8th International Conference on Mass Customization and Personalization – Community of Europe (MCP-CE 2018)
Digital Customer Experience
September 19-21, 2018, Novi Sad, Serbia

DEVELOPING A CONSTRAINT-BASED SOLUTION SPACE FOR PRODUCT-SERVICE SYSTEMS

Daniel Schreiber, Paul Christoph Gembarski, Roland Lachmayer
Leibniz University of Hanover, Institute of Product Development, Hannover, Germany

Abstract: *Product-Service Systems (PSS) are considered as problem-oriented solutions for individual customer needs. As such, they may be considered as mass customization offerings consisting of intertwined product and service components. Literature reports about single holistic approaches for development and design of PSS, which are either conceptual or discussed on simple and restricted examples. Compared to merely product offerings, solution space development for PSS is still in its infancy. It is agreed, that product configuration is a key tool for the success of a mass customization business model. From a product development point of view, parametric modeling and knowledge based engineering are technologies for its realization. In PSS development, those are only minor developed due to a missing common data model for products and services. In this paper the authors bridge this gap introducing a constraint-based modeling approach.*

Key Words: ***Solution Space, Product-Service System, Constraint-based Modeling, Data Model, ARIS***

1. INTRODUCTION

Nowadays, the rapid dissemination of knowledge and the international harmonization of standards leads to the problem that it is more and more difficult for companies to distinguish themselves from competitors only by technical product characteristics [1].

One possibility to challenge this is to combine products with services to provide an integrated solution. These so-called Product-Service Systems (PSS) are solutions, which fulfill individual customer needs [2, 3]. Here it is secondary wheather the value proposition and revenue is achieved primarily by the product or service components [2]. To benefit the advantages of PSS an integrative relationship between product and service components has to be developed [3, 4].

PSS are seen as customer-specific solutions, this would lead lead to a very high development effort if the entire development process have to be executed for each individual customer and each changing customer requirement [5, 6, 7]. One way to challenge this problem is to model a configurable PSS [4, 8, 9, 10].

Based on the conceptual similarity of the enterprise types of PSS and Mass Customization (MC), PSS can be understood as an MC offer. So MC development processes and modeling tools could be applied to PSS [12].

One of the the key principals of MC is the solution space modeling [11]. Which have already been applied for physical products. How a solution space is explored is determined by the control knowledge. This knowledge is implemented by knowledge-based engineering (KBE) principles like rule-based, case-based or model-based reasoning [13].

In the PSS literatur approaches for rule-based and case-based configurators (eg, in the work of Laurischkat [8]) are documented, but here is a lack of model-based configuration for PSS [7]. For this a model is requiered which combines product and service parts and all their dependencies. Like mentioned by Wagner et. al [16] it is an important requiered competence for the development of PSS is to combine product and service parts in a appropiate way with all its dependencies dependencies [16].

The advantage of a model-based configuration with a parametric model would be the possible adaptation and variant design of PSS which could be as powerful as it is already state of the art with the CAD systems for purely physical products. In addition, such a data model would additionally enable computer-aided product optimization [12].

In this paper an approach for a solution space modeling of products and services in a joint parametric model is shown and an example presented. The paper is part of a research which is structured by the design research methodology (DRM) of Blessing et Al. [17]. The Stages “Research Clarification” and the “Descriptive Study I” are already executed and presented in previous papers [12, 15, 18], the research of this paper takes place in the Stage “Prescriptive Study”.

The paper is structured as follws. After the introduction an overview about the relevant state of the art (section 2) is given. This contains an introduction to PSS, CAD-based solution space modeling for physical artefacts, service modeling with the architecture of integrated information systems (ARIS) and PSS configuration approaches. In section 3 a solution space modeling for PSS with a model-based configuration approach is presented and in section 4 an application example described. The paper ends with a conclusion and outlook at further research in section 5.

2. STATE OF THE ART

In order to correctly classify the example presented in this paper and the underlying approach, at first an overview of relevant areas of the state of the art is shown. This includes basics on PSS, solution space modeling for physical artifacts, service modeling and as well existing approaches to PSS configuration.

2.1. Product-Service Systems

In literature various characterizations and approaches for the development and configuration of PSS are discussed [5, 6, 19, 20, 21]. A characterization which is often used for PSS, was developed by Tukker [21] in 2004. He destinguishes three different types of PSS and the corresponding business models according to the content of physical product and service components of the value creation (Fig. 1):

- *Product-oriented*: Product-related services such as startup and initial operation; maintenance contracts; supply of consumables; financing plans; Advice and consultancy such as training, logistics and optimization.
- *Use-oriented:* Product lease; product sharing; product pooling.
- *Result-oriented:* Activity management; Outsourcing; pay per unit; functional result.

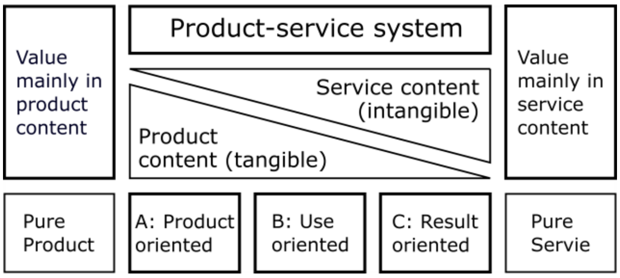

Fig. 1. *Main PSS categories based on Tukker [21]*

Literature agrees that the structure of the PSS development process influences the quality of a PSS [2]. To address the individual customer needs and due to the fact that a combined product and service development is necessarry a PSS specific development is needed. The literature agrees to that, but the existing approaches remain mainly vague and conceptual [22]. Additionaly they are discussed on very simple or very specific examples, which makes the transfer to relevant applications difficult [12].

A challange of PSS research and development is the multi-disciplinarity it includes researchers from various fields of interest. In the absence of the evaluation of the existing approaches, no one of them can be seen as a generally accepted and standardized approach for the development of PSS [23].

