INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.192

Integrating Model-Based Design of Mechatronic Systems with Domain-Specific Design Approaches

S. Husung ✉, C. Weber and A. Mahboob

Technische Universität Ilmenau, Germany

✉ stephan.husung@tu-ilmenau.de

Abstract

In addition to the known approaches for product development new or supplementary approaches have emerged. An important approach in this field is Systems Engineering (SE) and Model-Based Systems Engineering. Through these approaches, new procedures, level-focused description concepts and terms come into product development. However there are still some uncertainties as to how the known approaches of product development can be combined with the SE approaches. This paper aims to show how the known development approaches can be extended by and integrated with SE approaches.

Keywords: model-based systems engineering (MBSE), systems engineering (SE), design process, design models

1. Introduction

Product development is in a state of transformation since several years. Mechatronic or even cybertronic products, sometimes in combination with services (so called Product-Service Systems (PSS)), are usually necessary to meet the requirements of the stakeholders (Krause and Heyden, 2022). Thus, in addition to known, mostly domain-specific development approaches (models, procedures and methods), new or supplementary approaches are required to support the cross-domain development of mechatronic or even cybertronic products. An important approach in this field is Systems Engineering (SE) and, in connection with models, Model-Based Systems Engineering (MBSE). Through SE and MBSE new procedures and new description concepts for the product (see section 3) as well as new terms come into product development. However, both in academia and industry, there are still some uncertainties as to how the "traditional" approaches, among them quite elaborated methods for layout and detail design of products in the mechanical domain, can be combined with the SE approaches for the cross-domain development. This paper aims to show how the known development approaches can be extended by and integrated with SE approaches. The following three research questions are discussed in the paper:

- How do Systems Engineering approaches help extending existing methodologies and methods to deal with different mechatronic (cross-domain) and domain-specific system levels? (Answers, see section 3.1.)
- Which added value do semi-formal systems descriptions (like the ones provided by MBSE) offer at the mechatronic system level? (Answers, see section 3.2.)
- What information (requirements, decisions, development results) has to be exchanged between the different system levels, and how to do that? (Answers, see sections 3.1 and 4.)

For reasons of space, the considerations are mostly focussed on linking the mechatronic level to mechanical (sub-) issues. However, the same could be shown for the other domains involved in mechatronic systems (electric/electronic, information processing, hydraulics, optics, ...), which has also already been discussed much more intensively than the linking to the mechanical domain.

2. State of the art

Domain-specific models, methods and methodologies (in the sense of usage prescriptions of a consistent set of methods, guidelines and tools to produce solutions for a defined class of design tasks) have been around for a long time. Their aims were and are the support of designers in practice, provide a common base for teaching, and quite often enabling standardisation of products/components ("solution patterns", (Weber and Husung, 2016)). In the early days (starting in the 19th century), most approaches were dedicated to specific branches of industry or particular classes of products. Examples are models and methods dealing with Machine Elements, i.e. well-known and frequently used solution patterns in mechanical products. We find similar "pattern-based" approaches in other domains. These models and methods are continuously enhanced with new knowledge and transferred into contemporary tools (e.g. computer-based tools), and they are still useful and used today.
In the second half of the 20th century, more general approaches were developed: Not limited to particular branches or products and - more importantly - aimed at dealing with design tasks where the solution is not known from the beginning, thus focussing on invention and innovation (Design Theory and Methodology, DTM). These developments denied the previously prevailing view that designing is pure art (i.e. based on intuition which supposedly could not be taught and trained) and propagated that it can be systemised based on scientific findings and concepts, supported by appropriate models, methods and tools (maybe even automated).
The most widely spread approach to DTM goes back to early works at the Technische Universität Ilmenau (at that time: “Hochschule für Elektrotechnik”, “College of Electrical Engineering”) by Bischoff and Hansen (Hansen and Bischoff, 1953; Hansen, 1955) and was later picked up, extended and specialised by many other authors in Germany, Scandinavia, Switzerland and the Netherlands. The literature in this field is by far too numerous to be referenced here, important authors were Rodenacker, Roth, Koller, Andreasen, Eekels and Roozenburg. Because of translations into the English language, best known in the international context are the VDI guideline 2221 (VDI, 1987), the books of Pahl & Beitz (Pahl et al., 2007; Pahl et al., 1996) and of Hubka & Eder (Hubka and Eder, 1996). In (Blessing, 1996), a good overview is given over history, impact and also some open problems of the “European” (in this article designated “German”) approaches and models.
The core of these approaches is a phase model of the development/design process that proceeds from the clarification of the task to functional and principle considerations (conceptual phase) to embodiment (i.e. layout and detail) design issues. This model is often misunderstood as a “linear” one, while in reality several iteration loops between the phases have to be expected.
In the last 20 years, a number of authors have introduced considerable modernisations, particularly addressing computer support, distributed design processes and management issues. Prominent examples come from Ehrlenspiel (Ehrlenspiel and Meerkamm, 2017; Ehrlenspiel, 2009), Lindemann (Lindemann, 2009), Weber (Weber, 2014, 2005), Albers (Albers et al., 2005; Albers and Wintergerst, 2014). An overview over contemporary approaches is given in (Chakrabarti and Blessing, 2014).
All these approaches have their origin in the mechanical domain; as a result - even if they are decoupled from particular branches or products - they still remain quite domain-specific.
A first attempt to integrate several domains, i.e. to "go mechatronic", was made by VDI guideline 2006 (VDI, 2004b). VDI 2206 introduces the mechatronic system level above domain-specific levels (mechanical, electric/electronic, information processing, hydraulics, optics, ...). Based on the concept decisions at the mechatronic system level, domain-specific procedures are conducted for embodiment and detail design. In recent years, VDI 2221 has also been further developed into the direction of mechatronic system development (VDI, 2019). In this guideline, the development process is described as loops, independent of system levels, on the basis of Systems Theory (Ropohl, 1975).
Systems Engineering (SE) has its origins in the 1940ies and 1950ies (in a way in parallel to the origins of Design Theory and Methodology) as an approach to model and handle complex and increasingly multidisciplinary engineering systems and their behaviour. Today, SE is a well-known concept that is described in the ISO/IEC/IEEE 15288 and the INCOSE Systems Engineering Handbook (Walden et al.,

