vehicles

MDPI

Article

Innovative Vehicle Design Processes Based on the Integrated Framework for Abstract Physics Modeling (IF4APM)

Ralf Stetter

Department of Mechanical Engineering, Ravensburg-Weingarten University (RWU), 88250 Weingarten, Germany; ralf.stetter@rwu.de

Abstract: In industrial vehicle design processes, most companies have implemented model-based systems engineering (MBSE). As a consequence, design processes are nowadays not driven by documents, but by digital models of the vehicle to be developed and its components. These models exist on different levels of abstraction. The models on the requirements level are already well defined as well as the models of the defined product behavior and product properties. In recent years, the specification of models on the level of product functions was largely clarified, and elaborate frameworks already exist. However, this is not yet true for the level between functions and definite properties; this level can be referred to as "abstract physics". The enormous importance of this level, which, amongst others, can represent the physical effect chains which allow a vehicle component to function, is expressed by several researchers. Several research works aim at specifying models on this level, but, until now, no general consensus can be identified, and the existing model specifications are less appropriate for the early stages of vehicle design. This paper explains an Integrated Framework for Abstract Physics Modeling (IF4APM), which incorporates different perspectives of abstract physics and is suited for the early phases. The explanation is based on typical components of several kinds of vehicles. The main advantages of the proposed approach are the consistent interconnection of abstract product models, the clearness and understandability of the resulting matrices, and the aptitude to be used in the early phases of a vehicle design process.

Keywords: vehicle development; abstract physics; design models; model based design

check for updates

Citation: Stetter, R. Innovative Vehicle Design Processes Based on the Integrated Framework for Abstract Physics Modeling (IF4APM). *Vehicles* **2024**, *6*, 1345–1363. https://doi.org/10.3390/vehicles6030064

Academic Editor: Patrick Gruber

Received: 4 July 2024
Revised: 26 July 2024
Accepted: 31 July 2024
Published: 3 August 2024

Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In today's vehicle design processes, a gap can be observed between the rather clearly and concisely described functionality of the vehicle and its modules and the defined behavior and properties of them. This gap is caused by a lack of information that describes which physical phenomena realize the functions by which as well as how and to what extend these phenomena are acting. In recent years, the complexity of vehicle development is further increasing because of a growing share of intelligent components, because of increasing technical complexity of vehicles and their modules, and because of steadily increasing model ranges and product families [1]. Current advancements in production technologies, growing ecological constraints, and globally diversified customers require early design methods which allow designers to explore the available design space systematically [2]. Systematic design methods need to be based on concise and reliable products, production processes, and production system information on all levels of abstraction. Concerning the virtual models of technical systems (such as vehicles and their components), a wide consensus that four levels of abstraction can be distinguished can be observed (compare e.g., [3–6]). It is the current consensus in design science that these levels are not appropriate to be the basis for a general project planning of design, but that all levels may be used in all phases of design processes with changing intensity. The project definition and concept phases are usually characterized by a strong emphasis on requirements, functions, and abstract physics, while the detail design and production preparation phases mainly employ

detailed descriptions of geometry, material, and structure. Still, changing customer expectations and unexpected testing outcomes can always lead to changed requirements and can require a revisit to other more abstract levels. The four levels of abstraction of product models are shown in Figure 1.

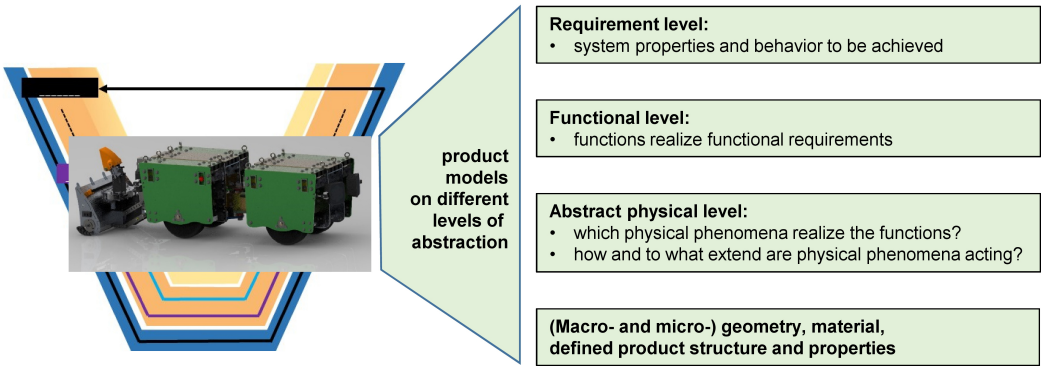


Figure 1. Vehicle product models on different levels of abstraction.

As visible in Figure 1, the most abstract level of systems models contains the requirements, i.e., the intended system properties and system behavior, in order to fulfill the customer expectations. The integration of the entities on this level of abstraction—the requirements—into engineering frameworks has been the focus of research projects for several years [7–9]; detailed specifications of model structures and contents are established. The so-called Requirements Interchange Format (ReqIF), for instance, defines a file format, which may be used for the exchange of requirements with metadata [10]. A part of these requirements will lead to functions that allow the realization of these functional requirements (second level of abstraction). For this level, detailed frameworks have been developed in the last decade [11–16]). The main focus of this paper is the third level—abstract physics. As stated above, the main content of this level is information, where physical phenomena realize the functions, as well as how and to what extend these phenomena are acting. For an analysis and understanding on the micro-level, working surfaces can also be of great interest on this level of abstraction. The least abstract level (in many publications, thhis is also described as the most "concrete" level; however, in this paper, this adjective is avoided in order to prevent confusion with a prominent building material) concerns the macro- and micro-geometry, material, product structure, and product properties. This abundant amount of information is stored in established data formats such as computer aided design (CAD) files, and the model specifications are well documented.

One may ask, why the gap between the second and the fourth level presents a real challenge for innovative vehicle design processes. Albers et al. [17] report a prominent problem for engineers to embody functional ideas in sub-systems and to localize these functions in mechanisms and their interfaces. Drave et al. [14] point out that technical systems fulfill certain functions through physical effects acting between components and emphasize a strong correlation between these physical effects and the architecture of the system. In connected research, Hoepfner et al. [18] report a lack of emphasis on this abstraction level, and that current model-based systems engineering (MBSE) frameworks do not integrate formal models for modeling the physical behavior in order to allow a seamless link to digital system models. Additionally, Zerwas et al. [19] point out that principle solutions are essential because they can efficiently describe possible solutions by specifying effects as well as active surfaces and material(s) without losing a functional orientation. The early phases are decisive because so-called breakthrough products (extremely successful products which completely replace prior products, such as fuel injection systems instead of carburetors) are usually characterized by a major change in the physical structure [20]. Especially in the early phases, it is important to note that not all aspects of a technical system can be represented with concrete parameters. In industrial companies, often so-called "engineering samples" are used to represent the intended realization of certain compo-

nents, especially if their appearance is decisive for customer satisfaction [21]. In industrial vehicle development processes, mature V-model-based development architectures can be observed, especially in the field of software and control development [22]. However, in the scope of mechanical and mechatronic engineering, certain levels of abstraction are not yet covered [18]. Generally, models on all level of abstraction and during all phases of product development are desirable because the employment of models as the primary artifacts can lead to an improved semantic integration of digital assets and robust early analysis [23]. Consequently, the proposed approach aims to close the gap between the second and the fourth level visible in Figure 1. On the given abstraction level of product models, detailed information concerning material selection or manufacturing processes and necessities are consciously omitted in order to reduce complexity and to enhance transparency. It can be concluded that intensified and detailed scientific activity on this level of abstraction may lead to more efficient design processes, better knowledge reuse, and seamless digital processes in the early phases of design. This also leads to the formulation of the following research question:

> *How can different perspectives of abstract physics be combined in a manner that allows integration in engineering frameworks for the design of vehicles?*

This research question is analyzed using several vehicle design processes focusing on remote controlled vehicles and lifting devices for vehicles (Figure 2).

