coming Home Service builds on a recipe, ie, a simple chain of conditional statements from the free web-based platform IFTTT, which connects certain smart car models with a Nest service system (IFTTT, 2017). The smart car then serves as boundary object. The purpose of the Coming Home Service is to change the temperature at home when the house owner leaves a predetermined location. To this end, the house owner needs to connect the Nest Thermostat and the smart car through a recipe. The Service Description Model shown in Figure 9 provides details on the Coming Home Service. Therein, the name of the service is stated in the top left corner. To the right of this,

Rush Hour Reward Service

Involved Service Systems (top level): Smart Home, Nest, Bank, Energy Provider, Meteorological Office

Purpose: House Owner receives reward payments from the energy provider, as the energy usage for the smart home is adapted to demand peaks during rush hour events using internal data (e.g. heating history) and external data (e.g. weather forecast).

Service Description:

(1) Smart Home Devices *update* Nest Thermostat regarding energy consumption property
(2) Nest Thermostat *updates* Nest Cloud regarding energy consumption property
(3) Weather Cloud *updates* Nest Cloud about weather forecast
(4) Nest Cloud *updates* EP Server about energy consumption
(5) EP Server applies forecast function to predict next demand peak and *notifies* Nest Cloud about next rush hour event
(6) Nest Cloud *parameterizes* Nest Thermostat to energy saving mode before and during next rush hour event using its energy saving calculation function
(7) Nest Thermostat *notifies* House Owner about next rush hour event
(8) Nest Thermostat *sets* itself and the properties of the Smart Home Devices to the energy saving mode before and during rush hour event by using its self-regulation function
(9) Smart Home Devices *update* Nest Thermostat about energy consumption after rush hour event
(10) Nest Thermostat *updates* Nest Cloud regarding energy consumption property
(11) Nest Cloud uses energy usage calculation function and *updates* EP Server about difference in energy usage
(12) EP Server uses reward payments calculation function to determine the financial bonus and *credits* reward payments to the banking account of the House Owner on the Bank Server

FIGURE 11 Service Description Model for the Rush Hour Reward Service

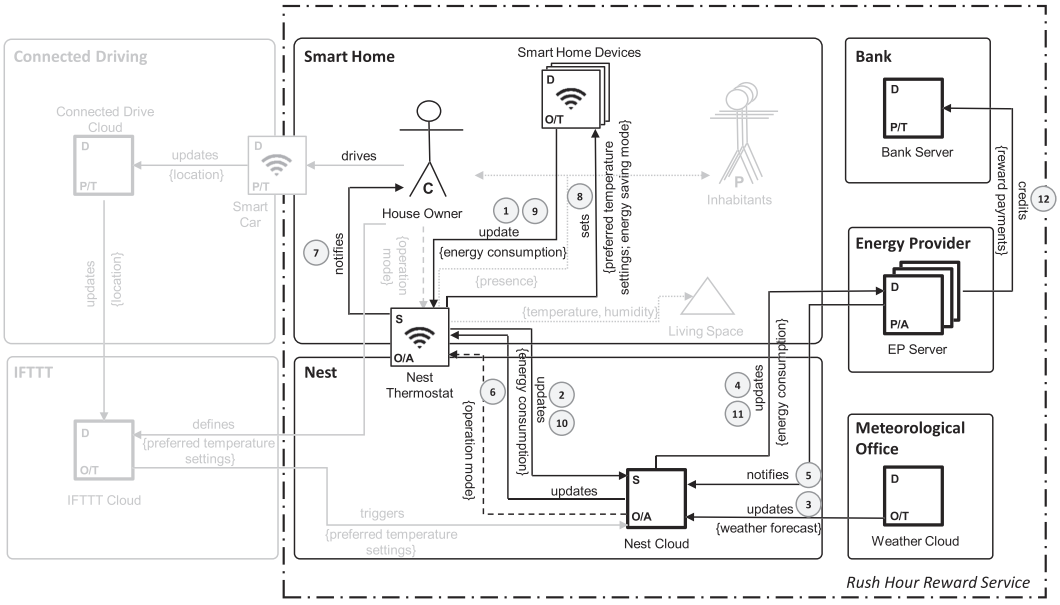


FIGURE 12 Integrated Service System Model with highlights for the Rush Hour Reward Service

information about the service is provided in three sections: first, all involved service systems on the top-most level of aggregation are listed; second, the purpose of the service is described to provide a shared understanding; third, the value cocreation process is detailed through a sequential description including relationships and attributes. The resources and relationships involved in the Coming Home Service can also be highlighted in the Integrated Service System Model (Figure 10). While the modelling of the relationship sequence complies with the metamodel (Figure 7), the graphical highlighting using shades of grey is not included in the metamodel but can be implemented in a modelling tool. The same holds for the underlining of functions. Comparing Figure 9 and Figure 10, it becomes evident that initial Service Description Models can be derived from Integrated Service System Models, but must be refined and extended afterwards.

Next, we examine the Rush Hour Reward Service (Figure 11). In the Unites States, energy providers collaborate with Nest to implement demand-side energy management, a technique balancing network loads in times of excess demand. Nest's value proposition here is the collection and sharing of data, which energy providers can use to better predict demand peaks as well as the self-dependent coordination of local devices in the case of "rush hours"—ie, situations where energy demand is anticipated to exceed energy supply. If the house owner approves and the energy provider registers via its monitoring systems that energy has been saved, the house owner receives a monetary reward. This example demonstrates how SSS can facilitate the balancing of energy network loads. With the use of renewable energy, demand-side management is more cost-efficient and eco-friendly than supply-side energy management (Strbac, 2008). To make our example easier to understand, we have again added a representation of the Integrated Service System Model highlighting involved resources, relationships, and service systems (Figure 12).

6 | EVALUATION

We evaluated our DSML via multiple methods. In addition to the modelling of real-world scenarios during the development phase, we discussed the DSML against Frank's (2013) GRs and domain-specific DOs. When discussing the DSML against DOs, we also compared it with selected conceptual modelling approaches as competing artefacts. Finally, we conducted interviews with industry experts to collect empirical data for further developing the DSML and for assessing to what extent it meets GRs and DOs.

6.1 | Interviews with industry experts

We conducted semistructured interviews with experts from eight organizations and diverse industries (Myers & Newman, 2007). Experts were selected based on their experience with SSS in specific IoT domains. By selecting experts in various roles, ranging from IT Architect to Head of Marketing and Operations, we covered many challenges faced by organizations when engaging in SSS. Table 2 shows all organizations that participated in the evaluation. Each interview took between 1.5 and 2 hours and was recorded or attended by two researchers. The interviews were structured around a presentation that laid out the research problem, presented the DOs, outlined the abstract and concrete syntax, and provided a simple and a complex real-world scenario. The simple scenario was centred around Nest Protect, a smart smoke and carbon monoxide detector, whereas the complex scenario referred to the Nest Learning Thermostat. To use interview time efficiently, we chose both examples from one domain. As all experts had an academic background in information systems or computer science, they easily understood the metamodel notation. When conducting the interviews, we used the verbal protocol technique (Ericsson & Simon, 1999), asking interviewees to "think aloud" when answering questions or applying the DSML. To challenge the metamodel, we discussed all concepts and relationships, asking the experts to provide examples from their organizations. In addition, we asked for missing concepts and potential inconsistencies. The same was done for the concrete syntax. By discussing the scenarios, we ensured an appropriate level of detail and rather focused on entire models than on individual concepts.

To summarize the results, all experts confirmed the relevance of the research problem and appreciated the development of a DSML for SSS. The experts also attributed huge economic potential to SSS and affirmed that an understanding of SSS can create competitive advantage. They also agreed with the DOs. Based on the experts' assessment, organizations lack a means for modelling and communication in SSS initiatives. For example, the expert from SERVICE II mentioned that it is common practice to use DSMLs when pitching project ideas to management. He underscored the usefulness of our DSML for such purposes. The expert from PRODUCTION II highlighted the importance of transparent design decisions on how smartness can be realized in SSS scenarios. With construction sites often lacking a stable internet connection, organizations cannot follow standard approaches that depend on cloud platforms to exchange data. A third expert, working for SERVICE I, appreciated the possibility of capturing nested service system structures. In the context of SERVICE I, legal requirements force organizations to store personal and service-usage data in separate systems. Only trusted third-party providers are allowed to combine data from both systems.

We incorporated most of the experts' recommendations regarding the abstract and concrete syntax. To ensure that the DSML remains parsimonious, we abstracted from scenario-specific peculiarities as far as reasonable. We also checked whether the literature supported the changes we planned to make. The most significant recommendations are listed below. Details can be found in Appendix E. Regarding the resource concept, the metamodel initially only distinguished between transactional and analytical data usage for digital hubs. In response to the advice of some experts, we also applied this distinction to smart things. As for relationships, the initial metamodel only allowed for smart things' observations of individuals and the physical environment. As experts provided examples wherein smart things observed other smart things—eg, a smart car observing other smart cars—we also incorporated this feature. Based on the experts' feedback, goal-based parameterization was renamed parameterization, as the initial nomenclature seemed to be confusing. One expert recommended including the service concept in the structural view to link both views more closely and to clarify the relation between service system and service. This recommendation had the strongest impact on our DSML as the service concepts had so far only been included in the behavioural view. We changed the metamodel such that services are highlighted in Integrated Service System Models by encircling involved service systems and relationships. We also dropped a second model type from the behavioural view that focused on service hierarchies. The revised structural and behavioural view complement each other more seamlessly.

As for the concrete syntax, we followed the experts' recommendation to distinguish between service customers and participants by using "C" and "P" as qualifiers. We also changed the notational element for service systems to a

TABLE 2 Organizations involved in the evaluation

Organization	Company Type	Employees	Revenue (EUR)	Job Title
(1) SERVICE I	Automotive service start-up	30 (2016)[a]	?[b]	Head of Marketing and Operations
(2) SERVICE II	IT consulting	410 (2016)[a]	100 M (2016)[a]	IT Architect
(3) PRODUCTION I	Manufacturer for optical systems and medical devices	25 000 (2016)[a]	4.88 B (2016)[a]	Lead Digital Business and Ecosystems Development
(4) SERVICE III	Globally operating bank	99 700 (2016)[a]	30 B (2016)[a]	IoT Expert
(5) PRODUCTION II	Construction	23 000 (2015)[a]	4 B (2015)[a]	Project Manager IoT Product Lifecycle Management
(6) SERVICE IV	FinTech bank	48 (2015)[a]	33 M (2015)[a]	Head of Marketing and Operations
(7) SERVICE V	Strategy consulting	11 000 (2015)[a]	8.4 B (2015)[a]	IoT Consultant
(8) SERVICE VI	Telecommunications	218 000 (2016)[a]	73.1 B (2016)[a]	Action Line Leader (Digital Industry)

Abbreviations: B, Billion; IT, information technology; M, Million.
[a]Year of latest available information.
[b]No information available.

rectangle with rounded corners as the experts opined that the symbol we initially used from UML use-case diagrams seemed to be too similar to the symbols used for digital hubs and smart things. Further, one expert recommended to graphically highlight service systems, resources, and relationships involved in a service. We incorporated this feedback in the demonstration example, but it needs to be fully tackled in future research, when the semiformal metamodel is further developed into a formal metamodel.

