Modeling Languages for Automotive Digital Twins

A Survey Among the German Automotive Industry

Jérôme Pfeiffer
University of Stuttgart
Stuttgart, Germany
jerome.pfeiffer@isw.uni-stuttgart.de

Dominik Fuchß
Karlsruhe Institute of Technology
Karlsruhe, Germany
dominik.fuchss@kit.edu

Thomas Kühn
Martin Luther University
Halle-Wittenberg
Halle, Germany
thomas.kuehn@informatik.uni-halle.de

Robin Liebhart
University of Stuttgart
Stuttgart, Germany
st166871@stud.uni-stuttgart.de

Dirk Neumann
Karlsruhe Institute of Technology
Karlsruhe, Germany
dirkneumann98@gmail.com

Christer Neimöck
T-Systems GmbH
Frankfurt, Germany
C.Neimoeck@t-systems.com

Christian Seiler
Mercedes-Benz AG
Sindelfingen, Germany
christian.seiler@mercedes-benz.de

Anne Koziolek
Karlsruhe Institute of Technology
Karlsruhe, Germany
koziolek@kit.edu

Andreas Wortmann
University of Stuttgart
Stuttgart, Germany
wortmann@isw.uni-stuttgart.de

ABSTRACT

The demand for digital twins and suitable modeling techniques in the automotive industry increases rapidly. Yet, there is no common understanding of digital twins in automotive, nor are there modeling techniques established to create automotive digital twins. Recent studies on digital twins focus on the analysis of the literature on digital twins for automotive or in general and, thus, neglect the industrial perspective of automotive practitioners. To mitigate this gap between scientific literature and the industrial perspective, we conducted a questionnaire survey among experts in the German automotive industry to identify i) the desired purposes for and capabilities of digital twins and ii) the modeling techniques related to engineering and operating digital twins across the phases of automotive development. To this end, we contacted 189 members of the Software-Defined Car research project and received 96 responses. The results show that digital twins are considered most useful in the usage and support phase of automotive development, representing vehicles as-operated. Moreover, simulation models, source code, and business process models are currently considered as the most important models to be integrated into a digital twin alongside the associated, established tools.

KEYWORDS

modeling languages, digital twins, automotive, survey

MODELS '24, September 22–27, 2024, Linz, Austria

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Jérôme Pfeiffer, Dominik Fuchß, Thomas Kühn, Robin Liebhart, Dirk Neumann, Christer Neimöck, Christian Seiler, Anne Koziolek, and Andreas Wortmann. 2024. Modeling Languages for Automotive Digital Twins: A Survey Among the German Automotive Industry. In *Linz '24: ACM Conference on Model Driven Engineering Languages and Systems, September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Digital Twins (DTs) [25, 40] are complex software systems that connect to an actual system, e.g., a (simulated) vehicle, obtain data from it, represent it based on this data, and use this data to optimize its behavior [13, 15, 22]. They are used in a variety of domains, including agriculture [3], manufacturing [8], medicine [27], and many more [11]. Typical purposes of digital twins include monitoring [31], validating [20], predicting [34], and optimizing the behavior of the actual system. For this, DTs require different properties [13], such as collecting data, model processing, simulation, bi-directional communication, and more. At the center of digital twins are models, representing the actual system, and data, obtained from it, to which the models give meaning. Hence, digital twins are model-centric software systems and modeling techniques shape the capabilities and properties digital twins can express and reason about. This is also reflected by large software vendors, such as Microsoft or Amazon, as well as the Eclipse Foundation that actively develop dedicated modeling languages for DTs [28, 33]. While the debate on the nature and components of digital twins is ongoing, the question of whether different domains require conceptually and methodically different kinds of digital twins, related to their modeling technologies and modeling languages, is looming as well.

Related studies collected various definitions of DTs across domains from the literature [11], and investigated how DTs can be engineered for the production of automotive and what their added-value could be [7]. While other related studies focusing on modeling languages [44] and model-based engineering [29] exist, they either only query existing literature or are tailored to a certain domain within the automotive development life cycle, e.g., to embedded systems [2, 29]. A study surveying automotive industry experts on the expectations, use, and modeling techniques for DTs in their domain is missing

To remedy this, we investigate the purposes, expected properties, and modeling techniques for DTs in the context of the German research project Software-Defined Car (SofDCar).[1] While SofDCar features German partners only, its consortium includes (i) worldwide leading automotive companies and suppliers, such as Bosch, Mercedes-Benz, ZF Friedrichshafen; (ii) renown software companies working in automotive, including T-Systems International and Vector Informatik, as well as (iii) leading research institutes on software engineering, such as Karlsruhe Institute of Technology, the University of Stuttgart, and the FZI Research Center for Information Technology. With many of the consortium participants contributing to the shape of digital twins in automotive internationally, we expect the insights from this survey to generalize beyond Germany.

To survey the experts in SofDCar, we designed and conducted a questionnaire that inquires about their understanding of DTs and the modeling languages and tools they currently employ in practice. To this end, we answer the following research questions (RQs) based on the responses from our industry experts:

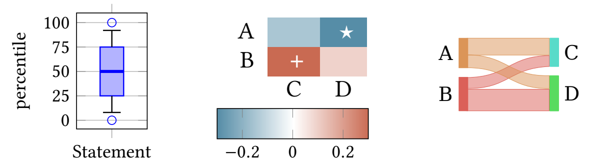
RQ1: How is the DT understood in the automotive industry?
- **RQ1.1:** For which phases of automotive development are DTs important?
- **RQ1.2:** What are desired properties of DTs?
- **RQ1.3:** What are desired purposes of using DTs?
- **RQ1.4:** How do these purposes change in relation to different phases of automotive development?

RQ2: Which modeling languages and modeling tools are currently employed in the automotive industry?
- **RQ2.1:** Which kinds of models are important during automotive development?
- **RQ2.2:** How important are which models in the phases of automotive development?
- **RQ2.3:** Which tools are used to create and maintain these models?

The results of our survey indicate that DTs are especially important when monitoring and collecting data from the vehicle in its operation. Furthermore, at least one kind of modeling language is currently employed by all participants, indicating that leveraging these models in the context of digital twins will be the key to their successful deployment in practice.

In the remainder, Section 2 introduces background information before Section 3 explains the design of our questionnaire survey, and Section 4 presents its results. Afterward, Section 5 discusses the results as well as threats to validity, whereas Section 6 discusses related studies. Finally, Section 7 concludes this paper.

[1] SofDCar website: https://sofdcar.de/language/en/.



(a) Box plot **(b) Heatmap** **(c) Sankey diagram**

Figure 1: Illustrative examples of the employed diagrams in our study.

2 BACKGROUND

2.1 Digital Twins

DTs are software systems comprising data, models, and services to interact with a cyber-physical system (CPS) for a specific purpose [6, 23, 25]. Twinning means that changes to the counterpart are automatically reflected in the DT and changes in the DT are automatically reflected in the counterpart. To this end, DTs must establish a representation of their corresponding system, along with a mechanism for transmitting changes between both systems [14]. Hence, DTs enable a variety of value-adding services, such as monitoring, behavior prediction [24], process optimization [9], predictive maintenance, and more [41, 45].