But based on literature studies (presented in previous papers [12, 15, 18]) on the existing characterizations of PSS and the existing theses and approaches in PSS design research, the following main implications can be named for the PSS development [15, 18]:

- *coequal development of product and service components* [15, 18]
- *integration and addressing of individual customers and their needs* [15, 18]
- *monitoring and addressing of the customers requirements during the whole life-cycle of the PSS* [15, 18]

2.2. CAD-based Solution Space Modelling

In contrast to rigid geometry modelling that cannot represent solution spaces, parametric CAD is able to do so. Especially, the possibility of defining mathematical and logical constraints between parameters in a CAD system makes it possible to implement knowledge in explicit form within digital prototypes. In relation to variant design, the designer does not only specify the product shape but also the control and configuration concept for his component and thus describes a solution space [24, 25].

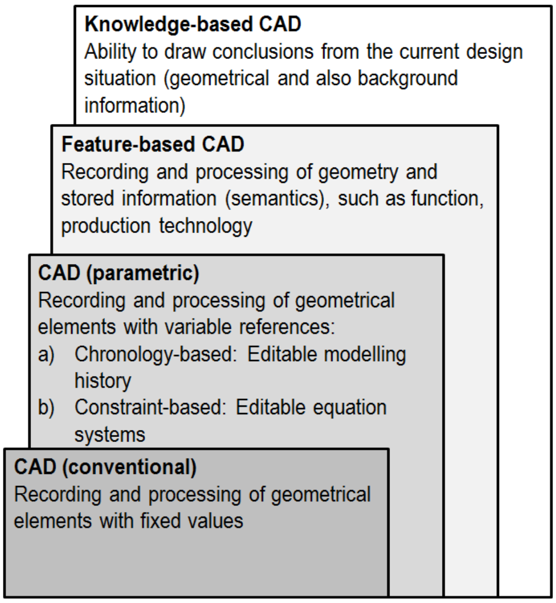

Fig. 2. *Overview of the principles of 3D modelling* [26]

The German VDI guideline 2209 [26] mentions two other types of CAD systems, which provide additional functionality for creating variable geometry models and for mapping design knowledge (fig. 2). Feature-based CAD systems are an extension of parametric systems. A feature can be understood as semantic information object which consists of multiple geometry elements with parametrics and behavioral rules [25]. Features can adapt themselves to their environment to a limited extent. Knowledge-based engineering (KBE) is the capstone in variable geometry modelling and aims at the automation of routine design tasks. Therefore, two different knowledge categories have to be considered (fig. 3): Domain knowledge describes a solution space in which a solution for a design problem may be found [27]. It may be expressed e.g. by dimensioning formulae that constrain parameters of the CAD-model, templates as reusable building blocks or design rules [28].

Control knowledge is different since it determines the way a solution space is explored. In literature this is also referred to inferences and reasoning since the system reasons about new or adapted requirements. Therefore, three different techniques may be used [13, 29]:

- **Rule-based reasoning**: The knowledge representation relies on IF-THEN-ELSE-statements. Rules are fired procedurally and can execute subordinate rules or delete them from the working memory in order to realize more complex tasks.
- **Model-based reasoning**: The possible solution space is described as physical and/or logical model (constraint-based) or by representation of resource consumption and allocation.
- **Case-based reasoning**: Here, knowledge is not explicitly modelled as rules or constraints. The knowledge necessary for reasoning is stored in cases that represent former approved solutions. A simple case-based reasoning system can either retrieve single already existing cases or assort a set of cases, which represent the best-fit. Highly developed ones are able of mixing or altering exiting cases in order to adapt them to new situations.

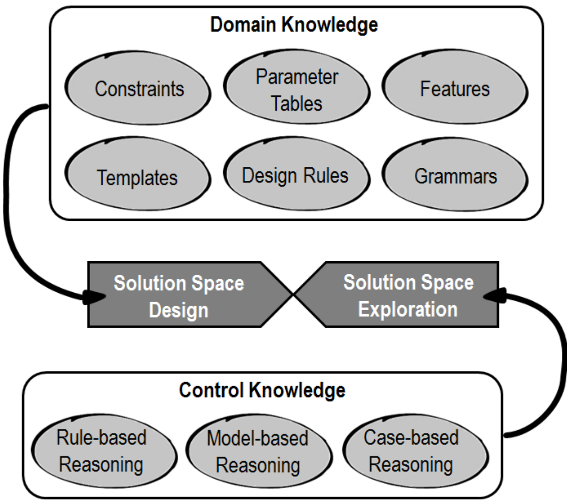

Fig. 3. *Knowledge Modelling in KBE and KBD*

2.3. Service Modeling

Compared to product development, there is just little software support in service development. The representation of services is realized by diagram-based methods which are data flow based, object-oriented or control flow-oriented. The service modeling includes the documentation and presentation of the processes as well as further necessary information like data which is needed in the process and organizations or people which are involved. The goals here are the targeted detection of weak spots (like media breaks within a process), or the analysis of certain properties of the processes (for example, throughput times or the costs of a process (activity costing)). In addition, the simulation of planned processes is possible with information about activities as well as further information such as throughput or set-up times and an exact process description [30, 31]

In the design of information systems the modeling language and the business process models become more and more important compared to the system implementation. They display the service processes and were already seen 15 years ago as important enabler to fulfill the requirements of generating, customizing and configuration techniques [32].

A modeling language for services is the event-driven process chain (EPC), which is based fundamentally on approaches of stochastic network procedures and Petri networks. The EPC has a high application orientation and therefore a high degree of dissemination and acceptance in science and practice. The EPC is regarded as the central modeling language of the architecture of integrated information systems (ARIS) [30].

The ARIS-approach by Scheer [32] originally provided a framework for the modeling of computer-aided information systems. ARIS offers a holistic view on process design, management, workflow and application processing through the generic methodological framework. The ARIS House (Fig. 4) forms the reference framework and is divided into five sections, which include the functional, organizational, data, performance and control views [32, 33, 34].

