INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.204

Capturing the Design Rationale in Model-Based Systems Engineering of Geo-Stations

A. Zech [1], R. Stetter [2], S. Rudolph [3] and M. Till [2,✉]

[1] EKS InTec GmbH, Germany, [2] University of Applied Sciences Ravensburg-Weingarten, Germany,
[3] University of Stuttgart, Germany

✉ till@rwu.de

Abstract

The design rationale describes the justification of design decision or selection. To avoid unnecessary design iterations, a capturing and documentation of this rationale is highly desirable. In digital engineering processes it is of imminent importance not only to document the evaluation processes behind this rationale but to make them repeatable and digitally executable. This allows to design a variety of product variants within an engineering framework. This paper explains an approach based on graph-based design languages and presents it based on a section of an automotive assembly system.

Keywords: design automation, graph-based design languages, design rationale, model-based systems engineering (MBSE), systems design

1. Introduction

A design rationale describes the reasons behind a design selection or decision as well as the justification for this decision, i.e. considered alternatives and their evaluation (compare Lee 1997). For industrial companies the ability to provide the rationale of past design solutions may be crucial (Heisig et al. 2010). A lack of re-use of the design rationale can result in increased product and quality costs, less competitive products and continuous rush in the design department (Ellman et al. 2018). In digital engineering, it is therefore not only necessary to capture the design rationale, but to store it in a manner which allows both the digital access and a far-ranging re-evaluation for parametrical as well as topological product variations. In current engineering approaches, most design decisions and selections are based on elaborate simulations and multi-objective evaluations, which are commonly digitally executed. The simulations range from finite elements analysis (FEA) over multi body system (MBS) analysis to collision detection simulation. Additionally, synthesis processes are frequently digitally executed, e.g. by means of topology optimization. Today, digital engineering is frequently implemented in a synergetic form as model-based systems engineering (MBSE), i.e. digital models are used to create, manage, review and advance product designs from system to component level across all disciplines (Gräßler et al. 2021). This papers explains an approach to capture the design rationale in MBSE which is based on the conscious application of graph-based design languages (GBDLs). These GBDLs can be digitally executed in a framework and allows to close the design loop by means of an evaluation. A central objective in such applications is thus the automated re-use of the underlying evaluation processes which lead to certain design decisions or selections. The research question underlying this approach can therefore be formulated as follows:
How can the rationale of design decisions or selections be captured in digital engineering frameworks in a manner that allows automated re-evaluation even for topological different design variants?

The scientific results are explained using the example of the digital engineering process of a geo-station. A geo-station is a station within the assembly line of an automotive car body, which is used to ensure the positions and orientations of several product components that need to be assembled. The active elements in a geo-station (usually an industrial robot in combination with centering and clamping units) realize components positions and orientations which are specified locally in the product design. The research results were generated in a large-scale research project which is characterised in the next section. Section 3 explains the underlying engineering framework of the developed approach which is based on GBDLs. The underlying engineering process is described in Section 4. The core of the application includes the design rationale for the definition of a joining sequence (Section 5), the design rationale for clamping and centering positions (Section 6) and the design rationale for clamping and centering units (Section 7). The paper is concluded in Section 8 with a summary and outlook.

2. Underlying research project

The aim of the underlying research project is the holistic digital representation and machine execution of the product life-cycle (PLC). In concrete terms, all stations, starting with the design of a product through its basic architecture and geometry, the simulation and validation, the production in the (digital) factory and the comprehensive cost and energy balancing are to be integrated into a consistent overall digital model (Till et al. 2016). In order to explore the application of GBDLs based on the Unified Modelling Language (UML) the project comprised 6 Universities of the south-west of Germany with several small and mid-size enterprises as well as large industrial companies (see https://dip.rwu.de for details). The central research topic of the project "Digital Product Life-Cycle (DiP)" was the consistent integration of the individual engineering domains in the PLC (requirements, design, product architecture and geometry, virtual testing, digital factory, holistic evaluations of costs and energies) into an overall model. Four example products were analysed in the project: an automotive car body, a gear-system for sub-urban trains, a two-wheel in-line vehicle and a quadrocopter. Figure 1 gives an overview of the project.

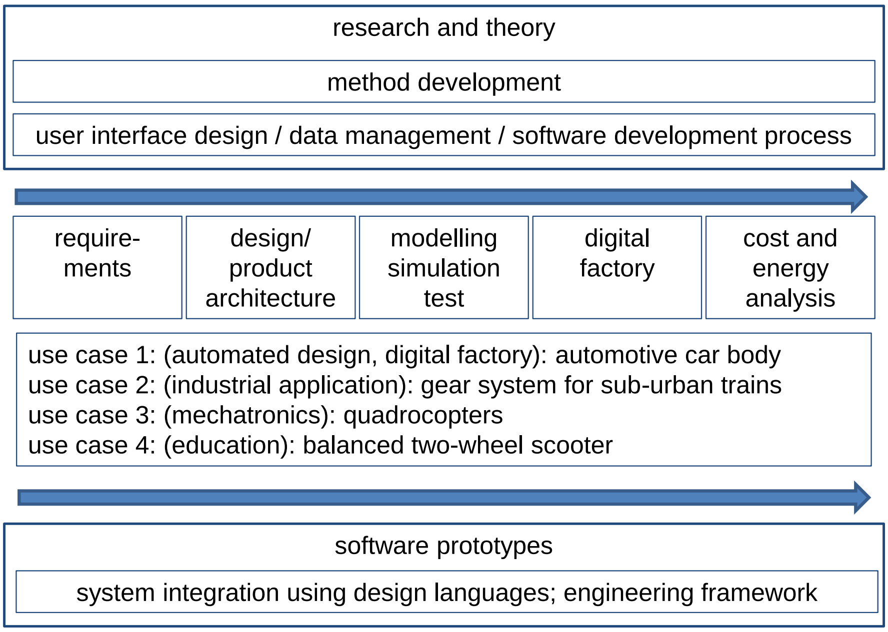


Figure 1. Overview of the underlying research project DiP

The research project focused on the mapping of the PLC and included the difficulty of having to integrate a variety of different technical domains. In this research project, the subdivision into the domains “requirements”, “design / product architecture”, “modelling / simulation / test”, “digital factory” and “cost and energy analysis” was chosen. The investigation of this main process was supported by a research and theory segment concentrating on method development as well as user interface design, data management and the software development process. The second supporting section investigated the system integration using design languages and the engineering framework.

