Software and Systems Modeling
https://doi.org/10.1007/s10270-026-01410-9

SPECIAL SECTION PAPER

A survey of modeling languages for automotive digital twins

Jérôme Pfeiffer[1] · Dominik Fuchß[2] · Thomas Kühn[3] · Dirk Neumann[2] · Christer Neimöck[4] · Anne Koziolek[2] · Andreas Wortmann[1]

Received: 5 May 2025 / Revised: 3 June 2026 / Accepted: 25 June 2026

Abstract

The demand for digital twins and suitable modeling techniques in the automotive industry is increasing 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, (ii) the modeling techniques related to engineering and operating digital twins across the phases of automotive development, and (iii) the role informal models play during 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 the most important models to be integrated into a digital twin alongside the associated, established tools. Furthermore, informal models are frequently created digitally, e.g., using PowerPoint™, mainly employed for communication and documentation during the conception and development phase.

Keywords Modeling languages · Digital twins · Automotive · Survey

1 Introduction

Digital Twins (DTs) [34, 56] 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 [18, 21, 29]. They are used in a variety of domains, including agriculture [4], manufacturing [11], medicine [36], and many more [16]. Typical purposes of digital twins include monitoring [41], validating [27], predicting [47], and optimizing the behavior of the actual system. For this, DTs require different properties [18], such as, collecting data, model processing, simulation, bi-directional communication. Related studies collected various definitions of DTs across domains from the literature [16] and investigated how DTs can be engineered for the production of automotive and what their added-value could be [10]. While other related studies focusing on modeling languages [5, 62] and model-based engineering [39] 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 [3, 39]. Concerning informal models, dif-

Communicated by Marsha Chechik and Benoit Combemale.

✉ Jérôme Pfeiffer
jerome.pfeiffer@isw.uni-stuttgart.de

Dominik Fuchß
dominik.fuchss@kit.edu

Thomas Kühn
thomas.kuehn@informatik.uni-halle.de

Dirk Neumann
dirk.neumann@kit.edu

Christer Neimöck
C.Neimoeck@t-systems.com

Anne Koziolek
koziolek@kit.edu

Andreas Wortmann
wortmann@isw.uni-stuttgart.de

1 University of Stuttgart, Stuttgart, Germany

2 Karlsruhe Institute of Technology, Karlsruhe, Germany

3 Martin Luther University Halle-Wittenberg, Halle, Germany

4 T-Systems GmbH, Frankfurt, Germany

Published online: 03 August 2026

ferent studies have examined the use of informal models [55], sketches [7, 59], and diagrams [14] in software development. However, no study focusing on the utilization of informal model in the automotive development exists.

At the center of digital twins are formal 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 [37, 46].

In addition, informal models are already widely used in software development [26], as they ease the communication and documentation of software. Furthermore, they support multiple software development phases [7, 54]. Consequently, research on approaches to deal with informal diagrams and sketches often focus on software development [30, 49, 58, 63]. Notably though, the use of informal models for the development of CPSs or the development with a digital twin is rarely studied. A future automotive digital twin could not only include formal engineering models but also the rich ecosystem of informal models–sketches, diagrams, slides, notes, and conceptual drawings–by automatically interpreting, structuring, and transforming them into actionable, machine-readable knowledge. Consequently informal models would become a living input source that continuously enriches the twin throughout the vehicle's lifecycle.

A study surveying automotive industry experts on the expectations, use, and formal, as well as, informal modeling techniques for DTs in their domain is missing.

To remedy this, we investigate the purposes, expected properties, formal, and informal 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 gather insights from the SofDCar experts, we designed and administered a questionnaire. Because we assume that digital twins are still uncommon in automotive practice, the first part focuses on experts' understanding of and expectations for digital twins. This also sets the context for digital twins. The second part examines current modeling practices, including the use of modeling languages, tools, and informal models. This allows us to identify existing models that could potentially support future digital twin development. Based on the experts' responses, we address the following research questions (RQs):

RQ: When developing digital twins for the automotive industry, what are the expectations of digital twins, and which modeling languages, tools, and practices should be taken into account for future automotive digital twins?

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?

RQ3: How pervasive are informal models in the automotive industry?

- **RQ3.1:** For which purpose are informal models used?
- **RQ3.2:** For which phases of automotive development are informal models employed?
- **RQ3.3:** How are informal models persisted?
- **RQ3.4:** Which kinds of models are informally designed?
- **RQ3.5:** How are informal models re-/used?

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 using these models in the context of digital twins will be the key to their successful deployment in practice. Last but not least, most participants employed informal models during the conception and development phase to communicate and document design decisions. Surprisingly, most participants frequently create informal models digitally, e.g., using PowerPoint.

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

This extended study has the following lessons learned:

- Model integration is crucial, with mathematical/physical and simulation models seen as essential.
- Automotive DT modeling is domain-specific, with surprisingly low use of UML/SysML compared to general DT literature, and strong dependence on specialized tooling.
- Tool integration is a major challenge, indicating the need for research on connecting diverse models and tools (e.g.,, via knowledge graphs, APIs, exchange formats).
- Informal models remain common for communication, creating opportunities for AI to convert them into formal models and support intelligent modeling assistants in future workflows.

This paper is based on a conference paper [45] and a preliminary report on informal models [22], that we combined, as in the initial conference paper we omitted the inclusion of the inquired use of informal models. In this extension we include the results and the analysis of the third part of our questionnaire about informal models. This part about informal was handed out to the automotive experts from the SofDCar consortium together with the questions about DTs and formal models. Extending our study with this part entails a new research question and two sections in the results and discussion section that present and analyze our findings.

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

2 Background

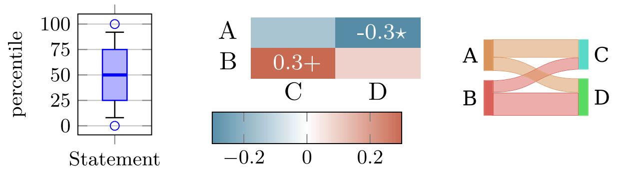
Before diving into the conducted study, in this section, we give a brief introduction to the notion of Digital Twins as well as to the diagrams we employ, i.e., box plots, heatmaps, and Sankey diagrams. The knowledgeable reader can safely skip this section.

2.1 Digital twins

Henceforth, we define a digital twin (DT) as: *"a software comprising models and services to purposefully represent and manipulate its original system across its complete lifecycle"* [9, 31, 34]. This definition acknowledges that a DT mirrors an original system, providing means to digitally represent that system, and to manipulate it. Moreover, it highlights that the models and services of a DT fulfill a specific purpose, which governs the models' abstraction and pragmatism. Furthermore, twinning means that changes to the original system are automatically reflected in the DT and changes in the DT are automatically reflected in the original system. To this end, DTs must establish a representation of their counterpart, along with a mechanism for transmitting changes between both systems [20]. Based on this, DTs enable a variety of value-adding services, such as, monitoring, behavior prediction [32], process optimization [12], predictive maintenance, and many more [57, 64]. As parts of our survey are based on our definition and experience with digital twins, the last question in our survey requested the agreement of the participants with our definition. We discuss the results to this answer in Sect. 5.

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, Fig. 1 illustrates each of them. First, box plots (Fig. 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 (Fig. 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 Pearson correlation coefficient between the two options. A legend indicates which color corresponds to which coefficient. In addition, we perform a T-test analysis to determine the significance of correlations. In case of a significant correlation the coefficient and a strength indicator is shown, the strong significance ($p < 0.05$) of the correlation between B and C is signified with a star in the cell, whereas the weak significance ($p < 0.1$) between A and D with a plus. Last but not least, Sankey diagrams (Fig. 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 Fig. 1c, the flow from option A to option C highlights that most participants who chose A also choose C in the second question, whereas only a few opted for D.



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

Fig. 1 Illustrative examples of the employed diagrams in our study

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 the 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 [11, 15, 16, 23, 46] and model-driven software engineering [25, 51, 62], 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. This questionnaire included the questions about the understanding of digital twins, the employed modeling languages and tools as well as the leveraged informal models. This questionnaire was handed out as one coherent document to the recipients. 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 Sankey diagrams. We performed a correlation analysis using the Pearson correlation coefficient and each time two-sided T-test.

3.1 Preliminary studies

We performed two preliminary studies before arriving at 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 ensure 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 min 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 three 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; the second part focuses on modeling techniques and tools; and the third part is concerned with the use and reuse of informal models.

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 (+) and completely agree (++). Participants could propose additional

[2] Replication packages: https://zenodo.org/doi/10.5281/zenodo.12529996 and https://doi.org/10.5281/zenodo.8199850.

[3] Tool for the questionnaire: https://www.limesurvey.org/.

[4] Tool for analysis of participant's answers: https://www.ibm.com/spss.

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.

