Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 52 (2020) 209–214

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

5th International Conference on System-Integrated Intelligence

Application of System Dynamics for Holistic Product-Service System Development

Daniel Kloock-Schreiber[a,b], Paul Christoph Gembarski[a], Roland Lachmayer[a]

[a] *Gottfried Wilhelm Leibniz Universität Hannover, Institut für Produktentwicklung und Gerätebau, Welfengarten 1A, 30167 Hannover, Germany*
[b] *Corresponding author: schreiber@ipeg.uni-hannover.de*

Abstract

In order to develop Product Service Systems (PSS), a holistic view on the system and a coequal development of service and product parts is necessary. Particularly for the beginning of the development of PSS, existing approaches show lacks and start with vague defined initial phases. This leads to inadequate methodological support for the consistent design of the overall system and simultaneous elaboration of the requirements down to the parameters of individual components. Therefore, a procedure is required that completely maps the PSS and enables detailed development for relevant individual areas, taking into account existing constraints. At the beginning of the development a model is necessary, which first defines the system boundaries of the PSS and maps the performance and control flows of the system. In addition, the integration of further actors into the PSS must be made possible. This paper presents an approach that uses System Dynamics (SD) to design a PSS. With this approach, the representation of the system is initially possible at a high level of abstraction, whereby the representation can be further refined and detailed. Parallel to this, a preliminary design for planning and controlling media flows can be carried out from the first system representation and further detailed parallel to the system representation. An essential advantage is that the detailing can also only be carried out for individual areas, which can be displayed in sub-models, but can also be reintegrated into the overall representation. The sub-models can be implemented function-specifically on the basis of resources and competencies of individual actors. For system-relevant areas, planning and design can be concretized in the sub-models (which can be realized by products as well as services) down to the lowest hierarchy level. This can take place up to the definition of individual physical component parameters and has thus up to the phase of the elaboration effects on the development of the parts. In return, the effects of changes in system-relevant parameters on the overall system can also be examined. For the PSS, a model is built in which system-determining functions and principles are represented and developed. The model is constructed in such a way that non-system-determining functions and principles are defined as variables or black boxes. Requirements and parameters are derived from this system development. These are used for the further development steps in the development process. Depending on whether it concerns system-relevant areas or not, the entry into the development process takes place later in the elaboration phase (e.g. in the area of detailed design) or partly earlier in the concept phase (e.g. function development). It is also possible to enter an early phase in the development process of the individual parts, accompanied by already defined functions, sub-functions or parameters that must not be changed in the course of development. With this approach a holistic development of the system with all product and service parts as well as their connections and dependencies is possible.

Keywords:
Product-Service Systems; PSS; System Dynamics; Model-Based Development, PSS Development; Simulation-Based Design, Product Development

1. Introduction

PSS are described in scientific literature as socio-technical systems that are life cycle-, sustainability- and customer- oriented [16, 17, 28]. They represent an integrated solution that meets individual customer needs by combining product and service parts. It is irrelevant whether the respective value proposition is mainly realized by the product or service components [26, 27]. Thus, engineering is evolving from a development process for individual products or components to the development of systems with the design of distributed product, software and service processes as an integrated solution [31]. An essential point for the development of a PSS is the co-equal development of product and service parts [21]. Due to the fact that PSS requirements are dynamic [30] and in order to be able to offer various customer-specific solutions, it is necessary to provide and develop a PSS solution space which contains the PSS com-

10.1016/j.promfg.2020.11.036

ponents [20] and their dependencies and relationships, which must be taken into account during development [2, 6].

Various approaches and tools already exist for the development of PSS, which have been examined in previous papers (e.g. with literature reviews) [7, 19, 20]. The existing approaches can be divided into two main types: The first includes approaches that describe a retrospective PSS development where the system is built using an existing physical artifact (e.g. [11, 12, 22]). The second type includes approaches that start directly with the system development and neglect previous steps. Creating a transition between PSS business model development and system development is not achieved. Although this is particularly important, especially for the aspect of equal development of product and service components (without early commitment to value creation through predefined individual product or service components). Only a system development that achieves this makes it possible to exploit the full potential of a PSS.

