



ABSTRACT PHYSICS REPRESENTATION OF A BALANCED TWO-WHEEL SCOOTER IN GRAPH-BASED DESIGN LANGUAGES

M. Ramsaier¹, R. Stetter^{1,✉}, M. Till¹ and S. Rudolph²

¹ University of Applied Sciences Ravensburg-Weingarten, Germany, ² University of Stuttgart, Germany

✉ ralf.stetter@rwu.de

Abstract

This paper presents a novel approach to include a holistic description of abstract physics in a digital engineering framework. Physical phenomena realize the numerous functions of technical systems and are an important link between rather abstract product functions and the concrete product geometry and material. Until now, a possibility to integrate the analysis and synthesis on this level of abstraction into a holistic engineering frameworks is not existing. The novel approach employs graph-based design languages using UML for this endeavour; the product example is a two-wheel scooter.

Keywords: design methods, design models, design process, physical effects, graph-based design languages

1. Introduction

In systematic design and the product development methodology, a distinction of product models based on their level of abstraction is generally agreed upon (Ehrlenspiel and Meerkamm, 2013; Ponn and Lindemann, 2011; Lindemann, 2009; Pahl et al., 2007). On the most abstract level are the requirements a technical system has to fulfil, followed by the functional level. The most concrete level are the actual geometry and material of the technical system. In between this level and the function level, an intermediate level can be found, which consists of physical phenomena or effects which realize the functions; the network of this physical effects leads to the physical behaviour of the technical system. In the last years both the requirements and the functions were integrated in a digital engineering framework (Holder et al., 2017; Ramsaier et al., 2017; Elwert et al., 2019). The most concrete level “geometry and material” forms the centre of a digital engineering framework, as this kind of concrete information can be stored digitally in a straight-forward manner. Until now, an integration of the intermediate level “abstract physics”, which includes physical effects, networks and effects and the resulting physical behaviour, was not realised. It is the central objective of this paper to propose an approach to achieve this integration. The integration is based on the application of graph-based languages using UML, because this approach was found to offer certain advantages also on the levels of requirements (Holder et al., 2017) and functions (Ramsaier et al., 2017; Elwert et al., 2019). The approach is demonstrated using the example of a balanced two-wheel scooter (compare Segway). Two different solution principles are described on the level of abstract physics and are integrated in an existing engineering framework (Wünsch et al., 2018). Additionally, support functionalities for choosing optimum physical effects and synthesizing effect networks were integrated into the logical

structure of the engineering framework. The paper is structured as follows: the next two sections provide a background - firstly concerning abstract physics and secondly graph-based design languages. Section 4 describes the product development problems and solutions of the two-wheel scooter. Two variants of the resulting abstract physics are integrated in an engineering framework - this is described in Section 5. Section 6 concludes the paper and delivers an outlook on future research activities.

2. Analysis and synthesis of abstract physics

For the analysis and synthesis of abstract physics, three perspectives can be found in literature and can be clearly distinguished: a phenomenon oriented perspective, a behaviour oriented perspective and an interface oriented perspective.

2.1. Phenomenon oriented abstract physics

In product development methodology, a consideration of abstract physics in form of physical effects (elementary demarcated physical phenomena) and the resulting effects chains is proposed. The analysis of physical effects is recommended for fostering a better understanding of the technical system and for developing innovative solutions, which use different physical principles (Ehrlenspiel and Meerkamm, 2013). It is essential for engineers to be familiar with the physical cause-and-effect chains and to know the applications of physical energies and effects. Additionally, the solution search with physical effects can dissolve mental blocks.

An in-depth analysis of innovative break-through products even resulted in the insight that such products in most cases rely on different physical principles. Wulf (2001) emphasizes that a discursive, hierarchical search for solution which includes the levels function and physical structure will probably lead to a “good” design. Nearly a decade ago, the analysis of physical effects was expanded in order to include uncertainties in form of disturbances (Mathias et al., 2011). In the area of computational design synthesis (CDS) an inclusion of phenomenon oriented abstract physics in the form of physical effects was realized by Helms (2013). In recent years it became obvious that an analysis and understanding of physical effects can be a decisive factor for principal investigations, which are empirical analyses in the early phases of a product development process which intended to quantitatively describe physical phenomena (Lang, 2016). It is a characteristic of such principal investigations that real processes are reduced to isolated physical principles or effects. A synthesis process including physical effects is also described for the realization of function integration in technical systems (Wagner, 2018).

2.2. Behaviour oriented abstract physics

For an analysis and synthesis of abstract physics also the middle level of the Function-Behaviour-Structure (FBS) ontology (Gero and Kannengiesser, 2014) can be applied. The FBS ontology is a design ontology that describes designed artefacts. It consists of three fundamental constructs – Function (F), Behaviour (B) and Structure (S). The construct “function” describes the objective of the artefact (“what the artefact is for”). The construct “behaviour” stands for the artefact’s attributes which can be derived from its structure (“what the artefact does”). It is important to note that the definition of this construct in literature does not include notions such as “operation”, “process” or “physical effect”; the explanations and examples indicate clearly that this concepts would also be appropriate to describe the phenomena on this level. The last construct “structure” represents the components of a technical system and their relationships (“what the artefact consists of”) and corresponds to the concrete design of the technical system.

