Enrichment of Geometric CAD Models for Service Configuration

Daniel Kloock-Schreiber [1] and **Lukas Domarkas** and **Paul Christoph Gembarski** and **Roland Lachmayer**

Abstract. The boundaries between products and services are vanishing and offers such as hybrid product service bundles, are becoming increasingly important. These solutions are referred to as Product Service Systems (PSS), which address individual customer needs as problem-oriented solutions.
In order to enable the configuration of such systems as well as the possibility of planning and supporting services on the basis of a holistic model, a data model that contains both product and service information as well as their dependencies, is required. For this purpose, existing CAD models must be extended with further information. These are e.g. maintenance intervals of parts, costs of the parts, or also index numbers for the complexity of installation. This paper shows how to enrich a CAD model by integrating information into it and creating an interface with Excel. Thereby, the model can be used by different actors in the PSS for configuration and development, service planning and support of service technicians. Finally the approach for enriching a CAD model is implemented using the example of an engine test bench.

1 INTRODUCTION

In mechanical and plant engineering it is becoming increasingly difficult for a company to distinguish the offering from competitors only by technical product features [14]. A development from the recent years is extending and strengthening the (after-sales) service activities. Since service has not to be seen as an add-on in order to leverage its full economic potential, a joint development of product and service is beneficial. In the scientific literature, this is introduced and discussed under the term of "Product-Service Systems" (PSS) [18, 33]. Literature describes PSS as solutions that meet individual customer needs, regardless of whether the value proposition and revenue are primarily achieved through the product or service components [30, 32]. PSS may be regarded as customer specific problem solution. As such, relations between product and service components must be taken into account during development. In order to reduce development and adaptation costs, the configuration of PSS is a possible way [1, 3, 15, 17].

1.1 Motivation and Aim

Due to the conceptual similarity of the enterprise types PSS and MC, PSS can be understood as a MC offer and thus MC development processes and modeling tools can be applied to PSS [8]. One of the the key principles of MC is the solution space modeling. The development and configuration of PSS can benefit from MC techniques like choice navigation and solution space modelling [22]. In order to deal with the upcoming complexity and to allow co-creation between PSS-supplier and consumer, the application of Knowledge-based-Engineering (KBE) and the implementation of reasoning mechanisms into product models is a promising approach [7, 12].
In the area of Mass Customization (MC), solution spaces and product configurators for physical products have already been described, furthermore there are already approaches to service configuration. For example, there are papers dealing with the bidding process and configuration ([10]) and the effects to assemble/make-to-order up to engineer-to-order situations ([31]). In this article, however, the focus is on services that occur at a later point in time, the service is regarded as a component of the usage phase of products (e.g. maintenance and repair as well as documentation of existing product versions).
With the Service Explorer, Sakao provides a computer-aided service modelling tool based on a provider-consumer system. The main point of this approach is to change the state of the receiver. In the system, the requirements and condition of a buyer are first modeled and transformation rules are designed based on these[21]. But without effect or direct dependence to the physical product model.
In the PSS literature rule-based and case-based configurators can be found (e.g. in the work of Laurischkat [16]), but a model-based configuration for PSS is missing [28]. For such a configuration a parametric model is needed that represents product and service parts of a PSS and also documents all their dependencies. Using a rule-based or case-based configurator without a parametric model leads to a very high effort in the creation, or to the fact that the configurators only operate with a small data base and therefore cannot use their strengths or only use them to a limited extent.
As mentioned by Wagner [35], it is an important prerequisite for the development of PSS to adequately combine product and service parts with all their dependencies [35]. In the area of MC and configuration existing domain models which are suitable for the development of solution spaces for products.
Important factors for the design of PSS is the coequal development of product and service and the addressing of individual customers and their needs. To realize a coequal product and service development as well as the configuration of the system for service planning and support, an enriched CAD model is a promising approach. Such a CAD model can be a start for a constraint-based model which includes the data about the physical product as well as service data. Beside the CAD-model this service data is part of a modeling language and process model of the service. They map the service processes and are an important prerequisite for meeting the requirements of the generation, customizing, and configuration techniques [23].
An approach how this can be build on CAD-model and extended with the event-driven process chain (EPC), will be shown in this paper.

[1] Leibniz University of Hannover, Institute of Product Development, email: schreiber@ipeg.uni-hannover.de

The EPC is a modeling language that can be parameterized within limits and is therefore well suited to extend a (physical) CAD model, because CAD and EPC can be modelled similarly and knowledge can be integrated into this model in a similar way (e.g. by formulas, rules). The EPC can be used with single extensible templates up to the parameterization of the displayed services. The discussion and outlook takes a look how this approach will be transfered in to a domain approach with a constraint-based model.

1.2 Structure of the Paper

In the following section 2, the theoretical background to PSS from the literature is presented, as well as an overview of geometry based solution space modeling in modern CAD systems, service modeling and existing approaches for PSS configuration. Based on this state of the art in section 3 the enrichment of CAD data is described and how a data model can be built up. As well as the description how this model can be used for service modeling and for the support of the service planing and execution. Afterwards in section 4 an exemplary implementation for a HIL (hardware in the loop) test bench provider, which offers engine test benches, is described. The paper is concluded with a discussion and an outlook on further research potentials in section 5.

