INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.61

Methodology for Early Design Phase Cost and Performance Trade-Off Analysis of a New Variant in a Product Family

M. Schaffers, E. Rosseel, S. Burggraeve✉, J. Pelfrene, K. Gadeyne and F. Petré

Flanders Make, Belgium

✉ sofie.burggraeve@flandersmake.be

Abstract

Given the Industry 4.0 trend towards smaller lot sizes and large product families, there is a need to increase the design efficiency in terms of cost and lead time. Design decisions taken in the early or conceptual design phases are shown to have the greatest impact on the final product cost, however the choices and trade-offs are often made in an ad-hoc way. This paper presents a structured methodology for formalizing early stage product design knowledge, which is key for designing and evaluating new variants in a product family in the early stages of the design process.

Keywords: design knowledge, product families, early design phase, cost-performance trade-offs, knowledge formalisation

1. Introduction

In the past decade, companies have seen a growing customer demand to offer more and more variants, a trend that continues given the Industry 4.0 market dynamics towards smaller lot sizes and larger product families. This increasing number of variants brings additional complexity in the design process, where a major driver for costs of nowadays mechatronic systems are variant driven complexity costs (Asadi et al., 2016). The trend towards lot size one at the cost of mass production requires a drastic increase in design efficiency since the development cost is spread across smaller product series.

When looking at increasing design efficiency in terms of cost and lead time, the phases of the product development process that would bring the greatest impact are the concept and design phases. Decisions taken in these early phases have the greatest impact on the final product cost, as confirmed by Thomke (2001): decisions made within the first 20% of the design cycle allocate up to 80% of the total production costs. It is therefore of utmost importance to take the right design decisions in these phases to avoid late and expensive design iterations causing delays and additional design cost. We observe however that often these decisions in the very early phases of design are taken in an ad-hoc way based on the intuition and experience of system architects and senior designers.

This paper presents a new methodology to support designers in the early design phase when introducing a new product variant in an existing product family. By creating a formal framework to capture the designer knowledge, cost and performance information of previous products in a product family can be reused in order to assess designs for new product variants. A big challenge in the formal modeling of a product family is to avoid redundancy in the specification of the product model for each variant. Conventional modeling methods fail to address this problem (Zeng et al., 2014). In particular there is a challenge in combining structural based modeling and feature based modeling. Amongst various existing solution approaches - for example the Modularized Generic Product Model (Zeng et al., 2014) or by using the UML language (Bonev et al., 2015) - there is no consensus yet. In this paper, a new metamodel is proposed to create a Domain Specific Language (DSL) that is built upon the

Eclipse Ecore modeling framework (see https://wiki.eclipse.org/Ecore), which provides a built-in interface for facilitating the interaction with the design end-users.
The proposed method allows one to determine whether it is worthwhile to introduce a new variant in a product family by efficiently and quantitatively trade-off cost and performance of the different product variant design candidates in an existing product family in an early stage of the design process. Moreover, an insight in which design choices yield the best trade-off between cost and performance is obtained. Furthermore, a preview is included on how the approach can be extended towards multiple levels of abstraction, which play a key role when the New Product Introduction Process consists of multiple design stages.
Note that this work focuses on knowledge formalization for product design, while there exist other solutions that deal with product family design and portfolio management. For instance, Jiao et al. (2007) and Windheim (2020) describe methodologies for optimizing the product offering, taking into account, for example, component standardization, customer value and production organisation. The difference in focus is reflected in the metamodels of both type of approaches. The proposed metamodel for product design is prescriptive and the concept of design choice and its link with cost and performance is key, while existing models for product family management and optimization are descriptive with respect to product design, with the consequence that design choices are implicitly dealt with or restricted to product or component attributes, but not tackling architectures. In contrast with Product Lifecycle Management (PLM) and Product Data Management (PDM) tools which are dominant methodologies to manage product data, models and processes, the present approach focuses on product knowledge management. Another differentiator between the here proposed approach and PLM and PDM tools is that the focus is on knowledge in the early design phase, when there is the need to deal with incomplete information or with restricted resources (time, personnel).
This paper starts by introducing the concepts and terminology on product design knowledge formalization in Section 2. Section 3 presents how the designer's knowledge can be formalized by using a metamodel. The management of product family design data is considered in Section 4, where we also discuss our approach to trade-off cost and performance of designs. We illustrate these concepts on the use case of designing a new exhaust cold end system. Section 5 is dedicated to a reflection on an early phase design process consisting of multiple design stages. Section 6 summarizes the main conclusions of this paper and gives an outlook on open research challenges.

