



A concept map to support the planning and evaluation of artifacts in the initial phases of PSS design

Maiara Rosa¹ · Wei Min Wang² · Rainer Stark² · Henrique Rozenfeld¹

Received: 28 May 2020 / Revised: 14 January 2021 / Accepted: 19 January 2021 / Published online: 13 February 2021
© The Author(s) 2021

Abstract

Designing Product-Service Systems (PSS) is associated with multiple problems and challenges, usually derived from its multidisciplinary and partially intangible nature. One particular issue is the high likelihood of ignoring the creation of relevant information regarding one or more PSS elements during the early design phases. Proceeding to later stages (e.g., detailed design) without generating the required information regarding all PSS elements and their relationships may lead to rework and lack of integration. Dealing with this problem requires adequate planning and evaluation of the artifacts (such as documents and models) created in the initial design phases. As a fundamental theoretical basis to support the creation of solutions that may help project managers dealing with this challenge, this paper presents a concept map to structure the concepts that compose artifacts resulting from the initial stages of PSS design and how those concepts interrelate. This concept map aims to structure which classes of information should be defined in the early phases of the design process before proceeding to a detailed design. The concept map was created by extracting concepts and relationships proposed in classifications, taxonomies, ontologies, meta-models, and concept maps in the PSS and servitization fields. Those documents were identified through a comprehensive systematic literature review. The resulting concept map was verified for completeness against formal documentation of two retrospective PSS design projects. The final proposed concept map is composed of 143 concepts interconnected through 278 relationships. In its current format, the concept map may be used as a checklist to support project managers in planning and evaluating early phases of PSS design based on information completeness. Researchers may also employ it to deploy ontologies, approach further knowledge and information-related challenges in PSS design, or structure PSS-related model-based systems engineering approaches. In future research, this concept map shall be deployed in a meta-model based on artifacts commonly used in PSS design, structuring a computational tool to allow and support practical application on planning and evaluating PSS design projects.

Keywords Concept map · Ontology · PSS design · Knowledge management · Planning · Evaluation

1 Introduction

Context A recent worldwide survey with about 190.000 manufacturers pointed out that 38% of them offer not only products but also services to their customers (Mastrogiacomo et al. 2019). Those numbers illustrate the servitization trend, which has been noticed many years ago and is still a theme. In this trend, companies are increasingly transforming their

business models from pure product offerings towards combining products and services or even selling products as services (Vandermerwe and Rada 1988; Neely 2008; Yang and Evans 2019). The term “servitization” refers to the company’s transformation process, which embodies designing new solutions composed of combined products and services, i.e., Product-Service Systems (PSS) (Baines et al. 2007, 2009; Vasantha et al. 2012, 2015; Raddats et al. 2019).

Problem and related challenges The PSS design is associated with a wide range of challenges, which usually originate from the nature of its design objects (i.e., the many tangible and intangible objects that compose the final solution) and its multidisciplinary. This paper deals with four challenges leading to a common problem: a high likelihood of ignoring the creation of relevant information on the early

✉ Maiara Rosa
maiara.rosa@usp.br

¹ University of São Paulo, Trabalhador são-carlense Ave, 400, São Carlos 13566-590, SP, Brazil

² Technical University Berlin, Pascalstr. 8-9, Berlin 10587, Germany

phases of PSS design, which generates rework and lack of integration (Aurich et al. 2006; Durugbo et al. 2010; Shen et al. 2010). Those four challenges are described in the following paragraphs, providing solutions proposed in the literature to deal with them.

First, services compose a meaningful part of PSS solutions. A particular characteristic of services is their intangible and heterogeneous nature (de Brentani 1991; Johne and Storey 1998; Jong and Vermeulen 2003; Alonso-Rasgado and Thompson 2006; Løkkegaard et al. 2016). This nature impacts the quality of service-related information generated during the PSS design, which often lacks completeness and structure (Shen et al. 2010). Employing process models is a good practice for structuring what information should be created in each moment of the design process, reducing the likelihood of forgetting relevant information (Browning et al. 2006; Gericke and Blessing 2011). There are several process models for PSS design available in the literature, such as the ones proposed by Hein et al. (2018), Alonso-Rasgado and Thompson (2006), Nguyen et al. (2014), Sutanto et al. (2015), among many others. However, formalizing the initial phases of design may hinder innovation, which is even more intense when market and technology uncertainties are high (Poskela and Martinsuo 2009). Therefore, for innovative solutions, employing formalized process models is more helpful in later stages of design (such as detailed design and implementation), which will benefit from repeatability and discipline. Nevertheless, different approaches are necessary for early phases to support the planning and evaluation of the created information to ensure completeness and structure.

Second, PSS are complex systems that encompass not only internal functional and physical relationships of a tangible product, but also interactions and dependencies among PSS elements, such as products, services, support systems, stakeholders, business elements, and processes (Vasantha et al. 2012). There is a growing perception that designing a PSS should be conducted by conceiving all those elements and their lifecycles as an integrated system, and not in a detached way (Aurich et al. 2006; Ericson and Larsson 2009; Cavalieri and Pezzotta 2012; McKay and Kundu 2014; Vasantha et al. 2015). This systemic perspective of the solution is mainly approached in the initial phases of PSS design (Komoto and Tomiyama 2009; Maussang et al. 2009; Mourtzis et al. 2015; Shimomura et al. 2015). It is essential to create underlying information architectures based on well-founded representation schemes to ensure integration in PSS design before detailing the solution (McKay and Kundu 2014; Hein et al. 2018). After a holistic definition of the integrated solution, the detailing phases can be separately conducted. There, experts in each field (e.g., product designers, service designers, programmers, and so on) will be individually designated to detail each element of the solution, but not entirely detached. They must keep in mind the interfaces and

relationships defined in the architecture and ensure that all elements together provide an appropriate integrated system, which is evaluated through experimentation, prototyping, and other validation techniques (INCOSE 2015).

Some authors propose modeling languages or diagrams to model the PSS as a system, aiming to enhance or depict integration. Examples are CAD-based approaches (Komoto and Tomiyama 2009; Sakao et al. 2009), modeling frameworks (Trevisan and Brissaud 2016), among other proposals (Tran and Park 2016; Medini and Boucher 2019). However, proper integration of products, services, and other PSS elements is still impaired (Zhu et al. 2017; Medini and Boucher 2019). One of the reasons for this lack of integration is the deficiency in PSS theory foundations and methods (Zhu et al. 2017). Another reason is not considering the information requirements for a PSS beyond the product/service scope, ignoring further relevant information for offering the PSS (Durugbo et al. 2010), such as support system requirements, necessary partners, service level agreements that should be fulfilled, among others. Similar to the first challenge, ensuring integration requires appropriate planning and evaluation of the relevant information, ensuring that all PSS elements, interfaces, and further relevant information are being created.

Third, developing such a system requires dealing with a meaningful amount of knowledge created during the process, which includes understanding customers, requirements, product and service technologies and behaviors, lifecycle perspectives, support systems, among others (Shen et al. 2010; Medini and Boucher 2019). This knowledge is collaboratively exchanged among stakeholders with distinct backgrounds and skills, such as engineers, service designers, marketing, programmers, but also partners, suppliers, and customers (Zhu et al. 2015; Hajimohammadi et al. 2017; Correia et al. 2018; Medini and Boucher 2019). Besides the quantity, there is a variety of knowledge manifested by the stakeholders through their domain-specific models and systems. Therefore, a challenge of PSS design is the appropriate gathering and exchange of a meaningful amount of varied knowledge, ensuring its completeness, re-use, sharing (Correia et al. 2018), and a unified perspective of the expected solution and its relevant information among all stakeholders (Hajimohammadi et al. 2017). A well-accepted strategy for elucidating knowledge in PSS design, providing a common understanding among all stakeholders, is visualization through artifacts (Bertoni et al. 2013; Wolfenstetter et al. 2018; Medini and Boucher 2019).

Artifacts are intermediate results of the design process, representing distinct abstraction levels of the solution space (i.e., the design objects, such as the PSS, products, services) and other related aspects regarding the problem space (e.g., requirements, context, constraints, needs). They may be physical representations of the solution (e.g., prototypes) or

documental representations (e.g., documents, models, simulations results) of entities in the solution space and problem space. Artifacts manifest the created knowledge in the shape of information and allow knowledge storage, exchange, and utilization (Visser 2006a; Pei et al. 2011). Those artifacts tend to evolve during the development process, possibly being submitted to modifications and updates.

Aiming at integrating the PSS perspectives and ensuring common understanding, some authors propose model-based frameworks, i.e., sets of artifacts that should be employed for modeling the solution in alignment with a proposed ontology or meta-model [e.g., Zhu et al. (2017), Medini and Boucher (2019)]. However, pre-establishing which artifacts the designers must employ may hinder the applicability of this approach. Designers decide how they will externalize a given piece of information based on their experience, background, and company's context (such as recommended best practices) (Andreasen 1994). Furthermore, artifacts are employed for different reasons during the design process (Rumbaugh et al. 2005), which may lead to distinct structures and styles depending on the person who created it and the goal that she/he wants to achieve. The flexibility of choosing what kind of artifact each designer will create in each situation is lost when artifacts are pre-defined.

However, allowing flexibility for designers to externalize their knowledge also poses a fourth challenge, which regards planning the PSS design. As discussed before, a meaningful amount of knowledge is generated during PSS design, which is manifested in the format of several physical or documental artifacts. Still, there is also a meaningful variety of potential alternative artifacts for each goal. For example, while a designer can elicit requirements by describing user stories, another may employ QFD, and a third may write a requirements list. Each artifact requires a specific procedure and may provide further information not shared by the other potential alternative artifacts. E.g., requirements elicitation through user stories includes information regarding the users' contextual characteristics that may not be covered in a standard requirements list. Furthermore, different artifacts that fulfill distinct goals may have information overlaps. Understanding what kind of information each artifact provides is essential in allowing appropriate planning for the PSS design. A path for systematizing the information contained in artifacts is through concepts¹

¹ Note that the term “concept” in this paper relates to an abstraction for possible pieces of information that an artifact may contain (such as requirements, functions, behavior). The term “concept” as employed in this paper should not be mistaken for the PSS concept (i.e., the PSS representation at a conceptual level). The PSS concept is an artifact, which describes pieces of information that may be generalized under some concepts identified in this paper, such as product, service, support system, among others.

(Medini and Boucher 2019), i.e., the pieces of information that artifacts have, such as requirements, functions, value, activities, resources, among others.

Concept maps, Ontologies, and Meta-models There are many solutions in the literature to deal with information and knowledge-related challenges. Some solutions and their obstacles were cited within the description of the challenges, such as prescribing design process models or model-based frameworks. Another approach usually employed in the design literature to deal with such challenges is creating information structures, such as concept maps (aka conceptual models), ontologies, and meta-models. The concept maps are natural for the users and visually represent knowledge as tangible concepts (Gómez-Gauchía et al. 2004). A concept map graphically represents concepts concerning a given domain and its relationships, aiming to demonstrate how knowledge is structured (Novak 1990). Brilhante et al. (2006) define those concepts as “regularities perceived in events or objects in the world.” Many authors point out conceptual mapping as a method to support capture, elicitation, communication, and organization of domain knowledge (Gómez-Gauchía et al. 2004; Hayes et al. 2005; Shen et al. 2010; Medini and Boucher 2019).

Other authors use ontologies for similar purposes (Pagopoulos et al. 2014; Hajimohammadi et al. 2017). Ontologies provide a common language to describe, communicate, manage and share knowledge (Vasanthi et al. 2015; Zhu et al. 2017; Correia et al. 2018). The difference between structuring a concept map or a formal ontology lies in their distinct complexity levels, but both characterize a given knowledge domain by structuring its concepts and relationships (Brilhante et al. 2006). Concept maps are more informal, simple, and accessible than ontologies (Brilhante et al. 2006). However, a concept map may be employed for incrementally constructing an ontology, mainly when there is a lack of understanding in the domain terms, a lack of experts, or when it is necessary to represent an ontology for domain experts out of the computer science field (Gómez-Gauchía et al. 2004). Finally, in a more artifact-oriented perspective, some authors employ meta-models to express the logical syntactic structures of domain-specific artifacts (Saeki and Kaiya 2006). A meta-model can be obtained by detailing a concept map based on a set of selected artifacts, including the concepts' attributes and their possible assumed values (Milton 2007).

Existing proposals and Gaps In this paper, we propose creating a concept map to depict the information required in the early phases of PSS design (i.e., from the beginning of the front-end of innovation until the end of the embodiment design with the creation of the system architecture).²

² This paper considers the generic phases of PSS design proposed by Rosa et al. (2017), which are the following: Front-end of innovation,

In the PSS literature, there are only a few information structures (i.e., concept maps, ontologies, and meta-models) published when compared with other PSS topics (Vasantha et al. 2015). When associated with knowledge-related challenges, most of those documents represent concepts regarding the PSS as a design object, focusing on functional and structural concepts of products and services [e.g., McKay and Kundu (2014), Zhu et al. (2017)]. Some authors are even more specific, covering only one or few PSS perspectives, such as service, and ignoring other dimensions or further relevant information due to the adopted focus [e.g., Kim et al. (2009), Shen et al. (2010)]. Other authors provide other perspectives of the PSS besides the object itself, such as its stakeholders, infrastructure, lifecycle, or further relevant information for its success [e.g., Hajimohammadi et al. (2017), Correia et al. (2018), Medini and Boucher (2019)]. All those papers have one common characteristic: because of their specific topics or approaches, they all provide an incomplete list of concepts and relationships regarding the relevant knowledge that must be created in the early phases of PSS design. Annamalai et al. (2011) proposed the broadest PSS ontology (on the concept point-of-view) identified in the literature. They present an ontological structure on the PSS design perspective based on eight root-concepts, subdivided into about 320 subconcepts. This structure was refined and agreed upon by 30 international PSS researchers. However, some aspects are still missing in this ontology, such as the services compositional structure and the interface perspective among PSS elements. Furthermore, Annamalai et al. (2011) cover only “is-a” relationships, not encompassing other types of relationships.

Goal and Methodology Therefore, we aim to create a concept map to structure the concepts that compose artifacts resulting from the initial phases of PSS design and how those concepts interrelate. Since there are many partial information structures with distinct foci in the literature, combining them may be a starting point for completeness. We created the concept map based on a comprehensive systematic literature review, analyzing classifications, taxonomies, ontologies, meta-models, and concept maps in the PSS and servitization fields (including non-knowledge-related information structures to enhance completeness). The concept map completeness was verified against formal documentation (artifacts) of two retrospective projects of PSS design (see Sect. 3.4). We present this concept map through a set of smaller graphic views to enhance its ease of use (Gómez-Gauchía et al. 2004) and textual description in Sect. 4.2. The proposed concept map is a fundamental

theoretical basis for creating solutions for planning and verifying the artifacts (and, consequently, the information) created in the initial phases of PSS design. Adequate planning and verification reduce the likelihood of forgetting relevant information, hence reducing rework and enhancing integration. As immediate contributions, the concept map provides a more complete set of relationships and concepts than other existing information structures in the literature, and it may be employed up to a certain point as a checklist to support project managers in planning and verifying the early phases of PSS design projects for completion. Other contributions of the concept map regard the possibility of using it to deploy PSS ontologies, create model-based systems engineering approaches, among others. A detailed description of this concept map’s contributions to theory and practice and its limitations is provided in Sect. 5.

2 Theoretical background

This section is divided into two subsections that provide relevant concepts and discussions to enhance understanding of the proposed methodology and obtained results in this paper. The first Sect. (2.1 The design process, knowledge, and artifacts) clarifies the dynamics of knowledge and information creation and modification in the design process, highlighting the artifacts’ role in these dynamics. The second Sect. (2.2 Information structures) emphasizes possible information structures that may be sources of concepts to build the aimed concept map and describes how those structures relate to each other.

2.1 The design process, knowledge, and artifacts

Design processes are knowledge-intensive processes typically carried out by groups of persons within organizations (Cooper 2003; Torry-smith and Mortensen 2011). To describe the dynamic flow of knowledge in design processes, Wang et al. (2017) propose a functional model for milestone-driven design processes, which considers three main perspectives: the “trigger model,” the “individual model,” and the “social model.” This functional model is illustrated in Fig. 1 and described in the following paragraphs.

First, the “trigger model” describes the organization’s specific *Design process* as the starting point for the conducted design activities, setting boundary conditions (e.g., process organization, resources, tools) and supporting the planning process. Many design processes follow phase-based models prescribing how the design process should proceed (Stacey et al. 2020). However, design processes have a multi-layered nature that should not be limited to the phase level only (McMahon 2015). The trigger model of Wang et al. (2017) covers three of McMahon’s (2015)

Footnote 2 (continued)

Embodiment design, Detailed design, Implementation, Use, and End-of-life.

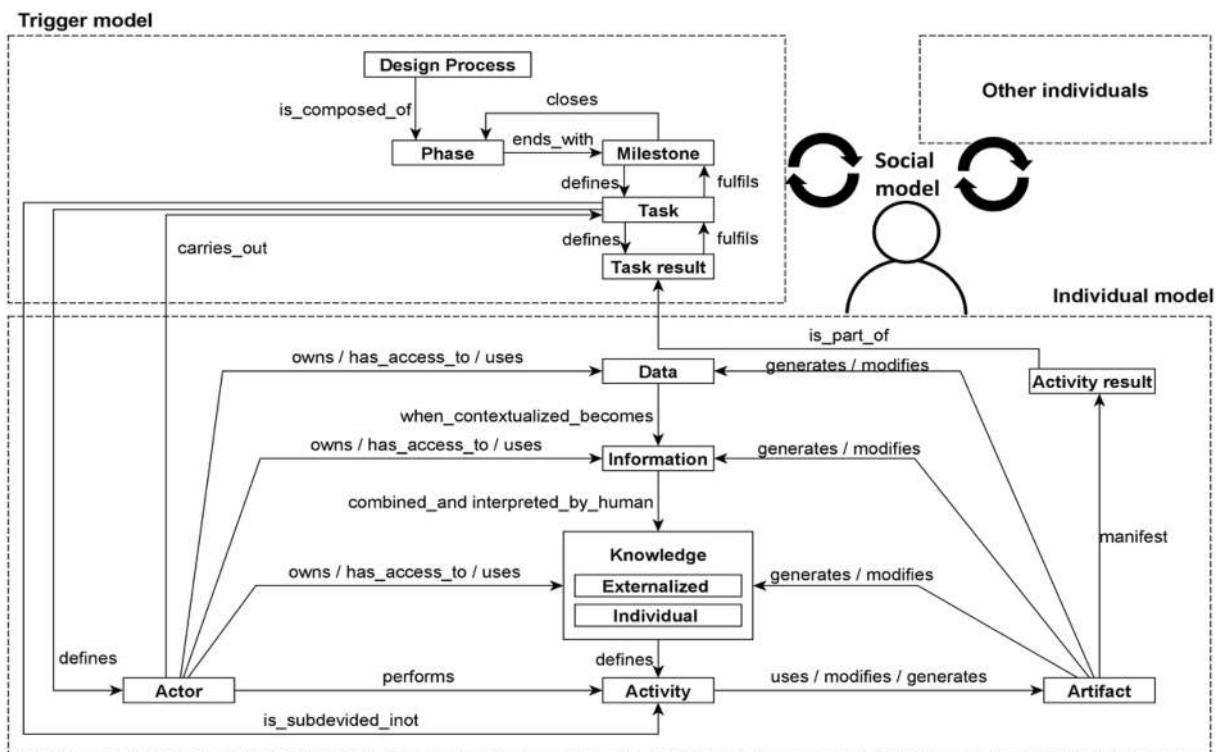


Fig. 1 Functional model for a milestone-driven design process [adapted from Wang et al. (2017)]

possible design process layers. Such a *Design process* is composed of *Phases* (cf. *Stages*, McMahon 2015), which end by achieving a set of *Milestones* (cf. *Work packages*, McMahon 2015), i.e., the completion of the expected *Milestones* closes the ongoing *Phase*. From abstract design goals, the project manager derives those *Milestones* and defines *Tasks* (cf. *Tasks*, McMahon 2015), which must be performed to fulfill those *Milestones*. Based on each *Task*, the project manager establishes a set of expected *Task results* (also known as deliverables), which will accomplish this *Task*, and defines the most appropriate *Actors*, with specific “knowledge profiles” (Gainsburg et al. 2010), to carry out the *Task*. The aggregated knowledge from the *Tasks’ results* must be distributed among all stakeholders to create a common baseline for subsequent phases of the process.

Many *Actors* may carry out each specific *Task*. However, when dealing with knowledge, it is interesting to limit our perspective to each *Actor* in the *Design process*. Therefore, the second perspective of the functional model proposed by Wang et al. (2017) is the “individual model,” which describes each *Actor’s* action and knowledge perspective. This perspective subdivides the *Tasks* into a fourth layer of the design process—the *Activities* (cf. *Activities*, McMahon 2015), which are the actions that each *Actor* performs individually as part of the scope of the *Task*, such as eliciting a set of requirements, creating a concept, or drawing a specific component. Apart from purely communicational *Activities*

(e.g., conversation), every *Activity* will encompass generating new *Artifacts*, modifying existing *Artifacts*, or using existing *Artifacts* (cf. *Information objects*, McMahon 2015). The *Artifacts* that result from the *Activity* express the *Activity result*. The resulting *Activity results* need to be integrated and evaluated, composing the *Task results*, i.e., when the *Actors* perform all *Activities* in the scope of a *Task*, all the *Results* of those *Activities* will constitute the *Task results* to fulfill the given *Task* (Lutters et al. 2014).

