ENGINEERING OF ASSEMBLY SYSTEMS USING GRAPH-BASED DESIGN LANGUAGES

Breckle, Theresa (1); Kiefer, Jens (1); Rudolph, Stephan (2); Manns, Martin (3)
1: Ulm University of Applied Sciences, Germany; 2: University of Stuttgart, Germany; 3: The University of Siegen, Germany

Abstract

Car Manufacturers are subject to continuous and fundamental changes. Already today increasing time pressure, rising complexity and a soaring cost pressure require a shorter time to market. As assembly planning is one of car manufacturers' core competence an innovative approach to adapt the processes is needed. This paper presents the early idea of a novel approach for an automated design process covering the early design phase of assembly systems. Integrating various input data as well as requirements leads to a are a base for this approach for designing an assembly system. Uncertain input data, which are fact within early planning phases, need to be considered in order to reach a holistic planning alternatives for evaluation, optimization and afterwards decision-making. With so-called graph-based design languages an automated and efficient design process will be implemented. This leads to a faster designing process in order to reduce planning time and planning costs and reach resilient and sustainable decisions.

Keywords: Uncertainty, Requirements, Early design phases, Design process

Contact:
Theresa Breckle
Ulm University of Applied Sciences
Production Engineering and Production Economics
Germany
breckle@hs-ulm.de

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 1: Resource-Sensitive Design | Design Research Applications and Case Studies, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

Manufacturing companies are in a dramatic change, triggered by emerging markets, new technologies, sustainability policies and a change within consumer preferences, in particular around ownership. New fundamentally questions about diverse mobility, autonomous driving, electrification and connectivity are arising for car manufacturers (Mohr et al., 2016). Due to a global and customer-driven market, car manufacturers are already permanently confronted with the challenge to reduce the duration and cost of the product creation process. Market-driven demands are linked with an increasing complexity (e. g. increasing product, process and resource complexity) and risks due to increasing quality demands, product variety and technologies (ElMaraghy et al., 2012). Furthermore, the frequency of new product introductions is increased. To sum up, fundamental changes, increasing time pressure, rising complexity and a soaring cost pressure affect the production processes of automotive industries.

This setting requires new design methods and strategies, which have to be efficient, fast and resource-sensitive. In order to cope with these challenges, this paper presents a new model-based engineering approach using graph-based design languages focusing on the early design phase of assembly systems (concept planning). This paper describes the most important characteristics of this new engineering approach as well as its implementation and evaluation using a practical example from an automotive body shop.

2 STATE OF THE ART

2.1 Engineering of assembly systems

In the field of engineering the term of assembly is used in different ways. An assembly can be defined as "the action of fitting together the component parts of a machine or other objects". Furthermore, it can be seen as "a unit consisting of components that have been fitted together" (Lit and Delchambre, 2003). In the context of this paper assembly will be used in the sense of action. As one subdomain of a manufacturing system in an assembly system, parts or preassembled products are joined together. These are integrated into a semi-finished or into the final product. Figure 1 illustrates the components of an assembly system.

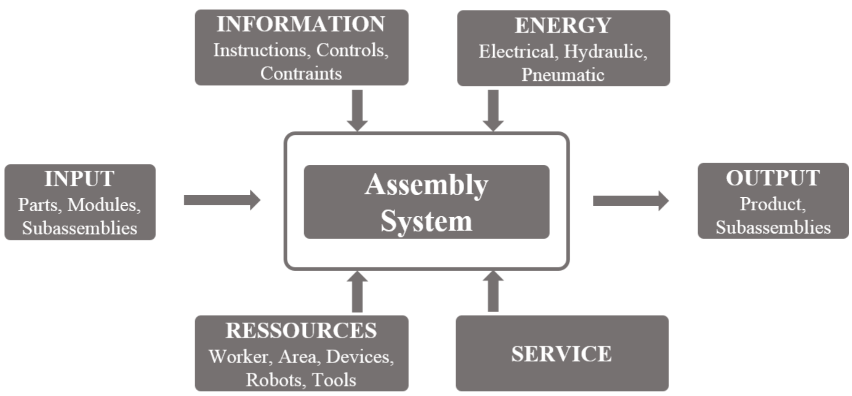

Figure 1. Basic components of an assembly system

Assembly systems are mainly classified into three types, which are manual, automated and hybrid. During the assembly, a product receives a large amount of value creation. Concurrently, technical and organizational problems and failures may occur within the assembly, which originate from earlier steps of the production process or within product design. In pertinent literature, many classical process models for production system planning can be found, e. g. by (Kettner et al., 2010; Grundig, 2015; Wiendahl et al., 2010; Schenk et al., 2014). Based on these process models, there are further approaches e. g. by (Suh *et al.*, 1998), approach by (Schuh et al., 2007) and related to the approach of assembly planning by (Konold and Reger, 2009). These process models vary in number of phases, beginning, endpoint and their level of detail. Fundamental content and the way of analytical methods is common. These models can generally be classified into preparation, structural planning, detail planning, execution planning and execution (Bergholz, 2005).