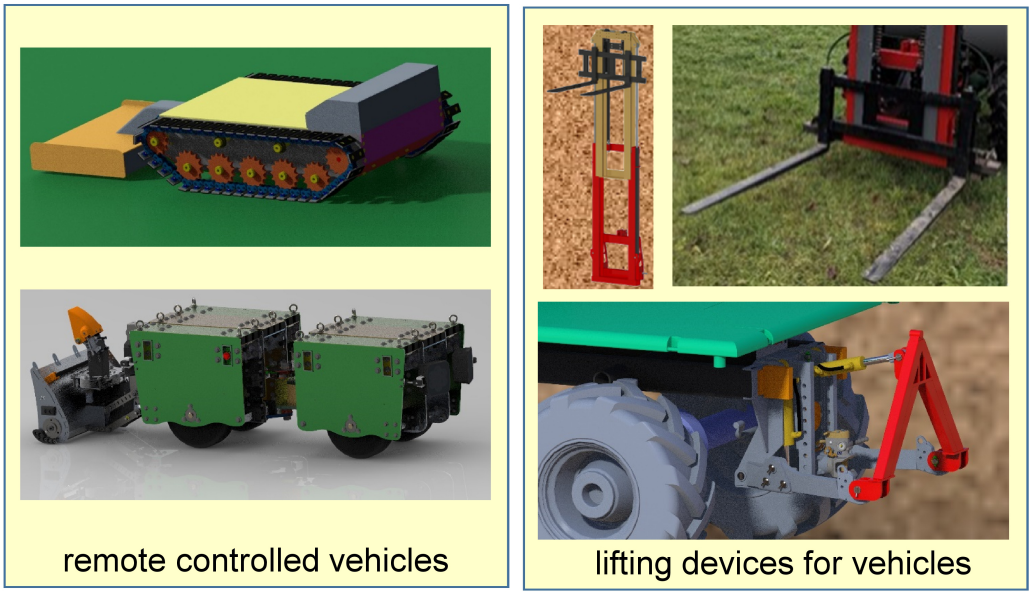


Figure 2. Vehicle and vehicle component development processes.

These vehicle design processes consider remote-controlled vehicles for purposes such as snow plowing or lawn mowing [24] (Figure 2 left side) and lifting systems in the area of agricultural technology—lifting masts for tractors [25] and power lifts for universal agricultural vehicles [26] (Figure 2 right side). As stated above, the levels of abstraction of product models are not appropriate to be the basis for a general project planning of design, but explicit project schedules need to be developed. A simplified timeline of vehicle design processes is visible in Figure 3.

In Figure 3, two main process flows are represented; the project-independent technology development flow as well as the project-oriented flow, which may be sensibly divided into the concept phase and vehicle development phase. For the project-oriented flow, typical mile-stones of a vehicle development process are given (compare [1,27]). An early phase concerns the concept development. In this phase, the important vehicle characteristics are derived, employing methods such as market research and benchmarking. The important product specifications are developed and important concept decisions are made. For a given project, product models on all levels of product concretization are employed and a

consideration of abstract physics is possible. However, the development of products components, which employ novel physical effect chains, very often is carried out outside the scope of a single vehicle development process, and can be understood as an independent ongoing technology development process. After the concept definition, the vehicle development process in the close sense takes place. On the basis of the product specification, a detailed system description is elaborated together with the preparation of all manufacturing and assembly processes. Still, changes of specifications are frequent, and, in some cases, the physical effect chains also need to be altered in this phase. However, the main area of interest concerning abstract physics are the technology development and the earlier phases of project oriented design. It is important to point out that the process planning and control in vehicle design processes is usually realized employing model-based system engineering (MBSE) [28–30]. In MBSE, product models on the different levels of abstraction are the primary process management objects and take over the role of other documents which are not directly product-related. The fact that models, which are generally agreed upon, are not yet available on the level of abstract physics can lead to mistakes, inefficient knowledge management, double work, and misunderstandings. Several research groups worldwide have started to address this issue; their main contributions will be described in Section 2 and form the state of the art. Based on this, the integrated framework is presented in Section 3. The vehicle design processes, which are analyzed in the ongoing research and serve for explaining the integrated frameworks, are characterized in Section 4, together with elements of the integrated framework. A final section (Section 5) concludes the paper and delivers an outlook on further research perspectives.

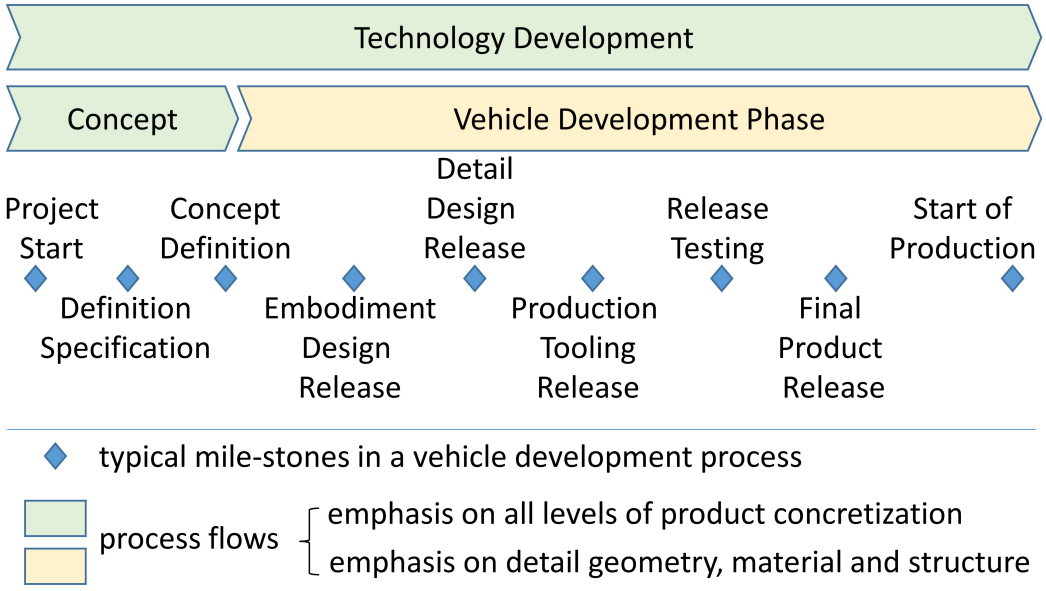


Figure 3. Timeline of vehicle design processes.

2. Perspectives of Abstract Physics

In earlier research [31], the different perspectives of abstract physics were analyzed based on the design processes of wind turbines. An overview of those perspectives is given in Figure 4.

In general, five perspectives can be distinguished as follows:

- a phenomenon-oriented perspective which consists of physical effects, effect chains, and effect networks and supports the understanding of physics and innovative system concepts;
- a control-oriented perspective which focuses on the elements of a control cycle (actor, process, and senor) and supports the synthesis of intelligent controlled systems [5];
- a behavior-oriented perspective which employs mathematical models such as differential equations in order to allow simulation and optimization;

- an interface-oriented perspective which focuses on interacting surfaces as part of a physical effect chains and allows a detailed analysis of these surfaces and their role in realizing the effect chain;
- a logic-oriented perspective which employs certain methods such as fault-tree analysis in order to analyze reliability and safety. These analyses frequently cover detailed phases of project-oriented design (compare Figure 3) and are therefore not included in the current state of the IF4APM.

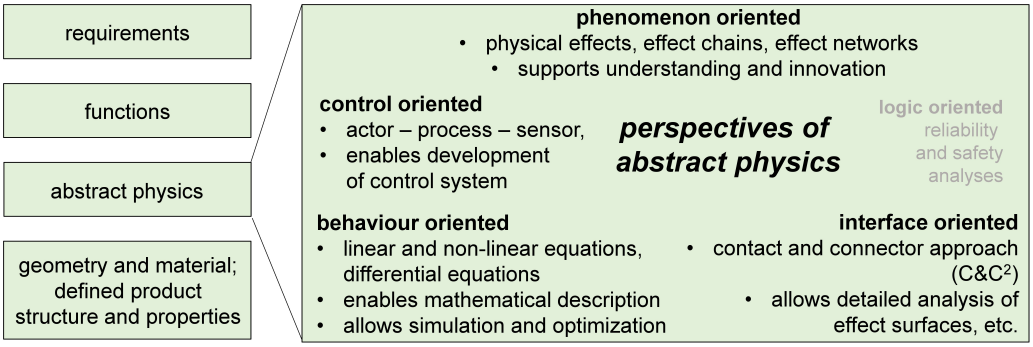


Figure 4. Perspectives of abstract physics.

2.1. Phenomenon-Oriented Perspective

The important entities of the phenomenon-oriented perspective are visible in Figure 5.

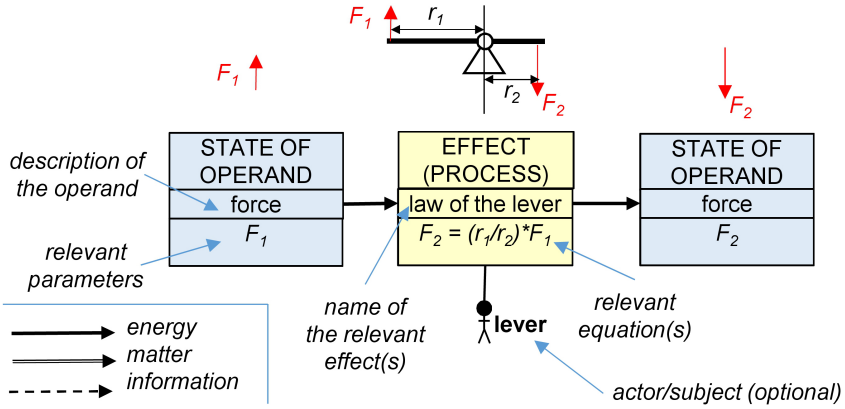


Figure 5. Entities of the phenomenon-oriented perspective.