2. Product design knowledge formalization

2.1. Product family concepts

Many different definitions and interpretations of 'product family' are present in manufacturing companies and in academic literature. For instance, by Simpson (2004), ***a product family*** is a group of related products that is derived from a product platform to satisfy a variety of market niches, while a ***product platform*** can be seen as the collection of assets, i.e., components, processes, knowledge, people and relationships, that are shared by a set of products (Robertson et al., 1998).
In this paper, we consider another definition of product family, where each set of products that satisfies at least one of the following constraints is considered as a product family:

- set of products with the same physical structure, i.e., consisting of the same type of components,
- set of products with the same functionality structure, i.e., the functionality is measured by the same set of indicators,
- set of products that form a context for a cost and performance evaluation.

Such a product family can be represented by a ***variability model*** which describes the set of decisions and possible choices to select the different products of the considered product family. These choices control the alternatives in the family. Compliant with the international standard ANSI/ISA-95.00.02 (2010), a ***product definition*** consists of all the information required to make a product, including a bill of resources, bill of materials, and a product production rule.

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

A ***variant*** corresponds to a product defined based on another product by specifying a few deltas with respect to the original product. When designing a product variant, multiple ***design alternatives*** can be proposed to fulfil some given requirements and for which the optimal design must be found.

2.2. Introduction of a new variant in a product family

To assess whether it is worthwhile to introduce a new variant into an existing product family when looking at all the associated costs and to determine which design choices will result in the best trade-off between cost and performance for the new variant, we formulated 4 research challenges:

1. Cost and performance models are typically not linked: at an early design stage it is important to link design choices to cost and performance trade-offs.
2. There is little reuse of design knowledge across a new product introduction (NPI) process for a product family or this knowledge is only available in the head of senior designers.
3. Appropriate abstraction levels are not existing. A design process typically follows a stage gate procedure where design decisions are being made at different stages in the process. Therefore, it is important to have initial cost/performance info available at higher levels of abstraction to account for depending design decisions made in a later design stage.
4. There is no reuse or traceability across the abstraction levels or stages in a design process: there is a need to trace back why certain changes to cost or performances occurred and to compare different scenarios for product variants.

This paper proposes a methodology to solve the first two identified research challenges and gives a preview on how the approach can be extended to include the third research challenge on dealing with multiple abstraction levels as well.

When introducing a new variant in an existing product family, early phase design cost-performance trade-offs can be supported by using available knowledge from the product family. Different types of reuse can be considered:

- using knowledge on the structure of the product family, for example the available functionalities, modules, performance metrics or incompatibility conditions;
- using knowledge on the behaviour of the product family, for example analytical formulas or logic models that express the relation between the structure of the product family and its cost or performance;
- using historical product data in order to derive cost and performance for new combinations of design choices.

In order to reuse information on the structure and behaviour of the product family, we introduce a new approach to ***formalize the design knowledge*** of the product family domain in Section 3. Separating the definition of a product family from its product instances is key to our methodology as it offers a structured view on a product family and a centralization of the domain knowledge so that all available knowledge can be reused without being dependent on the expertise of senior designers.

In the early or conceptual phases, we observe that quantitative models for estimating cost and performance values are often lacking, for example performance metrics that deal with acoustic noise levels or durability require access to detailed CAD or finite element models in order to numerically simulate the expected performances. Also, a completely detailed bill-of-materials (BOM) together with the sequence of assembly operations is only available during later design stages in order to compute the total product cost. To circumvent the need for such a high level of detail, we will reuse information from existing products in the product family. Based on historical design data, a ***regression analysis*** can be set-up in order to build parametric models for linking cost and performance properties to design choices.

3. Formalisation of design knowledge

This section introduces a meta-model for building a Domain Specific Language that formalizes product family design knowledge. This model deals with the definition of the product family and will be linked in Section 4 to a model that is used for handling the actual design data. This close link between the definition and data models requires in practice a synchronization mechanism so that updates to the definition of the

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

product family triggers updates in the design data as well, for example in the case of changing the cost calculation models.