2.2 Design methodology

The design process of a precise object represents the synthesis of the final design, which considers early stages specified definitions of the designed object. Requirements and constraints, which occur within the design process, have to be fulfilled (Lawson, 2006). Constraints are defined as limiting conditions or restrictions, which reveal within the design process. Despite of manifold definitions and approaches of the design process, there is neither a generally accepted and standardised theory of a design process, nor in practice (Lawson, 2006). For an engineering design process, there is no generally accepted and standardised design methodology (Rudolph, 2002).

The presented approach uses the design methodology of Pahl and Beitz (Pahl et al., 2005). Based on this, the design process can be split up into the following steps: Planning and task clarification, conceptual design, embodiment design, detail design and overall design process. During the conceptual design phase, a principle solution is fleshed out. For this purpose, all essential problem statements are disassembled into subtasks (e. g. sub functions). The goal is to search for suitable solutions or to use existing experience and knowledge. A combination of partial solutions leads to different solution alternatives. Afterwards, the solutions alternatives are evaluated on the basis of defined criteria (e. g. technical or economic criteria) in order to find the most suitable solution (Feldhusen and Grote, 2013). Complex design tasks require iteration loops (Pahl et al., 2005; Lindemann Udo, 2011). An increasing complexity of the designs as well as a target improvement in efficiency postulate computer aided methods for automation within the design process (Feldhusen and Grote, 2013). According to (Feldmann, 1997), this design methodology can be transferred to the design of assembly systems.

2.3 Current challenges

Integrating the real and the virtual world, planning models are fundamental in order to create complex systems (BITKOM et al., 2016). Within the meaning of the digital factory, the concept planning aims for a holistic planning considering all requirements and uncertainties as well as the product concept and all relevant structures regarding processes and resources (Bracht et al., 2011). Numerous models come into existence in this process. This could lead to inconsistency. Basically, there are two scenarios for a data exchange. The first possibility is that every data model exchanges data with every other model. This leads to a quadratic (n*(n-1)/2) rise of the required interfaces. Alternatively, a central model can be used, which contains all relevant data. An exchange between the central model and the domain models requires n interfaces (bidirectional). By means of a central model, the consistency issue between different models is solved (Reichwein, 2011).

As described above, complex design tasks require iteration loops, in order to consider e. g. uncertainties or changes regarding requirements. In many cases, current approaches do not allow to process these changes in an automated way. This leads to a lack of acceptance, as it requires manual and error-prone work. Changes within the model or results of simulations can often not reimported in the original data management system. Modelling approaches, which focus on the hole lifecycle, are missing according to (Straßburger et al., 2010; BITKOM et al., 2016). In this context, the question regarding a common abstract meta model is relevant. The reuse of expert knowledge (e. g. experiences from previous projects) is favoured. This postulates an approach for the design of assembly systems, which follows a formalised, holistic design approach. In order to allow a (partly) automation of the design, the approach needs to be build up a central model. The idea of such a requested holistic engineering approach as well as the essential role of a central model is depicted in Figure 2.