Generally, the respective level of abstraction can be understood as a level describing the physical behaviour of technical systems. In today’s engineering practice, powerful possibilities for behaviour modelling are existing, if concrete geometrical and physical parameters are available, such as Modelica or MATLAB/Simulink. Several commercial software products allow a behavioural description of technical systems which is based on Modelica (e.g. Dymola, LMS Imagine.Lab AMESim, MapleSim, Wolfram SystemModeler, Modelon Inside, SimulationX). These tools allow the

development of complex simulation models (compare e.g. [Fritzson, 2014](#)). However, these systems allow no holistic link to the requirements and functions of a technical system and require concrete geometrical and physical parameters. Several approaches for the holistic connection of requirements, functions, structural parameters and performance evaluation can be found in the area of CDS, e.g. including bond-graph based simulation models for performance evaluations on the behaviour level ([Munzer and Shea, 2015](#)). A solution for the integration between graph-based languages and behaviour modelling in form of continuous mechanics is proposed by [Vogel \(2019\)](#); the main emphasis is on concrete geometry and advanced analyses of this geometry.

It can be concluded that a behaviour oriented perspective on abstract physics in many cases allows the application of mathematical models for simulation and optimization. This perspective can also be connected to the early approach towards describing abstract physics by [Atkin \(1965\)](#).

2.3. Interface oriented abstract physics

On a similar level of abstraction, the Contact and Connector Approach (C&C²-A) is proposed for analysing the interfaces and physical structures within technical systems ([Albers and Wintergerst, 2014](#); [Gładysz et al., 2017](#)). In this approach an effect network is described which stores, transforms and exchanges inputs and outputs, e.g. energy, material and information flows, but no concrete physical effects are included. In the field of cyber-physical systems development, research addressing the management of heterogeneous information also includes aspects of abstract physics ([Guerineau et al., 2017](#); [Zheng et al., 2017](#)).

For the even more abstract level “function”, a comprehensive framework - the integrated function modelling framework (IFM) - was developed in recent years and meanwhile was successfully implemented in graph-based design languages (compare [Eisenbart et al., 2016](#); [Ramsaier et al., 2017](#); [Elwert et al., 2019](#)). This framework already considers “effects” which can be applied in order to realise functions. The IFM even includes a so-called effect view, which main emphasis is also on the interfaces. In the IFM, effects are defined as a representation of physiochemical effects or principles that are required or have to be provided, respectively, in order to enable or support the execution of transformation and/or interaction processes ([Eisenbart et al., 2016](#)). The contents of the effect view (physiochemical effects; effects related to different processes) were considered useful by more than fifty percent of the participants in a large scale study investigating the industrial applicability of the IFM framework ([Eisenbart et al., 2016](#)). The research in this paper will build upon this initial model of a physical structure.

2.4. Combined abstract physics

Form the literature listed above, it can be concluded that an increasing scientific interest can be observed to analyse, reflect, understand and support the analysis and synthesis of the physical effect chains and networks which may realize the required functions in the domain of physics and physical behaviour. It is also obvious that more than one perspective is required in order to allow a holistic model of abstract physics. Figure 1 combines the perspectives listed above, their elements and their central advantages.

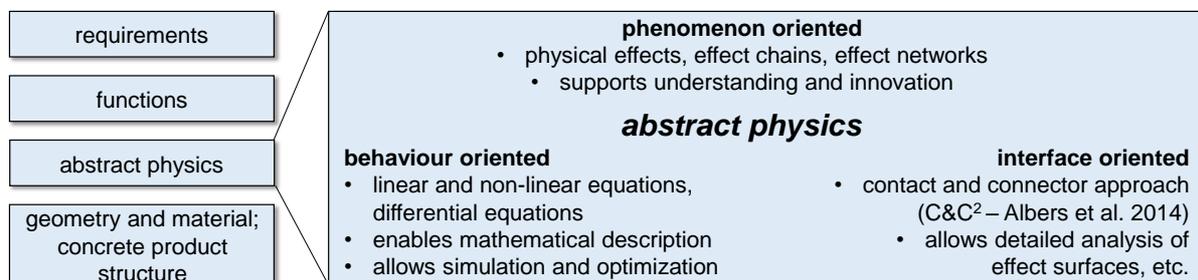


Figure 1. Perspectives on abstract physics, their elements and their advantages

A conclusive approach in the area of CDS is reported by [Wökl and Shea \(2009\)](#), who propose a computational product model in SysML for conceptual design which also includes abstract physics. Further research activities in this field include phenomenon oriented (compare [Helms, 2013](#)) and

behaviour oriented aspects (compare Muenzer and Shea, 2017); these activities could be successfully applied for the creation of multiple solution alternatives of hybrid car concepts (Rigger et al., 2016). From the literature analysis, it can be concluded that several scientific approaches concern the analysis and synthesis of physical structures or the physical behaviour, but that the existing approaches which aim at supporting engineers to consciously chose appropriate physical effects and the design of effect chains and effect networks which is connected to the other models in the product development processes, are not yet applied in many industrial product development processes. A promising approach for an integration of all aspects of abstract physics is the application of graph-based languages using UML for this endeavour, because it allows a compilation leading to a detailed geometrical model; the background of this concept will be explained in the next section.