6.2 | Competing artefact analysis

When discussing the DSML against DOs, we also compared it with selected competing artefacts. Thereby, we followed the proposed way of del-Río-Ortega, Resinas, Cabanillas, and Ruiz-Cortés (2013). Because of the high number of service- and IoT-related modelling approaches, we made a selection to ensure comparability. With the IoT and SSS being emergent phenomena, we only considered lately developed or recently updated approaches. We also focused on established approaches in terms of high-quality publications, citations, or adoption in industry. Finally, we only considered conceptual modelling approaches with a broad domain coverage deliberately excluding technical modelling approaches or those with a narrow domain focus. Table 3 shows the results of the competing artefact analysis.

Based on the selection criteria just outlined, we chose Alter's (2012b) metamodel for service activities and service systems, as it has a broad domain focus, builds on work systems theory (Alter, 2008), and has been published in prime outlets. We also included the e3family, proposed by Gordijn, Akkermans, and van Vliet (2001), which has been continuously extended and covers many concepts for modelling service systems. Next, Oberländer et al. (2018) contributed to development of our DSML based on their findings related to business-to-thing interactions and their initial thoughts on the in-depth modelling of SSS. Finally, the FMC is well-known and used in industry for modelling software-intensive systems. Although this selection may not include all relevant approaches, we are confident to have covered diverse and the most comparable works.

Alter (2012b) proposed a metamodel for service systems that builds on work systems theory and the idea of viewing systems as service (Alter, 2008, 2010). He offers an integrated metamodel with concepts related to service systems and a view that textually illustrates which concepts are involved in service. To capture the nature of service and service systems, the metamodel includes concepts such as value constellation, process and activity, service system, and resource. The metamodel captures important concepts related to service systems but does not cover concepts related to SSS such as (self-dependent) smart things. Relationships such as observation and parameterization are not covered either. Thus, DO1 is partially fulfilled. The metamodel allows for nested service system structures, ie, service systems can interact with and be part of other service systems. However, it includes no means that would allow for coping with dynamic adaptation as required in SSS. Thus, DO2 is partially fulfilled. Finally, DO3 is not fulfilled, the reason being that the metamodel does not deal with smartness in the sense of extended data analysis and self-x capabilities.

The e3family is a prominent modelling approach, which has been extended over the years (Pijpers, Leenheer, Gordijn, & Akkermans, 2012). An important feature is the ability to model services (ie, value bundles) enabled by multiple parties through the interaction of IT and humans. The e3value ontology, the foundation of the e3family, focuses on exploring, analysing, and evaluating value networks (Akkermans & Gordijn, 2003). With e3service, modellers can map customer needs on services and service compositions (Razo-Zapata, Gordijn, Leenheer, & Wieringa, 2015). With the e3alignment framework, the e3family has been extended to enable the design and analysis of value webs. Thereby, e3alignment accounts for four perspectives: strategic, value, process, and IS interactions (Pijpers et al., 2012; Razo-Zapata et al., 2015). Although the e3family contains even more frameworks, we focused on the most important ones for our analysis. Regarding DO1, the e3family covers the roles and implications of IT. Yet there is a lack of concepts and relationships enabled by the IoT. Thus, DO1 is partially fulfilled. With the modelling of nested service networks - especially value webs, DO2 is partially fulfilled, too. Another reason for partial fulfilment is that there is no means for coping with dynamic adaptation. As for DO3, the e3familiy does not account for different facets of smartness. In particular, data analysis and self-x capabilities are not covered.

Oberländer et al. (2018) proposed a taxonomy of business-to-thing interaction patterns and used these patterns for modelling SSS from a structural perspective and on a high level of abstraction. They also cover the behavioural view by highlighting the sequence of interactions and involved concepts and relationships in the structural view. Their approach partially meets DO1 as it covers novel resources such as smart things but not resources that only exist in the digital world. Moreover, all smart things are treated equally and not distinguished regarding their capabilities. The same holds true for relationship types. As for DO2, Oberländer et al. (2018) show how the proposed interaction patterns can be combined in complex scenarios. They also use multiobjects to account for dynamic adaptation. However, they do not support nested service system structures, which is why DO2 is only partially met. Regarding DO3, Oberländer et al. (2018) model smart things independently from their data analysis and self-x capabilities.

TABLE 3 Results of the competing artefact analysis

	Capturing Diversity of Resources nd Relationships (DO1)	Capturing Nested Service System Structures (DO2)	Capturing Different Facets of Smartness (DO3)
Alter (2012b)	(✓)	(✓)	-
e3family	(✓)	(✓)	-
Oberländer et al. (2018)	(✓)	(✓)	(✓)
FMC	(✓)	(✓)	-
Our DSML for SSS	✓	✓	✓

Note: - = not fulfilled; (✓) = partially fulfilled; ✓ = fulfilled.
Abbreviations: FMC, Fundamental Modelling Concepts; DSML, domain-specific modelling language; SSS, smart service systems.

However, they pose that ever more smart things are self-dependent actors and autonomous interaction partners. Hence, DO3 is partially fulfilled, too.

The last approach analysed here is FMC, a framework for modelling software-intensive systems (FMC Modeling, 2017). Based on a precise terminology and notational elements, FMC aims at introducing developers, users, or customers into complex software systems. To that end, it distinguishes between a structural view (ie, block diagrams that represent models of the static compositional structure of service systems) and a behavioural view (ie, extended Petri nets aligned with the block diagrams). With a focus on software-intensive systems, the FMC covers relationships between individuals and information systems but lacks concepts relevant for SSS. Especially, resources such as (self-dependent) smart things and digital hubs as well as relationships such as observation and parameterization are missing. Therefore, DO1 is partially fulfilled. The structural view supports nested service system structures. However, as there is no means that would allow for coping with dynamic adaptation, DO2 is partially fulfilled. Finally, DO3, which is about capturing different facets of smartness, is not addressed, as FMC does not cater for data analysis or self-x capabilities.

Our DSML meets all DOs to the full extent. Regarding DO1, it covers the sociotechnical spectrum of resources, including smart things as boundary objects, and a broad range of relationships including observation and parameterization. What is more, the DSML features attributes for all resources and relationships, including the distinction between self-dependent and dependent smart things and digital hubs. As for DO2, the DSML distinguishes between atomic and composed service systems and it includes multiobjects to support the modelling of dynamic adaptation. Finally, the DSML deals with data analysis and self-x capabilities by means of a data usage attribute and functions as well as by distinguishing dependent and self-dependent smart things and digital hubs. Based on these properties, our DSML outperforms the selected conceptual modelling approaches (competing artefacts) with respect to all DOs. It particularly adds to DO1 by introducing novel concepts and relationships as well as to DO3 by focusing on the data analysis and self-x capabilities of smart things and digital hubs. Overall, the discussion of our DSML against Frank's (2013) GRs, which is shown in Appendix F, and the competing artefact analysis confirm that the DSML answers the research question and extends existing knowledge. In the next section, we discuss our contribution and limitations in detail and outline ideas for future research.

7 | DISCUSSION AND CONCLUSION

Our research was motivated by the recent uptake of SSS and the IoT. In line with the emergent nature of both phenomena, their interplay is poorly understood. Building the basis for explanatory and design-led research as well as for the analysis and design of SSS, a means for the conceptual modelling of SSS, which also accounts for novel IoT-enabled concepts is in high need. Hence, we developed, demonstrated, and evaluated a DSML with an abstract syntax specified in terms of a metamodel and a concrete syntax with notational elements that enable modelling SSS as diagrams. The DSML includes concepts such as service, service system (eg, atomic and composed), resource (eg, individual, smart thing, digital hub, and physical environment), and relationship (eg, interaction, observation, and parameterization). All concepts are backed by the literature and approved by industry experts. As specified in our domain-specific DOs, the DSML covers the sociotechnical spectrum of resources involved in SSS, nested structures of service systems, and different means for realizing smartness. As for the evaluation, we used the DSML to model simple and complex real-world scenarios from all functional IoT domains, we sought the opinion of industry experts, discussed the DSML against GRs for DSMLs, and we analysed whether the DSML outperforms selected competing artefacts regarding the DOs.

The DSML is the first to enable the conceptual modelling of SSS, not only for simple and complex scenarios but also while accounting for novel IoT-enabled concepts. It contributes to the descriptive knowledge as it reconstructs concepts and relationships from the SSS domain (Becker et al., 2013; Matook & Brown, 2017). It also synthesizes and extends knowledge from the IoT and service domains by, first, introducing a unified nomenclature

understandable for modellers and model users from academia and industry and, second, by proposing construction rules in terms of a metamodel. Our DSML makes SSS tangible and offers modellers and model users a means for communication. This is notable as the academic discourse on technology-enabled services is fragmented (Bardhan et al., 2010; Larson, 2016). Furthermore, the DSML adds to the prescriptive knowledge as we subjected its abstract and concrete syntax to testing in multiple real-world scenarios and to the scrutiny of experts from diverse organizations. Hence, we claim the DSML to have a referential character, meaning that it provides modellers and model users with guidance on the modelling of SSS.

The DSML adds to the knowledge on the IoT and service domains. From an IoT perspective, it enhances our understanding of the role of smart things in broader contexts. This is important as IoT research has so far mainly focused on technical and engineering challenges (Atzori et al., 2010; Kortuem et al., 2010), simple scenarios, and individual smart things (Oberländer et al., 2018; Püschel et al., 2016). The few works that allow for the modelling of IoT-enabled scenarios are technical and restricted to specific domains (eg, Christoulakis & Thramboulidis, 2016; De et al., 2011; Meyer et al., 2013; Xu et al., 2012). For the same reasons, the DSML increases service-related knowledge. First, it fosters our understanding of the role of smart products in value cocreation. Second, it extends the latest conceptual works on SSS by enabling an in-depth modelling of SSS (Beverungen, Müller, et al., 2017) and existing service modelling approaches by incorporating IoT-enabled concepts.

The DSML and its limitations stimulate future research. Therefore, we compile limitations of our DSML and propose avenues for future research:

- With the development of DSMLs being an iterative process, we cannot formally claim to have reconstructed all concepts and relationships from the SSS domain. However, based on the experience we gained when applying the DSML and in line with the industry experts' feedback, we are confident to have covered the most important concepts and relationships. Furthermore, all included concepts and relationships are backed by the literature. As we pragmatically reused the notational elements included in the concrete syntax, they should be refined and further evaluated in the future. As the IoT is a fast-moving field, we also recommend updating the DSML from time to time. When doing so, it should be challenged whether the list-like textual representation of Service Description Models is too restrictive for the conceptual modelling of SSS. We also recommend extending our evaluation activities by using the DSML for modelling additional scenarios and by subjecting it to the scrutiny of further experts from industry and academia.
- As the DSML aims to capture SSS from a conceptual perspective, we deliberately omitted technical features relevant for distinct target groups and purposes. For example, although the DSML covers data and functions as attributes, both concepts could have been described in more detail, eg, in terms of interfaces, data structures, or algorithms. While this is necessary when using the outcomes of a conceptual modelling project as input for a technical specification, it is beyond the scope of our DSML. For the same reason, we omitted notational elements for collapsed service systems or colours used for highlighting concepts and relationships. Such features should be considered when a formal metamodel is constructed.
- From a methodological perspective, the DSML is a modelling language, not a modelling method. A modelling method includes a step-by-step procedure, role definitions, and a modelling tool. Building on our DSML, a modelling method would offer guidance for the analysis and design of SSS. Hence, we recommend complementing the DSML trough a modelling method. At the same time, the abstract syntax, which is specified in terms of a semiformal metamodel, should be further developed into a formal metamodel that can be implemented by modelling tools.
- Besides the design-oriented research streams mentioned above, our DSML can also serve as foundation for explanatory research. For example, the concepts included in the metamodel can be used to derive hypotheses about which factors drive or inhibit the adoption of SSS in practice. Related insights may lead to modelling heuristics that enrich the modelling method to be designed.