2.2 Diagrams

For the presentation of our results we utilize a variety of diagrams. Besides histograms and bubble charts, we employ box plots, heatmaps, and Sankey diagrams to visualize the information we gathered. As the latter might be unfamiliar to some readers, Figure 1 illustrates each of them. First, box plots (Figure 1a) visualize the distribution of answers given, especially for the four-point Likert scales. On the x-axis, we show the different properties or statements the participants could rate, and on the y-axis, we show the rating of the participants. The box indicates the range from 75th to 25th percentile, whereas the thick line indicates the median. The whiskers represent the highest and lowest answer in the interquartile range, i.e., 1.5 times the 75th minus the 25th percentile. Any answer that is beyond the interquartile range from the median is considered an outlier, indicated by a circle. Second, heatmaps (Figure 1b) visualize the correlation between different aspects of a dataset. Here, we show the correlation between different participants answering A or B in the first question and C or D in the second question. The x and y-axis show the different options whereas the cell color indicates the correlation coefficient between the two options. A legend indicates which color corresponds to which coefficient. In addition, we indicate a strong significance ($p < 0.05$) of the correlation between B and C with a star in the cell and a weak significance ($p < 0.1$) between A and D with a plus. Last but not least, Sankey diagrams (Figure 1c) visualize participants giving the same answers as a flow from one question to another. Consequently, the width of the flow indicates the number of participants giving the same answers. In Figure 1c, the flow from option A to option C highlights that most participants who chose A also chose B in the second question, whereas only a few opted for D.

3 METHODOLOGY

The creation of the questionnaire started with the selection of our research questions to be answered. They are based on the research gap in current literature where no study exists that surveys experts from the automotive industry on their understanding, expectations, and employed modeling techniques for DTs in automotive development. Next, we created the initial version of the questionnaire based on previous work on DTs [8, 10, 11, 17, 33] and model-driven software engineering [19, 37, 44], and discussions with industry partners within the SofDCar project. After that, we conducted two preliminary studies, to end up with our final questionnaire comprising 54 questions. The average time required to fill out the survey is 15 minutes. Since all members of the project are German, the survey language is German, too. A PDF export of our questionnaire in German as well as all answers can be found online.[2] We anonymized the data to tackle re-identification risk. We created the survey with the tool LimeSurvey[3]. LimeSurvey offers a CSV export that facilitates the import into other tools for data analysis. In particular, we utilized the *Statistical Package for the Social Sciences* (SPSS)[4] and custom Python scripts, for instance, to create the Sanky diagrams.

3.1 Preliminary Studies

We performed two preliminary studies before we came up with our final version of the questionnaire. We handed out the first draft of our questionnaire to institute members and collected feedback. The results showed that open-ended questions for both of our research questions hamper the analyzability of the responses. Therefore, we reworked the questionnaire and introduced Likert scales that allow for better analyzability. In the second preliminary study, we handed out the reworked survey to institute members and selected members of the project consortium. In this iteration, we got the feedback that answering the complete survey takes 30 minutes and thereby too much time. The reasons for that were too many questions for all the different working areas, the utilized kinds of models in these areas and the tools associated with the models. To tackle this concern, the questions for which models are used in which work area and with which tool were changed into dependent questions. Consequently, a participant first chooses his/her work areas and afterward selects the kinds of models used in each selected work area. Likewise, we only ask participants for a selected kind of model to name the tools employed and rate their importance in the different phases of automotive development. In addition, in both preliminary studies, we addressed feedback regarding misleading formulations or spelling errors. To avoid bias, participants in the two preliminary studies were excluded from the final questionnaire.

3.2 Questionnaire Design

The DT in the SofDCar project includes designing a DT information management layer that manages car configurations and information. To formalize this information and variants, models should be used. For this purpose, we designed a questionnaire to get an overview of employed models and associated tools in the consortium. Concerning their employment by DTs, we furthermore, wanted to gain the importance of purposes, properties, and development phases that DTs should fulfill or be used in. Therefore, we split the questionnaire into two parts comprising in total of 54 questions. Since the project consortium consists of experts from different domains, our first question asked for the scientific background of the participant, i.e., cybernetics, business administration, physics, mechatronics, mechanical engineering, electrical engineering, or computer science. The first part of the questionnaire focuses on questions regarding the purposes, properties, and important development phases of DTs, and the second part focuses on modeling techniques and tools.

3.2.1 Properties, Purposes and Phases important for Digital Twins. The first part of the survey started with questions concerning the DT. We asked for the agreement on properties of a DT using a four-point Likert scale ranging from disagree completely (--), rather disagree (-) to rather agree (+), completely agree (++). Participants could propose additional properties in an open text field. After the properties, we asked for the importance of purposes. Here the participants, again, could rate on a four-point Likert scale whether they consider a purpose unimportant (--), rather unimportant (-), rather important (+), or very important (++). The participants could enter additional purposes in a text field. The last question of the first part of the questionnaire was about the importance of the DT in different development phases. Here the same four-point Likert scale as before for importance was used.

3.2.2 Models and Tools in the Automotive Industry. For the second part of the questionnaire, we commenced by asking for the utilization of models in various work areas in automotive development. Here, the participants could select multiple work areas using checkboxes, whereas a text field permitted participants to provide additional work areas. Depending on the selected work area, one additional question asked participants to name the kinds of models they employ in each work area. Again, checkboxes were used, and additional model kinds could be provided in a text field. Depending on the model kind a participant selected, two model-kind-specific questions were asked for each model kind. First, which tools the participant uses to work on this model kind. Second, how important they consider each model kind for the different phases of development, employing the same Likert scale as before ranging from unimportant to very important.

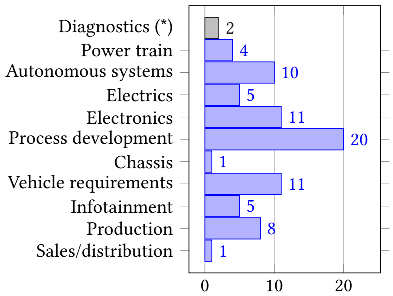
3.3 Selection of Participants

To reach as much participants from diverse work areas as possible, we invited the participants by email via an internal mailing list of 189 members of the SofDCar project. To improve the response rate, we implemented two deadline extensions from the initial one-month-long time frame with 2 weeks of additional response time. Together with the deadline extension, we sent a mail to inform and remind the members of the consortium of the survey.

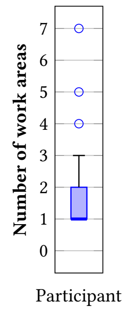
4 RESULTS

This section presents the results of our survey. In total, we had 96 participants, of which 43 completed the questionnaire. Figure 3 shows the scientific background of all participants of our survey. For

[2] Replication package: https://github.com/jerome-pfeiffer1/modeling-languages-for-digital-twins/tree/main (Will be uploaded to Zenodo for camera-ready version)
[3] Tool for the questionnaire: https://www.limesurvey.org/
[4] Tool for analysis of participant's answers: https://www.ibm.com/spss



(a) Participants' work areas



(b) Per participant

Figure 2: Number of participants in different work areas (a), where the * indicates answers from the free-form field, and a box plot highlighting the distribution of work areas per participant (b).

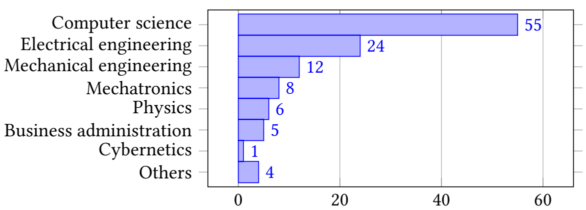


Figure 3: Overview of the professional backgrounds of the participants, whereas one participant could select multiple backgrounds.

