NordDesign 2014
August 27 – 29, 2014
Espoo, Finland / Melbourne, Australia

Implementation and automation of the characteristic-property-concept within a PDM-system

Johannes Kößler[1], Jochen Reitmeier[2], Kristin Paetzold[3]

[1]Institut für Technische Produktentwicklung, Universität der Bundeswehr München
Johannes.koessler@unibw.de
[2]Institut für Technische Produktentwicklung, Universität der Bundeswehr München
jochen.reitmeier@unibw.de
[3]Institut für Technische Produktentwicklung, Universität der Bundeswehr München
kristin.paetzold@unibw.de

Abstract
In this paper a basic concept of simulation planning with the use of the characteristic-property-model is presented. The used characteristic-property-model has been extended with context factors to describe the current development situation. Based on the research results, a relational database structure has been developed. This structure contains characteristics and properties as well as some context factors. In addition an assignment to a single part has been created. The objective of this structure is its possibility to be available in PDM-Systems. Finally this structure has been implemented in a PDM-System to present its possible application. This application consists of automated data integration of parameterized CAD-files into the PDM-System and a graphic display of the current status of a part in terms of the current simulation status.

Keywords: pdm-system, property validation, data processing

1 Introduction

Manifold requirements for technical systems lead to both increasingly complex products and diversity of development processes. The necessary cooperation within multidisciplinary teams, based on the trend towards mechatronic products, intensifies such a process complexity. Therefore, it is important to provide methodical support for engineers to enable them to purposefully control the process in specific development situations.

1.1 Problem Description

Continuous validation of product functionality must be ensured throughout the development process [6]. Due to the increasing competition and the shrinking development times, the validation by time-consuming and costly physical prototypes has been increasingly questioned. Today, there exists a huge variety of methods and tools for virtual validations. Nevertheless, it is still a challenging decision which simulations should be performed at a certain time of development. The effective support of development processes helps to reduce the development risk for purposeful and early validation of product functionality [6]. In addition, simulation results must be indicated transparently. Most important, the used input data has to be clearly pointed out to avoid misunderstandings. This supports a targeted iteration management and a more accurate identification of the actual degree of product maturity.

1.2 Objective

The aim of current research is the development of a holistic and effective approach to a process-attendant validation of product functionality by means of simulations. This should help to significantly reduce development processes and to raise the product quality, too. In this context, several aspects are considered:

- Classical project management (e.g. demand/availability of resources)
- Input data for simulations (e.g. availability/quality of data)
- Information value of simulations

Based on the characteristic-property-modelling approach of [13], development data should be processed and linked to additional information allowing an efficient planning and triggering of virtual validations.

Usually development data are stored in product data management systems (PDM-systems). It has been proven to be advantageously when modern PDM-systems are coupled to workflow functionality. Hence, they also assist in the control of development processes within a specific project.

In this paper the following research questions are discussed:

- How can the characteristic-property-modelling approach be implemented in a PDM-System?
- How can the data and information flows be automated to support the development process?

2 State of the art and related work

2.1 Characteristics and Properties

Starting point of any development are the requirements that can be according to [13] understood as desired properties to be fulfilled by the technical system. The product development process itself can be represented by the alternating steps of synthesis and analysis. Characteristics are the result of the synthesis process and serve as input data to the analysis process, in its result, the actual (realized) properties are identified (Figure 1). It is crucial that the characteristics are determined in a way which ensures that the required properties are achieved.

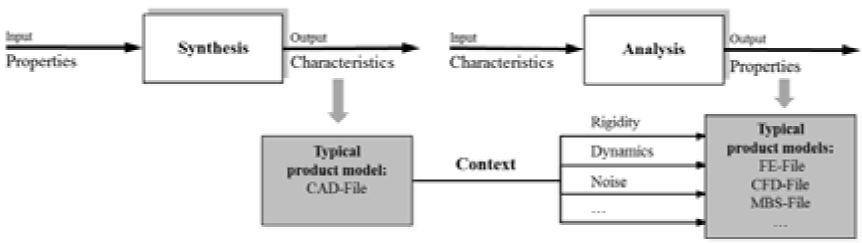


Figure 1. Data flow between steps of synthesis and analysis

Beside the CPM-approach there are many other concepts explaining the development process. Quoting all of them would go beyond the size of this paper. Therefore only two of them are mentioned. The first concept has been developed by Hubka and Eder [14]. It uses also different types of data called internal and external parameters. The classification of data in this concept is comparable to the CPM-approach. Another interesting concept has been introduced by Suh [15, 16] named Axiomatic Design. It also uses different types of data called design parameters and functional requirements. The classification of data in this concept allows a determination between structure and behavior. We are using the CPM-approach, because it additionally enables the structuring of the flow of data.