ORCID

Maximilian Röglinger https://orcid.org/0000-0003-4743-4511

REFERENCES

Ackoff, R. L. (1971). Towards a system of systems concepts. *Management Science*, 17(11), 661–671. https://doi.org/10.1287/mnsc.17.11.661

Akkermans, J. M., & Gordijn, J. (2003). Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering*, 8(2), 114–134. https://doi.org/10.1007/s00766-003-0169-x

Allmendinger, G., & Lombreglia, R. (2005). Four strategies for the age of smart services. *Harvard business review*, 83(10), 131–145.

Alter, S. (2008). Defining information systems as work systems: Implications for the IS field. *European Journal of Information Systems*, 17(5), 448–469. https://doi.org/10.1057/ejis.2008.37

Alter, S. (2010). Viewing systems as services: A Fresh approach in the IS field. *Communications of the Association for Information Systems*, 26, 195–224.

Alter, S. (2012a). Challenges for service science. *JITTA: Journal of Information Technology Theory and Application*, 13(2), 22–37.

Alter, S. (2012b). Metamodel for service analysis and design based on an operational view of service and service systems. *Service Science*, 4(3), 218–235. https://doi.org/10.1287/serv.1120.0020

Ashby, W. R. (1991). Principles of the self-organizing system. In G. J. Klir (Ed.), *Facets of Systems Science* (pp. 521–536). Boston, MA: Springer US.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity*. Cambridge: MIT Press.

Bardhan, I. R., Demirkan, H., Kannan, P. K., Kauffman, R. J., & Sougstad, R. (2010). An interdisciplinary perspective on IT services management and service science. *Journal of Managent Information System*, 26(4), 13–64. https://doi.org/10.2753/MIS0742-1222260402

Barile, S., & Polese, F. (2010). Smart service systems and viable service systems: Applying systems theory to service science. *Service Science*, 2(1-2), 21–40. https://doi.org/10.1287/serv.2.1_2.21

Barrett, M., Davidson, E., Prabhu, J., & Vargo, S. L. (2015). Service innovation in the digital age: Key contributions and future directions. *MIS Quarterly*, 39(1), 135–154.

Batool, K., & Niazi, M. A. (2017). Modeling the internet of things: A hybrid modeling approach using complex networks and agent-based models. *Complex Adaptive Systems Modeling*, 5(1), 393–411. https://doi.org/10.1186/s40294-017-0043-1

Becker, J., Beverungen, D., & Knackstedt, R. (2010). The challenge of conceptual modeling for product–service systems: Status-quo and perspectives for reference models and modeling languages. *Information Systems and e-Business Management*, 8(1), 33–66. https://doi.org/10.1007/s10257-008-0108-y

Becker, J., Beverungen, D., Knackstedt, R., Matzner, M., Müller, O., & Pöppelbuß, J. (2013). Designing interaction routines in service networks. *Scandinavian Journal of Information Systems*, 25(1), 37–68.

Bennett, N., & Lemoine, J. (2014). What VUCA really means for you. *Harvard Business Review*, 92(1-2), 27.

Beverungen, D., Lüttenberg, H., & Wolf, V. (2018). Recombinant service systems engineering. *Business & Information Systems Engineering*, 60(5), 377–391. https://doi.org/10.1007/s12599-018-0526-4

Beverungen, D., Matzner, M., & Janiesch, C. (2017a). Information systems for smart services. *Information Systems and e-Business Management*, 15(4), 781–787. https://doi.org/10.1007/s10257-017-0365-8

Beverungen, D., Müller, O., Matzner, M., Mendling, J., & Vom Brocke, J. (2017b). Conceptualizing smart service systems. *Electronic Markets*, 83(10), 1–12. https://doi.org/10.1007/s12525-017-0270-5.

Bitsaki, M., Danylevych, O., van den Heuvel, W.-J., Koutras, G. D., Leymann, F., Mancioppi, M., … Papazoglo, M. P. (2009). Model transformations to leverage service networks. In G. Feuerlicht, & W. Lamersdorf (Eds.), *International Conference on Service-Oriented Computing Workshops (ICSOC 2009)* (pp. 103–117). Berlin, Heidelberg: Springer.

Böhmann, T., Leimeister, J. M., & Möslein, K. (2014). Service systems engineering. *Business Information System Engineering*, 6 (2), 73–79. https://doi.org/10.1007/s12599-014-0314-8

Böhmann, T., Leimeister, J. M., & Möslein, K. (2018). The new fontiers of service systems engineering. *Business Information System Engineering*, 60(5), 373–375. https://doi.org/10.1007/s12599-018-0553-1

Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues. *Computer Communications*, 54, 1–31. https://doi.org/10.1016/j.comcom.2014.09.008

Bouhdadi, M., Balouki, Y., & Chabbar, E. M. (2007). Meta-modelling syntax and semantics of structural concepts for open networked enterprises. In O. Gervasi, & M. L. Gavrilova (Eds.), *Computational Science and Its Applications – ICCSA 2007* (pp. 45–54). Berlin, Heidelberg: Springer.

Boulding, K. E. (1956). General systems theory—The skeleton of science. *Management Science*, 2(3), 197–208. https://doi.org/10.1287/mnsc.2.3.197

Briggs, R. O., & Schwabe, G. (2011). On expanding the scope of design science in IS research. In H. Jain, A. P. Sinha, & P. Vitharana (Eds.), *Service-Oriented Perspectives in Design Science Research: DESRIST 2011. Lecture Notes in Computer Science* (pp. 92–106). Berlin, Heidelberg: Springer.

Caputo, A., Marzi, G., & Pellegrini, M. (2016). The internet of things in manufacturing innovation processes: Development and application of a conceptual framework. *Business Process Management Journal*, 22(2), 383–402. https://doi.org/10.1108/BPMJ-05-2015-0072.

Cardoso, J. (2013). Modeling service relationships for service networks. In W. van der Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, J. Falcão e Cunha, et al. (Eds.), *Exploring Services Science* (pp. 114–128). Berlin, Heidelberg: Springer.

Cardoso, J., Pedrinaci, C., & de Leenheer, P. (2013). Open semantic service networks: modeling and analysis. In W. van der Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, J. Falcão e Cunha, et al. (Eds.), *Exploring Services Science* (pp. 141–154). Berlin, Heidelberg: Springer.

Chandler, J. D., & Lusch, R. F. (2015). Service systems: A broadened framework and research agenda on value propositions, engagement, and service experience. *Journal of Service Research*, 18(1), 6–22. https://doi.org/10.1177/1094670514537709

Christoulakis, F. & Thramboulidis, K. (2016) IoT-based integration of IEC 61131 industrial automation systems: The case of UML4IoT. In: 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), pp. 322–327.

Cilliers, P. (2001). Boundaries, hierarchies and networks in complex systems. *International Journal of Innovation Management*, 5(2), 135–147. https://doi.org/10.1142/S1363919601000312

Danylevych, O., Karastoyanova, D., & Leymann, F. (2010). Service networks modelling: An SOA & BPM Standpoint. *Journal of Universal Computer Science*, 16(13), 1668–1693. https://doi.org/10.3217/jucs-016-13-1668.

De, S., Barnaghi, P., Bauer, M., & Meissner, S. (2011). Service modelling for the Internet of Things. In *Proceedings of the Federated Conference on Computer Science and Information Systmes* (pp. 949–955). Szczecin, Poland: IEEEE.

del-Río-Ortega, A., Resinas, M., Cabanillas, C., & Ruiz-Cortés, A. (2013). On the definition and design-time analysis of process performance indicators. *Information Systems*, 38(4), 470–490. https://doi.org/10.1016/j.is.2012.11.004

Drucker, P. F. (2009). *The practice of management*. New York: HarperCollins.

Edvardsson, B., Tronvoll, B., & Gruber, T. (2011). Expanding understanding of service exchange and value co-creation: A social construction approach. *Journal of the Academy of Marketing Science*, 39(2), 327–339. https://doi.org/10.1007/s11747-010-0200-y

Encarnação, J. L., & Kirste, T. (2005). Ambient intelligence: Towards smart appliance ensembles. In M. Hemmje, C. Niederée, & T. Risse (Eds.), *From Integrated Publication and Information Systems to Information and Knowledge Environments: Lecture Notes in Computer Science 3379* (pp. 261–270). Berlin, Heidelberg: Springer.

Ericsson, K. A., & Simon, H. A. (1999). *Protocol analysis: Verbal reports as data*. Cambridge: MIT press.

Eriksson, O., Henderson-Sellers, B., & Ågerfalk, P. J. (2013). Ontological and linguistic metamodelling revisited: A language use approach. *Information and Software Technology*, 55(12), 2099–2124. https://doi.org/10.1016/j.infsof.2013.07.008

Fleisch, E. & Thiesse, F. (2007) On the management implications of ubiquitous computing: An IS perspective. In: Proceedings of the 15th European Conference on Information Systems: (ECIS 2007), St. Gallen, Switzerland. Österle, H., Schelp, J., Winter, R. (eds.), pp. 1929–1940.

FMC Modeling (2017) Home of fundamental modeling concepts [WWW document]. URL http://www.fmc-modeling.org/.

Frank, U. (2013). Domain-specific modeling languages: Requirements analysis and design guidelines. In I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, & J. Bettin (Eds.), *Domain Engineering* (pp. 133–157). Berlin, Heidelberg: Springer.

Gartner (2013) Gartner says the Internet of Things installed base will grow to 26 billion units by 2020 [WWW document]. URL http://www.gartner.com/newsroom/id/2636073

Geerts, G. L., & O'Leary, D. E. (2014). A supply chain of things: The EAGLET ontology for highly visible supply chains. *Decision Support Systems*, 63, 3–22. https://doi.org/10.1016/j.dss.2013.09.007

Gordijn, J., Akkermans, H., & van Vliet, J. (2001). Designing and evaluating e-business models. *IEEE intelligent Systems*, 16(4), 11–17.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2), 337–355.

Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: Foundations and developments. *Electronic Markets*, 25(3), 179–188. https://doi.org/10.1007/s12525-015-0196-8

Hamilton, J. (2004). Service value networks: Value, performance and strategy for the services industry. *Journal of Systems Science and Systems Engineering*, 13(4), 469–489. https://doi.org/10.1007/s11518-006-0177-8

Heise, D., Strecker, S., & Frank, U. (2014). ControlML: A domain-specific modeling language in support of assessing internal controls and the internal control system. *International Journal of Accounting Information Systems*, 15(3), 224–245. https://doi.org/10.1016/j.accinf.2013.09.001

Henfridsson, O., & Lindgren, R. (2010). User involvement in developing mobile and temporarily interconnected systems. *Information Systems Journal*, 20(2), 119–135. https://doi.org/10.1111/j.1365-2575.2009.00337.x

Heß, M., Kaczmarek, M., Frank, F., Podleska, L. & Taeger, G. (2015) Towards a DSML for clinical pathways in the realm of multi-perspective hospital modelling. In: Proceedings of the 23rd European Conference on Information Systems: (ECIS 2015), Münster, Germany. Becker, J., Vom Brocke, J., de Marco, M. (eds.), pp. 1–17.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.