2 THEORETICAL BACKGROUND

2.1 Product Service Systems

The literature agrees that the quality of a PSS is influenced by the structure of the PSS development process [32]. In order to respond to individual customer needs and since a combined product and service development is necessary, a PSS-specific development is necessary. The literature agrees, but the existing approaches remain mostly vague and conceptual [4]. In addition, they are discussed using very simple or very concrete examples, which makes it difficult to transfer them to relevant applications [8]. Its multidisciplinarity, which involve researchers from different fields of interest is a challenge for the research and development of PSS. With respect to the evaluation of existing approaches, none of them can be regarded as a generally accepted and standardized approach to the development of PSS [9]. However, based on literature studies (documented in earlier papers [8, 24]) on the existing characterizations, the existing theses and approaches in PSS design research, the following main implications for PSS development can be identified [26]:

- coequal development of product and service components
- integration and addressing of individual customers and their needs
- monitoring and addressing of the customers requirements during the whole life-cycle of the PSS

2.2 CAD-based Solution Space Modeling

A parametric CAD, in contrast to rigid (conventional) geometry modeling, is able to represent a solution space. To do this, knowledge must be explicitly translated into digital prototypes. This is made possible by the parametric systems in particular by the fact that mathematical and logical constraints and boundary conditions can be defined between the parameters in a CAD system. For development, the designer must not only specify the product shape, but also the variant design and the associated control and configuration concept for the components. Thus a solution space is described by the developer [27, 11].

In addition to the above mentioned CAD systems (conventional and parametric), VDI Guideline 2209 [34] includes two other types of CAD systems that provide additional functionality for creating variable geometry models and mapping design knowledge (see figure 1). Feature-based systems are an extension of parametric CAD systems.

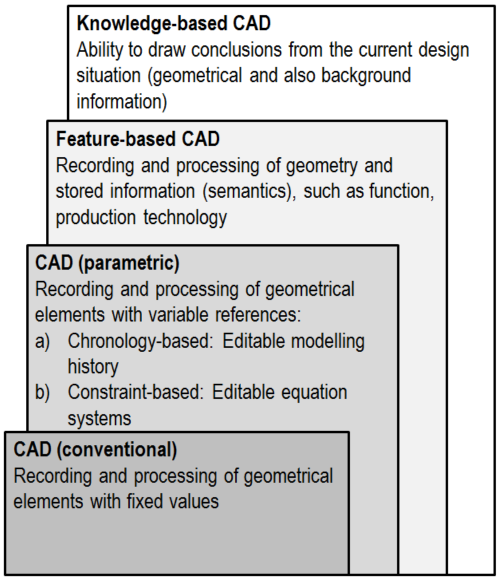

Figure 1. Overview of the principles of 3D modeling [34]

A feature consists of several geometric elements with parametrics and behavior rules and can be understood as a semantic information object [11]. Features can (to a limited extent) adapt to their environment.
The fourth principle is the knowledge-based engineering (KBE) with the ability to draw conclusions from the current design situation (geometrical and also background informations). It aims the automation of routine design tasks. To realize this two different knowledge categories have to be considered, which are shown in figure 2, domain knowledge and control knowledge.

The Domain knowledge describes a solution space build up with constraints (e.g. by dimensioning formulas that constrain parameters of the CAD-model), templates (as reusable building blocks), parameter tables, features, design rules or grammars. In this solution space a suitable solution for a design problem may be found [20, 5].
The control knowledge is the knowledge which determines the way a solution space is explored. In literature it is referred to inferences and reasoning techniques to adapted the system to new or adapted requirements. Basically, three different techniques may be used [12, 20]:

- Rule-based reasoning: Rules are executed procedurally and can perform subordinate rules or delete them from the working storage in order to realize more complex tasks. The knowledge representation is based on IF-THEN-ELSE-statements.
- Model-based reasoning: The possible solution space is described as a constraint-based physical and/or logical model or by the representation of allocation and resource consumption.
- Case-based reasoning: The knowledge is not explicitly modeled as a constraint based model or by a rule based. The knowledge necessary for reasoning is stored in examples (former approved solutions). A simple case-based reasoning system can assort a set

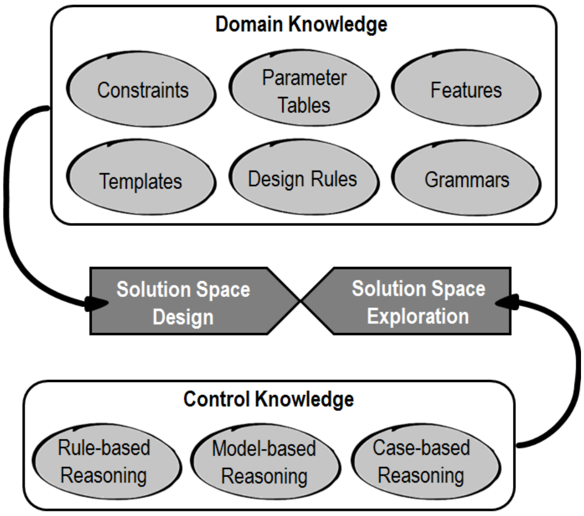

Figure 2. Knowledge Modeling in KBE and KBD [25]

of cases, which represent the best-fit or retrieve single already existing cases. Highly developed systems can mix or alter exiting cases and adapt them to new situations.