https://doi.org/10.1017/pds.2022.204 Published online by Cambridge University Press

3. Model-based systems engineering with GBDLs

Model-based systems engineering (MBSE) approaches are a step forward in the evolution of digital engineering, because they employ formalized models (Shaked and Reich 2021) and may lead to significant benefits in terms of engineering time and cost as well as product quality, process quality and process stability. Current research in this area is focussing, amongst others, on the integration of verification processes into MBSE (Laing et al. 2020) as well as the tracking and communicating risks through MBSE tools (Darpel et al. 2020). Important insights concerning MBSE can be gained from the guideline VDI/VDE 2206 which was republished in 2020 as a draft under the new name "Development of cyber-physical mechatronic systems (CPMS)" (VDI 2020). Essentially, this guideline supports system development engineers in all tasks in the development of cyber-physical mechatronic systems by means of representing the main logical relationships. These relationships form an inherent flow logic and may be represented as an updated and extended V-model (Gräßler and Hentze 2020).
One possibility for the implementation of MBSE is an engineering framework based on graph-based design languages (GBDLs). GBDLs allow the representation of knowledge in the Unified Modelling Language (UML) and the compilation by means of a design compiler (Design Cockpit 43, IILS Company, 2021). The underlying concept and the essential aspects of graph-based design languages were described in several scientific publications in the last years (compare e.g. Rudolph 2006, Holder et al. 2017, Riestenpatt and Rudolph 2019). The connection between formal languages from software engineering (UML, SysML) was also explored by several research groups (e.g. Cao et al. 2011, Kruse and Shea 2016). The distinctive quality of an engineering framework based on GBDLs is that it enables a machine executable representation of a topological diverse product family. This allows a holistic, multi-domain representation in connection with powerful simulation, topology optimization and evaluation processes. The knowledge representation can include product requirements (Holder et al. 2017), function (Elwert et al. 2019), abstract physics (Stetter 2020) and also the results of evaluation processes applying e.g. pareto fronts (Holder et al. 2021). The basic concept of a graph-based design language is illustrated in Figure 2 (adapted from Rudolph 2002).

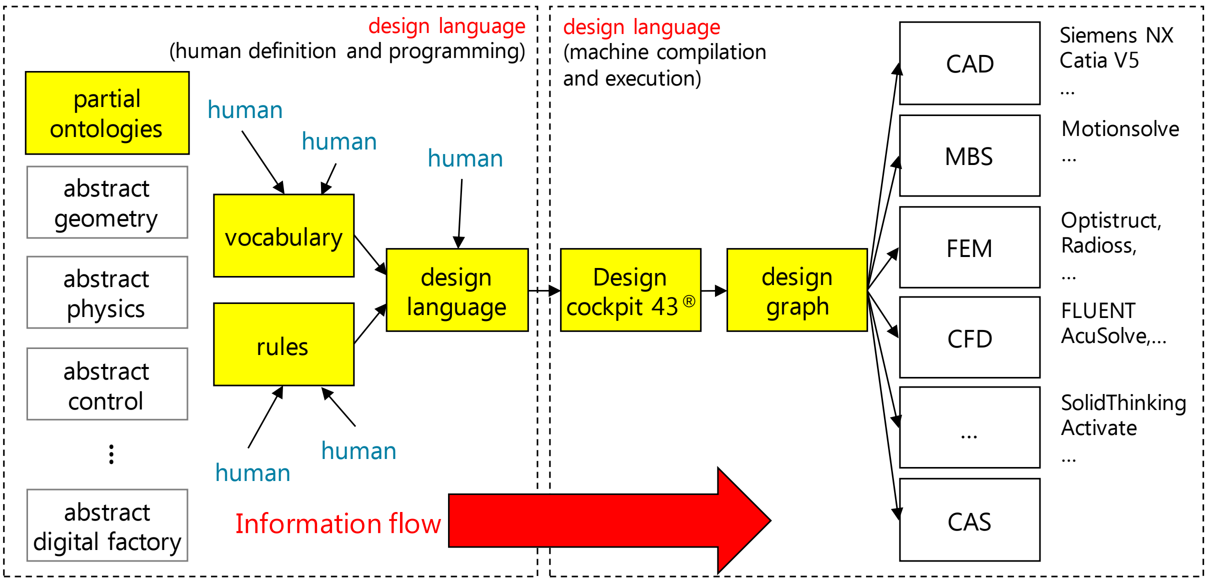


Figure 2. Information flow in graph-based design languages

A knowledge representation in form of a design graph is compiled using design compiler (here in the Design Cockpit 43) which is built from vocabulary (in UML: classes) and rules (in UML: model transformations). The basis for this process are partial ontologies, i.e. formal descriptions of the concepts and relationships that exist for a certain area of knowledge or application, and it defines the basic terms, concepts and relations comprising the vocabulary of a domain. From the resulting representation further domain-specific models (for example geometry (e.g. CAD) and simulation (e.g. FEM) models) can be generated automatically; the design graph represents the central data model.

https://doi.org/10.1017/pds.2022.204 Published online by Cambridge University Press

As a meta-model to store the data for the so called graph-based design languages an international standardized modelling language, the Unified Modelling Language (UML) is used (Gross & Rudolph 2011). The UML has been developed by software engineers and provides many features to model data and software behaviour in an object-oriented manner (Gross&Rudolph 2012). One extremely important characteristic of UML is its meta object facility - each system is an instantiation of a model and each model is the instantiation of a meta-model and each meta-model is the instantiation of a meta-meta-model (Figure 3 – adapted from Störrle (2005)).