IFTTT (2017) If this then that—BMW applet [WWW document]. URL https://ifttt.com/applets/374507p-if-you-leave-work-then-set-your-nest-thermostat-to-78-f-so-it-s-warm-when-you-arrive-home

Kees, A., Oberländer, A. M., Röglinger, M. & Rosemann, M. (2015) Understanding the Internet of Things: A conceptualisation of business-to-thing (B2T) interactions. In: Proceedings of the 23rd European Conference on Information Systems: (ECIS 2015), Münster, Germany. Becker, J., Vom Brocke, J., de Marco, M. (eds.), pp. 1–15.

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1), 41–50. https://doi.org/10.1109/MC.2003.1160055

Kolokotsa, D. (2016). The role of smart grids in the building sector. *Energy and Buildings*, 116, 703–708. https://doi.org/10.1016/j.enbuild.2015.12.033

Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1), 44–51. https://doi.org/10.1109/MIC.2009.143

Kurpjuweit, S., & Winter, R. (2007). Viewpoint-based meta model engineering. In M. Reichert, S. Strecker, & K. Turowski (Eds.), *EMISA 2007: Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures* (pp. 143–161). Bonn: Gesellschaft für Informatik.

Larson, R. C. (2016). Commentary—Smart service systems: Bridging the silos. *Service Science*, 8(4), 359–367. https://doi.org/10.1287/serv.2016.0140

Lea, R. & Blackstock, M. (2014) City hub: A cloud-based IoT platform for smart cities. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, pp. 799–804.

Leonardi, P. M. (2013). Theoretical foundations for the study of sociomateriality. *Information and Organization*, 23(2), 59–76. https://doi.org/10.1016/j.infoandorg.2013.02.002

Lim, C., & Maglio, P. P. (2018). Data-driven understanding of smart service systems through text mining. *Service Science*, 10 (2), 154–180. https://doi.org/10.1287/serv.2018.0208

Maglio, P. P., & Spohrer, J. (2008). Fundamentals of service science. *Journal of the Academy of Marketing Science*, 36(1), 18–20. https://doi.org/10.1007/s11747-007-0058-9

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

Mannadiar, R., & Vangheluwe, H. (2010). Domain-specific engineering of domain-specific languages. In *Proceedings of the 10th Workshop on Domain-Specific Modeling* (pp. 1–6). New York: ACM.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Matook, S., & Brown, S. A. (2017). Characteristics of IT artifacts: A systems thinking-based framework for delineating and theorizing IT artifacts. *Information Systems Journal*, 27(3), 309–346. https://doi.org/10.1111/isj.12108

Mattern, F., & Floerkemeier, C. (2010). From the internet of computers to the internet of things. In K. Sachs, I. Petrov, & P. Guerrero (Eds.), *From Active Data Management to Event-Based Systems and More* (pp. 242–259). Berlin, Heidelberg: Springer.

Medina-Borja, A. (2015). Editorial column—Smart things as service providers: A call for convergence of disciplines to build a research agenda for the service systems of the future. *Service Science*, 7(1), ii–v. https://doi.org/10.1287/serv.2014.0090

Meyer, S., Ruppen, A., & Magerkurth, C. (2013). Internet of things-aware process modeling: Integrating IoT devices as business process resources. In C. Salinesi, M. C. Norrie, & Ó. Pastor (Eds.), *Proceedings of Advanced Information Systems Engineering—25th International Conference, (CAiSE 2013), Valencia, Spain* (pp. 84–98). Berlin, Heidelberg: Springer.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, 17(1), 2–26. https://doi.org/10.1016/j.infoandorg.2006.11.001

National Science Foundation (2014) Partnerships for innovation: Building innovation capacity [WWW document]. URL https://www.nsf.gov/pubs/2014/nsf14610/nsf14610.pdf

NEST (2017) Nest homepage [WWW document]. URL https://nest.com/

Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J., & Peleska, J. (2015). Systems of systems engineering: Basic concepts, model-based techniques, and research directions. *ACM Computing Surveys*, 48(2), 1–41. https://doi.org/10.1145/2794381

Nordstrom, G., Sztipanovits, J., Karsai, G. & Ledeczi, A. (1999) Metamodeling—Rapid design and evolution of domain-specific modeling environments. In: Proccedings of the IEEE Conference and Workshop on Engineering of Computer-Based Systems (ECBS 99), pp. 68–74.

Oberländer, A. M., Röglinger, M., Rosemann, M., & Kees, A. (2018). Conceptualizing business-to-thing interactions—A sociomaterial perspective on the Internet of Things. *European Journal of Information Systems*, 27(4), 486–502. https://doi.org/10.1080/0960085X.2017.1387714

OMG (2017) Object modeling group—Documents associated with Unified Modeling Language (UML)—Version 2.5 [WWW document]. http://www.omg.org/spec/UML/2.5/PDF

Ouyang, Q. C., Spohrer, J., Caraballo, J., Davis, D., Perelgut, S., Mindel, M., … Subbanna, S. (2017). Collaborative innovation centers (CICs): Toward smart service system design. In Y. Sawatani, J. Spohrer, S. Kwan, & T. Takenaka (Eds.), *Serviceology for Smart Service System* (pp. 385–391). Tokyo: Springer.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014a). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454.

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014b). Sensing as a service model for smart cities supported by internet of things. *Transactions on Emerging Telecommunications Technologies*, 25(1), 81–93.

Peters, C., Maglio, P., Badinelli, R., Harmon, R. R., Maull, R., Spohrer, J. C., Tuunanen, T., Vargo, S. L., Welser, J. J., Demirkan, H., Griffith, T. L. & and Moghaddam, Y. (2016) Emerging digital frontiers for service innovation. *Communications of the Association for Information Systems*, 39 (1), 136–149. https://doi.org/10.17705/1CAIS.03908.

Pijpers, V., Leenheer, P. de, Gordijn, J. & Akkermans, H. (2012) Using conceptual models to explore business-ICT alignment in networked value constellations. *Requirements Engineering*, 17 (3), 203–226. https://doi.org/10.1007/s00766-011-0136-x.

Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, 92(11), 64–88.

Porter, M. E., & Heppelmann, J. E. (2015). How smart, connected products are transforming companies. *Harvard Business Review*, 93(10), 96–114.

Powell, W. W. (1990). Neither market, nor hierarchy: Network forms of organization. *Research in Organizational Behavior*, 12, 295–336.

Püschel, L., Röglinger, M. & Schlott, H. (2016) What's in a smart thing? Development of a multi-layer taxonomy. In: Proceedings of the 37th International Conference on Information Systems: (ICIS 2016), Dublin, Ireland. Agerfalk, P., Levina, N., Kien, S. (eds.), pp. 1–19.

Rai, A., & Sambamurthy, V. (2006). Editorial notes—The growth of interest in services management: Opportunities for information systems scholars. *Information Systems Research*, 17(4), 327–331. https://doi.org/10.1287/isre.1060.0108

Razo-Zapata, I. S., de Leenheer, P., Gordijn, J., & Akkermans, H. (2012). Service network approaches. In A. Barros, & D. Oberle (Eds.), *Handbook of Service Description: USDL and Its Methods* (pp. 45–74). New York: Springer.

Razo-Zapata, I. S., Gordijn, J., Leenheer, P. de & Wieringa, R. (2015) e^3 service: A critical reflection and future research. *Business Information Systems Engineering*, 57 (1), 51–59. https://doi.org/10.1007/s12599-014-0360-2.

Rijsdijk, S. A., & Hultink, E. J. (2009). How today's consumers perceive tomorrow's smart products. *Journal of Product Innovation Management*, 26(1), 24–42. https://doi.org/10.1111/j.1540-5885.2009.00332.x.

Rolls-Royce (2018a) Engine health management [WWW document]. URL https://www.rolls-royce.com/products-and-services/civil-aerospace/airlines/trent-7000.aspx#/

Rolls-Royce (2018b) Homepage [WWW document]. URL https://www.rolls-royce.com/products-and-services/civil-aerospace/aftermarket-services.aspx#/

Shaw, D. R., & Holland, C. P. (2010). Strategy, networks and systems in the global translation services market. *The Journal of Strategic Information Systems*, 19(4), 242–256. https://doi.org/10.1016/j.jsis.2010.08.001

Siau, K. & Rossi, M. (1998) Evaluation of information modeling methods—A review. In: *Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS 1998)*, pp. 314–322, Los Alamitos.

Slama, D., Puhlmann, F., Morrish, J., & Bhatnagar, R. M. (2015). *Enterprise IoT: Strategies and best practices for connected products and services*. Boston: O'Reilly.

Spohrer, J. C., & Maglio, P. P. (2008). The emergence of service science: Toward systematic service innovations to accelerate co-creation of value. *Production and Operations Management*, 17(3), 238–246. https://doi.org/10.3401/poms.1080.0027

Star, S. L. (2010). This is not a boundary object: Reflections on the origin of a concept. *Science, Technology, & Human Values*, 35(5), 601–617. https://doi.org/10.1177/0162243910377624.

Strbac, G. (2008). Demand side management: Benefits and challenges. *Energy policy*, 36(12), 4419–4426. https://doi.org/10.1016/j.enpol.2008.09.030

Strecker, S., Frank, U., Heise, D., & Kattenstroth, H. (2012). MetricM: A modeling method in support of the reflective design and use of performance measurement systems. *Information Systems and e-Business Management*, 10(2), 241–276. https://doi.org/10.1007/s10257-011-0172-6

Streitz, N. A., Rocker, C., Prante, T., van Alphen, D., Stenzel, R., & Magerkurth, C. (2005). Designing smart artifacts for smart environments. *Computer*, 38(3), 41–49. https://doi.org/10.1109/MC.2005.92.

Suchman, L. (2009). *Human-machine reconfigurations: Plans and situated actions*. New York: Cambridge University Press.

Uckelmann, D., Harrison, M., & Michahelles, F. (2011). *Architecting the internet of things*. Berlin, Heidelberg: Springer.

Vargo, S. L., & Lusch, R. F. (2004). Evolving to a new dominant logic for marketing. *Journal of Marketing*, 68(1), 1–17. https://doi.org/10.1509/jmkg.68.1.1.24036

Vargo, S. L., & Lusch, R. F. (2008a). Service-dominant logic: Continuing the evolution. *Journal of the Academy of Marketing Science*, 36(1), 1–10. https://doi.org/10.1007/s11747-007-0069-6

Vargo, S. L., & Lusch, R. F. (2008b). Why "service"? *Journal of the Academy of Marketing Science*, 36(1), 25–38. https://doi.org/10.1007/s11747-007-0068-7

Vargo, S. L., & Lusch, R. F. (2016). Institutions and axioms: An extension and update of service-dominant logic. *Journal of the Academy of Marketing Science*, 44(1), 5–23. https://doi.org/10.1007/s11747-015-0456-3

Vargo, S. L., Maglio, P. P., & Akaka, M. A. (2008). On value and value co-creation: A service systems and service logic perspective. *European Management Journal*, 26(3), 145–152. https://doi.org/10.1016/j.emj.2008.04.003

Velamuri, V. K. (2013). *Hybrid value creation*. New York: Springer Science & Business Media.

von Bertalanffy, L. (1976). *General system theory: Foundations, development, applications*. New York: George Braziller Inc.