To clarify the knowledge perspective, first, it is essential to point out the relationships among *Data*, *Information*, and *Knowledge* (North and Kumta 2018). *Data* are facts or observations that, alone, lack meaning, being unorganized, and unprocessed (Rowley 2007). When *Data* is organized and structured, becoming meaningful and relevant, it becomes *Information* (Rowley 2007), i.e., *Information* is contextualized *Data*. Some authors refer to *Information* as “weak-knowledge” (Frické 2019). *Knowledge* is combined information sets analyzed by a human through her/his understanding, experiences, skills, and values (Rowley 2007), i.e., *Knowledge* is combined and interpreted *Information*.

There are two moments in the individual model where *Data*, *Information*, and *Knowledge* play different roles: before performing a given *Activity* and while performing it. Before performing the *Activity*, the *Actor* has an individual tacit *Knowledge*, i.e., she/he owns a set of *Individual Knowledge*, *Information*, and *Data*, which is dependent on

her/his background, experiences, and learnings (Snowden 2000). However, this *Individual Knowledge, Information, and Data* may not be enough to perform the *Activity*. In this case, during the *Activity*, the *Actor* may use the *Artifacts* created in previous *Activities* of the *Design process*. Those *Artifacts* manifest the *Externalized Knowledge* applied or created during the *Design process*. The *Artifacts* contribute substantially to the *Design process* progress, as they represent intermediate states of the intended result and enable collaboration across disciplinary contexts (Visser 2006a; Pei et al. 2011; Mariano and Awazu 2016). By using those *Artifacts*, the *Actor* has access to the *Data, Information, and Externalized Knowledge* that were documented in the *Artifacts*. The *Actor* uses them by interpreting, transforming, and integrating this *Data, Information, and Externalized Knowledge* to her/his individual tacit *Knowledge*, with which the *Actor* performs the *Activity* and which will influence how the *Actor* will realize it (Visser 1992). While performing the *Activity*, new *Individual Knowledge* (e.g., experience with a given tool) and *Externalized Knowledge* (e.g., simulation results) are created. A meaningful part of this *Knowledge* is made explicit by generating a new *Artifact* (e.g., creating a simulation report from simulation results), or by modifying an existing *Artifact* (e.g., update a simulation report that already existed previously) (Lünnemann et al. 2017). This new *Artifact* may lead to the generation of new *Data, Information, and Knowledge (Individual and Externalized)* or to evolving existing *Data, Information, and Knowledge (Individual and Externalized)* when used in a subsequent activity of the design process.

Third, the “social model” describes the interactions between individuals and groups that enable the transformation of knowledge created during the process that leads to organizational learning. This model is based on the transformation of tacit knowledge (i.e., the inexpressible personal knowledge) and explicit knowledge (i.e., the expressible and recordable knowledge documented as artifacts) (Frické 2019). According to the SECI model proposed by Nonaka and Takeuchi (1995), organization learning takes place by the combination of the processes of Socialization (tacit to tacit knowledge), Externalization (tacit to explicit knowledge), Combination (explicit to explicit knowledge), and Internalization (explicit to tacit knowledge). Through social interactions during daily work, a person gains and creates knowledge, which is implicit at first. By externalizing this knowledge, it becomes explicit and can be transferred and shared. By combining the externalized knowledge, the organization extends its knowledge base, and new knowledge is made available for the entire organization. Then, the organization can use this knowledge to create further tacit and explicit knowledge.

Artifacts—such as documents, models, simulation results, and prototypes – partially represent the knowledge created

through a design activity (Lutters et al. 2014), allowing for its storage, exchange, and utilization (Visser 2006a; Pei et al. 2011). Therefore, artifacts contain information regarding the explicit knowledge created during the design process (Ullman 2010). This information represents different characteristics of the desired design object,³ its environment, and its problem space, with varying levels of abstraction depending on the maturity that was achieved in the design process (Visser 2006b). For example, in a product design process, part of the knowledge regarding this product during the conceptual design phase may be represented by a sketch of the product layout. In contrast, at the end of the detailed design phase, part of the explicit knowledge may be represented in a detailed drawing from the entire product, showing components’ interfaces, dimensions, and tolerances.

The content of those artifacts may be broken down into different classes of information—which are labeled as concepts in this paper—such as needs, requirements, functions, behavior, structure, subsystems, product components, among others. Therefore, this paper approaches artifacts under the *Information* perspective. Furthermore, each artifact depicts a set of attributes for each concept that composes this artifact. For example, when analyzing a detailed CAD drawing (artifact) that represents a product component (concept), it may contain attributes as shape, dimensions, tolerances, material, among other characteristics that, when instantiated, provide detail to the concept “product component.”

Finally, concepts interrelate. First, a concept may be a subclass of another concept, e.g., “stakeholders’ requirement” is a subclass of “requirement.” Second, one or more concepts may be input to or influence the instantiation of a new concept. For example, “stakeholders” leads to identifying “needs,” which will be translated into “requirements” for the desired design object. Simultaneously, “functional requirements” may be an input for establishing the desired design object’s functions.

2.2 Information structures

This paper focuses on identifying concepts that compose the artifacts resulting from the early phases of the PSS design process and identifying their relationships. This section aims at providing a theoretical background to justify why classifications, taxonomies, ontologies, meta-models, and concept maps are good sources for identifying concepts. The relationships among such documents and the concepts are illustrated in Fig. 2 and explained in the following paragraphs.

³ We employed the term “design object” in this paper as the tangible or intangible artifact resulting from the design process as its primary goal (Rosa and Rozenfeld 2019), e.g., a product, a service, a PSS, among others.

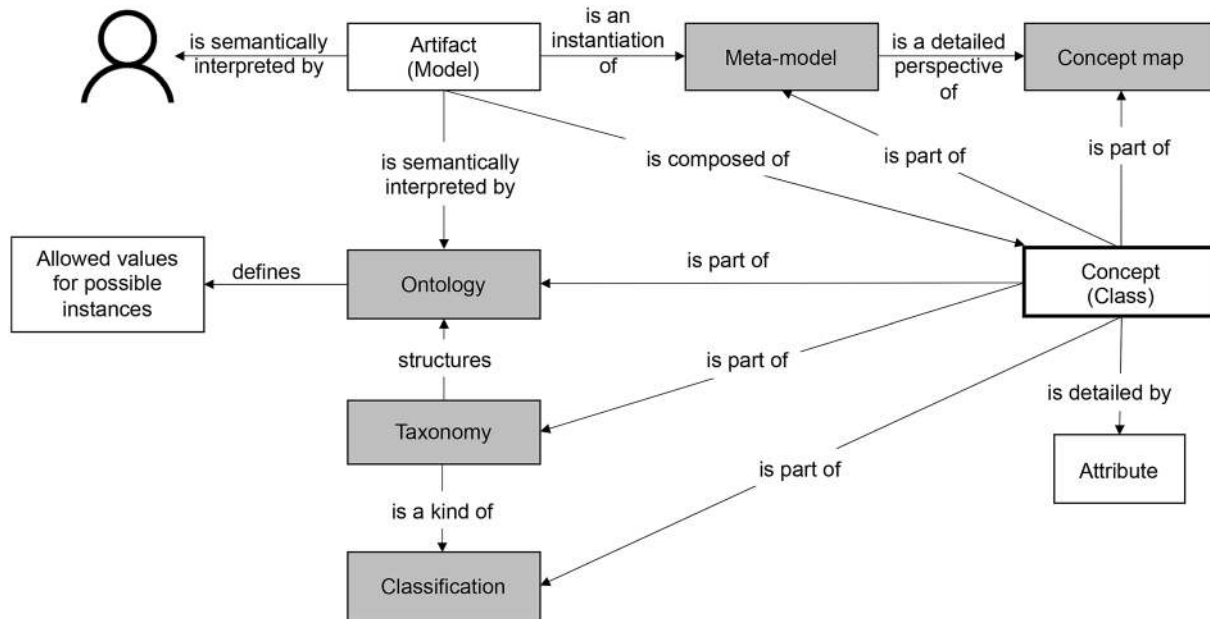


Fig. 2 Relationship among the terms “classification,” “taxonomy,” “ontology,” “meta-model,” and “concept map” based on the definition of “concept” (elaborated by the authors)

The terms “concept” or “class” are recurrent in classifications, taxonomies, ontologies, meta-models, and concept maps, and, therefore, it is essential to define them. A concept is “a mental representation of a simple class” of an object, where, in the classic view, “all instances of a concept share common properties that are necessary and sufficient conditions for defining the concept” (Medin and Smith 1984). A class is a generalization of “a set of objects with similar structure, behavior, and relationships” (Cho and Lee 2011). Based on these definitions, it is possible to notice that both the terms “concept” and “class” are synonyms. As pointed out before, this paper is interested in concepts representing the information of artifacts resulting from the initial phases of PSS design. An example to allow a better visualization may be based on the artifact “requirements list”. A requirements list is composed of instantiations of the concept “requirements.” The attributes detail each concept, such as requirement title, ID, description, requirement type, and source. The project team’s goals will outline which attributes are necessary to provide all the required information regarding a concept. Other examples of concepts that are relevant for the objective of this paper could be “need,” “stakeholder,” “process,” or “subsystem.”

In the context of a PSS design, each resulting artifact is an abstraction of a given perspective of the PSS solution or problem space, i.e., each artifact is a model. A model is an abstraction of reality representing the selected properties of a tangible or intangible object (da Silva 2015). Each model conforms to a meta-model (Bézivin 2004). A meta-model can “express the logical syntactical structures

of domain-specific models” (Saeki and Kaiya 2006). An example is a meta-model proposed by Maleki et al. (2018a) to represent PSS solutions based on the systems engineering approach. According to Saeki and Kaiya (2006), “a model is an instantiation of a meta-model, and the domain ontology semantically interprets it.”

We adopt the following definition for ontology: “a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations” (Princeton University 2010). Therefore, an ontology’s structure is a “subclass based taxonomic hierarchy” (Rees 2003). An example of a PSS ontology is the one proposed by Annamalai et al. (2011).

A “taxonomy” is “a hierarchy created according to data internal to the items in that hierarchy” (Rees 2003). A well-known taxonomy is the Linnaean taxonomy for categorizing organisms. There are also some taxonomies in the PSS research field, such as the taxonomy of technology roadmaps in service areas proposed by Cho and Lee (2011). We highlight, however, that many authors employ the label “ontology” to incorrectly label taxonomies. An ontology should be composed of classes structured in a taxonomic hierarchy and define the allowed values for the possible instances (Noy and McGuiness 2001).

This paper employs WordNet’s definition of “classification” (Princeton University 2010)—“a group of people or things arranged by class or category.” A taxonomy is a kind of classification. The difference between taxonomy and classification derives from the categories that classify the concepts and their structure. Classifications may employ arbitrary external

Formalism

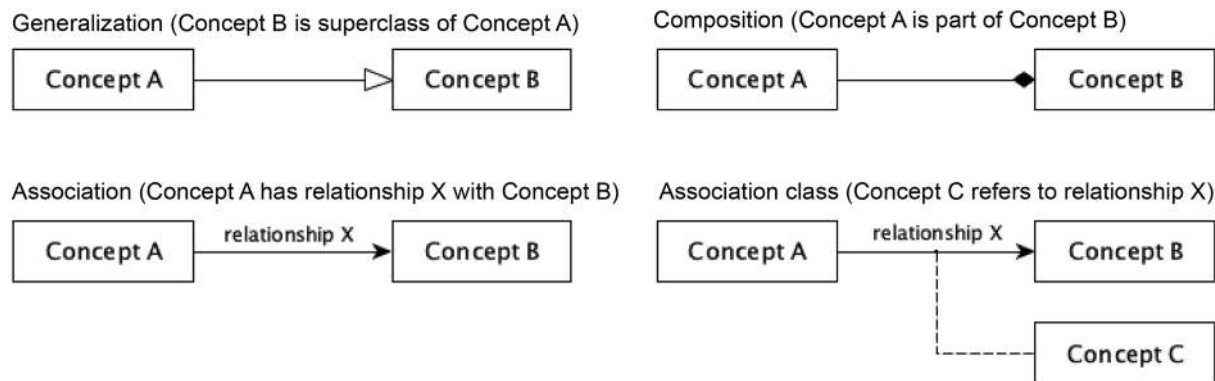


Fig. 3 Formalism for UML class diagrams [adapted from Purchase et al. (2003)]

criteria for grouping concepts (e.g., categorizing a consulting service as a business service based on the service sector branches). Differently, a taxonomy uses the concepts' private properties to describe their hierarchical relationships (e.g., subdividing a service into online services based on their channels and subdividing them again into automate services based on how their operation is structured) (Rees 2003). An example in the PSS literature is the classification proposed by Tukker (2004), which categorizes the types of PSS.

A concept map, aka conceptual diagram or conceptual model, is a tool that illustrates conceptual understanding, arising in the science education field (Novak 1990). A concept map represents concepts and their relationships, aiming at demonstrating how knowledge is structured. Concept maps are also useful for capturing, organizing, and representing knowledge from different areas and understanding unstudied or understudied fields (Novak 1990; Milton 2007; Donnelly 2017). In the knowledge management theory, Milton (2007) states that a meta-model is a detailed perspective of a concept map representing concepts, relationships, attributes, and possible values. Under this perspective, while the concept map may stop its detail level in the concepts, a meta-model should also bring attributes and possible values.

Based on the discussion presented in this section, we propose that systematically reviewing existent meta-models, ontologies, taxonomies, classifications, and concept maps may provide a set of concepts compatible with this paper's objectives.

3 Methodology

This paper followed a methodological approach composed of four main steps, which are shortly described below and expanded in the following subsections. The

methodological references employed in each step are also presented in each respective subsection.

Pre-analysis (Sect. 3.1): Identification of classifications, taxonomies, ontologies, meta-models, and concept maps proposed in the literature, from which we can extract concepts and their relationships;

Material exploration (Sect. 3.2): Identification, codification, and categorization of the identified documents, their proposed concepts, and relationships;

Treatment, interpretation, and inference (Sect. 3.3): Identification of which codified concepts compose artifacts resulting from the initial phases of PSS design. In this step, unnecessary relationships and bridge concepts were removed, and missing relationships were derived. The concepts and their relationships were graphically illustrated with the formalism for UML class diagrams for enhancing the process reliability;

Completeness verification against retrospective PSS design cases (Sect. 3.4): Verification of the concept map capacity of providing the necessary concepts and relationships to depict all artifacts resulted from two retrospective PSS design projects. Missing concepts and relationships that were identified during the verification were used to complement the concept map. The final results were also illustrated graphically with the formalism for UML class diagrams to ease readability.

Besides the final concept map, some diagrams in the methodology and results of this paper also follow the formalism for UML class diagrams. Figure 3 illustrates this formalism.

In the following topics, we exemplify each of the relationships from Fig. 3:

Table 1 Search parameters for the systematic literature review (elaborated by the authors)

| Search parameter | Used value |
|----------------------|---|
| Database | Scopus & Web of Knowledge |
| Time frame | Until September 2019 |
| Type of publications | Journal papers, Peer-reviewed conference proceedings, Book chapters |
| Fields | Title, Keywords, Abstract |
| Search String | TITLE-ABS-KEY(("servicization" OR "servicification" OR "service transition" OR "service infusion" OR "service growth in product firms" OR "value migration" OR "customer solutions" OR "servici?ing" OR "service engineering" OR "product?service system" OR "serviti?ation" OR "product-service offering" OR "value bundle" OR "bundles of benefits" OR "complex service system" OR "hybrid offering" OR "systems selling" OR "full service" OR "service package" OR "installed base service" OR "integrated solution" OR "eco-efficient producer service" OR "functional sales" OR "functional product" OR "product and service offering" OR "product service offering" OR "integrated product service" OR "product and service engineering" OR "product service bundling" OR "producti?ation" OR "service product engineering" or "total care product") AND ("taxonomy" OR "ontology" OR "meta-model" OR "concept map" OR "conceptual diagram" OR "conceptual model" OR "classification")) |

- *Generalization* Automotive vehicle (Concept B) is the superclass of Passenger Car (Concept A), i.e., Passenger Car (Concept A) is a type of Automotive vehicle (Concept B).
- *Composition* Wheel (Concept A) is part of Passenger Car (Concept B), i.e., Wheel (Concept A) is a compositional part of Passenger Car (Concept B).
- *Association* Customer (Concept A) purchases (relationship X) Passenger Car (Concept B). In this case, a customer is not a compositional part of a car; neither is the customer a type of car. They have a clear association relationship (i.e., "purchases").
- *Association class* Whenever a Customer (Concept A) purchases (relationship X) a Passenger Car (Concept B), some information might be associated with it (for example, the data of an invoice). Those data cannot be attributed to the customer itself nor to the car itself, but to the relationship between both concepts. In this case, the relationship leads to the association class "Purchase" (Concept C); i.e., this class is attributed to a relationship between two existing concepts.

3.1 Pre-analysis

As explained in the theoretical background (see Sect. 2.2), some documents structure knowledge as interconnected concepts. Those documents are a starting point for this research. Therefore, the first step of this methodology consists of a systematic literature review (Tranfield et al. 2003) to identify classifications, taxonomies, ontologies, meta-models, and concept maps proposed in the literature, from which we can extract concepts. The search string was

composed of those terms combined with words commonly associated with PSS and servitization (Boehm and Thomas 2013; Annarelli et al. 2016; Rabetino et al. 2018). The search was conducted in October 2019, covering papers published until September 2019 and exploring the Scopus and Web of Knowledge databases. Table 1 presents the search parameters employed for this literature review.

The search resulted in 473 documents. We employed the following inclusion criteria to identify relevant papers for this research:

1. The publication proposes a PSS or servitization-related classification, taxonomy, ontology, concept map, or meta-model composed of concepts, possibly presenting their relationships or not.
2. At least one of the proposed concepts refers to information that might be contained in an artifact of the initial PSS design phases.

Based on those inclusion criteria, we followed two screening steps to exclude all publications that are not relevant to this research. In the first screening step, all 473 documents had their title, abstract, and keywords read, resulting in 78 papers that proceeded to the second step. The second screening step encompassed reading the 78 documents in their totality, leading to 29 selected papers. We extracted the references from those 29 publications, also submitting them to a two-step analysis based on the inclusion criteria, ensuring a backward search approach, as recommended by Webster and Watson (2002). Three other documents were identified and included in the analysis.

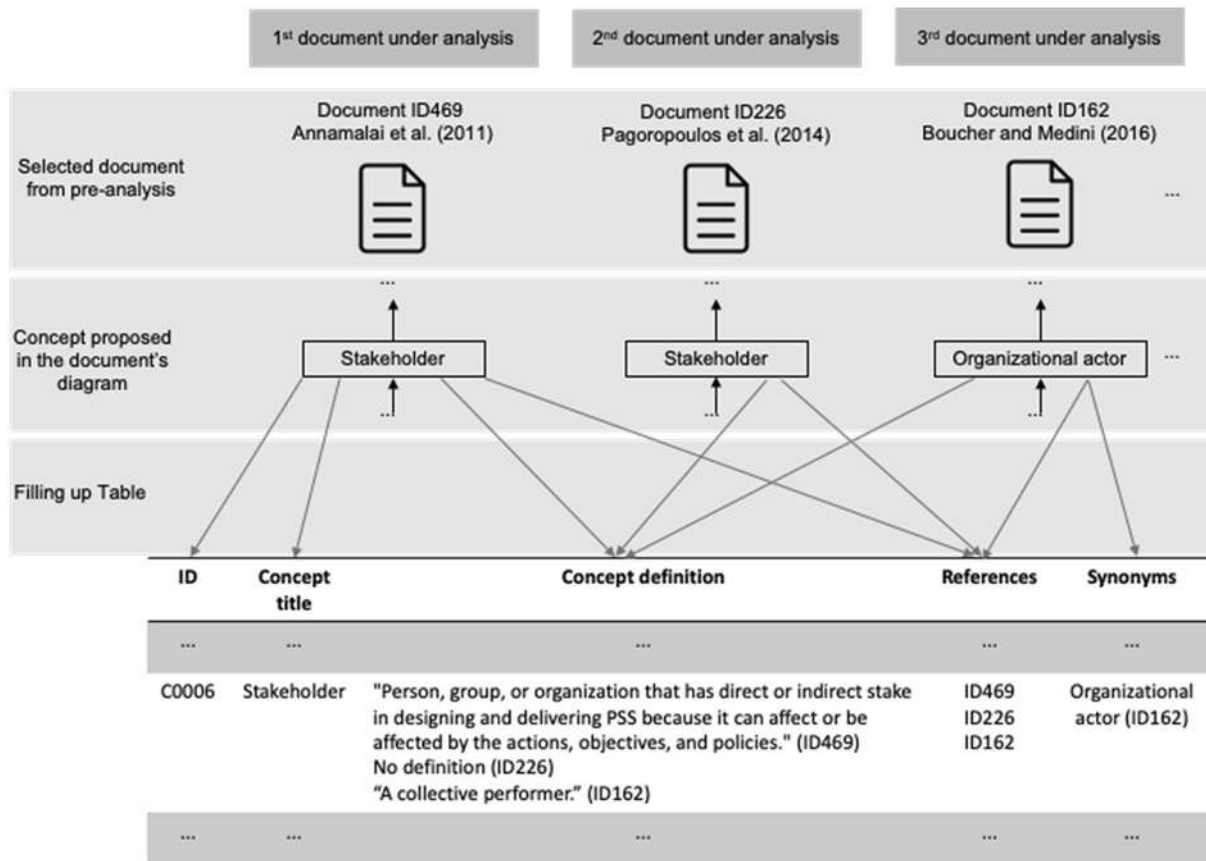


Fig. 4 Example of identification and codification of the concept “Stakeholder” based on the extraction of three of the identified documents (elaborated by the authors)

Appendix 1 provides the complete list of 32 publications, followed by a summarized description of their content.