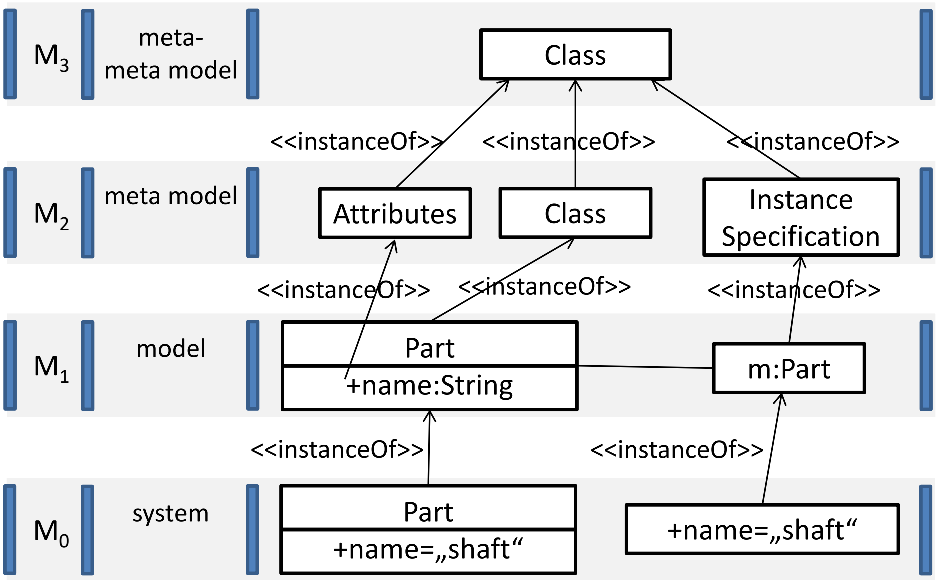


Figure 3. Meta object facility of UML

The main benefit of UML is the integration and standardization of a certain fields of knowledge. It is possible to realise an engineering framework for MBSE applying GBDLs based on UML; potential benefits of the application are shown in Figure 4 (adapted from Rudolph 2021).

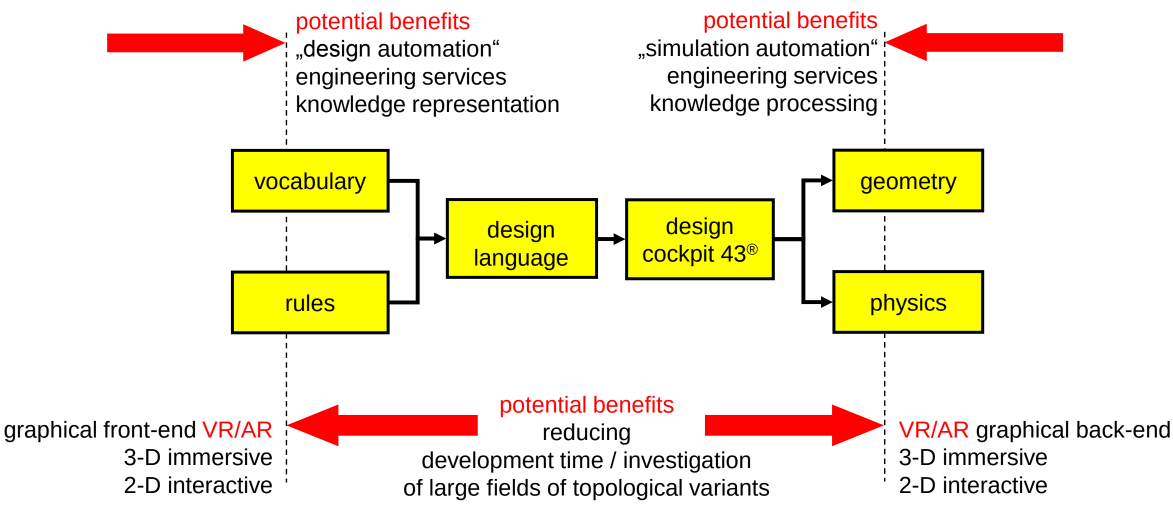


Figure 4. Potential benefits of the application of GBDLs in MBSE

The main benefits include the automation, i.e. digital execution of design and simulation processes thus reducing the development time and allowing an automated investigation of large fields of topological diverse product variants. For the application of GBDLs, they need to be integrated in the product development process; this aspect is described in the next section.

4. Product development process

The product development process (PDP) of a geo-station is based on the product model of the component to be assembled as well as a set of requirements and leads to a complete geo-station design. The essential elements of this process are shown in Figure 5.