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling—A research agenda. *Information Systems Research*, 13(4), 363–376. https://doi.org/10.1287/isre.13.4.363.69

Want, R., Schilit, B. N., & Jenson, S. (2015). Enabling the internet of things. *Computer*, 48(1), 28–35. https://doi.org/10.1109/MC.2015.12

World Bank (2017) Services value added (% of GDP) [WWW document]. URL http://data.worldbank.org/indicator/NV.SRV.TETC.ZS?locations=EU

Wuenderlich, N. V., Heinonen, K., Ostrom, A. L., Patricio, L., Sousa, R., Voss, C., & Lemmink, J. G. A. M. (2015). "Futurizing" smart service: Implications for service researchers and managers. *Journal of Services Marketing*, 29(6/7), 442–447. https://doi.org/10.1108/JSM-01-2015-0040.

Xu, H., Xu, Y., Li, Q., Lv, C. & Liu, Y. (2012) Business process modeling and design of smart home service system. In: *International Joint Conference on Service Science (IJCSS 2012)*, pp. 12–17.

Yoo, Y. (2010). Computing in everyday life: A call for research on experiential computing. *MIS Quarterly*, 34(2), 213–231.

Yoo, Y., Boland, R. J., Lyytinen, K., & Majchrzak, A. (2012). Organizing for innovation in the digitized world. *Organization Science*, 23(5), 1398–1408. https://doi.org/10.1287/orsc.1120.0771

Yoo, Y., Henfridsson, O., & Lyytinen, K. (2010). Research commentary—The new organizing logic of digital innovation: An agenda for information systems research. *Information Systems Research*, 21(4), 724–735. https://doi.org/10.1287/isre.1100.0322

AUTHOR BIOGRAPHIES

Rocco Xaver Richard Huber studied Information-Oriented Business Administration (BSc and MSc) with a major in Finance & Information Management at the University of Augsburg and further acquired an MBA degree from the University of Dayton. During his studies, he acquired practical experiences as an intern at Allianz in Munich and Guangzhou as well as at the consulting firm Zeb. Since October 2017, he works as a research assistant at the Research Center Finance & Information Management and the Business & Information Systems Engineering group of Fraunhofer FIT. He focuses his research on topics such as value-based process management, digital transformation, and smart service systems.

Louis Christian Püschel holds a PhD in Business and Information Systems Engineering from the University of Bayreuth and a Diploma in Industrial Engineering from the Brandenburg University of Technology Cottbus-Senftenberg (Germany). Since May 2015, Louis is working as research assistant at the Research Center Finance & Information Management (FIM) with a focus on business process management and the Internet of Things.

Maximilian Röglinger is a professor of Information Systems at the University of Bayreuth (Germany). He serves as Deputy Academic Director of the Research Center Finance & Information Management (FIM) and works with the Project Group Business & Information Systems Engineering of the Fraunhofer FIT. Most of Maximilian's work centers around business process management, customer relationship management, and digital transformation, including the Internet of Things. He publishes in journals like *Business & Information Systems Engineering*, *Decision Support Systems*, *European Journal of Information Systems*, *Journal of the Association for Information Systems*, *and Journal of Strategic Information Systems*. Maximilian is highly engaged in publicly and privately funded research projects. He has collaborated with companies such as Allianz, A.T. Kearney, Deutsche Bahn, Deutsche Bank, Fujitsu, Hilti, Infineon Technologies, Nord LB, Radeberger, REHAU, and Siemens. Maximilian earned his PhD at the University of Augsburg and holds a diploma in Business and Information Systems Engineering from the University of Bamberg.

How to cite this article: Huber RXR, Püschel LC, Röglinger M. Capturing smart service systems: Development of a domain-specific modelling language. *Info Systems J*. 2019;29:1207–1255. https://doi.org/10.1111/isj.12269

APPENDIX A

DESIGN RATIONALES REGARDING THE DOMAIN-SPECIFIC MODELLING LANGUAGE

Below, we provide design rationales for why and how we included concepts related to smart service systems (SSS) in our DSML. To that end, we focus on three main concepts: *service*, *service system*, and *smart service system*. The reasons for choosing these concepts are that they are most important for our DSML and that their definitions include almost all concepts relevant for the domain-specific modelling language (DSML). We proceed as follows: we provide a table for each main concept. At the top of each table, we repeat verbatim definitions of the respective main concept from the literature and highlight all subconcepts relevant for our DSML. After that, we provide design rationales for the main and subconcepts. We only repeat as much of the subconcepts' definitions as necessary as they have already been provided in the theoretical background. In case subconcepts relate to several main concepts, we include references between the tables. Based on the design rationales, it becomes clear that all main and sub-concepts are covered in our metamodel, either directly or indirectly. Like the DSML presented in the manuscript, the design rationales not only reflect the results of our design process, which draws on the literature and the modeling of real-world scenarios (Appendix Table 2), but also the expert feedback (Appendix Table A5) we received during the evaluation. Hence, they do not reflect the iterative design process of our DSML, but its final outcome.

APPENDIX B

REAL-WORLD SCENARIOS USED FOR THE DEVELOPMENT OF THE DSML

Table A4 lists the real-world scenarios that influenced the development of the abstract and concrete syntax of our DSML. It also indicates how these scenarios influenced our DSML. As we modelled the real-world scenarios shown in Table A4 in the specified order, earlier scenarios entailed more changes to the DSML than latter ones, which is why we argue that conceptual saturation has been reached.

TABLE A1 Design rationales related to the "Service" concept

"Service is the application of specialized competences (operant resources—knowledge and skills), through deeds, processes, and performances for the benefit of another entity or the entity itself." (Vargo and Lusch 2008a, p. 26)
"[T]he 'applied' designation makes operant resources—resources that can act on or in concert with other resources to provide benefit (create value), as distinguished from operand resources—resources which require action to create benefit—primary." (Vargo and Lusch 2008a, p. 31)
"From S-D logic, service is the application of competence for the benefit of another. So service involves at least two entities, one applying competence and another integrating the applied competences with other resources (value-cocreation) and determining benefit." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
Service	*Service*	Service is at the centre of SSS. Hence, the metamodel includes a service concept equipped with a purpose attribute. Please see "mutual benefit" for details on the meaning and use of purpose. In the metamodel, service is connected with the service system and relationship concepts to reflect that service is enacted through service systems in terms of interactive and collaborative processes. To account for the circumstance that service systems are connected through value propositions, the aggregation metarelationship related to service and service system is equipped with a "value proposition" association class, which simultaneously serves as attribute of service systems. Please see "mutual benefit" for details on value propositions. In the metamodel, we decided not to differentiate between service and smart service, but to generally refer to service as a specific set of interacting service systems and related resources. This complies with the conceptualization of service as the general process of resource application and integration for mutual benefit. In line with our focus on SSS, we split the service system concept into smart and ordinary service systems as well as into atomic and composed services systems. Our rationale was that a distinction for service and service system would have entailed unnecessary redundancies. More information can be found in Table A2 (service system).
Resources	*Resource*	With service systems being resource configurations, modelling resources is essential for SSS. Thus, resources are a distinct concept in the metamodel. The fact that service systems are resource configurations is reflected in the aggregation metarelationship between resources and service systems. Further, resources are connected through relationships, involving resources from one or many service systems. To cover the range of resources that might be involved in SSS, we split them into individuals, smart things, and digital hubs. We also decided to include the physical environment in the metamodel because individuals and smart things can observe the physical environment. This is a constitutive characteristic of SSS as opposed to digital service systems. Our rationale for treating the physical environment as a resource was that this allows for using the same relationships as for other resources. Otherwise, the metamodel would have become unnecessarily complex. In some cases, resources are service systems themselves. For our purposes, operant resources are resources that can act in a self-dependent manner without external intervention (sometimes even without external triggers) *and* that can learn from experience. In our DSML, this holds true for individuals and for self-dependent smart things and digital hubs. As highlighted in the literature, the smallest service system is an individual as a single operant resource. We treat all other resources as operand. Accordingly, we define operant resources not on the level of skills or functions, but on the level of self-dependent

(Continues)

TABLE A1 (Continued)

"Service is the application of specialized competences (operant resources—knowledge and skills), through deeds, processes, and performances for the benefit of another entity or the entity itself." (Vargo and Lusch 2008a, p. 26)
"[T]he 'applied' designation makes operant resources—resources that can act on or in concert with other resources to provide benefit (create value), as distinguished from operand resources—resources which require action to create benefit—primary." (Vargo and Lusch 2008a, p. 31)
"From S-D logic, service is the application of competence for the benefit of another. So service involves at least two entities, one applying competence and another integrating the applied competences with other resources (value-cocreation) and determining benefit." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
		resources. The rationale was that individuals, smart things, and digital hubs are the key resources of SSS and that skills and functions can only be meaningfully interpreted if associated with these resources. In line with the literature, functions of smart things and digital hubs can use data transactionally or analytically. Analytical functions can but need not serve purposes of self-dependency. More information about self-x functions can be found in Table A3. In the metamodel, we do not distinguish between operand and operant resources. Although we acknowledge the value of this distinction for theoretical purposes, it does not guide modellers when modelling SSS. Given the number of concepts and attributes modellers must consider, the distinction between operand and operant resources would have imposed further cognitive load. What is of operational importance, however, is the distinction between analytical and transactional data usage as well as between self-dependent and dependent smart things and digital hubs. As just discussed, this distinction directly relates to that between operant and operand resources. Hence, operant and operand resources are covered indirectly in the metamodel. As the distinction between self-dependent and dependent smart things and digital hubs is essential for SSS, we decided to account for it through distinct subconcepts (specialization metarelationship) for self-dependent smart things and digital hubs. Our rationale was that self-dependent smart things and digital hubs not only have additional attributes but differ from their dependent counterparts in their very nature. They can act and decide in line with goals without external intervention (sometimes even without external triggers) *and* learn by refining internal models of themselves. Both characteristics are enabled by extended data analysis and self-x functions, eg, self-monitoring, self-diagnosis, self-configuration, or self-optimization. Beyond the distinction between being self-dependent and dependent, it is important whether smart things and digital hubs process data analytically or transactionally and how compatible they are with products/services of other manufacturers/providers. To account for these characteristics, both concepts are included as attributes. We refrained from using further specialization metarelationships as this would have made the metamodel very complex and distracted from the key differentiation between self-dependent and dependent smart things and digital hubs. More information about data usage can be found in Table A2 (information).
Application of competences	*Relationship Attributes* (function and skill)	The application of competences is essential for SSS as service becomes manifest through interactive and collaborative processes of resource application and integration to cocreate value and mutual benefit for involved service systems. In our metamodel, the application of competences is covered in multiple ways. As mentioned, we define all resources as operant that can act and decide in a self-dependent manner

(Continues)

TABLE A1 (Continued)