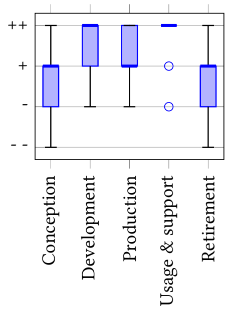
incomplete survey responses, we took only the available answers and did not include the missing answers in our data analysis. The majority of participants were computer scientists (55), followed by electrical (25) and mechanical engineers (11). The participants work in a diverse set of areas, shown in Figure 2a, ranging from *sales and distribution* and *chassis* over *power train* and *infotainment* to *vehicle requirements* and *autonomous systems*. Overall, most participants work on improving the automotive development processes and only work in up to two work areas (cf. Figure 2b). Including automotive diagnostics, named as an additional work area, we reached participants of 11 distinct work areas.

4.1 RQ1: How is the DT understood in the automotive industry?

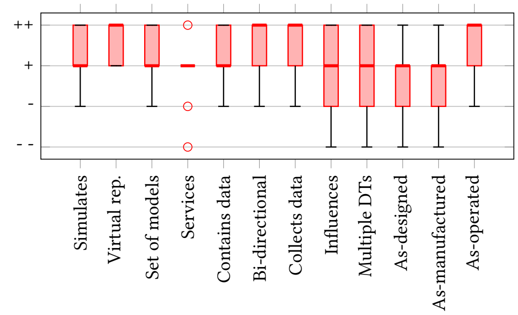
4.1.1 RQ1.1: For which phases of automotive development are DTs important? To answer the question of how important the participants consider DTs in different phases of automotive development, we asked them to rate the relevance of DTs in the concept phase, development phase, production phase, usage & support phase, and retirement phase. The answers ranged from unimportant to very important. The box plot in Figure 4a shows our results. We can see that the DT is considered important in all phases of automotive development. Nevertheless, the results show that it is less important during the concept and retirement phase. Furthermore, it is considered very important in the usage & support phase as well as in the development phase.

4.1.2 RQ1.2: What are desired properties of DTs? To answer the question of which properties a DT should have, we asked the participants to rate the importance of various properties via a Likert scale from unimportant to very important. We selected the following properties based on current literature and discussions with our partners in the project: (1) Simulates: the DT simulates the actual vehicle. (2) Virtual representation: the DT is a virtual representation of the actual vehicle. (3) Set of models: the DT encapsulates a set of models. (4) Services: the DT provides services for the vehicle. (5) Contains data: the DT contains data that is relevant for the vehicle. (6) Bidirectional: a bi-directional connection between the DT and the vehicle. (7) Collects data: the DT can collect current data of the vehicle. (8) Influences: the ability of the DT to influence the vehicle. (9) multiple DTs: one vehicle can have multiple DTs. (10) As-designed: the DT describes the vehicle in its state at design time. (11) As-manufactured: the DT describes how the vehicle is manufactured. (12) As-operated: the DT describes how the vehicle behaves during runtime. The box plot in Figure 4b shows the results regarding the different possible properties. We can see that the participants consider all properties as rather important or very important (median). The results show that the properties of virtual representation, as-operated, collects data, and bi-directional synchronization are considered very important.

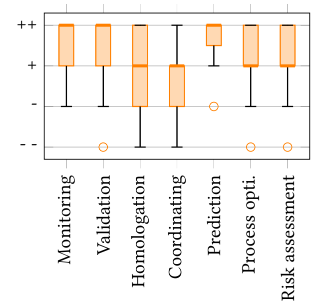
4.1.3 RQ1.3: What are desired purposes of using DTs? DTs are employed for achieving a multitude of purposes. Based on the existing literature and discussion with industry partners in our consortium, we provided the following purposes in our questionnaire: (1) Monitoring: the analysis of runtime and historical data emitted from the vehicle. (2) Validation: the DT can test the qualities and functionalities of the vehicle. (3) Homologation: helps car manufacturers to get admission by authorities for their car, its components and software. (4) Coordinating: the DT can coordinate and control multiple vehicles, e.g., for fleet management. (5) Prediction: the DT can make predictions on the vehicle, e.g., for erroneous behavior or the next maintenance required. (6) Process optimization: the DT can improve development, sales and operational processes. (7) Risk assessment: the DT can point out possible risks depending on the phases of development in automotive. The participants could rate how important they consider the respective purposes based on a Likert scale ranging from unimportant to very important. 1) For monitoring the answers ranged from 2 rather not important to 4 very important. However, the median and the majority of participants considered that this is a very important purpose of a DT. 2) For the purpose of validation, the median and majority considered it very important, too. Only a few participants voted that this validation is not an important purpose of a DT. 3) For the purpose of homologation the answers ranged from not important to very important. The median, however, is rather important. 4) This is true for the purpose of coordination, too. 5) Prediction has the most answers considering it as very important and no answer considering it not important at all. 6) For process optimization, the median of answers considers this purpose as rather important. While most answers were between rather important and very important, some participants replied with not important and rather not important.



(a) Phases of development



(b) Properties



(c) Purposes

Figure 4: Box plots of the perceived importance of DTs in the different phases of development (a), of the properties of DTs (b), and of purposes by employing DTs (c). The importance ranges from *unimportant* (– –), via *rather unimportant* (–) and *rather important* (+) to *very important* (++).

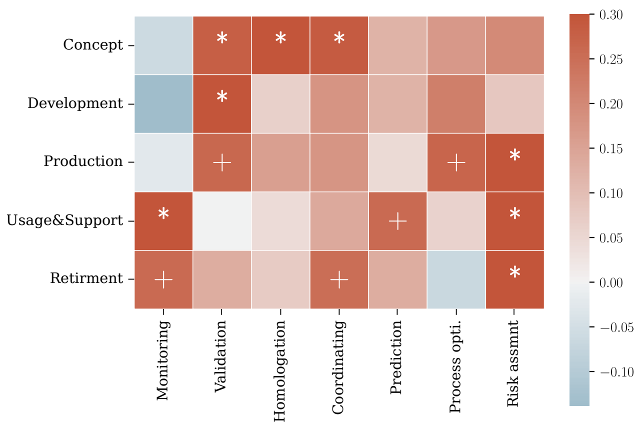


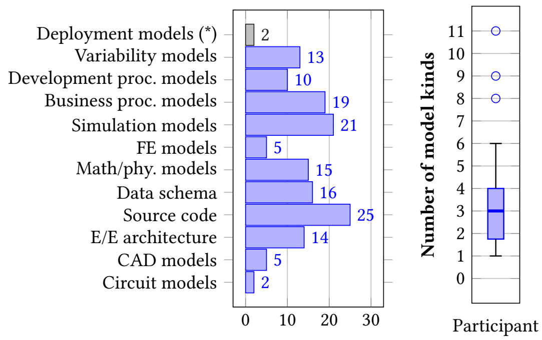
Figure 5: Heatmap of the correlations between phases of development and purposes for DTs. Hue indicates a positive or negative correlation. Strong significance ($p < 0.05$) is indicated with a * and weak significance ($p < 0.10$) with a +.

7) For the purpose risk assessment the results are the same as for process optimization.

4.1.4 RQ1.4: How do these purposes change in relation to different phases of automotive development? To answer the question of which DT purposes are relevant in specific development phases, we related the answers for both questions to one another, resulting in the heatmap depicted in Figure 5. It shows that the purpose of monitoring is especially relevant in the post-production phases of support and retirement. In contrast, validation is important in the pre-operational phases of concept, development and production. Homologation is at all relevant in the concept phase. Coordinating as a purpose of the DT is considered relevant in all development phases but especially in the concept and retirement phases. The purpose prediction is a purpose that is mostly relevant in the usage and support phase. Process optimization is highly relevant to the production process and, thus, there is a strong relation between this purpose and the production phase. Risk assessment has a strong correlation in the phases of production, support, and retirement.