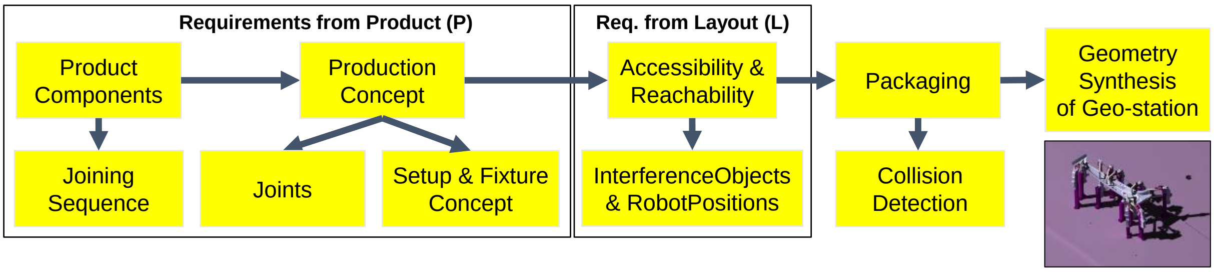


Figure 5. Overview of individual process steps of a PDP for geo-station design

The requirements concern both the product to be assembled and the layout of the assembly system. As use-case for the product serves a b-pillar assembly. For the product components as sensible joining sequence needs to be developed. The production concept includes joints and a clamping (fixture) concept. The design rationale is based on evaluations of accessibility and reachability as well as detailed collision detection. The next sections explain main aspects of this synthesis and evaluation processes in detail.

5. Design rationale for the definition of a joining sequence

The so-called body in white (BiW) describes a phase in which the components of a car body are connected by means of different technologies such as welding (mainly spot welding), gluing, riveting, clinching and folding. The product design describes all components including geometry and material as well as their final orientation towards each other and the joining technologies. From this a joining sequence needs to be developed. The main criteria for this optimization process are accessibility and joining time. An initial sequence can usually be derived from the structure of the product CAD model. There is however no guarantee that this sequence is optimal. Therefore, an iterative search for alternative joining sequences is carried out and the best solution in terms of accessibility and joining time is selected. An exemplary definition of a joining sequence is shown in Figure 6.

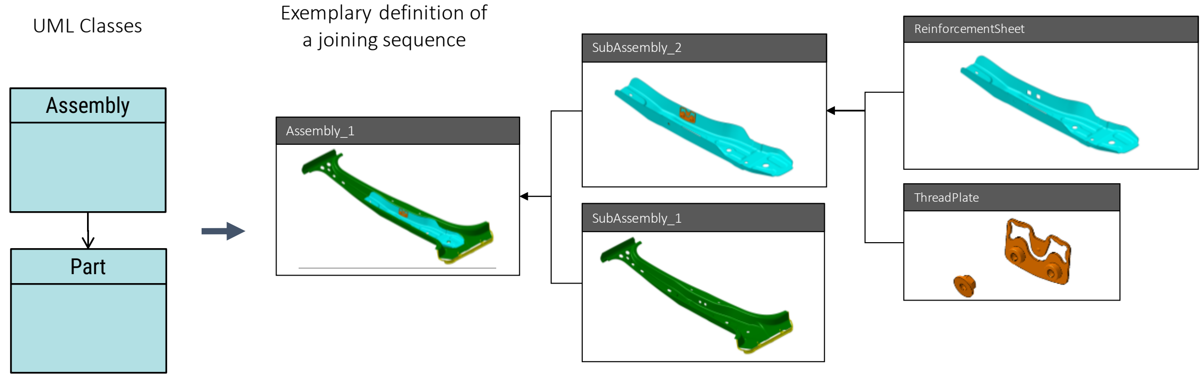


Figure 6. Definition of a joining sequence

Based on the selected sequence, clamping and centering positions can be defined.

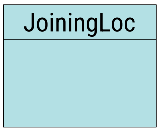
6. Design rationale for clamping and centering positions

The components of a car body are joined by means of different technologies. The joining positions are defined during the product design and are required for fulfilling several requirements concerning the car body e.g. the behaviour in case of a crash. The joining positions are consequently a result of an earlier process, are documented in a so-called joining plan and are considered as given in this paper. These joining positions are also represented by the *JoiningLoc* UML class in the engineering framework based on GBDLs (Figure 7).

Definition of Joints

Point-Nr.	X	Y	Z
001000_GP	1462,32mm	-744,16mm	739,52mm
002000_GP	1473,92mm	-726,77mm	807,67mm
003000_GP	1504,35mm	-726,26mm	647,07mm
005000_GP	1470,38mm	-769,95mm	225,18mm
004000_GP	1393,8mm	-791,1mm	229,56mm
002000_AP	1409,04mm	-739,92mm	650,31mm
003000_AP	1311,24mm	-766,38mm	190,94mm
004000_AP	1429,6mm	-779,28mm	445,66mm
005000_AP	1456,7mm	-779,38mm	443,26mm
001000_AP	1439,3mm	-774,54mm	556,27mm

UML Class



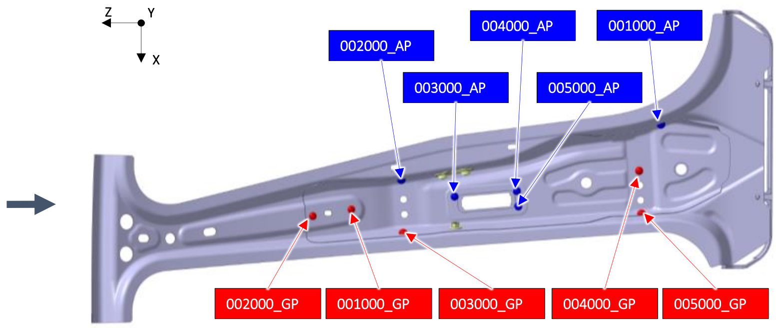


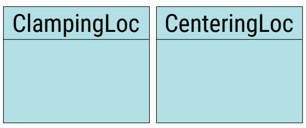
Figure 7. Representation of joining positions

It is important to note that two kinds of joints are defined: geo-points (in red in Figure 7) are important for guaranteeing the right positioning and need to be realised in the first station. Additional points are mainly required for a close connection e.g. in case of a crash (weld out points); they are represented in blue in Figure 7. On the basis of this required joining positions, the clamping and centering positions need to be defined or selected. This is also an iterative process; usually it is ideal to position clamping positions close to the joining positions, because then deformation because of the welding heat are prevented or reduced. Figure 8 shows the definition of the clamping and centering positions.