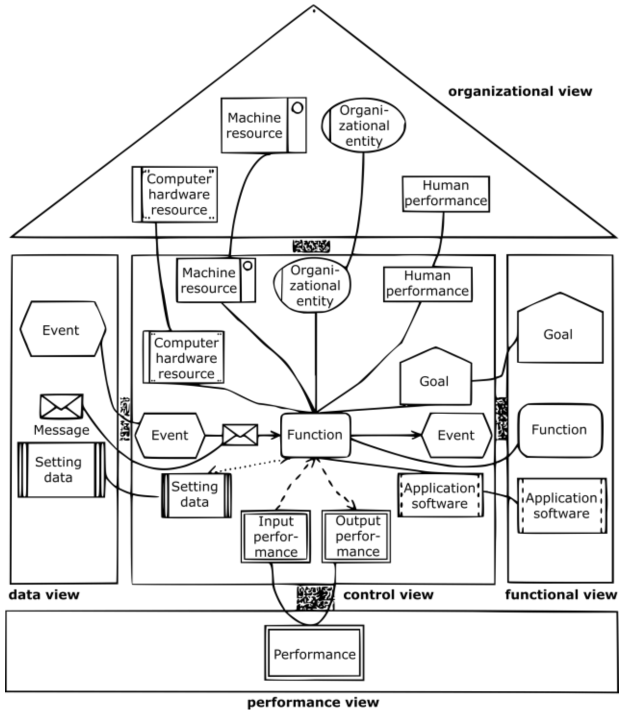

Fig. 4. *The five ARIS views (ARIS-house) based on Scheer [33]*

The area of the organization view (in Fig. 4 the roof of the ARIS-house) represents the organization in which the resources required to execute a function (here a task or activity to support one or more business goals associated with time and resource consumption) are assigned. In the area of the data view (left side in Fig. 4), the mapping and structuring of the information objects that are required or arise during the transformation process takes place. In the centric area of Fig. 4 (the control view) the elements of the four other views are arranged in the overall business process and placed in relationships. The function view (on the right side of Fig. 4) shows the processes that transform input performance into output performance. It also shows the goals that are closely related to the functions, as they are supported by functions but also control them. In the performance view (the basement in Fig. 4), the structural design of the tangible and intangible input and output performance required or created in the transformation process takes place [32, 33].

The modeling language EPC connects all the views in the ARIS-model and describes the dynamics of the business process [32]. In such a process model, the process-related relationship of functions is presented.

In the EPC, the functions are represented by the function block (see Fig. 5). Functions are triggered by an event and result in another event. Events are represented in the EPC by their own separate block (see Fig. 5) [35].

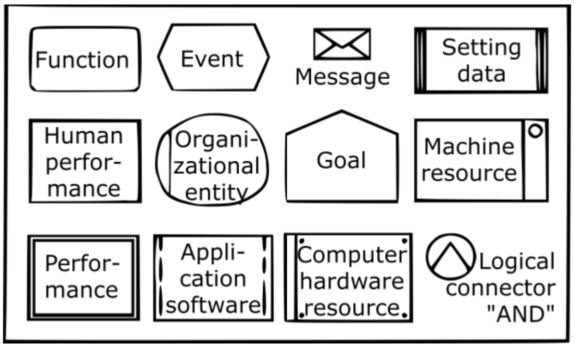

Fig. 5. *EPC blocks based on Scheer [32]*

Functions and events are linked by control flows. In addition, the connectors AND, OR and XOR are used. Control flows, connectors, events and functions are the basic building blocks of the EPC, which is extended by further flows and building blocks (see Fig. 5 and Fig. 6) [32].

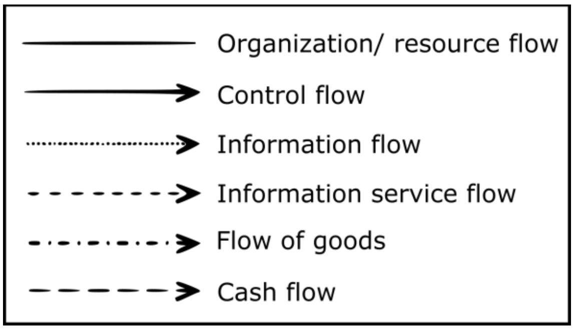

Fig. 6. *EPC flows based on Scheer [32]*

In Fig. 7 the EPC is shown by the example of a component inspection. First, the "component is submitted" (event), this triggers the function "component check". It is followed by the connector "exclusive or", since there is either the event "no errors found", or "error found". The event "errors found" is followed the function "rework component" which is followed by the event "Errors fixed". Afterwards an "exclusive or" linked this string to the second string and triggers the function "component output". This results in the event "component released" and the component inspection is completed.

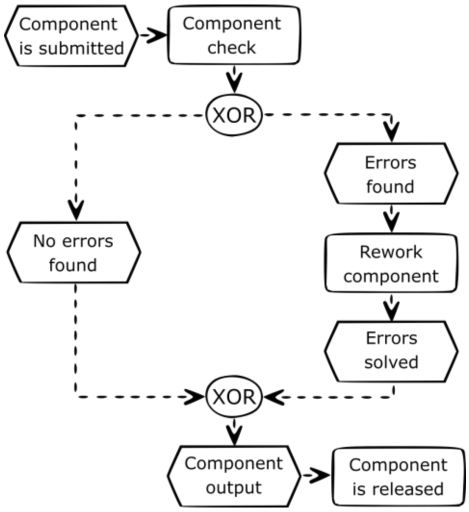

Fig. 7. *EPC example »component inspection«*

In addition to the EPK, there are other languages, such as the BPMN (Business Process Modeling Notation). But the application of the BPMN is limited to process modeling and for example the mapping of further information about used infrastructure or resources are not covered by the language [34].

2.4. PSS Configuration

As mentioned in section 2.1 the addressing of the individual customers and their needs is an important point for PSS development. To achive this with an acceptable workload is to provide configurable PSS [12]. The main approaches of the configuration of PSS, which are discussed in the literature, will be shown in the following. A detailled discussion can be found in [12].

Laurischkat [8] concentrates in her approach on the configuration of service components of PSS which is modeled by decision tables or production rules. Bochnig et al. [9] published an integrated PSS development approach with the possibility of PSS configuration. They introduced a CAE tool, in which variants are generated by combining existing PSS modules. In their work, Aurich et al. [36] identified modularization as a promising strategy, they use combination matrices and focuses on possible product and service architectures for PSS. The approach of Mannweiler [10] presents the idea of a development of industrial PSS with predefined blocks which are predominantly product components.

Compared to the three reasoning techniques for CAD-based solution space modeling which were shown in section 2.2 there are only two of them used in the approaches documented in the PSS literature. So far the identified PSS configuration approaches uses ether rule-based or case-based techniques. For a model-based configuration of PSS a constraint-based PSS model is necessary. To develop a constraint-based model the approach of Steinbach should be used as a starting point as evaluated in [7].