https://doi.org/10.1017/pds.2022.192 Published online by Cambridge University Press

2015). SE includes processes and related methods, with the aim “to enable the realisation of successful systems” (Walden et al., 2015). A wide spectrum of methods is described in (Haberfellner et al., 2019). The enhancement of SE with models is called Model-Based Systems Engineering (MBSE). The most widely used modelling language in MBSE is the Systems Modeling Language (SysML) (Friedenthal et al., 2015). SysML is a semi-formal graphical modelling language to describe products on the mechatronic system level(s) as well as the system context. The SysML model can additionally facilitate analysis, verification and validation activities on the design (Husung et al., 2021; Hick et al., 2019).
Seen in relation to the earlier, domain-specific approaches - especially the ones from mechanical engineering that were outlined above - MBSE provides a new way of systematically capturing requirements and functions, at the same time promising to link these to more concrete and detailed representations of the product being developed and to subsequent activities of the development process. This will be an important starting point for further considerations in this paper.

3. Description of products and methods on different system levels

For the development of mechatronic or cybertronic products, their description using Systems Theory is recommended (Ropohl, 1975). Based on this, products are described as models of so-called mechatronic systems (systems are themselves models of the product, possibly in combination with processes). The introduction of systems thinking implies a decomposition of the mechatronic system into sub-systems (Pohl, 2012). The decomposition is, in principle, arbitrary and different decompositions can be chosen for a product depending on the objectives (Ariyo et al., 2008). The decomposition is often carried out through organisational structures or by focussing on specific properties (such as functions) (Browning, 2001). This may result in sub-systems for which specific departments or suppliers are responsible. In any case, the decomposition of a system into sub-systems results in hierarchically structured levels, whereby for some products several levels with mechatronic sub-systems are possible (see also section 5). Eventually, from a certain level down, the mechatronic (sub-) system is split up into domain-specific sub-systems (including mechanical sub-systems). The domain-specific sub-systems can also be further decomposed, but this is not addressed in this paper because knowledge on this exists inside the domains. Figure 1 shows a simple example of three levels, of which two are mechatronic (sub-) systems. It may be noted that further up in the hierarchy, the more the models are related to functions, further down, the more they include principle and embodiment information.

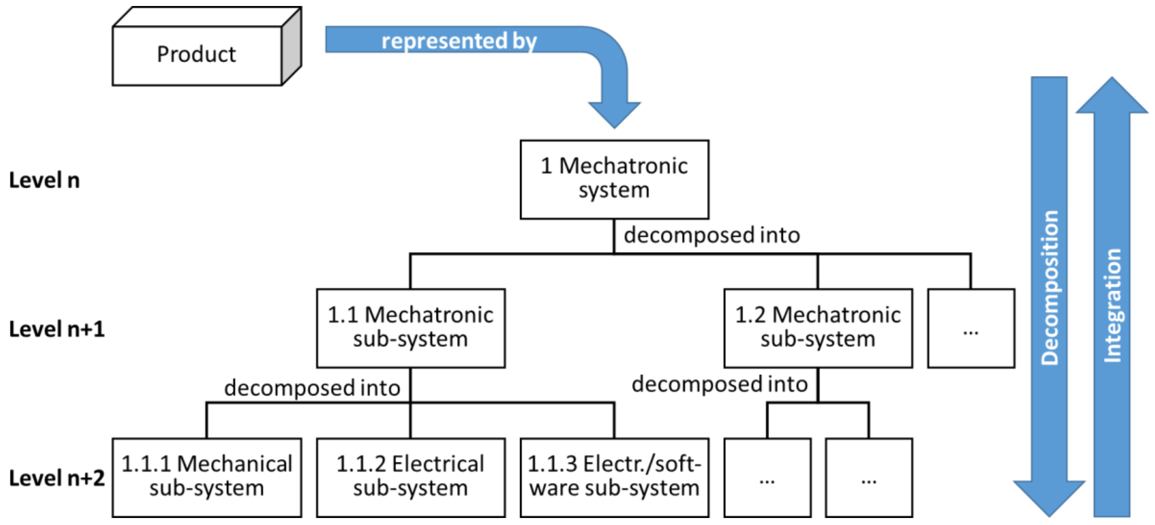

Figure 1. Decomposition of a mechatronic system (example) - Interactions between the sub-systems are not displayed (according to (Walden et al., 2015))