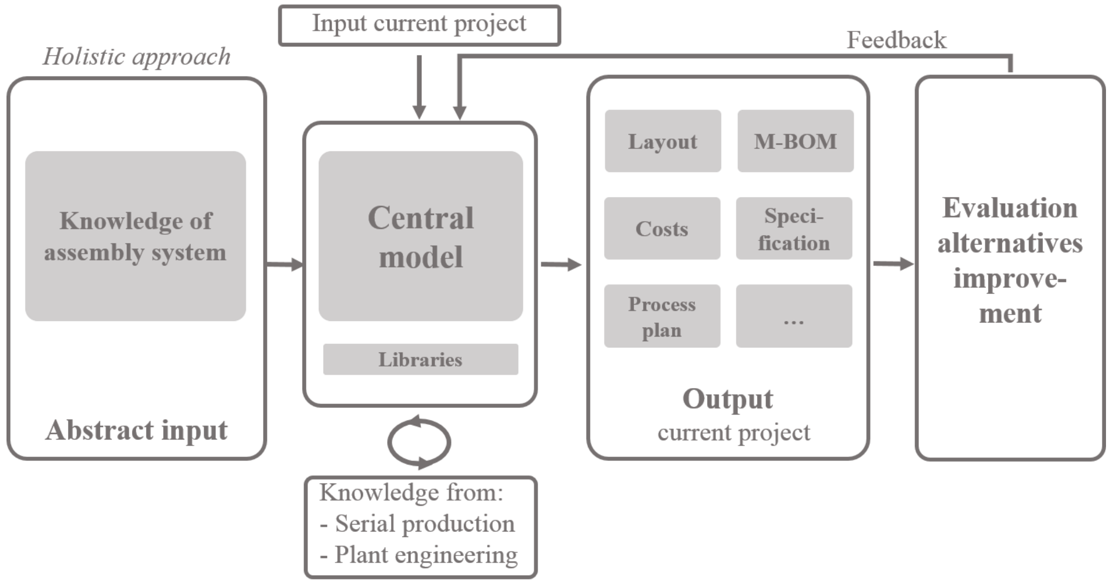

Figure 2. Idea of a holistic engineering approach for assembly systems

3 HOLISTIC ENGINEERING OF ASSEMBLY SYSTEMS USING GRAPH-BASED DESIGN LANGUAGES

3.1 Graph-based design languages

As a knowledge-based engineering method, design languages offer a rule-based and digital execution of the design process, which can be also re-executed. For the definition of computerized design processes, the concept of graph-based design languages (see e. g., (Kröplin and Rudolph, 2005; Rudolph, 2002)) has evolved into a generic framework over the last fifteen years. Examples of a successful application can be found in (Arnold and Rudolph, 2012; Groß, 2014; Vogel, 2016). This concept has been adapted from natural languages, spoken by humans, as one possibility to reproduce this complex and creative process. A human language can describe highly complex and interwoven issues with a high level of design freedom. This can be achieved by sentence formation, in which a grammar is composed of vocabulary and rules. A valid sentence is a legal combination of vocabulary and represents a valid design. In this sense, the vocabulary represents data. Information, knowledge and competence are depicted by syntactic, semantic and pragmatic aspects of design rules. Within the design process, there are iterative loops from data to information to knowledge and finally to wisdom or rather competence, as described in the Knowledge Ladder (Ackoff, 1989). A design language is developed by finding the relevant vocabulary for the current design problem. Defining explicit, formal rules (i. e. under laying design patterns) from often implicit human knowledge is mandatory for processing. Knowledge (on the design activities) and information (on the design data) have to be connected in order to create executable design rules. Exploration of the design space enables to achieve permissible solutions of the initial design problem (Rudolph, 2016).

The implementation, evaluation and processing of design languages are executed with so-called design compilers. One example of such a design compiler is the Design Compiler 43 (DC 43) developed by the IILS GmbH in cooperation with the Institute of Statics and Dynamics of Aerospace Structures of the University of Stuttgart (Alber and Rudolph, 2003). Design languages consist of a meta-model, which stores all relevant design information (Arnold and Rudolph, 2012). Based on the graphical modeling framework Unified Modeling Language (UML), the vocabulary is represented by UML classes. Ontologies are represented by the UML class diagram and the rules are represented by graphical UML rules or java rules. Instructions are included in the activity diagram. The use of UML within the design language is shown in Figure 3. Further information regarding UML can be found in (Object Management Group OMG, 2015).