3.2 Material exploration

The second step (material exploration) was conducted by employing the content analysis method (Bardin 2013) to identify, codify, and categorize concepts and their relationships in the selected documents. Those documents usually present concepts in the shape of diagrams (such as UML) or tables, already discretizing them under a given title. Each concept was included in a table, following the process explained below and exemplified in Fig. 4.

One of the researchers conducted the extraction, associating each concept to an ID (e.g., C0006), its title (e.g., *Stakeholder*), its definition according to its source (e.g., "Person, group, or organization that has a direct or indirect stake in designing and delivering PSS"), and its source [e.g., Annamalai et al. (2011)]. Whenever another author cited the same concept, the new definition and the reference

were added to the table under the same ID. Some authors have referred to the same concepts with different words (e.g., *Stakeholder* or *Organizational Actor*). However, based on the proposed definitions, it was possible to state whether two distinct titles referred to the same concept. Whenever the author did not present an explanation for a new concept, we employed a definition from the English lexical database Wordnet (Princeton University 2010) to verify whether the concept was a synonym of another one or not.

After filling the entire concepts table, the same researcher conducted a similar procedure to analyze the relationships, which is explained below and exemplified in Fig. 5. The analyzed documents also present the relationships in the shape of diagrams (such as UML), easing the extraction process. Each relationship presented by each document was also included in a table, being identified by an ID (e.g., R00377), its title (e.g., “is superclass of”), the input concept name (e.g., *Service*), the output concept name (e.g., *Elementary Service*), and its source [e.g., Letia and Marginean (2008)].

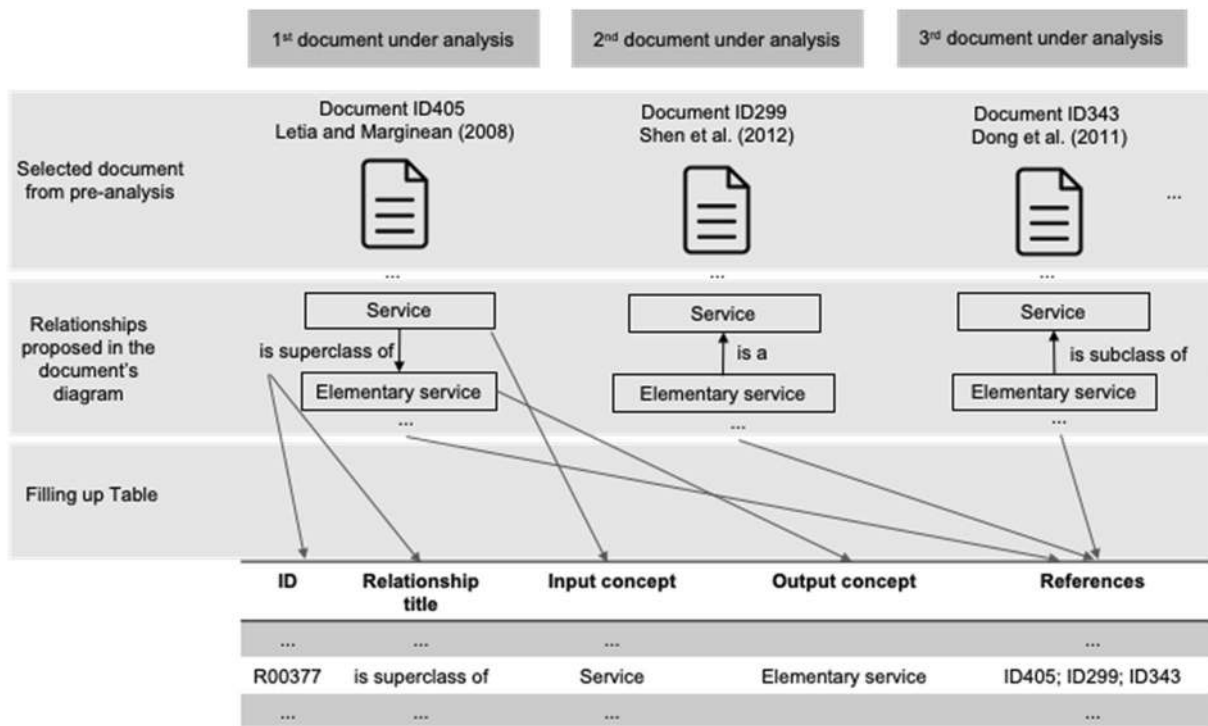


Fig. 5 Example of identification and codification of the relationship “Service is superclass of Elementary service” based on the extraction of three of the identified documents (elaborated by the authors)

In some relationships, the output might not be a concept but another relationship. For example, as illustrated in Fig. 5, when a stakeholder performs an activity of her/his activities lifecycle,⁴ she/he is submitted to a given experience. Therefore, the concept *Experience* is part of the relationship “*Stakeholder performs Stakeholder’s activity lifecycle*,” i.e., “*Experience*” is an association class (see Fig. 3). Association classes were also included in the table.

Attempting to follow the UML standards, inheritance relationships were always resumed to “is superclass of” relationships (e.g., relationships that state that a concept A “is a” concept B were reframed to state that concept B “is superclass of” concept A). Meanwhile, compositional relationships were always resumed to “is part of” relationships, independently of the authors’ terminology or symbolism (e.g., relationships that state that a concept A “has” concept B were reframed to state that concept B “is part of” concept A). Further association relationships kept their titles (e.g., “satisfies,” “provides,” among others). In some cases, the

same relationship was provided by different authors with distinct synonyms or with two words that are hypernyms⁵ of each other. In such cases, they were unified under the same ID employing one of the synonyms or hypernyms (e.g., “interfaces with” is a hypernym of “collaborates with”).

To provide a deeper understanding of how the concepts and relationships are extracted, we offer a visual example in the following paragraphs. We have selected the final concepts *Service*, *Service package*, and *Elementary service* (see Sect. 4.2, Fig. 15) to illustrate how the material exploration was performed. When exploring the papers resulting from the systematic literature review, six of all documents presented diagrams that contained the extracts shown in Fig. 6.

As explained in this section, we first extracted all concepts, codified them in a table, and established which concepts were synonyms. An extract of the final table with a focus on the concepts of Fig. 6 is presented in Table 2. Please, note that, since this is an extract of the original concepts table, the concept IDs are not in a subsequent order. The original concept table may be accessed in the supplementary electronic material.

⁴ The concept *Stakeholder’s activity lifecycle* encompasses all activities performed by a given *Stakeholder* in the entire solution lifecycle, including pre-purchase, purchase, and post-purchase (Annamalai et al. 2011).

⁵ Saying that “A is a hypernym of B” means that “B is a kind of A” (Ex: “car” is a hypernym of “cab”) (Ignatow and Mihalcea 2017).

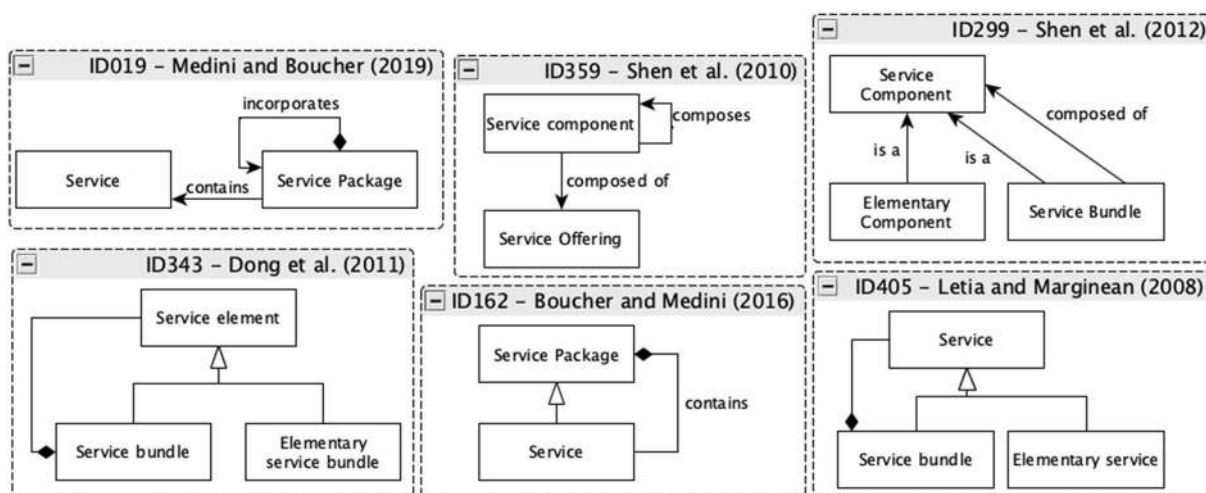


Fig. 6 Example of material exploration regarding the final concepts Service, Service package, and Elementary service and their interrelationships (elaborated by the authors)

After extracting all concepts and associating their synonyms, it is necessary to codify the relationships among those concepts, already treating which relationships are synonyms. As explained before, relations of the type “X is a Y” are equivalent to “Y is a superclass of X.” Furthermore, some relationships in the example are compositional (e.g., composes, is composed of, contains, incorporates). As explained before, compositional relationships, such as “X composes Y,” are equivalent to “X is part of Y.” Based on those equivalences, the relationships in Fig. 6 were codified, as illustrated in Table 3, which contains an extract of the relationships table. Please, note that, since this is just an extract of the original relationships table, the relationship IDs are not in a subsequent order. The original relationship table is provided in the supplementary electronic material. This example continues in the next section, following the process of treatment, interpretation, and inference.

3.3 Treatment, interpretation, and inference

The third step (Treatment, interpretation, and inference) employed the content analysis method (Bardin 2013) to identify which from the extracted concepts may compose artifacts resulting from the initial PSS design phases. It also aimed to reduce unnecessary relationships, add missing relationships, and remove bridge concepts. This step was conducted by the same researcher who led the second step.

Only concepts that are instantiable in design artifacts were considered in this paper since we deal with the design process domain. For example, in a requirements list (artifact), it is possible to identify many instances of requirements. Therefore, *Requirement* is a concept of interest for this paper. Excluding concepts that do not match this scope was essential to keep the concept map in a manageable size

and consistent with this paper’s goal. The following types of concepts were excluded:

Concepts that referred to artifacts (e.g., contract) Many artifacts may be created during the PSS design process. As explained in this paper’s introduction, designers decide how they will externalize a given piece of information based on their experience, background, and company’s context (Andreasen 1994). Companies may also create unique artifacts based on their know-how. Therefore, if concepts equivalent to artifacts were kept in the concept map (such as contract, bill of material, design brief, among others), it would be necessary to ensure that other possible artifacts were also included. Therefore, we decided to keep the concept map in a level of abstraction representing only the content of artifacts (information) at a class level. For example, a contract may be composed of concepts such as *Service level agreements*, *Price*, and *Renewal contract characteristics*. Those concepts were kept in the concept map, but the artifact (i.e., contract) was excluded. In the conclusion section, we clarify how the concept map will connect with the artifact level in future research.

Low-level concepts that already constituted attributes of another concept (e.g., adaptability) Many extracted concepts regard very specific and detailed information, which may be simplified as an attribute of another concept. For example, the customer profile may be described as an attribute of the *Customer* instead of a new concept. Therefore, to keep the concept map in a manageable size, low-level concepts were excluded.

Instances of a concept (e.g., maintenance, which is an instance of a business process) The exclusion of instances was performed, aiming at generalization. For example,

Table 2 Extract of the concepts table focused on the concepts *Service*, *Service package*, and *Elementary service*

| ID | Concept title | Concept definition | References | Synonyms |
|-------|--------------------|---|--|--|
| C0077 | Service | No definition (ID019); “The intangible result of activities realized by some actor(s) with the intention to create and deliver value for some other actor(s), resulting in a change of state for this (these) actor(s).” (ID162); “ServiceComponent is a constituent element, of which PESs (product extension services) consist. It serves as a building block to support on-demand service composition, which means one service component can be composed of other components.” (ID299); “Service elements, which are defined as a service in business literature, represent what a supplier offers to its customers. [...] A service element can be decomposed into smaller service elements, as long as these small elements can be offered to customers separately, possibly by different suppliers.” (ID343); “The ServiceComponent may be an atomic element that cannot be further decomposed, or a composite one that can be decomposed into smaller ServiceComponents, as long as these smaller ServiceComponents can be offered to customers separately with specific functions and processes.” (ID359) No definition (ID405); | ID019; ID162; ID299; ID343; ID359; ID405; | Service element (ID343); Service component (ID299; ID359); |
| C0372 | Service package | No definition (ID019); No definition (ID162); “Service bundles are just composite services which employ and synthesize a set of core components, and possibly supporting and enhancing components through various binding rules.”(ID299) “Service bundle is a set of one or more service elements that can be provisioned together as a whole compatible with the rules.”(ID343); “ServiceOffering provides a hierarchy of ServiceComponents from a provider’s perspective. It describes what a provider offers to its customers based on the component-based structure.” (ID359); “A service bundle aggregates services based on relations like enhancing, supporting, and substitute or excluding.” (ID405); | ID019; ID162; ID299; ID343; ID359; ID405; | Product Extension Service (ID299); Service bundle (ID343; ID405); Service offering (ID359); |
| C0375 | Elementary service | “Elementary component is the smallest service unit that, from a commercial point of view, can be meaningfully offered to customers by a ServiceProvider.” (ID299); No definition (ID343); No definition (ID405); | ID299; ID343; ID405; | Elementary component (ID299; ID359); Elementary service bundle (ID343); |

Table 3 Extract of the relationships table focused on the concepts *Service*, *Service package*, and *Elementary service*

| ID | Relationship title | Input concept ID | Input concept title | Output concept ID | Output concept title | References |
|--------|--------------------|------------------|---------------------|-------------------|----------------------|---------------------------------------|
| R00376 | Is superclass of | C0077 | Service | C0372 | Service package | ID299; ID343; ID405; |
| R00377 | Is superclass of | C0077 | Service | C0375 | Elementary service | ID299; ID343; ID405; |
| R00386 | Is part of | C077 | Service | C0372 | Service package | ID019; ID162; ID299; ID343; ID405; |
| R00388 | Is part of | C0375 | Elementary service | C0372 | Service package | ID244; |
| R00435 | Is part of | C0372 | Service package | C0077 | Service | ID359; |
| R00463 | Is part of | C0077 | Service | C0077 | Service | ID299; ID359; |
| R00847 | Is superclass of | C0372 | Service package | C0077 | Service | ID162; |
| R00928 | Is part of | C0372 | Service package | C0372 | Service package | ID019; |



Fig. 7 Example where reducing unnecessary relationships may be applicable (elaborated by the authors)

given instances of a *Business process*, such as refurbishing or maintenance, are applicable in specific PSS design projects. Even though those two specific processes are usually very relevant to ensure the feasibility of many PSS business models, they might not be necessary in some cases. Each specific PSS will require a different set of *Business processes*. Therefore, it is essential to state in the concept map that *Business processes* should be defined in the initial phases of PSS design (i.e., *Business process* is a concept). Then, during each specific PSS design project, the design team may define which specific *Business processes* must be created for that PSS, which may include maintenance and refurbishing. However, we highlight that readers interested in using or reading those instances may access them by filtering the workbook available as supplementary electronic material.

Abstract concepts that are challenging to instantiate (e.g., mindset) the exclusion of abstract concepts was led by the concept map goal (i.e., depicting the concepts that compose artifacts resulting from the initial PSS phases design, as well as their relationships). If something is too abstract to be represented as an artifact's information, it cannot compose the artifacts. For example, even though the mindset transformation is essential during servitization, how can a team describe a mindset in a document that results from the PSS design? The mindset should be slowly developed within the team, but it does not apply to the concept map's goals proposed in this paper.

Another task in the process of treatment, interpretation, and inference is to reduce unnecessary relationships. It regards removing connections that are already covered by other relationships. All remaining concepts and their relations were graphically illustrated to visualize whether they were redundant. As shown in Fig. 7, an example is when the relationships “*Capability is part of Stakeholder*” and “*Stakeholder is superclass of Partner*” are proposed. In this case, also stating that “*Capability is part of Partner*” would be redundant, since *Partner* is a type of *Stakeholder*, inheriting its relationships. Therefore, the relationship “*Capability is part of Partner*” may be excluded.

Some of the remaining relationships were also excluded for being incompatible with other relationships. For example, Annamalai et al. (2011) propose that the concept *Requirement* may be subclassified as *Stakeholder requirement*, *Product-service requirement*, and *Support system*

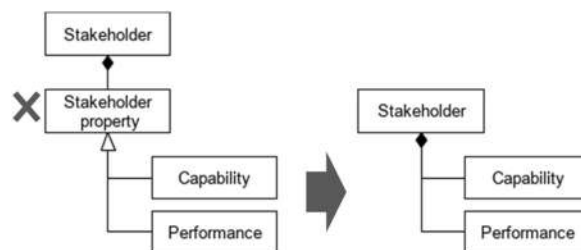


Fig. 8 Example of bridge concept exclusion (elaborated by the authors)

requirement, while *Functional requirement* is a type of *Product-service requirement*. In contrast, Shen et al. (2012) propose that *Functional requirement* is a type of *Stakeholder requirement*. In such cases, we have followed a three-step verification. First, we verified if the conflicting relationships (for instance, relationships A and B) were also contradicting other relationships or the definition of the involved concepts. If relationship A did not contradict other existent relationships or concept definitions, while relationship B does, then relationship B was excluded. If the first step was not enough to make a decision, we verified if the conflicting relationships were cited by more than one source in the systematic literature review, giving preference to the most cited one. Finally, if the same number of authors cited both conflicting relationships, then the concepts connected by the conflicting relationships were employed as keywords in a new literature search, keeping the relationship corroborated by the most cited papers in this search.

After reducing unnecessary relationships, we looked for missing relationships based on the other relationships depicted in the concept map. For example, the three following concepts appeared in the references: *PSS quality*, *Service quality*, and *Product quality*. With an intense focus on service development, one of the authors stated that “*Service quality is part of PSS quality*.” However, the concept *Product quality* was never connected to *PSS quality* by any of the authors. In such a case, it is clear that the relationship “*Product quality is part of PSS quality*” is missing, being derived based on the existing relationship “*Service quality is part of PSS quality*.”

The last task in the treatment, interpretation, and inference process is to remove “Bridge concepts.” These have the sole function of organizing a given number of concepts, not representing a possible class of information. One example of a bridge concept, as illustrated in Fig. 8, is the *Stakeholder property* proposed by Annamalai et al. (2011), which is part of the concept *Stakeholder*. The *Stakeholder property* is a superclass of some concepts, such as *Capability* and *Performance*. In this analysis, bridge concepts organizing less than five concepts were removed, connecting the subclasses of the bridge concept directly to the concept of which the bridge

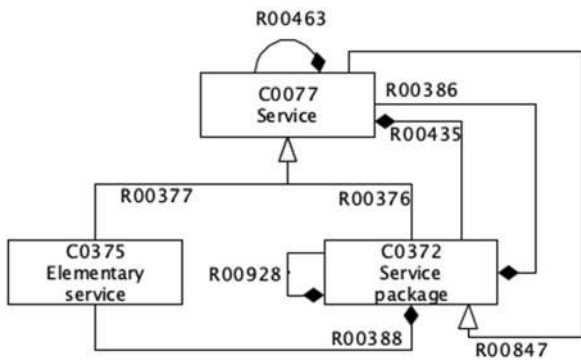


Fig. 9 Graphical representation of the outcomes from the example represented in Tables 2 and 3

concept is part, i.e., *Capability* and *Performance* are part of *Stakeholder*. Other bridge concepts organizing more than five concepts were kept to ease the concept map readability.

After the first researcher performed the treatment, interpretation, and inference process, a second researcher verified all concepts and relationships of the resulting concept map aiming to identify possible missing concepts and relationships, as well as questionable connections. All points highlighted by the second researcher were discussed with the first researcher, retrieving the sources and identifying whether the relationships and concepts could have been treated, interpreted, or inferred differently. The discussion was conducted until achieving agreement between the researchers based on the literature sources.

To provide a deeper understanding of how the concepts and relationships were treated, interpreted, and inferred, we offer a visual example in the following paragraphs. We have selected the concepts *Service*, *Service package*, and *Elementary service* to illustrate how the treatment process proceeded as a continuation of the example provided in Sect. 3.2. If the concepts and relationships resulting from Sect. 3.2 (see Tables 2 and 3) were graphically depicted, it would result in the diagram illustrated in Fig. 9.