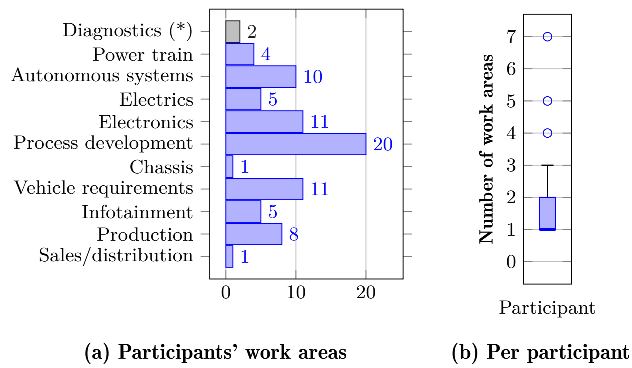


Fig. 2 Number of participants in different work areas (**a**), where the * indicates answers from the free-form field, and a box plot highlighting the distribution of work areas per participant (**b**)

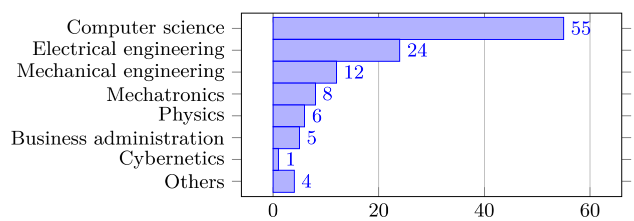


Fig. 3 Overview of the professional backgrounds of the participants, whereas one participant could select multiple backgrounds

3.2.3 Informal models in the automotive industry

In the third part, we ask participants about the prevalence and utilization of informal models, e.g., hand-drawn on whiteboards or digitally sketched in PowerPoint™, used in the automotive industry. In detail, participants should rate how often informal models were used for the purpose of communication, documentation, design of the DT, and requirements analysis of the DT. Moreover, they should rate how frequently informal models were utilized in each phase of development. Furthermore, participants should state how often an informal model is stored as a digital sketch (e.g., in PowerPoint), as a photograph, as a transcribed model (manually transferred using a modeling tool) or in place (e.g., on the whiteboard). For the above questions, we provided a five-point Likert scale ranging from very often (++), rather often (+), rather rarely (−) to rarely (−−) and never (Ø). Additionally, participants could enter custom purposes and development phases in a text field. Next, participants should select all model kinds they sketch or state custom model kinds. To limit the number of selectable kinds, we picked those model kinds selected in the prestudies. Finally, we tasked participants to rate their agreement to the following seven statements regarding issues using and persisting informal models: (1) *informal models are not reused because they are not stored;* (2) *informal models are not reused although they were stored;* (3) *photos of informal models were taken and reused later;* (4) *photos of informal models were taken yet never reused;* (5) *manually transcribing models was time consuming;* (6) *manually transcribing models were useful for later tasks;* (7) *stored informal models were difficult to find/retrieve.* Again, we employ the same four-point Likert scale for agreement and provide an extra field for participants to list other encountered issues wrt. re-/using informal models. The statements arose from discussions within the project that date before the conduction of the study. We formulated these hypotheses to assess their validity.

3.3 Selection of participants

To reach as many 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 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 Fig. 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. Fig. 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?

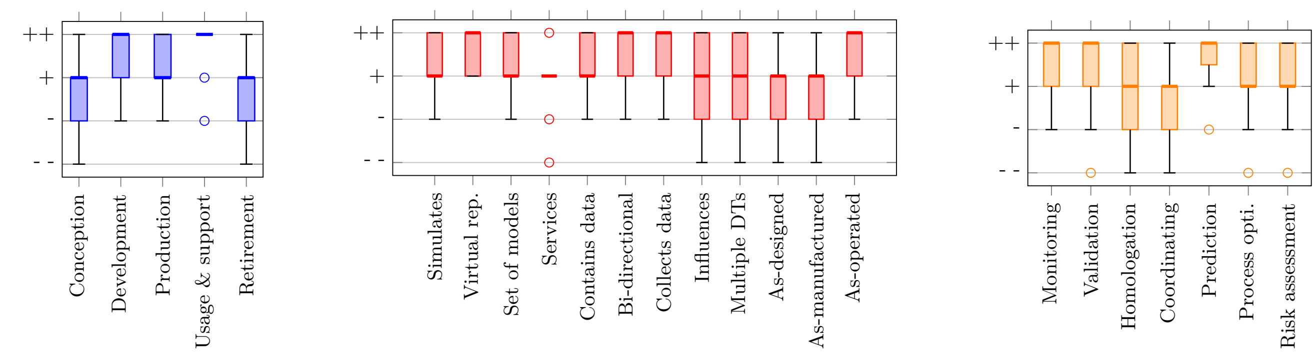
To answer the question of how important the participants consider DTs in different phases of automotive development, we asked them to rate the relevance of DTs in the phases *concept*, *development*, *production*, *usage & support*, and *retirement*. The answers ranged from *unimportant* to *very important*. The box plot in Fig. 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 Fig. 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. (7) For the purpose of risk assessment, the results are the same as for *process optimization*.



(a) Phases of development **(b) Properties** **(c) Purposes**

Fig. 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* (++)

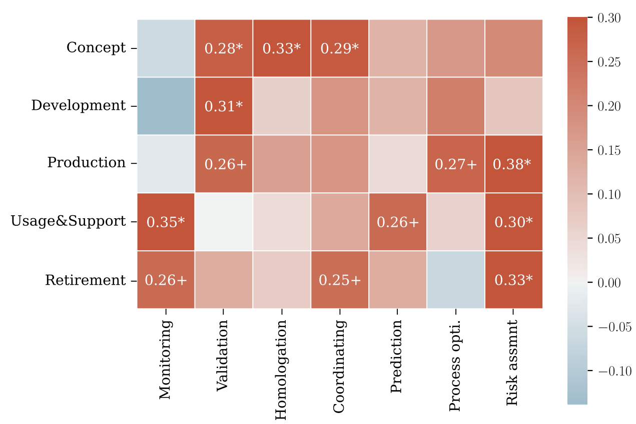


Fig. 5 Heatmap of the correlations between phases of development and purposes for DTs. Cells show the Pearson correlation, whereas the hue indicates a positive or negative correlation. Strong significance ($p < 0.05$) is indicated with a * and weak significance ($p < 0.10$) with a +

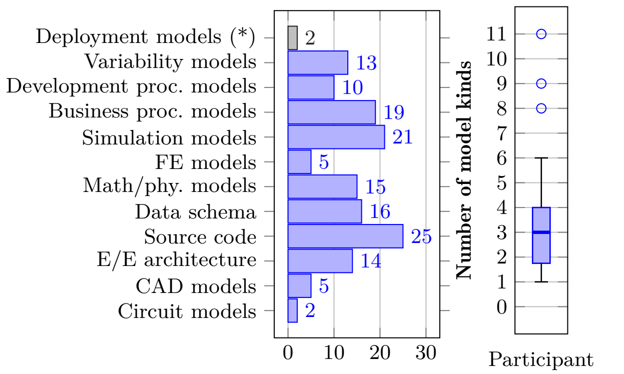
4.1.4 RQ1.4: How do these purposes change in relation to different phases of automotive development?

To answer the question of which DT purposes are relevant in specific development phases, we related the answers to both questions to one another, resulting in the heatmap depicted in Fig. 5. It shows that the purpose of *monitoring* is especially relevant in the post-production phases of *support* ($p = .011$) and *retirement* ($p = .066$). In contrast, *validation* is important in the pre-operational phases of *conception* ($p = .048$), *development* ($p = .026$) and *production* ($p = .064$). *Homologation* is at all relevant in the *conception* phase ($p = .023$). *Coordinating* as a purpose of the DT is considered relevant in all development phases but especially in the *conception* ($p = .047$) and *retirement* ($p = .081$) phases. The purpose *prediction* is a purpose that is mostly relevant in the *usage & support* phase ($p = .068$). *Process optimization* is highly relevant to the production process and, thus, there is a strong relation between this purpose and the *production* phase ($p = .06$). *Risk assessment* has a strong correlation in the phases of *production* ($p = .006$), *support* ($p = .03$), and *retirement* ($p = .018$).