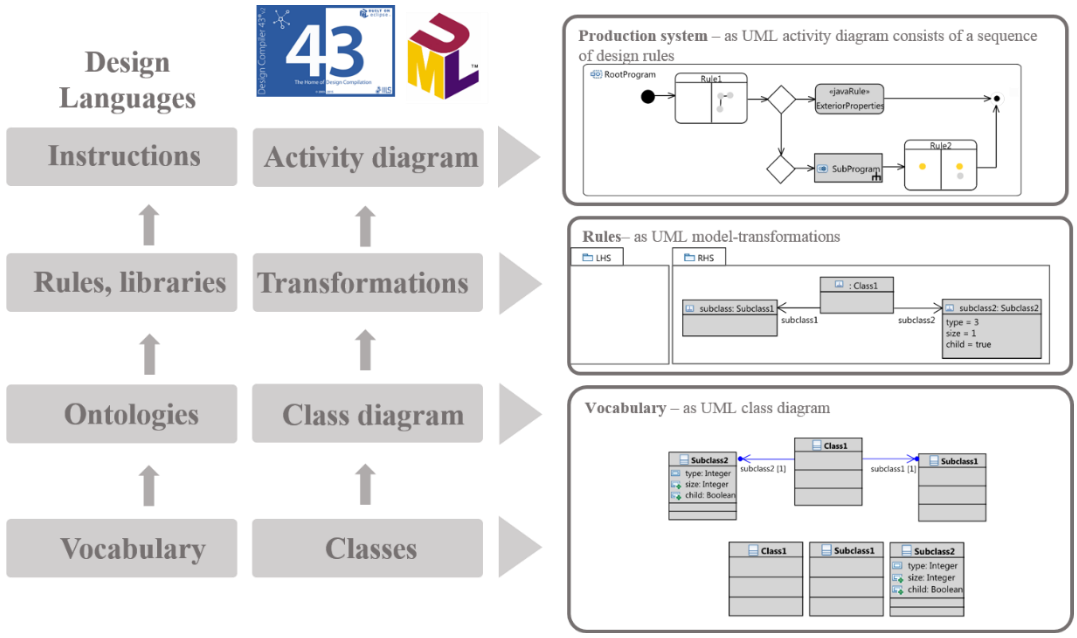

Figure 3. From vocabulary to activity diagram - use of UML

As depicted in Figure 4, the complete design process is contained in a design language. According to the principle of compilation, the design representation is generated by the design compiler. This is similar to modern computer programming languages. Constraints can be attached at classes. The design compiler executes the design rules and derives automatically by model-to-model transformations (M2M) a design graph. This holistic meta-model includes all relevant data. By model-to-text transformations (M2T), the data can be exported into a domain specific language (e. g. CAD), compare also Figure 4. Results of the analysis within the domain specific language can be fed back in the design graph. Defining and programming the design language is an engineering task in opposite to generating models, which is done by the design compiler.

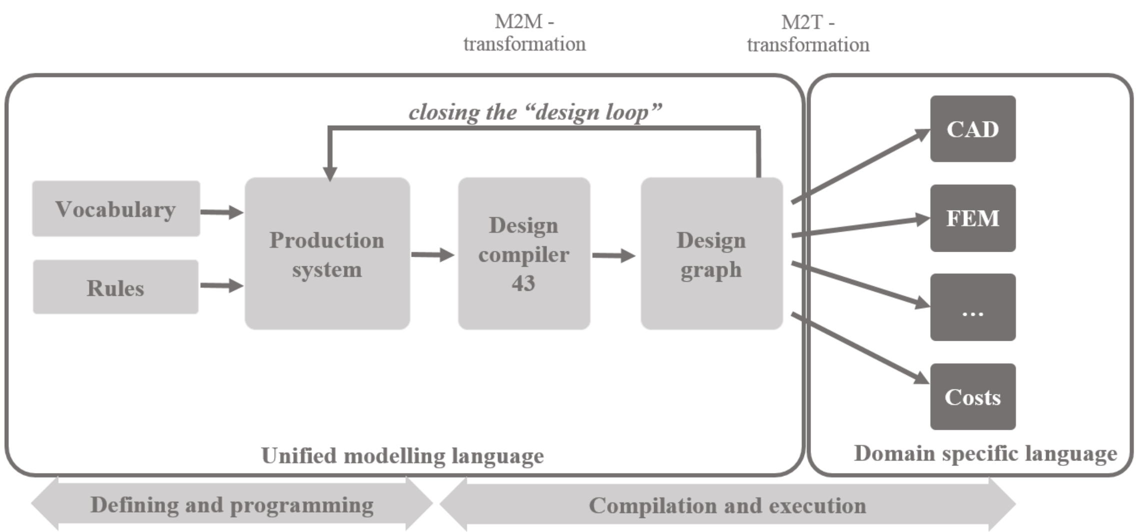

Figure 4. Design compiler architecture

With this approach, model consistency can be ensured as well as the re-execution. Results from M2T transformation can be fed back into the design graph. With this feedback the design loop can be closed. Thereby, an iterative process is enabled, which represents a complete design process (Rudolph, 2016, 2012). Within the concept of graph-based design languages, a graph represents the set of objects (as graph nodes), where some of these objects are connected (as graph edges). By changing the design parameters, which are contained in a node, the topology of the graph can be modified quickly (Arnold and Rudolph, 2012).

3.2 Graph-based engineering of assembly systems

The holistic planning, evaluation and ongoing improvement of all structures, processes, transfor-mations and resources of the real factory in conjunction with the product is the goal of the future digital factory. By applying the method of graph-based languages to design an assembly system, a new approach of the early, conceptual phase of the assembly planning is developed. According to the surrounding conditions, the use of graph-based design languages described above, prevents consistency issues due to a central data model. The ability of applying a consistent and re-executable design process improves the results quality of the early planning phase as well as design time is shortened. Due to shortening design time, multiple planning alternatives can be evaluated. With this way to express knowledge, changes within the process can be adopted. Thereby, a holistic approach can be realized. Holistic is meant here in this sense, that the definition of the digital factory includes and considers business calculations and economic efficiency as well as all related requirements as an output of the target planning. Knowledge from former engineering projects is integrated as well as current knowledge and information from serial production. As described above, knowledge processing needs to be adapted for a design of an assembly system. The design language process is depicted in Figure 5. Outcomes of this process are for example a layout, costs and a specification for the later detail design of e. g. equipment, a working schedule or a manufacturing bill of material (M-BOM).