First, it is necessary to analyze whether the concepts refer to artifacts, low-level concepts, instances of a concept, or abstract concepts. In this case, all three concepts are adequate for the abstraction level desired in the final concept map. Therefore, no concepts were excluded. Then, we removed unnecessary relationships and incompatible relationships, as described in the following paragraphs.

The first set of relationships that should be analyzed comprehends R00388 (Elementary service is part of Service package), R00928 (Service package is part of Service package), R00435 (Service package is part of Service), R00386 (Service is part of Service package). First of all, Elementary services and Service packages are possible types of Services. Thus, the relationship “*Service is part of Service package*” (R00386) already inherits the relationships “*Elementary*

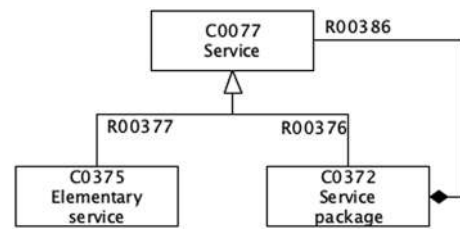


Fig. 10 Interconnection regarding the concepts *Service*, *Service package*, and *Elementary service* after treatment, interpretation, and inference

service is part of Service package” (R00388) and “*Service package is part of Service package*” (R00928). Therefore, the relationship R00386 is kept while R00388 and R00928 are excluded.

Some relationships in Fig. 9 are also incompatible. As defined by Shen et al. (2012), an *Elementary service* “is the smallest service unit that, from a commercial point of view, can be meaningfully offered to customers.” Therefore, an *Elementary service* could not be part of another *Elementary service*. Knowing that a *Service* may be an *Elementary service* or a *Service package*, it is incompatible with this definition to state that a “*Service is part of Service*” (R00463) or “*Service package is part of Service*” (R00435). Otherwise, we would be stating that a *Service package* may be part of an *Elementary service*. Therefore, the relationships R00463 and R00435 were deleted due to incompatibility. Finally, while relationship R00376 states that “*Service is superclass of Service package*,” the relationship R00847 states the exact opposite, i.e., “*Service package is superclass of Service*.” To identify which of the relationships will be employed in the concept map, we retrieved their sources. As the reader may confirm in Table 3, R00376 is recommended by three of the documents, while only one source prescribes R00847. Therefore, R00847 was excluded due to incompatibility.

Finally, no missing relationships were identified to be added in this scope nor bridge classes that should be deleted. Thus, the final interconnection regarding the concepts *Service*, *Service package*, and *Elementary service* is illustrated in Fig. 10.

3.4 Completeness verification against retrospective PSS design cases

The last step of the methodology regards verifying the resulting concept map’s completeness compared to the concepts that compose artifacts resulting from the initial phases of PSS design or their relationships. As mentioned previously, many information structures from the literature represent partial perspectives of the PSS, and all of them have an incomplete set of concepts and relationships. Even though

their combination might enhance completeness, some concepts and relationships might still be missing. The strategy employed for verifying the concept map completeness was to check it against formal project documentation from two retrospective cases conducted by the researchers previously: Case A and Case B.

Case A regards the servitization of diagnostic imaging equipment currently manufactured and sold by a health equipment manufacturer (we omit details about the company and the resulting solution for confidentiality reasons), with the Integrated Engineering Group's support, from the University of São Paulo. In this project, the company aimed to servitize its business model, moving from selling the equipment towards offering it as a service. This process covered the initial design phases until creating the solution's architecture, and it followed the servitization methodology proposed by Rozenfeld et al. (2018). For further information about this case study, please refer to the publication of Pieroni et al. (2016). One of the authors of this paper actively participated in this design process, having access to all versions of all created artifacts. Only the final versions of the artifacts resulting from the initial phases of the design process were selected, i.e., management reports or documents from other processes that run parallel to the PSS design were not used. Thus, 39 artifacts were considered in the analysis of Case A.

Case B regards the ELSA project, a research project at the Fraunhofer Institute for Production Systems and Design Technology (IPK). In this project, the team developed a robotic entity combined with digital services to simulate the smart product engineering complexity. The solution may be characterized as a product-oriented PSS, as proposed by Tukker's (2004) typology. ELSA was conceived to fulfill two main tasks: welcoming and guiding guests and watering plants in offices. The process covered all initial phases of the development until establishing the solution's architecture, following an approach inspired by the V model (VDI-Richtlinie 1993). Similar to Case A, one author of this paper also actively participated in the design process, having access to all versions of all created artifacts, filtered according to the same criteria employed in Case A. In total, 37 artifacts were included for analysis.

All selected artifacts were submitted to content analysis (Bardin 2013), which two researchers initially conducted to ensure a systematic procedure and, subsequently, one researcher. The analysis is illustrated with an example in Fig. 11 (for confidentiality proposes, we altered the real content from the artifact, just keeping its structure) and elucidated in the following paragraphs.

First, we tagged all pieces of information in each artifact (e.g., a simplified and modified version of an empathy map of Case A in Fig. 11) according to the concepts of the concept map, i.e., they were tagged as instances

of a given concept. Figure 11 shows that the identified concepts were *Stakeholder*, *Stakeholder's activity lifecycle*, *Need*, *Desire*, *Problem*, and *Experience*, all of which already existed in the concept map as extracted from the literature. However, in the analysis from other artifacts, whenever no concept was pertinent to a piece of information, a new concept was created to complement the concept map. Following the procedure proposed in Sect. 3.2, each new concept received an ID associated with its title, definition, and source (i.e., Case A or Case B). Second, we have analyzed the relationships among the tagged concepts in each artifact. If the relationships in the concept map that resulted from the literature could not cover all direct relationships among the pieces of information in the artifact, new relationships were created. In the example, three relationships were added to the concept map: "*Stakeholder performs Stakeholder's activity lifecycle*," "*Problem is part of 'Stakeholder performs Stakeholder activity lifecycle'*," "*Experience is part of 'Stakeholder performs Stakeholder's activity lifecycle'*." Following the process presented in Sect. 3.2, each new relationship was identified by an ID, its title, the input concept name, the output concept name (or output relationship name), and its source (i.e., Case A or Case B). As proposed in Sect. 3.3, we also verified whether the new relationships inherited other existing associations. If true, the new relationship was kept, and the inherited ones were deleted.

Similar to Sect. 3.3, after verifying the concept map for completeness, a second researcher verified all new concepts and relationships of the resulting concept map, aiming to identify possible missing concepts, relationships, and questionable connections. All points highlighted by the second researcher were discussed with the first researcher, retrieving the artifacts from the retrospective case studies and verifying the artifacts that originated those points. The discussion was conducted until achieving agreement between the researchers based on the artifacts.

We emphasize that this process cannot ensure the definite completeness of the concept map. The inclusion of new case studies may lead to new concepts and new relationships. Ensuring absolute completeness in a research project would require many applications in distinct contexts, which is not feasible in a research project. Those limitations are further discussed in Sect. 5.

4 Results

This section presents the resulting concept map, which structures the concepts that compose artifacts resulting from the initial phases of PSS design and how those concepts interrelate. In the first subsection (4.1 Overall

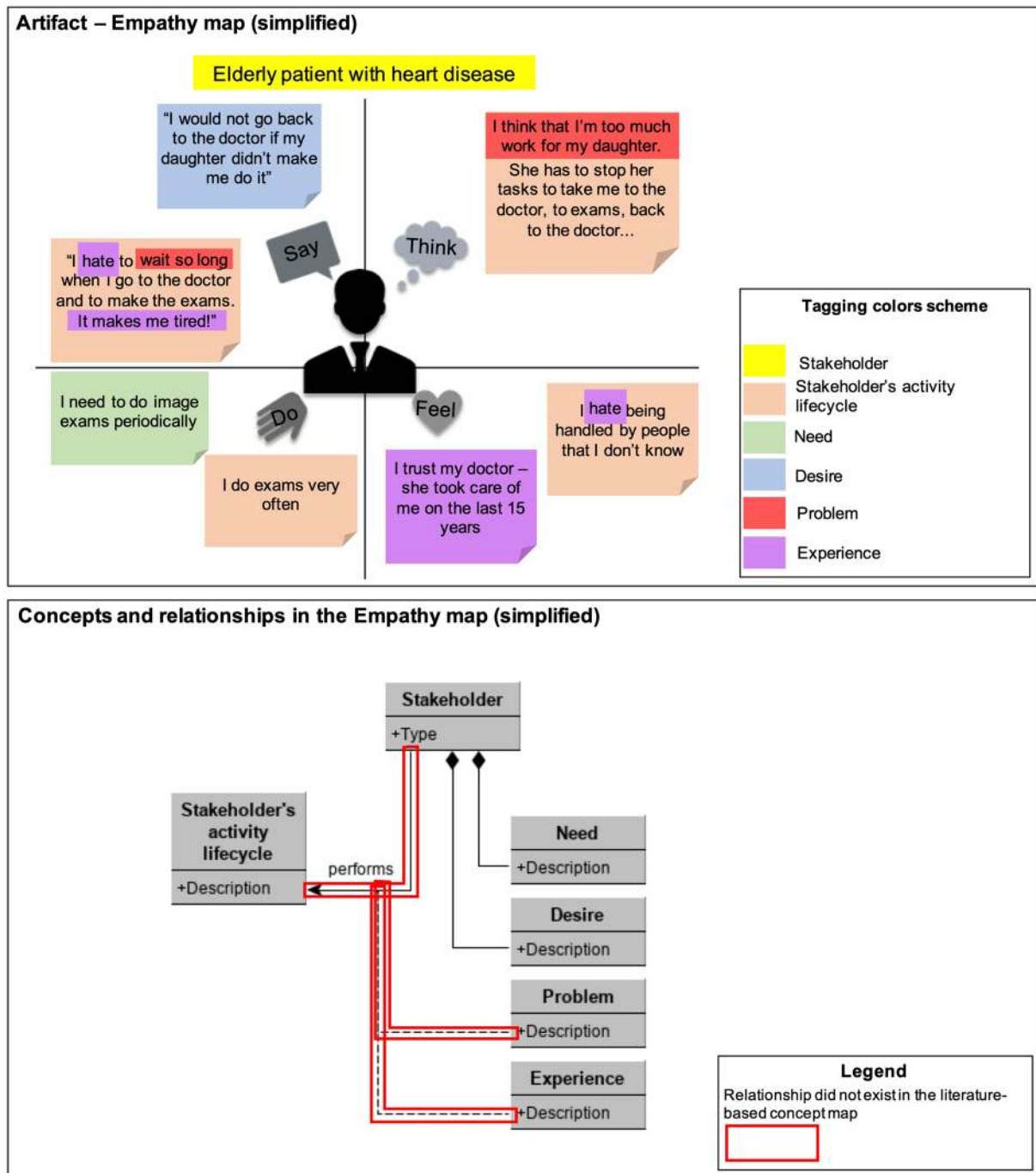


Fig. 11 Example of concepts and relationships extraction based on a simplified and modified Empathy Map from Case A (elaborated by the authors)

results), the reader may observe a quantitative view of the outcomes regarding each step of the methodology. The second subsection (4.2 Concept map) provides a set of views that depict the final concept map.

4.1 Overall results

The concept map proposed in this paper is composed of 143 concepts with 278 interrelationships. As summarized in Table 4, we identified 32 publications containing, in total, 671 concepts with 1109 relationships among themselves. From those, 519 concepts were not instantiable in design

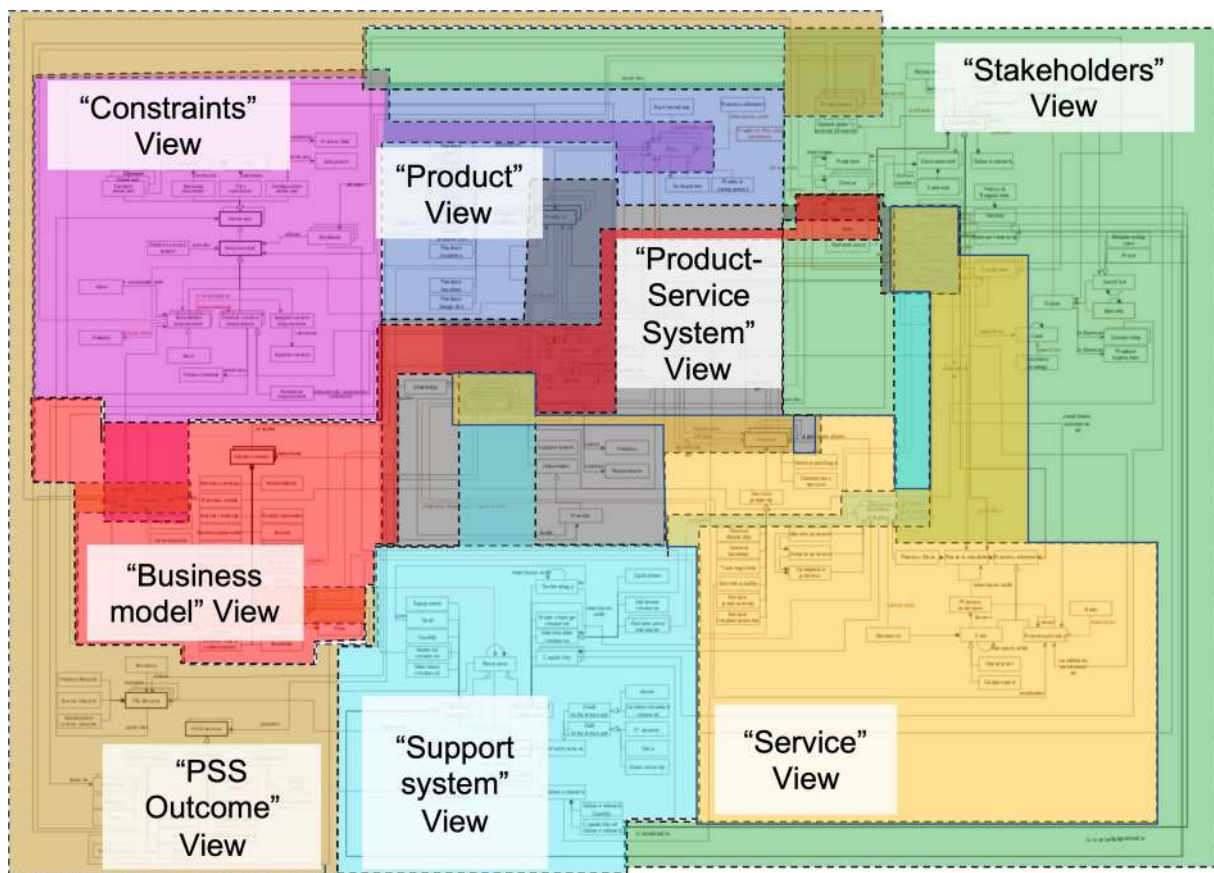


Fig. 12 Concept map and its partial views as depicted in this section (elaborated by the authors)

Table 4 Result parameters for each step of the methodology

| Results parameters | # |
|--|------|
| Step 1—Pre-analysis (Sect. 3.1) | |
| Included documents | 32 |
| Step 2—Material Exploration (Sect. 3.2) | |
| Extracted concepts | 671 |
| Extracted relationships | 1109 |
| Step 3—Treatment, interpretation, and inference (Sect. 3.3) | |
| Concepts excluded | 519 |
| Relationships excluded with the concepts | 743 |
| Bridge concepts | 11 |
| Relationships excluded due to bridge concepts | 53 |
| Relationships included due to bridge concepts | 38 |
| Unnecessary relationships | 165 |
| Missing relationships | 28 |
| Step 4—Completeness verification against retrospective cases of PSS design (Sect. 3.4) | |
| New concepts | 2 |
| New relationships | 63 |
| Final concept map | |
| Final concepts | 143 |
| Final relationships | 278 |

artifacts resulting from the initial phases of PSS design, i.e., they were instead artifacts, low-level concepts, instances, or abstract concepts. Therefore, they were excluded together with their 743 relationships. Furthermore, 11 concepts were bridge concepts and were removed, leading to the exclusion of 53 relationships, which were substituted by 38 relationships that superposed the bridge concepts. About 165 relationships were unnecessary since other relationships already covered them, while 28 missing relationships were identified and included. The verification of the concept map extracted from the literature against project documentation from two retrospective PSS design cases derived two new concepts and 63 new relationships.

4.2 Concept map

The concept map has a complex interrelationship network. Dividing concept maps into smaller graphic views enhances their usability (Gómez-Gauchía et al. 2004). Therefore, in this paper, the concept map was split into eight views (see Figs. 13, 14, 15, 16, 17, 18, 19 and 20) to allow its visualization, as conceptually illustrated in Fig. 12. We also provide the resulting concept map with a full list of concepts,

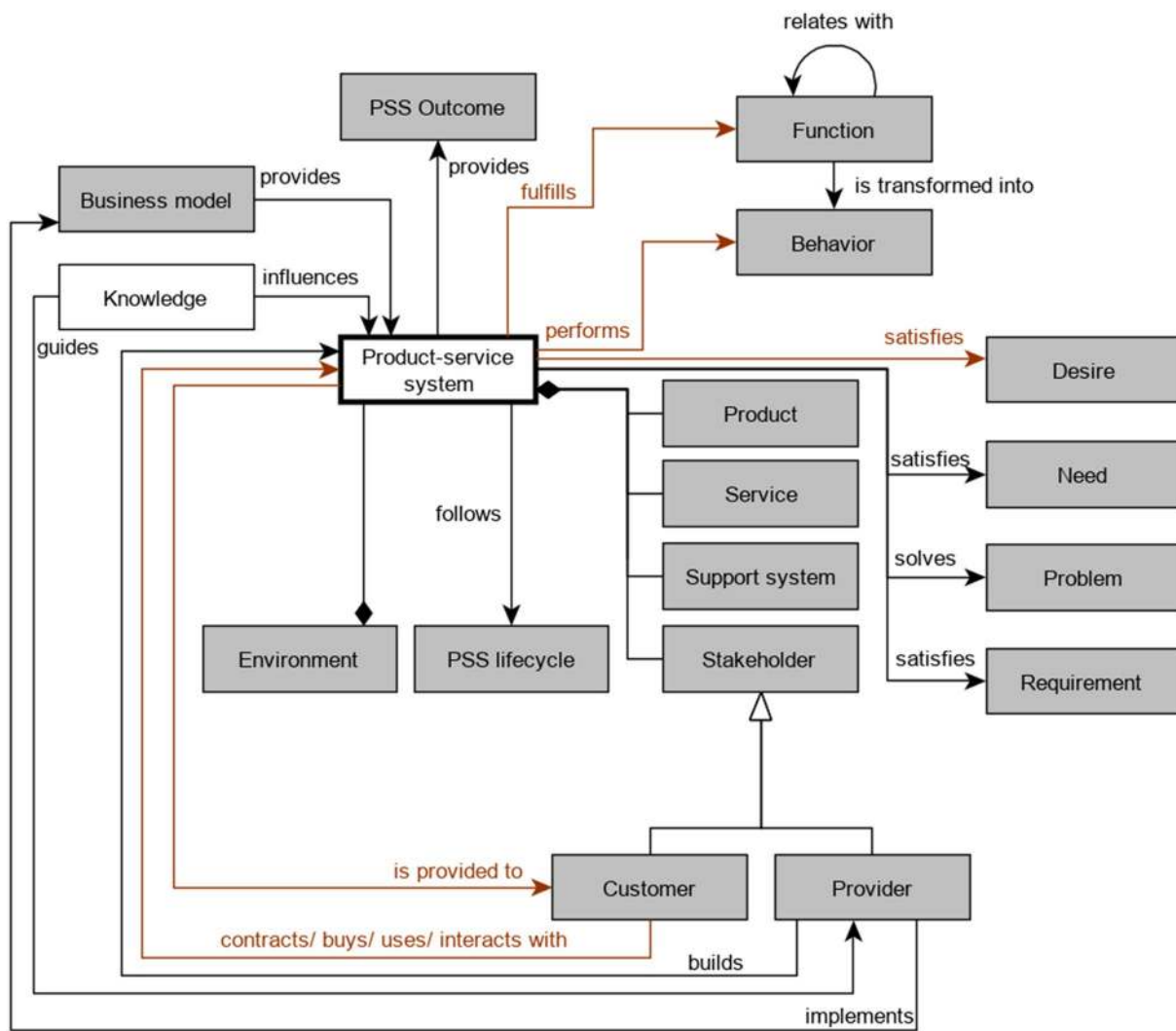


Fig. 13 Concepts that are directly connected to “Product-service system” in the concept map (elaborated by the authors)

definitions, and relationships in a workbook provided as supplementary electronic material. Before detailing the views, we reinforce that conceiving a PSS requires developing a system composed of products, services, and an entire support system. Therefore, some concepts in the concept map may also apply to other types of design processes (e.g., product design, service design, among others). However, the concept map as a whole focuses on PSS design. The information structures that served as a source for the concept map originated from the PSS and servitization literature.