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 Fig. 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 [28] 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*



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

Fig. 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**)

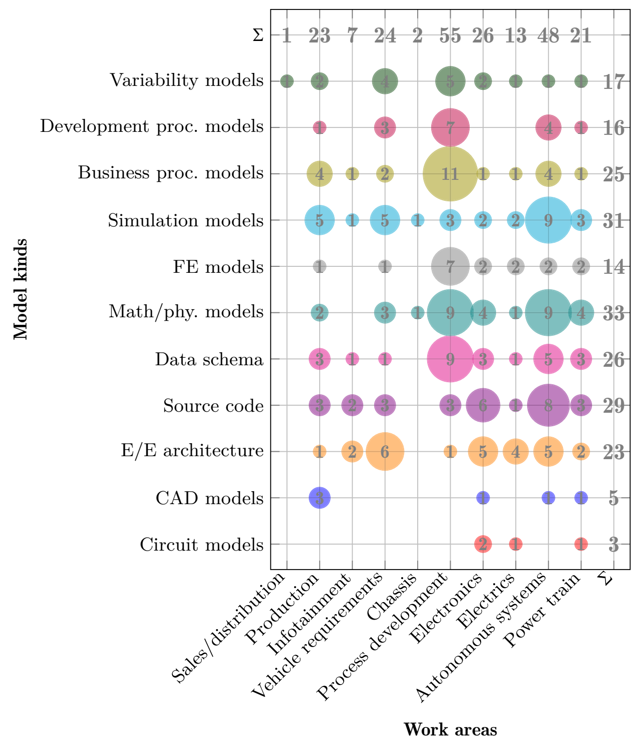


Fig. 7 Bubble chart relating the model kind (y-axis) to the various work areas of automotive development (x-axis)

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 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 Fig. 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 Fig. 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 Fig. 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

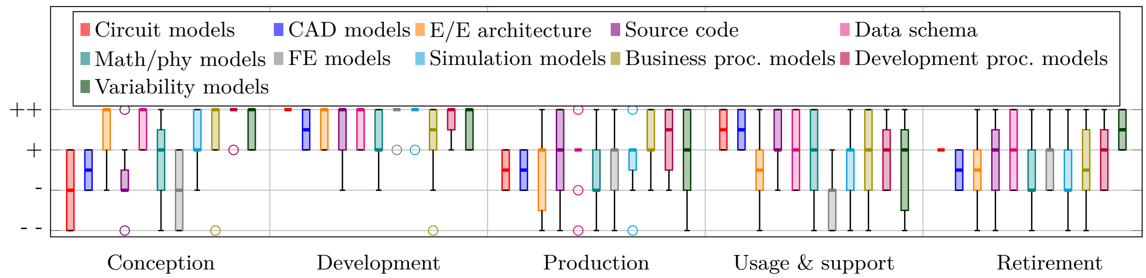


Fig. 8 Box plot of the importance of each model kind for the five phases of automotive development. The importance ranges from *unimportant* (−−), via *rather unimportant* (−) and *rather important* (+) to *very important* (++). The number of answers per model kind differs, as participants could select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

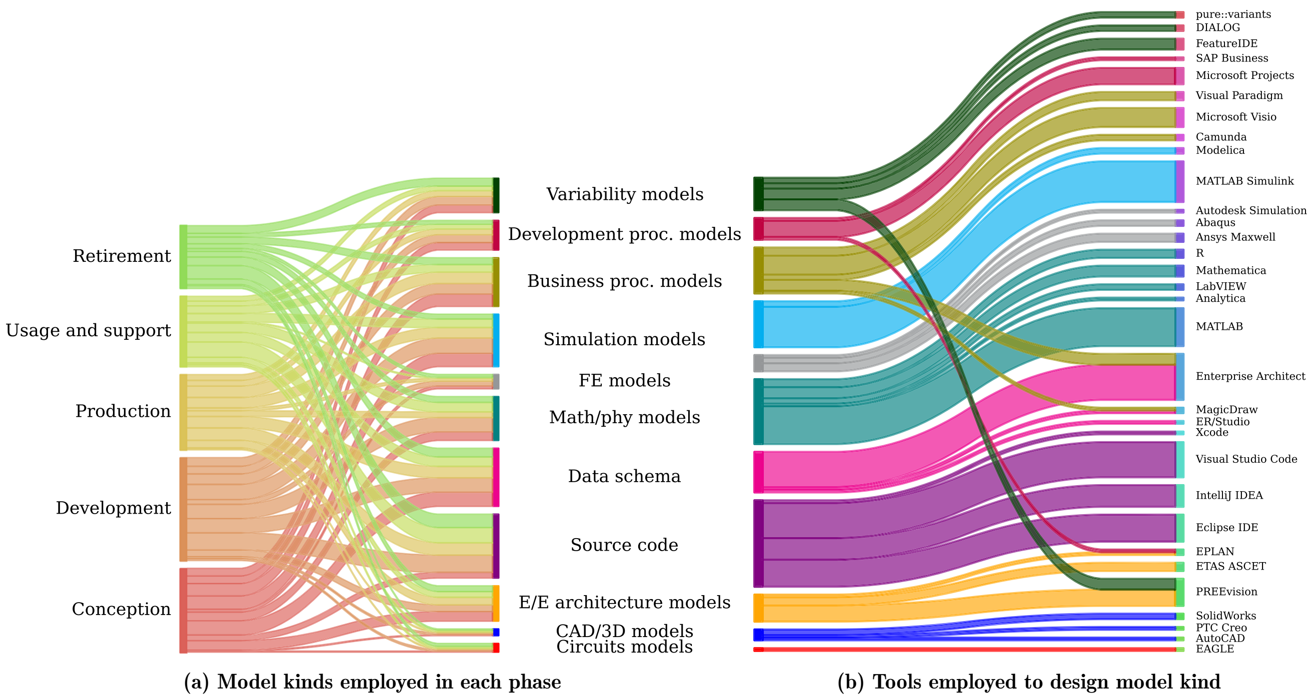


Fig. 9 Sankey diagram relating the development phases to the various model kinds employed (**a**) as well as the tools used to model them. Thickness corresponds to the number of answers with *rather important* or *very important*. Participants named all model kinds they employ

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, Fig. 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, that 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 mathematical/physical (math./phy.) models (14). To highlight the tools used in the automotive domain, Fig.

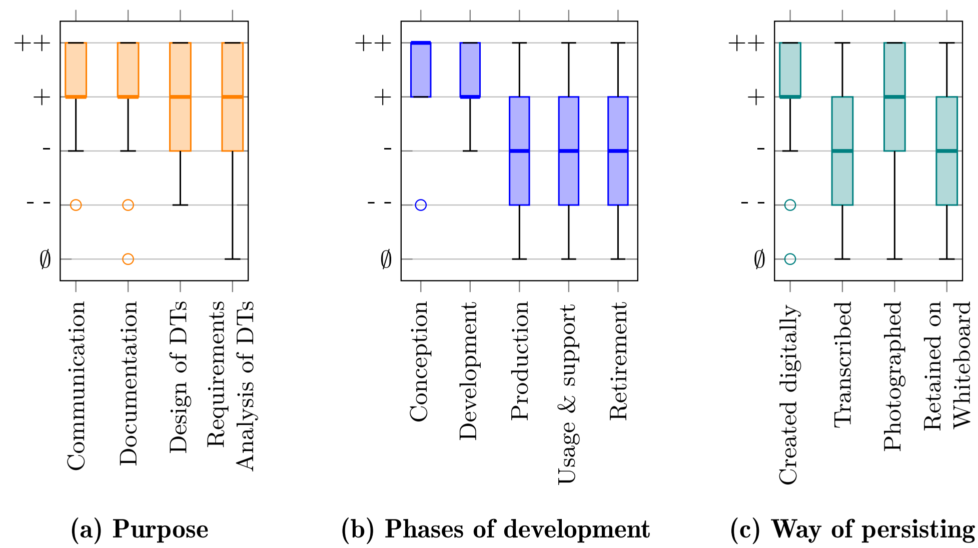


Fig. 10 Box plots of the purposes informal models (**a**) serve, of their use in different phases of development (**b**), of the way they are persisted (**c**). The frequency ranges from *never* (∅), via *rarely* (−−), *rather rarely* (−), and *rather often* (+) to *very often* (++)

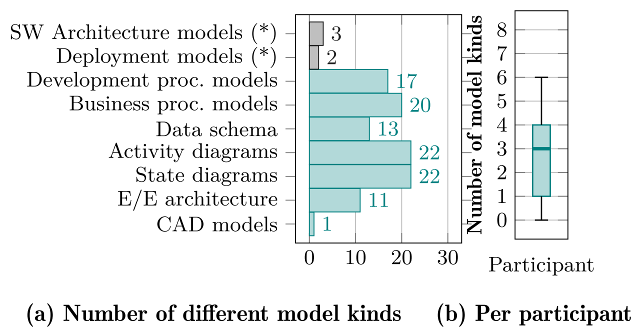


Fig. 11 Number of different model kinds informally depicted by participants (**a**), where the * indicates answers from the free-form field, and a box plot highlighting the distribution of model kinds used per participant (**b**)

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.

4.3 RQ3: How relevant are informal models in the automotive industry?

4.3.1 RQ3.1: For which purpose are informal models used?

To investigate how informal models are employed in practice, our first question asked participants to rate how frequent they emply informal models for each purpose. Figure 10a plots their answers indicating that informal models are often used for communication and documentation. Participants who chose ”rarely” or ”not at all” are considered outliers. Besides that, informal models were often used for the design and requirements analysis of DTs, although 25 % of the participants never or rarely use informal models for these purposes. In addition, three participants entered other purposes stating that informal models were used for *designing methods for continuous improvements or for updates*, for *modeling development process models*, for *evaluating use cases* or for *developing software architecture*. Although we would consider the latter two to correspond to the *design of DTs*.