Steinbach's work [37] adapts the definitions of characteristics and properties of Weber's [38] Characteristics Properties Modeling/ Properties Driven Development (CPM/ PDD) approach to PSS. The approach of Weber models products from a developer point of view (the characteristics (Ci) which can be affected by the developer) and from a customer point of view (the properties (Pi) which are noticed by the customer). These are connected and translated by relations (Ri) which are restricted by external conditions (ECi) (shown in Fig. 8) [37, 38].

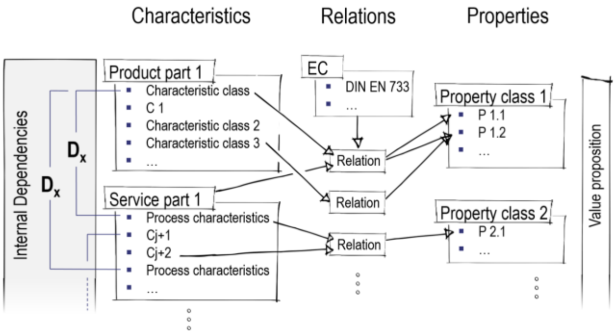

Fig. 8. *PDD for PSS based on Steinbach [37]*

For PSS, Steinbach extends the model with internal relations of product and service parts which can also be visualized (shown in Fig. 8 on the left side). The product parts are subdivided in characteristic classes and the service parts in process characteristics. The internal relations mostly exist between characteristic classes and process characteristics [37].

The approach provides a good theoretical foundation for further research on the representation and modeling of PSS. It helps to capture the PSS and to sketch it in an integrated way [18]. The internal dependencies as well as the relations including the external conditions can be used to create a constaint-based CAD model.

So far there is no documented example of a realization in a CAD environment. Within the scope of this paper this gap is closed, Steinbach is used as theoretical basis and extended with further methods from product and service development. On the one hand, the ideas of solution space modeling for constraint models for physical artifacts (introduced in section 2.2) are used. On the other hand, the ARIS approach (with the EPC) are used for service part modeling, due to the fact, that the EPC is already seen as an important tool for generating, customizing and configuring services in the area of pure service development(see section 2.3). Based on these three principles, an approach is presented that closes the gap of a contraint-based PSS configuration. A basis for this is a solution space modeled due a joint PSS CAD model with coequal service and product parts.

3. MODEL-BASED CONFIGURATION APPROACH FOR PSS

After the theoretical basics, the approach for a model-based configuration of a PSS is presented. In the following, the procedure of the procedure of the approach is described and a graphical overview is given in figure 9.

For the schematic representation of the PSS, the approach based on the work of Weber and Steinbach [37] is used. In this the needs of the customer (the so-called properties) are represented. These needs are formulated independently of a specific product or service, with the target to develop these coequal afterwards. The properties addresses the value proposition of the PSS, which is related to the business model of the PSS provider.

The decision which business model is targeted and therefore where in the characterisation of Tukker the offered PSS can be classified has an influence on the requirements for the PSS and thus also its development. However, these decisions are made before or in early stages of the development of the actual PSS and are not considered in detail in this paper but are included as given in the requirements. To take the individual requirements of the different customers into consideration, all possible requirements have to be considered. This creates a kind of requirement space.

The properties of the customers are linked by the relations with the characteristics of the PSS components. These components and their "characteristic classes" (product parts) and "process characteristics" (service parts) are designed in a way that all properties are indirectly controlled by them.

Possible tools for capturing the PSS components and their "characteristic classes" and "process characteristics" are excerpts from the specifications for individual physical artefacts and the representation of the services by the EPC including the associated resources, etc.

It is important here that the internal dependencies between "characteristic classes" and "process characteristics", i.e. between the product and service components of the PSS, are already part of the design. This leads to the situation, that the interactions (internal dependencies) are taken in to consideration at an early stage and the developer is forced to develop product and service components on an equal foundation and related to each other. To work out the internal dependencies, tools such as the creation of constraint networks, a parameter plan, or the formulation of rules are used.

In order to present the PSS in a data model, the product and service parts, including the "characteristic classes" and "process characteristics" and the internal dependencies, are transferred to a CAD system. The Autodesk Inventor Professional system is used for this purpose, since Gembarski et Al. already showed that building technical product configurators or engineering configurators within a CAD environment is possible [39]. Based on this the technical system will be extended to a PSS.

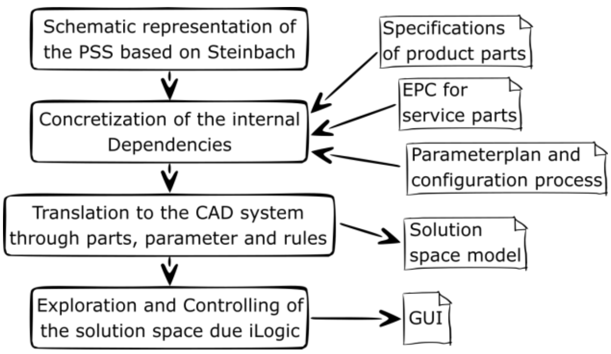

Fig. 9. *Procedure of the approach*

To transmit the PSS into the CAD system the possibility to insert parameters into the system is used. This can be done using the implementation of MS excel spreadsheets, or directly in Inventor's parameter tables. In order to integrate the service into the model, the component parameters are expanded in a first step with the help of user parameters. For the components in the CAD system, model parameter, reference parameter or user parameter can be defined (they can be numerical, boolean or text variables). The added parameters contain information such as e.g. the assembly complexity, required tools for mounting the part, or maintenance intervals. The individual parts of the PSS are combined in CAD assemblies (similar to physical products). The main assembly contains all PSS subassemblies and parts including the further informations which are deposited in each part.

In the main assembly Inventor iLogic is used to create a GUI on which further information and choices to the services are realized. The iLogic environment is used to integrate service parts and to model the solution space of the PSS. The iLogic programming language uses constructs like if-then-else rules, select-case decision trees, while loops, and offers sub procedures and a class concept. This language is similar to common script languages. As command library the snippets include code templates for almost every modeling context within Inventor.

The service informations and choise possibilities realized by iLogic are related and attached to the parameter in the model. A more detailed description of this is given in Section 5 in which the application example is documented.