Definition of Clamping- & Centering-Positions

Point-Nr.	Type	Direction
001	Centering	X, Z
002	Centering	X
003	Centering	X, Z
004	Centering	X
005	Clamping	Y
006	Clamping	Y
007	Clamping	Y
008	Clamping	Y
009	Clamping	X, Y
010	Clamping	X, Y
011	Clamping	X, Y
012	Clamping	X, Y

UML Classes



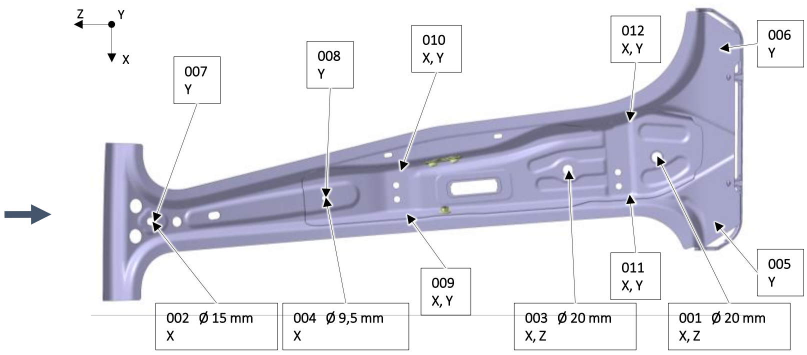


Figure 8. Definition of clamping and centering positions

In conventional engineering, the results of the prior steps (which are an important input for the design rationale of the clamping and centering stations) are represented in three different plans: a joining plan, a clamping plan and a centering plan. In the engineering framework the contents of these plans are represented in a consistent model.

7. Design rationale for clamping and centering units

Once all clamping and centering positions are found and defined, the automated design of the individual centering and clamping units can be executed (compare process steps *packaging* and *geometry synthesis* in Figure 5). The packaging-process deals with a collision free placement of clamping and centering units alongside respectively below the product contour. The *geometry synthesis* afterwards deals with creation of geometric models of *clampingUnits* and *centeringUnits*. It is appropriate in the engineering

https://doi.org/10.1017/pds.2022.204 Published online by Cambridge University Press

framework to decompose these units and to represent them in the engineering framework in an abstract manner (Figure 9).

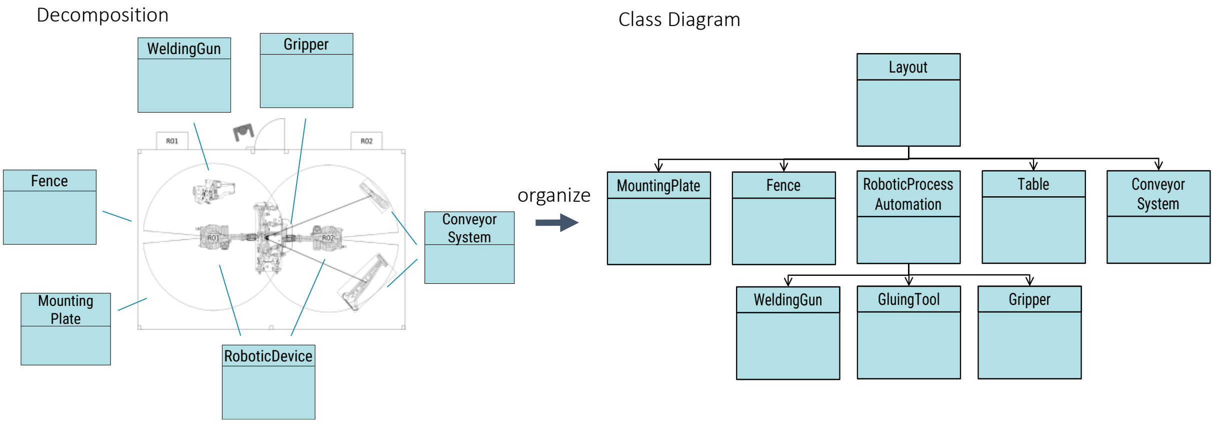


Figure 9. Decomposition and abstraction of the components of a physical centering and clamping unit

A main challenge in the design of a geo-station is to avoid collisions between the different units, i.e. the welding guns on an industrial robot as well as between centering and clamping units themselves which are used for aligning and fixing the car body components in space. The number of possible solutions in the packaging process step is extremely high and an optimum solution can only be found by an iterative search based on certain heuristics. In order to allow fast evaluations, it is sensible to use abstract and simplified interference objects (Figure 10).