4.3.2 RQ3.2: For which phases of automotive development are informal models employed?

To survey in which phase of development informal models are typically utilized, participants rated the frequency they use informal models in each phase. Figure 10b shows that informal models are often used in the *conception* and *development* phase. In contrast, the median frequency during the *production*, *usage & support* or *retirement* phase is *rarely used*, while the frequency in conception or development phase is *very often* respectively *rather often*.

4.3.3 RQ3.3: How are informal models persisted?

To gain insights into how informal models are retained, participants rated how often an informal model was created digitally, transcribed to a formal notation, stored as photograph and retained on the white board. The boxplots in Fig. 10c show that 75% of the participants create informal models digitally, whereas only outliers rarely or never create informal models digitally. Moreover, 50% of the participants rather often stored photographs of informal models. Still 25% rather rarely or never photograph informal models. In regard to transcribing informal models and retaining them on white-

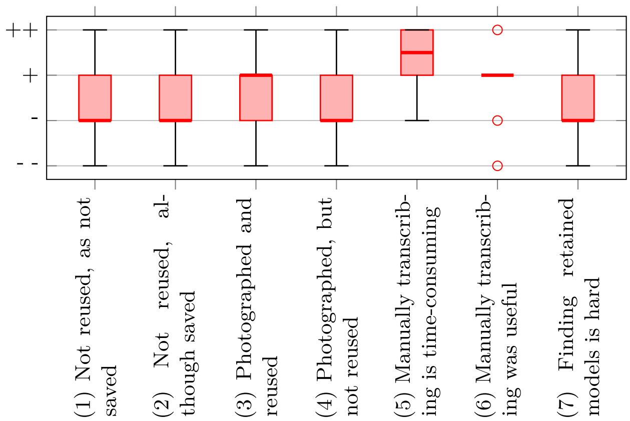


Fig. 12 Boxplot of the agreement with statements on the use and reuse of informal models. The agreemant ranges from *disagree at all* (−−), via *rather disagree* (−) and *rather agree* (+) to *agree at all* (++)

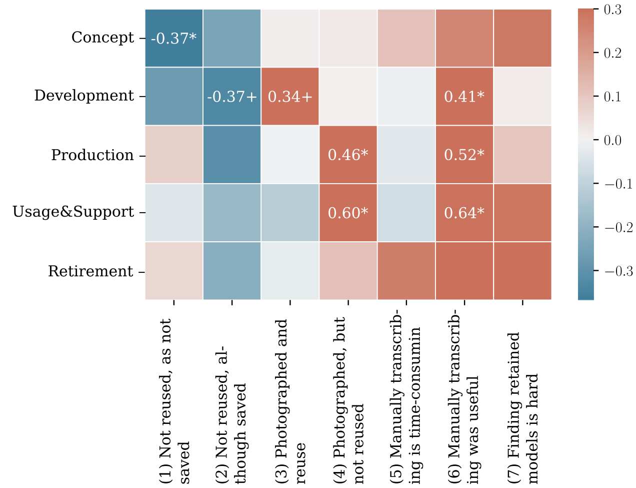


Fig. 13 Heatmap of the correlations between how frequently informal models are used in each phase of development and how informal models are used and reused. Hue indicates a positive or negative correlation. Strong significance ($p < 0.05$) is indicated with a * and weak significance ($p < 0.10$) with a +

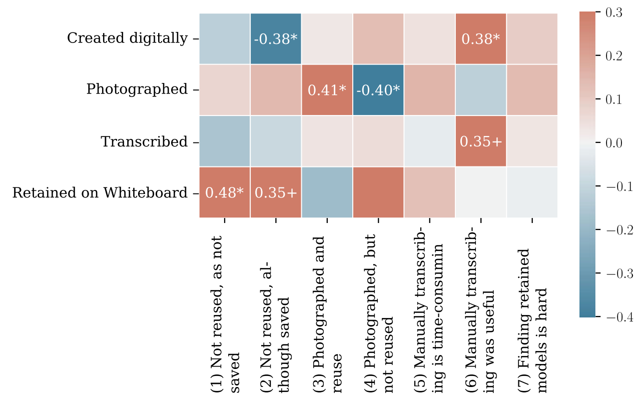


Fig. 14 Heatmap of the correlations between the way informal models are persisted and how they are used and reused. Hue indicates a positive or negative correlation. Strong significance ($p < 0.05$) is indicated with a * and weak significance ($p < 0.10$) with a +

boards, the answers of participants ranged from very often to never, whereas the median answer was rather rarely.

4.3.4 RQ3.4: Which kinds of models are informally designed?

To collect the kinds of informal models participants use, we asked participants to select any number of model kinds from a subset of seven model kinds or name an additional model kind. Figure 11a summarizes the answers, whereas *activity diagrams* and *state diagrams* were selected most often, closely followed by *business process models* and *development process models*. These three model kinds describe a system's or process' behavior. All other model kinds were selected less often. Additionally, three participants named *software architecture models* (or the synonym component models) and two *deployment models* as additional models. Besides that, Fig. 11b indicates that the number of different model kinds selected per participant is three, ranging from none to six. Only 6 participants did not select or named any model kind.

4.3.5 RQ3.5: How are informal models re-/used?

Finally, to investigate how participants use and reuse informal models in practice, we asked the participants to rate the agreement with seven statements on how informal models are stored and reused (cf. Sect. 3.2.3). In Fig. 12, we collected the results. The results indicate that the median *rather disagrees* with statements (1), (2), (4), and (7). Thus, disagreeing with statements including not reusing stored informal models. In contrast, the median of participants *rather agreed* with Statement (3) indicating that informal models are photographed and reused. Moreover, the median of the agreement with Statement (5) regarding the time consumption of manually transcribing informal models is between rather agree and agree at all, whereas no participants completely disagreed. Furthermore, all participants (except outliers) rather agree with Statement (6) regarding the usefulness of manually transcribing informal models.

Besides agreement with the seven statements, we also asked participants to describe obstacles hindering the reuse of informal models. Four participants provided such descriptions. We paraphrase their descriptions as:

- *Versioning is another problem*,
- *Lack of analytical ability and potential misunderstandings*,
- *(Partly) incorrect assumptions made for informal models*, and
- *Learning of rarely used modeling tool to transcribe informal model*.

In addition to the presented results, we conducted a correlation analysis to identify relations between how frequently different kinds of informal models are employed in each phase of development as well as the way informal models are persisted and (re)used. In Fig. 13, we depict the correlations between how often informal models are used in a particular phase of development and the agreement with the statements of reuse. Here, we observe a statistically significant correlation between the agreement with Statement (6), i.e., that manual transcription of informal models is useful, and the frequency of their use in *development* ($p = .027$), *production* ($p = .006$) or *usage & support* ($p = .0005$). Moreover, we see a statistically significant correlation between Statement (4) regarding not reusing photographed informal models and the frequency of their use in *production* ($p = .026$) and *usage & support* ($p = .0023$). Finally, we found a statistically significant negative correlation between how often informal models are used in the conception phase and agreement with Statement (1) ($p = .046$), i.e., that informal models are not reused as they are not persisted. Besides that, we found correlations between the frequency of using informal models in the development phase and Statement (2) ($p = .049$) and Statement (3) ($p = .097$). These, however, have a weak statistical significance.

The second correlation analysis, shown in Fig. 14, illustrates correlations between the frequency of persisting informal models in a particular way and the agreement to statements on re-/use. We identified a statistically significant negative correlation between the frequency of digitally created informal models and agreement with Statement (2) ($p = .032$), i.e., that informal models are not reused, although they were saved, as well as a positive correlation with Statement (6) ($p = .037$), i.e., that manually transcribing an informal model was useful. Similarly, we found a statistically significant positive correlation between the frequency of informal models retained as a photograph and agreement with Statement (3) ($p = .028$), i.e., that informal models are photographed and reused, as well as a negative correlation with Statement (4) ($p = .034$), i.e., not reusing photographs of informal models. Moreover, we identified a statistically significant correlation between how often informal models are retained on a whiteboard and agreement with Statement (1) ($p = .008$), i.e., not reusing the informal model as it was not saved. In addition, we found two correlations of weak significance: The first, between the frequency of informal models being retained on white boards and agreement with Statement (2) not reusing informal models ($p = .056$). The second, between how often are transcribed using an appropriate tool and Statement (6) stating the usefulness of manual transcription of informal models ($p = .0.06$).

5 Discussion

In this section, we discuss our findings and answer our research questions. Moreover, we discuss how we avoided threats to validity. Furthermore, we reassess the presented correlations to address potential multiple correlation biases.

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. Fig. 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. Fig. 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. Fig. 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 toward the importance of the topic, while other participants did not consider these purposes important.

