Enterprise Modelling and Information Systems Architectures
Vol. 13, No. 14 (2018). DOI:10.18417/emisa.13.14

Special Issue on Smart Service Engineering

A Classification Framework for Service Modularization Methods

Jens Poeppelbuss*,[a], Aleksander Lubarski[b]

[a] Ruhr-Universität Bochum, Germany
[b] University of Bremen, Germany

Abstract. *Service modularity has been suggested as a promising concept that can resolve the dilemma between increasing diversification of customer demands and the provider's need for standardization and efficiency gains. Despite having been in the center of attention amongst service researchers for the past decade, service modularity still remains a rather theoretical concept with little application in practice. Previous publications have contributed conceptual and enterprise modeling methods to achieve modular service architectures by both adjusting product modularization methods to the service domain as well as designing new ones specifically for services. However, up to date, there exists no framework that would systematize and classify these methods concerning their premises as well as underlying modularity principles and objectives. The main contribution of this paper is the development of a framework that can be used to classify existing and future methods for service modularization based on two key dimensions, i. e., the phases of the modularization process and the types of structuring the modular architecture. The developed framework further points out which phases of the overall modularization process are still underdeveloped and how future research can contribute to making service modularity more accessible for practitioners.*

Keywords. Service Modularity • Modularization • Service Engineering • Classification Framework

Communicated by O. Thomas. Received 2017-03-10. Accepted after 2 revisions on 2018-06-10.

1 Introduction

The current trend of individualization forces service providers to satisfy increasingly diversified demands of heterogeneous customers by offering tailored products and services (Bask et al. 2011a). At the same time, companies are urged to standardize internal processes (Böttcher and Klingner 2011) and control the variety of their offerings in order to maintain profitability and competitiveness. The concept of modularity is generally considered to be a suitable solution for this dilemma as it promises a wide market coverage with limited additional costs (Pekkarinen and Ulkuniemi 2008). The main principle of modularity is that a complex system can be built from smaller parts (modules), which can be designed, improved and substituted independently, yet function together as a whole (Baldwin and Clark 1999). This idea of loosely coupled modules results in interchangeability and flexibility in the value creation process, as long as the interfaces between separate modules are well-defined and standardized (Arnheiter and Harren 2005).

Apart from its origin in manufacturing, the concept of modularity has also been widely used in other fields, especially software engineering. As for the service domain, however, the concept still remains in its infancy both in terms of academic

---

* Corresponding author.
E-mail. jens.poeppelbuss@rub.de
This work was supported by the WFB Wirtschaftsförderung Bremen GmbH (grant number FUE0576B), the European Regional Development Fund from the European Union, and encoway GmbH.
Note: This work is based on Poeppelbuss and Lubarski (2016).

discussion and practical application (Carlborg and Kindström 2014). Previous studies have discussed the concept of modularity in general (Leseure et al. 2010; Lin et al. 2010; Pekkarinen and Ulkuniemi 2008), conducted literature reviews (Dörbecker and Böhmann 2013; Tuunanen et al. 2012) or assessed modularity potentials for specific application scenarios (Bask et al. 2011b; Böhmann and Loser 2005; Carlborg and Kindström 2014). A lot of attention has also been devoted towards modularization methods that allow for conceptually modeling modular service architectures. These have either been adapted from product modularity to the service domain (Dörbecker et al. 2014) or have been specifically designed for services (Lin and Pekkarinen 2011). The expected end result of applying such methods is typically a modular service architecture. A modular service architecture affects both the business model and the IT infrastructure of an organization and hence covers both management and technical aspects of providing a variety of services in an efficient manner (for a broader discussion on these two perspectives on service orientation see also Demirkan et al. (2009)).

Despite having been on the agenda of service researchers for the past decade, service modularity still remains a mostly theoretical concept with little attention in practice. Even though several distinct modularization methods have been introduced and applied in case studies (Dörbecker et al. 2014; Peters and Leimeister 2013), they typically do not cover the whole modularization process, but instead concentrate on single phases only (e. g., decomposition of monolithic service offerings, structuring of the elements, or module creation based on atomic service elements). In addition, most of the existing methods make simplified assumptions, which are not necessarily valid for service providers in reality (e. g., the existence of an already well-defined and clearly decomposed service portfolio or a comprehensive transparency of service processes). Up to date, there exists neither a clear overview about what stages a modularization process actually consists of and in how far these stages are interdependent in terms of inputs and outputs, nor is there a classification of modularization methods based on their specific characteristics that can help practitioners in selecting and adopting suitable methods.

The purpose of this article is therefore to develop an appropriate framework that fills this research gap. The framework that we present in this article is the result of an iterative research process and has been synthesized from existing literature on modularization methods from the field of enterprise modeling as well as operations research. It delivers a twofold contribution. First, practitioners are provided with a better understanding of the phases and characteristics of the modularization process, thus supporting its real-life implementation. Second, service researchers can use this framework as an orientation guide for identifying further academic void.

The remainder of this article is structured as follows. In Sect. 2, we elaborate on the distinctiveness of service modularity as well as its implications for modularization methods. Sect. 3 continues with the research methodology. The framework itself is explained in detail in Sect. 4, which is followed by the illustration of its applicability to existing methods in Sect. 5. In the final Sect. 6 6, we discuss the theoretical and managerial implications, point out the limitations of our research, as well as present future research opportunities.

2 Service Modularity

Modularity is a design principle used to build complex products or services out of separate components (modules). The modules can be improved and substituted independently without affecting the entire system and yet function as a whole (Baldwin and Clark 1999). Linked to "modularity" as the basic concept, "modularization" denotes the actual transformation process, and a "modular architecture" is the desired result of it. Previous applications of the modularity concept include product development (e. g., configurable cars that can be "built together" by the customer based on his individual preferences) and software engineering (e. g., customization of application systems and large software packages). Transfers to other domains have also been discussed (Voss and Hsuan 2009).

The benefits of modularization include, amongst others, the reusability of individual components for future offerings (Carlborg and Kindström 2014), faster development cycles (Böttcher and Klingner 2011), economies of scale and scope (Tuunanen et al. 2012) as well as cost efficiency in general (Bask et al. 2011a). Additionally, modularity can positively affect the user experience and value perception (Rahikka et al. 2011) since customers (both private and business) are enabled to configure their own individual bundle of products and/or services out of a limited number of possible alternative modules, thus having more transparency over what is possible and how different product/service options influence the final price. Specifically, Rahikka et al. (2011) find that this user-driven bundling based on a modular service portfolio not only enriches the overall user experience, but also makes quality evaluation easier, develops trust, and strengthens the long-term relationship between customer and service provider. Finally, modularization can also be used to improve calculations, decision making, and the preparation of quotes and tenders during service sales.

Similar to manufacturing firms, service providers face challenges in terms of developing offerings that are both flexible and customizable in order to fit the specific requirements of the customer without major additional investments or high operational costs (Edvardsson and Enquist 2007). Moreover, even manufacturing enterprises increasingly recognize the importance of (individualized) value-added and product-related services (Vargo and Lusch 2004) and can thus profit from a modular service architecture. Several case studies have already confirmed that there is a need for modular service portfolios amongst practitioners, especially in healthcare (de Blok et al. 2010; Peters and Leimeister 2013), remote monitoring (Carlborg and Kindström 2014) or logistics (Bask et al. 2011b; Lubarski and Poeppelbuss 2016). An extensive list of services with modular design and what industries have the highest potential for service modularity can be found in the works of Iman (2016) and Brax et al. (2017).

It is frequently argued that due to the specificity of services, the principles of product modularity cannot be transferred to services directly or easily, but require appropriate adjustments (Bask et al. 2011a; Iman 2016; Voss and Hsuan 2009). First of all, the intangibility of services makes pre-production and storage of services nearly impossible, which is a common practice with tangible products (de Blok et al. 2010). Second, due to the process nature of services and close interaction between provider and customer (integrating the so-called external factor), finding an appropriate level of decomposition (Leseure et al. 2010), as well as defining standardized interfaces between separate modules (Blok et al. 2014) is not as straightforward and intuitive as with tangible products. Lastly, a high heterogeneity and the allegedly uniqueness of single service instances is one of the main arguments for the general skepticism towards service standardization and modularity (Lubarski and Poeppelbuss 2016). A good overview of the key differences between modularization in the production and service industry in terms of modules, interfaces and an overall architecture can be found in Pekkarinen and Ulkuniemi (2008) and Leseure et al. (2010).

