Innovations in Systems and Software Engineering
https://doi.org/10.1007/s11334-022-00442-w

S.I. : SYSTEMS MODELLING AND MANAGEMENT

Replicability of experimental tool evaluations in model-based software and systems engineering with MATLAB/Simulink

Alexander Boll[1] · **Nicole Vieregg**[2] · **Timo Kehrer**[1]

Received: 25 January 2021 / Accepted: 4 February 2022

Abstract
Research on novel tools for model-based development differs from a mere engineering task by not only developing a new tool, but by providing some form of evidence that it is effective. This is typically achieved by experimental evaluations. Following principles of good scientific practice, both the tool and the models used in the experiments should be made available along with a paper, aiming at the replicability of experimental results. We investigate to which degree recent research reporting on novel methods, techniques, or algorithms supporting model-based development with MATLAB/Simulink meets the requirements for replicability of experimental results. Our results from studying 65 research papers obtained through a systematic literature search are rather unsatisfactory. In a nutshell, we found that only 31% of the tools and 22% of the models used as experimental subjects are accessible. Given that both artifacts are needed for a replication study, only 9% of the tool evaluations presented in the examined papers can be classified to be replicable in principle. We found none of the experimental results presented in these papers to be fully replicable, and 6% partially replicable. Given that tools are still being listed among the major obstacles of a more widespread adoption of model-based principles in practice, we see this as an alarming signal. While we are convinced that this situation can only be improved as a community effort, this paper is meant to serve as starting point for discussion, based on the lessons learnt from our study.

Keywords Model-based development · MATLAB/Simulink · Tools and techniques · Experimental evaluation · FAIR principles · Replicability · Systematic literature review

1 Introduction

Model-based development [1,2] is a much appraised and promising methodology to tackle the complexity of modern software-intensive systems, notably for embedded systems in various domains such as transportation, telecommunications, or industrial automation [3]. It promotes the use of models in all stages of development as a central means for abstraction and starting point for automation, e.g., for the sake of simulation, analysis or software production, with the ultimate goal of increasing productivity and quality.

Consequently, model-based development strongly depends on good tool support to fully realize its manifold promises [4]. Research on model-based development often reports on novel methods and techniques for model management and processing which are typically embodied in a tool. In addition to theoretical and conceptual foundations, some form of evidence is required concerning the effectiveness of these tools, which typically demands for an experimental evaluation [5–7].

In turn, experimental results should be replicable[1] in order to increase the validity and reliability of the out-

This work has been supported by the German Ministry of Research and Education (BMBF) under grant 01IS18091B in terms of the research project *SimuComp*.

✉ Alexander Boll
alexander.boll@inf.unibe.ch

Nicole Vieregg
nicole.vieregg@uni-flensburg.de

Timo Kehrer
timo.kehrer@inf.unibe.ch

[1] Software Engineering Group, University of Bern, Bern, Switzerland

[2] Special Education in the Field of Learning, Europa-Universität, Flensburg, Germany

[1] Concerning the different usages of the term replicability throughout the literature, we adopt the one which has been characterized as "B2" by Barba [8], where *replicability* refers to a different team arriving at

Published online: 25 March 2022

comes observed in an experiment [9,10]. Therefore, both the tool and the experimental subject data, essentially the models used in the experiments, should be made available following the so-called FAIR principles—Findability, Accessibility, Interoperability, and Reusability [11,12]—aiming at the replicability of experimental results.

In this paper, we investigate to which degree recent research on tools for model-based development of embedded systems meets the requirements for replicability of experimental results. We focus on tools for MATLAB/Simulink (referred to as Simulink, for short), which has emerged as a de facto standard for automatic control and digital signal processing. In particular, we strive to answer the following research questions:

RQ1: Are the experimental results of evaluating tools supporting model-based development with Simulink replicable?
RQ2: From where do researchers acquire Simulink models used as experimental subjects and what are their basic characteristics?
RQ3: Does the replicability of experimental results correlate with the impact of a paper?

We conduct a systematic literature review in order to compile a list of relevant papers from which we extract and synthesize the data to answer these research questions. Starting from an initial set of 942 papers that matched our search queries on the digital libraries of IEEE, ACM, ScienceDirect and dblp, we identified 65 papers which report on the development and evaluation of a tool supporting Simulink, and for which we did an in-depth investigation. Details of our research methodology, including the search process, paper selection and data extraction, are presented in Sect. 2.

In a nutshell, we found that models used as experimental subjects and prototypical implementations of the presented tools, both of which are essential for replicating experimental results, are accessible for only a minor fraction (namely 22% and 31%) of the investigated papers. We further found the results for none of these papers to be fully replicable (RQ1), achieving only partial replicability for 6%. The models come from a variety of sources, e.g., from other research papers, industry partners of a paper's authors, open source projects, or examples provided along with Simulink or any of its toolboxes. Interestingly, the smallest fraction of models (only 3%) is obtained from open-source projects, and the largest one (about 18%) is provided by industrial partners (RQ2). While we think that, in general, the usage of industrial models strengthens the validity of experimental results, such models are often not publicly available due to confidentiality agreements. These findings are confirmed by other research papers which we investigated during our study. Finally, we found papers having a better replicability also being cited more often (RQ3), confirming results of [13,14]. Our results are presented in detail in Sect. 3.

While we do not claim our results to represent a complete image of how researchers adopt the FAIR principles of good scientific practice and deal with the replicability of experimental results in our field of interest (see Sect. 4 for a discussion of major threats to validity), we see our findings as an alarming signal. Given that tools are still being listed among the major obstacles of a more widespread adoption of model-based principles in practice [15], we need to overcome this "replicability problem" in order to make scientific progress. We are strongly convinced that this can only be achieved as a community effort. The discussion in Sect. 5 is meant to serve as a starting point for this, primarily based on the lessons learnt from our study. Finally, we review related work in Sects. 6 and 7 concludes the paper.

This paper is a revised version of our previous conference paper [16], providing the following extensions:

- In our previous work, we only investigated the principal replicability of the experimental results presented in the research papers compiled by our systematic literature search. In this extended version, for each research paper which was deemed to be replicable in principle, we follow up with an actual attempt of replicating the results in terms of a replicability study (C1.4).
- In the extended version, we are not only interested in the origins of the models used as experimental subjects, but also in their basic characteristics such as size and active maintenance span (C2.3).
- We answer the new RQ3, which analyzes whether the replicability of experimental results correlates with the impact of a paper. The rationale behind this is to seek evidence whether carefully dealing with replicability of experimental results may positively influence the impact of the underlying research.

2 Research methodology

We conduct a systematic literature review in order to compile a list of relevant papers from which we extract the data to answer our research questions RQ1 and RQ2. Our research methodology is based on the basic principles described by Kitchenham [17]. Details of our search process and research paper selection are described in Sect. 2.1. To answer RQ3, we use the relevant papers found by the systematic literature

Footnote 1 continued
the same results using the original authors' artifacts (as opposed to *reproducibility* which refers to independent researchers arriving at the same results using their own data and methods, and which is out of the scope of this paper).