"Service is the application of specialized competences (operant resources—knowledge and skills), through deeds, processes, and performances for the benefit of another entity or the entity itself." (Vargo and Lusch 2008a, p. 26)
"[T]he 'applied' designation makes operant resources—resources that can act on or in concert with other resources to provide benefit (create value), as distinguished from operand resources—resources which require action to create benefit—primary." (Vargo and Lusch 2008a, p. 31)
"From S-D logic, service is the application of competence for the benefit of another. So service involves at least two entities, one applying competence and another integrating the applied competences with other resources (value-cocreation) and determining benefit." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
		without external triggers and learn from experience. This includes individuals and self-dependent smart things and digital hubs. These concepts are equipped with skills and functions that can be applied. Furthermore, the application of competences is covered through relationships, which are split into interaction, parameterization, and observation in line with the literature and which can connect resources from the same and different service systems. An interaction, for example, can express that a digital hub triggers a function of a smart thing. The observation of properties of individuals or the physical environment also reflects the application of skills or functions. While the structural view of the metamodel primarily supports the detailed analysis of how resources are connected through relationships and grouped into service systems, the behavioural investigates the application of competences by shedding light on skills and functions as well as on the sequence in which relationships occur. We chose to distinguish between a structural and a behavioural view because this is common practice in conceptual modelling for complexity reduction and as it is in line with General Systems Theory. As for Service Description Models, which belong to the behavioural view, we opted for a list-like textual representation (as opposed to a diagrammatic representation such as enabled by business process modelling languages) and for graphical highlights in the Integrated Service System Model, belonging to the structural view. In the metamodel, the sequence of relationships is covered via the "relationship sequence" association class. We admit that complex behavioural patterns (eg, parallelization and loops) cannot be captured this way. This is an avenue for future research. Our rationale was that our research primarily focuses on the reconstruction of concepts and relationships from the SSS domain. Moreover, highlighting selected relationships, resources, and service systems in the Integrated Service System Model does not require modellers to be familiar with additional modelling notations.
Mutual benefit	*Attributes* (purpose, value proposition)	The mutual benefit of involved service systems is covered through attributes. The service concept features an attribute called purpose and service systems feature an attribute called value proposition. We decided to model purpose as attribute because it should be associated with the respective service. We decided to use a purpose attribute per service instead of value-in-use attributes per involved service system. The rationale is that each service system can experience manifold values-in-use depending on the services in which it is involved and depending on which other service systems are involved. This would be highly complex to consider for modellers in case multiple services are modelled. In that sense, the purpose associated with a service pragmatically summarizes the values-in-use of the involved service systems (acknowledging that values-in-use may be very different for the

(Continues)

TABLE A1 (Continued)

"Service is the application of specialized competences (operant resources—knowledge and skills), through deeds, processes, and performances for the benefit of another entity or the entity itself." (Vargo and Lusch 2008a, p. 26)
"[T]he 'applied' designation makes operant resources—resources that can act on or in concert with other resources to provide benefit (create value), as distinguished from operand resources—resources which require action to create benefit—primary." (Vargo and Lusch 2008a, p. 31)
"From S-D logic, service is the application of competence for the benefit of another. So service involves at least two entities, one applying competence and another integrating the applied competences with other resources (value-cocreation) and determining benefit." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
		involved service systems). In line with their property as "invitations for engagement in service," value propositions primarily depend on the related service system. They exist even if the related service system is not engaged in service. If accepted, however, value propositions relate to one or more services, reflecting which and how resources of the related service system are involved. This is why we decided to model value propositions both as attributes of service systems and as association class attached to the aggregation metarelationship between service system and service. Compared with other approaches, our DSML does not specifically focus on the modelling of value propositions, but on the application and integration of resources through service systems and relationships. In that sense, our DSML complements approaches geared towards the modelling of value propositions. Nevertheless, the purpose of a service is explicitly mentioned in Service Description Model.
Value cocreation		See "application of competences" and "mutual benefit" above.
Entity		See "service system" in Table A2.

Abbreviations: DSML, domain-specific modelling language; SSS, smart service systems.

TABLE A2 Design rationales related to the "Service System" concept

"[A service system is] a dynamic value-cocreation configuration of resources, including people, organizations, shared information (language, laws, measures, methods), and technology, all connected internally and externally to other service systems by value propositions." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
Service system	*Service system*	Service systems are essential as they are the entities involved in value cocreation. Service is enacted through interacting service systems. Hence, the metamodel contains a concept for service systems equipped with value propositions as attribute. Please see "mutual benefit" in Table A1 for details on value propositions. With service systems being resource configurations, there is an aggregation metarelationship between service systems and resources. As service systems can be involved in various services based on their value propositions, the metamodel also includes an aggregation

(Continues)

TABLE A2 (Continued)

"[A service system is] a dynamic value-cocreation configuration of resources, including people, organizations, shared information (language, laws, measures, methods), and technology, all connected internally and externally to other service systems by value propositions." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
		metarelationship between service systems and resources. Some resources are service systems themselves. This includes individuals and self-dependent smart things and digital hubs. Hence, we included a generalization metarelationship between service systems and selected resources. Together with the generalization metarelationship between resources and its subconcepts, the generalization between service systems and selected resource subconcepts accounts for the fact that service systems are resources themselves. This in turn enables the modelling of nested service system structures, a constitutive characteristic of SSS. Hence, we decided to split service systems into atomic and composed service systems, depending on whether they include further service systems. In line with the importance of smart things in the SSS domain, we also decided to distinguish between smart and ordinary service systems through a specialization metarelationship, depending on whether smart things are involved. In the metamodel, atomic SSS feature an aggregation metarelationship with smart things. More information about smart things can be found in Table A3 (smart product).
People	*Individual*	People are an important resource involved in SSS equipped with knowledge and skills. Through relationships, people interact with and parameterize smart things and digital hubs, and they can be observed by smart things. Among many other things, people also observe the physical environment or other individuals. In the metamodel, we included people as a separate concept. In line with the roles that people can play in value cocreation, we distinguish between service customers and participants through a specialization meta-relationship. We decided to treat individuals as resources because service systems are resource configurations and individuals can be part of service systems. Furthermore, the service system definition provided above lists people among the resources involved in service systems. Individuals are service systems themselves, a circumstance that we cover through a specialization metarelationship between individuals and service systems.
Organization	*Service system*	The metamodel does not include a concept for organizations. Our rationale is that organizations are covered by service systems. In fact, organizations are dynamic resource configurations that offer their customers specific value propositions. This interpretation complies with the resource-based view of the firm and dynamic capability theory. In B2C contexts, service becomes manifest through relationships among customers (which represent service systems), the organization's employees (which represent service systems and at the same time belong to the organization), the organization's digital hubs, or smart things as shared resources serving as boundary objects. The same logic applies in B2B contexts where the resources of at least two organizations are involved.
Technology	*Resource* (smart thing and digital hub)	SSS are enabled by technology. This relates to the existence of smart things and digital hubs, the availability of communication infrastructure, and the collection and analysis of data. Due to the

(Continues)

TABLE A2 (Continued)

"[A service system is] a dynamic value-cocreation configuration of resources, including people, organizations, shared information (language, laws, measures, methods), and technology, all connected internally and externally to other service systems by value propositions." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
	Relationship	ubiquity of technology, the metamodel does not include a separate concept. Rather, the role of technology becomes manifest in the resource and relationship concepts. That is, smart things (existing in the physical and digital world) and digital hubs (solely existing in the digital world) are technology-enabled resources. Further, the differentiation between transactional and analytical data usage as well as between dependent and self-dependent smart things and digital hubs is technology-driven, involving hardware and software such as algorithms, data storage, sensors, and actuators. As for relationships, all subconcepts require technology. For example, interaction and parameterization require connectivity, human-computer interfaces, computer-computer interfaces, or application programming interfaces (APIs). In addition, observations require sensors to observe individuals and the physical environment.
Information	*Relationship* *Attributes* (data, function, and data usage)	Analogous to the role of technology, the collection, storage, exchange, and analysis of data and information for transactional and analytical purposes is at the heart of SSS. We use the terms data and information as synonyms. Due to the ubiquity of data, the metamodel includes data attributes for digital hubs and smart things and data attributes for interactions from a relationship perspective. Further, the processing of data for transactional and analytical (including self-x) purposes is modelled via function attributes. Finally, smart things and digital hubs include a data usage attribute that indicates whether, across all functions, data is processed transactionally or analytically, whereby self-dependent smart things and digital hubs always use data analytically. Our rationale for assigning the data usage attribute to smart things and digital hubs as opposed to individual functions was that the second option would have been too complex. As analytical functions can include self-x components (see Table A3, smart product for an example), a strict separation is impossible.
Dynamic resource configuration	*Resource* (multiobjects) *Relationship* (parameterization) *Attributes* (function, data usage)	Dynamic change is constitutive for service systems in general and SSS in particular. In practice, this means that the internal status of resources can change, that new resources can enter or leave, and that relationships emerge. Typically, runtime dynamics are hard to cover in conceptual models. Our main idea was to include features in the metamodel that grant modellers degrees of freedom at design time. Our DSML caters for runtime dynamics by allowing for the modelling of multiobjects as placeholders for smart things, digital hubs, and individuals. This covers, for example, that a smart thing observes an arbitrary number of individuals. Independent from the fact that self-dependent smart things and digital hubs feature self-x functions running in the background, their functions can also be used to model self-x functions such as self-monitoring, self-diagnosis, self-optimization, and self-configuration. Self-x functions use the status quo and internal model of a smart thing or digital hub as input, enabling runtime dynamics through self-optimization and self-configuration. Furthermore, the parameterization relationship enables setting and adjusting goals of self-dependent smart things and

(Continues)

TABLE A2 (Continued)

"[A service system is] a dynamic value-cocreation configuration of resources, including people, organizations, shared information (language, laws, measures, methods), and technology, all connected internally and externally to other service systems by value propositions." (Maglio et al., 2009, p. 399)

Concept from the Literature	Corresponding DSML Elements	Design Rationale
		digital hubs, a property that provides related resources with degrees for freedom for decision making at runtime. That is, the concrete behaviour of self-dependent smart things and digital hubs does not need to be specified at design time. More information about self-x functions can be found in Table A3.
Value cocreation/ value proposition		See "application of competences" and "mutual benefit" in Table A1.

Abbreviations: B2B, business-to-business; B2C, business-to-consumer; DSML, domain-specific modelling language; SSS, smart service systems.

TABLE A3 Design rationales related to the "Smart Service System" concept

"Smart service systems are service systems in which smart products are boundary-objects that integrate resources and activities of the involved actors for mutual benefit."(Beverungen, Müller, et al., 2017, p. 6)

"A 'smart' service system is a system capable of learning, dynamic adaptation, and decision making based upon data received, transmitted, and/or processed to improve its response to a future situation. The system does so through self-detection, self-diagnosing, self-correcting, self-monitoring, self-organizing, self-replicating, or self-controlled functions. These capabilities are the result of the incorporation of technologies for sensing, actuation, coordination, communication, control, etc." (National Science Foundation, 2014, p. 5)

"A smart service system is a service system capable of learning, dynamic adaptation, and decision making […] that requires an intelligent object […] and involves intensive data and information interactions among people and organizations." (Lim and Maglio 2018, p. 155)

Concept from the Literature	Corresponding DSML elements	Design Rationale
Smart service system	Service system (atomic and composed smart service system)	SSS are the central unit of analysis of our DSML. Hence, the metamodel includes a concept for SSS, which is split into atomic and composed SSS to capture nested service system structures. We already provided a rationale for this decision and an explanation for how SSS are linked with other concepts in the metamodel in Table A2 (service system).
Smart product/ intelligent object	Resource (smart thing)	Smart products, which are also referred to as smart things or smart devices, are constitutive components of SSS. Hence, we included them in the metamodel as a subconcept of resources. Smart things that are part of multiple service systems serve as boundary objects. In line with the capabilities of smart things discussed in the literature, we distinguish between dependent and self-dependent smart things by means of a specialization metarelationship. Self-dependent smart things can act in line with goals without external intervention (sometimes even without external triggers), *and* they can learn by adjusting internal models of themselves. Furthermore, self-dependent smart things can be parameterized with goals and represent service systems themselves. Self-dependent smart things can also parameterize other self-dependent resources—except for individuals. Further information

(Continues)

TABLE A3 (Continued)