While the need for adjusting methods from product modularization to the service domain and for designing new service-specific ones has been discussed widely (Iman 2016; Lin and Pekkarinen 2011; Pekkarinen and Ulkuniemi 2008; Peters and Leimeister 2013), it still remains unclear what exact phases constitute a modularization process and what preconditions (inputs, e. g., a list with identified basic elements of a service) or results (outputs, e. g., a module that is an aggregation of basic elements) are linked to each phase. Most of the previous publications only address a specific subtask of the overall modularization process, which points to the need for a framework that allows to structure existing and future modularization methods.

3 Methodology

We applied a hermeneutic approach for searching and analyzing the literature and developing the framework (Boell and Cecez-Kecmanovic 2014). The idea of the hermeneutic approach is that a literature review should be carried out in an iterative manner (so-called hermeneutic circle). The researchers' initial understanding of the topic is refined in the course of multiple iterations so that they do not get lost in the variety of articles and finally succeed in developing a distinct and unique contribution.

The starting point of this research was a comprehensive literature search for academic contributions on service modularity with a special attention to modularization methods. We scanned online databases (including ISI Web of Knowledge, Google Scholar, EBSCO, Science Direct, Elsevier, and JSTOR) for relevant journal and conference publications. For searching the online databases, we began with the search terms "*service*" in combination with one of the following terms "*modularity*", "*modularization*", "*modularisation*" and "*modular*". In addition, we conducted backward and forward searches based on the identified publications and included further relevant publications that were recommended to us by fellow researchers (e. g., during the review process of this article). Based on our ongoing analysis, we extended our search terms by including additional keywords (e. g., "*module building*", "*modularization interfaces*", "*service decomposition*", "*modular architecture*") and also searched for the application of the modularization concept in specific service domains (e. g., "*modular logistics*").

Following the hermeneutic approach, the processes of searching and analyzing the literature were intertwined. Relevant publications were used to identify dimensions and characteristics that can provide a structure for the framework to be developed. We also looked for additional attributes to create a method profile template. The identified methods were analyzed and described in such profiles, with these profiles being iteratively updated as long as further relevant attributes were identified (see Tab. 1). Along with the general data, each method profile contains the information regarding the placement of a method in the overall modularization process, its type of modular structuring, as well as what inputs and outputs it is associated with. Furthermore, we give practical suggestions on what visualization tools and generic enterprise modeling notation can be used with a selected method, and how it should be adjusted for services, in case it was originally suggested for the context of products.

During the course of searching and analyzing the literature, we continuously updated the structure of the framework and the assignment of the methods within the framework. The iterations were repeated until no further changes were observed. The overall research methodology is presented in Fig. 1.

Our sample of methods deliberately includes service-specific modularization methods as well as methods that originally were directed towards modular products. According to Müller and Lubarski (2016), the realms of manufacturing and information systems are the most influential roots for service modularity, which can be confirmed when looking at the outlets most of the early publications were published. Even the articles that appeared in traditional service and marketing journals (e. g., Tuunanen et al. (2012) and Tuunanen and Cassab (2011)) explicitly draw on insights and terminology from modularity in the manufacturing and information systems context. Therefore, in addition to the methods purposefully designed for service modularization, we also looked at methods from product modularity that have a potential to be employed in the context of services if certain adjustments are made. For example, Duckwitz et al. (2015) uses an adjusted version of the DSM for the modeling and simulation of knowledge-intensive service systems, thus confirming its general applicability. Similarly, Erixon (1996) points out that his method of a modular function deployment can be applied also in the context of services, if additional evaluation criteria for the modules are defined.

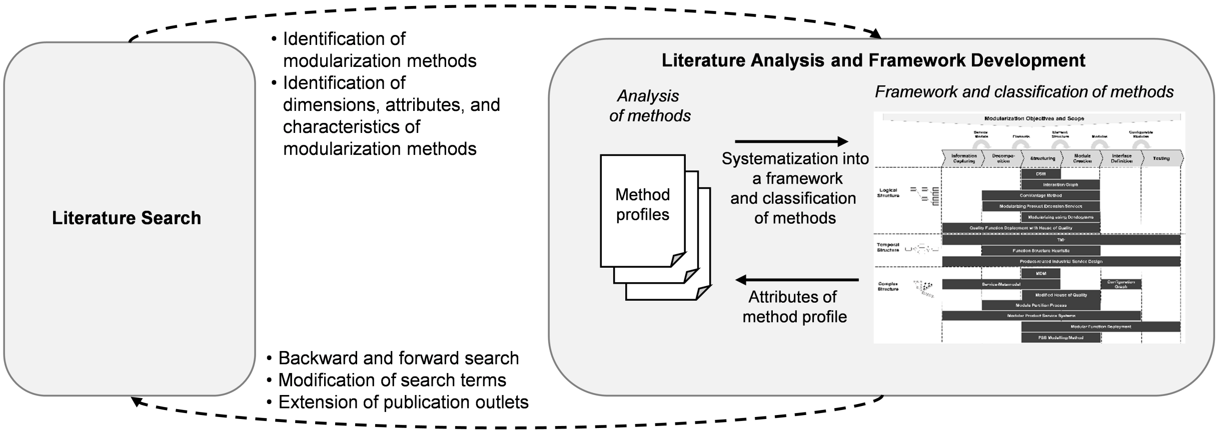

Fig. 1: Methodology for the development of the classification framework

Overall, we were able to identify 16 relevant candidates for our sample (Tab. 2). Based on the initial origin and motivation for their development, they can be categorized into three different groups:

- *Product methods (P)*. Methods falling in this category were originally developed for product modularization, driven by the idea of mass customization and global outsourcing strategies. The underlying principles are therefore usually product-oriented (e. g., related to design properties, material flows, or physical interconnections).
- *Service methods (S)*. As soon as the transfer of the modularity concept was discussed also in the realm of services, service researchers argued that service-specific methods were needed. This motivated them to develop appropriate methods for services in general, or even specific application areas within the service sector.
- *Product-Service combination (P+S)*. This research stream argues that most of the valuable (industrial) services like technical repairs or energy management are genuinely part of product service systems (PSS). They underline the inseparability of products and services and the need for their simultaneous engineering and, hence, modularization.

We excluded articles like Carlborg and Kindström (2014) or Bask et al. (2011a), as they solely present a classification of possible modularity strategies, but no actual distinct methods. Similarly, generic conceptual and enterprise modeling notations such as Business Process Model and Notation (BPMN), Event-driven Process Chain (EPC) or Petri nets were also not considered as distinct modularization methods. However, they can be considered as helpful visualization tools that can be applied for modeling tasks within the modularization process (e. g., during the initial phase of information capturing; see below).

4 Classification Framework

4.1 Build-Time and Run-Time of Modularity

Before introducing the framework itself, it is important to clarify the overall structure and the underlying motivation of modularization initiatives and for what purpose the results of such initiatives serve. In general, all required steps for the restructuring of a service portfolio can be categorized as *variant management* or *build-time*. The modularization process results in a modular architecture that will be used in *sales operations* or *run-time* (Fig. 2). Most of the existing literature on modularization methods focuses on variant management and, hence, the build-time of service modularity (Lubarski and Poeppelbuss 2016; Peters and Leimeister 2013). Correspondingly, the

Tab. 1: Exemplary method profile (the profiles of all methods are accessible under www.bakerstreet-projekt.de)

Name	Design Structure Matrix (DSM)
Source	Steward (1981), Corsten and Salewski (2013)
Purpose	Originally from product design, its purpose is to define the precedence of design variables and the dependencies between different design variables. In the service field, it is used to describe interdependencies between service elements or tasks.
Basic Idea	Variables or elements are listed in an array which forms both the horizontal and vertical axis of a matrix. Interdependencies between these variables or elements are defined in the cells of the matrix. They are typically rated with a score (e. g., a value that reflects the coordination costs resulting from the interdependencies) or a Boolean variable (reflecting yes/no or dependent/independent).
Phase	Structuring
Inputs	List of variables or service elements/tasks
Outputs	Matrix with interdependencies between variables/elements/tasks
Type	Logical structure
Visualization tool	Any software that allows a matrix representation. Complex matrix calculations can be executed using Matlab software environment, for instance.
Prerequisites	The DSM method requires a list of clearly defined and disjoint elements. It does not consider how to previously decompose services into their elements or tasks. The interdependencies between elements are often rated subjectively, e. g., by experts or service workers.
Procedure	1. Define list of elements 2. Rate relationships/interdependencies between elements 3. Generate modules by analyzing the interdependencies (e. g., numerical optimization with the objective to minimize inter-module coordination costs)
Adjustments for services	The DSM method can be applied to services if the smallest possible elements as well as their interfaces are well-defined. Special attention has to be devoted to the process sequence as the temporal dependency states an additional interface condition. An exemplary application of the DSM to knowledge-intensive service systems can be found in Duckwitz et al. (2015)

presented classification framework will refer to modularization methods used during build-time.