3. Graph-based design languages

Today's product development processes are characterised by a multiplicity of different domains, which leads to immense efforts for exchanging data and information between these domains. Research in the last years was able to show that it is possible to use a common language to describe the product lifecycle - for instance a graph-based design language based on the Unified Modeling Language (UML). The general concept of graph-based design languages for engineering design was firstly proposed by Rudolph (2002). Examples of this kind of rule-based digital engineering were described for power poles (Alber and Rudolph, 2004), a complete satellite system (Groß, 2013), gear systems for trains (Holder et al., 2017), multicopters (Ramsaier et al., 2017), balanced two-wheel scooters (Wünsch et al., 2018), exhaust systems (Vogel, 2019), automotive dashboards (Walter et al., 2019), car bodies (Zech et al., 2019) and automated guided vehicles (Stetter, 2020). The application of a graph-based design language leads to a central data model in form of a design-graph (which is created ("compiled") from the UML model) with interfaces to all domains in the product lifecycle. The central advantage of the application of graph-based design languages is the resulting inherent consistency of the generated graph because all different domains receive their information from this data model. It is possible to generate requirement models (e.g. conventional lists or ReqIf-models), to generate function models, the geometry in form of a 3D model and a bill of materials (BOM). In future also models of the abstract physics will be available. The underlying digital design process with graph-based design languages is illustrated in Figure 2.

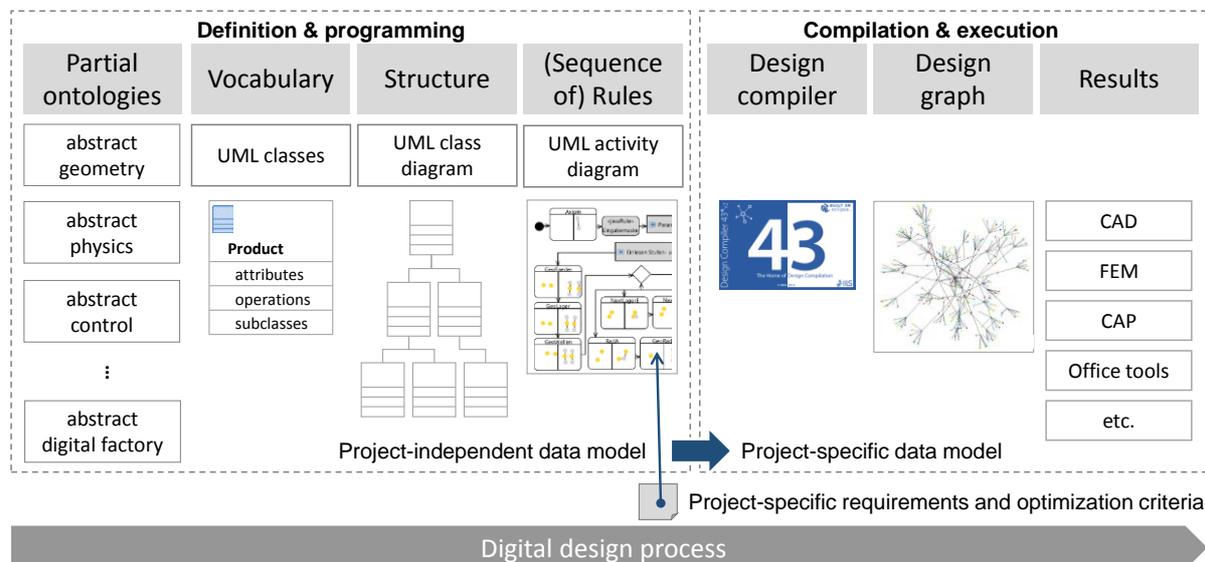


Figure 2. The digital design process with graph-based design languages

The basis for the creation of a graph-based design language are the so-called partial ontologies. An ontology defines the basic terms, concepts and relations comprising the vocabulary of a domain (Reichwein, 2011). For instance, the partial ontology "abstract geometry" defines the basis geometrical

terms, concepts and relations in the domain of industrial product development. On this basis, the responsible engineer can program the vocabulary (UML classes for product entities such as product components), the structure (UML class diagram - the structure of product entities) and rules (UML activity diagram - model transformations) for a whole product spectrum. This project-independent data model is then objected to project-specific requirements and optimization criteria and an automated compilation and execution process takes place. A computer software which is able to compile a graph-based design language is the “Design Compiler 43” (IILS mbH 2019). This design compiler was developed by the IILS mbH in cooperation with the University of Stuttgart and has the unique capability to create detailed and complex geometry models from UML diagrams. It is consequently possible to create all kind of domain-specific proprietary data formats from the design graph, which is the result of the compilation. In this paper, this engineering framework based on graph-based design languages will be expanded on the level of abstract physics; this expansion is explained using an exemplary product development - a steering system of a two-wheel scooter. This product development is explained in the next section.

4. Exemplary product development: Steering system for a scooter

The exemplary product development concerns the steering system of a self-balancing two-wheel scooter (Schuster and Pahn, 2018). The scooter serves as a main use-case of the underlying research project “Digital product life cycle” (ZAFH DiP). The system of interest is the so-called steering system. It has to be pointed out that this term may be misleading as the real steering system is a differential drive and steering is achieved just by different angular velocities of the two wheels. However, an ergonomic system is needed which will allow the user to express his/her turning wish. Experience with other balanced two-wheel scooters have shown that a concept is advantageous which allows to push a lever in sideward direction against a reverse force. Figure 3 shows the balanced two-wheel scooter first as digital model and secondly in two variations with different steering concepts.