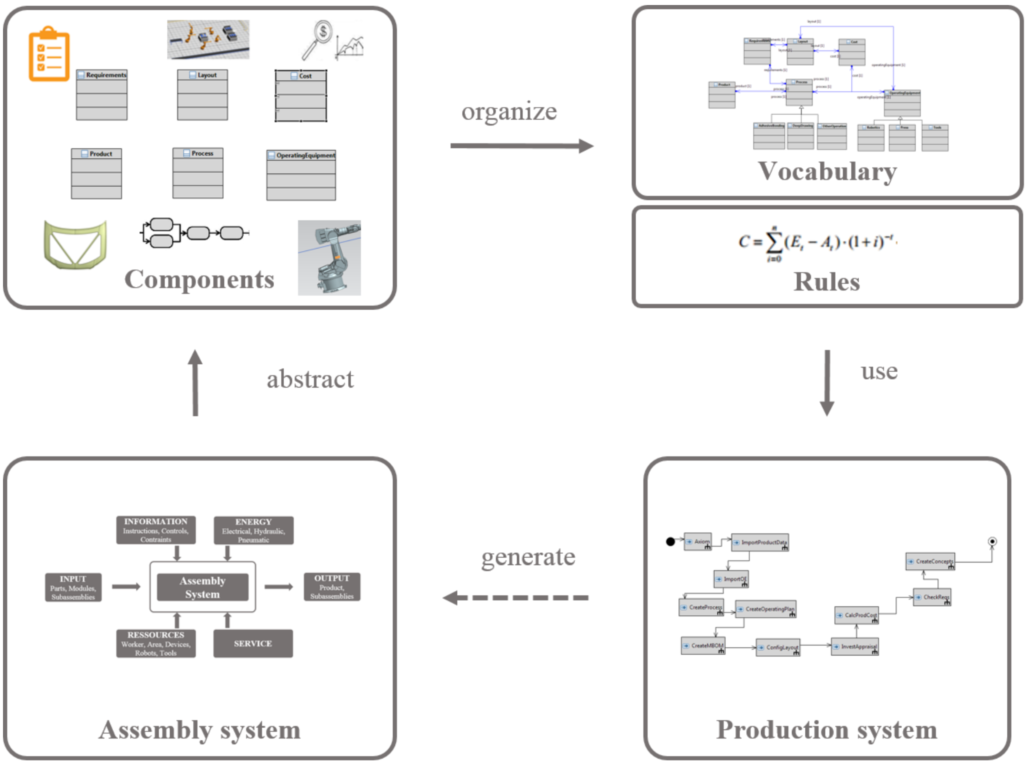

Figure 5. Design language process

To build up a graph-based design language, the assembly system is decomposed in its entities as shown in Figure 5. These entities represent the vocabulary of the assembly system on the highest level, e. g. a process concept. In order to describe the assembly system completely, each vocabulary of the highest level has to be decomposed in its components e. g. process-concept is built of technologies, assembly process plan and manufacturing bill of material. Each vocabulary has to be broken down into its basic components. For decomposition, the knowledge of the content, the structure and the dependencies of an assembly system is crucial. Within modelling, the vocabulary represents the UML-classes and is build-up in connection with their associations (relationships and dependencies) the UML-class diagram.

Constraints to the system are integrated into corresponding classes. An assembly concept is a complex system as described above. The decomposition of assembly system components is shown in Figure 5 on a high level. Requirements of an assembly system concept are placed as a component of the design language. The product concept is an essential component and linked to the product creation process. Within the process concept the assembly process plan and assembly times are created. The resource concept contains a concept for a layout, personal planning, operating resources and material flow. The basic component business calculation includes a concept of production costs, investment appraisal and lifecycle costs of the production equipment. Each component needs to be further detailed in a sub design language, in order to represent the design knowledge. Logically sequences of rules in order to build and rebuild the design graph are stored in the production system, which represents the design of the assembly system. Compilation creates the central model, which consists all relevant design information.

As described above, deep knowledge of designing a concept of an assembly system is the basis for developing the design language in order to pursue the holistic approach. Knowing the interwoven dependencies and relationships is vital. The conditions and limitations are described as constraints. Including a current design project, the modelling is realized and via feedback the current concept of an assembly system can be post-processed. Depending on requested results or targeted requirements, the output can be optimized, evaluated and alternatives can be designed before a final output is generated.

4 APPLICATION OF THE NEW ENGINEERING APPROACH USING A PRACTICAL EXAMPLE

The newly developed graph-based engineering approach has already been implemented using a real existing practical example. This example is an assembly station from an automotive body shop. Figure 6 shows the most important steps in the development process of the central data model and illustrates the generated outcomes of the configuration process of the assembly station.