The modularization process that takes place during build-time is usually an internal strategic back-office process closely linked to general service engineering activities. It generates the master data of the service portfolio, which, in turn, provides the necessary data base for the software used in the sales department during the quotation process at run-time. During the quotation process, the master data of the modular service portfolio provides the basis for configuring and pricing a service bundle and for generating the actual quotation document. In order to improve the efficiency of quotation processes, they can be supported by the use of so-called configure-price-quote (CPQ) software (Gartner IT Glossary 2012), as well as Customer Relationship Management (CRM) and/or Enterprise Resource Planning (ERP) systems (Bramham et al. 2005; Elgh 2012;

Tab. 2: Identified modularization methods

Authors (Year)	Method	Origin	Description
Boucher et al. (2016)	PSS Modeling Method	P + S	The method was specifically created for designing and managing industrial PSS. It follows a five-step algorithm resulting in a structural (containing product, service and organization perspectives) and a dynamic meta-model (containing offer, scenario and performance perspectives). The practical implementation of the algorithm is supported with an open-access tool provided by the authors.
Browning (2001); Corsten and Salewski (2013); Steward (1981)	Design Structure Matrix (DSM)	P	A DSM shows the interdependencies between (standardized) elements, from which a certain product or service is assembled. These relations are measured based on their strength and recorded in a (N * N) matrix.
Böttcher and Klingner (2011); Klingner and Becker (2014)	Service-Metamodel, Configuration graph	S	The Service-Metamodel describes the structure of service modules in order to support the modeling and configuration of services with the help of IT. Different connectors, as well as logical and temporal dependencies are introduced. The resulting configuration graph gives an overview over all available modules and can be used by the customer to configure his individualized service.
Buchmann (2016)	ComVantage method	P + S	The ComVantage method addresses the domain of PSS in an Internet of Things environment. In order to separate concerns and achieve modularity, the modeling language is partitioned in model types. The designed model types are organized in a stack across four different vertical enterprise “facets” capturing different views of the system.
Dörbecker et al. (2014)	Multiple Domain Matrix (MDM)	P	MDM is an extension of DSM, which integrates different domains together. It consists of an arbitrary amount of DSMs (describing elements of a single domain, so N * N) and Domain Mapping Matrices (DMM, describing elements of two different domains, so N *M).
Erixon (1996)	Modular Function Deployment (MFD)	P	The MFD method groups individual product elements into modules based on their functionality. The complex product is firstly decomposed and subsequently evaluated based on a set of pre-defined criteria. In order to apply MFD in the context of services, additional evaluation criteria have to be defined.

Authors (Year)	Method	Origin	Description
Geum et al. (2012)	Modified House of Quality	S	This method is an extension of the MFD by combining it with the idea of the House of Quality. Customer needs are transformed into functional requirements and evaluated with appropriate (product and service) criteria.
Ho et al. (2009)	Interaction graph	S	The main idea of the method is that service elements interact with each other via frequent service invocations. Modules are then created by minimizing the interaction between modules (Low Coupling) and maximizing the interactions between elements within modules (High Cohesion).
Hölttä et al. (2003)	Modularizing using Dendograms	P	Each product undergoes a functional decomposition, assigning inputs and outputs to its every component and labeling them as black boxes. In order to identify similar modules, the method compares inputs and outputs of every black box. The method can be applied for services if appropriate comparison measurements are defined.
Li et al. (2012)	Module Partition Process	P + S	Based on the customer needs, relevant products and services are modularized independently and are later combined into an Integrated-Service-Product (ISP). This method assumes an inseparability of services and products and thus the need of their simultaneous modularization.
Lin and Pekkarinen (2011)	Quality Function Deployment with House of Quality	S	The method is based on the publication of Lin et al. (2010), which divides industrial service provision into three layers – service, process, and activity layer. The modularization process is then conducted on all three layers in parallel using the House of Quality.
Peters and Leimeister (2013)	TM3	S	TM3 was specifically designed for telemedical services and presents a general procedure with five phases, which covers most of the modularization process. This is the only method that pays additional attention to the interfaces between individual phases of the process.
Song et al. (2015)	Modularizing Product Extension Services	P + S	The authors propose a method, which is specifically designed for product-extension services (PES). At first, the PES-blueprint is used to represent the whole PES scenario and identify all relevant service components. Afterwards, a module partition of the service components is executed using fuzzy graph theory.

Authors (Year)	Method	Origin	Description
Stone et al. (2000)	Function Structure Heuristic	P	The method identifies modules based on the relevant flows (e. g., energy flow, material flow). Functions that are connected by the same flows can be grouped into independent modules using three proposed heuristics.
Wang et al. (2011)	Modular Product Service Systems	P + S	The method is based on the concept of PSS, which is defined as the combination of tangible artefacts and intangible services. Using a functional modularization of the customer's requirements as a starting point, the method simultaneously performs product and service modularization to achieve a modular PSS.
Yu et al. (2008)	Product-related Industrial Service Design	S	The paper introduces a systematic design method, specifically developed for industrial services. The customer is supposed to be involved in the service module design from the early stage, thereby influencing the final service package configuration possibilities.

Hvam et al. 2006). Within the quotation process, they typically offer the following functions:

- *Configuration*: Based on the customer needs, a specification of the service or product is configured using available service modules. This can be done by a sales clerk relying on her/his discussions with the customer about her/his needs. The feasibility of the configuration has to be assessed and ensured; this may also lead to discussions of modifications with internal experts. In case of online configurators, the user can define her/his configuration through a web-based frontend as a self-service.
- *Pricing*: The pricing engine of the software calculates a price based on the configured specification of the service or product and additional variables. Depending on the industry and market environment, the pricing strategies to be implemented by the pricing engine can vary from rather simple cost-based and linear strategies, where the price of the configuration can be determined from the prices set for single modules, to more complex competition- and demand-oriented pricing strategies.
- *Quotation*: A document with the configuration and the calculated price is generated which is ready for transmission and presentation to the customer. This document may include additional explanations, illustrations, alternative configurations and options, as well as appendices and disclaimers. The quotation sent to the customer will typically also be stored in the CPQ, CRM or ERP system or archived in a document management system.

In what follows, we will focus on the build-time of service modularization.

4.2 Objectives and Scope of Service Modularization

Both the transformation process and the result of the modularization initiative of the company (i. e., the configurable modular service architecture) are influenced by the desired scope and objectives of modularization that provide an overarching framing. *Modularization objectives* can be divided into two different types with according modularization