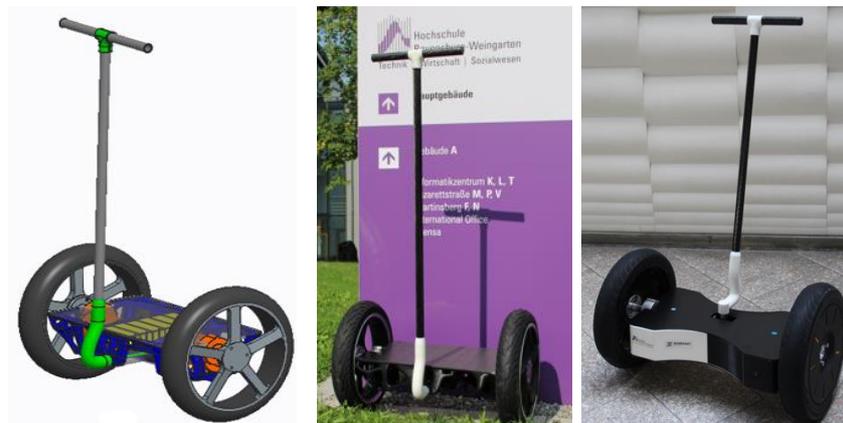


Figure 3. Balanced two-wheel scooter; digital model (left), realised variant 1 (middle), realised variant 2 (right)

A holistic functional analysis of the intended steering system is possible applying a slightly modified integrated function modelling framework (IFM), which has received increasing attention in the last years (compare Eisenbart et al., 2016; Ramsaier et al., 2017; Elwert et al., 2019). A part of this function model (the state view) is shown in Figure 4. It can be seen that the steering force from the user is transformed to a moment in the shaft. This moment leads to the rotation of an encoder, which can send information to the steering control. If the steering control detects a movement of the lever to the right, it will accelerate the left wheel and decelerate the right wheel in order to initiate the intended turning manoeuvre. Additionally, in the system some elastic element needs to be present in order to create the reverse force which is desired by the user of the scooter. It is obvious that the connection of functions and the flows of the operands energy and signal are clarified in this model.

In the product development process, the design of the elastic element was a challenge. The lever for the user of the scooter is rather long in order to accommodate large users, which want to stand upright. Consequently, the torque in the shaft is rather high and an elastic element also needs to create large

counter torques or forces. The development team created two alternative solutions, which both lead to the identical functionality but employ different physical effects and effect networks. The first solution idea was the application of two or four coil springs for creating the reverse force.

steering force from user	initial position	unloaded position: no reverse force	steering force from user	initial position encoder	state
P1			P1		process
moment in the shaft	initial position	initial position	moment in the shaft	initial position encoder	state
	P2		P2	process state of P2	process
	final position	unloaded position: no reverse force	movement of the shaft	final position encoder	state
		P3	P3		process
moment in the shaft	final position	loaded position; reverse force	reverse force from elastic element	final position encoder	state
lever and shaft	encoder	elastic element	energy	signal	
Actors			Operands		

Figure 4. Part of the function model of the steering system of the scooter (modified IFM model)

The principal solution, a digital model and the realised solution are shown in Figure 5.

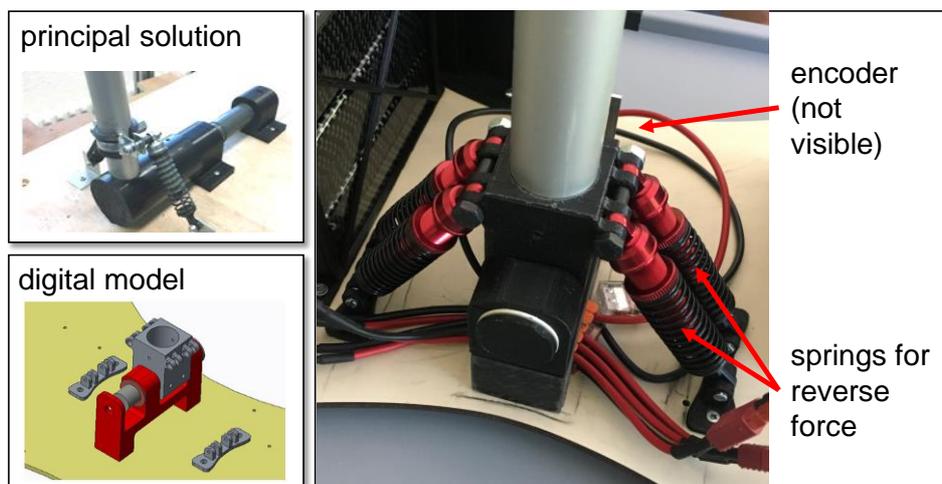


Figure 5. Solution concept with coil springs

The physical effect network for the coil spring solution is shown in Figure 6 (phenomenon oriented perspective).