3.1. Procedure at the different system levels

Approaches for the development of mechanical sub-systems are well known (see section 2), e.g. the procedure according to VDI 2221 (VDI, 2019) or on the layout and detail level according to VDI 2223

(VDI, 2004a). The same can be stated for the other relevant domains. At the end of the development activities of the domain-specific sub-system, a detailed description of them must be available for the further product life phases. However, the domain-specific approaches and models often reach their limits when connected to the mechatronic system level(s) because, among other things, the explicit differentiation between system levels is not made, the function description is not sufficient due to the numerous logic decisions (e.g. decisions which functions are active when) and the solution principles are often based on domain-specific physical effects (Koller, 1994; Roth, 2000). Against this background, it is often useful to extend the known approaches at the mechatronic system level(s) with systems engineering approaches. The resulting procedure is described in more detail below.

For the development of mechatronic (sub-) systems, approaches from Systems Engineering have become established (Huth and Vietor, 2020; Gausemeier et al., 2015). These approaches are specifically domain-spanning and are initially developed in a "top-down" way of working (Pohl, 2012). This implies that the detailed descriptions of the sub-systems, as specified later, must ultimately be brought together in "bottom-up" fashion (Hick et al., 2019). So, at the beginning of the development, the mechatronic view of the system is addressed with the goal of decomposing the required functions into sub-functions based on requirements (Srinivasan et al., 2012; Lamm and Weilkiens, 2014). Other required properties and boundary conditions may be added. The relations between the sub-functions have to be specified as functional chains (van Eck et al., 2008; Stone and Wood, 2000). The functional chains describe the temporal and logical dependencies between the sub-functions as well as the concrete flow variables (material, energy, information) that are exchanged between the sub-functions. The temporal and logical dependencies of the sub-functions are a characteristic feature at the mechatronic system level (Husung et al., 2022). Temporal dependencies are mostly series or parallel connections of functions. These are also very common in the domains. Logical dependencies can be significantly more complex. On the mechatronic system level, it often occurs that certain functions are only active in specific states (e.g. an actuation only takes place when the system is in the operating state (see section 5)) (Yildirim et al., 2017; Morkevicius et al., 2017). In addition, there are numerous decision-making elements between the sub-functions, e.g. determining which sub-function shall be currently active, as well as loops that control the activation of (sub-) functions depending on specific conditions (Morkevicius et al., 2017).

The sub-functions are passed "downwards" as required functions of the sub-systems. For this purpose, the (sub-) system is broken down into appropriate further sub-systems (Morkevicius et al., 2017). The sub-functions are assigned to these sub-systems. This decomposition is not to be understood as a waterfall: Depending on the possibilities of implementation incl. verification in the sub-systems, the decomposition into the sub-functions and assignment to the sub-systems must be adapted. This iterative procedure between the system levels is explained in detail in several sources (Pohl, 2012; Morkevicius et al., 2017).

When decomposing a system into sub-systems, the term "logical element" is often used in the field of Systems Engineering, as the concrete realisation of the sub-systems is often not known at the beginning of the development (Pohl, 2012). A logical element is an abstract representation of a sub-system to realise specific functions that will be concretised later (Estefan, 2008). If already existing components (real parts of the later product) are used, the sub-systems can already be concrete and corresponding descriptions can be inserted into the description hierarchy (Anacker et al., 2020).

In addition to the required functions, further required properties must also be passed on to the sub-systems. These required properties include, among other things, criteria like durability, reliability, etc. The required properties, together with design constraints, result in the requirements for each sub-system. The requirements can be passed on to the sub-systems at the next-lower level via model elements (e.g. model elements for the required functions) (Pohl, 2012) or (textual) requirements (see also figure 4). In this case, the requirements are distributed to N sub-systems (Böhm, 2021).

After the sub-system on the lower level has been developed (or is already developed because an existing solution is chosen) its As-is description is compared with the requirements and, if necessary, changes are made to the requirements or descriptions. Hereby, an important task at the mechatronic system levels is to combine the As-is descriptions of all related sub-systems (integration within the system descriptions at the appropriate place) and to analyse their interaction (keyword: emergent behaviour).