The inputs of a physical effect (i.e., a distinct physical phenomenon which can realize an operation within a technical system [3,32]) are the states of operands (operands are the entities that are processed). The effect itself is often a transformation of energy, matter, or signal, which is based on a physical phenomenon. Common examples for physical phenomena range from statics and dynamics, over elasticity, vibration, molecular forces to thermal effects, etc. [4,33]. Additionally, interaction processes are possible (e.g., processes of actors which are within the system with actors which are outside of the system) [11]. The output(s) of the process are again states of operands. The subject—the actor—realizes the process. In the early stages of design processes, the effect might already be clear and decided, but the definite actor might still be an element of discussion.

2.2. Control-Oriented Perspective

A complementary perspective is control-oriented. In contrast to the phenomenon-oriented perspective, which is mainly focused on the actors, important aspects of this perspective are also the sensors and the application of their measurements for stabilizing the behavior of the system. The main entities of this perspective are shown in Figure 6.

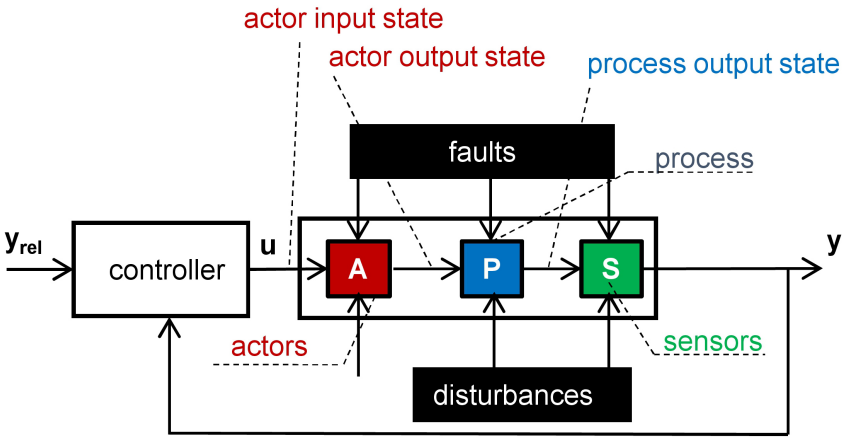


Figure 6. Entities of the control-oriented perspective.

The main element of this perspective are the control loops, consisting of actor, process, sensor, and controller. These control loops enable the use of feedback from sensor measurements for the control of the system with the goal to stabilize the behavior of the system or to follow a given trajectory [34]. Modern control systems also allow us to compensate disturbances and to detect and accommodate faults of actors, processes, and sensors [35,36]. In vehicle applications, complex control systems are present. A promising approach to tackle this enormous complexity are modularization and distribution approaches. Current research work concerning advanced vehicular mechatronic systems has led to the proposal of multi-agent systems in the control domain [37–39], including game-based control frameworks [40].

2.3. Interface-Oriented Perspective

Both the phenomenon-oriented and the control-oriented perspectives concentrate on the macro effects which realize the functionality and stability of a technical system. In contrast, the interface-oriented perspective focuses on the micro effects; the main entities are displayed and explained in Figure 7.

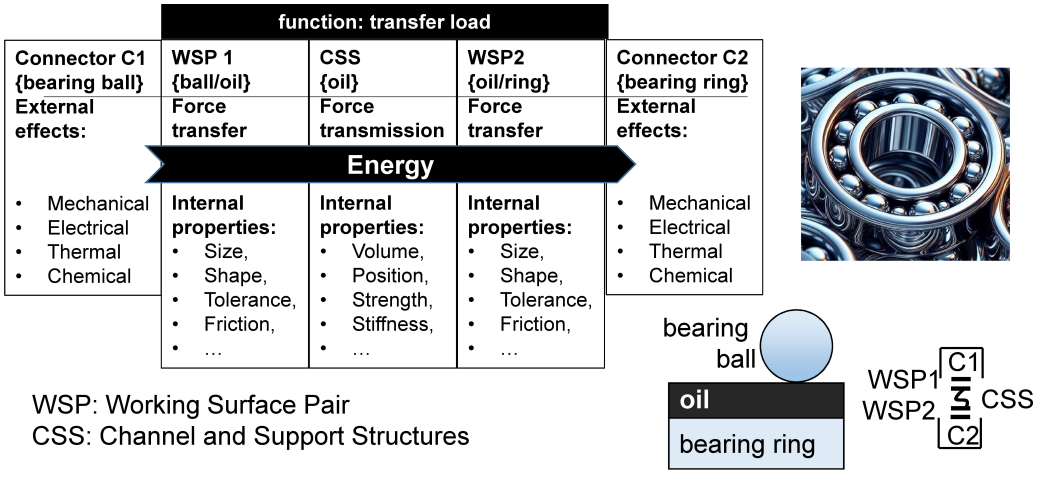


Figure 7. Entities of the interface-oriented perspective.

This perspective, which is referred to as the Contact and Channel Model (C&CM) and was introduced by Albers and Matthiesen [41], describes Working Surface Pairs (WSPs) that realize functions and Channel and Support Structures (CSSs) which connect the WSPs. WSPs are interfaces between components, and interfaces can be between solid surfaces or boundaries with the surfaces of liquids, gases, or fields, while CSSs are physical components or volumes of liquids, gases, or spaces that connect two WSPs [17,42–45].

2.4. Behavior-Oriented Perspective

Several researchers in engineering design science employ the function–behavior–structure paradigm (compare, e.g., [46–49]) and point out that functions and the concrete structure are connected by the behavior of the technical system. In some cases, this behavior can be described by means of mathematical equations. In Figure 8, an example using an electrical motor is given.

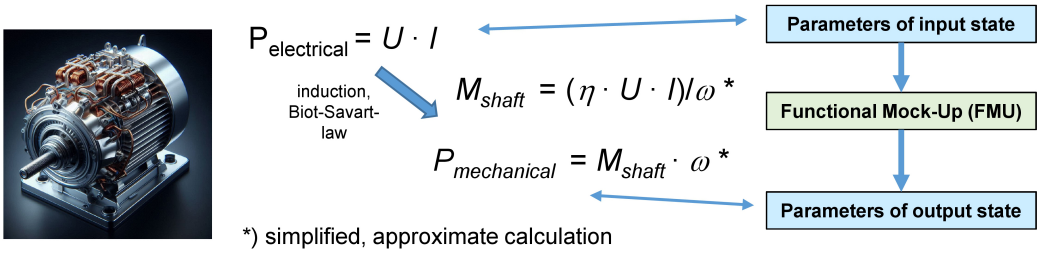


Figure 8. Entities of the behavior-oriented perspective.

Obviously, the simplified approximate calculation in Figure 8 does not represent the behavior of the electrical motor in detail. Usually, rather complex differential equations and simulations are necessary for a sensible representation and prognosis of the actual technical system. For several years, a modularization of simulation frameworks has been in development; for instance, in order to reduce complexity. A promising approach employs so-called functional mock-up (FMU) simulation components which are based on the functional mock-up interface (FMI) standard [50–53]. Today, some research groups have started to address the obvious gap between function and structure, and are employing more than one perspective of abstract physics in this endeavor. Detailed requirements concerning engineering frameworks that include abstract physics perspectives were formulated [21]. Ongoing research concerns a SysML-based modeling method that allows the mechanical domain to represent links between its requirements, functions, and principle solutions; the intended models not only contain elementary effects, but entire principle solutions with behavior models and appropriate test workflows [14,15,18,19]. Further relevant research is focusing on multi-domain matrices [54]. The IF4APM is employing the modeling intentions and solutions of earlier research and is focusing on early stages of vehicle design, where information and product structure are still unclear. It is a central intention to support design engineers with their accustomed procedure styles in vehicle development.

3. Description of the Framework

The IF4APM intends to integrate different perspectives of abstracts physics; its main entities are listed in Table 1 together with an explanation.

Table 1. Entities of the IF4APM.

Element	Explanation
Operand	The operand is the entity that is processed by the technical system, e.g., mechanical energy. Operands in a technical system appear in the general forms of matter, energy, and signal. The operand can also be a consequence or a prerequisite of energy, e.g., a force.
State	The state describes the condition of the operand e.g., the force on one side of a lever.
Process	The process describes the transformation of the operand, e.g., an enlargement of the force.