2.3 Service Modeling

In service development exist just little software support compared to product development. For displaying services diagram-based methods are used (these can be data flow based, object-oriented or control flow-oriented). For service modeling the documentation and presentation of the processes is necessary as well as further information like data needed in the process and involved organizations or people. Goals of the modeling are the targeted detection of weak spots which can be 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)). Furthermore, the simulation of processes is possible with information about included activities as well as further information (e.g. throughput or set-up times) and an exact process description [6, 19].
For this, modeling languages become more and more important. They are also seen as an relevant enabler to fulfill the requirements of generating, customizing and configuration techniques [23]. The event-driven process chain (EPC) is such a modeling language. It is based on approaches of stochastic network procedures and Petri networks and the central modeling language of the architecture of integrated information systems (ARIS) [6].
Originally the ARIS-approach provided a framework for the modeling of computer-aided information systems. It offers a generic methodological framework which allows a holistic view on process design, management, workflow and application processing. In figure 3 the ARIS-house is shown, it contains five different views on the modeling language with their parts and extensions. The **organization view** represents the resources required to execute a function. The **data view** contains the information objects that are required or arise during the transformation process. The **functional view** shows the processes that transform input into output performance, as well as the goals related to the single functions. The **performance view** includes the structural design of the tangible and intangible input and output performance required or created in the transformation process. The **control view** is the central view which combine the elements of the four other views and their relationships [23].
Central part of the house is the EPC, a process model which the

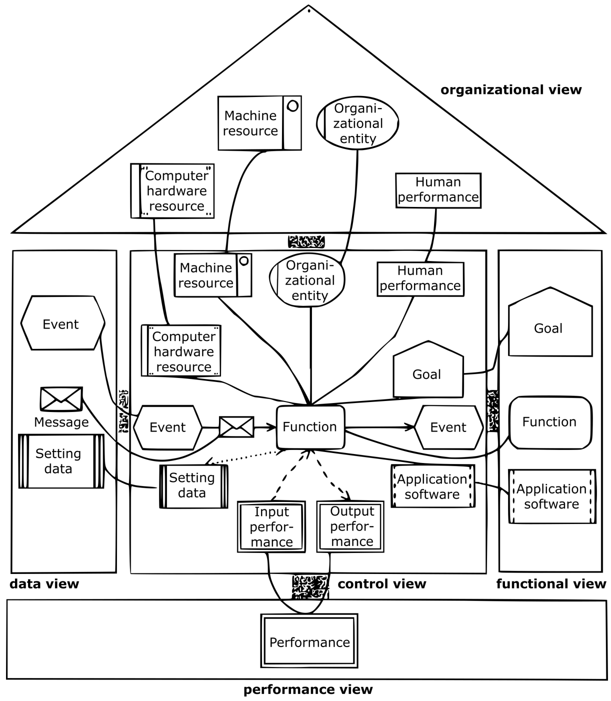

Figure 3. The five ARIS views (ARIS-house) based on Scheer and the elements of the EPC [23]

process-related relationship of functions presents. The functions are represented by the function block they are triggered by an event and result in another event (they are represented by event blocks). The functions and events are linked by control flows and the connectors AND, OR and XOR. Beside these fundamental parts the EPC can be extended by further informations which are already shown in the ARIS-house (see fig.3) [13].
Beside the EPC, there are other modeling languages, but they are not as accepted as the EPC or limited in mapping information about used infrastructure or resources. The EPC is promising for the use in combination with CAD models thus the characteristics of the service parts can be developed by means of the EPC [25].

2.4 PSS Modeling

Holistic development systems for PSS currently do not exist, predominantly the domains are processed side by side. Integration takes place through allocation mechanisms, e.g. simple combination matrices or simple rule-based configuration systems. The configuration of a PSS is an important part of the development in order to meet individual customer needs with a reasonable amount of work [8]. Existing approaches of PSS configuration discussed in literature are shown in the following. An approach based on the idea of modularization which uses combination matrices and focus on the possible product and service architectures for PSS is presented by Aurich et al. [2]. The configurability (of service components) of PSS, based on configuration rules (if-then rules) or decision tables, is part of the approach of Laurischkat [16]. She specifies that a generation (equivalent to a configuration) of PPS can be made out of five basic PSS types. Bochnig et al. [3] introduced a CAE tool, in which variants are generated by combining existing PSS modules, this is part of an integrated PSS development approach. An approach to develop an industrial PSS with predefined blocks (which are predominantly product components) is presented by Mannweiler [17].

The approaches, documented in the PSS literature are only using two of the three reasoning techniques for CAD-based solution space modeling. They uses ether rule-based or case-based techniques. For a model-based configuration of PSS a constraint-based PSS model is necessary [25]. An approach which is a helpful starting point to develop a constraint-based model, is the approach of Steinbach [26]. He adapts the definitions of characteristics and properties of Weber's Characteristics Properties Modeling/ Properties Driven Development (CPM/ PDD) approach to PSS and extends the model with internal relations of product and service parts [29]. With this approach a schematic documentation of the PSS is possible which can be transferred in to a CAD model. How such a model can be build up will be shown in the following.