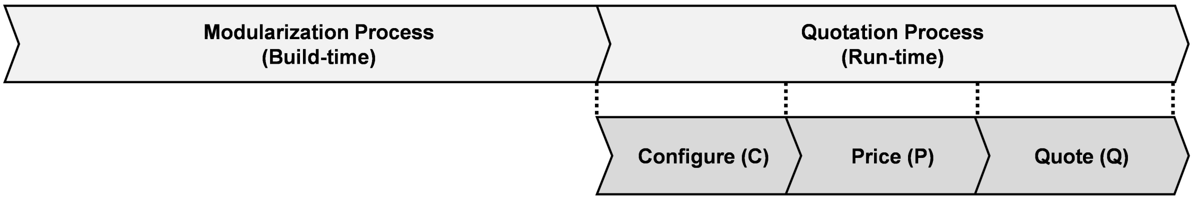

Fig. 2: Build-time and run-time of service modularity

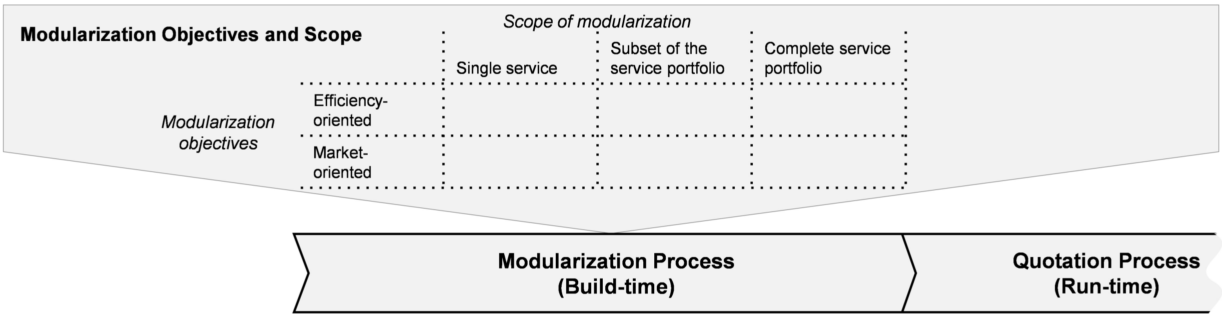

Fig. 3: Framing of the modularization initiative

strategies. The first type is the efficiency-driven modularization, which attempts to minimize the costs corresponding to multi-variant service offerings, e. g., through better resource utilization. This objective typically results in the reduction of the whole value-creation process of the company to a necessary amount of standardized service components or sub-processes, which can later be used in the customized service bundle. In other words, a company's motivation is to move from a "highly individualized" towards a "flexible, but standardized" service portfolio with the help of modularization. Alternatively, the aim of modularization can be a market-oriented variant management. A higher variant diversity will then be used for reaching out to yet unaddressed customer segments (Krebs and Ranze 2015), thus moving from "strictly standardized" towards a "flexible, but standardized" service portfolio.

Concerning the *scope of modularization*, a service provider has to decide which part of its service portfolio will be subject to the modularization initiative. On the one hand, there is the potential for enabling synergies that go beyond the boundaries of existing services, which calls for simultaneous modularization of multiple (if not all) services (Fig. 3). On the other hand, it may be desired and useful to concentrate the available resources on the most pressing and promising subareas of the portfolio first, thus increasing the overall success possibility of the modularization endeavor.

4.3 Framework for Systemizing Modularization Methods

4.3.1 Overview

The developed classification framework comprises two dimensions, which we were able to identify from our engagement with the literature and the 16 modularization methods from our sample. The first dimension reflects different stages of the modularization process which are to be completed within an initiative. The depicted process begins with the information capturing about the existing service portfolio of the service provider and ends with the test phase of the final set of modules including specified interfaces and rules for their configuration. The result of this process is a modular architecture which can be utilized by intended users (e. g., sales personnel or customers) through appropriate configuration and quotation processing tools (e. g., CRM plugins or online configurators).

The second dimension distinguishes between different types of how the modules are structured. *Logical structures* are widely used in the manufacturing industry, where the composition of a product from its components is described like in a bill of materials. On the other hand, the process nature of services suggests the *temporal structure* to be also or even more suitable for the service domain. In this case, due to the subsequent execution of process modules and accordingly sub-processes, there exist predecessor-successor-relationships. Furthermore, different logical structures, temporal structures, or both can be combined into *complex structures*. Fig. 4 gives an overview of the framework, which will be described in detail in the following.

4.3.2 Phases of the Modularization Process

The first dimension, the modularity process, consists of the six phases (1) information capturing, (2) decomposition, (3) structuring, (4) module creation, (5) interface definition, and (6) testing. In addition, the corresponding outputs of each phase are presented, which are, in turn, the inputs of the subsequent phase. The phases have their roots in the modularization method TM3 (Peters and Leimeister 2013) that was specifically designed for telemedical services and has been adjusted and refined in this publication based on the further identified modularization methods.[1]

The phase of *information capturing* gathers detailed information about the status quo of existing service offerings and service provision processes of the company and identifies customer requirements with respect to service variants. This can be achieved by using methods of qualitative research, e. g., interviews, observations, and document analysis (Peters and Leimeister 2013; Yu et al. 2008). The resulting outputs of this phase are documentations and service models, which can be recorded in forms of diagrams and text. For instance, existing service offerings are captured in a service catalog, giving an overview of possible service variants. Service provision processes can be recorded in process models that incorporate the temporal sequence of activities as well as their mapping to resources. Appropriate modeling notations include, e. g., Service Blueprinting (Wang et al. 2011), BPMN, or EPC. Such models provide an overview of the as-is situation, whereas customer needs may be transformed into *to-be* models, thus identifying the required service variety. In this context, service variants can refer to both different outcomes of services and varying service provision processes.

In the second phase termed *decomposition*, the gathered information is broken down into its elements (i. e., the most, but still meaningful, granular level), which will later be used to create independent modules. Services from the service portfolio are split into their components, e. g., a car inspection service can be divided into oil change, security check, and filter change. Similarly, processes can be divided into single activities (Lin and Pekkarinen 2011) and customer demands can be transformed into detailed functional requirements (Geum et al. 2012). The output of this phase is, therefore, a collection of unstructured atomic elements, which represents a company's service portfolio on a granular level. It is worth mentioning that each of the methods dealing with this phase defines its own level of granularity. Even though there exist several publications on the necessary service granularity level (Glöckner et al. 2016; Kim and Doh 2009), no consensus has been reached so far on what is an adequate level.

Once the elements have been identified and recorded, their categorization based on appropriate descriptive dimensions can begin in the phase *structuring*. A widely used way of structuring elements is with the help of matrices that can measure the elements' interdependencies either in a qualitative (e. g., "weak", "strong", "irrelevant") or quantitative (e. g., on a scale from 0 to 10)

---

[1] The original TM3 included five phases (1) status capturing, (2) decomposition, (3) matrix generation, (4) interface specification and (5) testing. The phase of module creation, which is explicitly mentioned in our framework, is a part of the matrix generation phase of TM3 (Peters and Leimeister 2013)

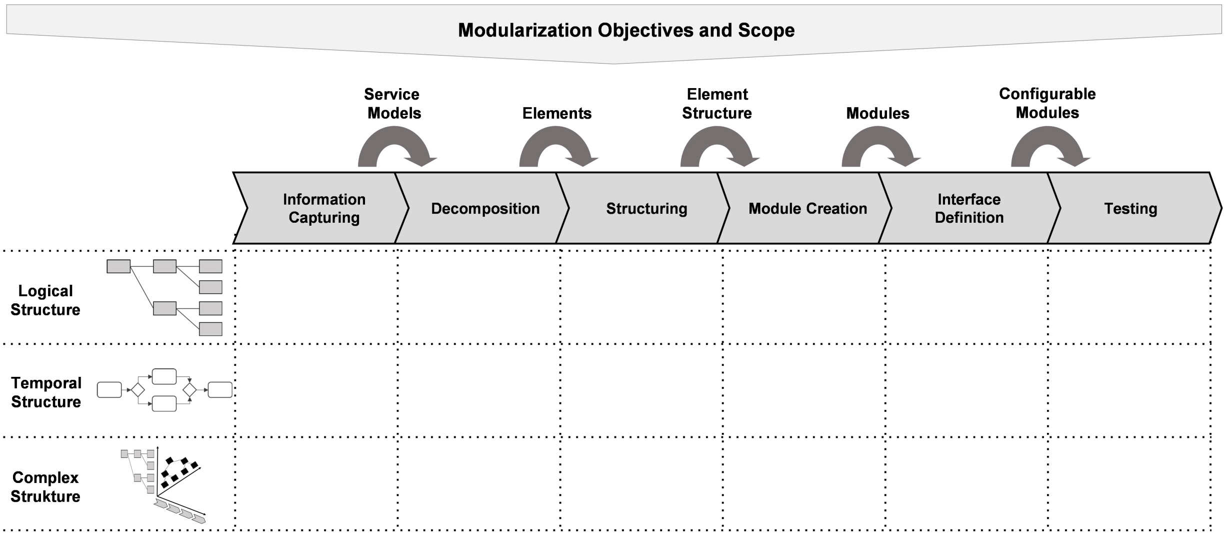

Fig. 4: Classification framework for service modularization methods

manner (Browning 2001; Corsten and Salewski 2013; Dörbecker and Böhmann 2013). Similarly, elements can be categorized in different clusters using predefined attributes, e. g., automated vs. manual activity or different levels of customer involvement (Erixon 1996; Stone et al. 2000). The output is an element structure, which is presented either in a form of a matrix, a classification, or a more complex structure if multiple attributes are used for element description. However, the service designer should keep in mind that with every additional descriptive dimension the complexity of the element structure is increased disproportionately, which may result in complexity problems in the next step of module creation (Dörbecker et al. 2014).