5.1.4 RQ1.4: How do these purposes change concerning different phases of automotive development?

For the correlation between the answers for relevant purposes and development phases of DTs (cf. Fig. 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 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*. As 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 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, and provisioned, and their consistency be ensured. *PREEvision* is a model-driven development environment[50] 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 [24], 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 RQ3: How relevant are informal models in the automotive industry?

5.3.1 RQ3.1: For which purpose are informal models used?

To understand why and how informal models are used, stored and reused, we first investigated which purpose informal models fulfill. We found that they often used for communication, documentation, the design of DTs as well as requirements analysis. However, it is evident that most participants use informal models for communication and documentation. This is in line with findings about software development processes [14]. Moreover, as participants named additional purposes, e.g., *modeling of development processes*, we conclude that these purposes are not exhaustive.

5.3.2 RQ3.2: For which phases of automotive development are informal models employed?

Regarding the phases of automotive development, we found that informal models are most often utilized during the conception and development phase. While we observe that informal models were also regularly employed in other phases of automative development, most particpants rather rarely employ them during the production, usage & support, and retirment phase. We argue that this is due to the way informal model can help participants to cost efficiently discuss and capture early design decisions. Thus, we see a chance in integrating informal models into digital twins to support the early development phases and to make documented design decisions available the whole life cycle of the DT.

5.3.3 RQ3.3: How are informal models persisted?

As informal models were mostly employed for communication and documentation, we surveyed how participants recorded them for later examination or revision. Surprisingly, we found that informal models were most frequently created using digital tools, e.g., PowerPoint™, and thus could be stored directly. Moreover, photographs of informal models sketched on flipcharts or whiteboards were still the second frequent option for persisting informal models. Although this observation is in contrast to past research on the use of informal models in software development [7] showing that most informal models are created using pen and paper or whiteboards, we argue this is due to the increased frequency of online meetings of engineers working in different locations. Granted, this could be a temporary shift following the COVID-19 pandemic. Notably though, some participants still manually transcribe informal models or simply keep them on the flipchart or whiteboard, although most participants *rather rarely* do either. In conclusion, DTs should be able to support the integration of digitally created informal models as well as digital images (e.g., a photograph or trace of a digital pen) of informal models. While the former could be easy, the latter would require image recognition pipelines, like [33], tailored to the different model kinds typically sketched.

5.3.4 RQ3.4: Which kinds of models are informally designed?

To investigate which model kinds are actually sketched by participants, we tasked them to select or name those model kinds. We expected that participants would name equally many structural model kinds, such as *E/E architecture models* or *data schemas*, as behavioral model kinds, such as *activity diagrams* or *business process models*. Yet, we found that behavioral model kinds were named significantly more often, i.e., *activity diagrams* and *state diagrams* followed by *business process models*. We conclude that this might be due to the emphasis on an automotive's behavior or business processes around the automotive during the *conception* phase. Notably, the number of different model kinds sketched by participants (Fig. 11b) is comparable to the number of different model kinds designed (Fig. 6b), although some participants stated to never draw informal models. In sum, a wide variety of model kinds are sketched, even though behavioral model kinds are more common. In turn, developers of DTs should start by supporting the integration of the most common behavioral model kinds sketched by practitioners in the automotive industry, i.e., *activity diagrams*, *state diagrams*, and *business process models*.

5.3.5 RQ3.5: How are informal models re-/used?

In our final question, we investigated how informal models are reused (after they were drawn and persisted) by means of seven statements. While most participants rather disagree with statements regarding not reusing informal models, most agree with the statement that the *manual transcription of an informal model is time-consuming yet is considered useful*. Additionally, a participant described the manual transcription as a *hurdle*, if the modeling tool is rarely used. In sum, we see a need for the automatic transcription of informal models, e.g., by means of image recognition. Besides that, from the agreement with Statement (3) and disagreement with Statement (4), we conclude that *if an informal model is photographed, it is more likely to be reused*.

From the correlations between the phase of development and the presented statements on reuse (cf. Fig. 13), we conclude that participants who frequently employ informal models in the conception phase usually reuse and save their models. In addition, we conclude that participants employing informal models in later development phases (except retirement) would not reuse photographed models but rather a transcribed model. Likewise, from the correlations between the way informal models are stored and the statements on reuse (cf. Fig. 14), we entail that informal models only retained on a whiteboard or flipchart are rarely reused. Surprisingly, whenever participants have created informal models digitally, they are more likely to reuse and transcribe them. Moreover, we imply that participants who often transcribed informal models also used a photograph of an informal model as reference, yet some do not consider this task to be useful. An experienced reader might interject that it is a common approach to sketch the most important concepts and relations on a whiteboard or in PowerPoint™leaving out unnecessary details. Afterward the sketch will be modeled and thus formalized with an appropriate tool, which complains about missing details or incomplete models. While we agree that this is a very common practice, we argue that this is a shortcoming of modeling tools not allowing for incomplete models or sketches, to be later completed. Because a sketch persisted in the modeling tool, albeit incomplete, it can be synchronized and retrieved within the digital twin.

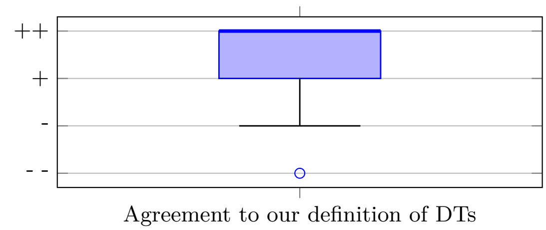


Fig. 15 Boxplot agreemant with the definition: <<*A digital twin is a software comprising models and services to purposefully represent and manipulate its original system across its complete lifecycle.*>>. The agreemant ranges from *disagree at all* (--), via *rather disagree* (−) and *rather agree* (+) to *agree at all* (++)

Answer to RQ3: The results of the survey indicate that informal models are frequently used in the automotive industry, in particular during the *conception* and *development* phase. Informal models are mostly used for the *communication* and *documentation* of design decisions. Surprisingly, most participants frequently *create informal models digitally*, e.g., using PowerPoint™, although they admit that *manually transcribing models is time-consuming*. Last but not least, participants sketch more behavioral models, most importantly, *activity diagrams* or *state diagrams*, then structural models.

5.4 Agreement to our definition of digital twins

As the last question, we asked participants to rate their agreement with our definition of DTs (cf. Sect. 2.1). Figure 15 shows that 75% of participants agree with the definition. Notably though, only one participant disagreed completely. We argue that this overall agreement stems from the fact that the definition was derived from the literature and has been discussed with experts in the field. Although we concede that this might be caused by the questions and aspects of our questionaire. However, the disagreement of one participant might indicate that our definition is not yet perfect and needs further refinement or that there are different opinions on the definition of digital twins in the automotive domain. Ultimately, we consider our definition to be a good starting point for further research on a common definition of DTs in the automotive domain.

5.5 Bonferroni correction of the presented correlations

Although the indicated correlations were meant to present an initial analysis, we concede that the significances of the presented correlations for RQ1.4 in Fig. 5 as well as for RQ3.5 in Figs. 13 and 14 are biased due to multiple correlations being considered. We applied the *Bonferroni correction* [19] to further lower the significance levels wrt. the number of correlations tested. In regard to RQ1.4 (cf. Fig. 5), we tested $m = 35$ correlations, lowering the significance level for a strong and weak significant correlation to $p < .05/35 \approx .00143$ and $p < .1/35 \approx .00286$, respectively. None of the reported p-values in Sect. 4.1.4 crossed these levels. For RQ3.5 (cf. Sect. 4.3.5), the correction of the correlation analysis presented in Fig. 14 with $m = 28$ requires a significance level of $p < .00179$ for a strong correlation and $p < .00357$ for a weak correlation. Again, the reported p-values did not cross these thresholds. In contrast, the correlation analysis presented in Fig. 13 (with $m = 35$) still indicates a strong correlation ($p < .00143$) between Statement (6) that manual transcription of informal models is useful when frequently employed for *usage & support* ($p = .0005$). Likewise, a weak correlation ($p < .0028$) is maintained between Statement (4) that photographs are not reused when they are taken during *usage & support* ($p = .0023$). Consequently, our survey only provided initial evidence for RQ1.4 and RQ3.5, yet most results are not significant under the Bonferroni correction. Please note that this does not impact the other 10 research questions (RQ1.1–1.3, RQ2.1–2.3, & RQ3.1–3.4).

5.6 Threats to validity

Our survey is subject to the four basic types of validity threats according to [61] 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 [48]. 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 before 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 original response data and a translated version of the data in English. Although the number of participants was small and their answers might not exhibit a normal distribution, according to Geoff [44], the Pearson correlation coefficient and Student's T-test are still applicable in our setting. As we were interested in initial correlations, we did not apply means to correct for false-positives of the found significances. To remedy this, in Sect. 5.5, we reassessed the three correlation analyses using the suggested Bonferoni correction. As a result, the presented correlations should be considered as initial results requiring further surveys and questionnaires to provide conclusive evidence.