Each view concentrates on the connections of a specific concept. For example, Fig. 13 presents concepts that are directly connected to the concept “Product-service system.” In each view, the concept where the focus lies is highlighted with a thicker black border. Since the views presented in this section represent partial views of the concept map, a concept illustrated in a view may have some of its relationships

omitted to enhance visualization. We employed a color schematic to identify which concepts have all their direct relationships depicted in that specific view (white boxes) or do not (gray boxes). Concepts are only highlighted in gray in the views when one or more of their direct relationships are omitted. Furthermore, all concepts and relationships added during completeness verification (i.e., through analysis of retrospective case studies described in Sect. 3.4) were highlighted in brown borders. The views employ the UML notation for class diagrams, which may be retrieved in Sect. 3 (Fig. 3). Throughout the text, we provide the definition only for terms that are not recurrently used in the PSS literature, illustrating them based on a bike-sharing example. However, it is possible to access a complete glossary in the workbook provided as supplementary electronic material. The first view, depicted in Fig. 13, portrays the concepts directly connected to *Product-service system* in the concept map.

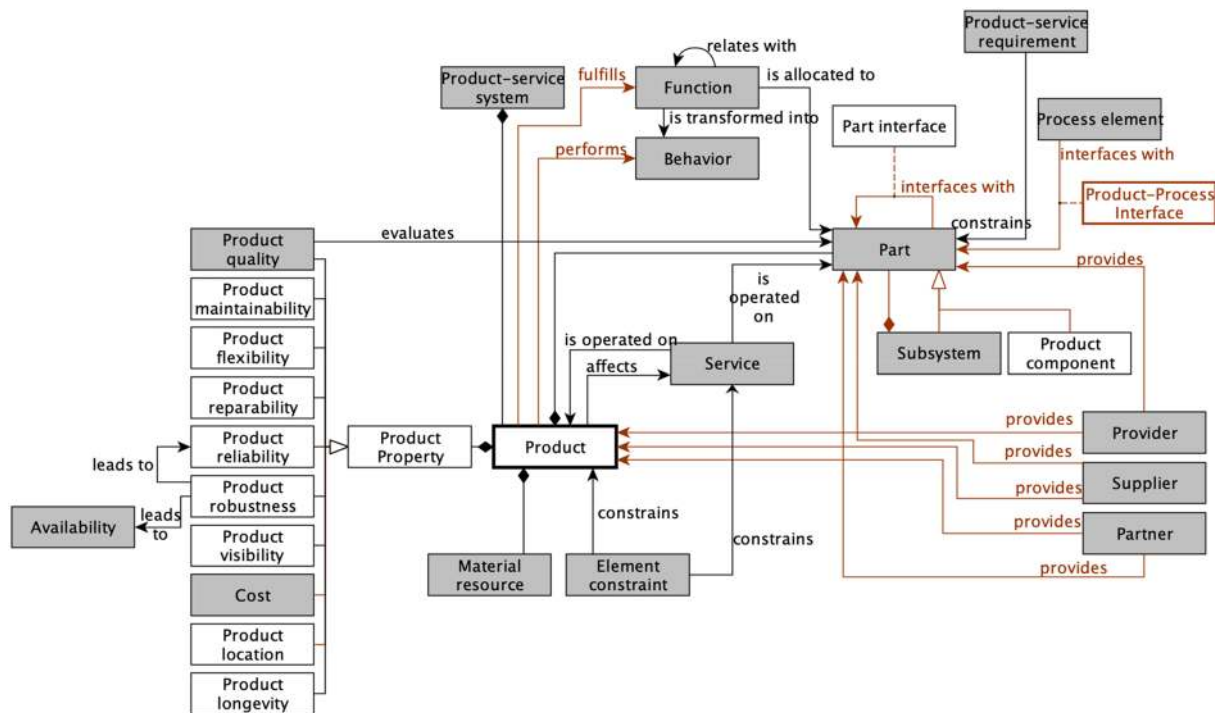


Fig. 14 Concepts that are directly connected to “Product” in the concept map (elaborated by the authors)

In a similar structure of the Function-Behavior-Structure (FBS) framework (Gero 1990), Fig. 13 illustrates that a PSS (represented under the combination of *Product*, *Service*, and *Support system* in an analogy to the “Structure” of the FBS framework) fulfills a *Function* (i.e., the purposes that the solution must satisfy, which may interrelate. E.g., in a bike-sharing solution, one of the *Functions* that must be fulfilled is “Track available bicycles”). In the design process, the *Functions* are transformed into *Behaviors* (i.e., how the structure achieves its *Functions* (Kan and Gero 2009). E.g., one of the possible *Behaviors* to fulfill the *Function* “Track available bicycles” is “Provide interface with geolocation of available bicycles.”). The *Behaviors* will be performed by the selected structure (again, the PSS as a combination of *Product*, *Service*, and *Support system*. E.g., the structure to fulfill the *Behavior* “Provide interface with geolocation of available bicycles” may require a GPS on the bike, an application to provide the appropriate interface, a stable GPS communication channel, among others). In this view, both *Function* and *Behavior* consider the system perspective of a PSS, i.e., what *Functions* the PSS fulfills as a system and what *Behaviors* it performs as a system. The figure also depicts two critical dimensions that are consistently cited in the PSS literature: the *PSS lifecycle*, followed by the PSS, and the related *Stakeholders*, which become part of the PSS. As pointed out previously, the *Stakeholders* and many other concepts are filled in gray because not all

their relationships are illustrated in Fig. 13. Figure 13 lays out only two *Stakeholders*: the *Provider* and the *Customer*. The *Provider* builds the PSS and implements a *Business model*, which provides the PSS. The *Customer*, on the other hand, contracts, buys, uses, or just interacts with the PSS provided to her/him (depending on the *Business model* in which the *Provider* offers the solution).

The *Knowledge* available in the PSS design context guides the *Provider* on creating a solution and influences the resulting PSS. Another meaningful dimension influencing PSS design is the *Environment* of which the PSS will be part. During PSS design, the PSS may also be seen through its intent perspective, which aims to satisfy the *Stakeholders’ Needs and Desires*, solve their *Problems*, and satisfy the established *Requirements*. Finally, implementing a PSS may generate desired and undesired *PSS Outcomes*, which will be further detailed in the other views. We highlight that the readers should carefully interpret gray boxes in all views of the concept map (including Fig. 13). For example, a reader familiar with the FBS framework (Gero 1990) might see Fig. 13 and ask why there is no relationship between *Requirements* and *Function* or *Behavior*. This particular relationship is omitted in Fig. 13 but illustrated in Fig. 20.

The view illustrated in Fig. 14 aims to detail the concept *Product*. Like Fig. 13, Fig. 14 depicts that the *Product* also fulfills *Functions* and performs *Behaviors*, just like the PSS does. However, in the design context, the *Functions* and *Behaviors* specifically related to the *Product* may be fulfilled

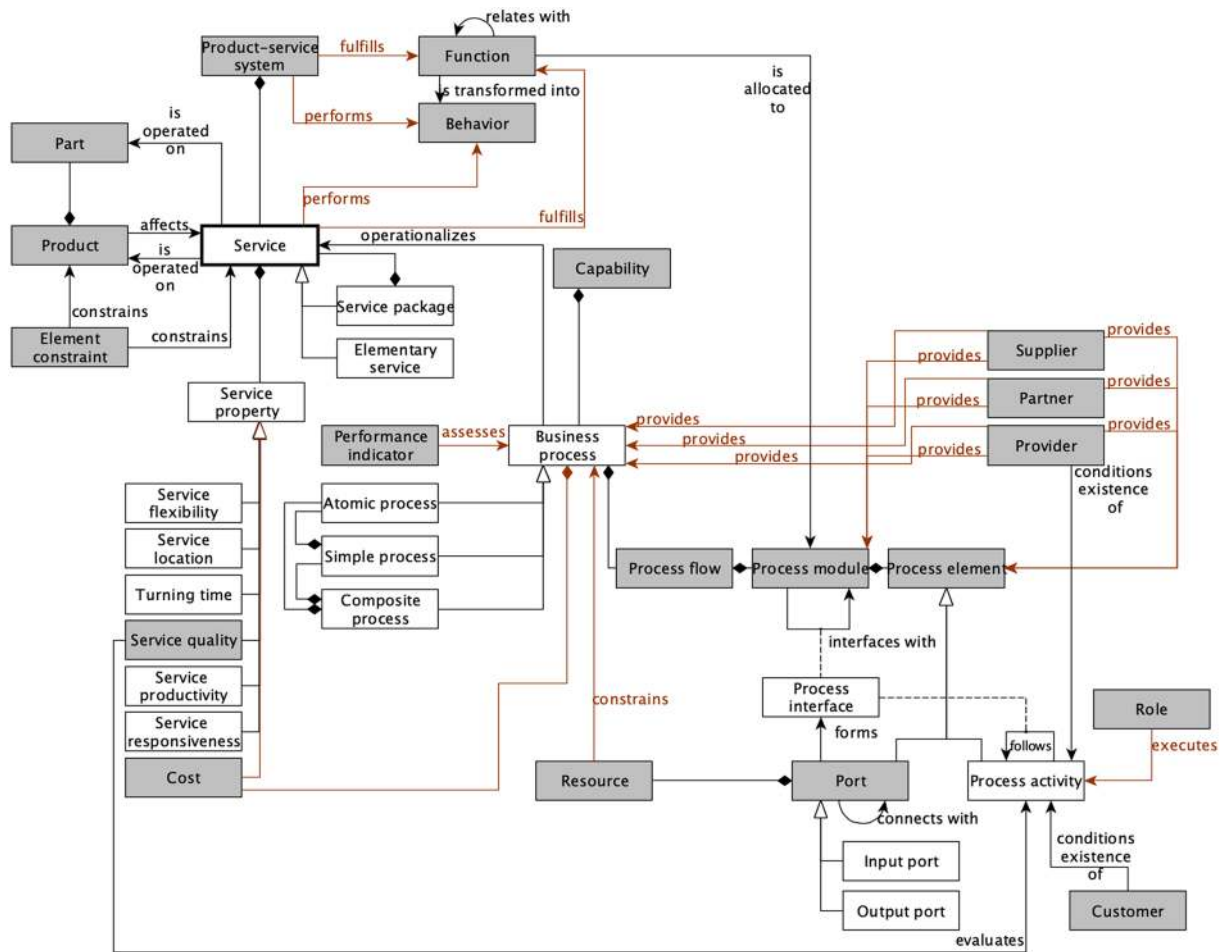


Fig. 15 Concepts that are directly connected to “Service” in the concept map (elaborated by the authors)

without the remaining PSS elements. The structure of the *Product* is composed of two possible generic entities: *Parts*, to which the product *Functions* are allocated, and *Material resources* (i.e., possible necessary supplies and spare parts during its lifecycle. E.g., spare parts for repairing damaged bicycles during maintenance). A *Part* may refer to distinct levels of the *Product’s* structure, such as a single *Product component* or a *Subsystem*. *Product components* refer to the most elementary parts of a *Product* (E.g., the tire of a shared bike) (Shen et al. 2010). A *Subsystem* concerns a group of *Product components* and/or other *Subsystems*, whose operation does not depend on other *Parts* (e.g., the entire lock system in a shared bike) (INCOSE 2015). Still composing the *Product’s* structure, we observe that the *Parts* have interfaces among themselves, represented by the concept *Part interface* (e.g., the lock system will lock the tire in a shared bicycle whenever a client is not using it).

Another perspective of the *Product* refers to the *Product properties*, which may be relevant to provide a feasible and successful PSS. Those properties are *Product quality* (which evaluates the *Parts*), *Product maintainability*, *Product*

flexibility, *Product reparability*, *Product robustness* (which leads to *Product reliability* and *Availability*), *Product visibility*, *Cost*, *Product location*, and *Product longevity*.

Different sources may provide a *Product* and its *Parts*, such as the *Provider* itself, a *Partner*, or a *Supplier*. Finally, there are relationships between the *Products* and the *Services*. Overall, the *Product* affects the *Service*, while the *Service* is operated on the *Product* or its *Parts*. On an architectural level, the *Process Elements* interface with the *Parts*, creating the concept *Product-Process Interface* (e.g., in a bike-sharing solution, the process of bicycle release is initiated by the customer in the software application and finished upon payment approval. This process will interface with the lock system, which will liberate the bike). Another concept illustrated as part of *Products* and *Services* in Fig. 14 (see also Fig. 15) is the *Element constraints*, which constrain the configuration of a solution. This constraint establishes which *Products* and *Services* are essential, optional, and how many are allowed to configure a specific PSS solution (e.g., a bike-sharing solution may provide a subscription service package for

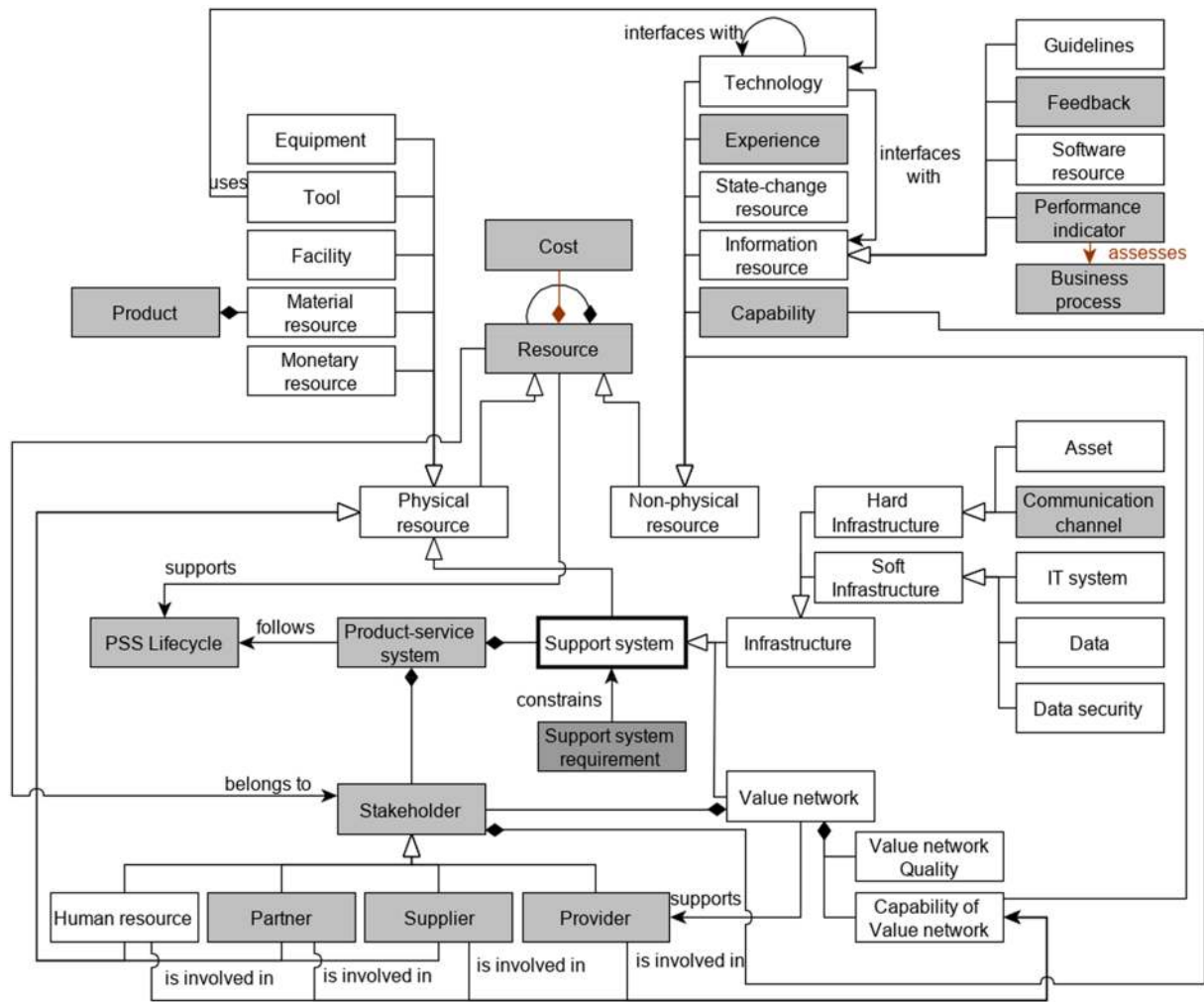


Fig. 16 Concepts that are directly connected to “Support system” in the concept map (elaborated by the authors)

customers, including access to safety equipment. However, access to safety equipment would be optional, while bike-sharing per se is the essential service) (Shen et al. 2012). Even though this information is relevant for implementing a *PSS* (after the early phases), it should be conceptually defined in the early phases of *PSS* design. Otherwise, it can cause meaningful impacts in later stages.

The view illustrated in Fig. 15 depicts the concepts that are directly connected to *Service*. It shows that a *Service* may be subclassified into two classes: *Elementary service* [the smallest service unit that can be offered to the *Customer*, such as paying per minute for a shared bike (Shen et al. 2012)] and *Service package* (a combination of *Elementary services* and/or other *Service packages*. E.g., a subscription package including unlimited use of the bicycles and access to safety equipment). Similar to the *Product*, the *Service* also fulfills *Functions* and performs *Behaviors*.

The *Business Process* operationalizes the *Services*, i.e., a *Service* is executed through a set of end-to-end activities

that delivers value to the customers. In turn, *Business Processes* may be subclassified as *Atomic processes* (cannot be subdivided, such as the payment process in a bike-sharing solution), *Simple processes* (composed of *Atomic processes*, such as the bicycle release process in a bike-sharing solution, which includes processing the user data and conducting the payment (both *Atomic processes*)), and *Composite processes* (which is composed of *Atomic processes*, *Simple processes* and/or even other *Composite processes*, such as the entire process of using the shared bike, which includes the bicycle releasing, time tracking during bicycle usage, and the bicycle locking). The structure of a *Business Process* is composed of a *Process flow* (a group of continuous activities), which consists of *Process modules* (e.g., the *Business Process* “Payment” may be performed through a set of *Process modules*, such as “Payment order,” “Payment processing,” “Invoice processing,” and “Payment confirmation”) (Wang et al. 2014). In turn, the *Process modules* are independent sets of loose coupling process steps (Wang et al. 2014), which are

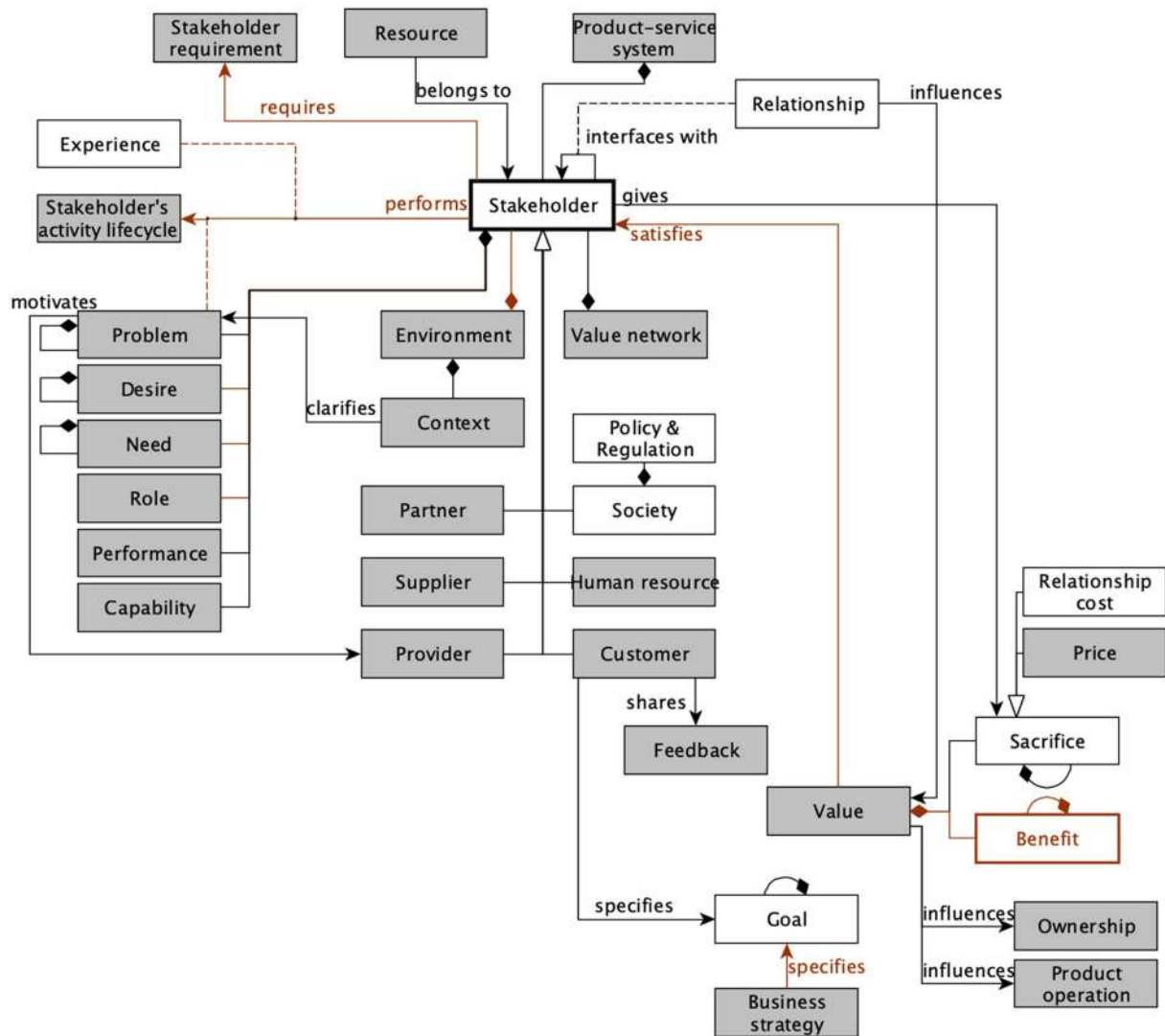


Fig. 17 Concepts that are directly connected to “Stakeholder” in the concept map (elaborated by the authors)

constituted of *Process elements*. *Process elements* may be *Process activities* or *Ports*. A *Process activity* is an action performed in the *Business Process*, which follows and is followed by other *Process activities*. A *Port* is necessary to connect two *Process activities* or two *Process modules* (Shen et al. 2010; Dong et al. 2011), being composed of *Resources*. It may be something required to perform a *Process activity* (*Input port*) or resulting from a *Process activity* (*Output port*). Therefore, a *Port* may connect with other *Ports*, and it forms the *Process interface*, which is the class comprising the interfaces between *Process modules* or *Process activities*. For example, consider the *Business Process* “Payment” encompassing a *Process module* “Invoice processing.” This *module* will be composed of a set of *Process activities*, such as “Emit invoice,” “Pre-process invoice,” “Approve invoice,” and “Document invoice.” The *Activity* “Document invoice” requires at least two *Input ports*: the invoice (which is an

Output port of the previous *activity*) and a storage system (a necessary *Resource*).