The phase of *module creation* marks the core step of the modularization process. Using the element structure from the previous phase, modules are built in a way that the elements within the modules are as homogenous as possible and are thus tightly interconnected (high cohesion), whereas the inter-modular relationship is minimized, enabling the principle of loose coupling (Ho et al. 2009). Modules are typically built using clustering algorithms (Hölttä et al. 2003; Song et al. 2015), heuristics (Stone et al. 2000), or via purely subjective aggregation, which is based on, e. g., expert knowledge (Yu et al. 2008) or card-sorting approaches with customer participation (Kohlborn and Poeppelbuss 2013). At this point, it is important to determine how many modules are desired at the end of the modularization initiative, so that the new architecture neither contains a too low number of excessively large and undifferentiated modules (or even one single module), nor ends up being too granular (an extreme form would be a 1:1 mapping of atomic elements to modules). Here, Dörbecker et al. (2014) offer a promising starting point to the discussion about an optimal amount of modules by introducing different scenarios based on the desired level of module granularity (not to be mixed up with the level of granularity of elements in the step of service decomposition; see above). Similarly, Erixon (1996) mentions a rule-of-thumb estimate that the number of modules should be approximately the square root of the number of elements, although no explanation for this rule is provided. The output of this phase is a set of modules, each consisting of at least one but usually multiple elements.

The *interface definition* phase determines how the set of created modules can be configured in order to offer the requested variety of services or service packages. This is a crucial step of the modularization process since this is where heterogeneous customer demands are linked with the service provider's need for standardization. In the course of the later configuration during the sales process at run-time, certain modules can enable or exclude each other (Böttcher and Klingner 2011; Erixon 1996), just as certain packages or module combinations can be tagged with special characteristics (e. g., flat rates or package prices). In addition, inter-modular interfaces have to be defined, which enable the interaction of modules and their combination in the first place (see also phase 4 "interface specification" of the TM3-method (Peters and Leimeister 2013)). It is of particular importance to ensure that the outputs of one module are compatible with the expected inputs of the consecutive or adjacent module (Wang et al. 2011) to ensure an appropriate information transmission and desired functionality. The result of this phase is, therefore, a modular set of configurable modules with pre-defined interfaces and configuration rules.

The final phase of the modularization process deals with testing the created set of configurable modules. At this point, it is important to examine whether the use of modules, respecting the defined interfaces and configuration rules, delivers valid and reasonable results while simultaneously excluding undesired combinations. Moreover, the applicability of the newly designed modular architecture needs to be evaluated in the specific context or usage scenario (Boucher et al. 2016). Finally, also the overall operational efficacy and efficiency at run-time needs to be assessed. This can be achieved e. g., by comparing the time spent for preparations of quotes and tenders, or the resources required for service delivery against the previous status quo as documented in the initial phase.

The introduced six phases represent an idealistic modularization process, which was derived from the 16 identified methods (Tab. 2). Considering actual modularization initiatives in practice, one can expect that, dependent on a company's context, some phases will be excluded or, on the contrary, executed in iterative loops. Similarly, starting with advanced phases or an early stop are also possible. This will probably be the case in a company with already a granular and well-documented service portfolio in place that just needs to be presented to their customers in an easy and intuitive manner. Moreover, it is worth mentioning that the six phases are also interdependent as visualized in Fig. 4. For instance, the element structure, which is the output of the structuring phase, will be derived from the information gathered in the phase of information capturing. Similarly, in case a specific type of structuring is pre-defined in the modularization objectives, this results in corresponding presets for the information capturing phase.

4.3.3 Types of Structuring

The second dimension distinguishes between different types of structuring the elements and modules. Based on the common distinction between product and process modularization, one can differentiate between a logical and a temporal kind of structure. This distinction is also presented in Böttcher and Klingner (2011), who write about logical and temporal interdependencies between service elements. As a third kind, complex structures result from a combination of several and potentially different approaches to structuring (Fig. 5).

As with bill of materials in manufacturing, *logical structures* can be represented as trees or graphs. Similarly, matrices that show the relationships between service elements or organizational units and corresponding resources with the help of certain evaluation criteria (e. g., coordination cost between activities, as shown in Corsten and Salewski (2013)) can also be considered logical structures.

Temporal structures, on the other hand, are used to define chronological sequences of activities or events, which is a typical approach for describing services. In other words, the underlying principle is the definition of predecessor-successor-relationships (Wang et al. 2011). Visualization tools for such structures include flow charts, Petri

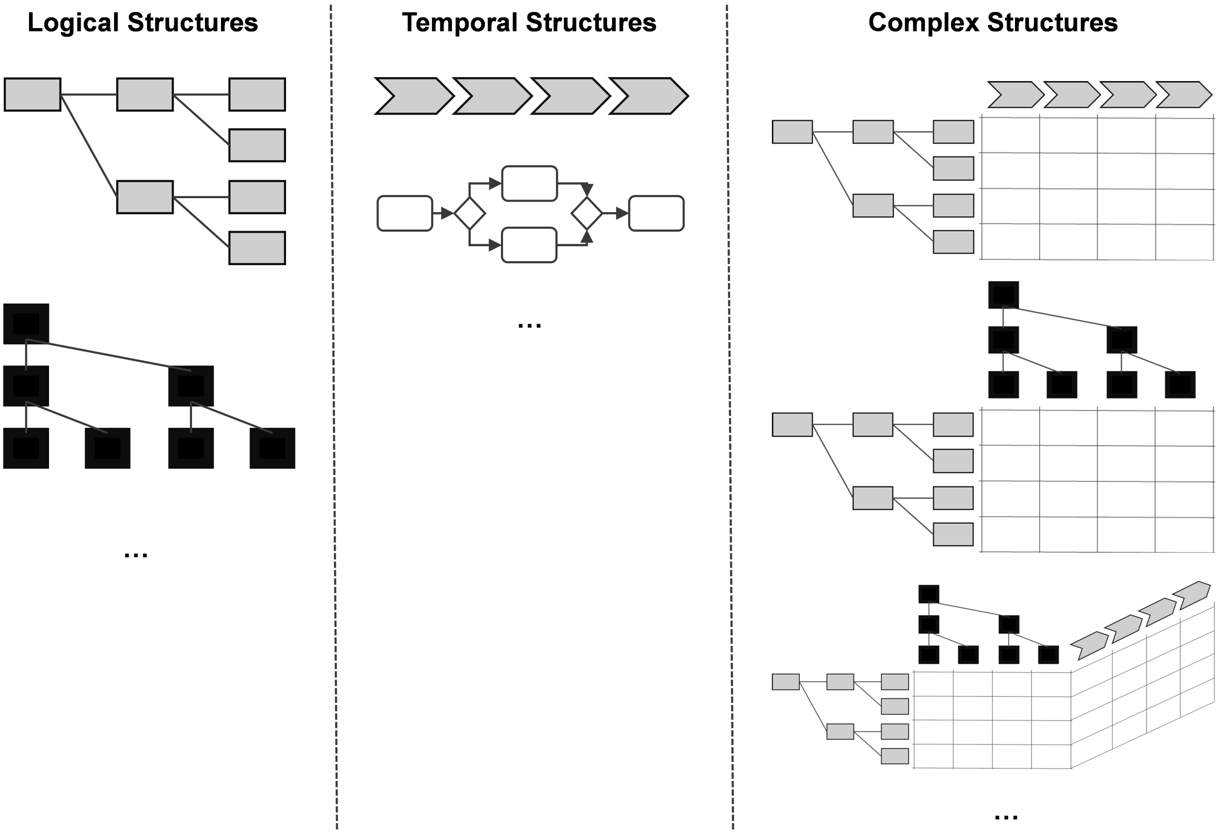

Fig. 5: Different structuring forms.

nets or specific modeling languages like EPC or BPMN as well as their adjustments to a certain domain context, as presented in Peters and Leimeister (2013).