3.1. Structures for cost, performance and design decisions

In order to uniquely define a product family, we distinguish the design choices to describe the product, from the cost incurred when making the product, and the performance output when using the product. This leads to the introduction of a design space, cost space and performance space, as graphically illustrated for the simplified design definition of a projector in Figure 1.

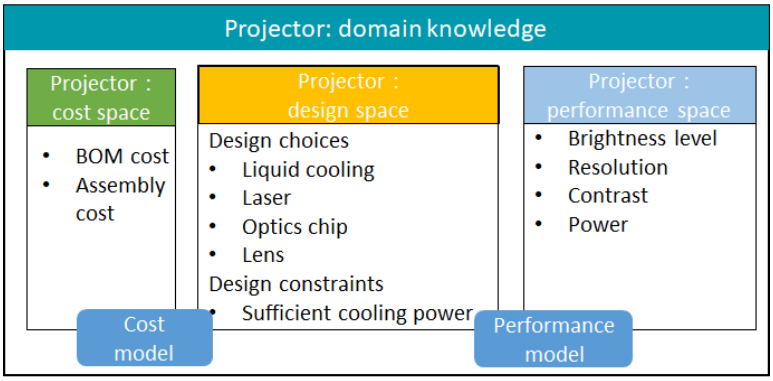

Figure 1. Example design definition for a projector product family

The ***design space*** represents the variability model, which was introduced in Section 2. It groups the design choices together and holds information on the feasibility of a design, which is expressed in the form of ***design constraints***. The design choices are either simple options such as the material type and sizing parameters, or can represent the physical model, i.e., the components which (on their turn) represent also products, typically from different product families. The latter enables a representation of the BOM structure in the design space. The design constraints contain the logic to verify whether certain combinations or options for the design choices are possible. Design constraints are hard constraints: unfeasible designs cannot be accepted.

The ***performance space*** holds the metrics to express the ***value*** of a design. A performance space or functionality structure can be hierarchical or constrained. We consider a constrained structure (D. Benavides et al., 2005), which implies that all performance dimensions are orthogonal: for example, the acoustic noise levels, fatigue resistance and weight of an exhaust system.

The ***cost space*** represents a set of orthogonal dimensions providing metrics for computing the cost of producing a product. These metrics are not necessarily expressed in monetary terms, but can also represent the resources (for example amount of energy in kW) needed during production or the number of labour hours.

The ***cost and performance models*** hold the logic on how the cost values, respectively performance values, can be derived from the selected design choices. These models can take the form of a mathematical relationship based on physical formulas or of parametric models whose parameters are derived from a regression analysis using historical data, as explained in Section 2.2. For example, via an aggregation model, the BOM cost can be obtained by summing the material cost of the components. Note that we make the assumption that designs are compositional with respect to cost and performance, that is, the cost and performances can be derived from the cost and performance properties of the components.

3.2. Metamodel

Using the formalization introduced in the previous section, we arrive at a metamodel for representing the design knowledge of a product family, as shown in Figure 2. This metamodel is implemented in the Eclipse Ecore Modeling Framework. The central ***domain*** block holds the knowledge for a particular domain, for example, all the domain knowledge on projector designs.

Noteworthy is the link between the ***components*** and the ***performance space***: indeed we consider a functional description of a component which expresses the required functionalities that a component

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

provides. This prevents the need for the designer to fix all the properties of the components at an early stage and allows to easily create new variants by exchanging components which offer the same functionality. For example, the cooling component of a projector is characterized by the amount of cooling power it provides and the designer is free to select later on different types of cooling components, such as air or liquid cooling devices, that fulfil the cooling power condition.