Three different *Stakeholders* can provide a *Business Process*. The *Provider* itself may provide the *Process activities*, *Process modules*, or the entire *Business Process*. However, *Partners* and *Suppliers* may also provide a *Business Process* or its structural elements, together or separately. Even though a *Provider*, a *Partner*, or a *Supplier* may be responsible for delivering a *Process activity*, executing it is under the responsibility of a specific *Role*, which may be attributed to any of the *Stakeholders*. Furthermore, a process is an intangible artifact. Therefore, the *Provider* and the *Customer* (in the *Service* context) condition the existence of a *Process activity*.

When the *Business Process* is implemented, it has a given *Cost*. It also becomes part of the company’s *Capability* or part of the *Capability* of the *Stakeholder* who is responsible

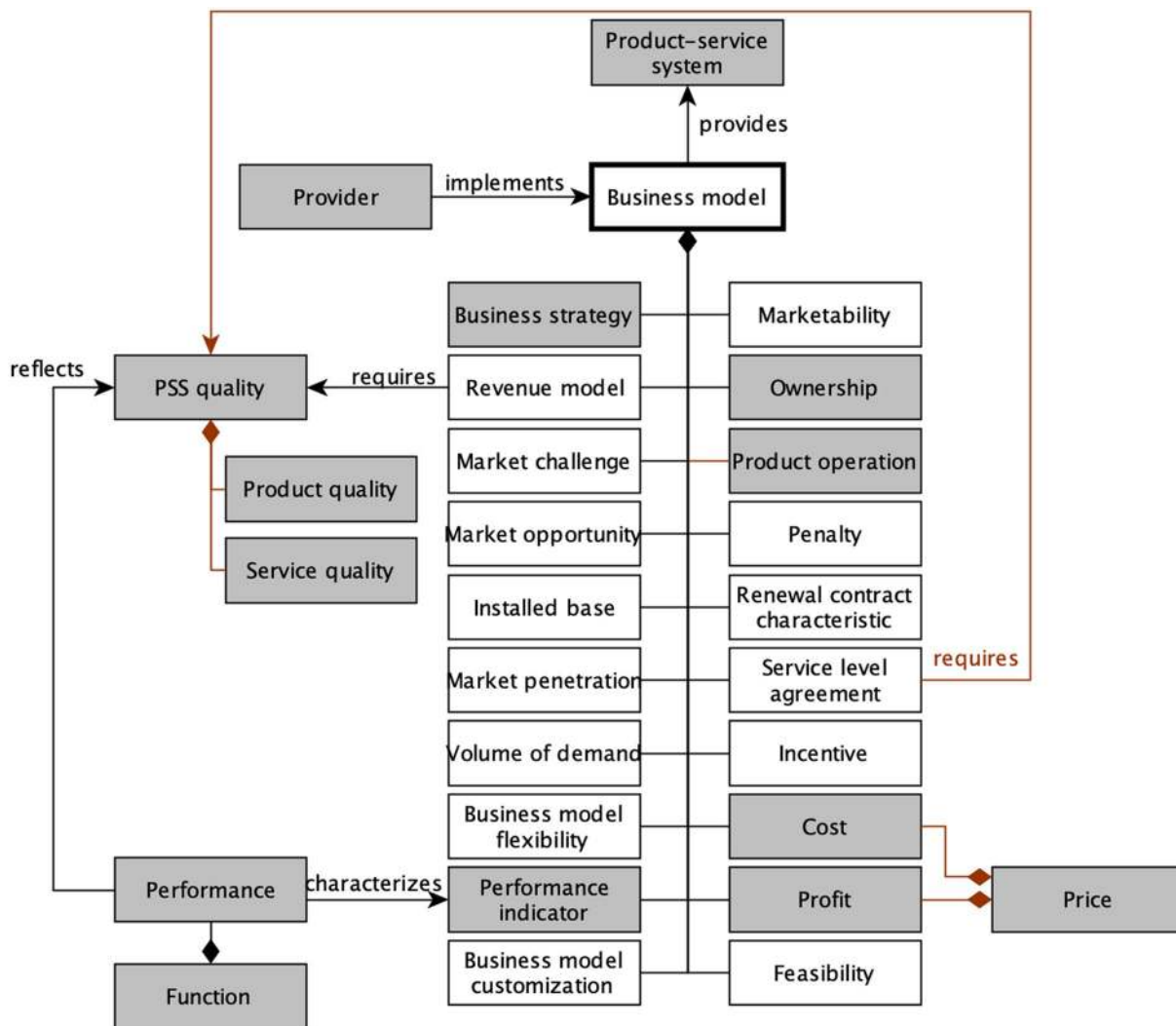


Fig. 18 Concepts that are directly connected to “Business model” in the concept map (elaborated by the authors)

for providing it. Each *Business Process* is constrained by the available *Resources* and is assessed by pre-defined *Performance indicators*. Finally, some *Service properties* influence the feasibility and success probability of offering a *PSS*, such as *Service flexibility*, *Service location*, *Turning time*, *Service quality* (which evaluates the *Process activity*), *Service productivity*, *Service responsiveness*, and *Cost*.

The next element that is part of the *PSS* structure is the *Support system*, illustrated with its related concepts in Fig. 16. Figure 16 shows the *Support system* being sub-classified as *Infrastructure* and *Value network*. The *Value network* is a people perspective of the *Support system*, composed of *Stakeholders* to support the *Provider*. When implemented, it has relevant properties for the *PSS* implementation, such as the *Value network quality* and the *Capability of the value network*, in which *Human resources*,

Partners, *Suppliers*, and the *Provider* are involved. The *Infrastructure* is a material (*Hard infrastructure*) and data (*Soft infrastructure*) perspective of the *Support system*. The *Hard infrastructure* comprehends *Assets* and *Communication channels*, while the *Soft infrastructure* covers the entire *IT system*, *Data*, and *Data security*. Together with all other *Requirements* and *Constraints* that constrain many elements of the solution (see Fig. 20), the *Support system* is constrained by the *Support system requirements*.

The *Support system* is a *Resource* – more specifically, a *Physical resource*. In general, *Resources* belong to one or more *Stakeholders* and are useful for supporting the entire *PSS Lifecycle*. A *Resource* may be a combination of *Resources*, and each *Resource* has a *Cost* attributed to it, influencing the total *Cost* of the solution. *Resources* may be subdivided into *Physical resources* and *Non-physical*

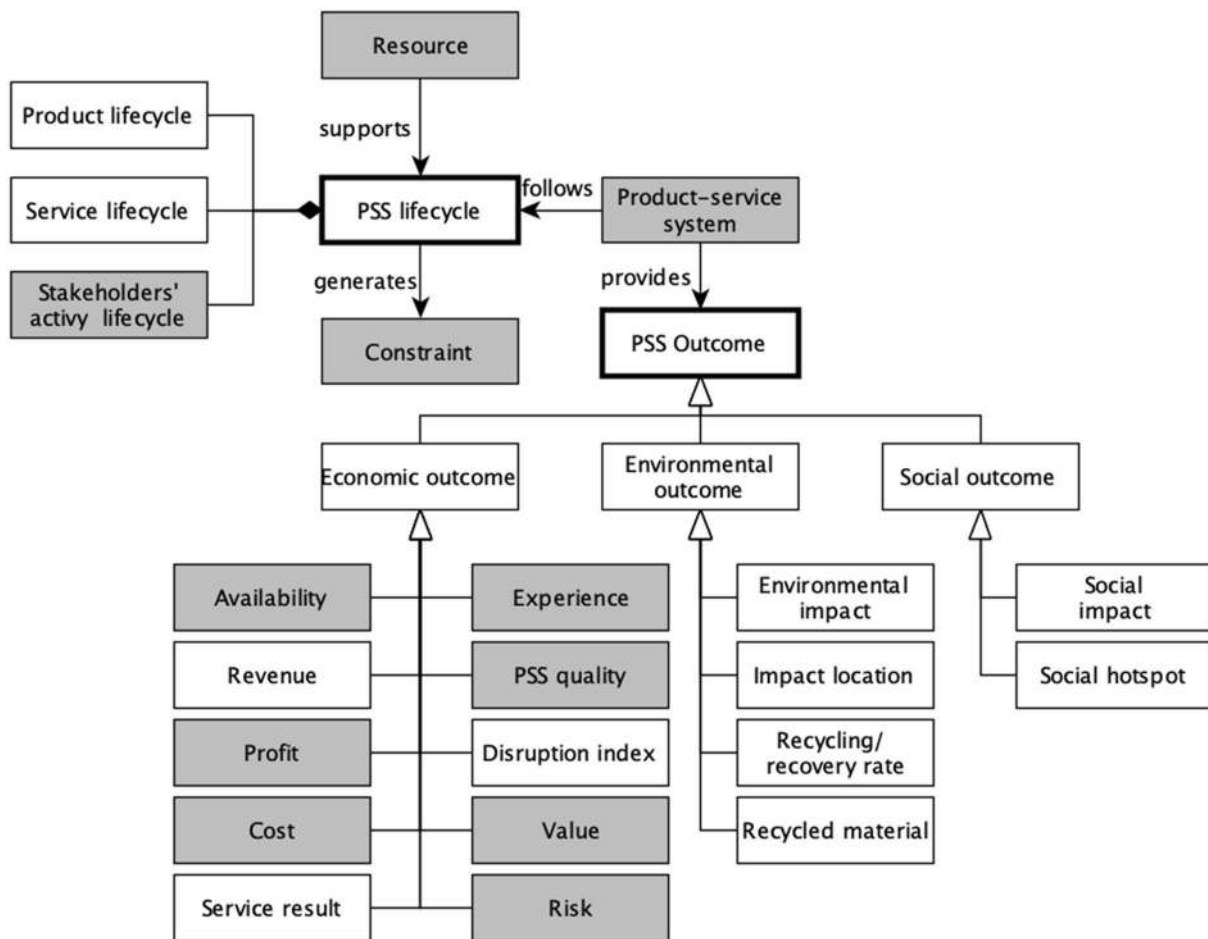


Fig. 19 Concepts that are directly connected to “PSS Outcome” and “PSS lifecycle” in the concept map (elaborated by the authors)

resources. Besides the Support system, the Physical resources encompass Equipment, Tools (which may use developed Technologies), Facilities, Monetary resources, and Material resources (which are employed to construct the Product, becoming part of it). The Non-physical resources, on the other hand, encompass Technology (which has interfaces with other Technologies), the current Experience of the Stakeholders (with focus on the Customer), State-change resources [i.e., Resources capable of changing a current state of the Customer or other Stakeholder, such as a medication (Dong et al. 2011)], Information resources (such as Guidelines, Feedback from the ongoing and old projects, Software resources, and Performance indicators), the Capability of the involved Stakeholders, and the Capability of Value network.

Figure 17 illustrates the concepts that are directly connected to the Stakeholder. Six main Stakeholders are discretized in the concept map: Partner, Supplier, Provider, Human resource, Society, and Customer. We highlight that the class Society regards not only the people in society but also their Policies and Regulations.

In a human-centered perspective, all Stakeholders have Needs (which may be broken down into smaller Needs), Desires (which may be composed of other Desires), and Problems (which may consist of other Problems and, usually, motivate the Provider). Those properties will reflect the Stakeholders’ requirements, which are required by the Stakeholders (for further detailing of the Requirements dimension, see Fig. 20). In the Stakeholders’ activity lifecycle, the Stakeholders perform many activities, from which they derive a given Experience that may be positive or negative. It is also possible to identify Problems in the Stakeholders’ activity lifecycle, which are clarified by the Context. The Context is part of the Environment, where the Stakeholders are embedded. Another concept, which is very highlighted in the human-centered design and the PSS literature, is the Value. In the literature, Value is mostly associated with the Customer, but it should be conceived to satisfy all Stakeholders, keeping in mind that their Relationships strongly influence it. A simple example is a product offered in a B2B business model being servitized to become B2C, focusing on the customers of their original clients. If the PSS provider

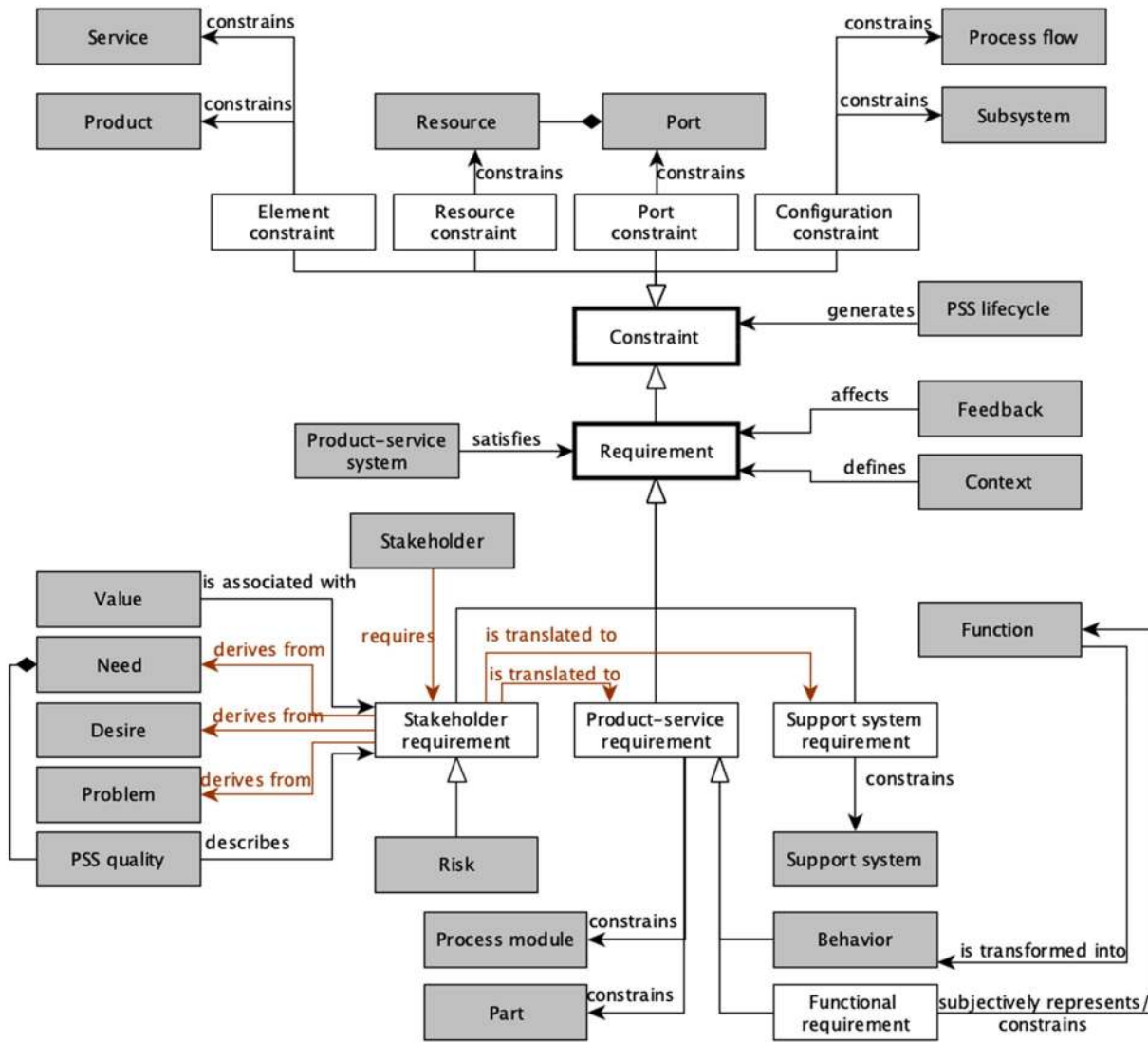


Fig. 20 Concepts that are directly connected to “Constraint” and “Requirement” in the concept map (elaborated by the authors)

decides to keep its initial business model concurrently, the customer from its first business model is the competitor from the new business model. Thus, the second business model will cannibalize the first. Building a partnership with the original clients would solve this problem (Pieroni et al. 2016). Therefore, the *Value* should be conceived with all *Stakeholders* in mind. Another typical example is developing an entire new PSS solution but not meeting the shareholders’ risk and profit expectations, which would lead to not implementing the solution.

Further detailing the concept *Value*, it is composed of a trade-off among the offered *Benefits* (which may be composed of more detailed *Benefits*) and the *Sacrifices* that *Stakeholders* may give (which may consist of other *Sacrifices*). Two main categories of *Sacrifices* are the *Price* to be paid and the *Relationship costs* [such as investments, time,

resources devoted to the solution, inconveniences, and other unpleasant feelings that the *PSS* may generate to any of the *Stakeholders* (Dong et al. 2011)]. While the *Customers* may have *Benefits* from a bike-sharing solution, such as not worrying about having their bikes stolen or about maintenance, they also have *Sacrifices*, such as being subject to a bicycle’s availability, which may not always be ensured. The *Value* also influences the product *Ownership* (i.e., whether it will be transferred to the *Customer* or kept by a *Partner* or the *Provider*). In a bike-sharing, for example, the *Provider* owns the *Product*) and the *Product operation*. For example, some companies that offer high-cost products in their PSS solutions do not transfer the *Ownership* to the *Customer* and also require that a *Partner* or the *Provider* itself operates the product. Even though the *Value* should satisfy the *Stakeholders*, the *Goal* of the design process (which may be divided

into sub-goals) should be specified based on the *Customer* and the *Business strategy*. Finally, the *Stakeholders* also have some properties capable of strongly influencing the PSS design process, such as the *Roles* they play in the *Business process*, their *Performance*, and their *Capabilities*.

As explained in Fig. 13, the *Business model* [i.e., the rationale for value creation, delivery, and capture (Diversity 2016)] allows the commercialization of a PSS. Figure 18 illustrates the concepts linked to the *Business model*.

The *Business model* is composed of 20 concepts: *Business strategy*, *Revenue model*, *Market challenge*, *Market opportunity*, *Installed base*,⁶ *Market penetration*, *Volume of demand*, *Business model flexibility*, *Performance indicator*, *Business model customization*, *Marketability*, *Ownership*, *Product operation*, *Penalty*, *Renewal contract characteristics*, *Service level agreements*, *Incentives*, *Cost*, *Profit*, and *Feasibility*. Some of those concepts interact with other concepts. The *Revenue model* and the *Service level agreements*, for example, require *PSS quality*, which is composed of *Product quality* and *Service quality*. *PSS quality* is a very relevant concept in the PSS design since it may influence the entire solution's feasibility. For example, customers contracting a pay-per-use product, like in the bike-sharing example, will provide higher revenues if the product requires less maintenance and repairs, for it will be longer available. A concept that reflects the *PSS quality* is the *Performance*, which characterizes the *Performance indicators* and is part of the *Function* regarding the PSS systemic perspective. Finally, the *Cost* and the *Profit* of the PSS commercialization will compose its *Price*. Some concepts of the *Business model* perspective are highly contractual, such as the *Service level agreements* that will be promised, the *Incentives* that will be proposed to attract customers, the *Ownership* of the *Product* and its *Operation*, and the *Penalties* for not fulfilling the contract.

A characteristic of this view differs from the other views shown so far. While the concepts from the other views are strongly interconnected, the reader may notice that the business model perspective is highly detached from the remaining concepts. Two reasons explain this particularity. First, distinct areas with different expertise are usually responsible for PSS design and business model design. Second, as a consequence of the first reason, most papers resulting from the systematic literature review are more design-related than business model-related, hardly ever pointing out concepts from both perspectives or interconnecting them. However, the business model perspective is still meaningfully connected with the design perspective through some of the

concepts. Therefore, it is fundamental to ensure the alignment of both perspectives during the PSS design.