3 CAD MODEL ENRICHMEND

The requirements for the development of PSS have already been mentioned, the coequal development of product and service components, the integration and addressing of individual customers and their needs and the monitoring and fulfillment of these during the entire life cycle. An important step for the coequal development of PSS is the integration of existing development tools. In the present work, parametric and knowledge-based CAD was linked with the EPK and thus a tool for the development of PSS was set up. In this, services can be developed and planned depending on the physical product, and the effects of services on the physical product can be documented. To implement this, simple references, formulas, matrix operations and hierarchical decision structures are used. Essentially, no additional tools are required for this and the implementation can be implemented using Autodesk Inventor 2017 (as CAD environment) with an Excel integration. The Excel-Inventor combination is sufficient and is used to keep the effort for the creation within limits. Additional the enriched CAD model is also a tool which helps to reach the third requirement, because it can be used for the monitoring and fulfillment of customer needs during the PSS life with a data model with a representation and documentation of the product and service interfaces, the documentation of changes on parts and their impact on other parts. Furthermore, the model helps to ensure a smooth exchange of information between the individual departments in the development of PSS components (product components or services). The structure of this CAD model is described in the next section.

3.1 Model Structure

The data model allows the configuration of a PSS likewise the support of service planing and the assistance of service technicians. To realize this a CAD-model is build up and enriched with additional data in the CAD environment, as well as an interface to a table calculation program. In this program data is stored and calculations are executed.

The structure of the model and parts of the PSS is divided into four main areas (shown in figure 4) in which information can be stored, entered and retrieved. The different actors in the PSS have access to the model in different places. The areas which can be identified in the data model are the product configurator, the CAD model, the product database and the service register.

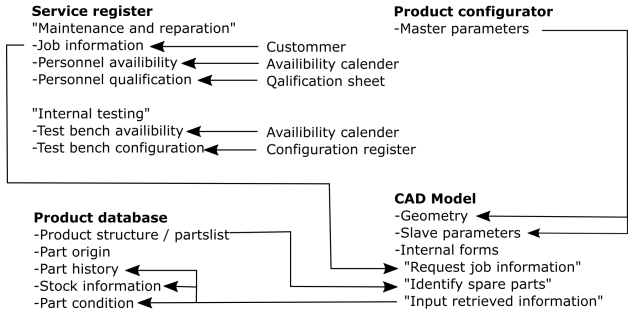

Figure 4. Structure of the data model

3.2 Configurator

The area, which is the starting part for realizing customer oriented solutions is the product configurator. It is working like known configurator for physical products and helps to adjust the system in a first step to the customer needs. To create configurations, a master CAD model in Inventor is created which contains the master parameter. In order to control the model with these parameters and to create reasonable configurations in Autodesk Inventor, the Inventor modeling language iLogic is used. In figure 5 the iLogic code for activating and deactivating parts of the model is displayed. The master parameters

```
'----------------Hier wird der MotorCart aktiviert (Case 1) oder deaktiviert (Case 0)
Select Case P:Cart
Case 0
'Teile aktivieren/deaktivieren
Component.IsActive("Motoraufnahmecart:1") = False
Component.IsActive("Bodenblech:1") = False
Component.IsActive("Quadrat_Profil:1") = False
Component.IsActive("Quadrat_Profil_MIR1:1") = False
Component.IsActive("rampe:1") = False
Component.IsActive("rampe_MIR:1") = False
Constraint.IsActive("Fluchtend:37")=False
Component.IsActive("L-Profil:1") = False
Component.IsActive("L-Profil:2") = False
Constraint.IsActive("Fluchtend:38")=True

'Schrauben und Mutter austauschen
For a=18 To 39
Component.iComponentIsActive("DIN 6921 M20 x 90:" & a & "")=False
Next
For b=1 To 6
Component.iComponentIsActive("DIN 6923 M16:" & b & "")=False
Component.iComponentIsActive("Nutenstein:" & b & "")=False
Next
For c=1 To 6
Component.iComponentIsActive("DIN 6921 M20 x 140:" & c & "")=False
Next
For d=12 To 17
Component.iComponentIsActive("DIN 6923 M20 x 1,5:" & d & "")=False
Next
For p=1 To 16
Component.iComponentIsActive("DIN 6921 M20 x 180:" & p & "")=True
Next
For f=13 To 28
```

Figure 5. Part of the iLogic code for the configurator

are embedded in a in Excel. By using the provided interface between Inventor and Excel a feasible realization of the configuration control can be realized without additional coding or external software. By varying the master parameters in Excel, a 3D model of the entered configuration is instantaneously created by updating the master CAD model and saving it as a new configuration.

Since the topic of the paper is located in the area of PSS and not restricted to pure products, the configurator has been extended to implement the services as well. The parameters in the CAD model are not tied explicitly physical properties. With the help of non-physical parameters and the tools of the programming languages VBA (Excel) and iLogic (Inventor), services are also be incorporated in the configurator and the CAD. Like for the physical parameters, non-physical parameters include constraints to ensure compatibility of the system. For example, commands like if component A is chosen, following Services are available are used. Such constrains can be based on physical (components) as well as economic reasons. A maintenance of a cheaper produced product might for example be possible, but from a economical view point not reasonable because during the whole lifetime the maintenance cost will be higher than the costs for a product with lower need of maintenance. The implementation of services in a system depends on different factors which needs to be considered.