Fig. 1 Digital libraries and corresponding search strings used to obtain an initial selection of research papers

IEEE: ("Abstract":Simulink OR "Abstract":Stateflow) AND ("Abstract":model) AND ("Abstract":evaluat* OR "Abstract":experiment* OR "Abstract":"case study") AND ("Abstract":tool OR "Abstract":program OR "Abstract":algorithm)

ACM: [[Abstract: simulink] OR [Abstract: stateflow]] AND [[Abstract: evaluat*] OR [Abstract: experiment*] OR [Abstract: "case study"]] AND [[Abstract: tool] OR [Abstract: program] OR [Abstract: algorithm]] AND [Abstract: model]

ScienceDirect: (Simulink OR Stateflow) AND (evaluation OR evaluate OR experiment OR "case study") AND (tool OR program OR algorithm)

dblp: (Simulink | Stateflow) (model (tool | program | algorithm | method))

review and how replicable we found them in RQ1. This is used to compare replicability of a paper to its impact. Section 2.2 is dedicated to our data extraction policy, structured along a refinement of our overall research questions.

2.1 Search process and research paper selection

2.1.1 Scope

We focus on research papers in the context of model-based development that report on the development of novel methods, techniques or algorithms for managing or processing Simulink models. Ultimately, we require that these contributions are prototypically implemented within a tool whose effectiveness has been evaluated in some form of experimental validation. Tools we consider to fall into our scope are supporting typical tasks in model-based development, such as code generation [18] or model transformation [19], clone detection [20], test generation [21] and prioritization [22], model checking [23] and validation [24], model slicing [25] and fault detection [26]. On the contrary, we ignore model-based solutions using Simulink for solving a specific problem in a particular domain, such as solar panel array positioning [27], motor control [28], or wind turbine design [29].

2.1.2 Databases and search strings

As illustrated in Fig. 2, we used the digital libraries of *ACM*,[2] *IEEE*,[3] *ScienceDirect*,[4] and *dblp*[5] to obtain an initial selection of research papers for our study. These platforms are highly relevant in the field of model-based development and were used in systematic literature reviews on model-based development like [30] or [31]. By using these four different digital libraries, we are confident to capture a good snapshot of relevant papers.

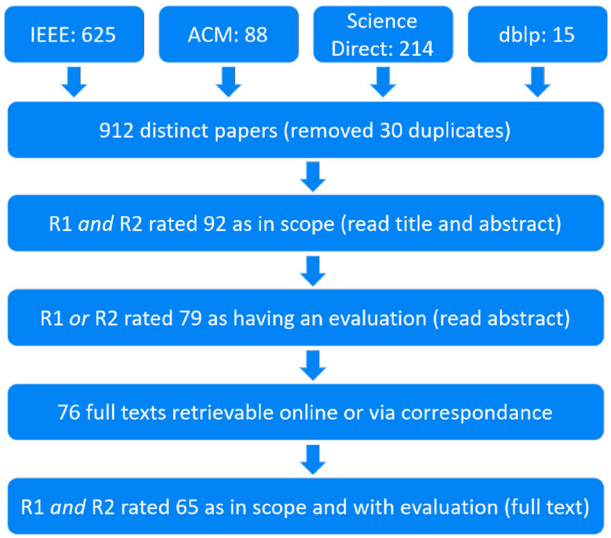

Fig. 2 Overview of the search process and research paper selection. Numbers of included research papers are shown for each step. After the initial query results obtained from the digital libraries of *ACM*, *IEEE*, *ScienceDirect* and *dblp*, the study has been performed by two researchers, referred to as R1 and R2

According to the scope of our study, we developed the search strings shown in Fig. 1. We use IEEE's and ACM's search feature to select publications based on keywords in their abstracts. Some of the keywords are abbreviated using the wildcard symbol (⋆). Since the wildcard symbol is not supported by the query engine of ScienceDirect[32], we slightly adapted these search strings for querying ScienceDirect. The same applies to dblp [32], where we also included the keyword "method" to obtain more results. To compile a contemporary and timely representation of research papers, we filtered all papers by publication date and keep those that were published between January 1, 2015, and February 24, 2020. With these settings, we found 625 papers on IEEE, 88 on ACM, 214 on ScienceDirect[6] and 15 on dblp.

[2] https://dl.acm.org.

[3] https://ieeexplore.ieee.org/Xplore/home.jsp.

[4] https://www.sciencedirect.com.

[5] https://dblp.uni-trier.de.

[6] ScienceDirect presented an initial selection of 217 papers on their web interface, out of which 214 could be downloaded.

Using the bibliography reference manager JabRef,[7] these 942 papers were first screened for clones. Then, we sorted the remaining entries alphabetically and deleted all duplicates sharing the same title. As illustrated in Fig. 2, 912 papers remained after the elimination of duplicates.

2.1.3 Inclusion and exclusion criteria

From this point onwards, the study was performed by two researchers, referred to R1 and R2 in the remainder of this paper (cf. Fig. 2).

Of the 912 papers (all written in English), R1 and R2 read title and abstract to see whether they fall into our scope. Both researchers had to agree on a paper being in scope in order to include it. R1 and R2 classified 92 papers to be in scope, with an inter-rater reliability, measured in terms of Cohen's kappa coefficient [17,33], at 0.86. To foster a consistent handling, R1 and R2 classified the first 20 papers together in a joint session, and reviewed differences after 200 papers again.

Next, R1 and R2 read the abstracts and checked whether a paper mentions some form of evaluation of a presented tool. Because such hints may be only briefly mentioned in the abstract, we included papers where either R1 or R2 gave a positive vote. As a result of this step, the researchers identified 79 papers to be in scope and with some kind of evaluation.

We then excluded all papers for which we could not obtain the full text. Our university's subscription and the social networking site ResearchGate[8] could provide us with 45 full text papers. In addition, we found 5 papers on personal pages and obtained 28 papers in personal correspondence. We did not manage to get the full text of 3 papers in one way or the other. In sum, 76 papers remained after this step.

Finally, we read the full text to find out whether there was indeed an evaluation, as indicated in the abstract, and whether Simulink models were used in that evaluation. We excluded 10 full papers without such an evaluation and one short paper which we considered to be too unclear about their evaluation. For this last step R1 and R2 resolved all differences in classification: concerning papers were read a second time, to decide together about their inclusion or exclusion. We did this so that R1 and R2 could work with one consistent set for the data extraction. After all inclusion and exclusion steps, R1 and R2 collected 65 papers which were to be analyzed in detail in order to extract the data for answering our research questions.

2.2 Refinement of research questions and data extraction

In order to answer our research questions, R1 and R2 extracted data from the full text of all the 65 papers selected in the previous step. To that end, we refined our overall research questions into criteria which are supposed to be answered in a straightforward manner, typically by a classification into "yes", "no", or "unsure".

2.2.1 RQ1: Are the experimental results of evaluating tools supporting model-based development with Simulink replicable?

To answer RQ1, we start with an investigation of the accessibility of models and tools, which are basic prerequisites for replicating experimental results, followed by full replication studies provided these basic prerequisites are fulfilled.

Accessibility of the models We assume that the effectiveness of a tool supporting model-based development can only be evaluated using concrete models serving as experimental subjects. These subjects, in turn, are a necessary precondition for replicating experimental results. They should be accessible as a digital artifact for further inspection. In terms of Simulink, this means that a model should be provided as a `*.mdl` or `*.slx` file. Models that are only depicted in the paper may be incomplete, e.g., due to parameters that are not shown in the main view of Simulink, sub-systems which are not shown in the paper, etc.