In Fig. 19, the concepts directly connected to *PSS Outcome* and *PSS lifecycle* are presented. The *PSS lifecycle* comprises three concepts: *Product lifecycle*, *Service lifecycle*, and *Stakeholders' activities lifecycle*. The available *Resources* support the entire *PSS lifecycle*. Furthermore, the planned *PSS Lifecycle* will generate *Constraints* for the whole solution (see more details about *Constraints* in Fig. 20).

When implemented, the PSS provides many outcomes that should be planned already in the initial phases of PSS design. The *Economic outcomes* encompass the solution's *Availability*, *Revenues*, *Profit*, *Cost*, *Service results*, the new *Experience* provided to the *Stakeholders* (and, mainly, to the *Customer*), *PSS quality*, *Disruption index*,⁷ *Value*, and *Risks* (which should be planned and mitigated). Regarding the environmental perspective, the expected outcomes are the *Environmental impacts*, the *Impact location*, the *Recycling or recovery rates* of the *Product* and *Resources* employed in the PSS provision, and the *Recycled material* used to provide the *Product* or as a *Resource*. The *Social outcomes* are the *Social impacts* that the PSS provision will cause and *Social hotspots*⁸ that may be generated.

Finally, the concept map's last view regards the *Requirements* that a PSS must satisfy and further *Constraints*. This perspective is illustrated in Fig. 20. *Constraints* are imposed limitations that establish the values allowed on PSS solution properties to ensure a correct configuration (Dong et al. 2011; Shen et al. 2012). Those *Constraints* may derive from inequalities between what has to be done and the environment (Simon 1996), generally keeping the label "constraint." For example, every physical object is subject to gravity, and this is a physical constraint. There might also be constraints regarding the available resources for the design process, such as money or people. The *Constraints* derive from inequalities and limitations regarding the *PSS lifecycle*. However, *Constraints* may also be artificially imposed, establishing specifications for the solution based on customer needs and desires, performance metrics, among others (Sommerville 2011). In this case, they are labeled *Requirements*.

Requirements may be defined in different levels of abstraction (Sommerville 2011). *Stakeholders' requirements*

⁶ The *Installed base* is the consolidation of how many products are operating in the field, where they are located, and further relevant information regarding each product instance.

⁷ The *Disruption index* refers to critical events that occur during the PSS provision, such as faults (Annamalai et al. 2011).

⁸ *Social hotspot* are unit processes performed in the environment of a specific sector or community, in which it brings great opportunities to address local issues or high potential of negatively impacting the sector or community (Benoit et al. 2010; Benoit-Norris et al. 2012). For example, if the bike sharing solution is implemented in a community where transportation is compromised, it might have a positive social impact in this community.

are elicited in a non-structured format and employing the *Stakeholders'* language. A *Stakeholder requirement* may be derived from the identified *Needs, Desires, and Problems*, but it also may be stated directly by the *Stakeholders* themselves. One subclass of *Stakeholders' requirements* is the *Risks*, which should be planned and mitigated. The *Stakeholders' requirements* are associated with the *Value* that the *Stakeholders* will perceive in the solution, and the *PSS's* capacity to fulfill those *Requirements* will reflect on the *PSS quality*.

After eliciting the *Stakeholders' requirements*, they are translated into *Product-service requirements* and *Support system requirements*. The first will establish constraints to the *Process modules* and the product's *Parts*, being possibly divided into *Functional requirements* (which will guide the *Functions* that the *Product-service system* must provide) and *Behavior* (which will describe how the *Product-service system* must realize the *Function*). The second will establish constraints to the *Support system*. The *Requirements* depend on the development *Context*, possibly being affected by *Customers' Feedbacks*.

Further *Constraints* regard the following classes. As explained in Fig. 14, *Element constraints* constrain the configuration of a solution, establishing which *Products* and *Services* are essential, optional, and how many *Products/Services* are allowed for configuring an instance of a *PSS* solution (Shen et al. 2012). *Configuration constraints* regard the configuration that the solution may assume, constraining *Subsystems* and *Process flows*. They establish which *Subsystems/Process flows* enhance each other, support each other, are bundled (i.e., must be offered together), or substitute each other (Wang et al. 2014). Similarly, *Resource constraints* impose rules on the *Resources* depending on the *PSS* configuration, and *Port constraints* impose restrictions on the required *Ports* for distinct configurations of the *PSS*.

5 Conclusions and outlook

The main problem approached in this paper is the high likelihood of ignoring the creation of relevant information on the early phases of *PSS* design, which generates rework and lack of integration. Four challenges mainly cause this problem: lack of completeness and structure on service-related information due to the intangibility and heterogeneity of services; lack of integration among the *PSS* elements due to not considering its information requirements; dealing with a significant quantity and variety of knowledge; and ensuring completeness without limiting the flexibility of designers to select the methods and artifacts they intend to use. Properly planning and evaluating the artifacts that result from the initial phases of *PSS* is a practical approach to deal with those four challenges.

As a fundamental theoretical basis to support the creation of solutions that may assist adequate planning and evaluation of *PSS* design, this study proposed a concept map consolidating concepts that should be defined in the early phases of the design process before proceeding to detailed design, as well as their relationships. The research methodology followed data-acquisition based on a systematic literature review of classifications, taxonomies, ontologies, meta-models, and concept maps in the *PSS* and servitization field. For extracting the concepts and relationships, we employed the method of content analysis. Finally, the concept map was verified for completeness based on formal project documentation analysis from two retrospective cases. This paper's main result is a concept map composed of 143 concepts, which interconnect through 278 relationships. Only two concepts were added to the concept map during completeness verification against retrospective projects (1.4% of all concepts), indicating good coverage. On the other hand, before completeness verification, 28 missing relationships were identified (10% of all relationships). During completeness verification, additional 63 missing relationships were identified (22% of all relationships). This fact highlights that the literature has significant gaps when regarding relationships among concepts. It also indicates that future improvement in the concept map might be more tightly connected to adding new relationships rather than new concepts. Based on the results obtained, it is possible to conclude that the proposed concept map covers many gaps of other published *PSS* information structures in the literature, providing a more complete schematic of concepts and relationships regarding the early phases of *PSS* design.

The proposed concept map is a theoretical model rather than a prescriptive solution oriented to daily practice. It consists of a meaningful fundamental contribution both to practice and theory. The possibilities of how this concept map may contribute immediately and future research are depicted in Fig. 21. Some of the proposed contributions may initially employ parts of this paper's methodology since a similar process was realized during the creation of the concept map. Those contributions are filled in gray in Fig. 21. The contributions are explained in the following paragraphs, being cited with the letters in the upper left corner of their respective boxes in Fig. 21.

As an immediate contribution to practice, project managers may employ the concept map as a checklist for planning or verifying *PSS* design projects (contribution A, Fig. 21). The concepts and relationships are provided in the workbook as a supplementary electronic material and may be filtered as desired. In this application, each concept of the concept map is a type of information that may be included in the initial phases of the *PSS* design project. In each *PSS* design project, the project manager should define which concepts apply to that specific project, i.e., since the concept map

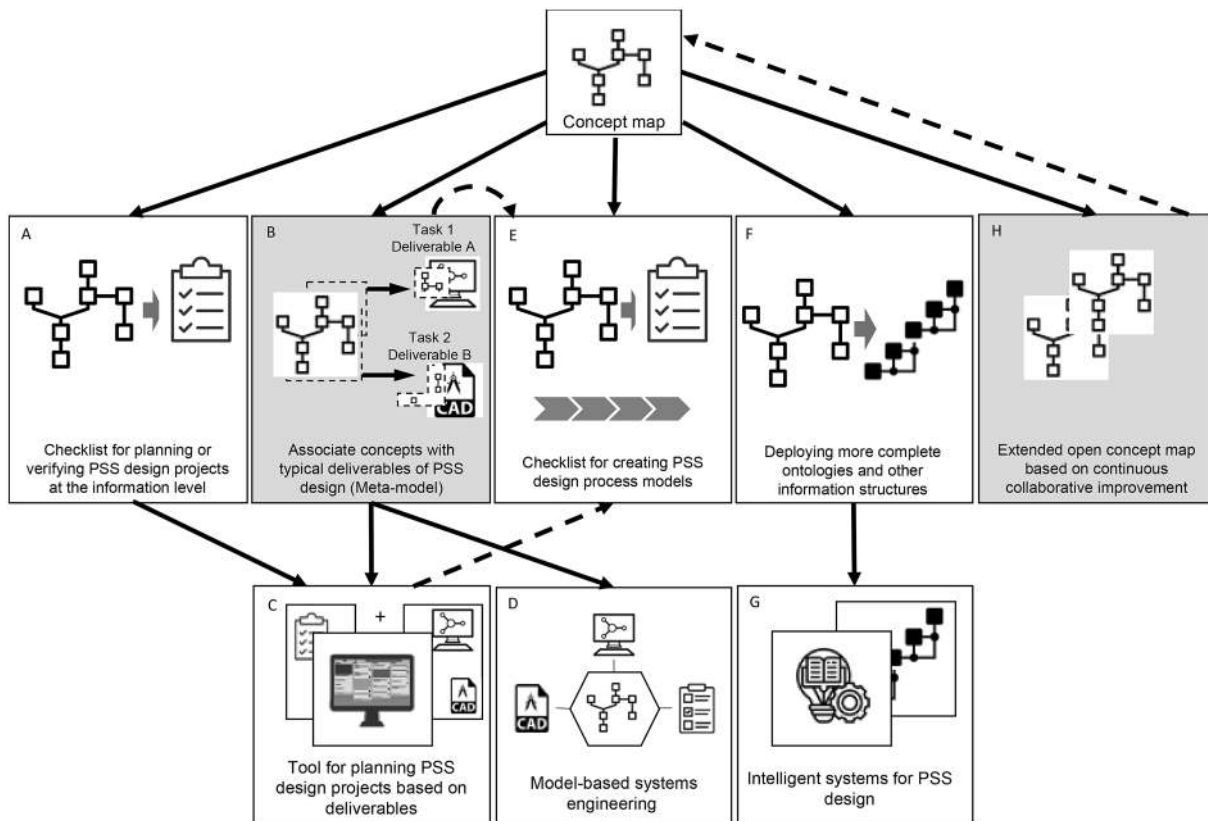


Fig. 21 Contributions of the concept map to theory and practice (elaborated by the authors)

contains a broad set of concepts, not every concept may apply to a given PSS design project. For example, suppose a product-oriented solution is to be created where no contract structures its provision (for example, products sold together with on-demand services of maintenance and repair). In that case, the concepts *Service level agreements* and *Renewal contract characteristics* are not necessary. The selected set of concepts can guide the project manager as a checklist to ensure that all required information in the initial phases of the PSS design project will be created through the planned tasks and deliverables. Furthermore, it is possible to employ this set of concepts as a checklist to support verification on each gate, i.e., checking which concepts were already defined up to that point.

Two main reasons may hinder the broad applicability of the concept map as a practical tool. First, the concept map depicts the information-level (content—e.g., process flow, process activities) of artifacts. However, project managers usually plan and verify design projects based on tasks (e.g., “draw service structure”) and artifacts (deliverables – e.g.,

“service blueprint”) (see Sect. 2.1). Therefore, to employ the concept map, the project manager should be able to bridge those two abstraction levels. E.g., when stating that the task “draw service structure” will generate the artifact “service blueprint” in the project plan, the project manager should understand that this artifact will provide detail regarding concepts such as “process flow” and “process activities.” Therefore, even though some practitioners may be capable of directly using the concept map, its broader application in practice may require establishing a communication bridge between those abstraction levels, i.e., correlating the information-level with the task/artifact level. Thus, as a continuation of this research, we propose contribution B (Fig. 21), which encompasses building this communication bridge by identifying typical deliverables of the PSS design, associating those deliverables to common tasks in PSS design process models, and identifying which concepts in the concept map are covered by each deliverable. This work is already in progress following an analysis method very similar to what is proposed in Sect. 3.4 to identify the concepts inside the

typical deliverables. This association will result in a holistic three-layered meta-model (concepts, artifacts, and tasks) to fulfill the communication bridge between both abstraction levels. The second reason that may hinder the concept map applicability is its complexity. Even if contribution B (Fig. 21) may ease its understandability, the meta-model will likely be very complex too. Thus, its broad use in daily practices may require its implementation as a software tool to allow visualization and selection of tasks and deliverables for planning the PSS design (see contribution C, Fig. 21). This tool should enable the visualization of what artifacts have overlaps of information and what artifacts complement each other. This tool should cover the deliverable-oriented planning of PSS design and encompass an automatic verification of which concepts in the concept map are not covered by the artifacts used in the early phases of the planned PSS design project. Furthermore, this tool should enable automatic filtering of concepts, artifacts, and tasks based on contextual factors (such as company size, innovation level, available tools, among others) to ensure that the meta-model will be tailored to the needs of different stakeholders.

Researchers in the field of model-based systems engineering with a focus on PSS may directly benefit from the meta-model described in contribution B (Fig. 21) to develop model-based systems engineering approaches based on traditional PSS design models (contribution D, Fig. 21). Researchers willing to employ new models to build model-based systems engineering approaches focusing on PSS may directly use the concept map, deriving their meta-models. This way, the relationships among concepts proposed in the concept map will reflect relationships among the models and facilitate the creation process. Furthermore, similar to contribution A (Fig. 21), the proposed concept map may also be employed by both researchers and practitioners as a checklist to create generic or specific PSS design process models (contribution E, Fig. 21). Due to the lack of a communication bridge and the concept map's complexity, this application may also benefit from the solutions proposed in contributions B and C (Fig. 21).

The concept map is also a possible basis for deploying PSS ontologies regarding the design process (contribution F, Fig. 21). Such ontologies are necessary to develop intelligent systems for PSS design regarding some support

processes, such as knowledge management and change management (contribution G, Fig. 21). For example, in change management regarding the project documentation, observing the artifacts as a set of interconnected concepts could help practitioners identify what documents are connected (and, therefore, might need revision) to a specific change that was performed.

Finally, this study's limitations encompass not ensuring the exhaustive completeness of the concept map since it was derived from a systematic literature review and may fail to cover all possible concepts and their relationships in PSS design artifacts. We verified the resulting concept map against artifacts from two retrospective PSS design projects to mitigate this limitation. However, additional retrospective analyses and future applications of the concept map may point out more missing concepts or relationships, possibly leading to modifications in the concept map. Furthermore, the concept map is available to the entire design community as a live editable file in open software in <https://flexmethod4innovation.com/collaborative-concept-map/>. We invite researchers who might identify missing relationships or concepts to contact us and provide their collaboration. This collaboration will lead to an extended version of this concept map (contribution H, Fig. 21). Every proposal will be integrated into the concept map following the methodology described in Sects. 3.2 and 3.3.

Another limitation of this paper regards the completeness verification, which does not evaluate the concept map in action (i.e., as a support for planning PSS design projects). Instead, the verification covers missing concepts and relationships only. This limitation is because, as described previously, the concept map is a fundamental theoretical basis rather than a prescriptive method or tool. Its verification in-action shall derive from future research after its deployment into a holistic meta-model.

Appendix 1

As described in this paper's methodology section, the documents that resulted from the systematic literature review are listed and shortly described in Table 5.

Table 5 Documents resulting from the systematic literature review

| Reference | Type | Goal |
|-----------------------------|----------------|--|
| Emili et al. (2016) | Classification | Classification regarding the most relevant dimensions of PSS based on distributed renewable energy |
| Cho and Lee (2011) | Taxonomy | Taxonomy for technology roadmaps in the service areas |
| Wang et al. (2019) | Ontology | Product, service, and condition ontologies to depict their in-context relationships |
| Correia et al. (2018) | Ontology | Ontology to facilitate knowledge management in collaborative PSS design |
| Hajimohammadi et al. (2017) | Ontology | A semantic model for PSS lifecycle management based on ontology |
| Marques et al. (2017) | Ontology | A reconfigurable PSS ontology enriched with knowledge models |
| Estrada and Romero (2016) | Ontology | A system quality attributes ontology to allow performance measurement of PSS functions |
| Zhu et al. (2015) | Ontology | A systematic strategy for PSS design with ontology-based knowledge representation |
| Vasanth et al. (2015) | Ontology | Summary of the most frequent classes of reviewed ontologies |
| Pagoropoulos et al. (2014) | Ontology | A PSS ontology for domain conceptualization |
| Wang et al. (2014) | Ontology | Meta-ontology for identifying customer demands on modular product-service |
| Shen et al. (2012) | Ontology | An ontology-based approach to represent and design a PSS configuration system |
| Dong et al. (2011) | Ontology | A meta-ontology for service product configuration |
| Annamalai et al. (2011) | Ontology | A PSS ontology to unify the PSS terminology and structure the PSS domain |
| Shen et al. (2010) | Ontology | An ontology-based approach for modeling Product Extension Services (servitization of products) |
| Kim et al. (2009) | Ontology | An ontological representation to support computer-aided tools and frameworks for PSS design |
| Letia and Marginean (2008) | Ontology | Needs and service offering ontologies in the context of client-provider collaboration for service bundling |
| Mien et al. (2005) | Ontology | Ontology for a sustainable development framework (IMPSS) |
| Anke (2019) | Meta-model | A meta-model for financial assessment of smart services |
| Medini and Boucher (2019) | Meta-model | A meta-model for developing a modeling language for PSS engineering |
| Maleki et al. (2018a) | Meta-model | A systems engineering-based semantic meta-model |
| Maleki et al. (2018b) | Meta-model | A meta-model, which proposes the definition and classification of interfaces in PSS architecture |
| Rizvi and Chew (2018) | Meta-model | A meta-model of the design process for cyber-physical PSS |
| Li et al. (2016) | Meta-model | A meta-model for modular structure data modeling |
| Boucher and Medini (2016) | Meta-model | A PSS meta-model for scenario modeling and analysis |
| McKay and Kundu (2014) | Meta-model | A meta-model for defining PSS service elements and relationships among product elements and service actors |
| Orellano et al. (2019) | Concept map | A comprehensive conceptual model, which integrates both strategic and operational perspectives of a business model |
| Verlaine (2017) | Concept map | A conceptual model of the key components and relationships of PSS solutions |
| Zhu et al. (2017) | Concept map | Concept map showing the interconnections among requirement, function, process, and structure |
| Diversity (2016) | Concept map | Concept map to provide a common understanding of PSS concepts |
| Goh et al. (2015) | Concept map | Concept map towards ontology to address uncertainty in estimating PSS cost |
| Komoto and Tomiyama (2009) | Concept map | Concept map of service CAD for PSS modeling |