4.2 RQ2: Which modeling languages and modeling tools are currently employed in the automotive industry?

4.2.1 RQ2.1: Which kinds of models are important during automotive development? As we were also interested in the kinds of models and modeling tools currently employed for automotive development, we first asked each participant to pick their work areas. From the provided work areas, participants worked in 10 different areas, i.e., power train, autonomous systems, electrics, electronics, process development, chassis, vehicle requirements, infotainment, production, sales/distribution. Overall participants, all 12 model kinds were stated to be employed within automotive development, as shown in Figure 6a: (1) Variability models: represent all possible variants of a system, usually by means of features and constraints. (2) Development process models: model all aspects of a development process, including project schedule and organization, employed to supervise the development process. (3) Business process models: represent business processes by means of activities, events, and flows; permits their analysis, management, and automation. (4) Simulation models: specify complex systems with continuous and discrete elements, such that the system's behavior can be simulated. (5) Finite element (FE) models: represent the physical shape and properties of objects to apply the finite element method [21] used for structural analysis, heat transfer or body deformation. (6) Mathematical/physical models: capture physical phenomena and dynamic behavior of a vehicle by means of mathematical equations. (7) Data schema: specify the layout and type of data used, stored, and/or shared within a system. (8) Source code: is a textual, human-readable representation of a machine-executable program. (9) Electrical/electronic (E/E) architecture: models the electrical and electronic hardware, the software components, and the wiring of a vehicle. (10) Computer-Aided Design (CAD) models: capture the physical shape and composition of 3D objects. (11) Circuit models: describe an electronic circuit used to design and emulate electronic hardware. In addition to the 11 model kinds we provided, two participants declared *deployment*



(a) Number of different model kinds **(b) Per participant**

Figure 6: Number of different model kinds employed by participants (a), where the * indicates answers from the free-form field, and a box plot highlighting the distribution of model kinds used per participant (b).

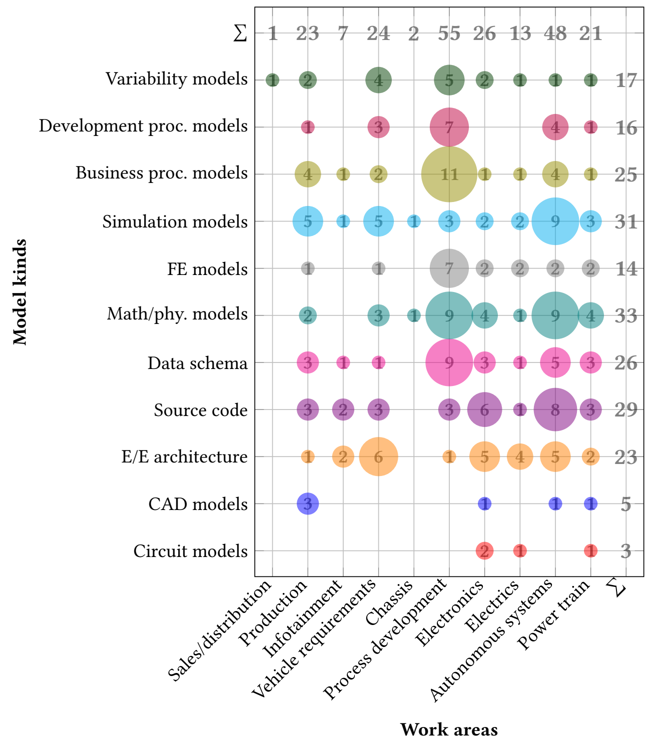


Figure 7: Bubble chart relating the model kind (y-axis) to the various work areas of automotive development (x-axis).

models as an additional model kind that denotes the run-time configuration of software components running on processing nodes used to deploy or reconfigure a vehicle. The most common model kinds were *source code* (25) followed by *simulation models* (21) and *business process models* (19). In contrast, FE models (5) and circuit models (2) were the least common model kinds employed. Moreover, we determined how many different model kinds each participant named. The result is depicted in Figure 6b, whereas the median number of different model kinds per participant is three. Notably, one participant selected all model kinds, which is a clear outlier.

To further distinguish, in which work area which model kind was used, the bubble chart in Figure 7 indicates, which model kinds are employed in which work area. Please note that several participants stated to work in multiple working areas, and, in turn, named the same model kind in different work areas. In the bubble chart, these mentions are counted separately. The data indicates that, except for sales & distribution, each work area employs a wide variety of different model kinds. Surprisingly, we found that the work area of process development also employs FE models and mathematical/physical models.

4.2.2 RQ2.2: How important are which models in the phases of automotive development. To answer how relevant each model kind for each phase of automotive development is, each participant rated the importance of each selected model kind for each of the five phases. The results, summarized in Figure 8, show that all model kinds are considered very or rather important in more than one phase. For the conception phase, E/E architecture models, data schemas and variability models are considered *very important* by most participants. For the development phase, all model kinds are considered *rather important* or *very important* by most participants. For the production phase, only source code, business process models, development process models and variability models are considered as *rather important* by most participants. For the usage& support phase, in addition to the model kinds of the production phase, circuit models, CAD models, data schemas, and math./phy. models are considered *rather important* by most participants. Finally, for the retirement phase, only source code, data schemas, development process models, and variability models are considered *rather important* by most participants. In essence, the participant answers indicate that data schemas, business process models, development process models, and variability models are considered *very important* or *rather important* throughout all phases of development when considering the median importance. In contrast, the median importance of FE models and math./phy. models varies per phase. To illustrate the variate of model kinds considered important in each phase, Figure 9a relates all development phases to the model kinds whereas the width of the line corresponds to the number participants considering the model kind as either *rather* or *very important.* In sum, all model kinds are considered important in each of the five development phases. This is true for CAD models and FE models despite the limited number of participants using them.

4.2.3 RQ2.3: Which tools are used to create and maintain these models? In addition to the importance of model kinds, participants were asked to name the corresponding modeling tools or development environments. In total, we collected individual 152 mentions of tools by participants resulting in 32 distinct modeling tools and development environments. Please note, many participants stated multiple tools for each model kind. The most common tools were MatLab Simulink for simulation models (15), Enterprise Architect for data schemas (14), Visual Studio Code for Source Code (14), and MatLab for math./phy. models (14). To highlight the tools used

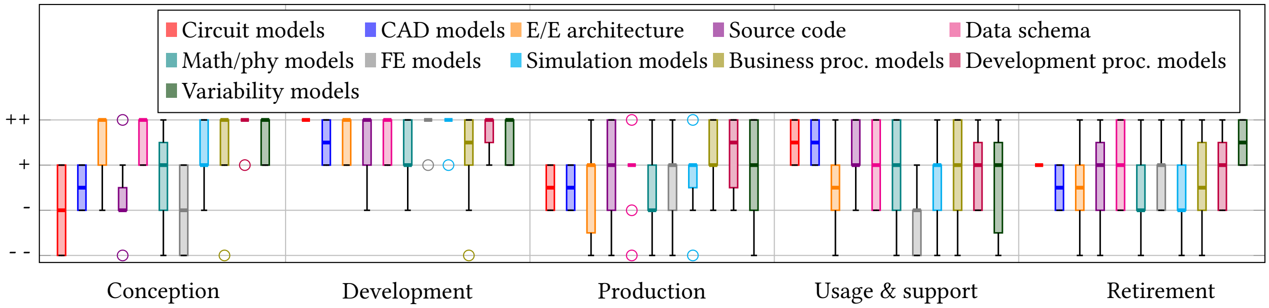


Figure 8: Box plot of the importance of each model kind for the five phases of automotive development. The importance ranges from *unimportant* (– –), via *rather unimportant* (–) and *rather important* (+) to *very important* (++). Number of answers per model kind differs, as participants could select to employ multiple model kinds.

in the automotive domain, Figure 9b shows the relation between model kinds and stated tools. Again, the width of the lines denotes the number of mentions. Notably, though, a plethora of tools are currently in use whereas many only focus on a specific model kind. Only four tools support multiple kinds of models, i.e., MagicDraw, Enterprise Architect, PREEvision and EPLAN.