The aim of C1.1 is to find out whether all models which are required for the sake of replication are accessible:

C1.1: Are all models accessible?

The accessibility of models can only be checked if the paper provided us some hint of how to access them. For a given paper, we thus read the evaluation section of a paper closely, and also looked at footnotes, the bibliography as well as at the very start and end of the paper. In addition, we did a full text search for the keywords "download", "available", "http" and "www.". Next we checked whether given links indeed worked and models could be found, there. For all papers, a positive answer to C1.1 requires that each of the models used in the paper's evaluation falls into one of the following categories:

- There is a non-broken hyperlink to an online resource where the Simulink model file can be obtained from.
- There is a known model suite or benchmark comprising the model, such as the example models provided by Simulink.

[7] https://www.jabref.org.

[8] https://www.researchgate.net.

– The model is taken from another, referenced research paper. In this case, we assume it to be accessible without checking the original paper.

Accessibility of the tool Next to the models, the actual tool being presented in the research paper typically serves as the second input to replicate the experimental results. In some cases, however, we expect that the benefits of a tool can be shown "theoretically", i.e., without any need for actually executing the tool. To that end, before dealing with accessibility issues, we assess this general need in C1.2:

C1.2: Is the tool needed for the evaluation?

We read the evaluation section to understand whether there is the need to execute the tool in order to emulate the paper's evaluation. For those papers for which C1.2 is answered by "yes", we continue with C1.3. All papers answered with "no" are treated as if they provide their tool in C1.3.

Similarly to our investigation of the accessibility of models, we also assess if a paper provides access to their presented tool:

C1.3: Is the tool accessible?

In contrast to the accessibility of models, which we assume to be described mostly in the evaluation section, we expect that statements on the accessibility of a tool being presented in a given research paper may be spread across the entire paper. This means that the information could be "hidden" anywhere in the paper, without us being able to find it in a screening process. To decrease oversights, we did full text searches, for the key words "download", "available", "http" and "www.". If a tool was named in the paper, we also did full text searches for its name. A tool was deemed accessible if a non-broken link to some download option was provided. If third-party tools are being required, we expected some reference on where they can be obtained. We considered Simulink or any of its toolboxes as pre-installed and thus accessible by default.

Replicability studies For those papers where we determined the models and tools to be publicly available, we also investigate whether the experiments are actually replicable by us or not:

C1.4: Are the experiments replicable?

Following the general meaning of replicability used throughout this paper, an experiment was deemed replicable if its results can be replicated by researchers different from those of the paper. To that end, all replication studies have been conducted by two graduate students R2 and R3[9] of our department, both of them holding a Bachelor's degree in computer science. As the replication studies were only done for 8 papers, we report on our experience for each of these studies from a qualitative point of view.

2.2.2 RQ2: From where do researchers acquire Simulink models used as experimental subjects and what are their basic characteristics?

Next to the accessibility of models as part of RQ1, we are interested in where the researchers acquire Simulink models for the sake of experimentation and what are their basic characteristics such as size and active maintenance span. By collecting these insights, researchers in need of models for analysis or tool validation may emulate successful ways of getting models. In order to learn more about the context of a model or to get an updated version, it may be useful to contact the model creator, which motivates C2.1:

C2.1: Are all model creators mentioned in the paper?

By the term "creator" we do not necessarily mean an individual person. Instead, we consider model creation on a more abstract level, which means that a model creator could also be a company which is named in a paper or any other referenced research paper. If creators of all models were named, we answered C2.1 with "yes".

Next to the model creator, C2.2 dives more deeply into investigating a model's origin:

C2.2: From where are the models obtained?

C2.2 is one of our sub-research questions which cannot be answered by our usual "yes/no/unsure scheme". Possible answers were "researchers designed model themselves"⋆, "generator algorithm", "mutator algorithm", "industry partner"⋆, "open source", "other research paper"⋆, "Simulink -standard example"⋆, "multiple" and "unknown". The categories marked with a ⋆ were also used in [31]. As opposed to us, they also used the category "none", which we did not have to consider, due to our previous exclusion steps. The category "multiple" was used whenever two or more of these domains were used in one paper. Note that even if C2.1 was answered with "no", we may still be able to answer this question. For example, if the model was acquired from a company which is not named in the paper (e.g. due to a non-disclosure agreement), we may still be able to classify it as from an industry partner.

Finally, with C2.3 we give an outline about the basic characteristics of the models that were used in the experiments:

[9] R3 was only involved for this criterion.

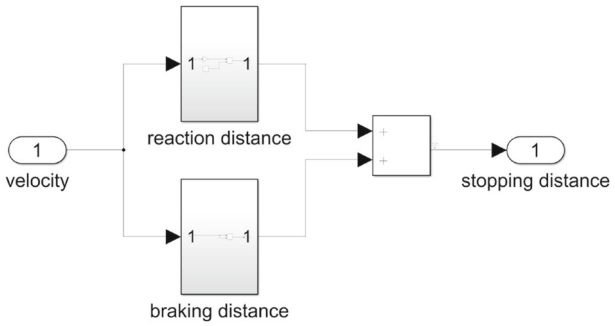

Fig. 3 A sample Simulink model showing Inport/Outport, Add and Subsystem blocks connected by signal lines. It computes the stopping distance from a car's velocity, by summing up reaction distance and braking distance. The details of the computation of reaction distance and braking distance are abstracted from in this view

C2.3: What are the basic characteristics of the experimental models?

With this, we are interested in what kinds of models researchers use: Are the models small toy examples created in a one-shot manner for the sake of illustration, or are they bigger and actively maintained over a certain period of time?

Simulink models are block diagrams used to model dynamical systems, where computing blocks are connected by signal lines (see Fig. 3 for a sample model). Blocks of various kinds (e.g., Sum, Logic, Switch, etc.) can apply transformations on their incoming signals, thereby producing modified outgoing signals. Inport and Outport blocks are specific blocks connecting a model with its surrounding context. Another special kind of block is the Subsystem block which can be used to modularize a Simulink model in a hierarchical manner. For more details, we kindly refer to Simulink introductions, e.g., [34]. We used basic MATLAB scripts to compute the following model metrics in order to characterize the models used as experimental subjects:

- Number of blocks,
- Amount of unique block types,
- Number of subsystems,
- Length of active maintenance span (creation date to last save date of a model).

We also counted, how many models were provided in the replication packages.

2.2.3 RQ3: Does the replicability of experimental results correlate with the impact of a paper?

With this RQ, we are interested whether a paper's impact may be influenced by its handling of replicability. We thus investigate whether papers that rank better in terms of replicability (using our results of C1.1, C1.3, C1.4) have a higher relative impact than the other papers. While our analysis cannot show an actual cause and effect relationship between replicability and impact, there are other studies [13,14] which let us hypothesize about a possible connection between the two.