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00163-021-00358-9>. **Funding** This work was supported by the São Paulo Research Foundation (FAPESP) under the processes 2017/12520-0 and 2018/09474-9 and by the German Research Foundation (DFG), project STA 1112/15-1. The opinions, hypotheses, and conclusions or recommendations expressed in this material are under the authors' responsibility and do not necessarily reflect the views of FAPESP or DFG. The authors would like to extend sincere thanks to São Paulo Research Foundation (FAPESP), the German Research Foundation (DFG), the Brazilian Council for Scientific and Technological Development (CNPq), and the Higher Education Personnel Improvement Coordination (Capes) for supporting this research.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Availability of data and material Supplementary electronic material is available in the journal.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes

were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alonso-Rasgado T, Thompson G (2006) A rapid design process for Total Care Product creation. *J Eng Des* 17:509–531. <https://doi.org/10.1080/09544820600750579>
- Andreasen MM (1994) Modelling—the language of the designer. *J Eng Des* 5:103–115. <https://doi.org/10.1080/09544829408907876>
- Anke J (2019) Design-integrated financial assessment of smart services. *Electron Mark* 29:19–35. <https://doi.org/10.1007/s12525-018-0300-y>
- Annamalai G, Hussain R, Cakkol M, Roy R, Evans S, Tiwari A (2011) An ontology for product-service systems. In: Hesselbach J, Herrmann C (eds) *Functional thinking for value creation*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19689-8_41
- Annamalai A, Battistella C, Nonino F (2016) Product service system: a conceptual framework from a systematic review. *J Clean Prod* 139:1011–1032. <https://doi.org/10.1016/j.jclepro.2016.08.061>
- Aurich JC, Fuchs C, Wagenknecht C (2006) Life cycle oriented design of technical Product-Service Systems. *J Clean Prod* 14:1480–1494. <https://doi.org/10.1016/j.jclepro.2006.01.019>
- Baines TS, Lightfoot HW, Evans S et al (2007) State-of-the-art in product-service systems. *Proc Inst Mech Eng Part B J Eng Manuf* 221:1543–1552. <https://doi.org/10.1243/09544054JEM858>
- Baines TS, Lightfoot HW, Benedettini O, Kay JM (2009) The servitization of manufacturing. *J Manuf Technol Manag* 20:547–567. <https://doi.org/10.1108/IJOPM-07-2010-0196>
- Bardin L (2013) *L'Analyse de Contenu*, 2nd edn. Presses Universitaires de France, Paris
- Benoît C, Norris GA, Valdivia S et al (2010) The guidelines for social life cycle assessment of products: just in time! *Int J Life Cycle Assess* 15:156–163. <https://doi.org/10.1007/s11367-009-0147-8>
- Benoit-Norris C, Cavan DA, Norris G (2012) Identifying social impacts in product supply chains: overview and application of the social hotspot database. *Sustainability* 4:1946–1965. <https://doi.org/10.3390/su4091946>
- Bertoni A, Bertoni M, Isaksson O (2013) Value visualization in product service systems preliminary design. *J Clean Prod* 53:103–117. <https://doi.org/10.1016/j.jclepro.2013.04.012>
- Bézivin J (2004) In search of a basic principle for Model Driven Engineering. *Novatica - Spec Issue UML (Unified Model Lang* 5:21–24
- Boehm M, Thomas O (2013) Looking beyond the rim of one's teacup: a multidisciplinary literature review of product-service systems in information systems, business management, and engineering & design. *J Clean Prod* 51:245–260. <https://doi.org/10.1016/j.jclepro.2013.01.019>
- Boucher X, Medini K (2016) Towards a generic meta-model for PSS scenarios modelling and analysis. *Procedia CIRP* 47:234–239. <https://doi.org/10.1016/j.procir.2016.03.038>
- Brilhante V, Macedo GT, Macedo SF (2006) Heuristic transformation of well-constructed conceptual maps into OWL preliminary domain ontologies. In: *Workshop on Ontologies and their Applications, WONTO*. pp 1–12
- Browning TR, Fricke E, Negele H (2006) Key concepts in modeling product development processes. *Syst Eng* 9:104–128. <https://doi.org/10.1002/sys.20047>
- Cavalieri S, Pezzotta G (2012) Product-service systems engineering: state of the art and research challenges. *Comput Ind* 63:278–288. <https://doi.org/10.1016/j.compind.2012.02.006>
- Cho C, Lee S (2011) Taxonomy of technology roadmaps in service areas. In: *Proc 1st Int Technol Manag Conf ITMC 2011* 462–467. <https://doi.org/https://doi.org/10.1109/ITMC.2011.5996013>
- Cooper LP (2003) A research agenda to reduce risk in new product development through knowledge management: a practitioner perspective. *J Eng Technol Manag* 20:117–140. [https://doi.org/10.1016/S0923-4748\(03\)00007-9](https://doi.org/10.1016/S0923-4748(03)00007-9)
- Correia A, Stokic D, Siafaka R, Scholze S (2018) Ontology for collaborative development of product service systems based on basic formal ontology. In: *ICE/ITMC 2017—Proceedings*, pp 1173–1180
- da Silva AR (2015) Model-driven engineering: a survey supported by the unified conceptual model. *Comput Lang Syst Struct* 43:139–155. <https://doi.org/10.1016/j.cl.2015.06.001>
- de Brentani U (1991) Success factors in developing new business services. *Eur J Mark* 25:33–59. <https://doi.org/10.1108/03090569110138202>
- De Jong PJ, Vermeulen PAM (2003) Organizing successful new service development: a literature review. *Manag Decis* 41:844–858. <https://doi.org/10.1108/00251740310491706>
- Diversity (2016) D1.4 Public System Concept—Cloud Manufacturing and Social Software Based Context Sensitive Product Service Engineering Environment for Globally Distributed Enterprise
- Dong M, Yang D, Su L (2011) Ontology-based service product configuration system modeling and development. *Expert Syst Appl* 38:11770–11786. <https://doi.org/10.1016/j.eswa.2011.03.064>
- Donnelly JP (2017) A systematic review of concept mapping dissertations. *Eval Program Plann* 60:186–193. <https://doi.org/10.1016/j.evalprogplan.2016.08.010>
- Durugbo C, Erkoyuncu JA, Tiwari A et al (2010) Data uncertainty assessment and information flow analysis for product-service systems in a library case study. *Int J Serv Oper Inf* 5:330. <https://doi.org/10.1504/IJSOI.2010.037002>
- Emili S, Ceschin F, Harrison D (2016) Product-service system applied to distributed renewable energy: a classification system, 15 archetypal models and a strategic design tool. *Energy Sustain Dev* 32:71–98. <https://doi.org/10.1016/j.esd.2016.03.004>
- Ericson ÅM, Larsson TC (2009) People, product and process perspectives on product/service-system development. In: Sakao T, Lindahl M (eds) *Introduction to product/service-system design*. Springer, London, London, pp 219–236
- Estrada A, Romero D (2016) A system quality attributes ontology for product-service systems functional measurement based on a holistic approach. *Procedia CIRP* 47:78–83. <https://doi.org/10.1016/j.procir.2016.03.215>
- Frické M (2019) The Knowledge Pyramid: the DIKW Hierarchy. *Knowl Organ* 46:33–46. <https://doi.org/10.5771/0943-7444-2019-1-33>
- Gainsburg J, Rodriguez-Lluesma C, Bailey DE (2010) A “knowledge profile” of an engineering occupation: temporal patterns in the use of engineering knowledge. *Eng Stud* 2:197–219. <https://doi.org/10.1080/19378629.2010.519773>
- Gericke K, Blessing LTM (2011) Comparisons of design methodologies and process models across disciplines: a literature review. In: *International Conference on engineering design, ICED11*. Copenhagen, pp 1–12

- Gero JS (1990) Design prototypes: a knowledge representation schema for design. *AI Mag* 11:26–36. <https://doi.org/10.1609/aimag.v11i4.854>
- Goh YM, Newnes L, Settanni E et al (2015) Addressing Uncertainty in Estimating the Cost for a Product-Service-System Delivering Availability: Epistemology and Ontology. In: Ebrahimipour V, Yacout S (eds) *Ontology modeling in physical asset integrity management*. Springer International Publishing, Cham, pp 199–219
- Gómez-Gauchía H, Díaz-Agudo B, González-Calero P (2004) Two-layered approach to knowledge representation using conceptual maps and description logics. In: *Concept Maps: Theory, Methodology, Technology*, Proc. of the First Int. Conf. on Concept Mapping
- Hajimohammadi A, Cavalcante J, Gzara L (2017) Ontology for the PSS lifecycle management. *Procedia CIRP* 64:151–156. <https://doi.org/10.1016/j.procir.2017.03.016>
- Hayes P, Eskridge TC, Saavedra R, et al (2005) Collaborative knowledge capture in ontologies. In: *Proceedings of the 3rd International Conference on Knowledge Capture - K-CAP '05*. ACM Press, New York, New York, USA, p 99
- Hein AM, Chazal Y, Boutin S, Jankovic M (2018) A Methodology for Architecting Collaborative Product Service System of Systems. In: *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE, pp 53–59
- Ignatow G, Mihalcea R (2017) *Text mining: a guidebook for the social sciences*. Sage Publications Inc, Los Angeles
- INCOSE (2015) *Systems engineering handbook: a guide for system life cycle processes and activities*, 4th edn. Wiley, San Diego
- Johne A, Storey C (1998) New service development: a review of the literature and annotated bibliography. *Eur J Mark* 32:184–251. <https://doi.org/10.1108/03090569810204526>
- Kan JWT, Gero JS (2009) Using the FBS ontology to capture semantic design information in design protocol studies. In: McDonnell J, Lloyd P (eds) *About: designing*. CRC Press, Analysing Design Meetings, pp 213–229
- Kim YS, Wang E, Park MW (2009) Value-function-structure modeling in an ontological representation of product-service systems. In: *International Conference on Engineering Design*. Stanford, pp 359–370
- Komoto H, Tomiyama T (2009) Systematic generation of PSS concepts using a service CAD tool. In: Sakao T, Lindahl M (eds) *Introduction to Product/Service-System Design*. Springer, London, pp 71–91
- Letia IA, Marginean A (2008) Client provider collaboration for service bundling. *Adv Electr Comput Eng* 8:36–43. <https://doi.org/10.4316/aecce.2008.01007>
- Li H, Ji Y, Luo G, Mi S (2016) A modular structure data modeling method for generalized products. *Int J Adv Manuf Technol* 84:197–212. <https://doi.org/10.1007/s00170-015-7815-6>
- Løkkegaard M, Mortensen NH, McAlone TC (2016) Towards a framework for modular service design synthesis. *Res Eng Des* 27:237–249. <https://doi.org/10.1007/s00163-016-0215-6>
- Lünnemann P, Stark R, Wang WM, Manteca PI (2017) Engineering activities - considering value creation from a holistic perspective. In: *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. IEEE, pp 315–323
- Lutters E, van Houten FJAM, Bernard A et al (2014) Tools and techniques for product design. *CIRP Ann* 63:607–630. <https://doi.org/10.1016/j.cirp.2014.05.010>
- Maleki E, Belkadi F, Bernard A (2018a) A meta-model for product-service system based on systems engineering approach. *Proc CIRP* 73:39–44. <https://doi.org/10.1016/j.procir.2018.04.016>
- Maleki E, Belkadi F, Bonjour E, Bernard A (2018b) Interfaces modeling for Product-Service System integration. In: *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE, pp 319–326
- Mariano S, Awazu Y (2016) Artifacts in knowledge management research: a systematic literature review and future research directions. *J Knowl Manag* 20:1333–1352. <https://doi.org/10.1108/JKM-05-2016-0199>
- Marques M, Agostinho C, Zacharewicz G, Jardim-Goncalves R (2017) Reconfigurable and updatable product-service systems: the path for sustainability and personalization. In: *Proceedings of the Symposium on Model-driven Approaches for Simulation Engineering*. pp 113–124
- Mastrogioacomo L, Barravecchia F, Franceschini F (2019) A worldwide survey on manufacturing servitization. *Int J Adv Manuf Technol* 103:3927–3942. <https://doi.org/10.1007/s00170-019-03740-z>
- Maussang N, Zwolinski P, Brissaud D (2009) Product-service system design methodology: from the PSS architecture design to the products specifications. *J Eng Des* 20:349–366. <https://doi.org/10.1080/09544820903149313>
- McKay A, Kundu S (2014) A representation scheme for digital product service system definitions. *Adv Eng Informatics* 28:479–498. <https://doi.org/10.1016/j.aei.2014.07.004>
- McMahon C (2015) Design informatics: supporting engineering design processes with information technology. *J Indian Inst Sci* 95:365–377
- Medin DL, Smith EE (1984) Concepts and concept formation. *Annu Rev Psychol* 35:113–138. <https://doi.org/10.1146/annurev.ps.35.020184.000553>
- Medini K, Boucher X (2019) Specifying a modelling language for PSS Engineering—a development method and an operational tool. *Comput Ind* 108:89–103. <https://doi.org/10.1016/j.compind.2019.02.014>
- Mien LH, Feng WL, Leng RGK (2005) An Integrated Manufacturing and Product Services System (IMPSS) Concept for Sustainable Product Development. In: *2005 4th International Symposium on Environmentally Conscious Design and Inverse Manufacturing*. IEEE, pp 656–662
- Milton NR (2007) *Knowledge acquisition in practice: a step-by-step guide*. Springer, London
- Mourtzis D, Fotia S, Doukas M (2015) Performance indicators for the evaluation of product-service systems design: a review. In: *Advances in Production management systems: innovative production management towards sustainable growth*. APMS 2015. IFIP Advances in Information and Communication Technology. Springer, Cham, pp 592–601
- Neely A (2008) Exploring the financial consequences of the servitization of manufacturing. *Oper Manag Res* 1:103–118. <https://doi.org/10.1007/s12063-009-0015-5>
- Nguyen HN, Exner K, Schnürmacher C, Stark R (2014) Operationalizing IPS2 development process: a method for realizing IPS2 developments based on process-based project planning. *Proc CIRP* 16:217–222. <https://doi.org/10.1016/j.procir.2014.01.024>
- Nonaka I, Takeuchi H (1995) *The knowledge-creating company: how Japanese companies create the dynamics of innovation*. Oxford University Press, New York
- North K, Kumta G (2018) *Knowledge management—value creation through organizational learning*, 2nd edn. Springer International Publishing, Cham
- Novak JD (1990) Concept mapping: a useful tool for science education. *J Res Sci Teach* 27:937–949. <https://doi.org/10.1002/tea.3660271003>
- Noy NF, McGuinness DL (2001) *Ontology development 101: a guide to creating your first ontology*. Stanford
- Orellano M, Medini K, Lambey-Checchin C, Neubert G (2019) A system modelling approach to collaborative PSS design. *Procedia CIRP* 83:218–223. <https://doi.org/10.1016/j.procir.2019.03.089>

- Pagoropoulos A, Andersen JAB, Kjær LL et al (2014) Building an ontology of product/Service-systems: using a maritime case study to elicit classifications and characteristics. In: 15th Working Conference on Virtual Enterprises (PROVE). Springer, Amsterdam, pp 119–126
- Pei E, Campbell I, Evans M (2011) A taxonomic classification of visual design representations used by industrial designers and engineering designers. *Des J* 14:64–91. <https://doi.org/10.2752/175630610X12877385838803>
- Pieroni M, Marques C, Campese C et al (2016) Transforming a traditional product offer into pss: a practical application. *Procedia CIRP* 47:412–417. <https://doi.org/10.1016/j.procir.2016.03.036>
- Poskela J, Martinsuo M (2009) Management control and strategic renewal in the front end of innovation. *J Prod Innov Manag* 26:671–684. <https://doi.org/10.1111/j.1540-5885.2009.00692.x>
- Princeton University (2010) WordNet. <http://wordnet.princeton.edu>. Accessed 3 Sept 2020
- Purchase HC, Colpoys L, McGill M et al (2003) UML Class Diagrams: An Empirical Study of Comprehension. *Software Visualization*. Springer, Boston, MA, pp 149–178
- Rabetino R, Harmsen W, Kohtamäki M, Sihvonen J (2018) Structuring servitization-related research. *Int J Oper Prod Manag* 38:350–371. <https://doi.org/10.1108/IJOPM-03-2017-0175>
- Raddats C, Kowalkowski C, Benedettini O et al (2019) Servitization: a contemporary thematic review of four major research streams. *Ind Mark Manag* 83:207–223. <https://doi.org/10.1016/j.indmarm.2019.03.015>
- Rizvi MAK, Chew E (2018) Towards systematic design of Cyber-Physical Product-Service Systems. In: Proceedings of International Design Conference, DESIGN. pp 2961–2974
- Rosa M, Rozenfeld H (2019) Using a design ontology to identify the terms that represent the design results across research communities. *Proc Des Soc Int Conf Eng Des* 1:2647–2656. <https://doi.org/10.1017/dsi.2019.271>
- Rosa M, Marques CAN, Rozenfeld H (2017) Commonalities and particularities of PSS design process and design thinking. *Procedia CIRP* 64:253–258. <https://doi.org/10.1016/j.procir.2017.03.020>
- Rowley J (2007) The wisdom hierarchy: representations of the DIKW hierarchy. *J Inf Sci* 33:163–180. <https://doi.org/10.1177/0165551506070706>
- Rozenfeld H, Rosa M, Fernandes S da C (2018) Servitization methodology: PSS design, change management or business model innovation? In: Proceedings of the 23rd International Seminar on High Technology. Piracicaba - Brazil, pp 91–116
- Rumbaugh J, Jacobson I, Booch G (2005) The unified modeling language reference manual. Addison-Wesley Publishing Company, Boston
- Saeki M, Kaiya H (2006) On relationships among Models, Meta Models and Ontologies. In: 6th OOPSLA Workshop on Domain-Specific Modeling (DSM'06). pp 1–10
- Sakao T, Shimomura Y, Sundin E, Comstock M (2009) Modeling design objects in CAD system for Service/Product Engineering. *Comput Des* 41:197–213. <https://doi.org/10.1016/j.cad.2008.06.006>
- Shen J, Wang L, Sun Y (2010) An ontology-based approach to product extension services modelling in servitisation. *Int J Serv Oper Inf* 5:373–399. <https://doi.org/10.1504/IJSOI.2010.037004>
- Shen J, Wang L, Sun Y (2012) Configuration of product extension services in servitisation using an ontology-based approach. *Int J Prod Res* 50:6469–6488. <https://doi.org/10.1080/00207543.2011.652744>
- Shimomura Y, Nemoto Y, Kimita K (2015) A method for analysing conceptual design process of product-service systems. *CIRP Ann Manuf Technol* 64:145–148. <https://doi.org/10.1016/j.cirp.2015.04.035>
- Simon HA (1996) *The Sciences of the Artificial*, 3rd edn. The MIT Press, Massachusetts
- Snowden D (2000) The ASHEN model: an enabler of action. *Knowl Manag* 3:14–17
- Sommerville I (2011) *Software engineering*, 9th edn. Pearson Education, Boston
- Stacey M, Eckert C, Hillerbrand R (2020) Process models: plans, predictions, proclamations or prophecies? *Res Eng Des* 31:83–102. <https://doi.org/10.1007/s00163-019-00322-8>
- Sutanto A, Yuliandra B, Tjahjono B, Hadiguna RA (2015) Product-service system design concept development based on product and service integration. *J Des Res* 13:1–19. <https://doi.org/10.1504/JDR.2015.067224>
- Torry-smith JM, Mortensen NH (2011) A mechatronic case study highlighting the need for re-thinking the design approach. In: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design. Lyngby/Copenhagen, Denmark
- Tran T, Park J (2016) Development of a novel co-creative framework for redesigning product service systems. *Sustainability* 8:434–450. <https://doi.org/10.3390/su8050434>
- Tranfield D, Denyer D, Smart P (2003) Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *Br J Manag* 14:207–222. <https://doi.org/10.1111/1467-8551.00375>
- Trevisan L, Brissaud D (2016) Engineering models to support product-service system integrated design. *CIRP J Manuf Sci Technol* 15:3–18. <https://doi.org/10.1016/j.cirpj.2016.02.004>
- Tukker A (2004) Eight types of product-service system: eight ways to sustainability? Experiences from SusProNet. *Bus Strateg Environ* 13:246–260. <https://doi.org/10.1002/bse.414>
- Ullman DG (2010) *The mechanical design process*. McGraw-Hill, Boston
- Van RR (2003) Clarity in the Usage of the Terms Ontology. *Taxonomy and Classification CIB Rep* 284(432):1–8
- Vandermerwe S, Rada J (1988) Servitization of business: adding value by adding services. *Eur Manag J* 6:314–324. [https://doi.org/10.1016/0263-2373\(88\)90033-3](https://doi.org/10.1016/0263-2373(88)90033-3)
- Vasantha GVA, Roy R, Lelah A, Brissaud D (2012) A review of product-service systems design methodologies. *J Eng Des* 23:635–659. <https://doi.org/10.1080/09544828.2011.639712>
- Vasantha GVA, Roy R, Corney JR (2015) Advances in designing product-service systems. *J Indian Inst Sci* 95:429–447
- Verlaine B (2017) A Conceptual Modelling of the Key Components and Relations of Service Systems. In: Za S, Drăgoicea M, Cavallari M (eds) *Exploring services science—8th International conference, IESS 2017*. Lecture notes in business information processing. Springer, Cham, pp 115–130
- Visser W (1992) Designers' activities examined at three levels: organization, strategies and problem-solving processes. *Knowl-Based Syst* 5:92–104. [https://doi.org/10.1016/0950-7051\(92\)90027-D](https://doi.org/10.1016/0950-7051(92)90027-D)
- Visser W (2006) Designing as construction of representations: a dynamic viewpoint in cognitive design research. *Human-Computer Interact* 21:103–152. https://doi.org/10.1207/s15327051hci2101_4
- Visser W (2006) *The cognitive artifacts of designing*. Lawrence Erlbaum Associates, New Jersey
- Wang PP, Ming XG, Wu ZY et al (2014) Research on industrial product-service configuration driven by value demands based on ontology modeling. *Comput Ind* 65:247–257. <https://doi.org/10.1016/j.compind.2013.11.002>
- Wang WM, Lünemann P, Preidel M, Stark PR (2017) Wissen in Produktentwicklungsprozessen – Ein aktivitätsbasierter

- Analyseansatz. In: Brökel K, Grote K-H, Stelzer R, et al. (eds) 15. Gemeinsames Kolloquium Konstruktionstechnik 2017. pp 183–192
- Wang Z, Chen C-H, Zheng P et al (2019) A graph-based context-aware requirement elicitation approach in smart product-service systems. *Int J Prod Res.* <https://doi.org/10.1080/00207543.2019.1702227>
- Webster J, Watson RT (2002) Analyzing the past to prepare for the future: writing a literature review. *MIS Q* 26:xiii–xxiii. <https://www.jstor.org/stable/4132319>
- Wolfenstetter T, Basirati MR, Böhm M, Krcmar H (2018) Introducing TRAILS: A tool supporting traceability, integration and visualisation of engineering knowledge for product service systems development. *J Syst Softw* 144:342–355. <https://doi.org/10.1016/j.jss.2018.06.079>
- Yang M, Evans S (2019) Product-service system business model archetypes and sustainability. *J Clean Prod* 220:1156–1166. <https://doi.org/10.1016/j.jclepro.2019.02.067>
- Zhu H, Gao J, Cai Q (2015) A product-service system using requirement analysis and knowledge management technologies. *Kybernetes* 44:823–842. <https://doi.org/10.1108/K-11-2014-0244>
- Zhu H, Li J, Gao J, Lu W (2017) Research on integrated product service system driven by customer value based on ontology technology. *Kybernetes* 46:1039–1051. <https://doi.org/10.1108/K-06-2016-0139>