According to [13], developers have to deal with two categories of data: characteristics and properties. Characteristics describe the product in terms of structure, shape, material and consistence. For example, if the geometry and material of a component are defined as characteristics, appropriate component properties such as weight or strength are a logical consequence of the decisions made before. Consequently, properties of the product may be affected by the developer only indirectly by a modification of characteristics.

Hence, product data emerge from both kinds of process steps, but are saved by different product models resp. data storages. In this connection the following difficulties can be recognized:

- A return of data from analysis to synthesis is difficult because of different used data types
- An additional step of synthesis is necessary which cannot be automated in many cases

PDM- systems serve as a key element for the data storage as well as support for data and information flows. The structure of a technical system is nowadays represented by these tools, which is often based on a component-oriented view. In the PDM-system mainly characteristics as a result of synthesis processes are stored in a structured way. Properties (e.g. results of simulations) are included as well, but just as a kind of semantic information (stored document-based). Therefore, they are not directly linked with the characteristics, so that an assignment is possible to a limited extent only.

2.2 Specific data processing for a process attendant property validation

The previous section clarified that an appropriate linkage of characteristics with properties is primarily required. For this purpose, a matrix-based representation seems reasonable (Figure 2). By using matrices, not only the link-up between characteristics and properties (i.e. a Domain Mapping Matrix, DMM) but also the connection between characteristics and properties itself (i.e. a Design Structure Matrix, DSM) can be displayed [3] [10]. In addition, these matrices can be supplemented with additional information (e.g. context factors) [10].

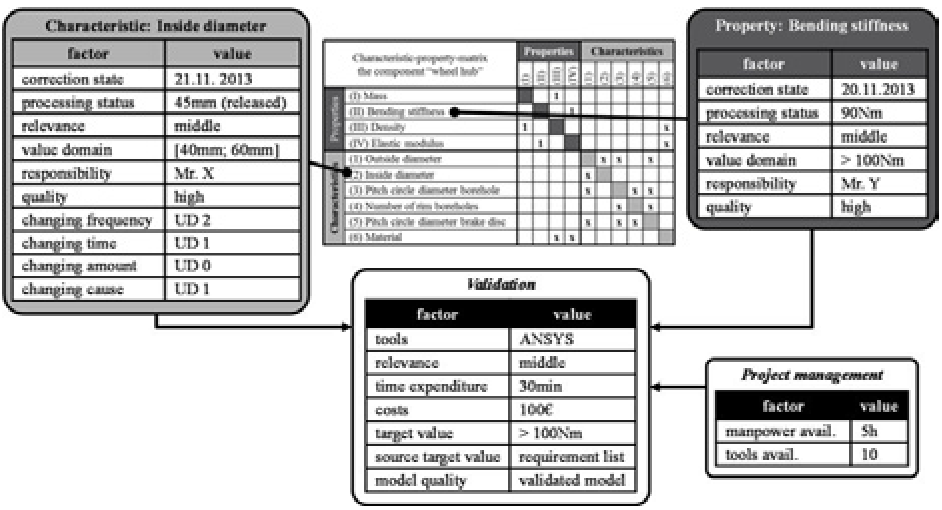


Figure 2. Link-up of context factors to the matrix-based product description [4]

As already mentioned, continuous validation of product functionality must be ensured throughout the development process [6]. However, the usefulness of a simulation is closely connected to the available data basis and thus to the process step. This proves that a situation-specific approach is required to evaluate the available data basis and to support the decision-making process concerning the execution of an intended simulation. According to [12], there is no universal description of the development situation. However, there is a tendency to describe situations on the basis of defined factors. These factors are adapted to the respective situation context [7]. Clearly different factors have different relevance depending on the specific perspective. Consequently, context factors were determined in [10] and expanded in [4] to describe the present situation with a specific view to the execution of simulations. These context factors address all important aspects to support decision-making situations with regard to the execution of virtual property validations.
In total, 27 context factors were defined. These factors include product-related (to identify the current development status), process-related (to support the simulation planning and the evaluation of simulation results) and resource-related (identification of available resources) aspects to describe the development situation in the simulation context. Exemplarily three factors are briefly described, which are partly taken up again in chapter 3:

- changing frequency [4]:
  allows drawing conclusions how often product changes and hence the degree of product maturity possibly occurs; this will be incorporated into the planning of simulations.
- value domain characteristic/ property [10]:
  shows the extent of freedom concerning a possible solution; this can be numerical/quantitatively, but also reflects a lot of approved individual solutions.
- release status [10]:
  provides information on the processing status of data; this is closely related to a specific data quality as well as the (limited) availability of data.

2.3 Micro cycle of a process attendant property validation

A theoretical construct for simulation planning, in terms of a micro cycle, was demonstrated in [11] (Figure 3).

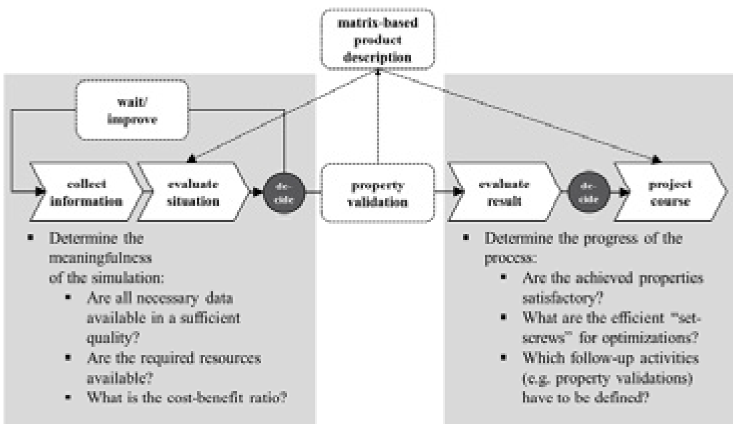


Figure 3. Basic concept of simulation planning following [11]

Two essential aspects are considered within simulation planning. The evaluation of the meaningfulness and feasibility of a simulation is put in front of a step of analysis. The subsequent evaluation of the validation results is supported by statements concerning the simulation quality. Here, the matrix-based data processing is used as a core element. These will initially be used to identify the data required for a simulation, which are then evaluated concerning their quality. Based on the weighted relationships between input data and property to be validated, statements are derived concerning the achievable quality of simulation results [8]. This micro cycle is linked [10] to the FORFLOW-process [2], which got constant positive feedback from industry. It is used as a reference process because it proved to be detailed and variable. It displays also a universally oriented approach to address current challenges of product development by means of process and workflow support. Beside the possibility of a situation-specific process planning the FORFLOW-process model explicitly refers to steps of analysis respectively to property validations. It is followed by a review of the analysis and decision-making situations concerning the progress of the process.

3 Utilization in a PDM-System

As mentioned above, development data are usually stored in PDM systems. It has been proven to be favourably when modern PDM systems provide workflow functionality. They can easily support the providing of data as well as the development process, as workflow elements like milestones, responsibilities or release mechanisms are included. In principle, these workflows help coordinating not only the work of development teams but also the interaction between them.

The basic elements of a simulation planning were shown in the previous chapter. To be able to use them in the context of workflow functionality, an appropriate data structure has to be developed firstly (chapter 3.1) and then mapped into a PDM system (chapter 3.2).

3.1 Development of a data structure

The first step to implement the data model is the decision which database system will be used. In general there are different systems available. These systems are relational databases, object oriented databases, semantic databases or graph databases. The typical systems, which are used in most PDM-Systems, are relational databases. Therefore we use this system for the development of the data structure. The structure is independent from a PDM-System and can be implemented in most of the existing PDM-Systems.

In relational database systems, the data are stored in relations. These relations can be understood as tables containing single rows called tuples [1]. In order to map the data processing for workflow support shown in chapter 2.2, it must be initially converted into a relational schema. This transfer is shown in Figure 4.