We compute a paper's relative impact with the normalized citation score of Waltman et al. [35], and use a paper's citation count as the basis of this score. We collected the citation counts of each included paper from GoogleScholar[10] and ResearchGate.[11]

To strengthen our confidence in our computed value of the relative impact (see threats to validity), we compared it to the Scopus Field-Weighted Citation Impact,[12] whenever Scopus provides it for a paper. We did not use Scopus for our computation because they did not list impacts for 4 papers, and their opaque computation of the Field-Weighted Citation Impact. Finally, we group our included papers into 5 groups according to our classification of C1.1/C1.3/C1.4: no replication package, only software provided, only models provided, both provided and replicable. Comparisons then take place based on the average relative impact of the groups.

3 Results

In this section, we synthesize the results of our study. All paper references found, raw data extracted and calculations of results synthesized can be found in the replication package of this paper [36]. The package includes all the Simulink models we found during our study.

3.1 Are the experimental results of evaluating tools supporting model-based development with Simulink replicable? (RQ1)

First we summarize the results for the criteria C1.1 through C1.4, before we draw our conclusions for answering the overall research question RQ1. Table 1 provides details for C1.1 through C1.3, while Table 2 shows the results for C1.4.

It can be seen that R1 and R2 generally had a high inter-rater reliability and agreed that most papers did not make their models accessible. Almost all paper's evaluations needed a tool to be executed for its evaluation. Finally with we were unsure for the majority of the papers whether they provide access to their tools.

While answering C1.3, R1 and R2 first used the additional category "no", but this produced an unsatisfactory inter-rater reliability of only 0.42. To remedy this, we revised the

[10] https://scholar.google.com.

[11] https://www.researchgate.net.

[12] https://scopus.com.

Table 1 Detailed summary of replicability in principle

Criterion	Title	"Yes"		"No"		"Unsure"		Kappa
		R1	R2	R1	R2	R1	R2	
C1.1	Are all models accessible?	13	16	52	49	0	0	0.78
C1.2	Is a tool needed?	62	58	2	2	1	5	0.58
C1.3	Is the tool accessible?	19	17	0	0	39	41	0.68

Table 2 List of papers whose experiments we examined for replicability

No.	Title	Replicable?	Reference
1	A Synchronous Look at the Simulink Standard Library	no	[37]
2	Automatically Finding Bugs in a Commercial Cyber-Physical System Development Tool Chain With SLforge	partially	[38]
3	Contract-based verification of discrete-time multi-rate Simulink models	No	[39]
4	Evaluating Model Testing and Model Checking for Finding Requirements Violations in Simulink Models	No	[23]
5	Multi-Objective Black-Box Test Case Selection for Cost-Effectively Testing Simulation Models	Partially	[40]
6	Pareto efficient multi-objective black-box test case selection for simulation-based testing	Partially	[41]
7	SyLVaaS: System Level Formal Verification as a Service	Partially	[42]
8	Test Suite Prioritization for Efficient Regression Testing of Model-based Automotive Software	No	[43]

answers, merging the categories "no" and "unsure", acknowledging that R1 and R2 interpreted "no" and "unsure" too differently.

C1.4: Are the experiments replicable?

Table 2 summarizes those papers for which the models and tools were determined to be publicly available by R1 or R2, and for which R2 and R3 attempted to fully replicate the experimental results. The replication studies have been conducted during October and November 2020, with two different hardware/software setups. We used a main setup running Windows 10 on a 64GB RAM, AMD Ryzen 7 3800x CPU desktop PC and MatlabR2020b. In case of a replication failure, we retried on a server running SuSe Leap 15 on a 1TB RAM, 4 Intel Xeon E7-4880 CPUs and MATLAB R2019a. When parts of the replication package were inaccessible,[13] we contacted the authors to provide us with access to models and tools for replication purposes. We will use the numbers listed in the first column of Table 2 to refer to each of the papers.

Paper No.1 provided a Docker container for all its files. Still, R2 and R3 were not able to replicate the experiments, because they lacked knowledge of the Zélus tool. More explicit instructions in the handling of it were needed. The authors of the paper acknowledge that showing an equivalence between the Zélus blocks and Simulink blocks is hard to prove.

Paper No.2 offers a detailed documentation on how to perform the experiments, however documentation was missing in how to edit the configuration file for experimental replication. Both R2 and R3 tried to map the configuration parameters to those given in the paper, but the replication still failed for RQ2. A more detailed description would be necessary, as well as instructions on how to acquire the CyFuzz data for comparison, which was not provided by the authors.

Paper No.3 could not be replicated because the link to the tool was not accessible at the time of conducting our replication studies.

Paper No.4 was not replicable. At the time of conducting our replication studies, we were unable to gain access to the necessary QVtrace package. Similarly no access to the author's Google Drive was granted to us.

[13] This means our prior designation of "replicable in principle" by R1 or R2 was too favorable, or models or tools were no longer accessible for R2 and R3.

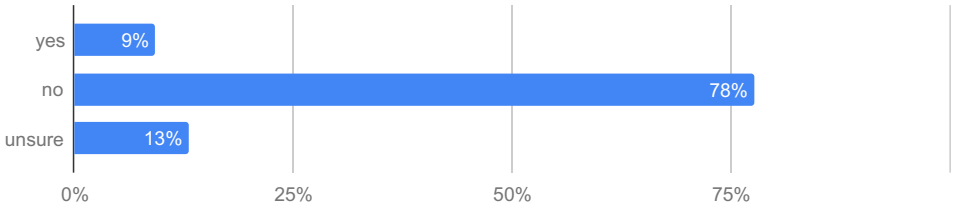

Fig. 4 Are studies of model-based development replicable in principle?

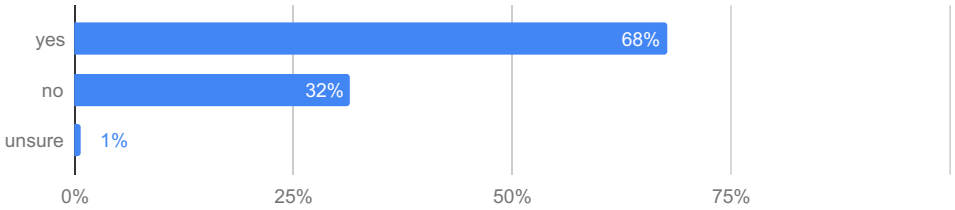

Fig. 5 C2.1: Are all model creators mentioned in the paper?

Paper No.5 offers all software components and models including a minimal documentation. While trying to replicate the experiments, we received numerous warnings and errors and were able to generate simulation times for only three out of four models. It was unclear how to deal with the simulation times, since the paper and documentation do not offer explicit instructions on how to aggregate the results such that they are comparable to those offered by the authors.

Paper No.6 is a revised version of paper No.5. It adds two more models to the experimental evaluation. As opposed to the first version of the paper, the ReadMe provided claims that scripts to execute the experiments are now being provided. However, we could only find one of the scripts, which was not executable. Thus, we were able to get simulation times for four out of six models, which were again not comparable with the authors' data.

Paper No.7's online tool described in the paper was not accessible. The authors instead provided us with a similar offline Docker container. Even though the documentation was very detailed, the important parameter τ was missing. Furthermore, models mentioned in the paper were not accessible to us anymore. Nevertheless, we were able to generate models with the tool, concluding that these experiments were the closest to replication.

Paper No.8 was not replicable, as implementation files were missing. As their model "Gearbox" and the Reactis tool are publicly available, only the very first step of their experiments could be replicated.