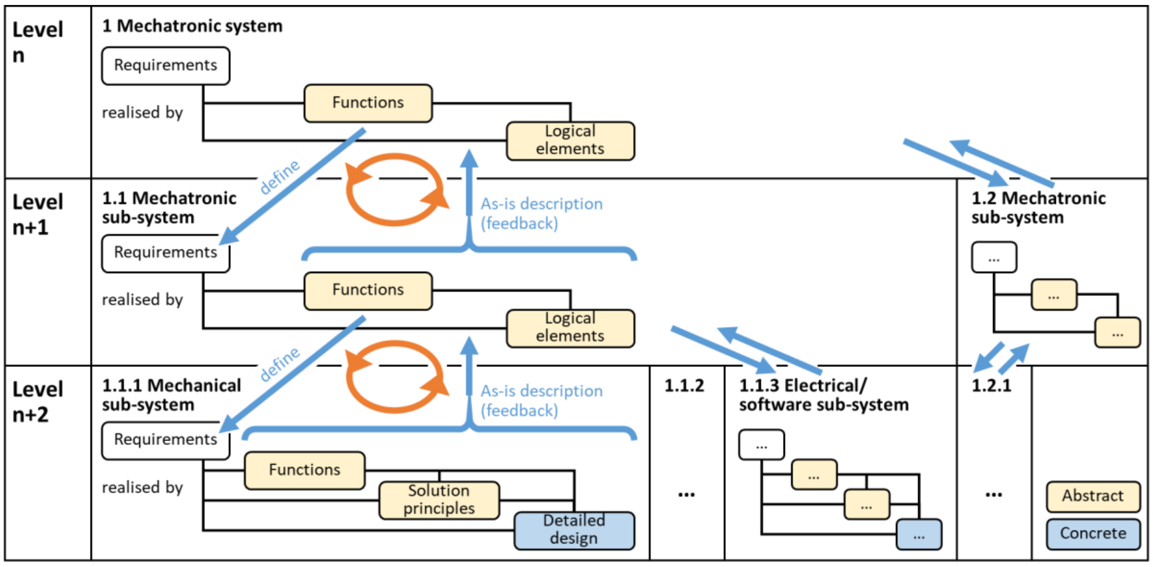

Figure 2. Interactions during development at the different levels of the system (expanded on the basis of (Pohl, 2012), concrete models on mechatronic levels are not visualised)

The descriptions of the systems and sub-systems on the mechatronic levels at the beginning of the development are usually mainly related to functions, thus still quite abstract. Therefore, many decisions about the realisation of required-properties (term according to the CPM model (Weber, 2005)) that go beyond the function (e.g. with regard to service life, reliability, suitability for manufacturing/assembly, etc.) cannot yet be made conclusively at this stage. For this, one has to wait for the results of the domain developments. It is therefore important that the As-is-properties of the sub-systems are passed back to the higher levels of the system in such a way that they can be compared with the required properties (Weber and Husung, 2016).

Table 1. Differences in the descriptions of the (sub-) systems and related procedures

	Mechatronic system level	**Mechanical (and other) system level(s)**
Descriptions of the (sub-) systems	The functions and logical elements are mostly abstract (i.e. not yet verified by concrete solutions). But the temporal and logical dependencies of the functions are usually more complex (i.e. implying, among other things, the assignment of functions to states as well as definitions of which functions are active and when).	The functions and solution principles (counterpart to the logical elements) are usually more concrete (among other things because they are specific to the domain).
Methods/ procedures	The first selection of the sub-systems is made according to their implementation status with the focus on the realisation of the necessary required functions and further properties. Further elaboration of the sub-systems can only be achieved by iterative collaboration with the organisations/people who are responsible for the sub-systems.	When selecting the solution principles, several required properties are already considered on the basis of experience (e.g. through the use of solution patterns).
	The main focus of the activities is the decomposition of the functions and the assignment of the sub-functions to the sub-systems, the necessary orchestration of the sub-systems and verification after integration.	The main focus of the activities is the selection of physical effects as a basis for solution principles as well as the elaboration of detailed design with regard to the design, considering numerous other (not directly functional) required properties - e.g. service life, reliability, suitability for manufacturing/assembly - as well as verification.

https://doi.org/10.1017/pds.2022.192 Published online by Cambridge University Press

The explanations show that different approaches at the mechatronic and mechanical (and other) system levels are all useful and complement each other. Significant differences in the procedure and in the descriptions of the (sub-) systems (always seen in relation to one another) on mechatronic and mechanical system level are shown in table 1.
When moving from a level with a mechatronic (sub-) system to a mechanical (sub-) system, i.e. moving top-down, at least the following information is required:

- Required functions that the mechanical sub-system has to take over, with the relevant required input and output flows (quality and quantity) (Albers and Wintergerst, 2014) as well as detailed information about the properties of the function (e.g. accuracy, reproducibility, etc.),
- Boundary conditions for the sub-system, e.g. with regard to interfaces,
- Requirements for other properties that do not result from functional decomposition (e.g. service life, manufacturing, etc.).

3.2. Models at the different system levels

The description of the elements in the mechatronic (sub-) systems differs slightly from the descriptions on the domain-specific (lower) levels due to different objectives. The domain descriptions, especially on the level of layout and detail design, require unambiguous formal models (e.g. CAD models with a mathematically unambiguous description of the geometry, but also CAE/CAO models for simulation/optimisation). On the mechatronic system levels, semi-formal descriptions are often used at the beginning of development (Hick et al., 2019), e.g. using the standardised modelling language SysML, among others due to

- the more abstract nature of the system elements and their relations (incl. the temporal and logical dependencies of the functions) as well as
- the need for simple adaptability in the event of changes during iteration loops.

The semi-formal models can (strictly speaking: have to) be supplemented or replaced by formal models as development progresses and the maturity of the sub-systems increases (Hick et al., 2019). This is, among other things, important in order to plan and perform final verifications based on these models.