"Smart service systems are service systems in which smart products are boundary-objects that integrate resources and activities of the involved actors for mutual benefit."(Beverungen, Müller, et al., 2017, p. 6)
"A 'smart' service system is a system capable of learning, dynamic adaptation, and decision making based upon data received, transmitted, and/or processed to improve its response to a future situation. The system does so through self-detection, self-diagnosing, self-correcting, self-monitoring, self-organizing, self-replicating, or self-controlled functions. These capabilities are the result of the incorporation of technologies for sensing, actuation, coordination, communication, control, etc." (National Science Foundation, 2014, p. 5)
"A smart service system is a service system capable of learning, dynamic adaptation, and decision making [...] that requires an intelligent object [...] and involves intensive data and information interactions among people and organizations." (Lim and Maglio 2018, p. 155)

Concept from the Literature	Corresponding DSML elements	Design Rationale
		about the attributes of smart things and the embedding of smart things in the metamodel can be found in Table A1 (resources) and Table A2 (service system, information, and dynamic resource configuration).
Self-x functions/ Learning	Resource (smart thing and digital hub) Relationship	Beyond the integration of smart things, self-x functions are constitutive for SSS. Self-x functions, which are also referred to as self-x capabilities, include basic self-x functions (eg, self-monitoring or self-diagnosis) and extended self-x functions (eg, self-optimization, self-reconfiguration, or self-healing). The metamodel caters for self-x functions in multiple ways. First, the functions of smart things and digital hubs can include self-x components. For example, a smart vacuum cleaning robot that cleans a room uses basic self-x functions embedded in ordinary functions to find its way in a self-controlled manner. In most cases, however, vacuum cleaning robots do not adjust their behaviour from one cleaning run to the other, as this would require extended self-x functions that enable learning. Such extended self-x functions are associated with self-dependent smart things or digital hubs. In self-dependent smart things and digital hubs, extended self-x functions are running in the background. The metamodel accounts for this circumstance by including separate concepts for self-dependent smart things and digital hubs. However, if deemed appropriate by modellers, extended self-x functions can also be modelled explicitly via the function attributes of smart things or digital hubs. Further information can be found in Table A2 (information, dynamic resource configurations).
Service system		See "service system" in Table A2.
Data/information		See "information" in Table A2.
Technology		See "technology" in Table A2.
Mutual benefit		See "application of competences" and "mutual benefit" in Table A1.
Dynamic adaptation		See "dynamic resource configuration" in Table A2 and "self-x functions/learning" above.
Resource		See "resource" in Table A1.
People		See "people" in Table A2.
Organization		See "organization" in Table A2.

Abbreviations: DSML, domain-specific modelling language; SSS, smart service systems.

TABLE A4 Real-world scenarios used for the development of the DSML

#	IoT Domain	Name	Source	Major Changes to the DSML
1	Smart City (*Smart Home*)	Nest's Remote Control Services	https://nest.com/support/article/Learn-how-Nest-products-work-together	• Decision to add "digital hubs" as a resource type to cover resources that only have a representation in the digital world
2	Smart City (*Smart Home*)	Nest's Smart Smoke Sensor Services	https://nest.com/support/article/Learn-how-Nest-products-work-together	• Decision to add "observation" as a relationship type in order to appropriately cover the sensing capabilities of smart things and individuals • Decision to add "physical environment" as a resource type to cover the observation of real-world properties and nonsmart objects
3	Smart City (*Smart Home*)	Nest's Motion Detection Service	http://www.howtogeek.com/249093/how-to-use-your-nest-thermostat-as-a-motion-detector/	• Decision to add relationship sequences to facilitate the modelling of value cocreation in SSS from a behavioural perspective
4	Smart City (*Smart Home*)	Nest's Smart Heating and Cooling Service	https://developers.nest.com https://nest.com/support/article/What-is-Rush-Hour-Rewards https://ifttt.com/applets/DQgDXhzt-a-warm-welcome-home	• Decision to add "parameterization" as a relationship type to complement transactional and functions-oriented "interactions" and to equip self-dependent resources with degrees of freedom for action and decision • Decision to specify detailed connection rules between relationships and resources as shown in Appendix A3 to cover that relationships and resources must not be combined arbitrarily
5	Smart City (*Smart Mobility*)	Tesla's Fleet Learning Service	http://fortune.com/2015/10/16/how-tesla-autopilot-learns/	• No major change
6	Smart City (*Smart Mobility*)	Pisa's Smart Parking Service	http://www.dailymail.co.uk/sciencetech/article-2667536/Smart-LAMP-POSTS-end-parking-woes-App-locates-space-lights-guide-spot.html	• Decision to model services via rectangular frames that include all involved service systems, resources, and relationships to enable a structural perspective on value cocreation in SSS and to enable the modelling

(Continues)

TABLE A4 (Continued)

#	IoT Domain	Name	Source	Major Changes to the DSML
				of nested service system structures
7	Industrial (*Logistic & Product Lifetime Management*)	Amazon's Smart Shopping Service	https://www.amazon.com/b?node=16008589011	• Decision to use multiobjects for modelling resources that are not known at design time but may enter or leave at runtime to cope with the dynamic adaptation of SSS
8	Industrial (*Logistic & Product Lifetime Management*)	Hospital 4.0	http://hospital40.net/	• Decision to distinguish between repeated and nonrepeated as well as between hierarchical and nonhierarchical interactions to cover a broader range of interaction relationships
9	Industrial (*Industrial Processing*)	Smart Devices Support Services	http://www.ibtimes.com.au/volkswagen-factory-workers-using-augmented-reality-tech-3d-smart-glasses-1487284	• No major changes
10	Industrial (*Real-time Vehicle Diagnostic*)	Engine Health Management	https://www.rolls-royce.com/products-and-services/civil-aerospace/aftermarket-services.aspx#/ https://www.sitaonair.aero/sitaonair-selected-rolls-royce-simplify-collection-engine-health-monitoring-data/	• No major changes
11	Health well-being (*Individual Well-being*)	Fitness Tracking Service	https://www.fitbit.com/de/home	• No major changes
12	Health well-being (*Elderly Assistance*)	Safety Watch Service For Elderly People	http://www.mylively.com/	• No major changes
13	Health well-being (*Smart Hospital Services*)	Smart Hospital Service Ecosystem	https://www.enisa.europa.eu/publications/cyber-security-and-resilience-for-smart-hospitals	• No major changes

Abbreviations: DSML, domain-specific modelling language; SSS, smart service systems.

APPENDIX C

METARELATIONSHIP BETWEEN THE CONCEPTS "RELATIONSHIP" AND "RESOURCE"

Resources cannot be connected arbitrarily. Admissible connections depend on the involved subconcepts of resources and relationships. Figure A1 provides an overview of admissible connections in terms of a detailed metamodel that complements the metamodel in the paper.

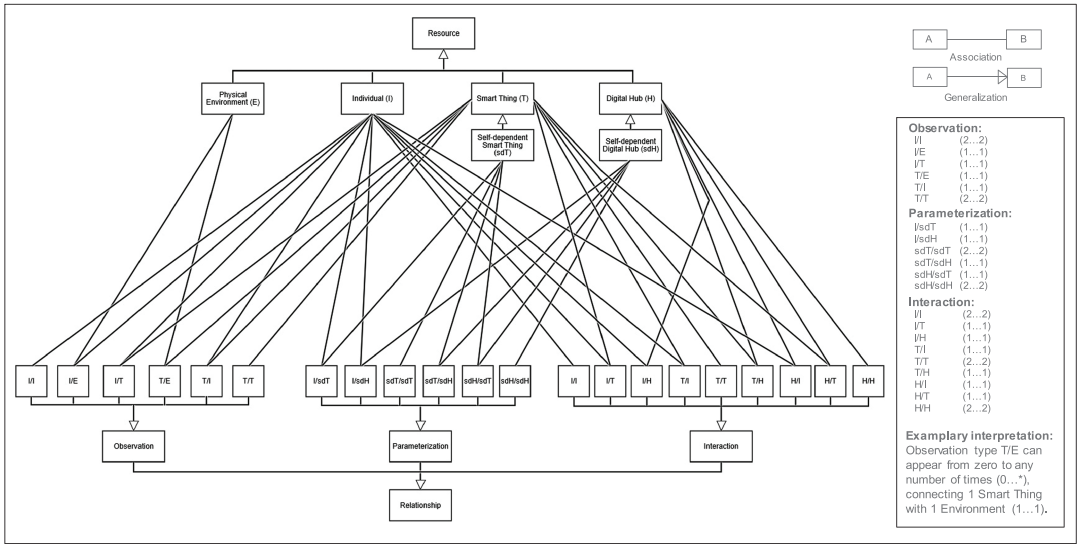


FIGURE A1 Detailed metamodel for connecting "Resources" via "Relationships"

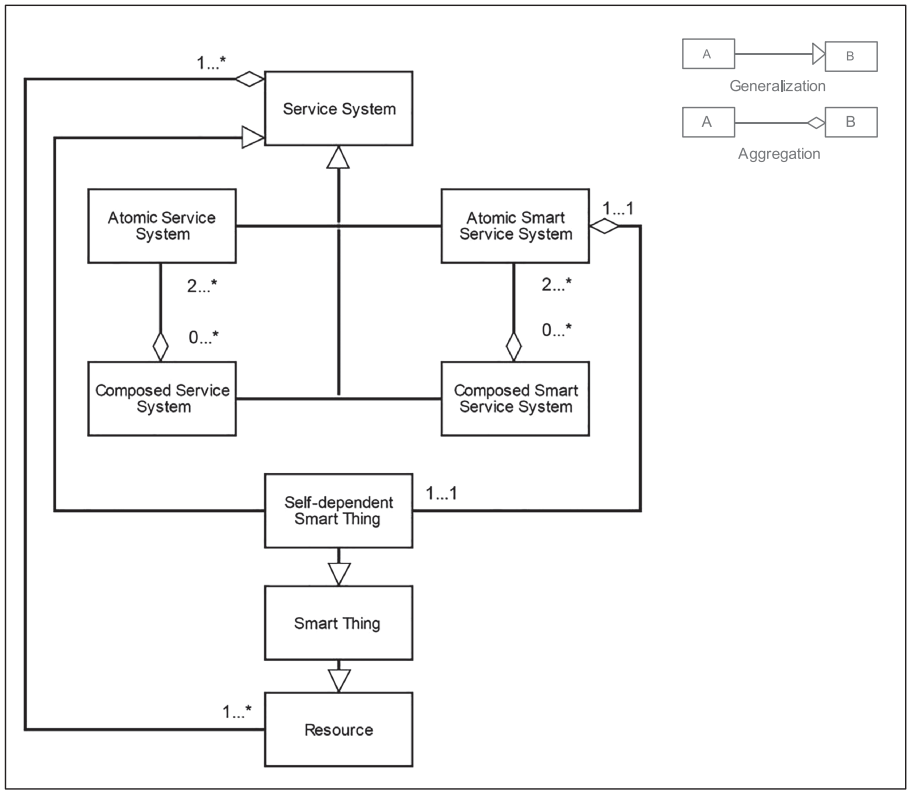


FIGURE A2 Detailed metamodel for connecting the subconcepts of "Service Systems"

APPENDIX D

METARELATIONSHIP BETWEEN THE SUBCONCEPTS OF "SERVICE SYSTEM"

Service systems cannot be connected arbitrarily with resources. Moreover, there are relationships among the subconcepts of service systems. Figure A2 provides an overview of admissible connections in terms of a detailed metamodel that complements the metamodel in the paper. It needs to be considered that the metamodel notation cannot capture all connection rules, eg, n-ary aggregation metarelationships with an "OR" or "XOR" logic or prohibited connections. Please refer to the manuscript for details.