3.3 CAD Model

The CAD model consists of (the already mentioned) master parameters and slave parameters which adapt depending on the user input automatically thus influencing the existence and geometry of components. The parameters need to be entered in a specific manner including the parameter name, unit, value and other optional fields. The number of parameters for each part are not limited and not restricted to geometric parameters, also parameters like neighborhood relations, number of parts, installation sequences or tightening torque can be derived from CAD models (these can also be transferred to the EPC to elaborate services (more in [25])). Once the part parameters are activated, they can also create and modify other databases such as a parts list in Excel. To create and modify data due an interaction of product and service a communication is needed.
To enable a communication between users (here service technicians) and Autodesk Inventor the software provides forms. They can be used to extend a user interface that allows the user to view, enter information or perform actions. The forms can be created using an editor integrated into the CAD system and usually associated with iLogic rules. The forms which are included in the system presented in this paper are ”request job information”, ”identify spare parts” and ”input retrieved information”, all of them realize the communication with a service technician and provides informations from the CAD model for the service or transfers informations from the service to the product data.
The first form, request job information, allows the technician to recall informations which are relevant for their next job by entering their identification number (ID) in an input field of the form. The data is stored in the service register and the iLogic code presents them all of the relevant information from the cells of the newest job to which they have been assigned and which suits to the technician (capabilities/ requirement comparison). The direct connection to the product data will be executed by the next form which provides the second support step for the technician. The form ”identify spare parts” allows the user to see the part dependencies of any component of the product (e.g. complexity of installation, neighborhood relations). Additionally due to the hierarchical structure of the part data base a simple combination of iLogic and VBA allows the identification of spare parts linked to the part of whichs ID has been entered in the form.
The first two forms were used to supply data to technicians, while the third form is used to return data collected during the service. In the form ”Input retrieved information” the entered information gets translated to parameters which automatically update specific cells in an Excel sheet. This allows the maintenance personnel to update the product data base with information depending on what has been done. If a part has been replaced, the stock, the status, the installation date and more will update.

3.4 Product Database

The product database was already mentioned in the sections of the configurator and the CAD model due to the internal relations of the data model. The product database contains the product-related data for individual products, so it implements the digital twin of the existing PSS. In contrast to the CAD model, the data stored here is not order-neutral.
The database is created parallel to the CAD model, the parts and informations generated configuration are stored in another Excel sheet. All of the components get automatically listed in a structured hierarchy thus splitting assemblies into sub-assemblies and single parts. In such way the dependencies between the components are easy to identify. The database gets filled out with relevant information such as amount installed, stock left, provider information, order date, maintenance interval and required maintenance certifications and more.

3.5 Service Register

Beside the product database exists the fourth area the service register because for the PSS, concrete services need to be implemented. For this purpose the service register, another data base in form of an Excel sheet, is created. The service job register includes information like the job description, job location, needed certification and the due date listed in a structured manner. In addition to that there is a cell to assign personnel for that specific job. To automate the assigning process a calendar has been created in another Excel sheet. It includes the information about the availability and the certification of the personnel. Based on the input in the service register a VBA code can easily identify suitable and available personnel with a push of a button.

3.6 Integration of the Views on the Model in a PSS

With the help of integrated iLogic commands information from excel sheets can be extracted and presented to the user in form of a message box directly in the CAD software. Such tool can for example improve the communication efficiency between departments that explicitly use a specific software and are dependent on it. In this case, the maintenance department could recall the relevant information for their job directly from the order stored in an Excel sheet (task date, problem description, task location). Built in iLogic the function of a text box allows display any cells in an excel sheet which can be identified automatically if the sheet has a defined a basic structure. Additionally the manual identification of cells is possible. For example the entering of a part ID in a form and the extraction of steps of the disassembly process. In this proposed system, the service personnel can retrieve relevant information entered by other departments. Concrete examples are provided in the next chapter as well.
The system proposed in this paper includes two roles representing some the most typical branches of service: sales and maintenance. The advantages of the proposed model enrichment techniques can be applied to any branch, these two have been chosen as example. The sales department is responsible for the service register in which they fill out the cells based on the customer input. Here an access to the standardized excel forms is necessary. The maintenance personnel is

in charge of the manual tasks in a company, the Inventor forms are intended to them as a simple but effective communication with the databases. In this way the power of the maintenance personnel is automatically restricted to only allow modification of the data which are needed or relevant to their job.

4 APPLICATION EXAMPLE

To illustrate the system described above an example of an industrial company producing test benches is used. They offers solutions of hardware in the loop (HIL) test benches for load test of engines. The following chapters shows concrete examples of the suggested CAD model enrichment techniques, their realization and advantages.

4.1 Configurator

The excel configurator containing four in the CAD model embedded parameters. The parameters describe the existence of following engine test bench parts or their size:

- An engine replacement cart
- A conditioning equipment
- An electric or a hydraulic brake
- Three different engine sizes

The input form of the excel configurator for these parts are shown in figure 6, as well as two models of the test bench (without displaying the engines).

Name	Gleichung	Einheit/Typ
P:Cart	1	oE
P:Kond	1	oE
P:Bremse	Elektrische Belastungsmaschine	oE
P:Motor	Größe A	oE

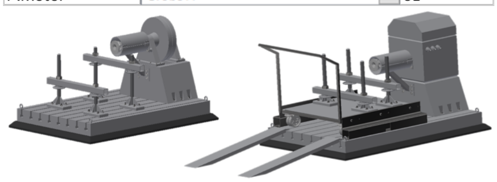

Figure 6. Configurator for test bench parts

With the mentioned parameters 16 different product configurations can be created initially. Since the CAD model is built parametrically, the compatibility is automatically ensured. For example if a larger engine size is chosen, heavier loads and dimensions are needed. For this, the support structures for the engine will adapt their position and the slot table will reduce or expand its size (dependent parameters are programmed in iLogic). Also the standard parts like screws will be replaced if the allowed loads are exceeded. The parameter and their dependencies of the configuration are shown schematically in figure 7.