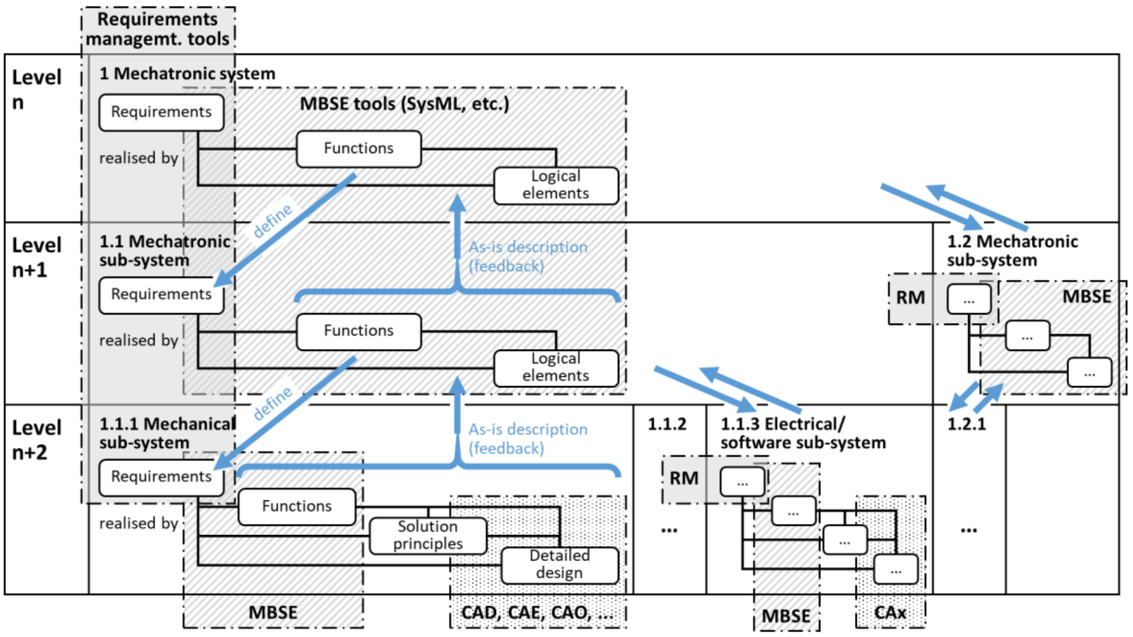

Figure 3. Description of the elements in the (sub-) systems and used tools

The standardised modelling language SysML finds its application mainly in modelling the logical system elements as well as the functions (see figure 3). Looking further "down" the hierarchy of models, SysML is not really suitable to describe solution principles, since they contain, besides physical effects,

geometric information on the arrangement and relations of the solution principle elements; SysML currently does not include an efficient possibility for capturing such information. Research on the model-based description of solution principles (see gap between MBSE and CAx in figure 3) is presented e.g. in (Meussen, 2021). Approaches to extend the SysML language, especially to represent geometric information, have been discussed in projects such as FAS4M (Grundel et al., 2014) or SysML4FMArch (Jacobs et al., 2022).
In some cases, SysML is also used to represent some additional requirements directly; however, this does not look very promising for some classes of requirements (e.g. for manufacturing) because, among other things, the versioning of the requirements cannot be ensured. However, at present it seems to be more adequate to use requirements models in special requirements' management tools; by the way, their application is independent of the level of the system.

4. Linking the elements

In order to trace the relevant information in the model elements (requirements, required and As-is-functions, interface descriptions, etc.), it is often useful to define links between the model elements. The links in the domains of the electronics/software sub-systems are usually very complex, as numerous sub-functions with specific temporal and logical dependencies are implemented here. The links in the mechanical domain focus primarily on functions that are implemented by individual components or assemblies, as well as interface specifications including the expected flows (e.g. torques or forces and velocities) (Albers and Wintergerst, 2014). Further links to specific expected properties are possible. If these are not explicitly present in the models at the mechatronic system level(s), it is still possible to connect them with the requirements. This type of linking will be called indirect linking here, as it is not done directly via model elements, but via requirements (see figure 4). Such linking helps achieve traceability in the overall development process.