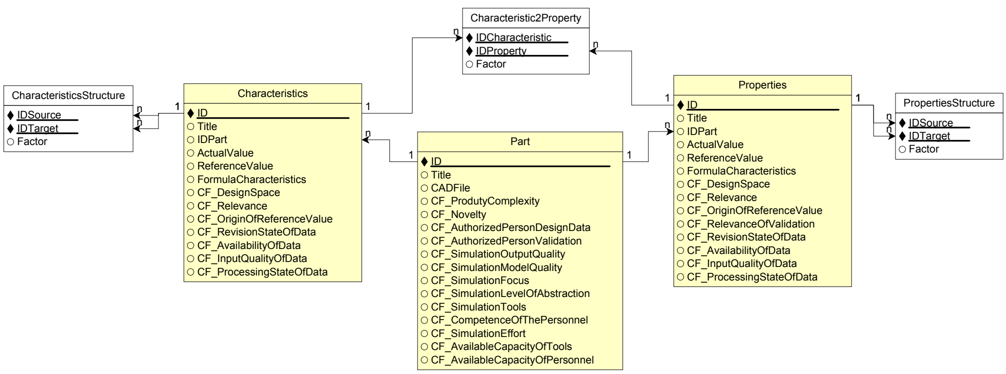


Figure 4. Relational data structure for simulation planning

The first relation required for this data structure is the relation "property". Here, all information is stored, which is necessary for a description and validation of properties in the context of simulations. The specific information that describes a relation is called attributes. Exemplarily, these include the actual value of a property, the predetermined range of values or the target value. In addition, a mean shall be provided so that a property (a tuple) can be uniquely identified. In a relational database, this is ensured by using a primary key named the property ID (automatically assigned by the system).
In contrast to the clearly assignable information (= attributes) to a property, characteristics cannot be clearly assigned to a single property. This is firstly due to the fact that a characteristic may influence various properties and secondly to the fact that a characteristic itself is defined by a series of additional information. The "characteristics" are therefore mapped as a second relation. In this relation, all information is also defined analogous to the properties needed for their description and simulation tasks. Exemplarily, the actual value of a characteristic or the predetermined range of values can be mentioned.
The third relation is used to display individual "parts". Beside an unambiguous identification, this relation contains context factors that apply to the entire component. (For example, the draftsman in terms of a responsible CAD designer). To support a planning of simulation, this relation also references to related simulation models and CAD models. Another function of this relation is the grouping of characteristics and properties to a single component. This grouping is essential because characteristics and properties of a component are initially not uniquely identifiable. For example, all components have the characteristics “length, width and height” and therefore the properties “volume and mass”. To identify the appropriate characteristics among a huge list of characteristics, they must be uniquely assigned to a component, which is realized by means of the relation "components".
In addition to the three previously described relations, a possibility to display the dependencies between characteristics and properties is required according to the model described in chapter 2.2. These dependencies are mapped in three other relations. The first relation is the relation between characteristics and properties (in terms of a DMM). In this relation, only the relationship between a property and its characteristics is included. The influence of a characteristic on a property is stored as part of the relation “property”. The last two relations are used to define the dependencies between the characteristics and the properties among each other (in terms of DSMs).

3.2 Prototypical implementation of the relational data structure in a PDM system

In the following chapter, a prototypical implementation of the data structure will be shown. For this implementation the PDM-System CIM Database was used.

3.2.1 Basic introduction of the used PDM/PLM-platform

The PDM/PLM-platform CIM DATABASE from Contact® is a comprehensive solution for collaborative product development [5]. It is designed as a turnkey system and permits individually tailored solutions due to the wide range of applications covered and the system's versatile adaptability. This PDM/PLM-system supports both product data management (PDM) and product lifecycle management (PLM) activities. It is firmly grounded in the integrated interaction of the areas “virtual product”, “documents”, “processes” and “projects”. The open, service-oriented and very customizable architecture based on the programming language Python is the main advantage of this PDM-platform. In addition this PDM-platform is based on a relational database system. It has also been chosen for the implementation because of the customizable architecture and the relational database system. But any other PDM-platform with a relational database system could also have been chosen.

3.2.2 Actual implementation in PDM-system

In the following, the concrete implementation will be presented. First of all, some customizing is done to make the new data structure available in the PDM-system. The next step shows an automatic import of properties and characteristics that were defined in a CAD-file. The last step particularly deals with a visualization, to gain the viewer's attention by highlighting important aspects.