5 DISCUSSION

Moreover, we discuss how we avoided threats to validity.

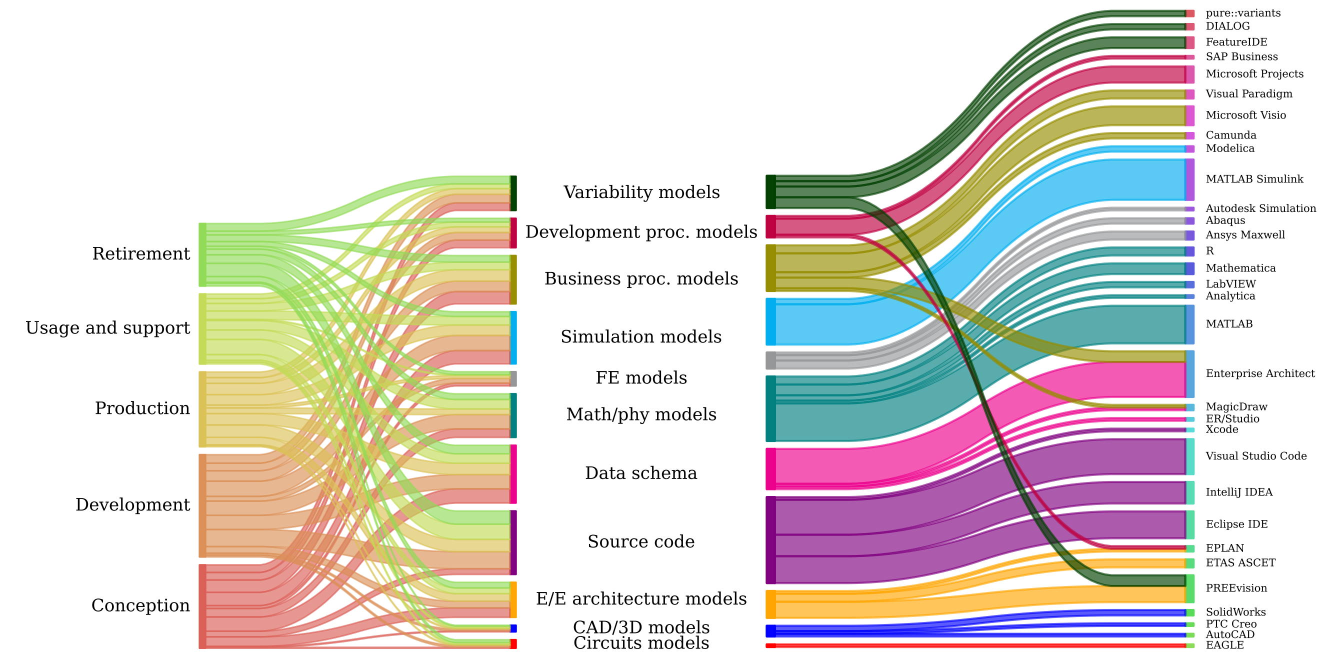
5.1 RQ1: How is the DT understood in the automotive industry?

5.1.1 RQ1.1: For which phases of automotive development are DTs important? Concerning the relevance of DTs in different phases of automotive development (cf. Figure 4a) our participants from the automotive domain agreed that the DT is relevant in all phases of development. Most interestingly, the DT is considered very important in the *usage & support* phase. This indicates that the DT is considered to be most useful when the vehicle is already in use. This might be related to the DT's ability to bridge the gap between the physical and the digital world and to provide insights into the physical system's behavior and performance. The second most important phase is the *development* phase, where the DT could prevent early errors and provide insights on the vehicle that would otherwise be only available after production. For instance, if the DT of the vehicle comprises a virtual representation, this could be used to simulate certain vehicle functionalities where otherwise, a physical prototype would be necessary in this phase of development.

5.1.2 RQ1.2: What are desired properties of DTs? Concerning the properties of DTs (cf. Figure 4b) our participants from the automotive domain agreed on all provided properties. Nevertheless, it is important to see the differences regarding their importance. In this regard, the data shows that the *as-operated* property is considered very important. In contrast to that, the properties *as-manufactured* and *as-designed* are considered important. This shows that DTs are especially interesting for the operational state of vehicles. The fact that the property *virtual representation* is considered very important, too, aligns with the answers for RQ1.1, where the *development* phase is considered one of the most important ones. Here, simulating the vehicle in development can be very helpful. The other very important properties *collects data* and *bi-directional* align with the automotive development phases *usage & support* and *development*. These provide valuable insights into the actual vehicle.

5.1.3 RQ1.3: What are desired purposes of using DTs? Concerning the purposes of DTs (cf. Figure 4c) our, participants from the automotive domain agreed that *prediction*, *monitoring*, and *validation* are the most important purposes. This indicates that the purpose of using the DT is focused on collecting data to observe and predict the behavior of a vehicle, but not actively controlling or interacting with the vehicle. The purposes of *homologation* and *coordination* of multiple twinned systems had the full range of answers, from not important to very important. Because some experts of the consortium actively work on improving homologation and fleet management, this might introduce some bias towards the importance of the topic, while other participants did not consider these purposes important.

5.1.4 RQ1.4: How do these purposes change in relation to different phases of automotive development? For the correlation between the answers for relevant purposes and development phases of DTs (cf. Figure 5), we found that the DT is considered useful in all automotive development phases. There is a strong correlation between the purpose *monitoring* and the post-production phases*usage & support* and *retirement*. This indicates that the DT should monitor the running system for instance to recognize erroneous behavior early and to gain knowledge from the vehicle when it is retired, for instance, to analyze historical data for a new product generation. Since *validation* is required and important especially when *conceptualizing* a product, its *development* and its *production*, there is a strong correlation between this purpose and these three development phases. The purpose of *homologation* strongly correlates with the *concept* phase. This indicates that automotive experts want to know as early as in the concept phase whether the product is going to be accepted or what might prevent it from becoming accepted by the authorities. Our findings indicate that the purpose *coordination* is relevant for the *concept* phase too. This enables the design concept to already interact with other systems. For *retirement*, it seems important to inform or to know the other systems the vehicle is



(a) Model kinds employed in each phase **(b) Tools employed to design model kind**

Figure 9: Sankey diagram relating the development phases to the various model kinds employed (a) as well as the tools used to model them. Thickness corresponds to the number of answers with *rather important* or *very important*. Participants named all model kinds they employ.

coordinated with. The purpose *prediction* only correlates with the *usage & support* of the vehicle. This indicates that the prediction is mostly useful during the usage of the vehicle, e.g., to predict future behavior or maintenance cycles. Concerning the purpose of *process optimization*, the answers mostly correlated with the phase *production*. This indicates that the participants mostly thought of the production process when aiming for process optimization. In fact, all phases include processes to be optimized. *Risk assessment* strongly correlates with the phases of *production*, *usage & support* and *retirement*. Ass such, this indicates that the automotive industry experts consider these phases as the most risk full, e.g., production errors or outages, malfunctions during the usage of the vehicle or expensive retirement costs.