A tool for achieving this in development is System Dynamics (SD). According to Sterman [25], SD is a method to better learn and understand complex socio-technical systems, especially in a complex business environment. It is an approach and simulation tool for modeling complex, dynamic systems.
Due to its interdisciplinarity, SD is a suitable simulation method for modeling and simulation of PSS business models and there are already approaches that use System Dynamics to map and simulate the PSS business model and its value proposition to the customer [15].

The approach presented in this paper goes beyond this and uses system dynamics not only to simulate the system, but uses system dynamics for development. This is done by drawing conclusions from the value proposition and the business model about the main function to be fulfilled and the sub-functions necessary for implementation. These are used to develop and elaborate the structure of the system and its parts. The approach for the development of PSS using SD is presented in this paper which is structured as follows. Starting with the motivation, section 2 contains the related work to the topic of this paper. Then in section 3 the approach to solution space development using System Dynamics is introduced and in section 4 an example is given. Section 5 gives a summary and an outlook on further work.

2. Related Work

In this section the basic idea of System Daynamics is summarized and existing literature on the use of SD for PSS is presented.

2.1. System Dynamics (SD)

SD is a method from business management, which was introduced in the 1960ies by Forrester [4] under the name Industrial Dynamics. It consists of flows, control loops and parameters that are linked in a model. The basic elements of SD are shown in Fig. 1, using a very simplified example (stock of cars), and are presented below.

Stocks are represented in the models by rectangles, in Fig. 1 the stock of cars is shown in the middle. Stocks give inertia and memory to the system, they create delays by accumulating the difference between the inflow to a process and its outflows. Inflows are represented by a pipe (shown as double line arrow) pointing into the rectangle and outflows by a pipe pointing out of the rectangle. Different stocks in a model can be connected by flows. If the stocks are outside the model boundaries, they are represented as clouds and defined as sources (with outflows) and sinks (with inflows). Sources and sinks are defined with an infinite capacity and can never restrict the flow they support. The flow is controlled by valves (shown in the middle of the pipes) that regulate the amount flowing in or out [24]. The controlled inflow in the example (Fig. 1) shows the production rate and the controlled outflow the scrapping rate of cars.

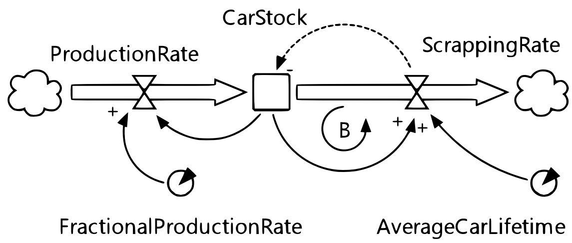

Fig. 1. Basic Elements of a System Dynamics Model

Beside flows, causal loops and parameters are important components of SD models. The parameters are represented by circles with a triangle (in the example, the fractional production rate and the average car lifetime) and dynamic variables by a clean circle. All model elements (stocks, parameters, variables and the valves of the flows) can be connected by arrows that represent causal links. A plus or minus on the arrowhead indicates the polarity of the connection. In the example, the car production rate is linked to the parameter "fractional production rate" and the car stock. The scrapping rate is positively influenced (amplified) by the average car lifetime and the car stock, while at the same time the car stock is negatively influenced by the scrapping rate. This leads to a loop, which is shown in Fig. 1 at the right half and marked with a B because it is a negative (balancing) loop. The opposite would be a positive (reinforcing) loop [25].

A large part of system dynamics modeling consists of modeling and representing the feedback processes and other complex elements that determine the dynamics of a system. These essentially arise from the interaction of the two types of feedback loops presented, the positive (or self-reinforcing) and negative (or self-correcting) loops. Positive loops tend to amplify or reinforce everything that happens in the system. Negative loops work against change, they describe all processes that tend to limit themselves, i.e. processes that create balance and equilibrium [23].

2.2. System Dynamics for PSS

In literature there are various works that use SD in the field of PSS research for the representation or simulation of PSS. A very general model is presented by Bianchi et al. [3], who show a SD model that simulates and represents the transition from a product-oriented to a PSS market. Parameters and variables of the market are defined and a model is built based on them.