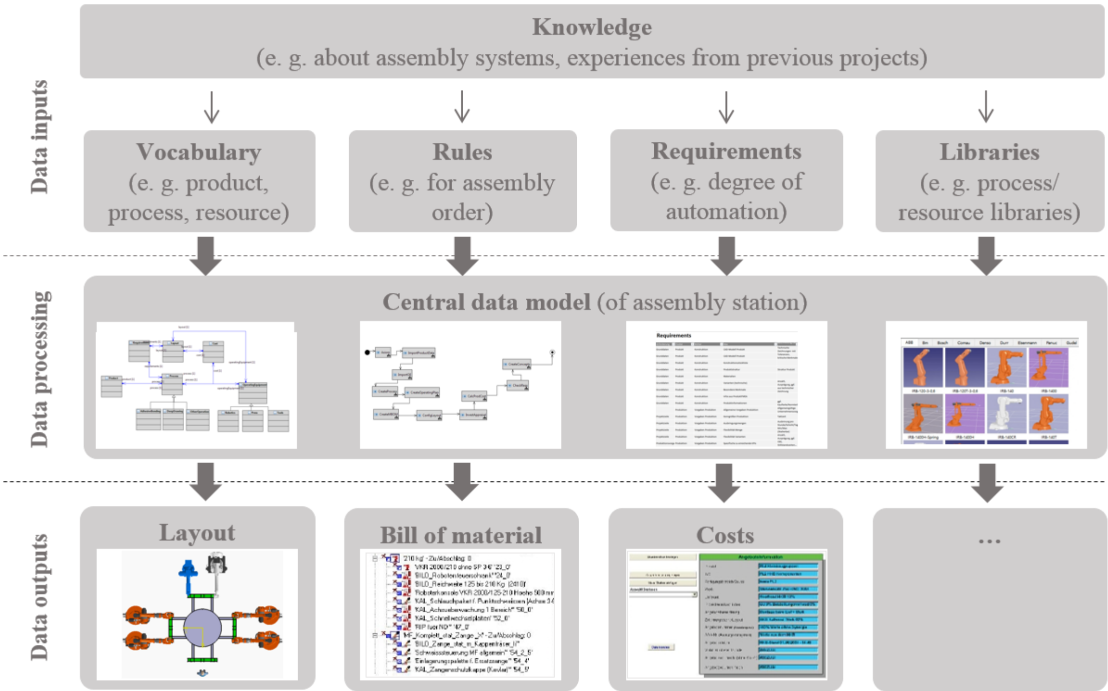

Figure 6. Application of the new engineering approach - Workflow and outcomes

In this example, the central data model of the assembly station has been created by a predominant manual process using the design compiler DC 43. The needed vocabulary of the assembly station (e. g. process and resource terms) was entered, the requirements (e. g. high degree of automation) and rules (e. g. further restrictions) were represented in the production system and preconfigured libraries (e. g. resource libraries including 3D geometry, costs, space requirements etc.) are used. To increase the efficiency of the modelling process, the authors are currently working to automate the generation process of the central model by using standardized data structures and interfaces.

As depicted in Figure 6, the 3D layout, the bill of material and the costs (e. g. acquisition costs, maintenance costs) of the assembly station can be automatically generated based on the central model. For the visualization of the 3D layout, there are basically two different possibilities: the internal viewer of the design compiler DC 43 (based on opencascade) or the use of an external CAD system. In case of using an external CAD system, the CAD data are currently transferred from DC 43 to the CAD system (e.g. CATIA V5, Siemens NX) using the data format step. At present, activities are being carried out to exchange data (e. g. CAD data, alphanumerical data) in the future using the standardized data exchange format Automation ML (www.automationml.org).

Figure 7 depicts an example of the model-to-model transformation for the layout modelling. This transformation is formulated in all rules within the production system. Firstly, the requirements for the available space is loaded (e. g. geometry of the existing floor-space). In a second step, the required operating equipment is imported from existing libraries (e. g. robots or tools). The placement of the operating equipment depends on the assembly process (especially on the assembly order). Lastly, the new layout concept is generated. In Figure 7, the design graph represents the layout with its operating equipment and its geometry (e. g. positions).