4. APPLICATION EXAMPLE

For a better understanding, the approach presented in section 4 is now used at an application example.

4.1. Description of the example

As an example, an end-suction centrifugal pump according to DIN EN 733 [40] is used. Just with the pump as a product the example is located on the left side in the characterization of Tukker (see section 3.1). By adding service which raises the service-value-ratio a PSS is created. So a product oriented PSS is possible with maintenance contracts added to the pump. Additionally, result oriented or use oriented PSS are possible, i.e. the customer pays for a guaranteed pump performance, independent of the material or service that is required.

For a better understanding the example is restricted to one (product oriented) PSS which contains a defined pump as the product part and additional services. The added services are: pump maintenance, damage repair and empowering the user for the repair.

4.1.1 Product part of the example

The product part of the PSS consists of the already mentioned end-suction centrifugal pump. Other physical parts such as the pipe system, or the drive of the pump could expand the PSS, but in this example they are not considered and taken as given. The pump is driven by an engine through the shaft, which is mounted in the housing and on whose end an impeller is mounted.

The pipe is axially connected to the inlet of the housing of the pump, through the impeller, the fluid is sucked and passed through the outlet of the pump back into the pipe system. In the example shown here, the focus is on the impeller of the pump and its installation and disassembly.

4.1.2 Service part of the example

As service part of the example the disassembly of the impeller is shown in excerpts. Two types of service are taken in consideration, first the process of the disassembly by a trained technician (no, or little support) and second the process of empowering the user for the repair (maximum support and protection against faulty operation).

4.2. Realization of the example

For the realization of the example, the four steps of the procedure presented in section 3 are executed.

4.2.1 Schematic representation

In the first step the PSS is sketched schematically with the approach of Steinbach (section 2). Fig. 10 shows a part of the system. It starts on the right side with the value proposition of the PSS, a guaranteed pump performance. At the properties level the required space or a maximum downtime is noted. The properties are connected to the characteristics through relations which are restricted by external conditions, in this example among others obviously the DIN EN 733 [40].

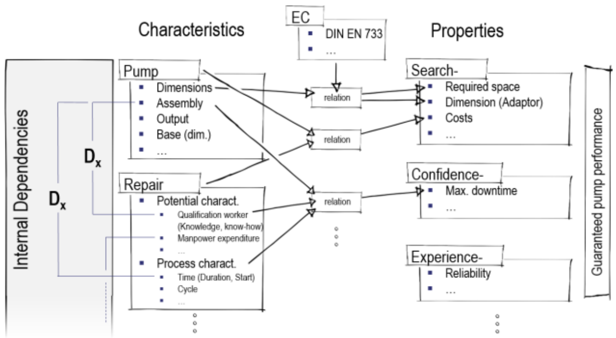

Fig. 10. *Detail from the schematic PSS depiction*

Next the characteristics level is shown, it includes for example the product part with the dimensions of the physical artefacts, or the assembly structure which can be manipulated directly by the developer. Beside the product part also the service parts of the PSS take place in the characteristic level. In fig. 10 the repair is shown with potential characteristics like the qualification of the worker and with process characteristics like time information.

On the left side of the representation the internal relations are displayed, where the relationship between "Assembly" and "Qualification worker" and "Assembly" and "Duration" is presented. The correlation of the complexity of the used components (and their assembly) and the education level of the technician is important when an appropiate technician is to be chosen as well as the amount of needed support. Also the dependencies of the assembly complexity and the duration of the service (an assembly with a higher complexity results in a higher duration) is formulated through internal dependencies.

4.2.2 Concretization of internal dependencies

To concretize the internal dependencies the second step of the procedure follows. With help of the specification, the characteristics of the product parts are elaborated. These characteristics are for example the dimensions of the artifacts which are defined among others by the length with a concrete value or a defined parameter which represents a value range. But also characteristics like maintenance intervals, the expected part lifetime, the monting difficulty, or the position in the mounting sequence. The characteristics of the service parts are developed by means of the EPC (section 2.4).

As mentioned before the service focuses on the disassembly of the impeller. As an example, the EPC for the process of the disassembly by a trained technician is shown (see fig. 11). Furthermore a part of a technical drawing of the pump with item numbers is depicted. The numbers are used in the EPC for a better orientation when one of the items is named.

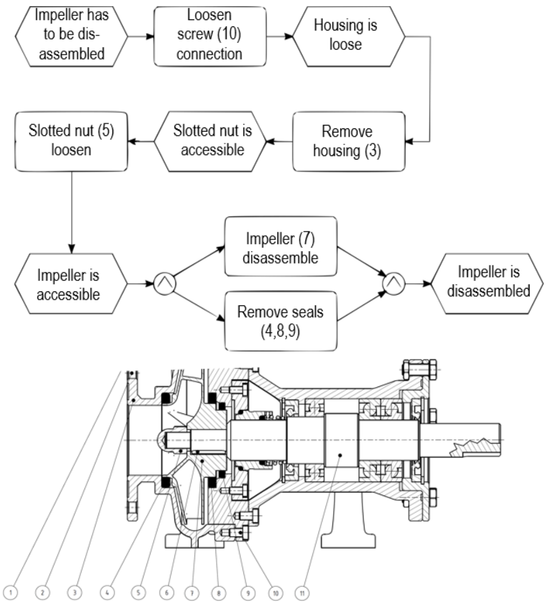

Fig. 11. *EPC of the Impeller disassembly*

The EPC starts with the event "Impeller has to be disassembled" which triggers the function "Loosen screw connection". This results in the event "Housing is loose", and is followed by further functions and events. The EPC ends with the event "Impeller is disassembled", which is preceded by an "and" connector, connecting the two previous functions "Demounting impeller" and "Demounting seals".

Fig. 12 shows the extension of the EPC on high detail level for the function "Loosen screw connection" in case of empowering the user for the repair service.