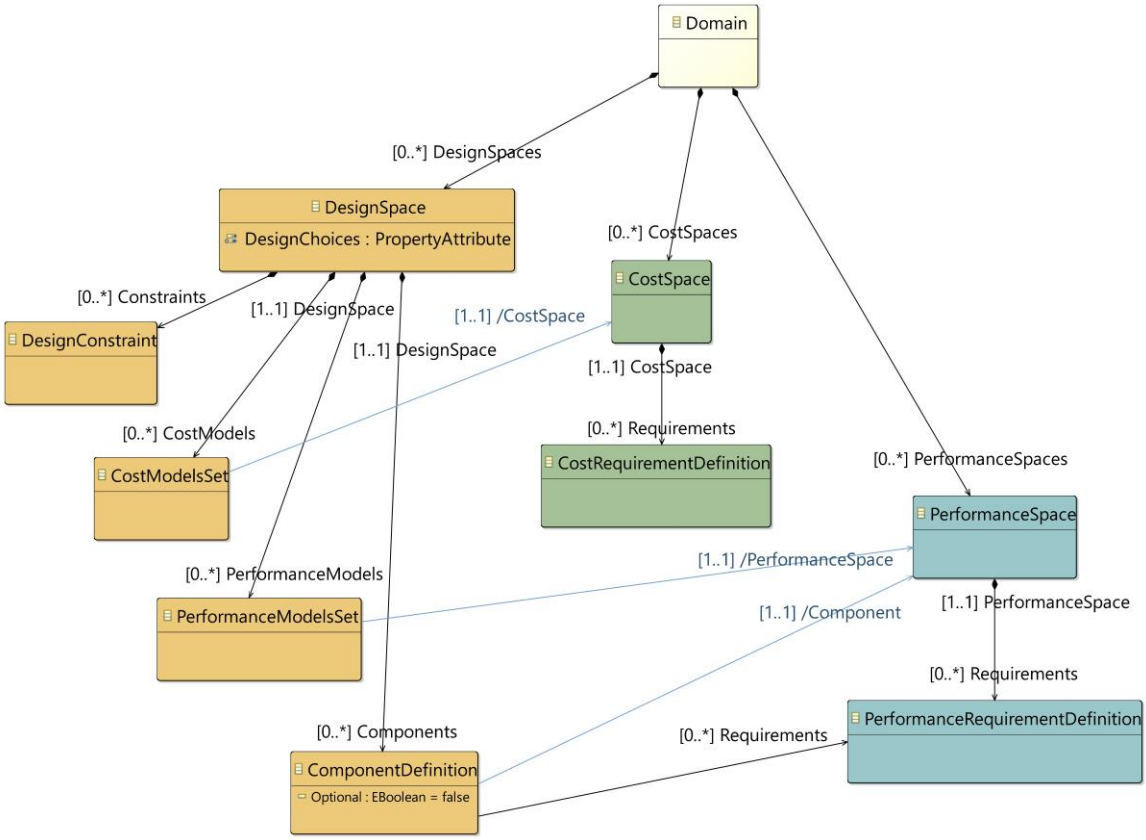

Figure 2. Metamodel formalizing the design knowledge in a product family

Another new element in Figure 2 are the ***cost and performance requirement definitions***. These requirement definitions formalize the types of requirements that could be made on the product. They specify logics in terms of cost and performance (e.g. 'light output ≥ B lumen', with light output a (parametrized) formula and B a placeholder). These definitions are later used to specify actual user requirements (meaning that the user chooses a concrete value for bound B). Such user requirements can be interpreted as the value of a product for the user. The present research does not go deeper in modeling the value of a product. These requirements are soft constraints on the design, in contrast to the design constraints.

4. Trade-off analysis of a new variant in a product family

4.1. Managing design data

The design data representing design alternatives for variants in a product family is to be managed in compliance with the formalization of design knowledge. Figure 3 shows the corresponding metamodel to handle the design data in a product family. The link between the design definition and the design data is illustrated by the connections between the blocks coming from the metamodel of Figure 2 and the new concepts introduced to manage the actual design data, see for example the relation between the `DesignSpace' concept and the `Design'. The ***design space*** defines the design choices and constraints for a product in the product family, while a ***design*** is an actual product design for a variant in the product family. For each of the design choices, an option has been fixed, as a result, the feasibility, cost and performance models from the product definition can be evaluated to assess the design constraints,

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

cost and performance values of the design at hand. Remark that the design represents a type of product, but not an actual product, for example it cannot be associated with a serial number.
A ***design component*** associates the design to the design selected for some part of the product. The selected component design adheres the functionalities of the component as defined in the design space. By construction, a design component is an instance of another product family. As such, the methodology is recursive, a design component has its own cost, performance and design space.
The ***design cost and design performance*** hold the values which result from applying a cost, respectively performance model, to the design instance.