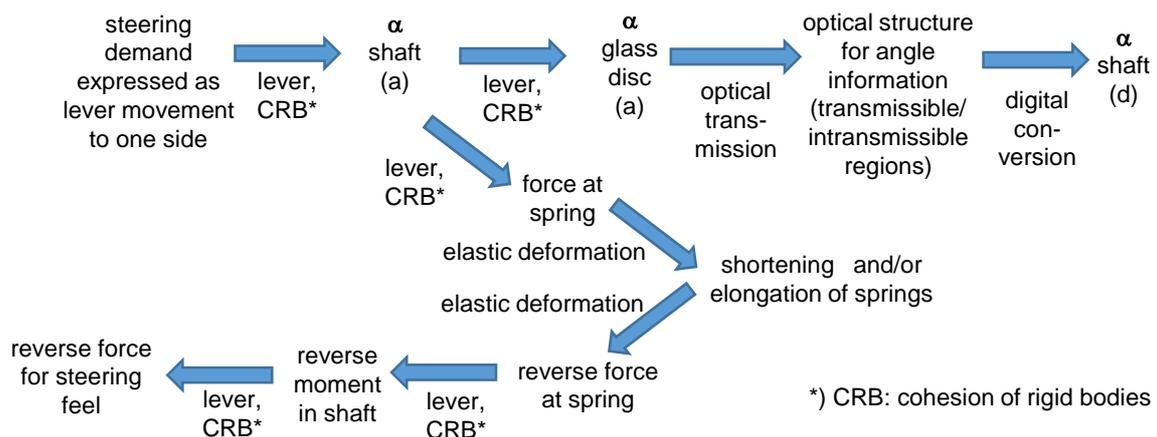


Figure 6. Physical effect network for the coil spring solution concept

Figure 7 shows the second solution concept. In this concept, the reverse torque is realised by means of an elastomer shaft. The torsion of his shaft leads to a reverse moment which will lead to a reverse force for the user, because of the physical effects “law of the lever” and “cohesion of rigid bodies”. Additionally, for reasons of cost reduction, a resistive potentiometer is used instead of the encoder. The physical effect network for the elastomer shaft solution is somewhat different, but could not be

shown in this paper due to space restrictions. The integration of the two different physical effect networks into the engineering framework is described in the next section.

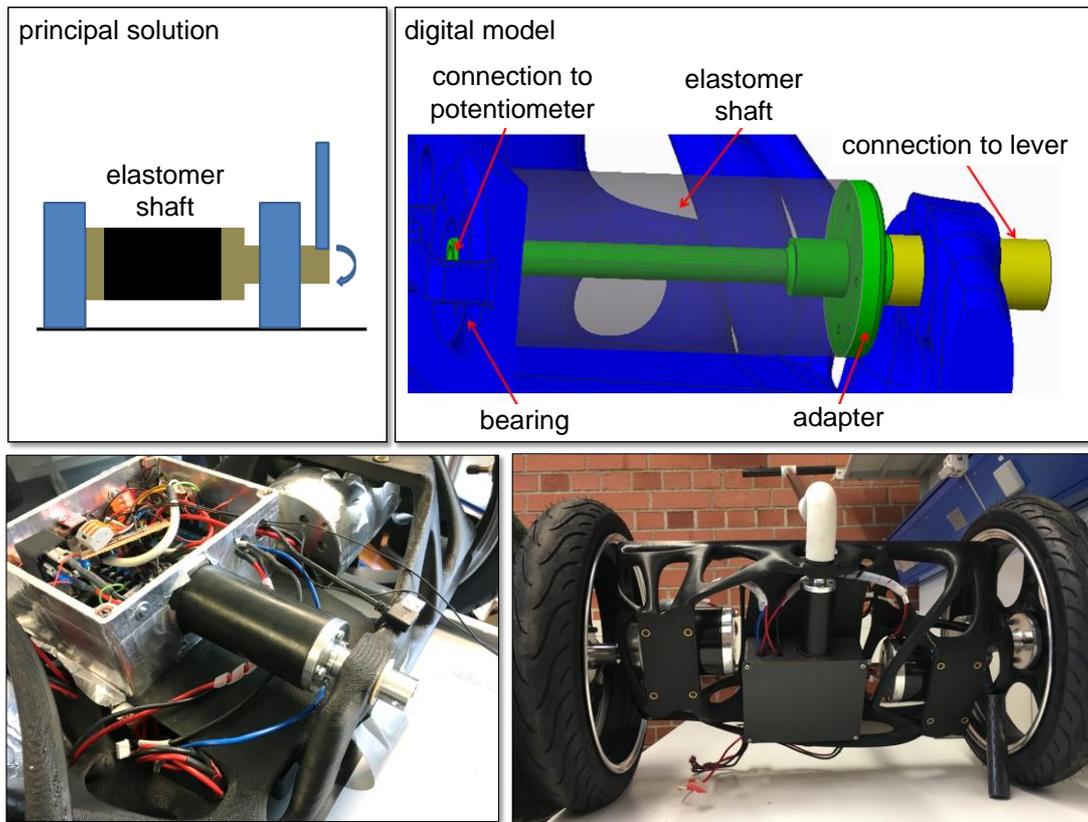


Figure 7. Solution concept with elastomer shaft

5. Integration of abstract physics into the engineering framework

In order to integrate abstract physics into the digital product design using design languages, the existing design language for functional modelling was extended to include abstract physics. The class diagram of the design language for function modelling is shown in Figure 8.