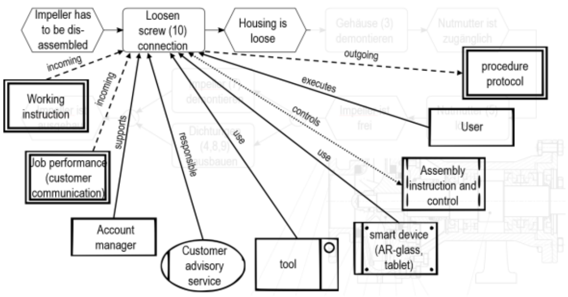

Fig. 12. *EPC extension with high detail level*

The extension takes place on the one hand by incoming ("Working instruction" and "Customer communication") and outgoing (entry into "Procedure protocol") information service flows. The human work performance of the "Account manager" ("supports") and the "User" ("executes") enter the function via control flows. The "responsibility" of the organizational unit "Customer advisory service" and the "use" of the tool "Screw-wrench" and the computer hardware "smart device" are also incoming through control flows. Incoming and outgoing is the information flow of the application software "Assembly instruction and control".

In addition to the EPC and specifications, the parameter plan and the configuration process plan are also used for specifying the internal dependencies. In the parameter plan, all parameters are listed, how they are passed on within the model and by which constraints and rules (global and local) they are connected to each other. An example of this is the required flow rate of the pump. The flow rate leads a certain norm impeller, which determines the diameter of the shaft at the corresponding shoulder by means of its standardized inner diameter.

The configuration process plan is a formal model that visualizes the process of the configuration and contains actions (or functions), input parameters and decision points.

In the next step the product and service parts including all parameter as well as the constraints and rules are translated to the CAD system (fig. 13).

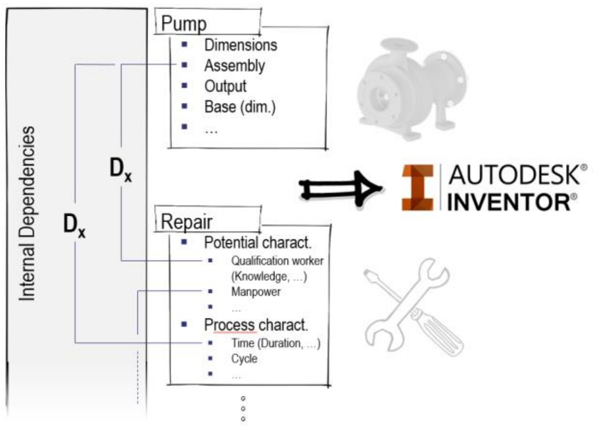

Fig. 13. *Transfer of the schematic PSS to a CAD-system*

4.2.3 Translation to the CAD system

To build a holistic data model in the CAD system the physical artifacts are created (as parts in Inventor). In the part document the parameters of each artifact are managed by the parameter table. In fig. 14. the part „impeller“ is shown including the parameter table which contains the physical part parameter like the geometric information and in the section “user parameter” also the extended information.

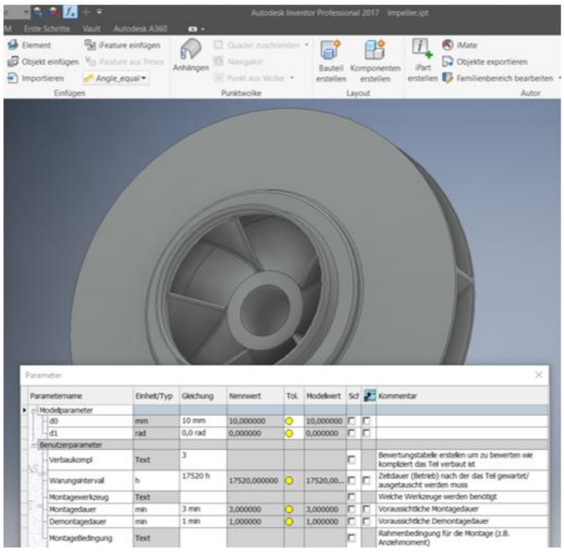

Fig. 14. *EPC extention with high detail level*

The extended information connected to the impeller are for example “assembly complexity” (which is related to the service technician), “maintenance interval” (which contains the information after how many working hours the part should be checked or changed) or “mounting tools” (which contains the information what kind of tools are needed). In the comment field these parameter are described in a short sentence and linked services can be named for a better orientation. This is executed analogical for the other parts of the pump.

The parts will be assembled in the main assembly, which includes all parts of the pump as well as the related parameters, so it contains all information. Additionally it also contains the GUI realized by the iLogic environment.

4.2.4 Exploration of the solution space (GUI)

The exploration and controlling of the solution space is realized with a configurator in the main assembly. The GUI is based on iLogic and on information which are stored in iLogic (by formulas and rules) as well as the linked parameters of the parts. Fig. 15 shows the GUI with which the PSS can be configured and the solution space is spanned. In the first step, the component and the type of service can be selected via the GUI. In a next step, the more precise specification of the service takes place, for the repair, for example, it is determined who executes it (either the service technician, or the user by itself).

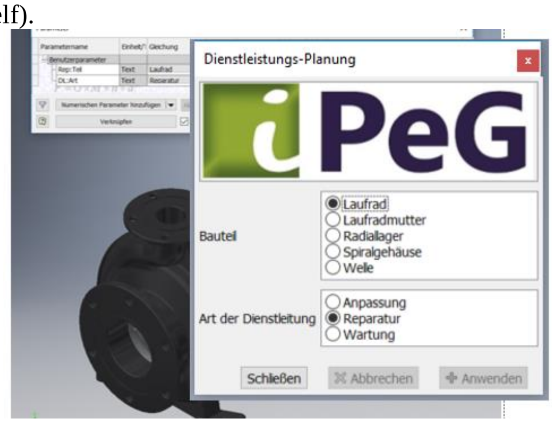

Fig. 15. *GUI of the PSS solution space*

An example of the constraints of product and service parts and how it is realized in the model is the relation of the assembly complexity and the education of the service technician or the user. In the configurator it is possible to choose a part (in fig. 15 the impeller) and type of service (repair). The next step is to assess whether this service can be done by the user or whether it has to be done by a service technician. For this purpose, different parameters for the qualification level of the users and technicians are defined (in iLogic) as well as parameters describing the assembly complexity of the parts, assigned to each part.

These parameters are compared by rules and thus it is to determine whether a repair can be executed by a person. If this is not possible, a higher qualification level may be required (service technician), or the person with a low qualification level (user) may be supported by assistance such as assembly instructions and tool or part lists.