4.2. User requirements

It is important to assess the value of a design in order to decide whether this design is a valuable design alternative for a product variant. A new product variant is typically designed to meet certain user requirements. In the framework, a user requirement is made for a request to a particular product, holding all the requirements for that product For example, when a new type of exhaust system is designed, it must satisfy some weight criteria according to the target vehicle platform and meet some acoustic noise levels according to the applicable norms and regulations. These requirements should not be confused with design constraints, since the actual target values will depend on the specific client, while design constraints are hard constraints that hold on all the products of the family. Figure 3 illustrates our approach for handling user requirements: a set of user requirements can be evaluated simultaneously on a selected group of designs. Each user requirement refers to a cost or performance requirement definition, which are part of the design knowledge formalisation metamodel, as shown in Figure 2.
Since the evaluation of a user requirement typically uses the cost or performance values of a design, the user requirements functionality is not limited to evaluating which designs meet the requirements, but at the same time an overview of cost and performance values of the different designs can be derived, which allows to sort the candidate designs that satisfy the user requirements according to cost or performance values.

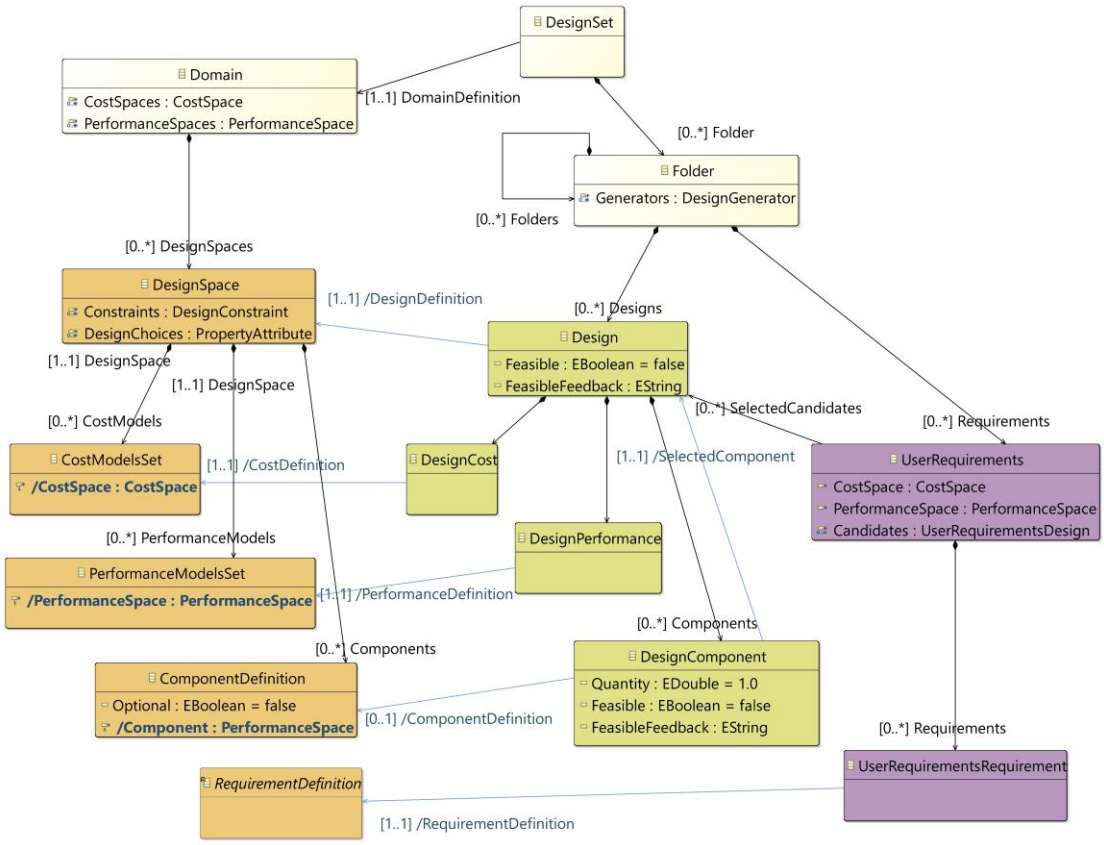

Figure 3. Metamodel on the management of design data and requirements

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

4.3. Example of an exhaust cold end design

To illustrate an application of the proposed methodology, we consider the design of exhaust cold end systems. Such an exhaust system consists of pipes and silencers, and is located under the car chassis, excluding the emission aftertreatment components such as catalytic converters. A large variety of exhaust systems results from different types of vehicle platforms, engine properties, gear boxes, chassis body variants and emission legislations. The constraints and performance requirements coming from these drivers for diversity need to be outbalanced against each other and against the associated product cost in order to arrive at the best product designs.