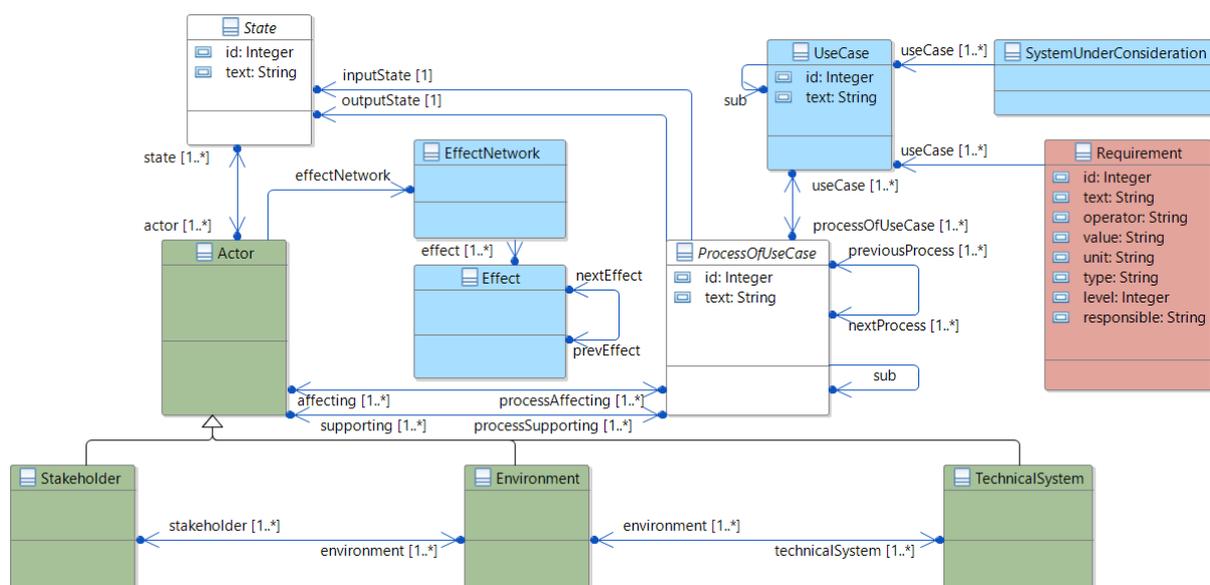


Figure 8. Class diagram of the function modelling design language

The abstract physics is integrated via the class “Effect” and described in more detail in a separate class diagram, shown in Figure 9. A physical effect automatically inherits from the “Effect” class. Each physical effect can therefore be used in the sense of function modelling.

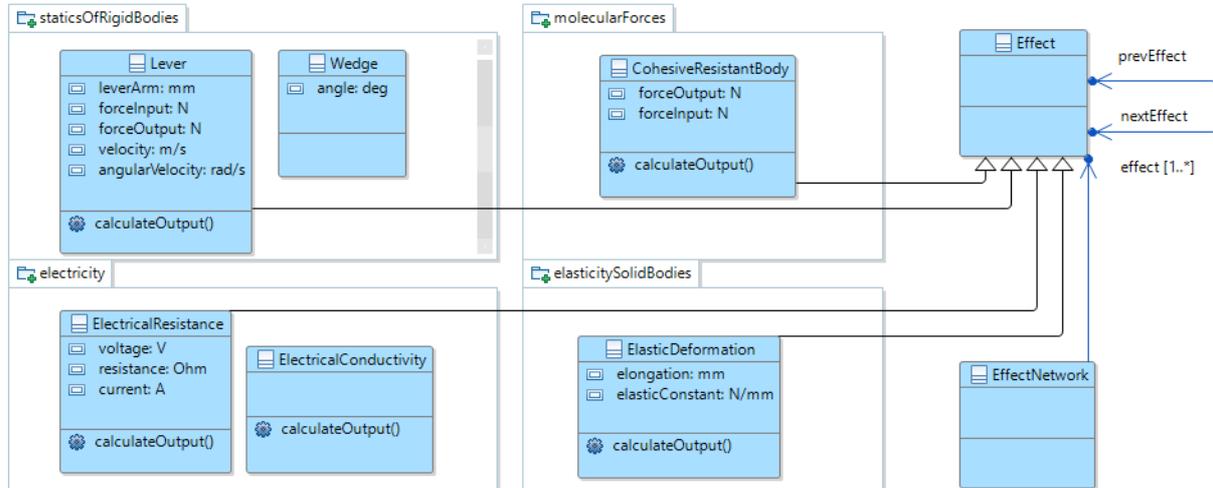


Figure 9. Class diagram of the abstract physics / physical effects

The physical effect class describes the logic of the effect. For example, the “Lever” class describes the relationship between input and output variables. The lever force is the applied force multiplied by the lever arm. One of the peculiarities of graph-based design languages can be used, namely the integrated Solution Path Generator for solving algebraic equations. To illustrate the modelling, Figure 10 summarizes the main elements activity diagram, rules and design graph (compare Figure 2).