Answer to RQ1: In the automotive industry, the DT is considered most useful in the *usage & support* phase of automotive development. This is supported by the fact that most participants considered that the DT should describe the vehicle *as-operated*. To this end, the most important purposes of the twins are the monitoring of data, performing validations and predicting future behavior. However, the purposes of the DT depend on the phase of automotive development.

5.2 RQ2: Which modeling languages and modeling tools are currently employed in the automotive industry?

5.2.1 RQ2.1: Which kinds of models are important in automotive development? Regarding the kinds of models relevant to the automotive domain, we found that besides *simulation models*, *source code* and *business process models* were the kinds used by most participants. Notably, though, we found that *deployment models*, previously not considered by us, were employed by some participants. While this might be in part explained by the distribution of professional backgrounds, we insist that this still highlights the variety of models employed in the automotive domain regardless of the total number of mentions. More importantly, our findings indicate that several model kinds are relevant for each working area and most participants work with two to four different kinds of models. Thus, for a DT used during automotive development, it must be able to include and relate multiple different kinds of models ranging from structural, e.g., *CAD models*, via behavioral, e.g., *business process models*, to quality models, e.g., *variability models*.

5.2.2 RQ2.2: How important is each model kind for the phases of automotive development? As the DT is supposed to support all phases of development, we especially asked for the importance of each model kind for each phase of automotive development. Although we initially assumed that many model kinds would be considered important in one or two phases, we found that most model kinds are considered important for at least three phases, except *CAD* *models*. Moreover, we were surprised to find that *data schemes*, *business process models*, *development processes* and *variability models* are considered important throughout all phases of automotive development. This indicates that for DTs in the automotive domain should consider the management of *data schemas*, *business process models*, as well as *variability models*. In particular, DTs must handle different versions and variants of automotive. From the participant's answers, we conclude that all model kinds are relevant for all phases of automotive development. While some are more important in a phase than others, we argue that all model kinds should be supported by a DT, regardless of whether it is used in one phase or multiple phases.

5.2.3 RQ2.3: Which tools are used to create and maintain these models? Finally, we were interested in the kinds of tools and development environments employed in the automotive domain and whether they already support multiple different model kinds. Surprisingly, we found that only a minority of tools integrate two (or more) model types, such that both models can be created, provisioned, and their consistency be ensured. *PREEvision* is a model-driven development environment[36] supporting multiple layers of abstraction from requirements down to the car's electrical wiring, the tool landscape is characterized by isolated tools tailored to one particular model kind. Consequently, to introduce a DT into the automotive domain would require connecting to a large variety of tools, as participants would want to continue using the existing tools. Although there are established standards in the automotive domain, e.g., AUTOSAR [18], used for the exchange of structural and to some extent behavioral information about a specific automotive, for many model kinds there is simply no standard. We conclude that, for a DT to be successful in the automotive domain it must be able to connect to and/or interact with many different established, commercial tools and development environments. Granted, this will be a major challenge for the modeling community.

Answer to RQ2: *Simulation models*, *source code*, and *business process models* are the most important models used. However, in all phases of automotive development, models are relevant and used in practice. Concerning tools, they are mostly limited to one modeling language. Since all models are relevant in the different phases of automotive development, the tools that allow their creation are too. Therefore, we conclude that establishing a DT in the automotive domain requires it to integrate and interact with many different modeling languages and tools.

5.3 Threats to Validity

Our survey is subject to the four basic types of validity threats according to [43] are construct validity (research design), internal validity (data extraction), external validity (generalizability), and conclusion validity (reliability).

Construct validity refers to how well the study represents what the researchers intended it to represent in the study construction. In this study, we relied solely on a questionnaire to measure the definition of DTs and the models and tools employed by the German automotive industry. To mitigate this issue, we did two pre-studies and discussed the questionnaire content with experts in the field of modeling and DTs. Another threat to construct validity could arise from discretizing the importance of the different properties of DTs, modeling languages, and used tools using balanced four-point Likert scales. This, however, has been investigated and identified as not compromising construct validity [35]. Another threat to construct validity could originate from too many questions overwhelming the participants and hindering them from completing the questionnaire. To mitigate this threat, we showed questions that depended on the agreement with another question only on the agreement, thereby presenting every participant with the most minimal version of the questionnaire.

A threat to *internal validity* could arise from the different understandings of what a DT is and the fact that the questionnaire was filled out online without supervision. To mitigate this effect, we performed multiple pilot studies prior to sending out the survey to the participants. Since our survey targeted German automotive experts within a German government-funded project we designed the survey in German language. To mitigate this issue, we provide the original questionnaire in German online, together with the extracted data and a table on how we translated the German terms into English.

Threats to *external validity* indicate to which extent the results of our study can be generalized. One threat to external validity is that most of our 55 of 115 participants are computer scientists and thereby the answers to our survey could be biased towards their view on the definition of DTs and the importance of certain models and modeling tools. Another threat to external validity is that we only surveyed members of the SofDCar research project and, thus, are not representative of the German automotive industry in general. To mitigate both threats, we call for replication to extend and generalize the results of the study to other disciplines that leverage DTs and models in practice and also to other companies involved in the German automotive industry and beyond.

Threats to *conclusion validity* entail avoiding wrong conclusions and permitting replicability. Regarding the former, because the answers to the questions were selected by the authors of this paper and could introduce some bias towards desired properties of DTs or the importance of modeling language or tools given by us. To mitigate this issue, we always included an open-answer field for input that was not initially part of the survey. For the study's replicability, we detailed the complete research method in Section 3, which enables replicating the methodology of this questionnaire survey. In addition, we provide the data extraction results and the original survey as PDF online.[2]

6 RELATED STUDIES

6.1 Surveys on Digital Twins

DTs are relevant in many domains and, as such, objective to current research activities. A systematic mapping study on software engineering for DTs [11] investigated the purposes and life cycle phases considered relevant for DTs in the current literature. They concluded that the main purpose of DTs is behavior optimization, monitoring, and prediction, especially in the runtime of the vehicle. These observations align with the responses from our survey where monitoring, prediction, and optimization together with validation are the main purposes of employing DTs. Another literature survey focused on the technical aspects of realizing DTs [30] and identified four essential categories of technologies that are required to work together that are communication, representation, computation, and microservices. Since the aim of the SofDCar research project is building a DT from scratch the technologies to build a DT were not the scope of our questionnaire. A survey among 22 participants from the production planning department of a major automobile manufacturer [7] showed that DTs increase the quality and lower the complexity in automotive production planning. The added value of DTs in production is also reflected in our study where our participants identified production as the second most important phase of development (cf. Figure 4a). Another systematic literature review on DTs in automotive application [12] differentiates between the purposes of a DT for manufacturing, which can monitor and optimize the production of an automotive, and a DT for the automotive in operation, where the DT can predict, monitor and optimize the energy consumption of an automotive. Our survey also investigated the question of which purposes of a DT is relevant in which phase of automotive development (cf. Figure 5). We identified validation, process optimization and risk assessment to be desired purposes in the production phase and monitoring, prediction, and risk assessment in the usage & support phase.