1) Customizing of the PDM-platform

The PDM-system already provides a lot of functions that facilitate the implementation of workflow support. These include technical and professional functions. In the technical field, the database system is emphasized. This is a relational system which allows flexible extension and so the implementation of the model is possible. In the professional field, the change history has to be mentioned as a classic function of a PDM-system. This allows, for example, identifying the changing frequency of specific development data and will be integrated in a simulation planning in terms of data consistency. The “changing frequency” is a context factor and allows drawing conclusions on how often product changes occur [4].
In addition the PDM-system provides simple options to expand existing structures or to define new structures within the PDM-platform (Figure 5).

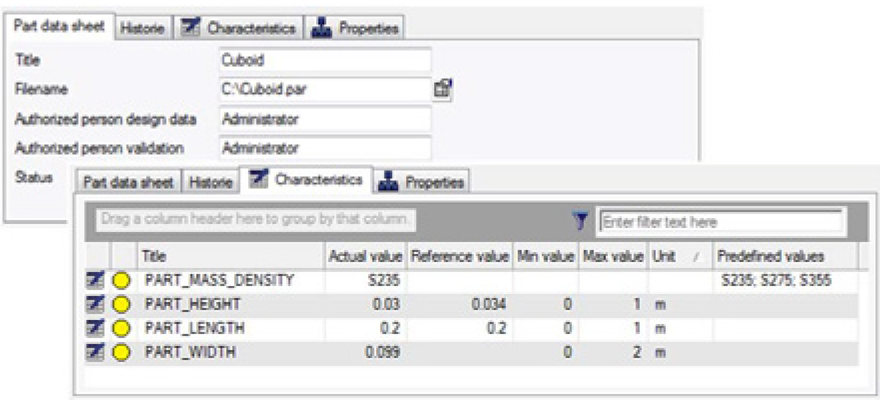


Figure 5. Customizable functionalities

New fields or structures can be defined by a configuration change depending on specific requirements. As an example for the definition of new fields, the contextual factors "authorized person design data", "authorized person validation" and “value range of characteristic” (e.g. usable options concerning material or maximal length due to installation space restrictions) are considered. By defining these fields and a description of the associated options, this information can be mapped to a component in the existing product management.
The implementation of dependencies between characteristics and properties can be considered as a definition of a new structure. Therefore, the relations “characteristic”, “property” and their “connection” are newly created. By the masks already provided by the PDM-system, the extensions can be used without any programming. Nevertheless, the applicability of these automatically generated masks is only limited practicable for the present context. Up to this point of implementation, the user can store and find a lot of necessary information. Concerning the required structure this link-up has to be administrated not only in the PDM-system but also in source systems such as CAD-systems. A further development of such a customizable implementation is therefore absolutely necessary and possible because of the open structure of the PDM-system. The objective is the extended integration of data and link-ups within the PDM-system that are already maintained in a CAD-file (e.g. the characteristics

concerning the dimensions and the properties volume or mass). For this purpose, however, a specific programming in terms of a modification of the system functionality is required exemplarily described in the following.

2) Functionalities to be programmed
Based on the programming language Python, the PDM-system provides an open platform so that the system behaviour can be extended and automated. The first objective of the developed extension is the partial automation of the maintenance of characteristics and properties as well as their dependencies (Figure 6).