Different logical and temporal structures can also be combined to form *complex structure*. TThis enables a multi-dimensional analysis of the elements. One of such combinations was introduced by Böttcher and Klingner (2011), who enrich their tree structure of service elements with detailed logical and temporal dependencies. Similarly, Dörbecker et al. (2014) combine resource and process dimensions with the help of the Multiple Domain Matrix (MDM) and differentiate between various target groups.

5 Classification of Existing Methods

The developed framework can be used to classify existing modularization methods in order to show which phases of the modularization process they cover and what kind of structuring they are based on. Fig. 6 illustrates the classification of the identified 16 methods, which were originally used to synthesize the framework. As for the modularity objectives and scope, the present methods cannot be categorized properly as they typically do not provide any specific information about the actual implementation, but instead assume general applicability.

5.1 Classification based on Modularization Phases

When looking at the modularization process, only five methods pay explicit attention (or at least give suggestions with regard to its execution) to *information capturing*, while the others act on the assumption that this will have already been accomplished by the company in an appropriate

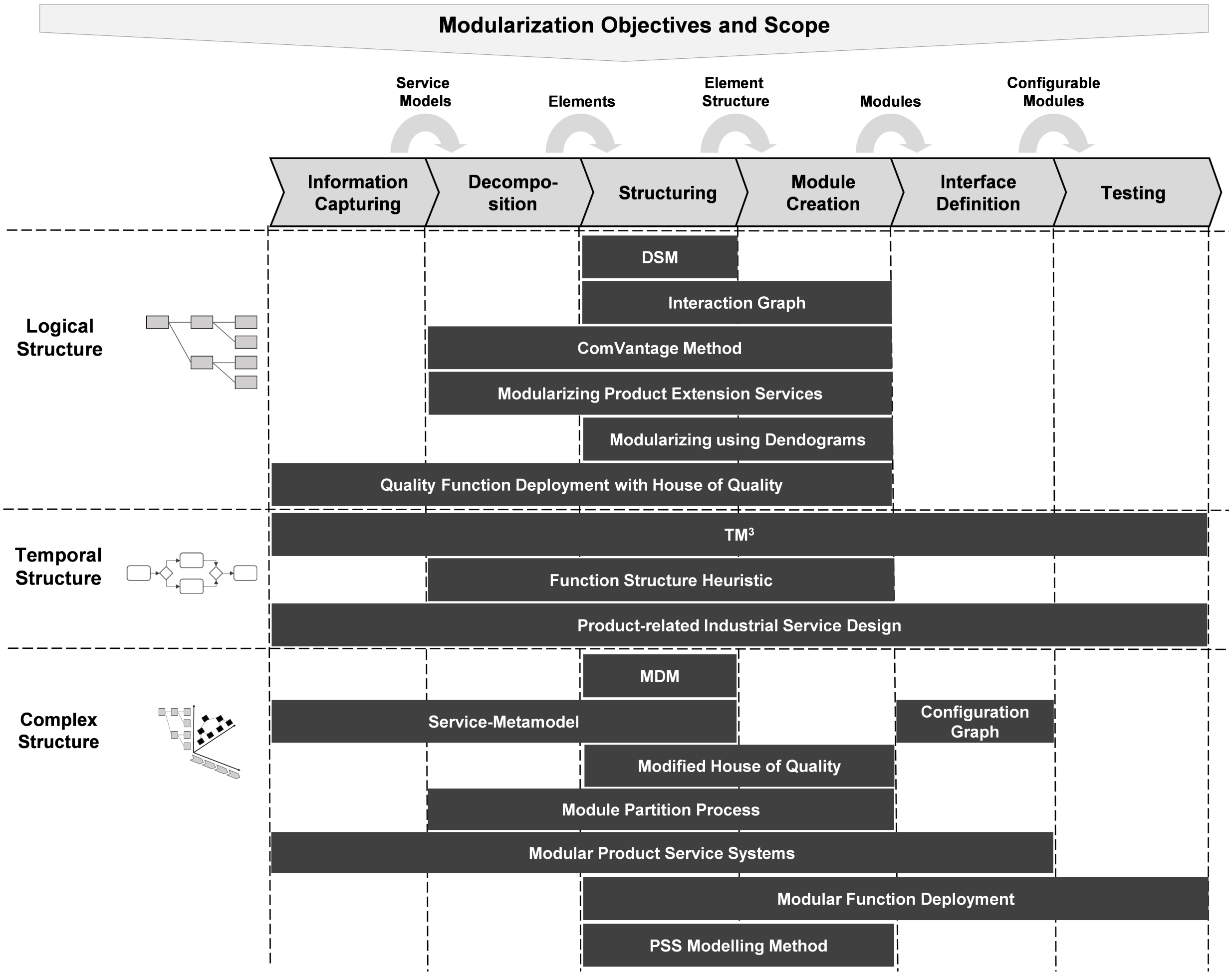

Fig. 6: Classification of methods for service modularization.

way. With regard to the kind of information that is captured, the Service-Metamodel and the TM3 focus on the process flow of the company's service provision (Böttcher and Klingner 2011; Peters and Leimeister 2013). In contrast, the Quality Function Deployment method sees the estimation of customer demands with the help of questionnaires and expert interviews as a starting point of the modularization process (Lin and Pekkarinen 2011). In addition, the remaining Product-related Industrial Service Design (Yu et al. 2008) and Modular Product Service Systems (Wang et al. 2011) not only analyze the customer requirements, but combine them with relevant information about the market and competition.

Nine out of 16 methods support the *decomposition* of previously monolithic services into their constituent elements. For instance, Klingner and Becker (2014) propose an approach based on their Service-Metamodel and using Workflow-Patterns (van Der Aalst et al. 2003) for decomposing monolithic services (that are documented with the help of process models) into smaller, functionally differentiated elements. Similarly, Song et al. (2015) introduce a modified service blueprint, which is used to represent the whole PES scenario and identify all relevant service components. In addition to the previous methods, the decomposition of products and services is also supported by the

Functional Structure Heuristics (Stone et al. 2000) and the Module Partition Process (Lin et al. 2010).

All of the presented methods cover the structuring phase, meaning that they are able to analyze the relationships between the identified elements. The proposed techniques range from the classic Design Structure Matrix (DSM) that can be used to capture the coordination costs between different activities (Corsten and Salewski 2013) up to a logical and hierarchical structuring of elements with the help of different abstraction layers like service modules, process modules, and activity modules (Lin and Pekkarinen 2011). Furthermore, methods that follow the idea of the inseparability and interdependence of services and products underline the need of structuring the information accordingly. Examples include the Modular Product Service Systems method (Wang et al. 2011) and Modular Partition Process (Li et al. 2012). The latter introduces the notions of functional and non-functional services, based on their dependence on physical modules.

The actual *module creation* can be achieved using either quantitative or qualitative methods. Methods like DSM and MDM quantify the relationships between elements in a way that enables an algorithmic module creation in the next step, e. g., with the help of clustering algorithms (Hölttä et al. 2003). The remaining qualitative methods likewise give indications on how to create modules from the available element structure. For instance, TM3 suggests criteria for aggregating elements, e. g., the need of particular premises, equipment, knowledge, or customer involvement during activity execution. Similarly, the Modular Function Deployment provides a rule-of-thumb estimate of how many modules shall be created from a certain amount of elements. The modules are then built from the identified elements using predefined evaluation criteria (so-called “module drivers” (Erixon 1996)). The idea of pre-defined evaluation criteria is extended in the Modified House of Quality method, where both product and service-specific module drivers are used (Geum et al. 2012).

The topic of *interface definition* is specifically addressed by the Configuration Graph (Böttcher and Klingner 2011). It offers a diagram-based notation of logical and temporal interdependencies between modules, which is then used for their combination to services. The Configuration Graph is based on the logic of the previously mentioned Service-Metamodel (Klingner and Becker 2014) and is therefore presented in the same row in the framework (Fig. 6). The phase of *interface definition* is also briefly mentioned in the methods TM3, Modular Product Service Systems and Product-related Industrial Service Design (Peters and Leimeister 2013; Song et al. 2015; Yu et al. 2008). Finally, being developed explicitly for product modularization, Modular Function Deployment differentiates between fixed, moving, and transmitting interfaces, which makes it necessary to introduce a service-specific classification of interfaces.