Newer versions of Inventor even have a function for automatic standard part replacement using its material library directly thus reducing the programming effort.
Parallel with a CAD model the database of the configuration is generated in an Excel sheet. This sheet contains all of the components which get automatically listed hierarchically splitting assemblies into single parts (For example the engine transportation cart gets split into the profiles for the frame, wheels, screws and bolts). In this way the dependencies between the components are easy to identify.

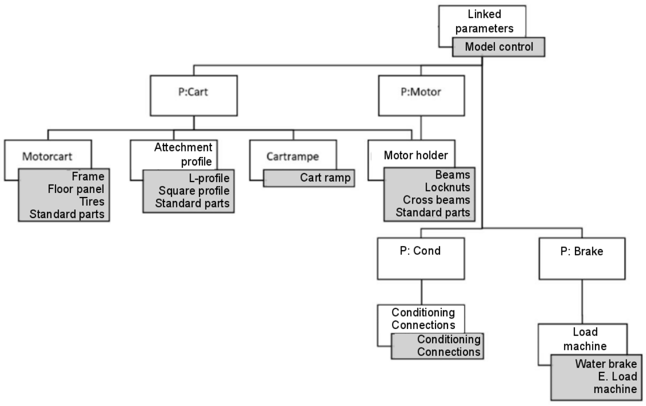

Figure 7. Parameter of the configurator

The database table gets filled with the information such as amount used, stock, provider information, order date, maintenance interval and more.

4.2 Maintenance and Repair

Until now the configurator does not differ much from the other already widely used configurators. Following the non-material factors have been implemented to the CAD model thus expanding the functionality of the configurator described above.
To upgrade the product to a PSS, the maintenance and repair of the products are integrated into the system. So if a maintenance need is known, qualified personnel must be employed to perform that maintenance.
To achieve this, the CAD data is extended by an Excel file in which a calendar has been created on an Excel sheet showing the availability of the personnel and the qualification of the personnel. When a maintenance need arises and a customer contacts the service department, a number of items are identified: the description of the problem, identification of a product, preferred maintenance dates and other information beyond. A VBA code can be activated in the Excel file at the push of a button, identifying all available and qualified employees from the calendar for the defined date and duration. If no employee is found, the system will also notify them and suggest a different date or duration. With this system, the customer can be given a confirmation for a specific date during the call, which increases communication effectiveness.

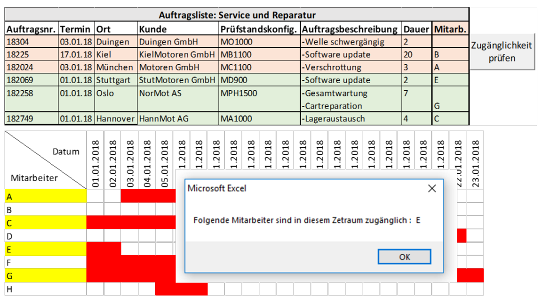

Figure 8. Job management in the PSS model

Figure 8 shows an exemplary extract from the Excel spreadsheet, both from sheet 1 with the existing orders and from sheet 2 with the employees and the employee-specific information and additionally a communication window. When the maintenance order is placed, the submitted data is stored in a separate Excel spreadsheet for the maintenance work. This information can now be retrieved by service personnel using a function integrated into the CAD model in Inventor. This is done using an iLogic code that displays a message box containing the information entered by Sales in the Maintenance Excel table.
In addition to planning service, an important part of the data model shown here is supporting the maintenance technician in the execution of his work. For this purpose, forms have been created in Inventor that realize the communication interface between the technician and the data model. The technician use the first form to request an order and Inventor provide the relevant informations including the order number, the date, information about the customer, the existing configuration of the hardware and the order description. Figure 9 shows the dialog windows of this form.