Table 1. *Cont.*

Element	Explanation
Actor	The actor is a component or sub-system of the technical system which is responsible for the realization of a process, e.g., a lever is responsible for the enlargement of the force.
Effect	The effect represents a physical phenomenon that contributes to the realization of the process, e.g., law of the lever.
Behavior	The behavior quantitatively represents the change in the elements of the technical system to realize the process. Usually, the alteration of certain parameters is the focus of the behavior.
Working Surface (WP)	The working surface is a part of an object of a system that is necessary for the realization of the process.
Working Surface Pair (WSP)	The WSP is the interface between objects that take part in the process.
Channel and Support Structure (CSS)	The CSS describes components or volumes of liquids, gases, or spaces that connect WSPs in order to realize the process.

The relations between the entities of the IF4APM and their contribution to the abstract physical structure of a technical system can be illustrated in a unified modeling language (UML)-based class diagram (Figure 9).

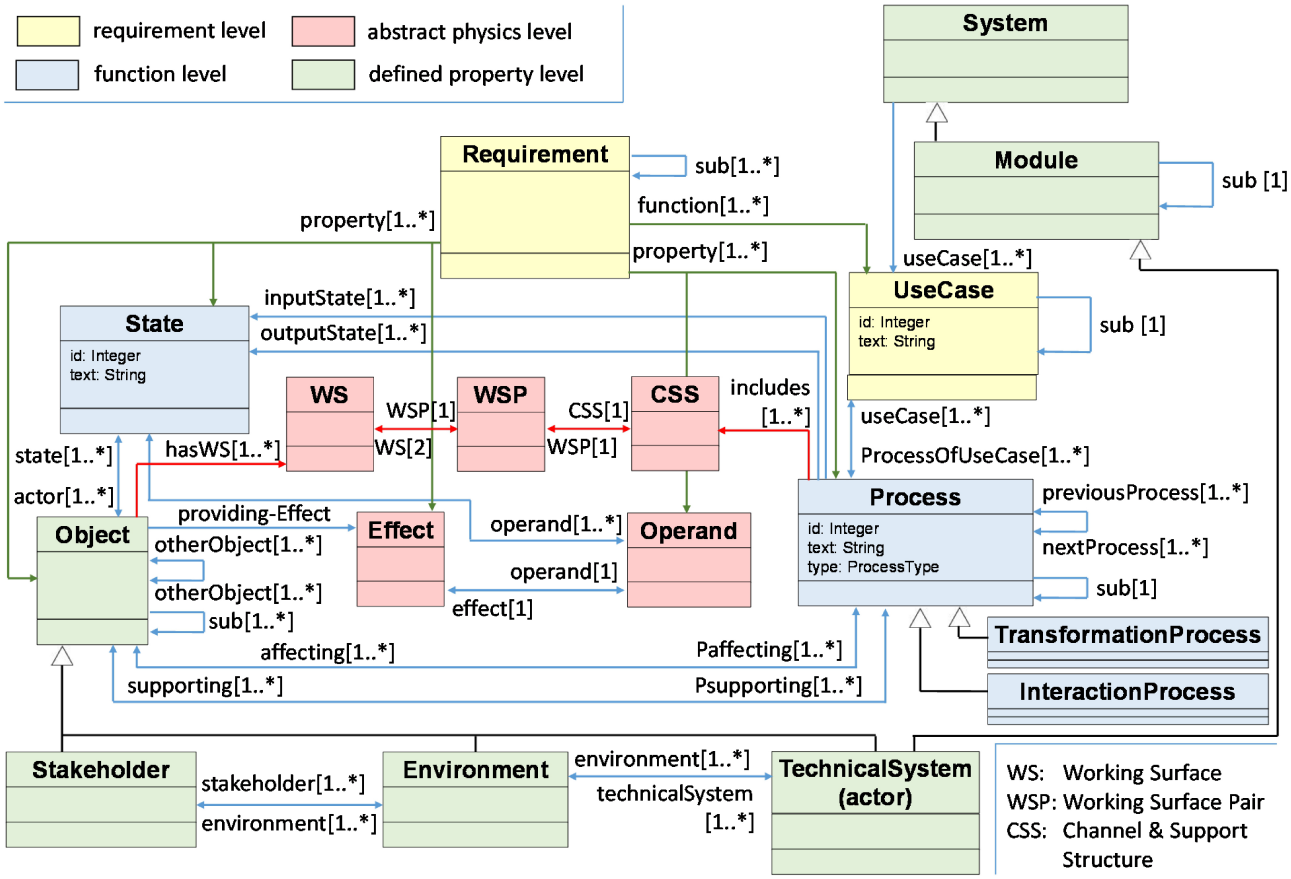


Figure 9. Relations between the entities of the IF4APM.

This class diagram was developed based on class diagrams of the framework for integrated function modeling (IFM) [13,55] and earlier research [21]. The different levels of abstraction of the respective product model are indicated with colors. The yellow entities *Requirement* (specific objectives of a system) and *Use Case* (scenarios of applying the technical

system for a specific purpose) concern the most abstract level. The blue entities *State* and *Process* are located on the function level, but also play an important role on the level of abstract physics. On this level, the red entities *Effect*, *Operand*, *Working Surface*, *Working Surface Pair*, and *Channel Support Structure* are employed in the IF4APM for describing the physical effect chain. The green entities *System*, *Module*, *Object*, *Stakeholder*, *Environment*, and *Technical System* concern the level of detail geometry and material. Similar to the well-known IFM, the main representation of the IF4APM is a combination of modular matrices because they enable a concise, structured representation [11]. Matrices are excellent tools for managing complex dependencies in technical systems [54]. Such matrices are a relatively intuitive means for modeling information in a structured manner. An overview of the structure of the IF4APM is given in Figure 10.

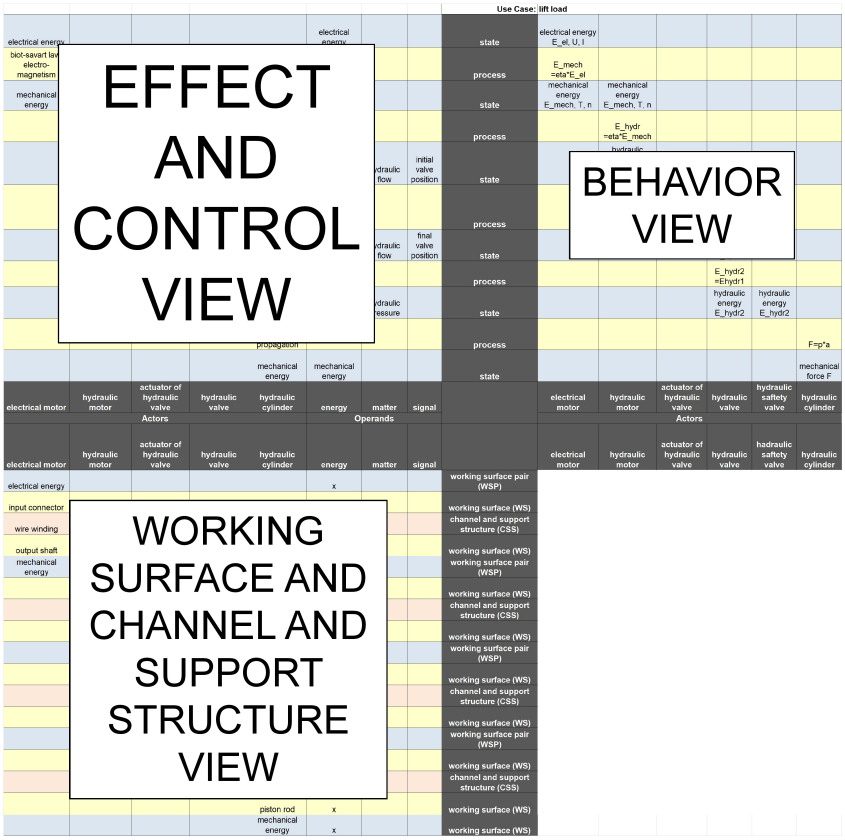


Figure 10. Overview of the structure of the IF4APM.

Three views can be distinguished as follows: the effect and control view, the behavior view, and the working surface, channel, and supports structure view. These views are explained in detail on the basis of a vehicle component example in the next section.

4. Application of the Framework

In this section, the different elements of the IF4APM are explained based on a sub-system development for an agricultural vehicle. The sub-system concerns the actuation of a lifting system (compare Figure 2), and a schematic diagram is given in Figure 11.