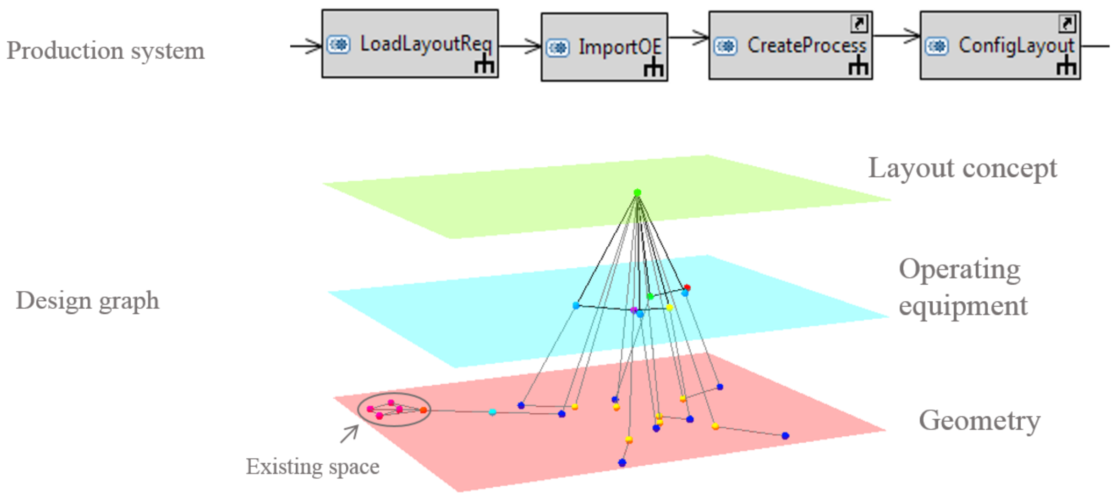

Figure 7. Model-to-model transformations for layout modelling

5 SUMMARY AND OUTLOOK

A new approach how to design an assembly system with graph-based design languages has been presented. Transferring this method of graph-based design languages to the engineering of assembly systems solves consistency issues by using one central, consistent, data model. A design language for assembly systems creates a widespread basis for modelling. This is achieved by integrating a holistic input, considering requirements and business calculations and adding knowledge from former projects as well as from current serial production. The capability of iterative loops leads to sustainable results, in order to manage rising complexity and be efficient.

Based on the practical example described in Chapter 4, the feasibility of the new model-based engineering approach could successfully demonstrated. Much progress has already been made in the context of the information-technical implementation of the new engineering process, the authors are currently working on further optimizations. These improvements are mainly concerned with the efficiency increase both regarding the creation of the central model (e. g. the import and automated processing of project-specific requirements using standardized data structures or formats) and regarding the generation of the domain-specific solutions within the respective IT systems (e. g. the data transfer via AutomationML). In addition to that, based on the central data model further project-specific data and documents shall be automatically generated in the near future, such as process plans and specification documents. In order to pursue the holistic idea, an evaluation and improvement loop, as shown in Figure 2, a deeper and more widespread solution for this loop needs to be worked out.

Since this development of a design-language has required a time-consuming pre-investment, it is inefficient to run it only once. In case of iterative design loops, it is efficient in time (project time and time to market) and costs (e. g. less budget for equipment, less project time due to less manual iterations). Besides, there are further benefits such as a traceable design process, a documented process and the ability to react to product changes and other changes within the product creation process and within the assembly planning project in an automated way. This might be especially valuable in the early stages of engineering of assembly systems.

REFERENCES

Ackoff, R.L. (1989), “From Data to Wisdom”, *Journal of Applied Systems Analysis*, No. 16, pp. 3–9.

Alber, R. and Rudolph, S. (2003), “A Generic Approach for Engineering Design Grammars”, *Proceedings of the AAAI Spring Symposium ”Computational Synthesis”.*

Arnold, P. and Rudolph, S. (2012), “Bridging the Gap between Product Design and Productmanufacturing by means of Graph-Based Design Languages”, *1st ACCM Workshop on Mechantronic, Karlsruhe, Germany, May 7-11. 2012, digital proceedings, Delft*, pp. 985–997.

Bergholz, M.A. (2005), *Objektorientierte Fabrikplanung*, Dissertation, RWTH Aachen, Germany.

BITKOM, VDMA and ZVEI (2016), “Umsetzungsstrategie Industrie 4.0. Ergebnisbericht der Plattform Industrie 4.0”.

Bracht, U., Geckler, D. and Wenzel, S. (2011), *Digitale Fabrik: Methoden und Praxisbeispiele, VDI-Buch*, Springer, Berlin, New York. 10.1007/978-3-540-88973-1

ElMaraghy, W., ElMaraghy, H., Tomiyama, T. and Monostori, L. (2012), “Complexity in engineering design and manufacturing”, *CIRP Annals - Manufacturing Technology*, Vol. 61 No. 2, pp. 793–814. 10.1016/j.cirp.2012.05.001

Feldhusen, J. and Grote, K.-H. (Eds.) (2013), *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung,* 8., vollst. überarb. Aufl., Springer Vieweg, Berlin. 10.1007/978-3-642-29569-0

Feldmann, C. (1997), *Eine Methode für die integrierte rechnergestützte Montageplanung*, Dissertation Technische Universität München, Germany. 10.1007/978-3-662-06845-8

Groß, J. (2014), *Aufbau und Einsatz von Entwurfssprachen zur Auslegung von Satelliten*, Dissertation, Universität Stuttgart, Germany. 10.18419/opus-3938

Grundig, C.-G. (2015), *Fabrikplanung: Planungssystematik - Methoden - Anwendungen,* 5., aktualisierte Aufl., Hanser, München.