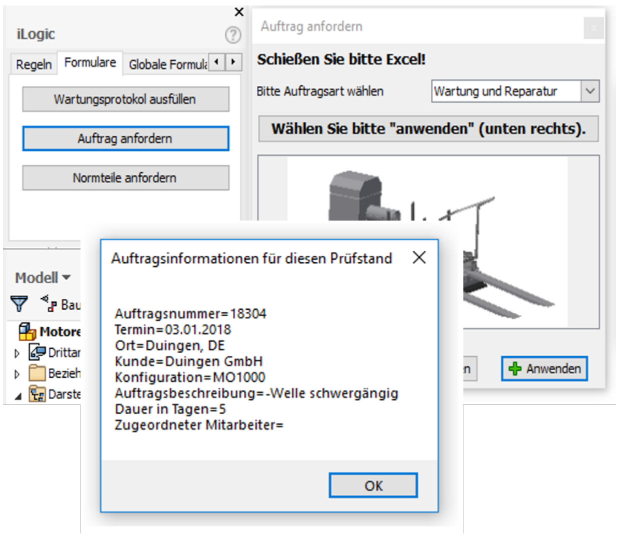

Figure 9. Job information providing for service technicians

The second form provides the technician additional informations about the maintenance task. With a relatively complex product like an engine test bench it can be hard to identify parts that also might be defect or need a replacement before a failure analysis has been performed. To assist in such task another configurator has been created in the Inventor environment. Firstly a button with an iLogic code has been created in Inventor which retrieves all the relevant job information that the sales engineer entered during the call with the customer. If the customer was able identify the broken or to be maintained component, it is then also included in the retrieved data. By entering that part or assembly number in another Inventor form, second iLogic code gets activated identifying that very same part/assembly in the data bank created at the beginning. Due to the hierarchical structure of the data bank the subcomponents or sub assemblies of that part can be identified and communicated to the user in a form of a message box. Dynamic machines often use parts that need to be replaced after every disassembly like special anti-friction bolts that cannot be reused due to the glue layer on the thread. Taking the exact amount of the right type of bolts on to the maintenance could save an additional trip or delay trying to get the needed bolts. The effectiveness of this tool does obviously depend on many factors like the experience level of the maintenance personnel, product art and its complexity, the detail of the problem description, etc.
The third form used by maintenance technicians implements a data feedback into the system and thus realizes the essential function of information feedback from the service into the model of physical components.
If the engine test bench has been repaired, serviced or parts replaced, the service personnel fill out a form in the Inventor. Input information is the identification of parts, the activity performed, the date of the activity, the reason for the activity, the new condition of the product and the proposed future activities, including any other parts that have not been repaired or maintained but may require attention. An iLogic code sends this information to the parts database in Excel and updates the relevant cells. For example, if a part has been replaced, the inventory will be reduced, the implementation date will also be updated, it will now indicate that the part has been replaced once and its condition will be set to "good". If a part has been replaced too often, or if it has been replaced before the end of its life, it may be a reason for a more detailed investigation of why this is happening. In this way, important information is exchanged immediately and automatically across different hierarchical levels (from maintenance personnel to project managers).
Figure 10 shows the data feedback parameters that are transferred to the documentation of the product (its digital twin).

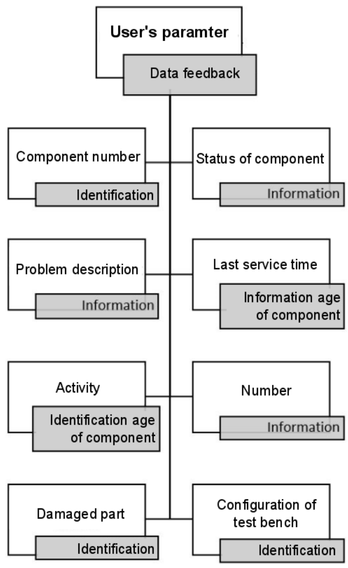

Figure 10. User parameter of the PSS

5 DISCUSSION AND CONCLUSION

In the context of this article, an approach was presented that shows the implementation of a parametric PSS data model based on a CAD application. This was applied to the example of an engine test bench and the advantages of the model were worked out.
Although CAD documents usually represent order-neutral data, this approach makes it possible to extend the CAD and create digital

twins based on the stored data of individual PSS models, which support the documentation, adaptation and execution of product and service components of the PSS during the life cycle phase.

The basic structure of a configurator makes it possible to initially respond to individual customer needs. By linking products and services in a model, it is possible to document changes to components and services and their effects. In this way the information exchange can take place without friction losses (by the translation into different models). Service planning and development also benefits from the model because it can use the information available in CAD about neighborhood relationships, number of components (e.g. screws) and additional information such as tightening torques and transfer it to tools such as the EPC. The common data model not only provides a common communication basis, but also guarantees consistency in the model and enables view management of the configuration in the PSS.

The problem with the model is that it is still a relatively rigid model that is limited to a specific application case. In further research, this model will be transformed into an approach that allows constraint-based creation of models. There, the individual parts of the PSS are to be built within the framework of a constraint network, so that an optimization of the system to different boundary conditions (e.g. Maintenance interval, costs, installation duration, remaining service life) is also possible.

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] J.C. Aurich and C. Wagenknecht C. Fuchs, 'Life cycle oriented design of technical product-service systems', *Journal of Cleaner Production*, **17**, 1480–1494.

[2] J.C. Aurich, N. Wolf, M. Siener, and E. Schweitzer, 'Configuration of productservice systems', *Journal of Manufacturing Technology Management*, **20**, 591–605.

[3] H. Bochnig, E. Uhlmann, and A. Ziefle, 'Assistenzsystem IPSS-CAD als informationstechnische unterstützung der integrierten sach- und dienstleistungsentwicklung in der IPSS-entwurfsphase', in *Industrielle Produkt-Service Systeme*, eds., Horst Meier and Eckart Uhlmann, 95–115, Springer Berlin Heidelberg.

[4] M. Boehm and O. Thomas, 'Looking beyond the rim of ones teacup: a multidisciplinary literature review of product-service systems, in information systems, business management, and engineering and design', *Journal of Cleaner Production*, **51**, 246–260.

[5] J.J. Cox, 'Product templates - a parametric approach to mass customization', *CAD Tools and Algorithms for Product Design*, 3–15.

[6] A. Gadatsch, *Grundkurs Geschäftsprozess-Management*, Vieweg+Teubner.

[7] P.C. Gembarski and R. Lachmayer, 'Designing customer co-creation: Business models and co-design activities', *International Journal of Industrial Engineering and Management (IJIEM)*, **13**(8.3), 121–130.