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 that are representatives of the German automotive industry and, thus, could prevent our results from being generalizable to the automotive industry of other countries. Furthermore, we did not collect information on whether respondents were from academia or industry. As a result, we cannot rule out potential bias introduced by a higher proportion of academic participants.

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.

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, e.g., by not including deployment models.

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 Sect. 3, which enables replicating the methodology of this questionnaire survey. In addition, we provide the data extraction results and the original survey online.[2]

6 Related studies

This section presents related studies that survey the properties, capabilities, and goals of DTs, modeling languages and tools that are employed in automotive development, and studies that investigate informal modeling practices. For each group of related studies, we provide a structured comparison table that classifies how each related finding relates to our survey results: *Confirmed* findings are corroborated by our data, *Extended* findings are covered by our survey with additional depth or scope, and *Contrast* marks findings that differ from, or are out of scope of, our study.

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 [16] 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. Besides, a systematic mapping study investigated the state of the art in employing models in the development of digital twins in research [38]. In comparison to this work, we focused on the automotive domain and surveyed experts from industry. Another literature survey focused on the technical aspects of realizing DTs [40] 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 [10] 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. Fig. 4a). Another systematic literature review on DTs in automotive application [17] 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. Fig. 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. Another investigation presents challenges for the development of digital twins [1]. It considers only papers from the 27th International Conference on Model Driven Engineering Languages and Systems and, consequently, limited in scope. However, it identifies the derivation of UML diagrams from informal models as one challenges, which aligns with our results that providing digital twins with informal models is beneficial. Table 1 systematically classifies which of these findings are confirmed, extended, or contrasted by our results.

6.2 Surveys on modeling languages and tools in automotive development

A systematic mapping study on modeling languages for Industry 4.0 [62] 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 [39] 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 [3] showed that informal models are widely used in the embedded industry.

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 [51], consistency preservation [60], handling uncertainty [2], modelling of variants and versions [6], or novel modeling and programming languages [35]. Besides surveys covering the state of the art of modeling in general, there exist surveys for

Table 1 Comparison of related surveys on DTs with the findings of our survey

Study	Key finding	Alignment with our survey
Dalibor et al. [16]	Systematic mapping of SE for DTs: main purposes are behavior optimization, monitoring, and prediction, especially at CPS runtime	*Confirmed*: monitoring, prediction, and validation reach a median of *very important*; process optimization is also rated as an important purpose, aligning with the behavior optimization finding of Dalibor et al. (cf. Fig. 4c)
Lehner et al. [38]	Surveys model-based DT development approaches in research literature	*Extended*: we focus on the automotive domain and on practitioners rather than research literature
Lim et al. [40]	Four essential technology categories for DTs: communication, representation, computation, and microservices	*Out of scope*: technology realization was not the focus of our questionnaire, which targeted purposes and model kinds used in practice
Biesinger et al. [10]	DTs increase quality and reduce complexity in automotive production planning (n = 22)	*Confirmed*: production is the second most important development phase (cf. Fig. 4a)
Deng et al. [17]	Distinguishes DT purposes for manufacturing (monitor, optimize production) from DT-for-vehicle purposes (predict, monitor, optimize energy)	*Extended*: our study correlates specific purposes with development phases, e.g., process optimization with production and prediction with usage & support (cf. Fig. 5)
Abrahão et al. [1]	Identifies challenges in DT development; highlights derivation of UML diagrams from informal models	*Confirmed*: providing DTs with informal models is beneficial, as also reflected in our RQ3 results

model-based security testing approaches [53], variability modeling [8], and requirements modeling [52]. 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 [42].

Table 2 classifies which of these findings are confirmed, extended, or contrasted by our results.

6.3 Studies on informal modeling practices

Numerous studies have examined the use of informal models, sketches, and diagrams in software development. Cherubini et al. [14] conducted semi-structured interviews and a survey to explore how software developers use diagrams. They found that diagrams are primarily used for communication and are often transient, meaning they are used for a specific task and then discarded. This finding is consistent with our survey results, where participants discussed how they use informal models. However, our we also found that informal models are more often created digitally. Walny et al. [59] investigated the lifecycles of diagrams and sketches in software development by conducting multiple interviews with researchers. They identified various diagram lifecycles, such as brainstorming, drawing a diagram, taking a photo of it, and uploading the photo to a common wiki. Our survey also addressed how participants persist informal models, reflecting these practices. Baltes et al. [7] conducted an exploratory study with three companies and an online survey to understand the role of sketches and diagrams in software engineering. They found that informal diagrams are valuable resources that document several aspects of the software development process. This aspect of documentation was also highlighted in our survey, where participants discussed the use of informal models for documentation purposes. Störrle [55] conducted an online survey to determine whether and how conceptual modeling languages are used in software engineering practice. He observed that conceptual modeling languages like UML are used in practice, but often as informal models (70-79%). In contrast to the aforementioned studies, our research focuses on the use of informal models in the context of digital twins and the automotive industry. Table 3 classifies which of these findings are confirmed, extended, or contrasted by our results.

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

Table 2 Comparison of related surveys on modeling languages and tools with the findings of our survey

Study	Key finding	Alignment with our survey
Wortmann et al. [62]	Systematic mapping for Industry 4.0: UML variants and DSLs most applied; validation underrepresented in research despite being an important concern	*Extended*: rather than mapping research literature, we survey industry practitioners and identify the most prevalent model kinds (source code, simulation models, business process models) and tools (e.g., MATLAB/Simulink, Enterprise Architect) in the automotive domain
Liebel et al. [39]	Survey of 113 embedded-systems practitioners: models are central; MATLAB/Simulink, UML, and SysML used for simulation, code generation, and consistency checking	*Confirmed*: MATLAB/Simulink is the most frequently named tool; we extend this to automotive-specific tools and to their relevance per development phase (cf. Fig. 9)
Akdur et al. [3]	Survey in the embedded industry: informal models are widely used	*Confirmed*: informal models are frequently used in the automotive industry as well (cf. Fig. 10)

Table 3 Comparison of related studies on informal modeling with the findings of our survey

Study	Key finding	Alignment with our survey
Cherubini et al. [14]	Semi-structured interviews and survey with software developers: diagrams are used primarily for communication and are often transient.	*Confirmed*: communication and documentation are the top-rated purposes in our survey (cf. Fig. 10a). *Contrast*: automotive practitioners create informal models predominantly digitally, indicating they are less transient in our context
Walny et al. [59]	Interviews investigating diagram lifecycles: brainstorming, sketching, photographing, and uploading to a shared wiki.	*Confirmed*: photographing is common (50 % of participants rather often); however, digital-first creation dominates over whiteboard retention in our survey (cf. Fig. 10c)
Baltes et al. [7]	Exploratory study across three companies and an online survey: informal diagrams are valuable resources for documenting software development.	*Confirmed*: documentation is a frequently cited purpose of informal models in our automotive survey (cf. Fig. 10a).
Störrle [55]	Online survey: conceptual modeling languages such as UML are used in practice, mostly as informal models (70–79 %)	*Contrast*: our study focuses on the automotive domain and the context of DTs; automotive practitioners actively employ dedicated formal modeling tools alongside informal models, which differs from the predominantly informal use of UML-like notations in general software engineering

(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 [16], which, e.g., also includes knowledge representation techniques. Surprisingly, in contrast to the existing literature, UML and SysML models are not that frequently used. 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. Therefore, future studies could investigate which approaches companies leverage to cope with the integration of these tools and how information across models is persisted, e.g., by using knowledge graphs, tool APIs, exchange formats, or language extension mechanisms [13]. We also see the need for follow-up studies. Since respondents rate many aspects, future research should investigate the reasons for these opinions. Concerning the adoption of digital twins and modeling languages in practice, future studies could also investigate the most significant technological challenges and barriers that the industry faces today. Regarding informal models our results indicate that they are mostly used for communication and documentation purposes. They are particularly used during the conception phase, are mostly drawn in digital tools such as PowerPoint, and describe the behavior of the CPS. Therefore, we see an opportunity in artificial intelligence technology [33] to transform these informal models in first versions of formal models in the future. Moreover, we see major opportunities in the development of intelligent modelling assistents [43] for the automotive industry. In general, we believe that the insights gained from our questionnaire survey can help practitioners and researchers engineering automotive DTs more systematically.