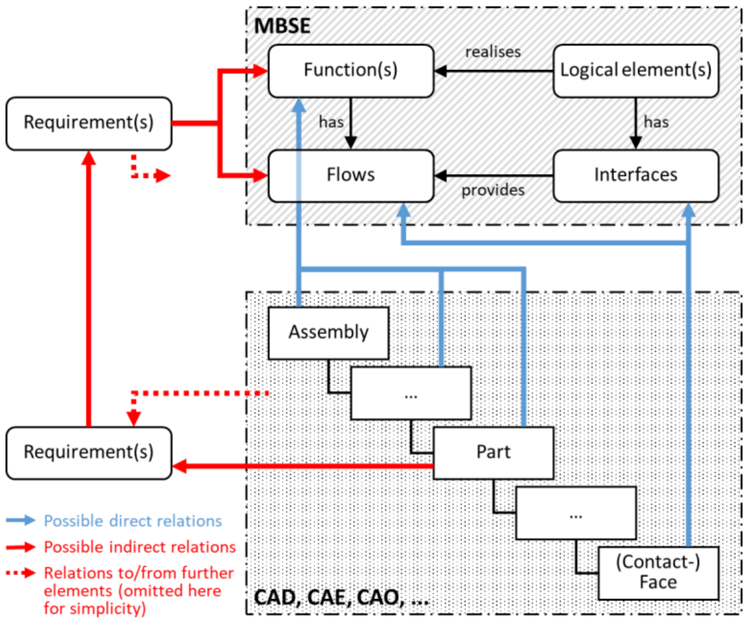

Figure 4. Possible traces/links between the model elements

5. Example

In order to show the interaction between the different system levels (especially between the levels representing the mechatronic and the mechanical (sub-) system(s)), a product from precision technology (see SFB 622) will be used as a use case (highly simplified). At the highest system level, based on the determined use case "Precision positioning of measuring object" it has to be identified what the product must offer as function(s) to the outside world. The overall function is the actual execution of the positioning, which has to become active when the system is in the corresponding state (e.g. after

https://doi.org/10.1017/pds.2022.192 Published online by Cambridge University Press

successful initialisation and when a start signal is actuated). On the next level, the overall function is divided into appropriate sub-functions for movement execution, measurement and control. The sub-function for movement execution is assigned to the corresponding sub-system (here *"Movement Sub-System"*). The sub-system is itself also a mechatronic system. Therefore, on the next sub-system level, there is a further decomposition to the direct drive and - in this case - a precision guidance that is realised purely mechanically. During the development of the product, it is important that the interactions at the interfaces between the respective sub-systems are coordinated. Top-down, the requirements for positioning lead to decisions in the control sub-system and measuring sub-system. These decisions must be coordinated with the direct drive. The effects of the direct drive on the precision guidance must be considered as a load spectrum. This requires coordination between the system levels and orchestration of the sub-systems.

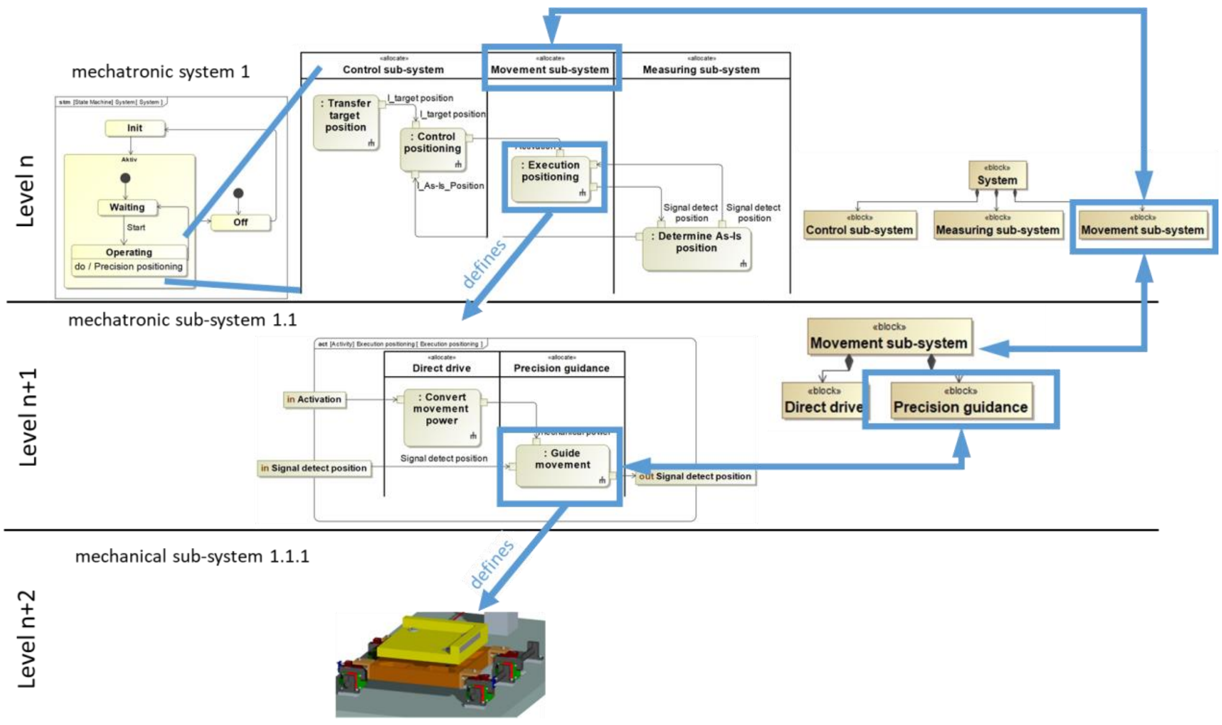

Figure 5. Simplified example of decomposition across levels for use case "Precision positioning of measuring object" (figure inspired by (Morkevicius et al., 2017))

6. Conclusion and next steps

The combination of "traditional" development approaches and Systems Engineering approaches is becoming increasingly important for overcoming the challenges in product development. The approaches have many parallels, but at the same time they have their own specific meaning and justification on different levels of the system. This paper discusses the reasons for the complementary approaches and describes basic procedures and descriptions used.