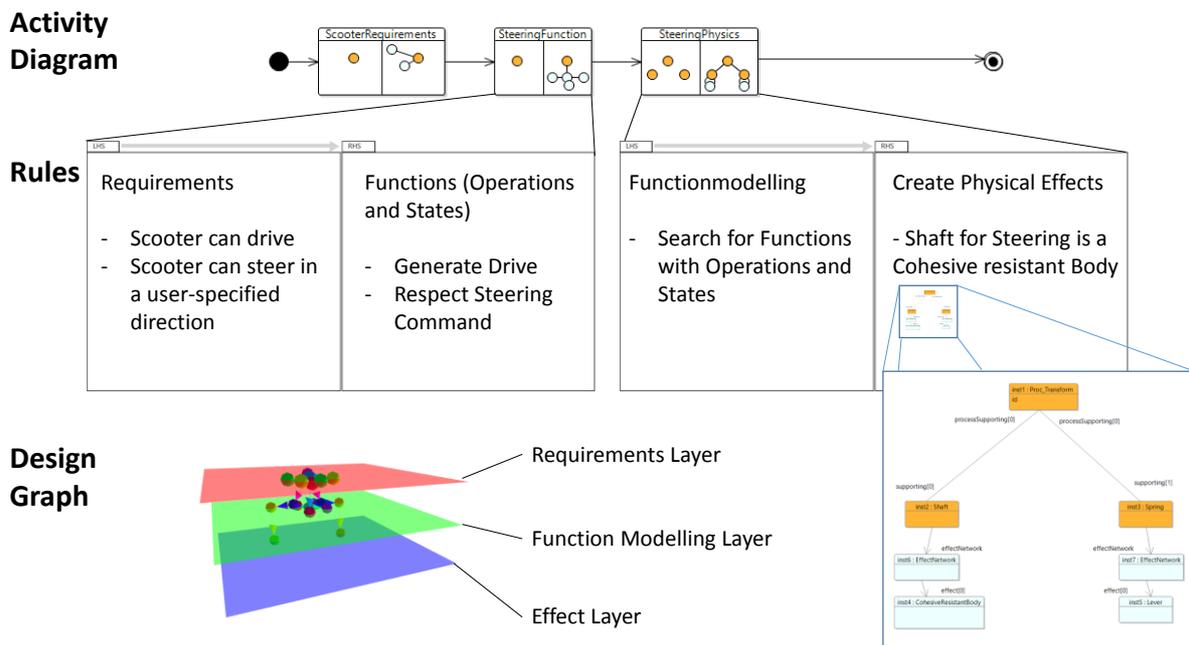


Figure 10. Main elements of graph based languages including the modelling of abstract physics

This integration of several aspects of abstract physics (phenomenon oriented, behaviour oriented and interface oriented) in a digital engineering framework can lead to considerable advantages in terms of product and process costs as well as product development speed and quality, because the integration allows an automatic choice and optimization on the decisive physical layer. The presented framework is characterised by the ability to compile numerous instances of information formulated in UML and to synthesize a design graph with interconnected geometrical, physical and structural information.

This engineering framework is able to generate and evaluate numerous product variants in relatively early phases of the product development cycle and can thus contribute to an optimised selection of solutions and a higher decision quality.

6. Conclusions and outlook

This paper demonstrates the integration of abstract physics models into a holistic engineering framework based on graph-based design languages. Such languages support an object oriented programming of a complete product spectrum. The integration of abstract physics allows the application of a Solution Path Generator for optimised solutions on the physical level and a synthesis of physical effect networks according to different criteria such as the robustness of the physical effects (compare Mathias et al., 2011). Further research will extend the investigation with additional product examples and will explore additional automatic optimisation algorithms.

Acknowledgement

The project “digital product life-cycle (ZaFH)” (information under: <https://dip.reutlingen-university.de/>) is supported by a grant from the European Regional Development Fund and the Ministry of Science, Research and the Arts of Baden-Württemberg, Germany (information under: www.rwb-efre.baden-wuerttemberg.de).

References

- Alber, R. and Rudolph, S. (2004), “On a Grammar-Based Design Language That Supports Automated Design Generation and Creativity”, In: Borg, J.C., Farrugia, P.J. and Camilleri, K.P. (Eds.), *Knowledge Intensive Design Technology*, Springer US, Boston, MA, pp. 19-35. https://doi.org/10.1007/978-0-387-35708-9_2
- Albers, A. and Wintergerst, E. (2014), “The Contact and Channel Approach (C&C2-A): relating a system’s physical structure to its functionality”, *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*, No. Kim 2010, pp. 61-72.
- Atkin, R.H. (1965), “Abstract Physcis”, In: *Il nuovo cimento*, Vol. XXXVIII No. 1, pp. 496-517. <https://doi.org/10.1007/bf02750478>
- Cross, N. (2008), *Engineering Design Methods: Strategies for Product Design*, John Wiley and Sons Ltd.
- Ehrlenspiel, K. and Meerkamm, H. (2013), “Integrierte Produktentwicklung”, Denkabläufe, Methodeneinsatz, Zusammenarbeit. Carl Hanser. <https://doi.org/10.3139/9783446436275>
- Eisenbart, B. et al. (2016), “A DSM-based Framework for Integrated Function Modeling: Concept, Application and Evaluation”, *Research in Engineering Design*, Vol. 28 No. 1, pp. 25-51. <https://doi.org/10.1007/s00163-016-0228-1>
- Elwert, M. et al. (2019), “Holistic Digital Function Modelling with Graph-Based Design Languages”, In: *Proceedings of the Design Society: International Conference on Engineering Design*, Vol. 1 No. 1, Cambridge University Press, pp. 1523-1532. <https://doi.org/10.1017/dsi.2019.158>
- Fritzson, P. (2014), *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley. <https://doi.org/10.1002/9781118989166>
- Gero, J. and Kannengiesser, U. (2014), “The Function-Behaviour-Structure Ontology of Design”, In: Chakrabarti, A., *L.T.M. Blessing: An Anthology of Theories and Models of Design*. Springer. https://doi.org/10.1007/978-1-4020-5131-9_21
- Gladysz, B., Spandl, L. and Albers, A. (2017), “A Function- and Embodiment-Based Failure Analysis Method for an In-Depth Understanding of Failure Mechanisms”, In *Proceedings of the 21st International Conference on Engineering Design, ICED17*, 21-25 August 2017, The University Of British Columbia, Vancouver, Canada.
- Groß, J. (2013), “Aufbau und Einsatz von Entwurfssprachen zur Auslegung von Satelliten”, *Dissertation, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen*, Universität Stuttgart, Stuttgart, 2013.
- Guerineau, B. et al. (2017), “Management of Heterogeneous Information for Integrated Design of Multidisciplinary Systems”, *Procedia CIRP*, Vol. 60 No. 2017, pp. 320-325. <https://doi.org/10.1016/j.procir.2017.02.020>
- Helms, B. (2013), “Object-Oriented Graph Grammars for Computational Design Synthesis”, *Diss. TU München*.
- Holder, K. et al. (2017), “Model-Based Requirements Management in Gear Systems Design Based On Graph-Based Design Languages”, In: *Applied Sciences*, Vol. 7, p. 1112. <https://doi.org/10.3390/app7111112>
- IILS mbH, Design Compiler 43 (www.iils.de), Last access: 18.11.2019
- Lang, H. (2016), “Die methodische Integration empirischer Analysen in die frühen Phasen eines Entwicklungsprozesses”, *Dissertation TU Graz*.