SD is not used for a whole market, but for modelling the business model of the PSS and the dynamic behaviour over the PSS life cycle in the approach of Meier and Boßlau [15]. They use SD as strategic modeling for developing new business models and adapting existing business models in the PSS life cycle.

The approach of Legnani et al. [14] also provides a consideration at management level. In this approach, the authors have created a general SD model at the management level that qualitatively simulates how the provision of PSS can affect the service performance of a company. Geum et al. [8] use SD to integrate it into scenario planning. They consider scenarios with a combination of products and services in their publication.

Another paper presents an approach for measuring the performance of functional dynamics (achieved through the structural interactions of the individual functions of products, services and relevant actors), which can be used to analyze the long-term behavior of PSS [13].
The approach of Grandjean et al. [9] use SD for the strategic planning of PSS. Two models are presented, the so-called resource model and the adaptation model. There is an approach for a multi-level consideration, but there is no impact on the individual product in the PSS or the exchange of PSS modules.

In summary, previous research on the combination of SD and PSS is primarily concerned with the representation of PSS markets or business models and their changes. SD is used to represent and simulate the systems. A development of the system using SD has not yet been documented.

3. PSS Solution Space Development with System Dynamics

In the development of physical products and their solution spaces, the goals of the development are defined at the beginning. These are usually closely related to the corporate goals or corporate strategy and thus the desired business model. The developed products generally serve the fulfillment of functions, whereby function is understood as the general and intended connection between input and output of a system with the aim of fulfilling a task [29]. In this context, complexity emerges when multiple variants of components and uncertainties in this fulfillment exist [1, 5]. For the development of a product and the decomposition in the development process this means that the main function of the product is clear in the ”planning phase”, but the sub-functions are still open and are developed after the development of product structure and design [18].

These basic relationships can also be applied to the development of PSS. Therefore, a model is needed that is able to represent a PSS where the general purpose is clear but the implementation and sub-functions are not yet defined. SD is a tool that is able to capture feedback processes, stocks and flows, time delays and other sources of dynamic complexity for systems. It supports the design and evaluation of new system structures and their consequences [23].

3.1. Approach

SD enables to map the essential processes and dependencies that occur in a business model. This is initially possible at a high level of abstraction and can then be refined and detailed. This is the basis for the approach presented here, which uses SD in the development of PSS and applies it not only for system simulation, but also for the development of the system and system solution spaces. In these solution spaces, parts of the PSS can be further detailed and different variants can be compared. For the individual parts, different value creation mechanisms (fulfillment of functions by service or product) can also be taken into account, without dependence on predefined products or services.

Thus, the use of SD for development makes it possible to derive sub-functions and system structures from the main function and to describe a PSS solution space. SD can be used not only for system simulation, but also for working out the solution space, right up to defining specifications for parts of the PSS.

When modeling PSS solution spaces with SD, the first step is to describe the main function of the PSS with the corresponding processes and control loops. This is followed by detailed specifications for the entire system or in sub-models that are restricted to individual system areas. In this way, different sub-functions can be compared in the system context and parameter studies can be carried out on the variation of individual sub-parameters and their effects in different PSS scenarios.

3.2. System Dynamics Model

The SD model is built iteratively and is further detailed with each step. This can be done in the initial model, or by including sub-models that were previously defined as black boxes. New actors can also be integrated into the PSS, or existing actors can be exchanged. The level of detail in the SD model is based on the levels of the functional structure of the system. Starting with the main function, up to a level of detail where all necessary specifications for relevant product and service parts of the PSS are defined. The system architecture can be described based on these parts.

4. Application Example

The example presented here includes a PSS that can be classified as a result-oriented PSS that provides an accountable service to the customer, regardless of how it is achieved. The supplier provides high quality filter coffee in coffee cups and is responsible for the preparation and therefore for the provision of machines if necessary. In addition to the machine supplier, the supplier of the coffee beans (the consumables) is also an actor in the PSS, as is, for example, the actor responsible for machine maintenance. Depending on the application scenario, different requirements for the machines become relevant, for example, to reduce the investment costs for machines and to adapt them to individual use cases.