Our methodology allows the exhaust experts to formalize their available design knowledge. This step requires input from experts of different departments, e.g. design engineers, acoustic engineers and costing engineers. In this restricted example, their expertise and analysis of existing products lead to the identification of six components together with the exhaust main design choices: pipe diameter, thickness and material. In addition three performance metrics (which are key when dealing with the OEM requirements) and one cost indicator are formalized, as illustrated in Figure 4. The BOM cost is computed by an aggregation of the material cost of the components. The value of the weight performance indicator can be estimated analytically from the component dimensions and material. Furthermore, the experts defined parametric models for computing the tailpipe noise and backpressure values, using historical data and regression analysis. Figure 4 gives a view on the formalized knowledge.

The added value of the proposed methodology becomes clear when a new request for quotation for a new variant is submitted to this company. The formalized knowledge will help a designer (less or more experienced) in an early design phase to keep the overview over all possible design choices and their impact on cost and performance. As such, the method will support the designer to come to one or more design alternatives that satisfy the user requirements. Furthermore, the method makes it possible to compare the alternatives for cost and performance, even in an early design phase.

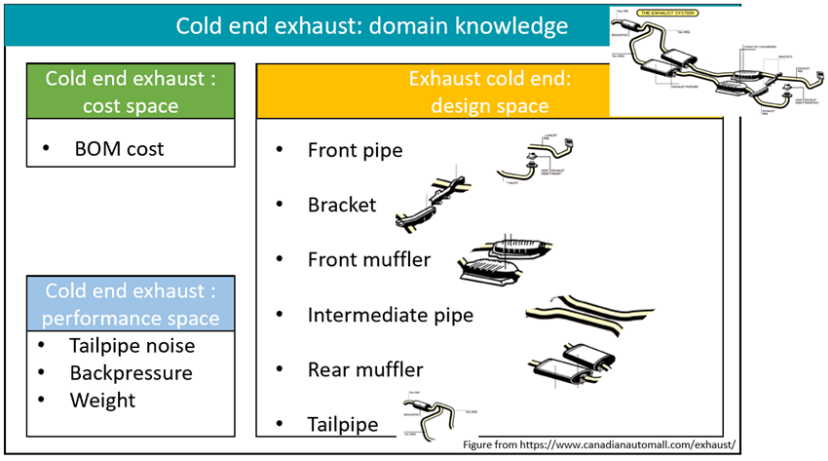

Figure 4. Use case: design of a exhaust cold end system

5. Towards multiple levels of abstraction and design stages

The design process when introducing a new product typically runs through different steps or design stages. In an initial stage, trade-off analyses during a value engineering exercise are based on rough performance evaluations - typically using rule-of-thumb heuristics - and cost estimation. A first selection takes place and the design of the remaining valuable candidates is refined using more detailed cost and performance models, and taking into account more detailed component designs. For a meaningful comparison of the design candidates and an optimization of the product family, it is important that the data models, cost and performance functions at their respective abstraction levels are linked together and kept consistent. Therefore the metamodels presented in Figures 2 and 3 need to be extended to support the evolution of design knowledge, technologies and processes in a modular way.