Aggregation and summary of the results To answer RQ1, we first assess whether the experiments described in the studied papers are replicable in principle. Therefore, we combine C1.1 and the revised answers of C1.3. For those papers where there was no tool needed (C1.2), C1.3 was classified as "yes". The formula we used is "If C1.1 = 'no' then 'no' else C1.3". This way, on average, 6 (R1: 5, R2: 7) papers have been classified as replicable in principle, 50.5 (R1: 52, R2: 49) as not replicable, and 8.5 (R1: 8, R2: 9) for which we were unsure (see Fig. 4), with Cohen's kappa of 0.67. In sum, 8 papers were classified to be replicable in principle by at least one of the researchers.

We were not able to fully replicate any of the experiments of those 8 papers, and achieved partial replicability for four of them. Three tools were not completely available to us due to denied access, timeouts and a missing implementation. One paper could not be replicated due to incomplete documentation. Four software setups were closely examined and principally functional, but the experiments could not be fully replicated due to incomplete documentation, errors or broken links to the models used as experimental subjects.

RQ 1: Are the experimental results of evaluating tools supporting model-based development with Simulink replicable?
We found 9% of the examined papers to be replicable in principle as software and models were provided. However, none of the experiments could be fully replicated, while we achieved partial replication for 6%. For 78% of the examined papers, either the tool or the models used as experimental subjects were not accessible, and we were not able to determine the principle replicability of the experiments presented in 13% of the investigated papers as we were unsure about the accessibility of tools.

3.2 From where do researchers acquire Simulink models used as experimental subjects and what are their basic characteristics? (RQ2)

C2.1 Are all model creators mentioned in the paper? As can be seen in Fig. 5, of the 65 papers investigated in detail, on average, 44 (R1: 43, R2: 45) papers mention the creators of all models. On the contrary, no such information could be

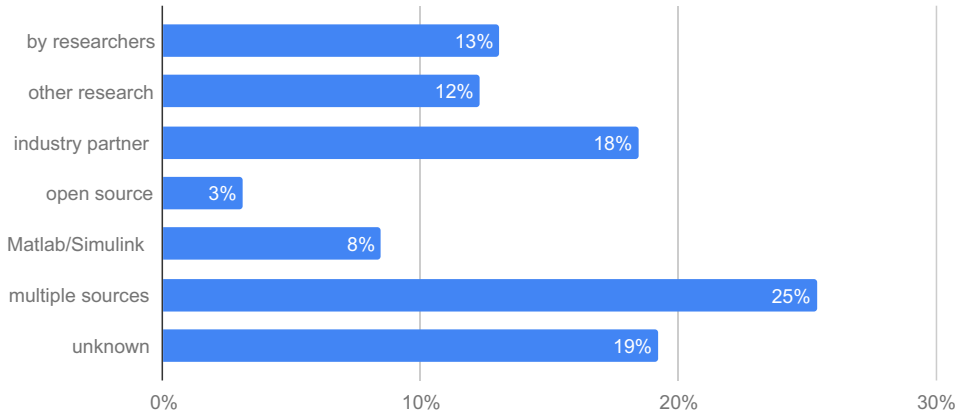

Fig. 6 C2.2: From where are the models obtained?

Table 3 Basic characteristics of the models used as experimental subjects in the papers which were found by our systematic literature search

Category	Min	Max	Mean	Median	Std.-dev.
#Models per paper	1.0	215.0	51.5	3.0	81.4
#Blocks per model	2.0	3276.0	305.5	236.0	364.6
#Subsystems per model	1.0	135.0	26.1	26.0	18.7
#Unique blocks per model	2.0	55.0	16.1	12.0	7.0
Maintenance span in days	0.0	7678.0	554.8	0.0	1076.5

found for an average of 20.5 (R1 22, R2 19) papers. Finally, there was one paper for which R2 was not sure, leading to an average value of 0.5 (R1: 0, R2: 1) for this category. In sum, this question was answered with an inter-rater reliability of 0.79.

C2.2 From where are the models obtained? As shown in Fig. 6, there is some variety for the model's origins. Only 3% used open source models, 8% used models included in Simulink or one of its toolboxes, 12% cited other papers, 13% built their own models, and 18% obtained models from industry partners. A quarter of all papers used models coming from two or more different sources. For 19% of the papers, we could not figure out where the models come from. This mostly coincides with those papers where we answered "no" in C2.1. For some papers, we were able to classify C2.2, even though we answered C2.1 with "no". E.g. we classified the origin of a model of [44] as "industry partner" based on the statement "a real-world electrical engine control system of a hybrid car", even though no specific information about this partner was given. C2.2 was answered with Cohen's kappa of 0.68.

An interesting yet partially expected perspective arises from combining C2.2 and C1.1. None of the models obtained from an "industry partner" are accessible. Three papers which we classified as "multiple" in C2.2 did provide industrial models though: [23] provides models from a "major aerospace and defense company" (name not revealed due to a non-disclosure agreement), while [40] and [41] use an open-source model of electro-mechanical braking provided by Bosch in [45]. Finally, [46] and [47] use models for an advanced driver assistance system by Daimler [48], that can be inspected on the project website.[14]

C2.3 What are the basic characteristics of experimental models? Table 3 lists basic characteristics of the models used as experimental subjects in the papers investigated by our study. For all of the characteristics, the standard deviation was high, and the distributions of the measures are right-skewed (median smaller than mean). Most of the papers used only a few models for their experiments, as the median is at only three. We found the models themselves to be not only toy examples, as there were 26 subsystems and 236 blocks per model in the median. A further indication of this is that the median model uses 12 different block types. Fig. 7 gives an impression of the 25 most frequently used blocks. The three most frequently used kinds of blocks are the Inport, Outport and Subsystem blocks; these are used for the sake of modularizing a model into different parts. All this shows some degree of sophistication in the models. Certainly, a "degree of sophistication" is not objectively measurable, but talking to Simulink experts in private conversation, they reported typical models in industry often having 1000-10000 blocks, most of the models we found in this study are smaller. Finally, we determined the maintenance spans of the models, which has the highest degree of deviation. There are many models with a maintenance span of "zero" days, but we also observe models with a maintenance spans of multiple years.

[14] https://www.se-rwth.de/materials/cncviewscasestudy.