Especially in the early development phases, abstract (i.e. mainly function-related) descriptions of the product are necessary at the mechatronic system level, as the federation of the sub-systems and the necessary changes can be carried out in a target-oriented way. The interaction of the approaches must be clearly understood so that the mechatronic and domain-specific levels can work together effectively and efficiently. Understanding the interactions between system levels and system elements is also an important basis for end-to-end model and tool support between system levels.

For future research activities, it should be discussed whether the existing models, methods and methodologies that are at present existing separately - as is even implicitly defined in VDI guideline 2006 (VDI, 2004b) - should be merged into one domain-spanning approach (also taking into account the other domains) or whether the separation of the different approaches with defined interfaces and interactions

is more purposeful. This, however, would lead to massive consequences in science, practice (including organisational issues) and teaching.
Further activities of the authors and others deal with the target-oriented development of meta-models for the continuous model description. Above all, the description of solution patterns in the domains and at the lower levels of mechatronic systems is an important basis for the further (re-)use of existing knowledge.

References

Albers, A., Burkhardt, N., Meboldt, M. and Saak, M. (2005) "SPALTEN Problem Solving Methodology in the Product Development", *15th International Conference on Engineering Design.* Melbourne, Australia, 15.-18.08.2005, pp. 553–554. DOI: https://doi.org/10.5445/IR/1000007075.

Albers, A. and Wintergerst, E. (2014) "The Contact and Channel Approach (C&C2-A): Relating a System's Physical Structure to Its Functionality", in: Chakrabarti, A. and Blessing, L. T. M. (eds) *An anthology of theories and models of design: Philosophy, approaches and empirical explorations*, London, Springer, pp. 151–171.

Anacker, H., Dumitrescu, R., Kharatyan, A. and Lipsmeier, A. (2020) "Pattern Based Systems Engineering – Application of Solution Patterns in the Design of Intelligent Technical Systems", *16th International Design Conference (DESIGN 2020)*, pp. 1195–1204. DOI: https://doi.org/10.1017/dsd.2020.107.

Ariyo, O. O., Eckert, C. M. and Clarkson, P. J. (2008) "Hierarchical decompositions for complex product representation", *10th International Design Conference*, pp. 737–744.

Blessing, L. (1996) "Comparison of design models proposed in prescriptive literature", in: *The role of design in the shaping technology*, 5th edn, Social Sciences Series, pp. 187–212.

Böhm, W. (2021) Model-Based Engineering of Collaborative Embedded Systems: Extensions of the SPES Methodology, Springer. DOI: https://doi.org/10.1007/978-3-030-62136-0.

Browning, T. R. (2001) "Applying the design structure matrix to system decomposition and integration problems: a review and new directions", *IEEE Transactions on Engineering Management*, vol. 48, no. 3, pp. 292–306.

Chakrabarti, A. and Blessing, L. T. M., eds. (2014) An anthology of theories and models of design: Philosophy, approaches and empirical explorations, London, Springer.

Ehrlenspiel, K. (2009) Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit, 4th edn, München, Hanser Verlag. 9783446420137.

Ehrlenspiel, K. and Meerkamm, H. (2017) *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*, 6th edn, Carl Hanser Verlag. DOI: https://doi.org/10.3139/9783446449084.

Estefan, J. A. (2008) Survey of model-based systems engineering (mbse) methodologies, INCOSE-TD-2007-003-02.

Friedenthal, S., Moore, A. and Steiner, R. (2015) *A practical guide to SysML: The systems modeling language*, 3rd edn, The MK/OMG Press. Burlington.

Gausemeier, J., Dumitrescu, R., Steffen, D., Czaja, A., Wiederkehr, O. and Tschirner, C. (2015) *Systems engineering in industrial practice* [Online], Paderborn, Germany.

Grundel, M., Abulawi, J., Moeser, G., Weilkiens, T., Scheithauer, A., Kleiner, S., Kramer, C., Neubert, M., Kümpel, S. and Albers, A. (2014) "FAS4M – No more: "Please mind the gap!"", *Tag des Systems Engineering.* Bremen, pp. 63–74. DOI: https://doi.org/10.3139/9783446443761.007.

Haberfellner, R., Weck, O. L. de and Fricke, E. (2019) *Systems engineering: Fundamentals and applications*, Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-13431-0.

Hansen, F. (1955) Konstruktionssystematik – Grundlagen für eine allgemeine Konstruktionslehre (Systematic Design – Fundamentals of a General Model of Designing), VEB-Verlag Technik.

Hansen, F. and Bischoff, W. (1953) *Rationelles Konstruieren (Efficient Designing)*, 5th edn, VEB Verlag Technik.

Hick, H., Bajzek, M. and Faustmann, C. (2019) "Definition of a system model for model-based development", *SN Applied Sciences*, vol. 1, no. 9.

Hubka, V. and Eder, W. E. (1996) *Design science: Introduction to the needs, scope and organization of engineering design knowledge*, 2nd edn, London, Springer. DOI: https://doi.org/10.1007/978-1-4471-3091-8.

Husung, S., Weber, C. and Mahboob, A. (2022) "Model-Based Systems Engineering – A New Way for Function-Driven Product Development", in: Krause, D. and Heyden, E. (eds) *Design Methodology for Future Products – data driven, agile and flexible*, Springer International Publishing.