Acknowledgements This work has been partly supported by the German Federal Ministry of Economic Affairs and Climate Action (BMWK, SofDCar – 19S21002L). The authors of the University of Stuttgart were partly supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) [grant number 441207927] and by the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg within the Innovations Campus Future Mobility (ICM). This work was supported by funding from the pilot program Core Informatics at KIT (KiKIT) of the Helmholtz Association (HGF) and supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - SFB 1608 - 501798263.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Abrahão, S., Staron, M., Michael, J., Combemale, B., Chechik, M.: Modeling and digital twins: insights and strategies for software engineers. IEEE Softw. **42**(2), 92–96 (2025)
2. Acosta, M., Hahner, S., Koziolek, A., Kühn, T., Mirandola, R., Reussner, R.: 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), Montreal, Kanada, pp. 569–578 (2022)
3. Akdur, D., Garousi, V., Demirörs, O.: A survey on modeling and model-driven engineering practices in the embedded software industry. J. Syst. Archit. **91**(2018), 62–82 (2018)
4. Alves, R.G., Souza, G., Maia, R.F., Lan Ho Tran, A., Kamienski, C., Soininen, J., Aquino, P.T., Lima, F.: A digital twin for smart farming. In: IEEE Global Humanitarian Technology Conference GHTC, pp. 1–4. IEEE, Seattle, WA, USA (2019)
5. Amrani, M., Blouin, D., Heinrich, R., Rensink, A., Vangheluwe, H., Wortmann, A.: Multi-paradigm modelling for cyber-physical systems: a descriptive framework. Softw. Syst. Model. **2021**, 1–29 (2021)
6. Ananieva, S., Greiner, S., Kehrer, T., Krüger, J., Kühn, T., Linsbauer, L., Grüner, S., Koziolek, A., Lönn, H., Ramesh, S., Reussner, R.: A conceptual model for unifying variability in space and time: rationale, validation, and illustrative applications. Empir. Softw. Eng. **27**(5), 101 (2022)
7. Baltes, S., Diehl, S.: Sketches and diagrams in practice. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 530–541. ACM, Hong Kong China (2014)
8. Berger, T., Rublack, R., Nair, D., Atlee, J.M., Becker, M., Czarnecki, K., Wasowski, A.: A survey of variability modeling in industrial practice. In: Gnesi, S., Collet, P., Schmid, K. (eds.) The Seventh International Workshop on Variability Modelling of Software-intensive Systems: VaMoS, p. 7:1-7:8. ACM, Pisa, Italy (2013)
9. Bibow, P., Dalibor, M., Hopmann, C., Mainz, B., Rumpe, B., Schmalzing, D., Schmitz, M., Wortmann, A.: Model-driven development of a digital twin for injection molding. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) International Conference on Advanced Information Systems Engineering (CAiSE'20) (Lecture Notes in Computer Science), vol. 12127, pp. 85–100. Springer, Grenoble, France (2020)
10. Biesinger, F., Kraß, B., Weyrich, M.: A survey on the necessity for a digital twin of production in the automotive industry. In: 2019 23rd International Conference on Mechatronics Technology (ICMT), pp. 1–8. IEEE, Salerno, Italy (2019)
11. Bolender, T., Bürvenich, G., Dalibor, M., Rumpe, B., Wortmann, A.: Self-Adaptive Manufacturing with Digital Twins. In: 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (Madrid. Spain) (SEAMS '21), pp. 1–4. Association for Computing Machinery, New York, NY, USA (2021)

12. Brockhoff, T., Heithoff, M., Koren, I., Michael, J., Pfeiffer, J., Rumpe, B., Uysal, M.S., Van Der Aalst, W.M.P., Wortmann, A.: Process prediction with digital twins. In: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 182–187. IEEE, Fukuoka, Japan (2021)
13. Butting, A., Haber, A., Hermerschmidt, L., Kautz, O., Rumpe, B., Wortmann, A.: Systematic language extension mechanisms for the MontiArc architecture description language. In: European Conference on Modelling Foundations and Applications (ECMFA'17) (Marburg) (LNCS 10376), pp. 53–70. Springer (2017)
14. Cherubini, M., Venolia, G., DeLine, R., Ko, A.J.: Let's go to the whiteboard: how and why software developers use drawings. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 557–566. ACM, San Jose California USA (2007)
15. Dalibor, M., Heithoff, M., Michael, J., Netz, L., Pfeiffer, J., Rumpe, B., Varga, S., Wortmann, A.: Generating customized low-code development platforms for digital twins. J. Comput. Lang. **70**(2022), 101117 (2022)
16. Dalibor, M., Jansen, N., Rumpe, B., Schmalzing, D., Wachtmeister, L., Wimmer, M., Wortmann, A.: A cross-domain systematic mapping study on software engineering for Digital Twins. J. Syst. Softw. **193**(2022), 111361 (2022)
17. Deng, S., Ling, L., Zhang, C., Li, C., Zeng, T., Zhang, K., Guo, G.: A systematic review on the current research of digital twin in automotive application. Internet Things Cyber Phys. Syst. **3**(2023), 180–191 (2023)
18. Digital Twin Consortium: Capabilities Periodic Table. https://www.digitaltwinconsortium.org/initiatives/capabilities-periodic-table/ Last accessed: 2023-12-30 (2022)
19. Dunn, O.J.: Multiple comparisons among means. J. Am. Stat. Assoc. **56**(293), 52–64 (1961). https://doi.org/10.1080/01621459.1961.10482090
20. Eramo, R., Bordeleau, F., Combemale, B., van den Brand, M., Wimmer, M., Wortmann, A.: Conceptualizing digital twins. IEEE Softw. **39**(2), 39–46 (2022)
21. Eramo, R., Bordeleau, F., Combemale, B., van Den Brand, M., Wimmer, M., Wortmann, A.: Conceptualizing digital twins. IEEE Softw. **39**(2), 39–46 (2021)
22. Fuchß, D., Kühn, T., Wortmann, A., Pfeiffer, J., Koziolek, A.: An expert survey on the use of informal models in the automotive industry. In: Karlsruher Institut für Technologie (KIT). ECSA TwinArch Workshop, (2023). (Technical Report)
23. Fur, S., Heithoff, M., Michael, J., Netz, L., Pfeiffer, J., Rumpe, B., Wortmann, A.: Sustainable digital twin engineering for the internet of production. In: Digital Twin Driven Intelligent Systems and Emerging Metaverse, pp. 101–121. Springer, Singapore (2023)
24. Fürst, S., Bechter, M.: 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), pp. 215–217. IEEE, Toulouse, France (2016)
25. Götz, S., Fehn, A., Rohde, F., Kühn, T.: Model-driven software engineering for construction engineering: Quo Vadis? J. Object Technol. **19**(2), 1–22 (2020)
26. Hasselbring, W.: Software architecture: past, present, future. In: Gruhn, V., Striemer, R. (eds.) The Essence of Software Engineering, pp. 169–184. Springer (2018)
27. Hua, E.Y., Lazarova-Molnar, S., Francis, D.P.: Validation of digital twins: challenges and opportunities. In: 2022 Winter Simulation Conference (WSC), pp. 2900–2911. IEEE, Singapore (2022)
28. Huebner, K.H., Dewhirst, D.L., Smith, D.E., Byrom, T.G.: The Finite Element Method for Engineers, Wiley (2001)
29. ISO/DIS: ISO/DIS 23247 Automation Systems and Integration-Digital Twin Framework for Manufacturing. Technical Report. International Standardization Organization (ISO) (2020)
30. Jolak, R., Vesin, B., Chaudron, M.R.V.: Using Voice Commands for UML Modelling Support on Interactive Whiteboards: Insights and Experiences. In: Pastor, O., Guizzardi, R.S.S., Kaplan, G.N., Molina, J.G., Moreira, A., Brito, I.S., de la Vara, J.L., Genero, M., da Silva, F.Q.B., Cernuzzi, L. (eds.) Proceedings of the XX Iberoamerican Conference on Software Engineering, Buenos Aires, Argentina, May 22–23, 2017, pp. 85–98. Curran Associates (2017)
31. Kirchhof, J.C., Michael, J., Rumpe, B., Varga, S., Wortmann, A.: 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, pp. 90–101. ACM, Virtual Event, Canada (2020)
32. Knapp, G.L., Tuhin Mukherjee, J.S., Zuback, H.L.W., Palmer, T.A., De, A., DebRoy, T.J.A.M.: Building blocks for a digital twin of additive manufacturing. Acta Mater. **135**(2017), 390–399 (2017)
33. Koziolek, H., Koziolek, A.: LLM-based control code generation using image recognition. In: LLM4Code '24: Proceedings of the 1st International Workshop on Large Language Models for Code (LLM4 Code), Lissabon, 20th, pp. 38–45. Association for Computing Machinery (ACM) (2024)
34. Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W.: Digital Twin in manufacturing: a categorical literature review and classification. Ifac-PapersOnline **51**(11), 1016–1022 (2018)
35. Kühn, T., Leuthäuser, M., Götz, S., Seidl, C., Aßmann, U.: A metamodel family for role-based modeling and programming languages. In: Software Language Engineering, pp. 141–160. Springer, Cham (2014)
36. Lauzeral, N., Borzacchiello, D., Kugler, M., George, D., Rémond, Y., Hostettler, A., Chinesta, F.: 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 (2019)
37. Lehner, D., Pfeiffer, J., Tinsel, E.-F., Strljic, M.M., Sint, S., Vierhauser, M., Wortmann, A., Wimmer, M.: Digital twin platforms: requirements, capabilities, and future prospects. IEEE Softw. **39**(2), 53–61 (2022)
38. Lehner, D., Zhang, J., Pfeiffer, J., Sint, S., Splettstößer, A.-K., Wimmer, M., Wortmann, A.: Model-driven engineering for digital twins: a systematic mapping study. Softw. Syst. Model. **24**(5), 1339–1377 (2025)
39. Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J.: Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice. Softw. Syst. Model. **17**(1), 91–113 (2018)
40. Lim, K.Y.H., Zheng, P., Chen, C.-H.: A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. J. Intell. Manuf. **31**(6), 1313–1337 (2020)
41. Magargle, R., Johnson, L., Mandloi, P., Davoudabadi, P., Kesarkar, O., Krishnaswamy, S., Batteh, J., Pitchaikani, A.: A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system. In: Jirí Kofránek and Francesco Casella (eds.) Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017 (Linköping Electronic Conference Proceedings, Vol. **132**), pp. 132:003. Linköping University Electronic Press, Prague, Czech Republic(2017)
42. Mussbacher, G., Combemale, B., Abrahão, S., Bencomo, N., Burgueño, L., Engels, G., Kienzle, J., Kühn, T., Mosser, S., Sahraoui, H., Weyssow, M.: Towards an assessment grid for intelligent modeling assistance. In: Proceedings of the 23rd ACM/IEEE Inter-