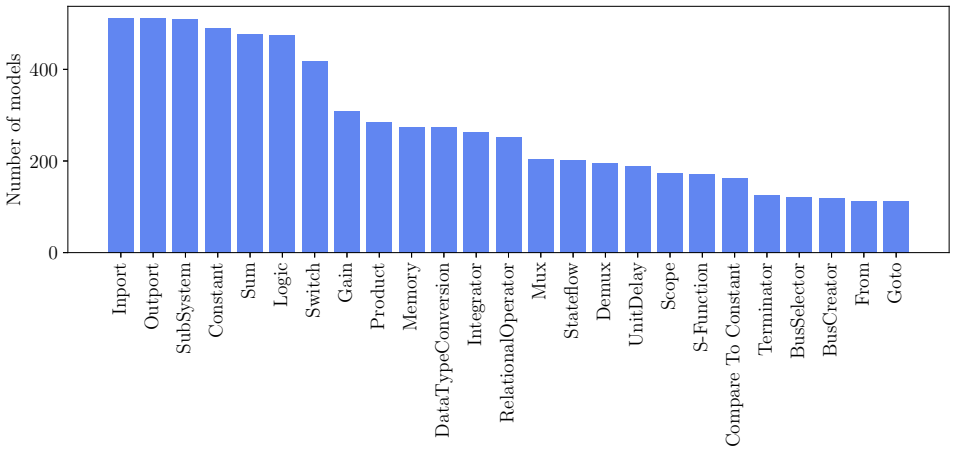

Fig. 7 Overview of the most frequently used kinds blocks in the Simulink models. The horizontal axis lists Simulink block types, and the vertical axis shows how many models did use this type of block

RQ 2: From where do researchers acquire Simulink models used as experimental subjects and what are their basic characteristics?
A wide variety of sources is used. 25% used multiple sources, another 25% used models of their own or from other researchers, 18% used models by an industry partner. 8% used models of Simulink or a toolbox and only 3% used open-source models. We could not determine 19% of the models' origins. A basic analysis of the models' characteristics showed them having a degree of sophistication containing 236 blocks, 26 subsystems, and 12 different block types in the median. On the contrary, the median maintenance span is at zero days, which suggests that most of the models have been created in a one-shot manner for the sake of illustration.

3.3 Does the replicability of experimental results correlate with the impact of a paper? (RQ3)

To compute the relative impact of a paper, we collected the citation counts on GoogleScholar and ResearchGate. GoogleScholar provided slightly higher counts of 8.6 average citations versus 5.8 on ResearchGate[15]. Because of the complete citation record and the generally higher values, we used the results of GoogleScholar for further computations of the relative impact. A histogram of the relative impact of the papers is shown in Fig. 8.

We compared our computed relative impacts with the Scopus Field-Weighted Citation Impact by computing the correlation between the two impacts with SPSS.[16] Spearman's rank correlation coefficient is 0.838 with significance $p < 0.001$, and Kendall's rank correlation coefficient is 0.668 with significance $p < 0.001$. We also applied

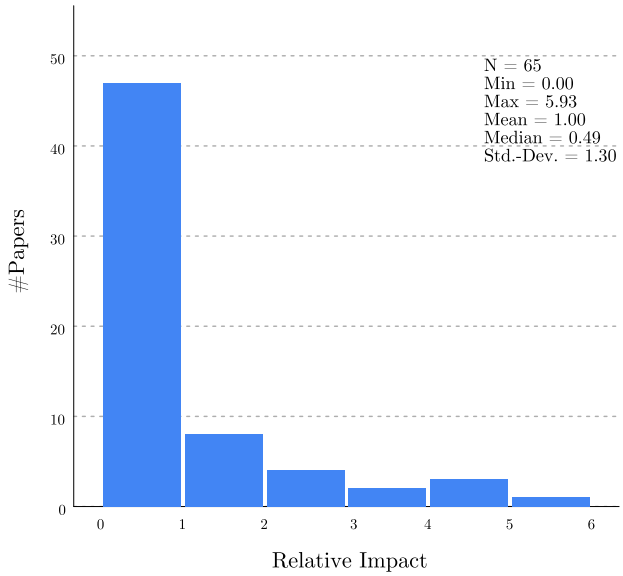

Fig. 8 Histogram of the relative impact of the papers. An average paper has a relative impact of 1.00, which means that it is cited as often as the average of its peers. One paper (the maximum) is cited 5.93 as often as its peers and some (the minima) were never cited and thus have a relative impact of 0.00

Wilcoxon's signed-rank test which indicates that the median of differences between our impact and the Scopus impact is the same, with significance $p = 0.074 > \alpha = 0.05$. Finally, the average value of our relative impact is 1.0 vs. 1.08 for the Scopus Field-Weighted Citation. With these similarities between the two measures, we concluded that our relative impact can be used for further analysis.

Next, we grouped the papers according to our findings of C1.1, C1.3 into the 4 groups: "nothing provided", "only models provided", "only software provided", "both provided". If at least one researcher found the models were provided, or at least one researcher found the software was provided or not necessary, the respective group is used. Note that if both researchers were "unsure", this was not the case. The aver-

[15] Three of the 65 papers could not be found on ResearchGate.

[16] https://www.ibm.com/analytics/spss-statistics-software.

Table 4 Average relative impacts grouped by our results of C1.1 and C1.3

Group	#Papers	Average relative impact
Nothing provided	27	0.590
Only models provided	9	0.739
Only software provided	21	1.369
Both provided	8	1.710

age relative impacts are presented in Table 4. The 4 papers achieving partial replicability (part of "both provided") even got an average impact of 2.072. The average relative impacts of the 4 groups are growing monotonically as listed in the table. The findings signal a positive correlation between a paper's replicative quality and its relative impact. Papers that only provided their software scored higher than papers that only provided their models. This does not imply a cause and effect relationship, though (cf. Sect. 4).

RQ 3: Does the replicability of experimental results correlate with the impact of a paper?
Grouped by our findings of C1.1 and C1.3, we found that groups of papers with higher replicability achieved a higher relative impact. Papers we could partially replicate also got the highest relative impacts.

4 Threats to validity

There are several reasons why the results of this study may not be representative, the major threats to validity as well as our countermeasures to mitigate these threats are discussed in the remainder of this section.

Our initial paper selection is based on selected databases and search strings. The initial query result may not include all the papers being relevant w.r.t. our scope. We tried to remedy this threat by using four different yet well-known digital libraries, which is sufficient according to [49,50] and using wild-carded and broad keywords in the search string.

For the first two inclusion/exclusion steps, we only considered titles and abstracts of the papers. If papers do not describe their topic and evaluation here, they could have been missed.

It turned out to be more difficult than originally expected to find out whether a paper provides a replication package or not. One reason for this is that just scanning the full text of a paper for occurrences of Simulink or the name of the tool is not very useful here since there are dozens of matches scattered all over the paper. In fact, almost every sentence could "hide" a valuable piece of information. We tried to remedy this problem by searching the *.pdf files for the key words mentioned in Sect. 2.2. We also merged our answers of "no" and "unsure" for C1.3 in reflection of this problem.

We were very strict in rating the accessibility, i.e., we expected not to have to contact a paper's authors for getting model or tool access. This may have lowered the number of papers we deemed to be replicable in principle.

More generally, the data collection process was done by two researchers, each of which may have overlooked important information or misinterpreted a concrete criterion. We tried to mitigate these issues through intermediate discussions. Furthermore, we calculated Cohen's kappa coefficient to better estimate the reliability of our extracted data and synthesized results.

Furthermore, the replication attempt of the experiments of 8 papers in C1.4 was only conducted by two researchers. The researchers did not have complete expert knowledge in the fields of the analyzed papers, which could have caused difficulties to reproduce experiments. This together with our strict grading of replicability may have led to a lower number of papers being replicable in principle (only 8 of 65) and papers we could fully replicate (none of the 8).

We did not investigate the paper's Simulink versions or hardware setups for our assessment of principal replicability. This is because in many cases newer versions of Simulink or our own hardware setups would produce equivalent results.

Our methodology section does not present a separate quality assessment which is typical in systematic literature review studies [51]. Thus, our results could be different if only a subset of high-quality papers, e.g., those published in the most prestigious publication outlets, would be considered. Nonetheless, a rudimentary quality assessment (paper's language, experimental evaluation instead of "blatant assertion" [5]) was done in our inclusion/exclusion process.