APPENDIX E

FEEDBACK FROM THE EXPERT INTERVIEWS

Table A5 includes the feedback from the expert interviews and how we tackled this feedback. The statements listed in the table have been translated verbatim from German into English as all interviews were conducted in German.

APPENDIX F

RESULTS OF FEATURE COMPARISON

As part of the evaluation, we discussed the characteristics of our DSML against Frank's (2013) generic requirements (GRs) for DSMLs. These requirements are familiarity (GR1), invariability (GR2), level of detail and abstraction (GR3), and mapping of language concepts (GR4). This discussion enabled us to assess whether our DSML meets the requirements of DSMLs as a specific artefact type. Based on this assessment, we conclude that our DSML appropriately meets the requirements of DSMLs.

(GR1) Familiarity. To ensure that modellers and model users are familiar with concepts, relationships, and notational elements, the DSML draws on extant knowledge from the internet of things (IoT) and service domains. We also reused concepts and notational elements of existing modelling approaches wherever appropriate, and we discussed the abstract and concrete syntax with industry experts based on a simple and a complex real-world scenario. The experts agreed that the concepts and relationships included in the DSML are clearly defined and that the notational elements can be intuitively understood. However, we admit that the development of our DSML primarily focused on the reconstruction of concepts and relationships from the SSS domain and their compilation in terms of a metamodel. Notational elements—except for those related to parameterization and observation—were reused. Hence, they need to be refined and further evaluated in future research.

(GR2) Invariability. Invariability requires the semantics of modelling concepts to be constant within the range of their intended use. This ensures that the DSML can be applied in different contexts within its target domain. To ensure invariability, we developed our DSML iteratively, not only by drawing on the literature but also on simple and complex real-world scenarios from various functional IoT domains. We covered all IoT domains listed by Borgia (2014). An overview of the used scenarios and how they influenced the DSML can be found in Appendix A2. Furthermore, we discussed the DSML with experts from various organizations and industries. Regardless of their background, the experts were asked to discuss the same examples and to provide additional examples based on their own experience. As the experts had no difficulties using the DSML and as they appreciated the clear definition of concepts and relationships, we pose that the semantics of our DSML are stable across different contexts.

(GR3) Level of detail and abstraction. The DSML enables modelling at different levels of detail and abstraction. Firstly, some characteristics of resources and relationships (eg, transactional vs analytical) are optional and should only be modelled if needed. The same holds for most attributes (eg, data and functions). Secondly, the nested structure of service systems enables switching between different levels of detail (ie, atomic vs composed service systems). Thirdly, by allowing for multiobjects, our DSML enables summarizing similar resources and working with

TABLE A5 Detailed expert feedback

Topic	Verbatims from the Expert Interviews	Our Response/Reaction
1. Understanding of SSS	**A:** "As there are different types of SSS, for example, ones which function without any human interaction as intelligent booking systems, I would like to know what your definition of SSS is and which ones you cover with your DSML?" (SERVICE II) B: "Per definition, which elements need to be part of an IoT-enabled SSS?" (SERVICE III)	**A + B:** We shared our definition of SSS with the expert and included it prominently in the manuscript. We also clarified that our DSML aims to model IoT-enabled SSS, ie, SSS that include smart things.
2.1. Core metamodel (Abstract syntax)	**A:** "In a business context, it would make sense to differentiate services between paid and unpaid ones" (PRODUCTION I) **B:** "Does every smart thing need a service which it is connected to? Wouldn't it make sense to just model the smart thing and then see what services are enabled by it?" (SERVICE IV)	**A:** The differentiation between paid and unpaid service is reflected in the relationships involved in a distinct service. For example, a paid service would include interactions that refer to payments. **B:** Every resource needs to be involved in at least one service, as defined by the metamodel. As the DSML is a language not a method, it does not prescribe a distinct procedure. Thus, it allows for modelling smart things first and identifying services afterwards.
2.2. Resources (Abstract syntax)	***Individual (I)*** **I1:** "I see more different roles of individuals, such as passive participants (eg, are observed by the smart thing) and active participants (eg, call agent who is phoning the customer)" (SERVICE II) ***Smart Thing (S)*** **S1:** "Differentiating smart things only into 'open' and 'closed' ones might be problematic as there are also mixed forms. For example, as far as I know car manufacturers grant their business partners only a limited access to their platform." (PRODUCTION I) **S2:** "The distinction between transactional and analytical smart things is not 100% clear. I am not sure if I can easily differentiate between both, as often it will be something in between" (SERVICE IV) **S3:** "In reality, it is often difficult to define the actual place of smartness as 'the magic' could be done in the smart thing, the connected server or even the cell phone which is used to address the smart thing. How does your DSML and especially the concept of smart things capture this aspect?" (SERVICE IV) **S4:** "Are there still closed smart things? Most smart things I know are used as a platform. If you find some, then the differentiation is valid." (SERVICE V)	***Individual (I)*** **I1:** The differentiation between active and passive service participants is reflected in the relationships that these individuals are involved in. We did not include a further specialization in the metamodel to keep it parsimonious. ***Smart Thing (S)*** **S1:** We sharpened the distinction between closed and open smart things in the manuscript. We highlighted that hybrid forms are likely to occur in industry. **S2:** We sharpened the distinction between analytical and transactional smart things in the manuscript. We highlighted that hybrid forms are likely to occur in industry. **S3:** We shared our understanding of smartness with the expert and included it prominently in the manuscript. We clarified that our DSML offers various means for covering smartness, ie, the distinction in analytical and transactional smart things and hubs, the usage of function and data attributes, and the usage of different relationship subconcepts. Thus, smartness is covered in the structural and the behavioural view. **S4:** Many smart things can be characterized as closed. This is supported by extant literature and by the examples we modelled. We included related references and examples in the manuscript.

(Continues)

TABLE A5 (Continued)

Topic	Verbatims from the Expert Interviews	Our Response/Reaction
	Digital Hub (D) **D1**: "Regarding your concept of digital hubs, the differentiation we have for digital hubs and platforms, we do not make a big difference among both, are those which allow access for third-party services and those which allow new content, such as the App Store or Google Play." (SERVICE II) ***Physical Environment (P)*** -	***Digital Hub (D)*** **D1**: We included the expert's thoughts in our definition of open and closed digital hubs. We have the same understanding as the expert. ***Physical Environment (P)*** -
2.3. Relationship (Abstract syntax)	**A:** "I can't perfectly make a distinction between the relationship types of parameterization and interaction. Isn't it possible to use interactions as a short-term parameterization?" (SERVICE III) **B:** "Isn't parameterization a specific form of interaction?" (PROCUCTION II) **C:** "Can smart things also observe other smart things? As for example one smart car another smart car? Currently this aspect is not enabled in your DSML as you limit resources to only observe the physical environment." (SERVICE V)	**A + B:** We shared our understanding of interaction and parametrization with the expert and included it prominently in the manuscript. We clarified that interactions are restricted to a distinct point in time and have a transactional nature, whereas parameterizations refer to a period and goals. Further, only analytical smart things and digital hubs can be parameterized. **C:** Based on this suggestion, we modified the metamodel such that smart things can observe other smart things, individual things, and the physical environment.
3. Concrete syntax	**A:** "The notations for service systems and digital hubs are very similar—it would make sense to give one a normal and the other one a round edge." (SERVICE II) **B:** "As the creation of a service includes different data and functions, it might make sense and benefit the DSML by including such one on the interaction arrows with a few words." (SERVICE I) **C:** "I would keep the phrasing on the interaction arrows more generic as one model should be reusable for many different services." (PRODUCTION I) **D:** "I would add a C and P to the notation of the individual to easily, visually different between both." (SERVICE V)	**A:** Based on this suggestion, we modified the concrete syntax and changed the notational element for service systems. B: Based on this suggestion, we added data and function attributes to the metamodel. **C:** We refrained from incorporating this suggestion as each the naming of relationships should depend on the modelling goals that hold for a concrete scenario. Nevertheless, we consider model reusability an important topic. **D:** Based on this suggestion, we modified the concrete syntax and changed the notational element for service customers and participants.
4. Nest example	**A:** "I do not really understand how service and service systems are connected as you have two different models, the Integrated System Model and the Service Description Model." (SERVICE I) **B:** "Does a system also exist if there is no particular service connected to it? This might be an important question as you differentiate between a service and system view." (PRODUCTION II) **C:** "As the Nest example is rather complex and the additional graphic for the service description makes it even more complex, I	**A:** Based on this suggestion, we modified the metamodel as well as the views on the metamodel to include the service concept. We also dropped a second model type from the behavioural view that focused on service hierarchies to avoid redundancies between views. **B:** We shared our understanding of service systems with the expert and included it prominently in the manuscript. Services systems are characterized by value propositions and exist independently from services.

(Continues)

TABLE A5 (Continued)

Topic	Verbatims from the Expert Interviews	Our Response/Reaction
	would recommend highlighting all systems, resources and relationships which are involved in a particular service and grey out the rest." (SERVICE V)	**C:** We accounted for this suggestion in the demonstration section. However, the graphical highlighting cannot be covered by the metamodel. It must be implemented in a modelling tool.
5. Further research	**A:** "A guidance for modeling would be useful as what to model first, second, third, and so on …" (SERVICE I) **B:** "However, to make it a success story in my company I would need a software tool for the modeling and a guarantee that someone is taking care of and updating it … so that my organization rely on it" (SERVICE VI)	**A:** We agree with this suggestion. We included the development of a modelling method for IoT-enabled SSS that complements our DSML as a topic for future research. **B:** We agree with this suggestion. We included the development of a formal metamodel and the implementation of a modelling tool as topics for future research.

Abbreviations: DSML, domain-specific modelling language; IoT, internet of things; SSS, smart service systems.

placeholders. Fourthly, the structural and behavioural views lead to different model types—the Integrated System Service Model and a Service Description Model per service—which reduce modelling complexity by focusing on specific modelling objectives and enable working at different levels of detail. Beyond these conceptual ways of switching between different levels of detail anchored in the metamodel, we discussed simple and complex real-world scenarios with industry experts. In both cases, the experts deemed the chosen level of detail to be appropriate, and they found the means for switching between different levels of detail to be of practical use. Although the DSML meets the invariability requirement, we admit that we omitted concepts such as "collapsed service systems" that would further support the switching between different levels of abstraction, because our primary objective was to reconstruct domain-specific concepts and relationships. More technical concepts should be added in future research when the semiformal metamodel is further developed into a formal metamodel that serves as foundation for IT-based modelling tools.

(GR4) Mapping of language concepts. The last GR involves the mapping of language concepts onto concepts of relevant target representations. If possible, all information required by target representations should be extracted from the model. We addressed this requirement by introducing different specifications for resources and relationships. For example, smart things and digital hubs play an essential role in SSS, and so we distinguish their most important characteristics to ensure a clear mapping of real-world examples to target representations (eg, smart thing with analytical capabilities and those without). We also allowed for the specification of data, functions, goals, and properties. As the DSML takes a conceptual rather than a technical perspective on SSS, we admit that the models created with our DSML do not contain all the information needed to implement an SSS. This reflects a deliberate design decision, as our objective was to take a conceptual perspective. Accordingly, more technical modelling approaches should be used when it comes to the implementation of SSS.