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

5.1. Abstraction level

To formalize the introduction of a new product variant in a multi-stage design process, we start from the concept of ***abstraction level***, which is defined as a space representing a set of observables or indicators for a system of interest (Floridi, 2008). Considering the concepts introduced in Section 3, we can identify the following systems of interest: a cost aspect (a set of cost indicators expressing a measure of resources needed to create a product), a performance aspect (a set of performance indicators measuring the value of a product), a design (a set of decisions - the structure of the product, the manufacturing or assembly process, logistics or packaging aspects - made to produce a product). We define a ***subsumption relation*** between a higher and lower abstraction level for a system of interest: the lower abstraction level is a refinement of a higher abstraction level and every observable of the higher abstraction level is also present at the lower abstraction level, possibly next to new observables. So every observable of the higher level is inherited by the lower level. Going down the abstraction levels, the domain is (possibly) reduced (this is the subsumption) while the space of observables is (possibly) augmented (this is the refinement).

Considering the use case of designing a new exhaust cold end system, we can identify for example a subsumption relation both for a series of design structures as well as for different performance aspects, as illustrated in Figure 5. In bold, the additional indicators for a lower abstraction level of the design structure are indicated. The performance aspect tailpipe noise is being refined by considering smaller tolerances on the noise level values. As a result, more accurate models will be required to compute the lower level performance values. At the highest abstraction level, the tailpipe noise results from a regression based on a quadratic model depending on the motor temperature and flow, while at the lower abstraction levels 1D to 3D CFD computations are used for computing the noise levels.

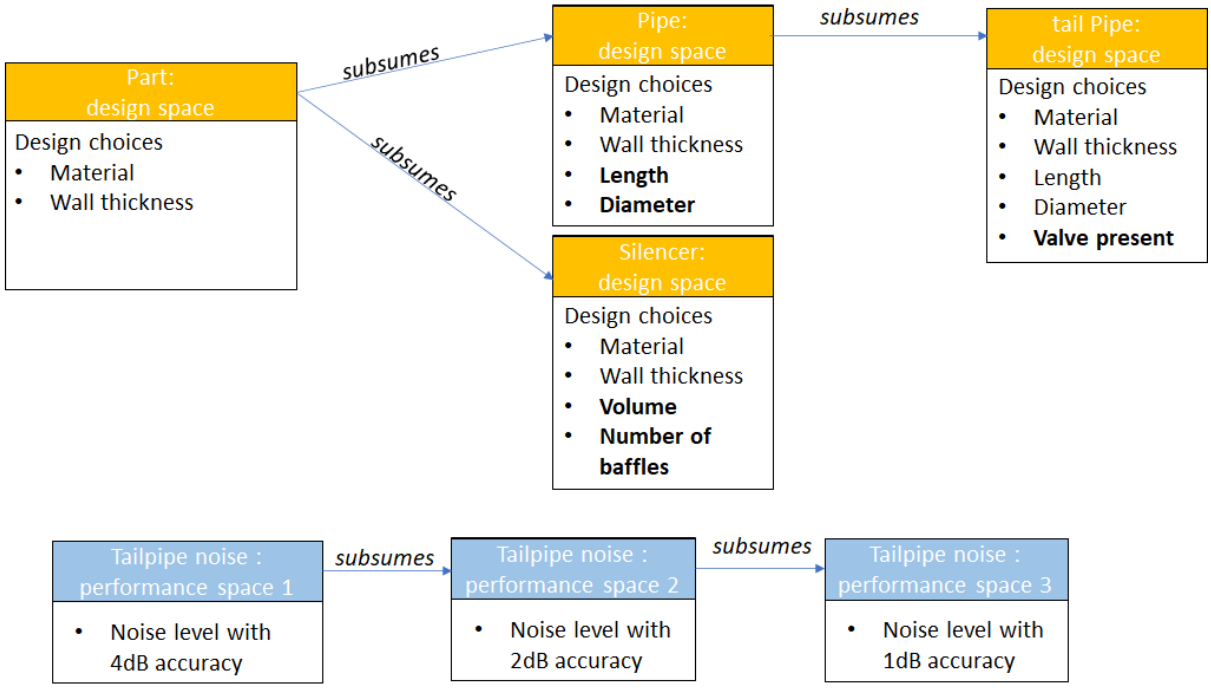

Figure 5. Abstraction levels for design and performance aspects of the exhaust use case

We propose to represent a multi-stage design process by introducing the concept of ***stage definition*** in the design knowledge formalization, such that designers can work with predefined stages at the data level. The exact scope, responsibilities and user guidance of the stages is still subject of investigation.