Kettner, H., Schmidt, J. and Greim, H.-R. (2010), *Leitfaden der systematischen Fabrikplanung: Mit zahlreichen Checklisten,* Unveränd. Nachdr. der Ausg. 1984, Hanser, München.

Konold, P. and Reger, H. (2009), *Praxis der Montagetechnik: Produktdesign, Planung, Systemgestaltung, Vieweg Praxiswissen,* 2., überarb. und erw. Aufl., korrigierter Nachdr, Vieweg, Wiesbaden. 10.1007/978-3-663-01609-0

Kröplin, B. and Rudolph, S. (2005), “Entwurfsgrammatiken –Ein Paradigmenwechsel?”, *Der Prüfingenieur*, Vol. 26, pp. 34–43.

Lawson, B. (2006), *How designers think: The design process demystified*, 4th ed., Elsevier/Architectural, Oxford, Burlington, MA. 10.4324/9780080454979

Lindemann Udo, P.J. (2011), Konzeptentwicklung und Gestaltung technischer Produkte: Systematisch von Anforderungen zu Konzepten und Gestaltlösungen, 2. Auflage, Springer, Heidelberg, Dordrecht, London, New York. 10.1007/978-3-642-20580-4

Lit, P. de and Delchambre, A. (2003), *Integrated Design of a Product Family and Its Assembly System,* Reprint, 1st edition 2003, Springer Science + Business Media, New York. 10.1007/978-1-4615-0417-7

Mohr, D., Kaas, H.-W., Gao, P., Wee, D. and Möller, T. (2016), “Automotive Revolution & Perspective Towards 2030”, *Auto Tech Review*, Vol. 5 No. 4, pp. 20–25. 10.1365/s40112-016-1117-8

Object Management Group OMG (2015), *OMG Unified Modeling Language (OMG UML): Version 2.5,* Available at: www.omg.org/spec/UML/2.5/PDF (accessed date: April 4, 2017).

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2005), *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung, Springer-Lehrbuch,* 6. Aufl., Springer, Berlin, Heidelberg. 10.1007/b137606

Reichwein, A. (2011), *Application-specific UML Profiles for Multdisciplinary Product Data Integration*, Dissertation, Universität Stuttgart, Germany. 10.18419/opus-3869

Rudolph, S. (2002), *Übertragung von Ähnlichkeitsbegriffen.*, Habilitationsschrift, Universität Stuttgart, Germany.

Rudolph, S. (2012), "On the problem of multi-disciplinary system design -and a solution approach using graph-based design languages", *1st ACCM Workshop on Mechatronic Design, Linz, Austria, Nov 30th*.

Rudolph, S. (2016), *lecture notes "Digital Engineering"*, Institute for Statics and Dynamics of Aerospace Structures, Faculty of Aerospace Engineering, University of Stuttgart, Germany.

Schenk, M., Wirth, S. and Müller, E. (2014), *Fabrikplanung und Fabrikbetrieb: Methoden für die wandlungsfähige, vernetzte und ressourceneffiziente Fabrik,* 2., vollst. überarb. und erw. Aufl., Springer Vieweg, Berlin. 10.1007/978-3-642-05459-4

Schuh, G., Gottschalk, S., Lösch, F. and Wesch, C. (2007), "Fabrikplanung im Gegenstromverfahren", *wt online*, Vol. 97 No. 4, pp. 195–199.

Straßburger, S., Bergmann, S. and Müller-Sommer, H. (2010), "Modellgenerierung im Kontext der Digitalen Fabrik - Stand der Technik und Herausforderungen", in Zülch, G. and Stock, P. (Eds.), *Integrationsaspekte der Simulation*, Fachtagung Simulation in Produktion und Logistik, Karlsruhe, 7. und 8. Oktober, Karlsruhe, pp. 37–44. 10.5445/KSP/1000019635

Suh, N.P., Cochran, D.S. and Lima, P.C. (1998), "Manufacturing System Design", *CIRP Annals - Manufacturing Technology*, Vol. 47 No. 2, pp. 627–639. 10.1016/S0007-8506(07)63245-4

Vogel, S. (2016), *Über Ordnungsmechanismen im wissensbasierten Entwurf von SCR-Systemen*, Dissertation, Universität Stuttgart, Germany. 10.18419/opus-8829

Wiendahl, H.-P., Reichardt, J. and Nyhuis, P. (2010), *Handbuch Fabrikplanung: Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten,* 1. Aufl., Carl Hanser Fachbuchverlag, München. 10.3139/9783446423237

ACKNOWLEDGMENTS

The approaches presented in this paper are part of the research project „digital product life-cycle (ZaFH)", which is supported by a grant from the European Regional Development Fund and the Ministry of Science, Research and the Arts of Baden-Württemberg, Germany (information under: www.rwb-efre.baden-wuerttemberg.de).