A hydraulic pump (which is driven by an electrical motor) is generating a certain hydraulic pressure and, under certain conditions), a hydraulic flow. A so-called 4/3 way (which is electrically actuated) valve is used for defining the motion direction of a double-acting hydraulic cylinder with a piston (up/down/none). This hydraulic cylinder creates motion which leads to the movement of devices on vehicles, e.g., lifting systems. The main physical principles are represented in Figure 12.

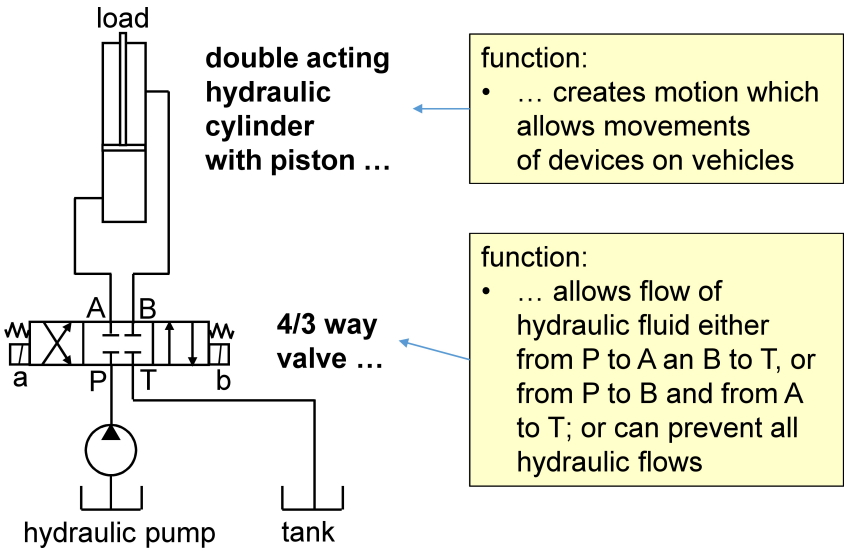


Figure 11. Sample vehicle component: lifting system.

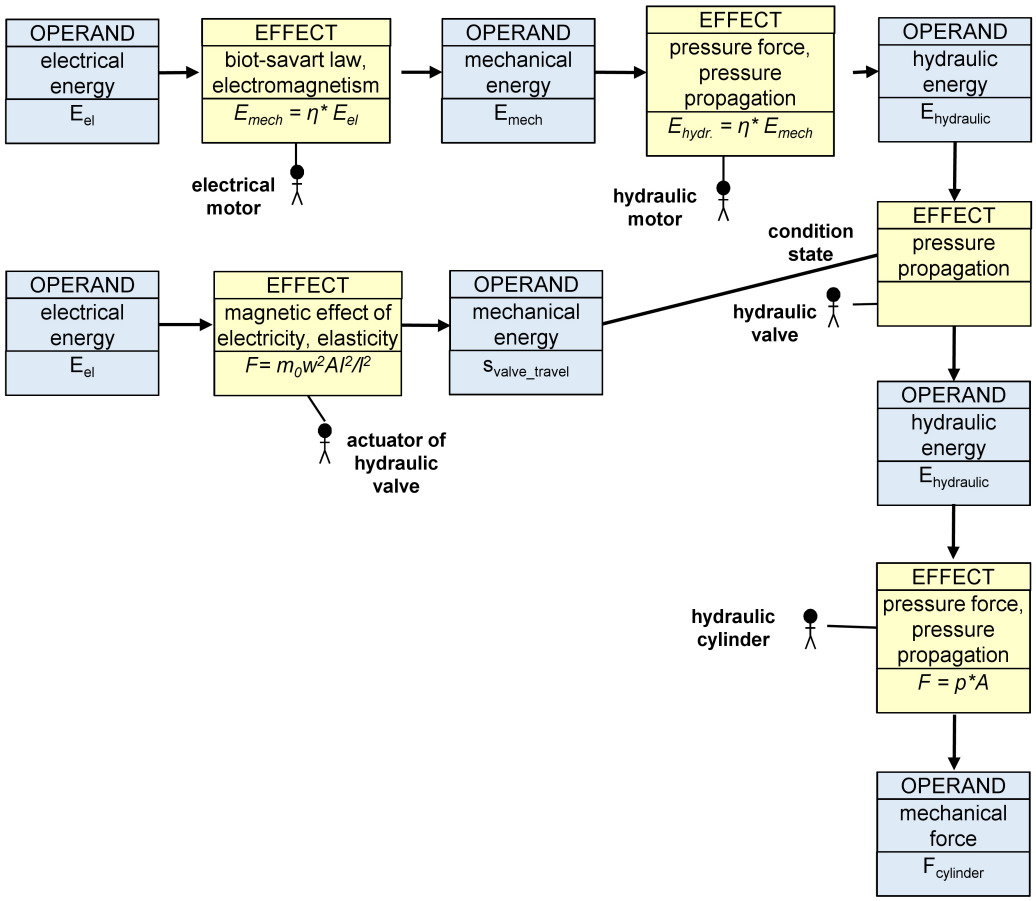


Figure 12. Main physical principles of the lifting system for vehicles.

The blue entities in Figure represent the states of operands, e.g., the electrical energy for an electrical motor. The yellow entities represent the processes and the distinct physical effects which enable these processes. For these processes, the actors, which are the function carriers and allow the physical effects to take place, are also provided in this representation. It is important to point out that it is not sensible in the early phases of design to define these actors; this should be completed during the concretization of the system design. The central view of the IF4APM is the effect and control view; an example is given in Figure 13.

									Use Case: lift load
electrical energy					electrical energy				state
biot-savart law, electro-magnetism					biot-savart law, electro-magnetism				process
mechanical energy	mechanical energy				mechanical energy				state
	pressure force, pressure propagation				pressure force, pressure propagation				process
	hydraulic energy (hydraulic pressure, hydraulic flow)				hydraulic energy	hydraulic flow			state
		electrical energy			electrical energy		initial valve position	actor input state	state (sub-process)
		magnetic effect of electricity, elasticity			magnetic effect of electricity, elasticity		magnetic effect of electricity, elasticity	actor	process
		movement of valve			electrical energy		final valve position	actor output state	condition state
			hydraulic energy		hydraulic energy	hydraulic flow			state
			pressure force, pressure propagation		pressure force, pressure propagation	pressure force, pressure propagation			process
			hydraulic energy		hydraulic energy	hydraulic flow			state
				pressure force, pressure propagation	pressure force, pressure propagation				process
				mechanical energy	mechanical energy				state
electrical motor	hydraulic motor	actuator of hydraulic valve	hydraulic valve	hydraulic cylinder	energy	matter	signal		
Actors					Operands			Control	

Figure 13. Effect and control view.

The effect and control view alternates between states of operands and processes. The initial state of the actor *electrical motor* is *electrical energy*; this motor transforms this energy by applying the physical effects (distinct physical phenomena) *biot-savart law* and *electro-magnetism* to *mechanical energy*, which is the final state of this first process. The operand in this process is *energy*. Additional states are necessary if sub-processes are necessary (this will be explained in detail at the end of this section). In the process lines, the physical effects that enable the process are listed, thus allowing us to describe the physical effect chain that realizes the respective functionality. The columns for actors are intended to be filled in later stages; initially, a solution-independent description of the main effects is possible in the columns for operands. The control column will be explained in detail at the end of this section. It is important to note that the IF4APM does not intend to model all states, properties, aspects, and processes of a vehicle or vehicle component, but only those which are relevant to a certain high level use case (see Figure 13, top line).

In general, a use case describes an application of a technical system for achieving a certain goal and may include different types of users and their activity in and relationship with the system [56]. Use cases can be decomposed into "sub use cases" [57] and summarized into "super use cases". For the IF4APM, a concentration on high level use cases (such as "lift load") (compare Figure 13) is sensible. Use cases allow the creation of IF4APM models as aspect models with a limited size and complexity. It is also important to point out that human beings can also act as actors in this system and can be represented in the IF4APM. They can be part of the respective use case and they can perform certain processes.

The second view of the IF4APM captures the behavior-oriented perspective (Figure 14).

Use Case: lift load					
state	electrical energy E_el, U, I				
process	E_mech =eta*E_el				
state	mechanical energy E_mech, T, n	mechanical energy E_mech, T, n			
process		E_hydr =eta*E_mech			
state		hydraulic energy E_hydr, V_dot, p			
state (sub-process)			electrical energy		
process			complex relationship (FMU)		
condition state			movement of valve		
state				hydraulic energy E_hydr1	
process				E_hydr2 =Ehydr1	
state				hydraulic energy E_hydr2	hydraulic energy E_hydr2
process					F=p*a
state					mechanical force F
	electrical motor	**hydraulic motor**	**actuator of hydraulic valve**	**hydraulic valve**	**hydraulic cylinder**
	Actors				

Figure 14. Behavior view.