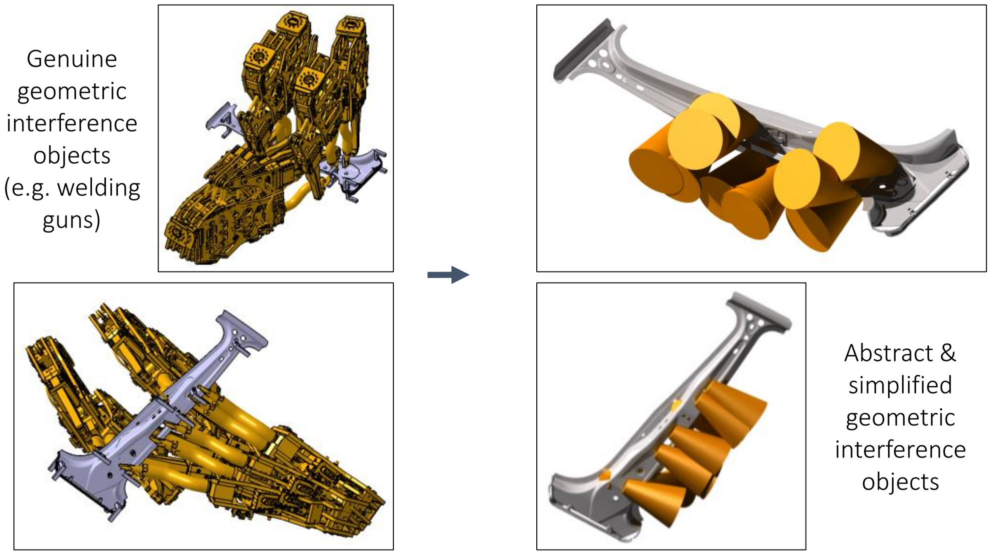


Figure 10. Dealing with interference objects

As the achievement of the required joining positions has the highest priority, it is initially evaluated which connection units (e.g. welding guns) in which orientation may allow a good access of all joining positions. These connection units are abstracted and simplified. For each joining point, an interference object (represented by geometrical cones (orange in figure 10)) is created. Initially, it is attempted to ensure a close contact by connecting a line normal to the outer contour with the respective joining point. Usually a collision between the interference objects will be present and an iterative search will be necessary to allow a placement of these interference objects relatively close to the joining positions and without collisions. This placement is than the starting point for the placement (packaging) of the clamping and centering units. The main part of these units is abstracted into a cylinder (turquoise and light green colour). The placement procedure of these abstract clamping and centering units is explained in Figure 11.

https://doi.org/10.1017/pds.2022.204 Published online by Cambridge University Press

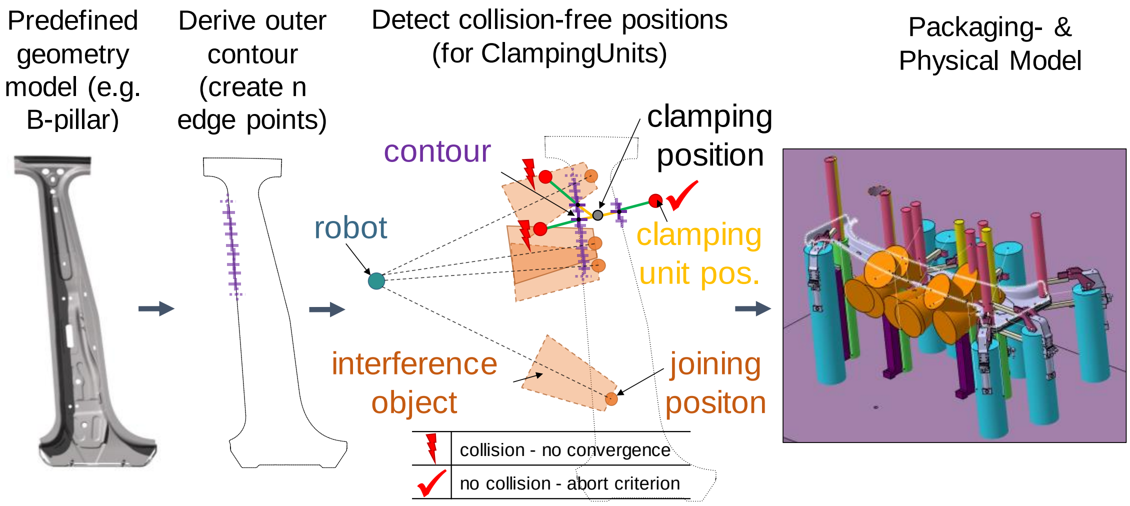


Figure 11. Synthesis of the packaging model

The main steps of the synthesis of a packaging model of the clamping and centering units are described in Figure 11. From a predefined geometry model (in this case a b-pillar of a car body) the outer contour is automatically derived and discretized by means of n contour point. Also an initial robot base position is defined based on rather simple heuristics (green point). In Figure 11 also four joining positions (light brown/orange points) are visible together with interference objects (cones) which represent the connection units (e.g. welding guns) in an abstract manner. Also a desired clamping position is visible (dark grey point). An algorithm now seeks to find a clamping unit position which is on a straight line from the clamping position and is as close as possible to the clamping position, because in this case a rather simple and rigid clamping mechanism can be used. Possible clamping unit positions (red points) are then checked whether there is a collision with the interference object (two such cases are visible in Figure 11). The green lines in Figure 11 show the distance from the clamping position to the clamping unit position. When a collision is present, new clamping unit positions are sought and tested. One collision-free position (convergence criteria of underlying algorithm) is also visible in Figure 11. It is important to note that this position is on the other side of the b-pillar as the other two investigated positions. It was found to be necessary in the testing of the algorithm that it is able also to "jump" to the other side, i.e. to investigate possible clamping unit positions on all sides of a geometry. In many cases the described procedure will lead to several possible clamping unit positions for each clamping position. In a second step a genetic algorithm is applied for creating suitable solutions considering all clamping and centering units.
From the resulting package model a concrete model of the geo-station can be derived (Compare process step *geometry synthesis* in Figure 5). A result of automatic designed geo station is shown in Figure 12).