Finally, the structure of the described data model in the CAD environment is shown schematically in figure 16. In the middle of the representation the PSS model is displayed which includes product (e.g. pump) and service parts (e.g. repair), such affiliations are indicated by lined arrows in the figure. The service includes service parts (e.g. disassembly of impeller) or activities (e.g. loosen screw), all of them are represented by ovals. The products as well as the included assemblies (e.g. shaft assembly) and parts (e.g. impeller) are represented by rectangles with rounded corners. Knowledge and data are represented in the figure by rectangles and the flow of information or data by dashed arrows.

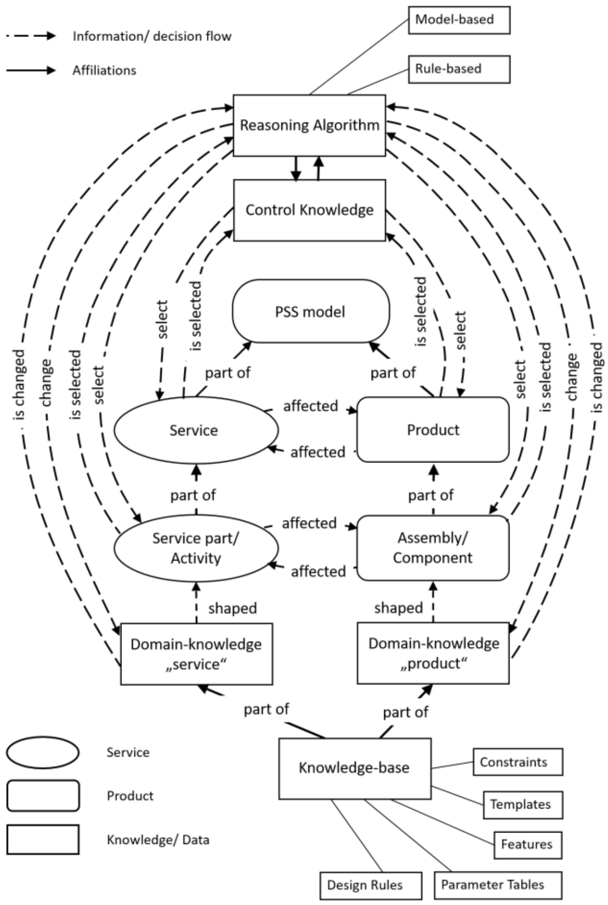

Fig. 16. *Shematic Representation of the datamodel*

With the control knowledge the PSS is defined and the included product and services are selected (by the GUI). Part of the control knowledge is the reasoning algorithm (realized by iLogic code) which select the necessary service and product parts and components. The knowledge-base which is part the domain-knowledge, can be realized by constraints, templates, features, design rules or parameter tables (e.g. by the extension of the part parameter by parameter tables or rules realized by iLogic code).

5. CONCLUSION AND FURTHER RESEARCH

In this paper, first the need for solution space modeling for PSS was outlined. Subsequently, the solutions space modeling of physical artifacts and ARIS with the EPC as a method for service modeling were considered. A promising approach for the development of a constraint-based CAD model for a model-based configuration was shown and extended with solutions space modeling of physical artifacts and ARIS. An approach to solution space modeling for PSS was presented and an application example described.

The description was a simple application example with only one product and a limited number of services. In the following research the method will be executed with bigger and more complex systems, furthermore a detailed description of the implementation of the model in the CAD environment will take place including a procedure model.

The approach shown in this paper will be further elaborated and detailed in the future and evaluated in further application examples. Furthermore, an extension of the model with more information, such as the integration of priority graphs and an automated optimization of the system, for example with regard to the adjustment of maintenance intervals of adjacent components, must be reviewed.

6. 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.

7. REFERENCES

[1] Y. Koren, "The global manufacturing revolution: product-process-business integration and reconfigurable systems" in *Wiley series in systems engineering and management*. Wiley, Hoboken, N.J. 2010.

[2] O. Thomas, P. Walter, P. Loos, "Konstruktion und Anwendung einer Entwicklungsmethodik für Product-Service Systems" in *Hybride Wertschöpfung*. Springer, Berlin, Heidelberg, 2010.

[3] F. Sturm, A. Bading, M. Schubert, *Investitionsgüterhersteller auf dem Weg zum Lösungsanbieter: eine empirische Studie ; fit2solve.* IAT, Univ, Stuttgart. 2007.

[4] J.C. Aurich, C. Fuchs, C. Wagenknecht, "Life cycle oriented design of technical Product-Service Systems" in, *Journal of Cleaner Production* no. 17, pp. 1480–1494, 2006.

[5] O.K Mont "Clarifying the concept of product–service system" in, *Journal of Cleaner Production* no. 3, pp. 237–245, 2002.

[6] P. Müller, R. Stark, *Integrated engineering of products and services: [layer-based development methodology for product-service systems]*, Berlin, Stuttgart, Fraunhofer Verlag, 2014.

[7] D. Spath, L. Demuß, "Entwicklung hybrider Produkte — Gestaltung materieller und immaterieller Leistungsbündel" in, *Service Engineering*, Springer, Berlin, Heidelberg, pp. 463–502, 2006.

[8] K. Laurischkat, "Product-Service Systems: IT-gestützte Generierung und Modellierung von PSS-Dienstleistungsanteilen" in *Schriftenreihe des Lehrstuhls für Produktionssysteme,* Ruhr-Universität Bochum. Shaker, Aachen, 2012.

[9] H. Bochnig, E. Uhlmann, A. Ziefle, "Assistenzsystem – IPSS-CAD als informationstechnische Unterstützung der integrierten Sach- und Dienstleistungsentwicklung in der IPSS-Entwurfsphase" in, *Industrielle Produkt-Service Systeme,* Springer, Berlin, Heidelberg, pp. 95–115, 2017.

[10] C. Mannweiler, *Konfiguration investiver Produkt-Service Systeme*, Techn. Univ, Kaiserslautern, 2014.

[11] F. Salvador, P. M. De Holan, F. Piller, "Cracking the code of mass customization" in *MIT Sloan management review* no. 50, pp. 71-78, Cambridge, MA, 2009.

[12] P.C. Gembarski, R. Lachmayer, "Mass Customization und Product-Service-Systems: Vergleich der Unternehmenstypen und der Entwicklungsumgebungen" in *Smart Service Engineering*. Springer Fachmedien Wiesbaden, Wiesbaden, 2017.