national 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 (2020a)
43. Mussbacher, G., Combemale, B., Kienzle, J., Abrahão, S., Ali, H., Bencomo, N., Búr, M., Burgueño, L., Engels, G., Jeanjean, P.: Opportunities in intelligent modeling assistance. Softw. Syst. Model. **19**, 1045–1053 (2020)
44. Norman, G.: Likert scales, levels of measurement and the "laws" of statistics. Adv. Health Sci. Educ. **15**(2010), 625–632 (2010)
45. Pfeiffer, J., Fuchß, D., Kühn, T., Liebhart, R., Neumann, D., Neimöck, C., Seiler, C., Koziolek, A., Wortmann, A.: Modeling languages for automotive digital twins: a survey among the German automotive industry. In: Egyed, A., Wimmer, M., Chechik, M., Combemale, B. (eds.) Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, MODELS 2024, Linz, Austria, September 22-27, 2024, pp. 92–103. ACM (2024)
46. Pfeiffer, J., Lehner, D., Wortmann, A., Wimmer, M.: Modeling capabilities of digital twin platforms—old wine in new bottles? J. Obj. Technol. **21**(3), 1–14 (2022). The 18th European Conference on Modelling Foundations and Applications (ECMFA 2022)
47. Rajesh, P.K., Manikandan, N., Ramshankar, C.S., Vishwanathan, T., Sathishkumar, C.: Digital twin of an automotive brake pad for predictive maintenance. Procedia Comput. Sci. **165**, 18–24 (2019)
48. Ray, J.J.: The construct validity of balanced Likert scales. J. Soc. Psychol. **118**(1), 141–142 (1982)
49. Samuelsson, S.G., Book, M.: Towards sketch-based user interaction with integrated software development environments. In: ICSE '20: 42nd International Conference on Software Engineering, pp. 181–184. ACM, Workshops, Seoul, Republic of Korea (2020)
50. Schäuffele, J.: E/E architectural design and optimization using PREEvision. Technical Report, SAE Technical Paper (2016)
51. Schindewolf, M., Wittler, J.W., Kühn, T., Grimm, D., Sax, E.: A Model-Based Approach to Automotive Feature Development for Updates and Upgrades. In: International Conference on Service-Oriented System Engineering SOSE, pp. 19–26. IEEE, Athens, Greece (2023)
52. Sepúlveda, S., Cravero, A., Cachero, C.: Requirements modeling languages for software product lines: a systematic literature review. Inf. Softw. Technol. **69**, 16–36 (2016)
53. Sommer, F., Kriesten, R., Kargl, F.: Survey of model-based security testing approaches in the automotive domain. IEEE Access **11**, 55474–55514 (2023)
54. Störrle, H.: How are conceptual models used in industrial software development? A descriptive survey. In: Mendes, E., Counsell, S., Petersen, K. (eds.) Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE 2017, Karlskrona, Sweden, June 15–16, 2017, pp. 160–169. ACM (2017)
55. Störrle, H.: How are conceptual models used in industrial software development? A descriptive survey. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, pp. 160–169. ACM, Karlskrona Sweden (2017)
56. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F.: Digital twin-driven product design, manufacturing and service with big data. Int. J. Adv. Manuf. Technol. **94**, 3563–3576 (2018)
57. Tao, F., Qi, Q., Wang, L., Nee, A.Y.C.: Digital twins and cyber-physical systems toward smart manufacturing and industry 4.0: correlation and comparison. Engineering **5**(4), 653–661 (2019)
58. Vesin, B., Jolak, R., Chaudron, M.R.V.: OctoUML: an environment for exploratory and collaborative software design. In: Uchitel, S., Orso, A., Robillard, M.P. (Eds.). Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20–28, 2017—Companion Volume, pp. 7–10. IEEE Computer Society (2017)
59. Walny, J., Haber, J., Dork, M., Sillito, J., Carpendale, S.: Follow that sketch: lifecycles of diagrams and sketches in software development. In: 2011 6th International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT), pp. 1–8. IEEE, Williamsburg, VA, USA (2011)
60. Wittler, J.W., Saglam, T., Kühn, T.: Evaluating model differencing for the consistency preservation of state-based views. J. Object Technol. **22**(2), 1–14 (2023)
61. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Berlin (2012)
62. Wortmann, A., Barais, O., Combemale, B., Wimmer, M.: Modeling languages in industry 4.0: an extended systematic mapping study. Softw. Syst. Model. **19**(1), 67–94 (2020)
63. Wüest, D., Seyff, N., Glinz, M.: FlexiSketch: a lightweight sketching and metamodeling approach for end-users. Softw. Syst. Model. **18**(2), 1513–1541 (2019)
64. Xu, Q., Ali, S., Yue, T.: Digital twin-based anomaly detection in cyber-physical systems. In: 14th IEEE Conference on Software Testing, Verification and Validation (ICST), pp. 205–216. IEEE, Porto de Galinhas, Brazil (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jérôme Pfeiffer is a research assistant and PhD candidate at the Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW) of the University of Stuttgart. His research interests are software language engineering, digital twins, and artificial intelligence. For more information, please visit: https://www.isw.uni-stuttgart.de/en/institute/team/Pfeiffer-00005/

Dominik Fuchß is a researcher and PhD candidate at the KASTEL - Institute of Information Security and Dependability of the Karlsruhe Institute of Technology (KIT). His research interests are traceability link recovery, consistency between formal and informal software artifacts, and large language models for software engineering. For more information, please visit: https://fuchss.org/

Thomas Kühn is a tenured postdoc at the Software-Engineering and Programming Languages group of the Martin-Luther-Universität Halle-Wittenberg. His research focuses on new ways to model and program future software systems with their increased complexity, heterogeneity, distribution, rate of change, and longev ity. In pursuit of this goal, he developed a family of role-based modelling and a family of role-oriented programming languages supported by a feature-aware modelling editor and a basic IDE, respectively. Following these topics, he aims to bridge the gap between classic compiler construction and software product line engineering, leading to the notion of language product line engineering. Aside from that, he is interested in good practices in scientific research and methods to improve his daily work process. thomas.kuehn@informatik.uni-halle.de https://swt.informatik.uni-halle.de/mitarbeiter/45865_3467283/

Dirk Neumann is a research assistant and PhD candidate at the Karlsruhe Institute of Technology (KIT). His research focuses on variability modeling and model-driven engineering, with an emphasis on consistency management in multi-domain, variant-rich systems. For more information, please visit: https://orcid.org/0000-0003-1999-639X

Christer Neimöck is Senior Manager of Business Development and Programs for Advanced Driver Assistance Systems (ADAS) and Automated Driving (AD) at T-Systems, where he helps custom ers bring the software-defined vehicle and autonomous mobility to life.

Anne Koziolek is a full professor and head of the Modelling for Continuous Software Engineering (MCSE) group at the KASTEL - Institute of Information Security and Dependability of the Karlsruhe Institute of Technology (KIT). Her research interests are software architecture, model-based design space exploration, model consistency and the consistency of informal views on software systems, and large language models for software engineering. For more information, please visit: https://mcse.kastel.kit.edu/staff_anne_koziolek.php

Andreas Wortmann is head of the chair "Model-Driven Engineering for Manufacturing Automation" at the Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW) of the University of Stuttgart. He conducts research on model-driven engineering, software language engineering, and systems engineering with a focus on Industry 4.0 and digital twins. For more information, please visit: www.wortmann.ac.