Finally, only three methods – TM3, Product-related Industrial Service Design and Modular Function Deployment – point to the necessity of *testing* the identified modules, their interfaces, and the new modular architecture as a whole. However, neither any of these methods provide specific test procedures, nor do they make any suggestions concerning the transition from the modularization process (build-time) to quotation processing (run-time). They also remain silent about the use of information systems to support the use of the modular architecture in day-to-day sales processes (e. g., online configurators).

From this classification it becomes obvious that only two methods cover all phases of the modularization process, which are TM3 (Peters and Leimeister 2013) and Product-related Industrial Service Design (Yu et al. 2008). These methods, however, give abstract suggestions on what aspects to keep in mind in each of the phases, rather than providing precise guidelines on their execution.

5.2 Classification based on Types of Structuring

The presented methods follow different types of structuring, which affect both inputs and outputs of each of the modularization phases. As mentioned before, the structuring framework differentiates between logical, temporal, and complex structures.

A typical example of a method based on a *logical structure* is the well-known DSM, in which service elements are mapped to each other in a matrix. Their relationships are assessed based on specific dimensions or criteria, e. g., coordination costs (Corsten and Salewski 2013). Other methods following a logical structure include Modularizing using Dendograms (Hölttä et al. 2003) that clusters elements into modules based on their input and output characteristics, as well as the Interaction Graph (Ho et al. 2009), which is constructed based on the number of invocations between elements. Finally, in the Modularizing Product Extension Services method (Song et al. 2015), different types of component interdependencies are combined in a comprehensive matrix, which is later visualized with the help of a maximum spanning tree and divided into modules according to a desired threshold.

TM3 (Peters and Leimeister 2013) and Product-related Industrial Service Design (Yu et al. 2008), on the other hand, can be seen as representatives of methods following a *temporal structure*. These methods analyze the chronological order of activities within service processes and visualize them as predecessor-successor-relationships. Moreover, the Function Structure Heuristic that was specifically designed for products also takes a process flow perspective.

The most widespread kind of structuring are *complex structures*, which combine different descriptive dimensions to group elements into modules. For instance, the MDM provided by Dörbecker et al. (2014) links process and resource domains together. Following the idea of the inseparability of services and products, both the Modular Partition Process and the Modular Product Service Systems method integrate service, functional, and product modularization simultaneously. Similarly, the Configuration Graph presents both logical and temporal interdependencies between elements or modules, respectively (Böttcher and Klingner 2011). Finally, the Modular Function Deployment method provides twelve different module drivers to evaluate the elements when creating modules (Erixon 1996).

6 Conclusion and Outlook

6.1 Implications and Limitations

Service modularity has attracted much attention from service researchers during the past decade and is becoming an interesting concept for practitioners to balance efficiency and variety of services. However, it has mainly stayed an academic concept so far with little application in practice. We believe that one main reason for this lies in a lack of transparency and understanding of the overall modularization process, its phases, and related activities. Obviously, there is a growing set of modularization methods, especially proposed from academia, which attempt to provide solutions for fine-grained tasks. At the same time, there has been a lack of the bigger picture of service modularization and practitioners find it difficult to assess the suitability of existing modularization methods to their actual challenges in practice. In order to address this research gap, we developed a comprehensive framework for classifying service modularization methods. This framework can be used for structuring existing and future work on this topic and serve as an implementation guideline of modularization initiatives for service providers.

We consider the framework to be useful for practitioners while it surely cannot offer a comprehensive handbook on how to modularize service portfolios. First, it is particularly valuable as it points to a set of basic decisions that have to be made at the beginning of modularization initiatives, i. e., defining the objectives and the scope. While these decisions do not necessarily determine the use of specific methods, we consider such decisions crucial in order to be able to measure the success of modularization initiatives in the end. This is also a novel contribution of our framework as these basic decisions have not been discussed in the existing method descriptions at all. Second, the framework discloses the complexity of modularization initiatives, which goes beyond the application of clustering algorithms and heuristics. It also shows in how far existing methods cover certain phases of modularization process. Based on this information, practitioners can choose which methods they may want to use in combination or deliberately opt for more comprehensive methods. Third, the distinction between different types of structuring also offers practitioners a starting point for checking the potential compatibility of different methods across a modularization process. While being classified to the same type of structuring does not guarantee unrestricted compatibility of methods, it at least discloses a consistent view of what is basically understood as the general format of the resulting modular architecture.

The presented research is beset with limitations. The developed framework proposes two key dimensions for the classification of modularization methods that are based on a detailed analysis of existing works. When defining these two key dimensions, we particularly built on the phases found in the TM3 method (Peters and Leimeister 2013) and the different types of element interdependencies suggested by the Service-Metamodel (Böttcher and Klingner 2011) However, we find the delimitation between logical and temporal structures to be sometimes difficult, as even processes, which are firstly modeled using a temporal orientation, are often reduced to logical hierarchies of processes and sub-processes in the end (Corsten and Salewski 2013; Lin and Pekkarinen 2011). Hence, there might be room for further improving the framework by additional research in the future, especially in terms of additional or refined classification dimensions.

6.2 Avenues for Future Research

This study further points to several opportunities for future research that will be of value to both research and practice. First, there is a need for improving existing methods and possibly also developing new service modularization methods. Obviously, the structuring phase has been addressed the most so far, even though many researchers point to the need of analyzing preceding and successive phases, too. Hence, more attention has to be paid towards the less covered modularization phases like information capturing, interface definition, or testing, e. g., by providing new methods or extending existing methods accordingly. When advancing or developing methods, the practical usefulness should be a main objective as existing methods have not found widespread adoption in practice yet.

Second, more research is also needed to empirically investigate modularization initiatives in real-life organizations. Especially case studies and action research projects are needed to test the actual applicability and effectiveness of existing modularization methods. As for now, too little is known about whether and how they can be successfully implemented by service providers. Such investigations will also support the identification of key factors that support or inhibit the application of the modularity concept to services and will further provide insights, which domains and what types of services are most amenable for this concept. This could, for instance, result in a set of evaluation criteria that help predicting whether and in what scope a company could benefit from modularization. Here, the typology of service process types by Carlborg and Kindström (2014) offers a promising starting point, although the modular strategies they describe might still be too vague to actually guide measures for implementing the modularity concept in organizations. Another opportunity is the design of a service modularization maturity model that would help organizations to carry out as-is assessments and define a roadmap towards a modular service architecture.

Third, more critical reflection is needed in how far the different characteristics of products and services actually require distinctive modularization methods or not. While there are methods that originate from product design, we also identified methods specifically developed for services. Some recent methods do not even distinguish or deliberately integrate both product and service elements. Here, it could for example be worthwhile to discuss the implications of the Service-Dominant Logic (Vargo and Lusch 2004) and the co-creation of value between providers and customers on service and product modularity.

Finally, this study can also spark further design-oriented work beyond the development of methods and constructs for defining modular architectures. There is an obvious need for software suites that support both the built-time and run-time of service modularization comprehensively and in an integrated manner. Only a few of the analyzed publications have reported on dedicated software prototypes for modeling and analyzing modular service structures at all. At the same time, initiatives in practice are typically limited by the data processing capabilities of spreadsheets and standard ERP systems. While there are already specialized CPQ software suites, they mostly focus on product and not service domains. It can be expected that a widespread adoption of service modularization methods in practice, in the end, will also be dependent on the availability of adequate and easy-to-use software support.