Husung, S., Weber, C., Mahboob, A. and Kleiner, S. (2021) "Using Model-Based Systems Engineering for need-based and consistent support of the design process", *23rd International Conference on Engineering Design (ICED21)*. DOI: https://doi.org/10.1017/pds.2021.598.

Huth, T. and Vietor, T. (2020) "Systems Engineering in der Produktentwicklung: Verständnis, Theorie und Praxis aus ingenieurswissenschaftlicher Sicht", *Verständnis, Theorie und Praxis aus ingenieurswissenschaftlicher Sicht*, no. 51, pp. 125–130.

Jacobs, G., Konrad, C., Berroth, J., Zerwas, T., Höpfner, G. and Spütz, K. (2022) "Function-oriented model-based product development", in: Krause, D. and Heyden, E. (eds) *Design Methodology for Future Products – data driven, agile and flexible*, Springer International Publishing.

Koller, R. (1994) Konstruktionslehre für den Maschinenbau: Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen, 3rd edn, Berlin, Heidelberg, Springer. DOI: https://doi.org/10.1007/978-3-662-08165-5.

Krause, D. and Heyden, E., eds. (2022) *Design Methodology for Future Products – data driven, agile and flexible*, Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-78368-6.

Lamm, J. G. and Weilkiens, T. (2014) "Method for Deriving Functional Architectures from Use Cases", *Systems Engineering*, vol. 17, no. 2, pp. 225–236.

Lindemann, U. (2009) Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden, 3rd edn, Berlin, Springer. 978-3-642-01423-9.

Meussen, B. (2021) "On the use of model based systems engineering and CAD for the design of physical products", *23rd International Conference on Engineering Design (ICED21)*. DOI: https://doi.org/10.1017/pds.2021.493.

Morkevicius, A., Aleksandraviciene, A., Mazeika, D., Bisikirskiene, L. and Strolia, Z. (2017) "MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems", *INCOSE International Symposium*. DOI: https://doi.org/10.1002/j.2334-5837.2017.00350.x.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007) *Engineering design: A systematic approach*, 3rd edn, London, Springer. DOI: https://doi.org/10.1007/978-1-84628-319-2.

Pahl, G., Beitz, W. and Wallace, K. (1996) *Engineering design: A systematic approach*, 2nd edn, London, Berlin, Heidelberg, Springer. DOI: https://doi.org/10.1007/978-1-4471-3581-4.

Pohl, K. (2012) *Model-based engineering of embedded systems: The SPES 2020 methodology*, Heidelberg, Springer. DOI: https://doi.org/10.1007/978-3-642-34614-9.

Ropohl, G. (1975) *Systemtechnik: Grundlagen und Anwendung*, München, Hanser. DOI: https://doi.org/10.1002/cite.330471315.

Roth, K. (2000) *Konstruieren mit Konstruktionskatalogen*, Berlin, Heidelberg, Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-642-17466-7.

Srinivasan, V., Chakrabarti, A. and Lindemann, U. (2012) "A Framework for Describing Functions in Design", *12th International Design Conference (DESIGN 2012).* Dubrovnik, Croatia, pp. 1111–1122.

Stone, R. B. and Wood, K. L. (2000) "Development of a Functional Basis for Design", *Journal of Mechanical Design*, vol. 122, no. 4, pp. 359–370.

van Eck, D., McAdams, D. A. and Vermaas, P. E. (2008) "Functional Decomposition in Engineering: A Survey", *19th International Conference on Design Theory and Methodology*, 4.-7.9.2007. New York, NY, ASME. DOI: https://doi.org/10.1115/DETC2007-34232.

VDI (1987) VDI 2221:1987: Systematic Approach to the Design of Technical Systems and Products, Düsseldorf.

VDI (2004a) VD 2223:2004: Methodisches Entwerfen technischer Produkte / Systematic embodiment design of technical products, Düsseldorf.

VDI (2004b) VDI 2206:2004: Entwicklungsmethodik für mechatronische Systeme / Design methodology for mechatronic systems, Düsseldorf.

VDI (2019) VDI 2221:2019: Entwicklung technischer Produkte und Systeme/ Design of technical products and systems, Düsseldorf.

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D. and Shortell, T. M. (2015) *Systems engineering handbook: A guide for system life cycle processes and activities*, 4th edn, Hoboken, NJ, Wiley. 978-1118999400.

Weber, C. (2005) "CPM/PDD - An Extended Theoretical Approach to Modelling Products and Product Development Processes", *2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes*, 07.-08.07.2005, Fraunhofer-IRB-Verlag, pp. 159–179.

Weber, C. (2014) "Modelling Products and Product Development Based on Characteristics and Properties", in: Chakrabarti, A. and Blessing, L. T. M. (eds) *An anthology of theories and models of design: Philosophy, approaches and empirical explorations*, London, Springer, pp. 327–352.

Weber, C. and Husung, S. (2016) "Solution patterns - their role in innovation, practice and education", *14th International Design Conference (DESIGN 2016)*, Volume: Design Theory and Research Methods. Cavtat, Dubrovnik, Croatia, pp. 99–108.

Yildirim, U., Campean, F. and Williams, H. (2017) "Function modeling using the system state flow diagram", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 31, no. 4, pp. 413–435.