Similar to the effect and control view, the behavior view alternates between states of operands and processes. The state *electrical energy* is expressed e.g., by means of an energy quantity parameter *E_el*. The simplified calculation also used in Figure 8 allows us to calculate the output parameter *E_mech* based on the energy efficiency parameter *eta* of the motor. In the process lines, the behavior is described either by means of equations or by links to external simulations (e.g., in form of FMUs compare Section 2). It is important to note that links to other software packages, simulation platforms, and co-simulation

frameworks, such as Matlab/Simulink [58], Modelica [59], and INTO-CPS [60], is also possible. The behavior entities intend to describe the abstract physical behavior that realizes the respective functionality. Only columns for actors are included because this view can usually only be sensibly filled when many properties of the product (such as the choice of actors) are already decided.

The structure of the third view of the IF4APM—the working surface and the channel and supports structure view—is different and is shown in Figure 15.

Actors					Operands			
electrical motor	hydraulic motor	actuator of hydraulic valve	hydraulic valve	hydraulic cylinder	energy	matter	signal	
electrical energy					x			working surface pair (WSP)
input connector					x			working surface (WS)
wire winding					x			channel and support
output shaft					x			working surface (WS)
mechanical energy	mechanical energy				x			working surface pair (WSP)
	input shaft				x			working surface (WS)
	rotor				x			channel and support
	output connector				x			working surface (WS)
	hydraulic energy		hydraulic energy		x	x		working surface pair (WSP)
			input connector		x	x		working surface (WS)
			cavitiy		x	x		channel and support
			output connector		x	x		working surface (WS)
			hydraulic energy	hydraulic energy	x	x		working surface pair (WSP)
				input connector	x	x		working surface (WS)
				piston	x	x		channel and support
				piston rod	x			working surface (WS)
				mechanical energy	x			working surface pair (WSP)

Figure 15. Working surface and channel and supports structure view.

The working surface and the channel and supports structure view starts with working surface pairs (WSPs) and then continues to distinct working surfaces (WSs), then listing channel and support structures (CSS) and continuing again with working surfaces (WSs) and working surface pairs. A WSP with an external source *electrical energy* brings this energy to the WS *input connector*. The CSS is the *wire winding* that represents the abstract physics of the electrical motor. The next WS is the *output shaft* which is connected to the WS *input shaft* of the hydraulic motor via the WSP *mechanical energy*. The working surface and the channel and supports structure view allow a step-wise and connected discussion of the acting physics on a detailed micro level.

As pointed out above, the control aspect of the effect and control view has not yet been discussed in detail. In order to explain the role of control functionalities in detail, an example of a controlled system will be employed. An effect chain enabling the generation of a controlled hydraulic pressure is shown in Figure 16.

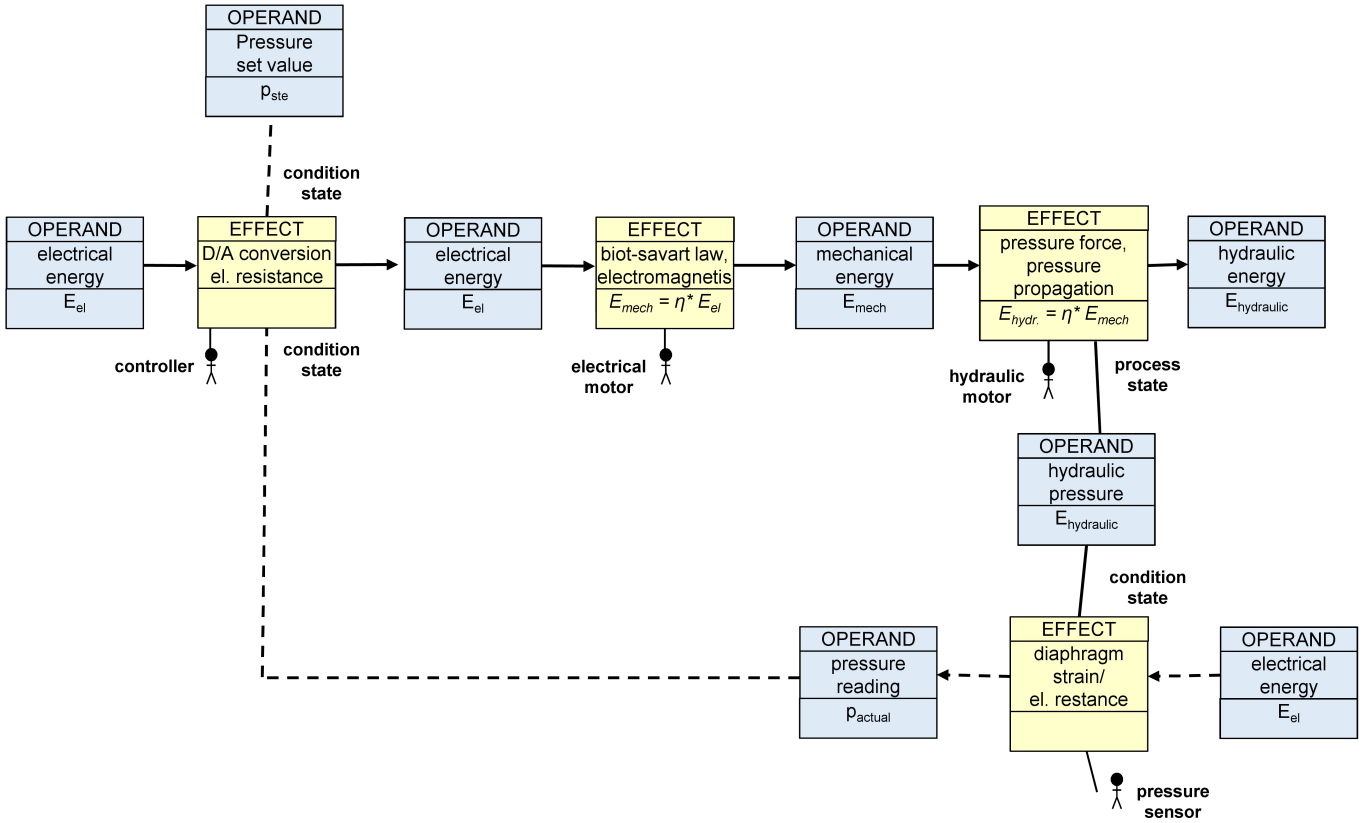


Figure 16. Physical effect chain for the generation of a controlled hydraulic pressure.

In this effect chain, the first actor is the controller of the control loop that compares the actual pressure reading with the set value and controls the flow of electric energy to an electrical motor (second actor). This motor drives an hydraulic motor which delivers the intended hydraulic pressure. For a clear and sensible representation of control functionalities, process and condition states can be employed. The connection types "condition state" (C) and "process state" (P) can be used to connect the auxiliary flows of operands to the main flow of operands [33,61]. The condition state is a prerequisite for the process to take place (e.g., the controller can only regulate the electrical flow if it has information concerning the current, actual pressure, and intended set value). A process state indicates that the state of one operand in the process is used for a second purpose; this link type allows us to assign diagnostic functions to flows of operands undergoing some kind of operation. The use of such link types allows us to clarify complex connections in functional models [61], but also in physical effect chains. The measured pressure is a process state of the main flow. This same state is used in a sub-process as a condition state in order to obtain an electrical voltage correlating with the pressure. This condition state can be used by the controller together with a condition state set value in order to realize the intended control loop. In Figure 17, how this control effect chain can be represented in the effects and control view is shown.

The set value and the measured value can be indicated in the given fields, thus representing and connecting the control effect chain. The condition states *control: measured value* and *control: set value* are a prerequisite for the controller to carry out its intended task *control output pressure*. One of the states *control: measured value* is also visible in the last line of the effect and control view; this clarifies the cyclic nature of control. The effect and control view allows an interlinked discussion of the acting physical effects and their contribution to the establishment of control loops that can guarantee a stable behavior.

							Use Case: hydr. pressure	
electrical energy				electrical energy				state
actual pressure						actual pressure	control: measured value	condition state
desired pressure						desired pressure	control: set value	condition state
D/A conversion el. Resistance							control output pressure	process
electrical energy	electrical energy		mechanical energy	mechanical energy				state
	biot-savart law, electro-magnetism		pressure force, pressure propagation					process
	mechanical energy	mechanical energy	hydraulic energy	hydraulic energy	hydraulic flow			state
		pressure force, pressure propagation						process
		hydraulic energy		hydraulic energy	hydraulic flow			state
		hydraulic pressure	hydraulic pressure			hydraulic pressure		proess state
			diaphragm strain/ el. restance					process
			actual pressure			actual pressure	control: measured value	state
controller	electrical motor	hydraulic motor	pressure sensor	energy	matter	signal		
Actors				Operands			Control	

Figure 17. Representation of a control effect chain in the effect and control view.