4.1. Used Modelling Environment

Various programs can be used as a modeling environment for SD, e.g. Anylogic and Vensim as commercial solutions. As free software exists, among others, the Eclipse based Semantics System Dynamics, but further development and support of this software is discontinued. When using the Anylogic software, it is possible to import Vensim models and link the SD simulation with discrete event-based or agent-based simulations.

The implementation of the development of the presented example is done in the mentioned software (Anylogic PLE), because it has proven to be promising also in the combination of SD with further simulations. In the following, the example models that have been built up are presented. The models are simplified for better clarity, or are only shown in excerpts.

4.2. Model of the Example

At the beginning of the development the main function is derived and defined from the described example. The first step of the modelling is to create a model that contains the main function of the Coffee PSS, as well as the essential parts of the PSS (the supply of consumer goods and consumption). Figure 2 shows this first stage of the SD model with the coffee flow and simple control loops where the coffee output (CoffeeConsumption) ends in the sink of the model and is influenced by the variable "NrConsumer". The coffee output has an influence on the consumption of coffee beans and thus on the delivery of consumer goods (BeanDelivery). In the system shown, it is also influenced by the variable "SupplyInterval" and is fed by the source of the system.

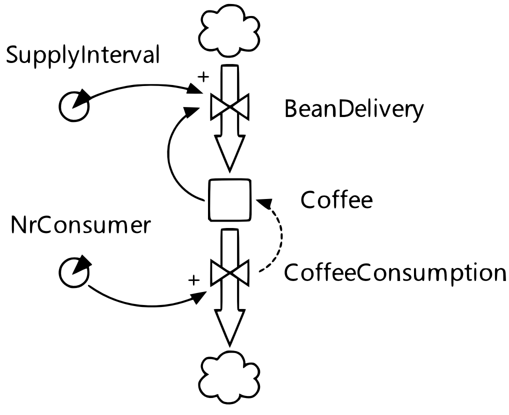

Fig. 2. First Model of the Coffee PSS

In the next steps the main function is divided into subfunctions. For the SD model this means detailing in different areas and adding parameters, actors and dependencies. Figure 3 shows the extended model of the example with the brewing chamber and the thermal tank of a filter coffee machine, which further processes the inputs (BeanDelivery and WaterSupply) of the system and makes them available for consumption.

In comparison to the first model, here in Figure 3 on the left-hand side, for example, the actor in the coffee supply was included in the model by a fixed set of variables (working hours, delivery routes, minimum or maximum delivery quantities) as a general condition. These influence the supply interval, which has changed from a fixed framework condition to a variable and is influenced not only by the supplier but also by the coffee cups provided. As in the first model, BeanDelivery is still dependent on the SupplyInterval.

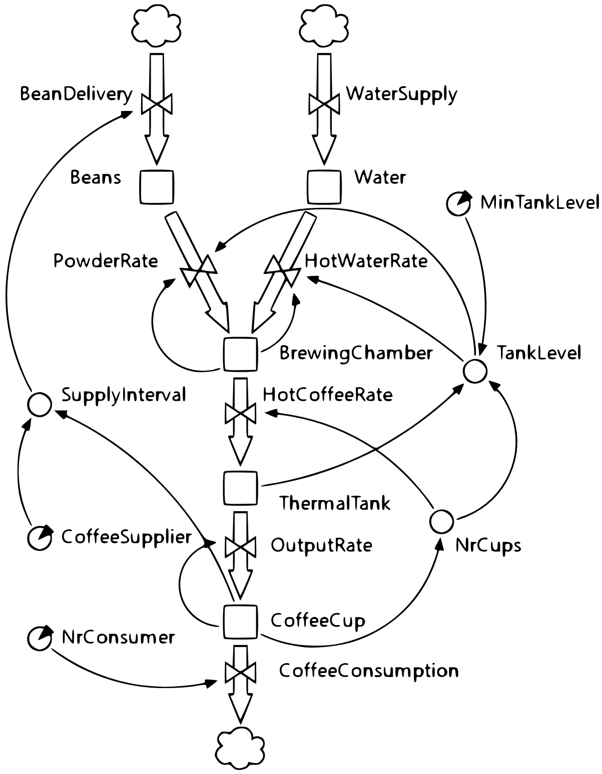

Fig. 3. Model including the Brewing Chamber and the Thermal Tank

On the right side of Figure 3, the dependencies, control loops and variables that trigger the brewing of a new batch are shown. To do this, the "PowderRate" and "HotWaterRate" flows are influenced by the "BrewingChamber" (with the dependencies maximum volume and ready for brewing) and the "TankLevel" variable, which depends on the "MinTankLevel" constraint, the "ThermalTank" and the "NrCups" variable (based on "CoffeeCup" dispensing and "CoffeeConsumption" flow). The control loops described in the state of the art can also be drawn in here, but since they primarily serve the system understanding and are not necessary for development and simulation, they have been omitted in the figure.