References

Arnheiter E., Harren H. (2005) A typology to unleash the potential of modularity. In: Journal of Manufacturing and Technology Management 16, pp. 699–711

Baldwin C., Clark H. (1999) Design Rules, Volume 1: The Power of Modularity. MIT Press, Cambridge, MA, USA

Bask A., Lipponen M., Rajahonka M., Tinnilä M. (2011a) Framework for modularity and customization: service perspective. In: Journal of Business and Industrial Marketing 26, pp. 306–319

Bask A., Lipponen M., Rajahonka M., Tinnilä M. (2011b) Modularity in logistics services: a business model and process view. In: International Journal of Services and Operations Management 10, pp. 379–399

de Blok C., Meijboom B., Luijkx K., Schols J., Schroeder R. (2014) Interfaces in service modularity: a typology developed in modular health care provision. In: Journal of Operations Management 32(4), pp. 175–189

Boell S., Cecez-Kecmanovic D. (2014) A hermeneutic approach for conducting literature reviews and literature searches. In: Communications of the Association for Information Systems 34(10), pp. 379–399

Böhmann T., Loser K. (2005) Towards a service agility assessment - modeling the composition and coupling of modular business services. In: Seventh IEEE International Conference on E-Commerce Technology Workshops, pp. 140–148

Böttcher M., Klingner S. (2011) Providing a method for composing modular B2B services. In: Journal of Business & Industrial Marketing 26 (5), pp. 320–331

Boucher X., Medini K., Fill H.-G. (2016) Product-Service-System Modeling Method. In: Domain-Specific Conceptual Modeling. Springer, pp. 455–482

Bramham J., MacCarthy B., Guinery J. (2005) Managing product variety in quotation processes. In: Journal of Manufacturing Technology Management 16, pp. 411–431

Brax S. A., Bask A., Hsuan J., Voss C. (2017) Service modularity and architecture–an overview and research agenda. In: International Journal of Operations & Production Management 37(6), pp. 686–702

Browning T. R. (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new direction. In: IEEE Transactions on Engineering Management, pp. 292–306

Carlborg P., Kindström D. (2014) Service process modularization and modular strategies. In: Journal of Business and Industrial Marketing 29, pp. 313–323

Corsten H., Salewski H. (2013) Dienstleistungsmodularisierung: Entwurf zweier alternativer Modellierungsvorschläge. In: Kompetenz, Interdisziplinarität und Komplexität in der Betriebswirtschaftslehre, pp. 313–323

de Blok C., Luijkx K., Meijboom B., Schols J. (2010) Modular care and service packages for independently living elderly. In: International Journal of Operations and Production Management 30, pp. 75–97

Demirkan H., Kauffman R. J., Vayghan J. A., Fill H.-G., Karagiannis D., Maglio P. P. (2009) Service-oriented technology and management: Perspectives on research and practice for the coming decade. In: Electronic commerce research and applications 7(4), pp. 356–376

Dörbecker R., Böhmann T. (2013) The Concept and Effects of Service Modularity – A Literature Review. In: Proceedings of the 46th Hawaii International Conference on Information Systems (HICSS 2013), pp. 1357–1366

Dörbecker R., Tokar O., Heddaeus D., Böhmann T. (2014) Evaluation der Multiple Domain Matrix Methode zur Modularisierung von Dienstleistungen am Beispiel eines Versorgungsnetzwerks für psychische Erkrankungen. In: Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2014). Paderborn, Germany

Duckwitz S., Petz A., Schlick C. M. (2015) Modelling and Simulation of Knowledge-Intensive Service Systems with Design Structure Matrices. In: Enterprise Modelling and Information Systems Architectures 10(1), pp. 49–66

Edvardsson B., Enquist B. (2007) Success factors in new service development and value creation through services. In: Advances in Service Innovations, pp. 165–183

Elgh T. (2012) Decision support in the quotation process of engineered-to-order products. In: Advanced Engineering Informatics 26(1), pp. 66–79

Erixon G. (1996) Modular function development mfd, support for good product structure creation. In: Proceedings of the 2nd WDK Workshop on Product Structuring 1996

Gartner IT Glossary (2012) Configure, Price, Quote (CPQ) Application Suites http://www.gartner.com/it-glossary/configure-price-quote-cpq-application-suites/

Geum Y., Kwak R., Park Y. (2012) Modularizing services: A modified HoQ approach. In: Computers and Industrial Engineering 62, pp. 579–590

Glöckner M., Ludwig A., Franczyk B. (2016) How low should you go? - conceptualization of the service granularity framework. In: Proceedings of the 24th European Conference on Information Systems (ECIS 2016). Paper 32. Istanbul, Turkey

Ho H., Haung C., Yang H. (2009) Development of Modular Services. In: Proceedings of the New Trends in Information and Service Science, pp. 1215–1220

Hölttä K., Tang V., Seering W. (2003) Modularizing product architectures using dendrograms. In: Proceedings of ICED 03, the 14th International Conference on Engineering Design

Hvam L., Pape S., Nielsen M. (2006) Improving the quotation process with product configuration. In: Computers in Industry 57(7), pp. 607–621

Iman N. (2016) Modularity matters: a critical review and synthesis of service modularity. In: International Journal of Quality and Service Sciences 8(1), pp. 38–52

Kim Y., Doh K. (2009) Formal identification of right-grained services for service-oriented modeling. In: International Conference on Web Information Systems Engineering, pp. 261–273

Klingner S., Becker D. (2014) Konfiguration von Dienstleistungen – Ein Ansatz zur Verbindung von Geschäftsprozessen mit Dienstleistungsmodellen. In: Dienstleistungsmodellierung, pp. 76–96

Kohlborn T., Poeppelbuss J. (2013) Designing the Information Architecture of Governmental One-Stop Portals: On the Application and Analysis of Card Sorting. In: International Journal of Electronic Government Research, pp. 47–62

Krebs T., Ranze C. (2015) Market-oriented Variant Management. In: Proceedings of the 17th International Configuration Workshop. Vienna, Austria, pp. 1–4

Leseure M., Hudson-Smith M., Bask A., Lipponen M., Rajahonka M., Tinnilä M. (2010) The concept of modularity: diffusion from manufacturing to service production. In: Journal of Manufacturing Technology Management 21(3), pp. 355–375

Li H., Ji Y., Gu X., Qi G., Tang R. (2012) Module partition process model and method of integrated service product. In: Computers in Industry 63, pp. 298–308

Lin Y., Luo J., Zhou L. (2010) Modular logistics service platform. In: Service Operations and Logistics and Informatics, pp. 200–204

Lin Y., Pekkarinen S. (2011) QFD-based modular logistics service design. In: Journal of Business and Industrial Marketing, pp. 344–356

Lubarski A., Poeppelbuss J. (2016) Modularization of Logistics Services – An Investigation of the Status Quo. In: Proceedings of the 5th International Conference on Dynamics in Logistics

Müller F., Lubarski A. (2016) Schools of Thought in Service Modularity. In: Proceedings of the 24th European Conference on Information Systems. Istanbul, Turkey

Pekkarinen S., Ulkuniemi P. (2008) Modularity in developing business services by platform approach. In: International Journal of Logistic Management 19, pp. 84–103

Peters C., Leimeister J. (2013) TM3 - A Modularization Method for Telemedical Services: Design and Evaluation In: Proceedings of 21st European Conference on Information Systems (ECIS 2013) Utrecht, The Netherlands

Poeppelbuss J., Lubarski A. (2016) Methoden der Dienstleistungsmodularisierung – Entwurf eines Ordnungsrahmens zur Systematisierung. In: Thomas O., Nüttgens M., Fellmann M. (eds.) Dienstleistungsmodellierung (DLM 2016). Springer, Berlin, Heidelberg, pp. 76–93

Rahikka E., Ulkuniemi P., Pekkarinen S. (2011) Developing the value perception of the business customer through service modularity. In: Journal of Business and Industrial Marketing 26, pp. 357–367

Song W., Wu Z., Li X., Xu Z. (2015) Modularizing product extension services: An approach based on modified service blueprint and fuzzy graph. In: Computers and Industrial Engineering 85, pp. 186–195

Stone R., Wood K., Crawford R. (2000) A heuristic method for identifying modules for product architectures. In: Design Studies 21, pp. 5–31

Tuunanen T., Bask A., Merisalo-Rantanen H. (2012) Typology for Modular Service Design: Review of Literature. In: International Journal of Service Science, Management, Engineering, and Technology 3, pp. 99–112

Tuunanen T., Cassab H. (2011) Service process modularization: reuse versus variation in service extensions. In: Journal of Service Research 14(3), pp. 340–354

van Der Aalst W., Ter Hofstede A., Kiepuszewski B., Barros A. (2003) Workflow patterns. In: Distributed and Parallel Databases 14, pp. 5–51

Vargo S., Lusch R. (2004) Evolving to a New Dominant Logic for Marketing. In: Journal of Marketing 68, pp. 1–17

Voss C. A., Hsuan J. (2009) Service architecture and modularity. In: Decision Sciences 40(3), pp. 541–569

Wang P., Ming X., Li D., Kong F., Wang L., Wu Z. (2011) Modular Development of Product Service Systems. In: Concurrent Engineering 19, pp. 85–96

Yu M., Zhang W., Meier H. (2008) Modularization based design for innovative product-related industrial service. In: Service Operations and Logistics, and Informatics, pp. 48–53