6.2 Surveys on Modeling Languages and Tools in Automotive Development

A systematic mapping study on modeling languages for Industry 4.0 [44] identified UML variants and domain-specific languages as the most often applied languages followed by knowledge representations, e.g., via data models. Furthermore, they state that despite validation being an important concern in Industry 4.0, there is little research in this direction. Compared to our study, the mapping study relies on literature, whereas we questioned experts from the German automotive industry on which modeling languages and tools they use most frequently. Concerning model-based engineering in the embedded domain, a survey among 113 participants from industry [29] highlighted that models are a central aspect of developing embedded systems. Common languages are MATLAB/Simulink, UML and SysML which are used for simulation and code generation, as well as behavioral and structural consistency checking. Another survey for modeling and model-driven engineering practices in the embedded industry [2] showed that informal models are widely used in the embedded industry. A recent expert survey [16] analyzed the use of informal models in the automotive industry and showed their importance in the context of DTs. In contrast to these studies, our study focuses on formalized modeling languages, their tools and their relevance for the different phases of development and work areas in the automotive industry. In particular, we did not investigate the relevance of current research efforts in the automotive domain, such as, automotive reconfiguration [37], consistency preservation [42], handling uncertainty [1], modelling of variants and versions [4], or novel modeling and programming languages [26]. Besides surveys covering the state of the art of modeling in general, there exist surveys for model-based security testing approaches [39], variability modeling [5], and requirements modeling [38]. However, we wanted to provide an overview of common tools and modeling languages in the automotive domain, and surveying the languages and tools in detail was not the scope of our survey and could be potential future work, e.g., employing the assessment grid for intelligent modeling assistance [32].

7 CONCLUSION AND FUTURE WORK

We have investigated purposes, expected properties, and modeling techniques for DTs in the automotive industry through the lens of German automotive companies and suppliers involved in the SofDCar project. Based on a questionnaire survey answered by 96 professional automotive (software) experts, out of which 43 completed all answers to the questionnaire, we found that digital twins are considered most important in the usage and support phase (i.e., once the vehicle exists) and less during other phases. Depending on the phase in which the DT is meant to be used, experts expect it to fulfill different purposes, which might entail that there will be different kinds of digital twins of a single vehicle throughout its life cycle. The relevance of monitoring during the conception phase of an automotive DT suggests that they also serve as a foundation for learning and improvement for future vehicles. To model digital twins, besides business process models, descriptive mathematical/physical models and predictive simulation models are most important to the participants. These are surpassed only by source code, which might hint at a lack of sufficiently expressive modeling languages for some aspects of automotive DT development. Future studies might shed light on the role of source code in the engineering and application of (automotive) DTs.

Overall, we conclude that integrating models into automotive DTs is highly relevant, as expected from the definitions of digital twins. Moreover, the modeling languages employed for such DTs differ from the modeling languages employed for DTs in general [11], which, e.g., also includes knowledge representation techniques. This suggests that automotive DTs indeed are domain-specific, at least, for the employed models. Furthermore, as most models are supported by a single tailored modeling tool, to be accepted by practitioners, automotive DTs must be coupled to a variety of existing tools. Overall, we believe that the insights gained from our questionnaire survey can help practitioners and researchers engineering automotive DTs more systematically.

ACKNOWLEDGMENTS

This work has been supported by the German Federal Ministry of Economic Affairs and Climate Action (BMWK, SofDCar – 19S21002L).

REFERENCES

[1] Maribel Acosta, Sebastian Hahner, Anne Koziolek, Thomas Kühn, Raffaela Mirandola, and Ralf Reussner. 2022. Uncertainty in coupled models of cyber-physical systems. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. Association for Computing Machinery (ACM), 569–578.

[2] Deniz Akdur, Vahid Garousi, and Onur Demirörs. 2018. A survey on modeling and model-driven engineering practices in the embedded software industry. *J. Syst. Archit.* 91 (2018), 62–82.

[3] Rafael Gomes Alves, Gilberto Souza, Rodrigo Filev Maia, Anh Lan Ho Tran, Carlos Kamienski, Juha-Pekka Soininen, Plinio Thomaz Aquino, and Fábio Lima. 2019. A digital twin for smart farming. In *IEEE Global Humanitarian Technology Conference, GHTC 2019, Seattle, WA, USA, October 17-20, 2019*. IEEE, 1–4.

[4] Sofia Ananieva, Sandra Greiner, Timo Kehrer, Jacob Krüger, Thomas Kühn, Lukas Linsbauer, Sten Grüner, Anne Koziolek, Henrik Lönn, S. Ramesh, and Ralf Reussner. 2022. A conceptual model for unifying variability in space and time: Rationale, validation, and illustrative applications. *Empirical Software Engineering* 27, 5 (2022), Article no: 101.

[5] Thorsten Berger, Ralf Rublack, Divya Nair, Joanne M. Atlee, Martin Becker, Krzysztof Czarnecki, and Andrzej Wasowski. 2013. A survey of variability modeling in industrial practice. In *The Seventh International Workshop on Variability Modelling of Software-intensive Systems, VaMoS '13, Pisa , Italy, January 23 - 25, 2013*, Stefania Gnesi, Philippe Collet, and Klaus Schmid (Eds.). ACM, 7:1–7:8.

[6] Pascal Bibow, Manuela Dalibor, Christian Hopmann, Ben Mainz, Bernhard Rumpe, David Schmalzing, Mauritius Schmitz, and Andreas Wortmann. 2020. Model-Driven Development of a Digital Twin for Injection Molding. In *International Conference on Advanced Information Systems Engineering (CAiSE'20)* (Grenoble) *(Lecture Notes in Computer Science, Vol. 12127)*, Schahram Dustdar, Eric Yu, Camille Salinesi, Dominique Rieu, and Vik Pant (Eds.). Springer International Publishing, 85–100.

[7] Florian Biesinger, Benedikt Kraß, and Michael Weyrich. 2019. A survey on the necessity for a digital twin of production in the automotive industry. In *2019 23rd International Conference on Mechatronics Technology (ICMT)*. IEEE, 1–8.

[8] Tim Bolender, Gereon Bürvenich, Manuela Dalibor, Bernhard Rumpe, and Andreas Wortmann. 2021. Self-Adaptive Manufacturing with Digital Twins. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)* (Madrid, Spain) *(SEAMS '21)*. Association for Computing Machinery, New York, NY, USA, 1–4.

[9] Tobias Brockhoff, Malte Heithoff, István Koren, Judith Michael, Jérôme Pfeiffer, Bernhard Rumpe, Merih Seran Uysal, Wil MP Van Der Aalst, and Andreas Wortmann. 2021. Process prediction with digital twins. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 182–187.

[10] Manuela Dalibor, Malte Heithoff, Judith Michael, Lukas Netz, Jérôme Pfeiffer, Bernhard Rumpe, Simon Varga, and Andreas Wortmann. 2022. Generating customized low-code development platforms for digital twins. *Journal of Computer Languages* 70 (2022), 101117.

[11] Manuela Dalibor, Nico Jansen, Bernhard Rumpe, David Schmalzing, Louis Wachtmeister, Manuel Wimmer, and Andreas Wortmann. 2022. A cross-domain systematic mapping study on software engineering for Digital Twins. *Journal of Systems and Software* (2022), 111361.

[12] Shutong Deng, Liang Ling, Caizhi Zhang, Congbo Li, Tao Zeng, Kaiqing Zhang, and Gang Guo. 2023. A systematic review on the current research of digital twin in automotive application. *Internet of Things and Cyber-Physical Systems* (2023).

[13] digital twin consortium. 2022. Capabilities Periodic Table. Last accessed: 2023-12-30.

[14] Romina Eramo, Francis Bordeleau, Benoit Combemale, Mark van den Brand, Manuel Wimmer, and Andreas Wortmann. 2022. Conceptualizing Digital Twins. *IEEE Software* 39, 2 (2022), 39–46.