Up to this model, a system development is carried out which defines parameters, dependencies, boundary conditions and the resulting requirements for individual system areas. No requirements for individual components of service or product parts of the PSS could be derived yet. Figure 4 shows a section of the system with further details and how the approach makes it possible to detail individual sub-areas for sub-functions to such an extent that requirements and parameters for product components can be derived.

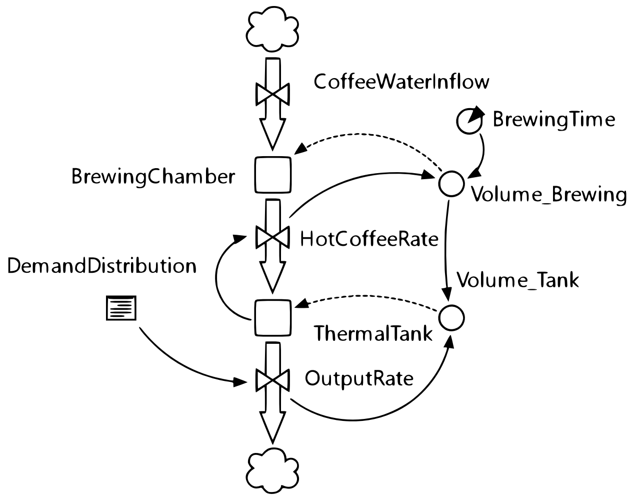

Fig. 4. Model with exemplary parameters for product parts

When considering, for example, the volume parameters of the brewing chamber and the thermal tank, with constant technical boundary conditions for the entire machine (here, the output quantities and output speeds are assumed to be the same and thus no additional parameters in the system are changed), different usage scenarios can have different effects on the design of the physical components. This demonstrates the advantages (in addition to system understanding and general system simulation) of PSS solution space development with SD.

The scenarios considered here in the example are on the one hand the provision of a continuous output of coffee and the output with expected peak times (high capacity utilization) combined with a break (with low demand). Furthermore, for this example it should be noted that in the coffee machine under consideration the brewing chamber is technically more complex than the thermal tank, and in production a larger brewing chamber is much more cost-intensive than a larger thermal tank.

The section from the simulation in Figure 4 is reduced to the ”CoffeeWaterInflow” and the ”OutputRate” as system input and output, with the ”BrewingChamber” and the ”ThermalTank” as stocks in between. The different scenarios are introduced into the simulation model via the ”DemandDistribution” and influence the ”OutputRate”. The relevant parameters here, which follow from the ”OutputRate” and influence the variable ”VolumeTank”, are the output per time (ml/t) and the duration (t) for which this is done. In addition to these, the ”VolumeTank” variable is also influenced by the ”VolumeBrewing” variable, as the tank must be able to hold the brewed quantity after the brewing process. The variable ”VolumeBrewing” in turn is influenced by the ”HotCoffeeRate” and the fixed variable ”BrewingTime” and thus also by an output per time (ml/t) and a process-related delay (t).

The result of the development taking into account the scenarios considered are two different configurations of the brewing chamber and the tank size. For the continuous dispensing of coffee, a brewing chamber volume of 1,000 ml and a tank volume of 1,500 ml is required. Thus, the brewing time can be buffered by the thermal tank and a continuous output can be realized.

For the scenario of an output with expected peak time, it is possible to reduce the brewing chamber to a volume of 400 ml. In this scenario, the thermal tank is considerably larger at 4,000 ml. This is because several batches are already pre-brewed in the system for the expected peak load and the output is then realized from the tank. However, these parameters only apply to the example case under consideration, since it must be noted that the size of the modules depends on the specific scenario and that the size of the modules varies depending on the duration and distribution of the peak times (as well as the maximum storage time for the coffee).