For the analysis of the Simulink models in C2.3, we used self-written Matlab scripts, which could be faulty. We especially cannot rule out that the maintenance span was computed correct for some models, as many models had a maintenance span of 0 days. This could naturally occur by a very short lived model or an automatic script creating and saving a model instantaneously. Another possibility is that the date feature of Simulink is buggy for some models.

As already indicated, there is a threat to the conclusion validity for answering RQ3 since there is not necessarily a cause and effect relationship between replicability of experimental results and the relative impact of a paper. Another possible cause could be the reputation or impact factor of the publication venue. If this is the case, however, then our results may point to higher replication standards for these venues. Moreover, our computed relative impact score is based on only 65 papers grouped by their publication year into 5 peer groups of size 9, 20, 15, 15 and 6. This is why we compared our relative impact with the Scopus metric for average, correlation and distribution and found it to be highly similar.

Another mitigating factor is that the papers were manually selected in our systemic literature review. This ensures comparing papers only with highly relevant peers in each peer group.

5 Discussion

Limited accessibility of both models and tools Although generally accepted, the FAIR guiding principles of good scientific practice are hardly adopted by current research on tools for model-based development with Simulink. From the 65 papers which have been selected for an in-depth investigation in our systematic literature review, we found that only 22% of the models and 31% of the tools required for replicating experimental results are accessible. Thus, future research that builds on published results, such as larger user or field studies, studies comparing their results with established results, etc., are hardly possible, which ultimately limits scientific progress in general.

Difficulties regarding replicability We found none of 8 thoroughly examined papers to be fully replicable. This was largely caused due to insufficient documentation on the experiment setups, parameters and tools, or missing parts of implementations, tools or models. We suggest providing Docker containers or similar for ease of experimental evaluation. These can come pre-installed and pre-configured. This way, the replicator simply has to download the container and start a script.

Replicability and relative impact We can confirm the finding of papers publishing their data being cited more often [13,14,52]. Our results further show, that publishing software or both data and software does have a higher correlation with citation counts, than just publishing the experimental datasets.

Open-source mindset rarely adopted One general problem is that the open-source mindset seems to be rarely adopted in the context of model-based development. Only 3% of the papers considered by our study obtained all of their models from open source projects. On the contrary, 18% of the studied papers obtain the models used as experimental subjects from industry partners, the accessibility of these models is severely limited by confidentiality agreements.

Selected remarks from other papers These quantitative findings are also confirmed by several authors of the papers we investigated during out study. We noticed a number of remarks w.r.t. the availability of Simulink models for evaluation purposes. Statements like "To the best of our knowledge, there are no open benchmarks based on real implementation models. We have been provided with two implementation models developed by industry. However, the details of these models cannot be open."[53]; "Crucial to our main study, we planned to work on real industrial data (this is an obstacle for most studies due to proprietary intellectual property concerns)."[46]; "[...] most public domain Stateflows are small examples created for the purpose of training and are not representative of the models developed in industry."[54]; or "Such benchmarking suites are currently unavailable [...] and do not adequately relate to real world models of interest in industrial applications."[24] discuss the problem of obtaining real-world yet freely accessible models from industry. Other statements such as "[...] as most of Simulink models [...] either lack open resources or contain a small-scale of blocks."[55] or "[...] no study of existing Simulink models is available [...]."[38,56] discuss the lack of accessible Simulink models in general.

Reflection of our own experience In addition, the findings reflect our own experience when developing several research prototypes supporting model management tasks, e.g., in terms of the SiDiff/SiLift project [57,58].

Likewise, we made similar observations in the SimuComp project. Companies want to save the intellectual property and do not want their (unobfuscated) models to be published.

As opposed to the lack of availability of models, we do not have any reasonable explanation for the limited accessibility of tools. Most of the tools presented in research papers are not meant to be integrated into productive development environments directly, but they merely serve as experimental research prototypes which should not be affected by confidentiality agreements or license restrictions.

Suggestions based on the lessons learnt from our study. While the aforementioned problems are largely a matter of fact and cannot be changed in a short-term perspective, we believe that researchers could do a much better job in sharing their experimental subjects. Interestingly, 12% of the studies obtain their experimental subjects from other papers, and 13% of the papers state that such models have been created by the authors themselves. Making these models accessible is largely a matter of providing adequate descriptions.

However, such descriptions are not always easy to find within the research papers which we considered in our study. Often, we could not find online resources for models or software. It should be made clear where to find replication packages. In some cases a link to the project's website was provided, but we couldn't find the models there. To prevent this, we suggest direct links downloadable files or very prominent links on the website. The web resource's language should match the paper's language: e.g., the project site of [59] is in German. Four papers referenced pages that did not exist anymore, e.g., a private Dropbox[17] account. These issues can be easily addressed by a more thorough archiving of a paper's replication package.

[17] https://www.dropbox.com.

We also suggest to name or cite creators of the models, so they can be contacted for a current version or context data of the model. In this respect, the results of our study are rather promising. After all, model creators have been mentioned in 68% of the studied papers, even if the models themselves were not accessible for a considerable amount of these cases.

Towards larger collections of Simulink models Our study not only reveals the severity of the problem, it may also be considered as a source for getting information about publicly available models and tools. In fact, we found a number of papers that did publish their models. This includes models with a degree of sophistication, e.g., models with more than a few hundred blocks. These models could be reused, updated or upgraded in other studies. We provide all digital artifacts that were produced in this work (.bibtex files of all papers found, exported spread sheets and retrieved models or paper references) online for download at [36]. Altogether we downloaded 517 Simulink models. We also found 32 referenced papers where models were drawn from. These models could be used by other researchers in their evaluation. Further initiatives of providing a corpus of publicly available models, including a recent collection of Simulink models, will be discussed in the next section.

6 Related work

The only related secondary study we are aware of has been conducted by Elberzhager et al. [31] in 2013. They conducted a systematic mapping study in which they analyzed papers about quality assurance of Simulink models. This is a sub-scope of our inclusion criteria, see Sect. 2.1. One research question of them was "How are the approaches evaluated?". They reviewed where the models in an evaluation come from and categorized them into "industry example", "Matlab example", "own example", "literature example" and "none". We include more categories, see Sect. 2.2, apart from "none". All papers that would fall in their "none"-category were excluded by us beforehand. Compared to their findings, we categorized 2 papers using open source models, one with a generator algorithm and 16 with multiple domains. Furthermore we found 11 papers, where the domain was not specified at all. They also commented on our RQ1: "In addition, examples from industry are sometimes only described, but not shown in detail for reasons of confidentiality."

The lack of publicly available Simulink models inspired the SLforge project to build the only large-scale collection of public Simulink models [38,60] known to us. To date, however, this corpus has been only used by the same researchers in order to evaluate different strategies for testing the Simulink tool environment itself (see, e.g., [61]). Another interesting approach was used by Sanchez et al. [62]. They used Google BigQuery[18] to find a sample of the largest available Simulink models on GitHub. In sum, they downloaded 70 Simulink models larger than 1MB from GitHub. Some authors created datasets of Simulink models as benchmarks. For example, Bourbouh et al. [63] compiled a set of 77 Stateflow models to demonstrate the effectiveness of their tool. Another benchmark of Simulink models was created as part of the Applied Verification for Continuous and Hybrid Systems (ARCH) workshop [64]. Regarding works describing characteristics of Simulink models, Dajsuren et al. [65,66] reported coupling and cohesion metrics they found in ten industrial Simulink models. They measured the inter-relation of subsystems as well as the inter-relation of blocks within subsystems.