6. Conclusion

This paper presents a new methodology for evaluating at an early design stage whether it is worthwhile from cost and performance point of view to introduce a new product variant in an existing

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press

product family. The structured representation of the design knowledge is key to our approach. The formalization of the product family design knowledge is centred around a cost, performance and design structure. After introducing a link between the design definition and the management of designs at data level, early-phase cost-performance trade-offs can be set-up in order to determine the best design alternative for a new product variant. We illustrated our methodology on the design of a new exhaust cold end system and we gave a preview how the formalization can be extended taking a multi-stage design process into account.

In order to make our method more robust, we will add in a next step traceability information to our model. As a first step, we want to create a link between designs on different levels of abstraction. This starts by defining the workflow for refining a design across design stages. When going from a design defined at a higher level of abstraction to a lower abstraction level, the design data needs to be kept consistent so that a refinement of design choices does not invalidate earlier design decisions. In the case that an update of design choices or computational models cause conflicts with earlier estimated cost and performance values, one needs to be able to retrieve the cause for these mismatches throughout the different stages of the design process.

Acknowledgment

This research was supported by Flanders Make, the strategic research centre for the manufacturing industry, and by the Hermes Fund. We would also like to thank Bosal (https://www.bosal.com/en) for delivering a generic use case to develop, test and validate the methodology.

References

Asadi, N. , Jackson, M. and Fundin, A. (2016), "Drivers of Complexity in a Flexible Assembly System- A Case Study", *Procedia CIRP Vol. 41*, CIRP Society, Ischia, pp. 189-194. https://doi.org/10.1016/j.procir.2015.12.082

Benavides, B., Trinidad, P. and Ruiz-Cortés, A. (2005), "Automated Reasoning on Feature Models", In: Pastor O., Falcão e Cunha J. (Eds.), *Advanced Information Systems Engineering (Lecture Notes in Computer Science, vol. 3520)*, Springer Verlag, Berlin, Heidelberg, pp. 361-373. https://doi.org/10.1007/11431855_34

Bonev, M., Hvam, L., Clarkson, J. and Maier, A. (2015), "Formal computer-aided product family architecture design for mass customization", *Computers in Industry*, Vol. 74, pp. 58-70. https://doi.org/10.1016/j.compind.2015.07.006

Floridi, L. (2008), "The Method of Levels of Abstraction", *Minds and Machines*, Vol. 18 No. 3, pp. 303-329. https://doi.org/10.1007/s11023-008-9113-7

ISA (2010), ANSI/ISA-95.00.01-2010: Enterprise-Control System Integration - Part 1: Models and Terminology, International Society of Automation, North Carolina.

Jiao, J., T.W. and Siddique, Z. (2007), "Product family design and platform-based product development: a state-of-the-art review.", *J. Intell Manuf*, Vol. 18, pp. 5-29. https://doi.org/10.1007/s10845-007-0003-2

Robertson, D. and Ulrich, K. (1998), *Planning for product platforms*. [online] MIT *Sloan Management Review*. Available at https://sloanreview.mit.edu/article/planning-for-product-platforms/ (accessed 05.10.2021).

Simpson, T. W. (2004), "Product platform design and customization: Status and promise", *AI EDAM*, Vol. 18 No. 1, pp. 3-20. https://doi.org/10.1017/S0890060404040028

Thomke, S. (2001), *Enlighted Experimentation: The New Imperative for Innovation. [online] Harvard Business Review*. Available at: https://hbr.org/2001/02/enlightened-experimentation-the-new-imperative-for-innovation (accessed 04.10.2021).

Windheim, M. (2020), "State of the Art in Product Family Design and Evaluation", In: Alexander Gruen (Ed.), *Cooperative Decision-Making in Modular Product Family Design (Produktenentwicklung und Konstruktionstechnik Vol 17*), , Springer Berlin Heidelberg, pp. 33-78. https://doi.org/10.1007/978-3-662-60715-2_3

Zeng, F., Li, B., Zheng, P. and Xie, S.S.Q. (2014), "A modularized generic product model in support of product family modeling in One-of-a-Kind Production", *Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation*, IEEE, Tianjin, pp. 786-791. https://doi.org/10.1109/ICMA.2014.6885797

https://doi.org/10.1017/pds.2022.61 Published online by Cambridge University Press