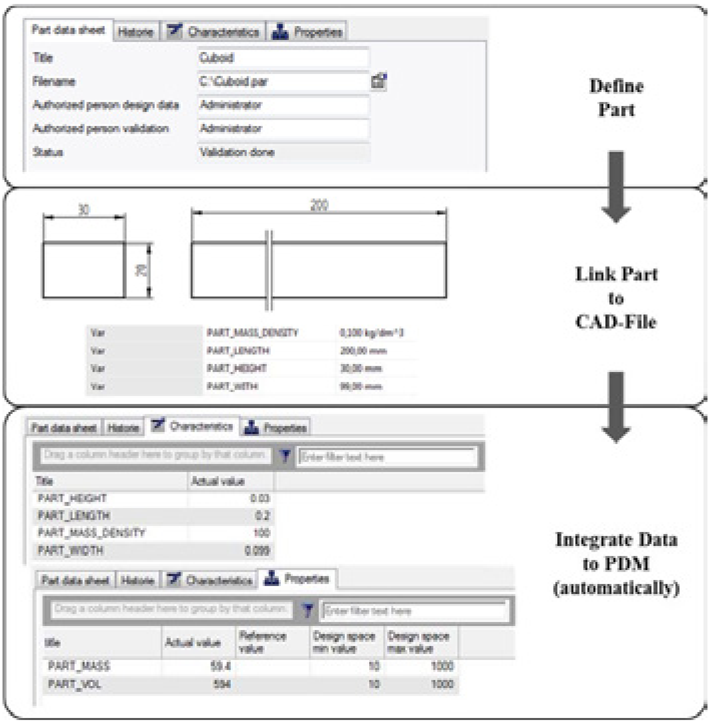


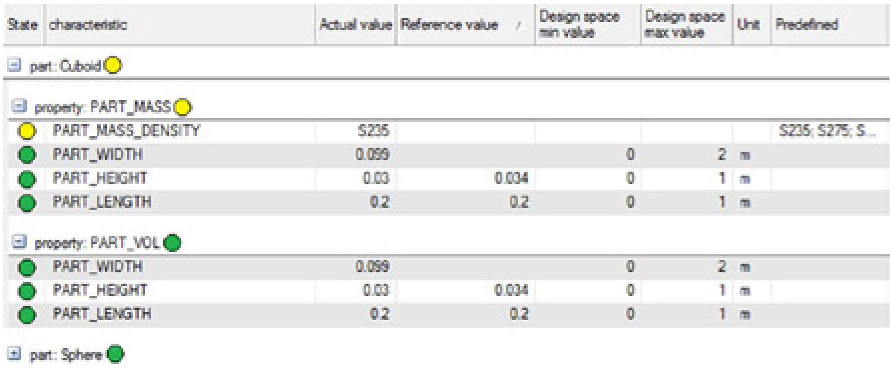
Figure 6. Automated acquisition of information from CAD-files

The automation is intended to include characteristics and properties of a parameterized CAD file (in the present case from Solid Edge) in the PDM-system. In addition, the relationship between characteristics and properties is recognized on the basis of formulas in the table of variable of the CAD-system and transferred to the PDM system. The aim of this implementation is to avoid redundant work and the acquisition of previously defined characteristics and properties of a CAD-system. In a first step, the implementation uses the expanded part table in the PDM-system linking to an existing CAD-file. When storing an entry in this table it is checked if a CAD-file was attached. If a file is attached, then the file is opened by the system and the table of variables is read out automatically. The information name, value and formula are applied to each variable. Based on the value of the formula and its unit, the system decides whether it is a characteristic or a property and sets the respective entries in the PDM-system. It should be noted that only the present properties in a CAD-file are automatically linked in the current early stage of implementation. A link-up with other properties (e.g. stiffness, in conjunction with the released requirements management with respect to nominal properties) is currently being developed.

3) Visualization

As demonstrated, an initial implementation of the characteristic-property-matrix in the PDM-system is performed. By extending the system functions, these extensions can be semi-automatically filled with information on the basis of CAD-files. In addition, general requirements like target values or ranges of values are stored. Experiences from industrial project show that reasonable visualizations have a great importance. This may concern the release status of data as well as open tasks. Hence, a traffic light scheme is often served, which will be used in the present research project as well.

To allow a support for simulation planning, a possibility is required to identify the status of a component, for example. The status of a part shall be dependent on the status of the associated characteristics and properties. This is shown in Figure 7.



State	characteristic	Actual value	Reference value	Design space min value	Design space max value	Unit	Predefined
	part: Cuboid						
	property: PART_MASS						
	PART_MASS_DENSITY	S235					S235; S275; S...
	PART_WIDTH	0.099		0	2	m	
	PART_HEIGHT	0.03	0.034	0	1	m	
	PART_LENGTH	0.2	0.2	0	1	m	
	property: PART_VOL						
	PART_WIDTH	0.099		0	2	m	
	PART_HEIGHT	0.03	0.034	0	1	m	
	PART_LENGTH	0.2	0.2	0	1	m	
	part: Sphere						

Figure 7. Visualization of a part status

In this figure, the component status of a cuboid is set to "in progress" (yellow). This is due to the fact that no final material selection (characteristic) has been met respectively no material was determined. Thus the status "released" cannot be assigned. As a consequence no component mass can be determined. Additionally, no statements can be made regarding the component stiffness. The latter could only be made based on assumptions. This consequently influences the quality of the simulation results (integrated in the context of a simulation planning) which should be used for the triggering as well as for the evaluation of simulation results [8].