[8] P.C. Gembarski and R. Lachmayer, 'Product-service-systems - what and why developers can learn from mass customization', *Enterprise Modelling and Information Systems Architectures*, **13**(16), 1–16.

[9] M. Grässle, O. Thomas, M. Fellmann, and J. Krumeich, 'Vorgehensmodelle des product-service systems engineering: überblick, klassifikation und vergleich', *Integration von Produkt und Dienstleistung - Hybride Wertschöpfung*, **51**, 246–260.

[10] D. Guillon, A. Sylla, E. Vareilles, Mi. Aldanondo, E. Villeneuve, C. Merlo, T. Coudert, and L. Geneste, 'Configuration and response to calls for tenders: an open bid configuration model', (09 2017).

[11] Mario Hirz, Wilhelm Dietrich, Anton Gfrerrer, and Johann Lang, *Integrated computer-aided design in automotive development*, Springer.

[12] L. Hvam, N.H. Mortensen, and J. Riis, *Product customization*, Springer Science + Business Media, 2008.

[13] G. Keller, M. Nüttgens, and A.-W. Scheer, 'Semantische prozessmodellierung auf der grundlage ereignisgesteuerter prozeketten (epk)', *Veröffentlichungen Des Instituts Fr Wirtschaftsinformatik*.

[14] Y. Koren, *The global manufacturing revolution: product-process-business integration and reconfigurable systems*, Wiley series in systems engineering and managements.

[15] K. Kuntzky, *Systematische Entwicklung von Produkt-Service-Systemen*, Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der TU Braunschweig, Vulkan-Verl.

[16] K. Laurischkat, *Product-Service Systems: IT-gestützte Generierung und Modellierung von PSS-Dienstleistungsanteilen*, number 2012,3 in Schriftenreihe des Lehrstuhls für Produktionssysteme, Ruhr-Universität Bochum, Shaker.

[17] C. Mannweiler, *Konfiguration investiver Produkt-Service Systeme*, number 2014,1 in Produktionstechnische Berichte aus dem FBK, Lehrstuhl für Fertigungstechnik und Betriebsorganisation, Techn. Univ.

[18] O.K Mont, 'Clarifying the concept of productservice system', *Journal of Cleaner Production*, **10**(3), 237–245.

[19] F.J. Nüttgens, M. und Rump, 'Syntax und semantik ereignisgesteuerter prozessketten (epk)', *Prozessorientierte Methoden Und Werkzeuge Für Die Entwicklung von Informationssystemen - Promise*.

[20] D. Sabin and R. Weigel, 'Product configuration frameworks - a survey', *IEEE intelligent systems*, 42–49.

[21] T. Sakao, Y. Shimomura, E. Sundin, and M. Comstock, 'Modeling design objects in CAD system for service/product engineering', **41**(3), 197–213.

[22] F. Salvador, P.M. De Holan, and F. Piller, 'Cracking the code of mass customization', *MIT Sloan management review*, **50**, 71–78.

[23] A.-W. Scheer, *ARIS Vom Geschäftsprozess zum Anwendungssystem*, Springer.

[24] D. Schreiber, P.C. Gembarski, and R. Lachmayer, 'Datamodels for pss development and configuration: Existing approaches and future research', *World Conference on Mass Customization, Personalization and Co-Creation (MCPC 2017)*, **9**.

[25] D. Schreiber, P.C. Gembarski, and R. Lachmayer, 'Developing a constraint-based solution space for product-service systems', *International Conference on Mass Customization and Personalization - Community of Europe (MCP-CE 2018)*, **8**.

[26] D. Schreiber, P.C. Gembarski, and R. Lachmayer, 'Modeling and configuration for product-service systems: State of the art and future research', *International Configuration Workshop (CWS 2017)*, **19**.

[27] J.J. Shah, 'Designing with parametric cad: Classification and comparison of construction techniques', *Geometric Modelling Proceedings of the Sixth International Workshop on Geometric Modelling*, **6**, 53–68.

[28] D. Spath and L. Demuß, 'Entwicklung hybrider produkte - gestaltung materieller und immaterieller leistungsbündel', in *Service Engineering*, eds., Hans-Jrg Bullinger and August-Wilhelm Scheer, 463–502, Springer-Verlag.

[29] M. Steinbach, 'Systematische gestaltung von product-service-systems: integrierte entwicklung von product-service-systems auf basis der lehre von merkmalen und eigenschaften'.

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

[31] A. Sylla, D. Guillon, E. Vareilles, M. Aldanondo, T. Coudert, and L. Geneste, 'Configuration knowledge modeling: How to extend configuration from assemble/make to order towards engineer to order for the bidding process', *Computers in Industry*, **99**, 29–41, (08 2018).

[32] O. Thomas, P. Walter, and P. Loos, *Konstruktion und Anwendung einer Entwicklungsmethodik fr Product-Service Systems*, Hybride Wertschpfung, Springer, Berlin, Heidelberg, 2010.

[33] Arnold Tukker, 'Eight types of productservice system: eight ways to sustainability? experiences from SusProNet', *Business Strategy and the Environment*, **13**(4), 246–260.

[34] VDI, *VDI Guideline 2209 - 3D Product Modelling*, Beuth.

[35] L. Wagner, D. Baureis, and J. Warschat, *Developing Product-Service Systems with InnoFuncs*, volume 1, 2013.