- Lindemann, U. (2009), *Methodische Entwicklung technischer Produkte*, Springer. <https://doi.org/10.1007/978-3-642-01423-9>
- Mathias, J. et al. (2011), "Selection of Physical Effects Based on Disturbances and Robustness Ratios in the Early Phases of Robust Design", In: *Proceedings of the International Conference on Engineering Design, Iced11*, 15-18 August 2011, Technical University of Denmark.
- Munzer, C. and Shea, K. (2015), "A Simulation-Based CDS Approach: Automated Generation of Simulation Models Based From Generated Concept Model Graphs", *Proceedings of the ASME, International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, Boston, MA, 2015. <https://doi.org/10.1115/detc2015-47353>
- Muenzer, C. and Shea, K. (2017), "Simulation-Based Computational Design Synthesis using Automated Generation of Simulation Models from Concept Model Graphs", *Journal of Mechanical Design*, Vol. 139 No. 7, p. 071101. <https://doi.org/10.1115/1.4036567>
- Pahl, G. et al. (2007), *Engineering Design: a systematic Approach*, Springer-Verlag. <https://doi.org/10.1007/978-1-84628-319-2>
- Ponn, J. and Lindemann, U. (2011), *Konzeptentwicklung und Gestaltung technischer Produkte*, Springer. <https://doi.org/10.1007/978-3-540-68563-0>
- Ramsaier, M. et al. (2017), "Digital representation of product functions in multicopter design", In: Maier A. et al.: *Proceedings of the 21st International Conference on Engineering Design (ICED 17), Vol 1: Resource Sensitive Design, Design Research Applications and Case Studies*, Vancouver, Canada, 21-25.08.2017, pp. 369-378. <https://doi.org/10.1016/j.procir.2016.06.008>
- Reichwein, A. (2011), "Application-specific UML Profiles for Multidisciplinary Product Data Integration", *Dissertation*, Universität Stuttgart.
- Rigger, E., Münzer, C. and Shea, K. (2016), *Estimating the Potential of State of the Art Design Automation - Tasks, Methodsw, and Benefits*, In: Marjanovic, D., Storga, M., Neven, G., Bojetic, N. and Stanko, S. (Eds), *Proceedings of the DESIGN 2016, 14th International Design Conference*, Vol. 1, pp. 421-432.
- Rudolph, S. (2002), Übertragung von Ähnlichkeitsbegriffen. Fakultät Luft- und Raumfahrttechnik und Geodäsie. Habilitation thesis, University of Stuttgart, Stuttgart, Germany.
- Schuster, J. and Pahn, F. (2018), *Entwicklung und Bau zweier konzeptionell unterschiedlicher Segways*. Bachelor-Thesis Ravensburg-Weingarten University (RWU).
- Stetter, R. (2020), *Fault-Tolerant Design and Control of Automated Vehicles and Processes. Insights for the Synthesis of Intelligent Systems*. Springer. <https://doi.org/10.1007/978-3-030-12846-3>
- Vogel, S. (2019), "An application-independent continuum mechanics interface for virtual engineering", *Engineering with Computers*, Vol. 35, pp. 551-565. <https://doi.org/10.1007/s00366-018-0617-3>
- Wagner, C. (2018), "Funktionsintegration im Rahmen einer fertigungsgetriebenen Produktentwicklung", *Dissertation*, TU Darmstadt.
- Walter, B., Kaiser, D. and Rudolph, S. (2019), "Machine-executable Model-based Systems Engineering with design languages", In: Banach, R., Razavi, J., Lesecq, S., Debicki, O., Mareau, N., Foucault, J., Correvon, M. and Dudnik, G. (Eds.) *Complex Systems Design & Management*, Springer. https://doi.org/10.1007/978-3-030-04209-7_25
- Wökl, S. and Shea, K. (2009), "A Computational Product Model for Conceptual Design using SYSML", In: *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2009*, August 30-September 2, 2009, San Diego, California, USA. <https://doi.org/10.1115/detc2009-87239>
- Wulf, J. (2001), "Elementarmethoden zur Lösungssuche", *Dissertation*, TU München. Dr. Hut, München.
- Wünsch, F. et al. (2018), "Executable Cost-Sensitive Product Development of a Self-Balancing Two-Wheel Scooter with Graph-Based Design Languages", In: Marjanović, D., et al. (Eds.): *Proceedings of the 15th International Design Conference DESIGN 2018*, Dubrovnik. <https://doi.org/10.21278/idc.2018.0409>
- Zech, A. et al. (2019), "Novel approach for a holistic and completely digital represented product development process by using graph-based design languages", *Procedia CIRP*, Vol. 79, pp. 568-573. <https://doi.org/10.1016/j.procir.2019.02.102>
- Zheng, C. et al. (2017), "Multidisciplinary design methodology for mechatronic systems based on interface model", *Research in Engineering Design*, Vol. 28, pp. 333-356. <https://doi.org/10.1007/s00163-016-0243-2>