4 Conclusion and Outlook

In this paper, the essential elements for an efficient approach to a process attendant property validation were shown. In addition, the initial work has been described, how to prototypically implement the basic characteristic-property concept in a PDM-system. It was also mentioned how to make the theoretically developed concepts for simulation planning available within development workflows. The implementation has been achieved by the transfer of the data model into a relational database structure. Furthermore a CAD-System has been connected to the PDM-System. With this connection an automated transfer of data from the CAD-Part to the PDM-System has been demonstrated. This transfer shows how the data and information flows can be automated to support the development process.

Currently, the relational data structuring and the practical implementation into the PDM-system is being continued and optimized based on new findings. In the following, workflows will be added, which should support a simulation planning. In addition, automated functions will be implemented to integrate simulation results into the PDM-System.

The integration of complete products, component assemblies and the relationships between parts into the data model is also part of the current research project.

References

[1] Date, C.J. “An Introduction to Database Systems”, Addison-Wesley, 8th edition, 2003

[2] Krehmer, H., Eckstein, R., Lauer, W., Roelofsen, J., Stöber, C., Troll, A., Weber, N., Zapf, J. “Coping with multidisciplinary product development - a process model approach”, In: Proceedings of the 16th International Conference of Engineering Design (ICED 09), August 24-27, 2009, Stanford University, Stanford, CA, USA.

[3] Luft, T., Krehmer, H., Wartzack, S., “An advanced procedure model for property-based product development”, Proceedings of the 19th International Conference on Engineering Design – ICED 2013, Lindemann, U., Venkataraman, S., Kim, Y.S., Ion, W., Malqvist, Y., (Eds.), Seoul, 2013.

[4] Luft, T., Reitmeier, J., Paetzold, K., Wartzack, S. „Uncertainty considerations by context factors for the monitoring of the degree of product maturity”, submittet for the 13th International Design Conference - DESIGN 2014, May 19th-22nd, 2014, Dubrovnik - Cavtat – Croatia.

[5] N.N. “CIM DATABASE Release 10 What’s new booklet”, user manual, Contact Software GmbH, 2013.

[6] Paetzold, K.; Reitmeier, J. “Approaches for process attendant property validation of products”, 1st International Conference on Modelling and Management of Engineering Processes MMEP 2010, Cambridge 2010.

[7] Ponn, J., “Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte“, Dr. Hut München 2007.

[8] Reitmeier, J., Paetzold, K. “Evaluation of data quality in the context of continuous product validation throughout the development process”, Proceedings of the 18th International Conference on Engineering Design 2011 (ICED´11). 15.-18. August 2011, Kopenhagen, Denmark, Vol. 9, 2011, pp. 143-152.

[9] Reitmeier, J.; Paetzold, K. “Reduction of Uncertainty by Sensitivity Analysis and context-specific Data Processing within Virtual Property Validation”, In: Kyvsgård Hansen, P.; Rasmussen, J.; Jørgensen, K. A.; Tollestrup, C. (Hrsg.): Proceedings of the 9th Norddesign, ISBN: 978-87-91831-51-5, 22.-24. August 2012 Aalborg, Denmark.

[10] Reitmeier, J., Paetzold, K. “Process-integrated analysis of the development situation for an efficient simulation planning”, Proceedings of the 19th International Conference on Engineering Design – ICED 2013, Lindemann, U., Venkataraman, S., Kim, Y.S., Ion, W., Malqvist, Y., (Eds.), Seoul, 2013a.

[11] Reitmeier, J., Luft, T., Wartzack, S., Paetzold, K. “Consideration of uncertainties in the product-oriented development process”, Proceedings of the 3rd International Conference on Modelling and Management of Engineering Processes, 20.-21. November, Magdeburg, Germany, 2013b.

[12] Roelofsen, J. „Situationsspezifische Planung von Produktentwicklungsprozessen“, Dissertation, München: Dr. Hut Verlag, 2011.

[13] Weber, C., “CPM/PDD - An extended theoretical approach to modelling products and product development processes”, Proceedings of the German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes, Bley, H., Jansen, H., Krause, F.-L., Shpitalni, M. (Eds.), Fraunhofer Stuttgart, 2005.

[14] Hubka, V.; Eder, W.E.: Einführung in die Konstruktionswissenschaft. Springer, Berlin-Heidelberg, 1992.

[15] Suh, N.P.: The Principles of Design. Oxford University Press, 1990.

[16] Suh, N.P.: Axiomatic Design. Oxford University Press, 2001.