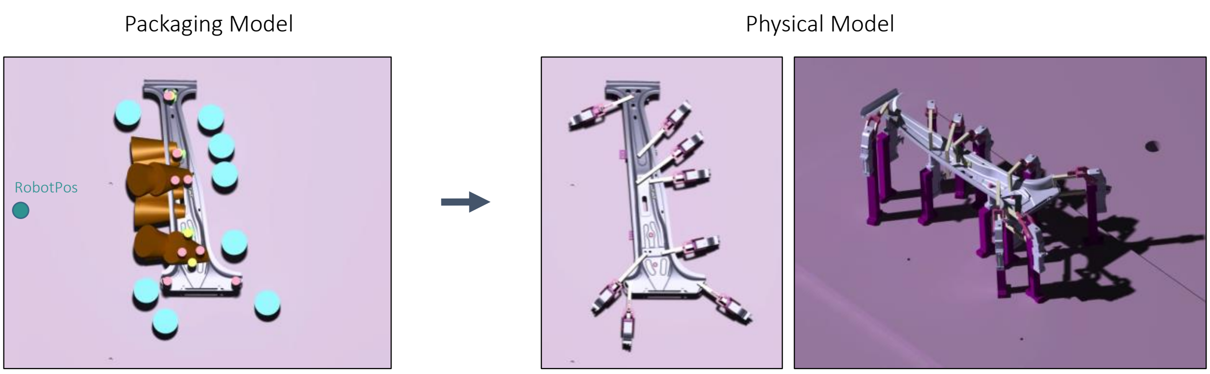


Figure 12. Synthesis of the concrete model of the geo-station

For all clamping and centering units the most important information is available (knowledge representation in design graph): the unit position, the length to the clamping or centering position which

https://doi.org/10.1017/pds.2022.204 Published online by Cambridge University Press

determines the length of the arm and the height of the clamping or centering point which determines the height of the unit column. The respective components are selected from a database of possible components and one specific part "contour part" is defined using the shape of the respective car body part. It is important to note that the robot base position (blue point) can be shifted and that the cones would adapt themselves accordingly. These would lead to an adapted package of the clamping units. The robot base position can then be optimised using inverse kinematics and the complete station and system layout can be developed (Figure 13).

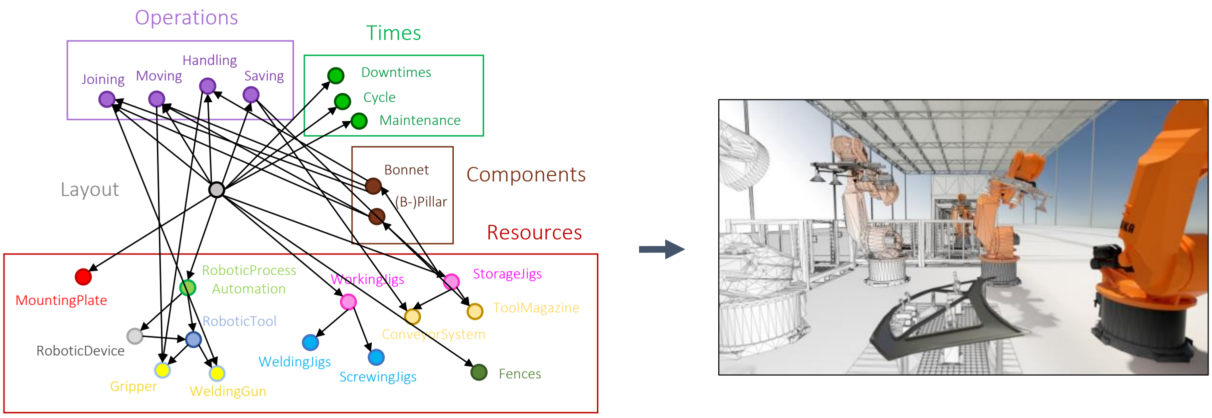


Figure 13. Development of the station and system layout (knowledge representation in the design graph)

8. Conclusions and outlook

For the parts of a car body, it is an extremely difficult task to develop the geo-stations which ensure the positions and orientations of several product components and allow their connection. It is nearly impossible to find an appropriate space for all clamping and centering units which avoids collisions with connection units such as welding guns. Usually, in an enduring and long-lasting iterative process, a solution is found by human expert designers.

This paper describes an automated algorithmic approach that will result in several solution possibilities, their evaluation and selection and the possibility to store the design rationale for a selected design in an engineering framework. This framework can on the one hand allow an automated execution of the design tasks. On the other hand, it allows to explore synthesis, evaluation and selection process steps and - by doing so - to explore the design rationale.

The realization of this framework is based on graph-based design languages in combination with UML. Currently, the underlying algorithms are optimized in order to generate further improved designs. It is important to note that the presented solution was mainly developed using the example of car body components and their assembly. However, the general nature makes this methodology appropriate for all design problems that concern the placement of certain elements in a restricted space. Further application areas will also be the focus of further research.

Acknowledgement

Some parts of this work were supported by a grant from the European Regional Development Fund and the Ministry of Science, Research and the Arts of Baden-Württemberg, Germany. (More information https://efre-bw.de/) as a part of the project “Digital Product Life-Cycle” (ZaFH, see https://dip.hs-weingarten.de for details). Further parts were supported in the scope of the project "Automatisierter Entwurf eines geometrischen und kinetischen digitalen Zwillings einer Rohbaufertigungsanlage für die Virtuelle Inbetriebnahme (TWIN)" which is funded by the German Federal Ministry of Education and Research.

References

Cao, Y.; Liu, Y.; Paredis, C. J. J.: System-level model integration of design and simulation for mechatronic systems based on SysML Mechatronics 21 (6) 2011, pp. 1063-1075, https://doi.org/10.1016/j.mechatronics.2011.05.003.