[13] L. Hvam, N.H. Mortensen, J. Riis, *Product customization*. Springer Science & Business Media, 2008.

[14] K. Laurischkat, "Product-Service Systems: IT-gestützte Generierung und Modellierung von PSS-Dienstleistungsanteilen" in *Schriftenreihe des Lehrstuhls für Produktionssysteme,* Ruhr-Universität Bochum. Shaker, Aachen, 2012.

[15] D. Schreiber, P.C. Gembarski, R. Lachmayer, "Datamodels for PSS Development and Configuration: Existing Approaches and Future Research" in *9th World Conference on Mass Customization, Personalization and Co-Creation (MCPC 2017),* Aachen, 2017.

[16] L. Wagner, D. Baureis, J. Warschat, "Developing Product-Service Systems with InnoFunc®" in *International Journal of Industrial Engineering and Management (IJIEM)* no. 1, pp. 1–9, 2013.

[17] L.T.M. Blessing, A. Chakrabarti, "*DRM, a Design Research Methodology*" Springer, London, 2009.

[18] D. Schreiber, P.C. Gembarski, R. Lachmayer, "Modeling and configuration for Product-Service Systems: State of the art and future research" in,

Proceedings of the 19th International Configuration Workshop, Paris, 2017.
[19] N. Morelli, “Developing New Product Service Systems (PSS): Methodologies and Operational Tools” in, *Journal of Cleaner Productio*, no. 17, 2006.
[20] H. Meier, R. Roy, G. Seliger, “Industrial Product-Service SystemsIPS2” in, *CIRP Annals - Manufacturing Technology*, no. 2, 2010.
[21] A. Tukker, “Eight types of product–service system: eight ways to sustainability? Experiences from SusProNet” in *Business Strategy and the Environment* no. 13, pp. 246–260, 2004.
[22] M. Boehm, O. Thomas, “Looking beyond the rim of one’s teacup: a multidisciplinary literature review of Product-Service Systems, in Information Systems, Business Management, and Engineering & Design” in, *Journal of Cleaner Production* no. 51, pp. 245–260, 2013.
[23] M. Gräßle, O. Thomas, M. Fellmann, J. Krumeich, “Vorgehensmodelle des Product-Service Systems Engineering: Überblick, Klassifikation und Vergleich“ *Integration von Produkt und Dienstleistung - Hybride Wertschöpfung*, MKWI, 2010.
[24] J.J. Shah, “Designing with parametric cad: Classification and comparison of construction techniques” in, *Geometric Modelling – Proceedings of the Sixth International Workshop on Geometric Modelling*, Tokyo, Japan, December 7–9, 1998, pp. 53-68, 2001.
[25] M. Hirz, *Integrated Computer-aided design in automotive development*, Berlin, Heidelberg, Germany: Springer, 2013.
[26] VDI, *VDI Guideline 2209 - 3D Product Modelling*, Berlin, Germany: Beuth, 2009.
[27] N.R. Milton, *Knowledge Technologies*, Monza, Italy: Polimetrica sas, 2008.
[28] J.J. Cox, “Product Templates - A Parametric Approach to Mass Customization” in *CAD Tools and Algorithms for Product Design*. Berlin, Heidelberg, Germany: Springer, pp. 3-15, 2000.
[29] D. Sabin, R. Weigel, “Product configuration frameworks - a survey” in IEEE intelligent systems on. 13.4, pp. 42-49, 1998.
[30] A. Gadatsch, Grundkurs Geschäftsprozess-Management. Vieweg+Teubner Verlag, Wiesbaden, 2013.
[31] M. Nüttgens, F.J. Rump, “Syntax und Semantikreignisgesteuerter Prozessketten (EPK)“ in, Prozessorientierte Methoden Und Werkzeuge Für Die Entwicklung von Informationssystemen – Promise, Potsdam, 2002.
[32] A.-W. Scheer, ARIS — Vom Geschäftsprozess zum Anwendungssystem. Springer Berlin Heidelberg, 2002.
[33]DIN, DIN-Fachbericht 75 - Entwicklungsbegleitende Normung (EBN) für Dienstleistungen, Berlin, Germany: Beuth, 1998.
[34] M. Lankhorst, Enterprise Architecture at Work, The Enterprise Engineering Series. Springer, Berlin, Heidelberg, 2013.
[35] G. Keller, M. Nüttgens, A.-W. Scheer, “Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK)“ in, Veröffentlichungen Des Instituts Für Wirtschaftsinformatik, Saarbrücken, 1992.
[36] J.C. Aurich, N. Wolf, M. Siener, E. Schweitzer, “Configuration of product service systems” in, Journal of Manufacturing Technology Management no. 20, pp. 591–605, 2009.
[37] M. Steinbach, Systematische Gestaltung von Product-Service-Systems: integrierte Entwicklung von Product-Service-Systems auf Basis der Lehre von Merkmalen und Eigenschaften, Saarbrücken, 2005.
[38] C. Weber, “CPM/PDD – An Extended Theoretical Approach to Modelling Products and Product Development Processes.” In, Advances in Methods and Systems for the Development of Products and Processes. at the 2. German-Israeli Symposium on Design and Manufacture, Fraunhofer IRB, Stuttgart, pp. 159–179, 2005.
[39] P.C. Gembarski, H. Li, R. Lachmayer, “KBE-Modeling Techniques in Standard CAD-Systems: Case Study—Autodesk Inventor Professional” in, Managing Complexity. Springer International Publishing, Cham, pp. 215–233, 2017.
[40] DIN, DIN EN 733 – End-suction centrifugal pumps, rating 10 bar with bearing bracke, Brüssel: Comite Europeen de Normalisation, 1995.

CORRESPONDENCE

Daniel Schreiber, M.Sc.
Institute of Product Development
Leibniz University Hanover,
Welfengarten 1A,
30167 Hanover, Germany
schreiber@ipeg.uni-hannover.de

Dipl.-Ing. (FH) Paul Gembarski
Institute of Product Development
Leibniz University Hanover,
Welfengarten 1A,
30167 Hanover, Germany
gemarski@ipeg.uni-hannover.de

Prof. Dr.-Ing. Roland Lachmayer
Institute of Product Development
Leibniz University Hanover,
Welfengarten 1A,
30167 Hanover, Germany
lachmayer@ipeg.uni-hannover.de