The variables that have been determined here as examples for the volumes are also defined in the system for other requirements. Thus, parameter and dependency networks can be derived from the variables and dependencies that are modeled around the flows of the system. These networks can be used to build parametric models for solution spaces in CAD environments.

5. Conclusion and Outlook

This paper gave an introduction to SD and showed in a short overview how SD is used in the PSS context, respectively in PSS development. Based on this, an approach was presented that goes beyond the previous applications of SD and develops PSS with the help of SD. The process of developing and building SD models is based on designing the functional structure of PSS. This makes it possible to develop a PSS with a focus on fulfilling the functions of the individual parts, regardless of whether the added value is realized through product or service components. It is also possible to integrate other actors into the model, by means of fixed parameters or a detailed modeling of the network extension. In addition, individual PSS parts can be exchanged in the created models, which enables the comparison of different product parts, or product and service variants. In future work, this can be used to automate scenario variation and thus optimize the system. A sensitivity analysis, which examines the effect of parameter variations on the overall system, should also be part of further research. Since Anylogic is already used as software, it is a goal to set up discrete event simulations (DES) for individual areas of the developed system and to integrate them into the model. Thus, the models presented here can also be used to optimize the PSS.

As already indicated in the last paragraph of the application example, the approach presented here can also be used to derive a parameter- and constraintnetwork from the model from the parameters and dependencies of the individual PSS parts in a further step. This can be used as a basis for building a parametric CAD model. In this way a continuous development of the PSS up to a link with the extended CAD models presented in [10] can be achieved in further work.

Acknowledgements

This research was conducted in the scope of the research project SmartHybrid Product Engineering (ID: 85003608) which is partly funded by the European Regional Development Fund (ERDF) and the State of Lower Saxony (Investitions- und Förderbank Niedersachsen NBank). We like to thank them for their support.

References

[1] Aldanondo, M., Vareilles, E., 2008. Configuration for mass customization: how to extend product configuration towards requirements and process configuration. Journal of Intelligent Manufacturing 19, 521–535.

[2] Aurich, J.C., Fuchs, C., Wagenknecht, C., 2006. Life cycle oriented design of technical product-service systems. Journal of cleaner production 14, 1480–1494.

[3] Bianchi, N.P., Evans, S., Revetria, R., Tonelli, F., 2009. Influencing factors of successful transitions towards product-service systems: a simulation approach. International Journal of mathematics and computers in simulation 3, 30–43.

[4] Forrester, J.W., 1968. Industrial dynamicsafter the first decade. Management Science 14, 398–415.

[5] Gembarski, P.C., Lachmayer, R., 2017. A business typological framework for the management of product complexity, in: Managing Complexity. Springer, pp. 235–247.

[6] Gembarski, P.C., Lachmayer, R., 2018. Product-service-systems–what and why developers can learn from mass customization. Enterprise Modelling and Information Systems Architectures (EMISAJ) 13, 16–1. doi:10.18417/emisa.13.16.

[7] Gembarski, P.C., Schreiber, D., Schoormann, T., Knackstedt, R., Lachmayer, R., 2018. are we talking about the same thing? analyzing effects of mass customization and product-service systems on sustainability, in: Proceedings of the 8th International Conference on Mass Customization and Personalization–Community of Europe (MCP - CE 2018), September 19-21, 2018, Novi Sad, Serbia, UNIVERSITY OF NOVI SAD–FACULTY OF TECHNICAL SCIENCES - Department of Industrial Engineering and Management.

[8] Geum, Y., Lee, S., Park, Y., 2014. Combining technology roadmap and system dynamics simulation to support scenario-planning: A case of car-sharing service. Computers & Industrial Engineering 71, 37–49.

[9] Grandjean, L., Alevifard, S., Steven, M., 2014. Strategic adaptability of industrial product-service-systems–dynamic effective ips2. Procedia CIRP 16, 314–319.

[10] Kloock-Schreiber, D., Domarkas, L., Gembarski, P.C., Lachmayer, R., 2019. Enrichment of geometric cad models for service configuration, in: 21th International Configuration Workshop, September 18-19, 2019, Hamburg, Germany.