Darpel, S.; Beckman, S.; Ferlin, T.; Havenhill, M. Parrot, E.; Harcula, K.: Method for tracking and communicating aggregate risk through the use of model-based systems engineering (MBSE) tools. Journal of Space Safety Engineering 7 (2020) 11–17, https://doi.org/10.1016/j.jsse.2020.01.001.
Ellman, A., Paronen, J.; Juuti, T.S., Tiainen, T.: Re-use of engineering design rationale in Finnish SME project based industry. In: Marjanović D., Štorga M., Škec S., Bojčetić N., Pavković N.: Proceedings of the DESIGN 2018 15th International Design Conference, 2018, pp. 1825-1832.
Elwert, M.; Ramsaier, M.; Eisenbart, B.; Stetter, R.: Holistic Digital Function Modelling with Graph-Based Design Languages. In: Proceedings of the Design Society: International Conference on Engineering Design / Volume 1 / Issue 1, Cambridge University Press: 26 July 2019, pp. 1523-1532.
Gräßler, I.; Hentze, J.: The new V-Model of VDI 2206 and its validation. at – Automatisierungstechnik 2020; 68(5): pp. 312–324, https://doi.org/10.1515/auto-2020-0015.
Gräßler, I.; Wiechel, D.; Pottebaum, J.: Role model of model-based systems engineering application, 2021 IOP Conference Series: Material Science and Engineering, 1097, 012003.
Groß, J., Rudolph, S.: Hierarchie von Entwurfsentscheidungen beim modellbasierten Entwurf komplexer Systeme. Tag des System Engineerings. 2011.
Groß, J.; Rudolph, S.: Generating simulation models from UML – a FireSat example. In: Proceedings of the 2012 Symposium on Theory of Modeling and Simulation – DEVS Integrative M&S Symposium. San Diego: Society for Computer Simulation International, 2012.
Heisig, P., Caldwell, N.H.M., Grebici, K. and Clarkson, P.J. (2010), "Exploring knowledge and information needs in engineering from the past and for the future - Results from a survey", Design Studies, Vol. 31 No. 5, pp. 499–532. https://doi.org/10.1016/j.destud.2010.05.001.
Holder, K., Zech, A., Ramsaier, M., Stetter, R., Niedermeier, H.-P., Rudolph, S. and Till, M. (2017) "Model-Based Requirements Management in Gear Systems Design based on Graph-Based Design Languages". Appl. Sci. 2017, 7, 1112, https://doi.org/10.3390/app7111112.
Holder, K.; Schumacher, S.; Friedrich, M.; Till, M.; Stetter, R.; Fichter, W.; Rudolph, S.: Digital Development Process for the Drive System of a Balanced Two-Wheel Scooter. Vehicles, 2021, Vol. 3, Issue 1, pp. 33-60. https://doi.org/10.3390/vehicles3010003.
IILS mbH. Design Compiler 43. Available online: https://www.iils.de (accessed on 04 Mai 2021).
Kruse, B., Shea, K.: Design Library Solution Patterns in SysML for Concept Design and Simulation. In: Proceedings of the 26th CIRP Design Conference, Procedia CIRP, vol. 50, pp. 695–700 (2016). https://doi.org/10.1016/j.procir.2016.04.132
Laing, C.; David, P.; Blanco, E.; Dorel, X.: Questioning integration of verification in model-based systems engineering: an industrial perspective. Computers in Industry 114 (2020) 103163 https://doi.org/10.1016/j.compind.2019.103163.
Lee, J., Design rationale systems: Understanding the Issues. IEEE intelligent systems, 1997, 12(3), pp78-84.
Reichwein, A.: Application-specific UML Profiles for Multdisciplinary Product Data Integration. PhD thesis, Universität Stuttgart, 2011.
Riestenpatt gen. Richter, M. and Rudolph, S.: A scientific discourse on creativity and innovation in the formal context of graph-based design languages. 13th Anniversary "Heron Island" Conference Workshop on Computational and Cognitive Models of Creative Design (HI'19), Heron Island, Queensland, Australia, December 15–18, 2019.
Rudolph, S. Übertragung von Ähnlichkeitsbegriffen. Habilitationsschrift, Fakultät Luft- und Raumfahrttechnik und Geodäsie. Habilitation Thesis, Universität Stuttgart, Stuttgart, Germany, 2002.
Rudolph, S. A Semantic Validation Scheme for Graph-Based Engineering Design Grammars. In: Gero, John (ed): Proceedings Design Computing and Cognition (DCC'06), Springer, Dordrecht, 541–560, 2006
Rudolph, S.: Vorlesungen Digitaler Produktentwurf. Universität Stuttgart 2021.
Shaked, A.; Reich, Y.: Using Domain-Specific Models to Facilitate Model-Based Systems-Engineering: Development Process Design Modeling with OPM and PROVE. Applied Sciences 2021,11, 1532. https://doi.org/10.3390/app1104153
Stetter, R.: Approaches for Modelling the Physical Behavior of Technical Systems on the Example of Wind Turbines. Energies (2020), Vol. 13, No. 8, 2087, https://doi.org/10.3390/en13082087.
Störrle, H.: UML 2 für Studenten. München: Pearson 2005.
Till, M.; Stetter, R.; Rudolph, S.: Multi-disziplinäre digitale Repräsentation des Produktlebenszyklus auf der Basis graphenbasierter Entwurfssprachen. In: Forschungsreport für den Maschinenbau in Baden-Württemberg 2016. Bingen: Public, 2016. S. 3 – 6. ISSN 2196-8659.
VDI/VDE 2206 – Entwurf: Entwicklung cyber-physischer mechatronischer Systeme (CPMS). Beuth: 2020.
Zech, A.; Stetter, R.; Till, M.; Rudolph, S.: Automated Generation of Clamping Concepts and Assembly Cells for Car Body Parts for the Digitalization of Automobile Production. Stuttgart Conference of Automotive Production SCAP 2020, 2nd to 10th November 2020.