[15] Romina Eramo, Francis Bordeleau, Benoit Combemale, Mark van Den Brand, Manuel Wimmer, and Andreas Wortmann. 2021. Conceptualizing Digital Twins. *IEEE Software* 39, 2 (2021), 39–46.

[16] Dominik Fuchß, Thomas Kühn, Andreas Wortmann, Jérôme Pfeiffer, and Anne Koziolek. 2023. *An Expert Survey on the Use of Informal Models in the Automotive Industry*. Technical Report. Karlsruher Institut für Technologie (KIT). https://doi.org/10.5445/IR/1000162389 ECSA TwinArch Workshop 2023.

[17] Shan Fur, Malte Heithoff, Judith Michael, Lukas Netz, Jérôme Pfeiffer, Bernhard Rumpe, and Andreas Wortmann. 2023. Sustainable digital twin engineering for the internet of production. In *Digital Twin Driven Intelligent Systems and Emerging Metaverse*. Springer, 101–121.

[18] Simon Fürst and Markus Bechter. 2016. AUTOSAR for Connected and Autonomous Vehicles: The AUTOSAR Adaptive Platform. In *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*. IEEE, Toulouse, France, 215–217.

[19] Sebastian Götz, Andreas Fehn, Frank Rohde, and Thomas Kühn. 2020. Model-driven Software Engineering for Construction Engineering: Quo Vadis? *J. Object Technol.* 19, 2 (2020), 2:1–22.

[20] Edward Y. Hua, Sanja Lazarova-Molnar, and Deena P. Francis. 2022. Validation of Digital Twins: Challenges and Opportunities. In *2022 Winter Simulation Conference (WSC)*. IEEE, 2900–2911.

[21] Kenneth H Huebner, Donald L Dewhirst, Douglas E Smith, and Ted G Byrom. 2001. *The finite element method for engineers*. John Wiley & Sons.

[22] ISO/DIS. 2020. *ISO/DIS 23247 Automation Systems and Integration-Digital Twin Framework for Manufacturing*. Technical Report. International Standardization Organization (ISO).

[23] Jörg Christian Kirchhof, Judith Michael, Bernhard Rumpe, Simon Varga, and Andreas Wortmann. 2020. Model-driven digital twin construction: synthesizing the integration of cyber-physical systems with their information systems. In *23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*. 90–101.

[24] GL Knapp, Tuhin Mukherjee, JS Zuback, HL Wei, TA Palmer, Amitava De, and TJAM DebRoy. 2017. Building blocks for a digital twin of additive manufacturing. *Acta Materialia* 135 (2017), 390–399.

[25] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. 2018. Digital Twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline* 51, 11 (2018), 1016–1022.

[26] Thomas Kühn, Max Leuthäuser, Sebastian Götz, Christoph Seidl, and Uwe Aßmann. 2014. A Metamodel Family for Role-Based Modeling and Programming Languages. In *Software Language Engineering*. Springer International Publishing, Cham, 141–160.

[27] Nathan Lauzeral, Domenico Borzacchiello, Michael Kugler, Daniel George, Yves Rémond, Alexandre Hostettler, and Francisco Chinesta. 2019. A model order reduction approach to create patient-specific mechanical models of human liver in computational medicine applications. *Comput. Methods Programs Biomed.* 170 (2019), 95–106.

[28] Daniel Lehner, Jérôme Pfeiffer, Erik-Felix Tinsel, Matthias Milan Strljic, Sabine Sint, Michael Vierhauser, Andreas Wortmann, and Manuel Wimmer. 2022. Digital Twin Platforms: Requirements, Capabilities, and Future Prospects. *IEEE Software* 39, 2 (2022), 53–61.

[29] Grischa Liebel, Nadja Marko, Matthias Tichy, Andrea Leitner, and Jörgen Hansson. 2018. Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice. *Softw. Syst. Model.* 17, 1 (2018), 91–113.

[30] Kendrik Yan Hong Lim, Pai Zheng, and Chun-Hsien Chen. 2020. A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. *J. Intell. Manuf.* 31, 6 (2020), 1313–1337.

[31] Ryan Magargle, Lee Johnson, Padmesh Mandloi, Peyman Davoudabadi, Omkar Kesarkar, Sivasubramani Krishnaswamy, John Batteh, and Anand Pitchaikani. 2017. A Simulation-Based Digital Twin for Model-Driven Health Monitoring and Predictive Maintenance of an Automotive Braking System. In *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017 (Linköping Electronic Conference Proceedings, Vol. 132)*, Jirí Kofránek and Francesco Casella (Eds.). Linköping University Electronic Press, 132:003.

[32] Gunter Mussbacher, Benoit Combemale, Silvia Abrahão, Nelly Bencomo, Loli Burgueño, Gregor Engels, Jörg Kienzle, Thomas Kühn, Sébastien Mosser, Houari Sahraoui, and Martin Weyssow. 2020. Towards an assessment grid for intelligent modeling assistance. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (Virtual Event, Canada) *(MODELS '20)*. Association for Computing Machinery, New York, NY, USA, Article 48, 10 pages.

[33] Jérôme Pfeiffer, Daniel Lehner, Andreas Wortmann, and Manuel Wimmer. 2022. Modeling Capabilities of Digital Twin Platforms - Old Wine in New Bottles? *Journal of Object Technology* 21, 3 (July 2022), 3:1–14. The 18th European Conference on Modelling Foundations and Applications (ECMFA 2022).

[34] PK Rajesh, N Manikandan, CS Ramshankar, T Vishwanathan, and C Sathishkumar. 2019. Digital twin of an automotive brake pad for predictive maintenance. *Procedia Computer Science* 165 (2019), 18–24.

[35] John J Ray. 1982. The construct validity of balanced Likert scales. *The Journal of Social Psychology* 118, 1 (1982), 141–142.

[36] Jörg Schäuffele. 2016. *E/E architectural design and optimization using PREEvision*. Technical Report. SAE Technical Paper.

[37] Marc Schindewolf, Jan Willem Wittler, Thomas Kühn, Daniel Grimm, and Eric Sax. 2023. A Model-Based Approach to Automotive Feature Development for Updates and Upgrades. In *International Conference on Service-Oriented System Engineering SOSE*. IEEE, Athens, Greece, 19–26.

[38] Samuel Sepúlveda, Ania Cravero, and Cristina Cachero. 2016. Requirements modeling languages for software product lines: A systematic literature review. *Inf. Softw. Technol.* 69 (2016), 16–36.

[39] Florian Sommer, Reiner Kriesten, and Frank Kargl. 2023. Survey of Model-Based Security Testing Approaches in the Automotive Domain. *IEEE Access* 11 (2023), 55474–55514.

[40] Fei Tao, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. 2018. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology* 94 (2018), 3563–3576.

[41] Fei Tao, Qinglin Qi, Lihui Wang, and AYC Nee. 2019. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering* 5, 4 (2019), 653–661.

[42] Jan Willem Wittler, Timur Saglam, and Thomas Kühn. 2023. Evaluating Model Differencing for the Consistency Preservation of State-based Views. *The Journal of Object Technology* 22, 2 (2023), 2:1–14. 46.23.01; LK 01.

[43] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[44] Andreas Wortmann, Olivier Barais, Benoit Combemale, and Manuel Wimmer. 2020. Modeling Languages in Industry 4.0: an Extended Systematic Mapping Study. *Software and Systems Modeling* 19, 1 (January 2020), 67–94.

[45] Qinghua Xu, Shaukat Ali, and Tao Yue. 2021. Digital Twin-based Anomaly Detection in Cyber-physical Systems. In *14th IEEE Conference on Software Testing, Verification and Validation (ICST)*. 205–216.