[11] Kuntzky, K., 2013. Systematische Entwicklung von Produkt-Service Systemen. Vulkan-Verlag.

[12] Laurischkat, K., 2012. Product-Service Systems: IT-gestützte Generierung und Modellierung von PSS-Dienstleistungsanteilen. Shaker.

[13] Lee, S., Han, W., Park, Y., 2015. Measuring the functional dynamics of product-service system: A system dynamics approach. Computers & Industrial Engineering 80, 159–170.

[14] Legnani, E., Cavalieri, S., Marquez, A.C., Díaz, V.G., 2010. System dynamics modeling for product-service systems: A case study in the agri-machine industry, in: Proc. of APMS.

[15] Meier, H., Bolau, M., 2012. Design and Engineering of Dynamic Business Models for Industrial Product-Service Systems. pp. 179–184. doi:10.1007/978-3-642-32847-3_30.

[16] Mont, O.K., 2002. Clarifying the concept of product–service system. Journal of cleaner production 10, 237–245.

[17] Müller, P., 2014. Integrated engineering of products and services: Layer-based development methodology for product-service systems. Fraunhofer Verlag.

[18] Pahl, G., Beitz, W., 2013. Engineering design: a systematic approach. Springer Science & Business Media.

[19] Schreiber, D., Gembarski, P.C., Lachmayer, R., 2017. Modeling and configuration for product-service systems: State of the art and future research, in: 19th International Configuration Workshop, September 14-15, 2017, Paris, France.

[20] Schreiber, D., Gembarski, P.C., Lachmayer, R., 2018a. Datamodels for pss development and configuration: Existing approaches and future research, in: Customization 4.0. Springer, pp. 55–74. doi:10.1007/978-3-319-77556-2_4.

[21] Schreiber, D., Gembarski, P.C., Lachmayer, R., 2018b. Developing a constraint-based solution space for product-service systems, in: Proceedings of the 8th International Conference on Mass Customization and Personalization–Community of Europe (MCP - CE 2018), September 19-21, 2018, Novi Sad, Serbia, UNIVERSITY OF NOVI SAD–FACULTY OF TECHNICAL SCIENCES - Department of Industrial Engineering and Management.

[22] Steinbach, M., 2005. Systematische Gestaltung von product-service-systems: integrierte Entwicklung von product-service-systems auf Basis der Lehre von Merkmalen und Eigenschaften. LKT.

[23] Sterman, J.D., 2002. System dynamics: systems thinking and modeling for a complex world .

[24] Sterman, J.D., 2002. Systems dynamics modeling: tools for learning in a complex world. IEEE Engineering Management Review 30, 42–42. doi:10.1109/EMR.2002.1022404.

[25] Sterman, J.D., 2010. Business dynamics. Irwin/McGraw-Hill c2000..

[26] Sturm, F., Bading, A., Schubert, M., 2007. Investitionsgüterhersteller auf dem Weg zum Lösungsanbieter: eine empirische Studie;[fit2solve]. Fraunhofer-IRB-Verlag.

[27] Thomas, O., Walter, P., Loos, P., 2010. Konstruktion und anwendung einer entwicklungsmethodik für product-service systems, in: Hybride Wertschöpfung. Springer, pp. 61–81.

[28] Tukker, A., 2004. Eight types of product–service system: eight ways to sustainability? experiences from suspronet. Business strategy and the environment 13, 246–260.

[29] Ullman, D.G., 2010. The mechanical design process: Part 1. McGraw-Hill.

[30] Wiesner, S., Hauge, J.B., Sonntag, P., Thoben, K.D., 2019. Applicability of agile methods for dynamic requirements in smart pss development, in: IFIP International Conference on Advances in Production Management Systems, Springer. pp. 666–673.

[31] Wiesner, S., Nilsson, S., Thoben, K.D., 2017. Integrating requirements engineering for different domains in system development: Lessons learnt from industrial sme cases, in: 9th CIRP IPSS Conference: Circular Perspectives on PSS, June 19-21, 2017, Copenhagen, Denmark, Elsevier. pp. 351–356.