5. Summary and Outlook

In the center of this publication is the presentation and explanation of the integrated framework for abstract physics modeling (IF4APM) in the scope of vehicle design. Current engineers working in vehicle design do not dispose of appropriate means to connect the function models of their vehicle components with the detailed geometry, material, and structure models. The lack of such possibilities can lead to misunderstandings, less-than-optimal knowledge management, inconsistencies, as well as isolated and incomplete simulations of the behavior of the vehicle. The presented research aims to provide engineers in vehicle design with a framework which allows the description and investigation of the physical effect chains linking functionality with detailed geometry, material, and structure. This framework intends to increase the transparency of the inner processes of a vehicle design and of the design decisions. The distinctive quality of the proposed approach is to consistently connect the product models on the levels function and detailed geometry/material in the scope of mechanical and mechatronic engineering; the specific realization of the framework leads to the aptitude being used in the early stages of the product development process, as well as to enhanced transparency and easy understandability. The knowledge that is usually captured in the minds of the engineers can be represented and stored; it can be available for further analysis, e.g., by means of big data analyses of artificial intelligence (AI). The ultimate vision concerning the application of AI for vehicle design could be that all models are available in a semantically rich representation (e.g., with all important attributes, annotations, and relationships, as well as all requirements and environmental

conditions) for an AI system to learn how vehicles are developed. From this vision, the main road blocks become apparent. Future research needs to develop even richer and more interconnected models of the technical systems in a form that is completely accessible to AI. The second road block is the development of AI approaches that are powerful enough to deal with the resulting abundance of data. The third road block concerns the trustability of the results; future AI approaches not only need to deliver the design, but also a concise documentation how the design was derived and how high the levels of safety and reliability are. The basis for the presented research are design processes of remote vehicles and vehicle components; these processes also serve as a basis for the explanation of the framework. Up to now, the framework was tested using several examples; the investigation of the general applicability will be the focus of further research. One future research field is the investigation of the possibility to include the logic perspective into this framework; this could be fruitful for a better connection of reliability and safety analyses with the physical effect chains present in the vehicle. A further promising research field is a sensible integration into engineering frameworks for vehicle design, which can be based on the unified modeling language (UML), the system modeling language (SysML), graph-based design languages (GBDLs), and standards such as AutomationML.

Funding: Parts of the described research were funded in the scope of the project "Automatisierter Entwurf eines geometrischen und kinetischen digitalen Zwillings einer Rohbaufertigungsanlage für die Virtuelle Inbetriebnahme (TWIN)", which is funded by the German Federal Ministry of Education and Research. Parts of the described research were funded by the Carl Zeiss Foundation in the scope of the project AI-based digital twin (KI-basierter digitaler Zwilling—KIDZ).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CAD	Computer-Aided Design
CSS	Channel and Support Structure
FMI	Functional Mock-up Interface
FMU	Functional Mock-up
GBDL	Graph-based Design Language
IF4APM	Integrated Framework for Abstract Physics Modeling
IFM	Integrated Function Modeling
MBSE	Model-based Systems Engineering
SysML	Systems Modeling Language
UML	Unified Modeling Language
WS	Working Surface
WSP	Working Surface Pair

References

1. Brunner, H.; Rossbacher, P.; Hirz, M. Sustainable product development: Provision of information in early automotive engineering phases. *Teh. Glas.* **2017**, *11*, 29–34.
2. Demirel, H.O.; Goldstein, M.H.; Li, X.; Sha, Z. Human-centered generative design framework: An early design framework to support concept creation and evaluation. *Int. J. Hum.-Interact.* **2024**, *40*, 933–944. [CrossRef]
3. Pahl, G.; Beitz, W. *Engineering Design: A Systematic Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
4. Ponn, J.; Lindemann, U. *Konzeptentwicklung und Gestaltung Technischer Produkte*; Springer: Berlin/Heidelberg, Germany, 2011.
5. Stetter, R. Fault-Tolerant Design and Control of Automated Vehicles and Processes. In *Insights for the Synthesis of Intelligent Systems*; Springer: Cham, Switzerland, 2020.
6. VDI/VDE. VDI 2221: Design of Technical Products and Systems. In *Model of Product Design*; Beuth: Berlin, Germany, 2019.
7. Holder, K.; Zech, A.; Ramsaier, M.; Stetter, R.; Niedermeier, H.P.; Rudolph, S.; Till, M. Model-Based Requirements Management in Gear Systems Design Based on Graph-Based Design Languages. *Appl. Sci.* **2017**, *7*, 1112. [CrossRef]

8. Malcher, P.; Viana, D.; Antonino, P.O.; dos Santos, R.P. Investigating Open Innovation Practices to Support Requirements Management in Software Ecosystems. In *Proceedings of the International Conference on Software Business*; Springer Nature: Cham, Switzerland, 2023; pp. 35–50.
9. Serna, A. *Requirements Engineering*; Editorial Instituto Antioqueño de Investigación: Medellín, Antioquia 2024.
10. Kosse, S.; Vogt, O.; Wolf, M.; König, M.; Gerhard, D. Requirements management for flow production of precast concrete modules. In *Proceedings of the International Conference on Computing in Civil and Building Engineering*; Springer: Cham, Switzerland, 2022; pp. 687–701.
11. Eisenbart, B.; Gericke, K.; Blessing, L.; McAloone, T. A DSM-based framework for integrated function modelling: Concept, application and evaluation. *Res. Eng. Des.* **2016**, *28*, 25–41. [CrossRef]
12. Gericke, K.; Eisenbart, B. The integrated function modeling framework and its relation to function structures. *AI EDAM* **2017**, *31*, 436–457. [CrossRef]
13. Elwert, M.; Ramsaier, M.; Eisenbart, B.; Stetter, R. Holistic Digital Function Modelling with Graph-Based Design Languages. In Proceedings of the 22nd International Conference on Engineering Design (ICED 19), Delft, The Netherlands, 5–8 August 2019; Volume 1.
14. Drave, I.; Rumpe, B.; Wortmann, A.; Berroth, J.; Hoepfner, G.; Jacobs, G.; Spuetz, K.; Zerwas, T.; Guist, C.; Kohl, J. Modeling mechanical functional architectures in SysML. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Virtual, 16–23 October 2020; pp. 79–89.
15. Wyrwich, C.; Boelsen, K.; Jacobs, G.; Zerwas, T.; Höpfner, G.; Konrad, C.; Berroth, J. Seamless function-oriented mechanical system architectures and models. *Eng* **2024**, *5*, 301–318. [CrossRef]
16. Krüger, M.F.; Zorn, S.; Gericke, K. Combining function modelling and requirements modelling with the ifm framework. *Proc. Des. Soc.* **2023**, *3*, 987–996. [CrossRef]
17. Albers, A.; Wintergerst, E. The Contact and Channel Approach: Relating a system's physical structure to its functionality. In *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*; Springer: London, UK, 2014; pp. 61–72.
18. Hoepfner, G.; Nachmann, I.; Zerwas, T.; Berroth, J.K.; Kohl, J.; Guist, C.; Rumpe, B.; Jacobs, G. Towards a Holistic and Functional Model-Based Design Method for Mechatronic Cyber-Physical Systems. *J. Comput. Inf. Sci. Eng.* **2023**, *23*, 051001. [CrossRef]
19. Zerwas, T.; Jacobs, G.; Spütz, K.; Höpfner, G.; Drave, I.; Berroth, J.; Guist, C.; Konrad, C.; Rumpe, B.; Kohl, J. Mechanical concept development using principle solution models. In *Proceedings of the IOP conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2021; Volume 1097, p. 012001.
20. Stetter, R.; Niedermeier, M. Approaches towards lean products. In Proceedings of the Knowledge, Innovation and Sustainability. Proceedings of the 16th International Conference on Engineering Design, Paris, France, 28–30 August 2007; Design Society: Paris, France, 2007.
21. Stetter, R.; Till, M. A Concept for an Integrated Framework for Abstract Physics Modelling (IF4APM). *Procedia CIRP* **2024**, *120* .
22. Vitale, G.; Hollander, M. The Software-Defined Vehicle: How to Verify and Validate Software Functions. In *Proceedings of the International Stuttgart Symposium*; Springer: Wiesbaden, German, 2023; pp. 399–405.
23. Cederblahd, J.; Cicchetti, A.; Suryadevara, J. Early Validation and Verification of System Behaviour in Model-based Systems Engineering: A Systematic. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 1–67. [CrossRef]
24. Kraft, J. *Entwicklung Eines Universellen Schwerlast-Roboter-Fahrzeugs*; Project Report; Ravensburg-Weingarten University: Weingarten, Germany, 2020.
25. Saiger, C. *Hydraulisches Hubgerüst als Anbaugerät für Einen Kleintraktor*; Project Report; Ravensburg-Weingarten University: Weingarten, Germany, 2018.
26. Reinprecht, M. Entwicklung Eines Heckkrafthebers mit dem Schwerpunkt Fehlertoleranz. Bachelor's Thesis, Ravensburg-Weingarten University, Weingarten, Germany, 2020.
27. Mario, H.; Dietrich, W.; Gfrerrer, A.; Lang, J. *Integrated Computer-Aided Design in Automotive Development: Development Processes, Geometric Fundamentals, Methods of CAD, Knowledge-Based Engineering Data Management*; Springer: Berlin/Heidelberg, Germany, 2013.
28. Eigner, M.; Ernst, J.; Roubanov, D.; Dickopf, T. An initial approach for the application of product assembly information in the early phases of the product development process by using methods of Model Based Systems Engineering. In Proceedings of the DS 81: Proceedings of NordDesign 2014, Espoo, Finland, 27–29 August 2014.
29. Henderson, K.; Salado, A. Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. *Syst. Eng.* **2021**, *24*, 51–66. [CrossRef]
30. Graessler, I.; Wiechel, D.; Oezcan, D.; Taplick, P. Tailored metrics for assessing the quality of MBSE models. *Proc. Des. Soc.* **2024**, *4*, 2545–2554. [CrossRef]
31. Stetter, R. Approaches for Modelling the Physical Behavior of Technical Systems on the Example of Wind Turbines. *Energies* **2020**, *13*, 2087. [CrossRef]
32. Koller, R.; Kastrup, N. *Prinziplösungen zur Konstruktion Technischer Produkte*; Springer: Berlin/Heidelberg, Germany, 2013.
33. Ehrlenspiel, K.; Meerkamm, H. *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit, 6, Vollständig überarbeitete und Erweiterte Auflage*; Carl Hanser Verlag GmbH Co KG: Munich, Germany, 2017.
34. Lunze, J. *Regelungstechnik 1*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 10.

35. Witczak, M. *Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems*; Lecture Notes in Electrical Engineering; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; Volume 266, p. 229.
36. Blanke, M.; Kinnaert, M.; Lunze, J.; Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*; Springer: New York, NY, USA, 2016.
37. Tang, C.; Khajepour, A. Wheel modules with distributed controllers: A multi-agent approach to vehicular control. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10879–10888. [CrossRef]
38. Tang, C.; Khajepour, A. Agent-based model predictive controller (AMPC) for flexible and efficient vehicular control. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9877–9885. [CrossRef]
39. Liang, J.; Lu, Y.; Yin, G.; Fang, Z.; Zhuang, W.; Ren, Y.; Xu, L.; Li, Y. A distributed integrated control architecture of AFS and DYC based on MAS for distributed drive electric vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5565–5577. [CrossRef]
40. Liang, J.; Lu, Y.; Wang, F.; Yin, G.; Zhu, X.; Li, Y. A robust dynamic game-based control framework for integrated torque vectoring and active front-wheel steering system. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7328–7341. [CrossRef]
41. Albers, A.; Matthiesen, S.; Lechner, G. Konstruktionsmethodisches grundmodell zum zusammenhang von gestalt und funktion technischer systeme. *Konstruktion (1981)* **2002**, *7–8*, 55–60.
42. Albers, A.; Braun, A.; Sadowski, E.; Wyatt, D.F.; Wynn, D.C.; Clarkson, P.J. Contact and Channel Modelling Using Part and Function Libraries in a Function-Based Design Approach. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, QC, Canada, 15–18 August 2010; Volume 44137, pp. 393–404.
43. Albers, A.; Zingel, C. Extending SysML for engineering designers by integration of the contact & channel–approach (C&C2-A) for function-based modeling of technical systems. *Procedia Comput. Sci.* **2013**, *16*, 353–362.
44. Grauberger, P.; Bremer, F.; Sturm, C.; Hoelz, K.; Wessels, H.; Gwosch, T.; Wagner, R.; Lanza, G.; Albers, A.; Matthiesen, S. Qualitative Modelling in Embodiment Design-Investigating the Contact and Channel Approach through Analysis of Projects. In *Proceedings of the Design Society: DESIGN Conference*; Cambridge University Press: Cambridge, UK, 2020; Volume 1, pp. 897–906.
45. Freudenmann, T. *Ontologien zur Validierung von Produkten Basierend auf dem Contact & Channel-Ansatz (C&C^2-Ansatz)= Ontologies for the Validation of Products Based on the Contact & Channel Approach (C&C^2-Approach)*; IPEK: Karlsruhe, Germany 2014.
46. Gero, J.S.; Kannengiesser, U. The function-behaviour-structure ontology of design. In *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 263–283.
47. Gero, J.; Milovanovic, J. The situated function-behavior-structure co-design model. *CoDesign* **2021**, *17*, 211–236. [CrossRef]
48. Russo, D.; Spreafico, C. Investigating the multilevel logic in design solutions: A Function Behaviour Structure (FBS) analysis. *Int. J. Interact. Des. Manuf.* **2023**, *17*, 1789–1805. [CrossRef]
49. Sanderson, D.; Chaplin, J.C.; Ratchev, S. A Function-Behaviour-Structure design methodology for adaptive production systems. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 3731–3742. [CrossRef]
50. Neema, H.; Gohl, J.; Lattmann, Z.; Sztipanovits, J.; Karsai, G.; Neema, S.; Bapty, T.; Batteh, J.; Tummescheit, H.; Sureshkumar, C. Model-based integration platform for FMI co-simulation and heterogeneous simulations of cyber-physical systems. In *Proceedings of the 10th International Modelica Conference*; Linköping University Electronic Press: Linköping, Sweden, 2014; Volume 96, pp. 235–245.
51. Larsen, P.G.; Fitzgerald, J.; Woodcock, J.; Fritzson, P.; Brauer, J.; Kleijn, C.; Lecomte, T.; Pfeil, M.; Green, O.; Basagiannis, S.; et al. Integrated tool chain for model-based design of Cyber-Physical Systems: The INTO-CPS project. In Proceedings of the 2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data), Vienna, Austria, 11 April 2016; pp. 1–6.
52. Miller, M.; Pfeil, M.; Kennel, R. *Trailer Electrification—A HIL Approach for MPC Powertrain Control to Ensure Driver Safety in Micromobility*; Technical Report; SAE Technical Paper: Warrendale, PA, USA 2023.
53. Saft, P.; Pfeil, M.; Stetter, R.; Till, M.; Rudolph, S. Integration of geometry modelling and behavior simulation based on graph-based design languages and functional mockup units. *Procedia CIRP* **2024**, *120* .
54. Beernaert, T.; Verlaan, A.; Etman, P.; Giesen, P.; Van Beekum, E.; Ribeiro, M.; Bola, I.; Moser, L.; De Bock, M.; Classen, I.; et al. From Physics to Project Management: A Multi-Domain Matrix Model for the Distributed Analysis of Nuclear Fusion Systems. In Proceedings of the DS 126: Proceedings of the 25th International DSM Conference (DSM 2023), Gothenburg, Sweden, 3–5 October 2023; pp. 29–38.
55. Elwert, M.; Ramsaier, M.; Eisenbart, B.; Stetter, R.; Till, M.; Rudolph, S. Digital Function Modeling in Graph-Based Design Languages. *Appl. Sci.* **2022**, *12*, 5301. [CrossRef]
56. Scalice, R.K.; Berkenbrock, G.R.; Mendoza, Y.E.A. Use case based methodology for conceptual design of industrial mechatronic products. In Proceedings of the DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, BC, Canada, 21–25 August 2017; pp. 347–356.
57. Eisenbart, B.; Gericke, K.; Blessing, L. Application of the IFM framework for modelling and analysing system functionality. In Proceedings of the DS 77: Proceedings of the DESIGN 2014 13th International Design Conference, Cavtat, Croatia, 19–22 May 2014; pp. 153–162.
58. The MathWorks Inc.: Natick, MA, USA. Available online: https://mathworks.com/ (accessed on 31 July 2024).
59. The Modelica Association: Linköping, Sveden. Available online: https://modelica.org/ (accessed on 31 July 2024).

60. INTO-CPS Association: Aarhus, Denmark. Available online: https://into-cps.org/ (accessed on 31 July 2024).
61. Stetter, R.; Simundsson, A. Design for control. In Proceedings of the DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, BC, Canada, 21–25 August 2017; pp. 149–158.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.