In a different context focusing on UML models only, Hebig et al. [67] have systematically mined GitHub projects to answer the question when UML models, if used, are created and updated throughout the lifecycle of a project. A similar yet even more restricted study on the usage of UML models developed in Enterprise Architect has been conducted by Langer et al. [68].

Apart from individual studies, there is an increasing community effort towards the adoption of open science principles within the field of software and systems engineering. One of the goals of such efforts is to create the basis for replicating experimental results. Most notably, a number of ACM conferences and journals have established formal review processes in order to assess the quality of digital artifacts associated with a scientific paper, according to ACM's "Artifact Review and Badging" policy[19]. Artifacts may receive three different kinds of badges, referred to as "Artifacts Evaluated", "Artifacts Available", and "Results Validated". In terms of our study, we investigated the accessibility of digital artifacts in terms of C1.1 and C1.3, which is an essential prerequisite for receiving an "Artifacts Available" badge. The notion of replicability used in terms of this paper and particularly addressed in C1.4 is one of the two possible forms of how experimental results may receive a "Results Validated" badge. As a natural side-effect of conducting our replication studies, we also investigated the documentation, completeness, consistency and exercisability of artifacts, which are the requirements for receiving an "Artifacts Evaluated" badge.

7 Conclusion

In this paper, we investigated to which degree the principles of good scientific practice are adopted by current research on tools for model-based development, focusing on tools

[18] https://cloud.google.com/bigquery.

[19] https://www.acm.org/publications/policies/artifact-review-and-badging-current.

supporting Simulink. To that end, we conducted a systematic literature review and analyzed a set of 65 relevant papers on how they deal with the accessibility of experimental replication packages.

We found that only 31% of the tools and 22% of the models used as experimental subjects are accessible. Given that both artifacts are needed for a replication study, only 9% of the tool evaluations presented in the examined papers can be classified to be replicable in principle. We found none of those papers to be fully replicable and only 6% of them partially replicable. Moreover, only a minor fraction of the models is obtained from open-source projects, but some of those open source models show a degree of sophistication and could be useful for other experimental evaluations. Altogether, we see this as an alarming signal w.r.t. making scientific progress on better tool support for model-based development processes centered around Simulink. Giving access to the models and tools also could potentially result in a higher impact in the scientific community—this may serve as another motivation to give more care to replicability.

While both tool and models are essential prerequisites for replication and reproducibility studies, the latter may also serve as experimental subjects for evaluating other tools. In this regard, our study may serve as a source for getting information about publicly available models. Other researchers in this field have even started to curate and analyze a much larger corpus of Simulink models [60,69]. Open-source models were found to be highly diverse in almost all metrics applied with some being complex enough to be representative of industry models.

Funding Open access funding provided by University of Bern.

References

1. Brambilla M, Cabot J, Wimmer M (2017) Model-driven software engineering in practice. Synth Lect Softw Eng 3(1):1–207
2. Völter M, Stahl T, Bettin J, Haase A, Helsen S (2013) Model-driven software development: technology, engineering, management. John Wiley & Sons
3. Liggesmeyer P, Trapp M (2009) Trends in embedded software engineering. IEEE Softw 26(3):19–25
4. Robert France and Bernhard Rumpe, "Model-driven Development of Complex Software: A Research Roadmap," Future of Software Engineering (FOSE '07), 2007, pp. 37–4, https://doi.org/10.1109/FOSE.2007.14
5. Shaw M (2002) What makes good research in software engineering? Int J Softw Tools Technol Transf 4(1):1–7
6. Tichy WF (1998) Should computer scientists experiment more? Computer 31(5):32–40
7. Meyer B (2010) Empirical research: questions from software engineering. In: 4th international symposium on empirical software engineering and measurement (ESEM 2010)
8. Barba LA (2018) Terminologies for reproducible research. arXiv preprint arXiv:1802.03311
9. Juristo N., Gómez O.S. (2012) Replication of Software Engineering Experiments. In: Meyer B., Nordio M. (eds) Empirical Software Engineering and Verification. LASER 2010, LASER 2009, LASER 2008. Lecture Notes in Computer Science, vol 7007. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25231-0_2
10. Basili VR, Shull F, Lanubile F (1999) Building knowledge through families of experiments. IEEE Trans Softw Eng 25(4):456–473
11. Mark D. Wilkinson., Michel Dumontier, IJsbrand J. Aalbersberg, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3, 160018 (2016). https://doi.org/10.1038/sdata.2016.18
12. Lamprecht AL, Garcia L, Kuzak M, Martinez C, Arcila R, Martin Del Pico E, Dominguez Del Angel V, van de Sandt S, Ison J, Martinez PA, et al. (2019) Towards fair principles for research software. Data Sci pp. 1–23
13. Piwowar HA, Day RS, Fridsma DB (2007) Sharing detailed research data is associated with increased citation rate. PloS One 2(3):e308
14. Piwowar HA, Vision TJ (2013) Data reuse and the open data citation advantage. PeerJ 1:e175
15. Whittle J, Hutchinson J, Rouncefield M, Burden H, Heldal R (2013) Industrial adoption of model-driven engineering: Are the tools really the problem? In: International Conference on Model Driven Engineering Languages and Systems. pp. 1–17. Springer
16. Boll A, Kehrer T (2020) On the replicability of experimental tool evaluations in model-based development. In: International conference on systems modelling and management. Springer, pp. 111–130
17. Barbara Kitchenham, Stuart Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report, Version 2.3, Keele University and University of Durham, 9 July 2007
18. Rebaya A, Gasmi K, Hasnaoui S (2018) A Simulink-based rapid prototyping workflow for optimizing software/hardware programming. In: 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). pp. 1–6. IEEE
19. Kuroki Y, Yoo M, Yokoyama T (2016) A Simulink to UML model transformation tool for embedded control software development. In: IEEE international conference on industrial technology, ICIT 2016, Taipei, Taiwan. IEEE, pp. 700–706. https://doi.org/10.1109/ICIT.2016.7474835
20. Stephan M, Cordy JR (2015) Identifying instances of model design patterns and antipatterns using model clone detection. In: Proceedings of the seventh international workshop on modeling in software engineering. MiSE '15, IEEE Press, pp. 48–53
21. Matinnejad R, Nejati S, Briand LC, Bruckmann T (May 2016) Automated test suite generation for time-continuous Simulink models. In: 2016 IEEE/ACM 38th international conference on software engineering (ICSE), pp. 595–606. https://doi.org/10.1145/2884781.2884797
22. Matinnejad R, Nejati S, Briand LC, Bruckmann T (2019) Test generation and test prioritization for Simulink models with dynamic behavior. IEEE Trans Softw Eng 45(9):919–944. https://doi.org/10.1109/TSE.2018.2811489
23. Nejati S, Gaaloul K, Menghi C, Briand LC, Foster S, Wolfe D (2019) Evaluating model testing and model checking for finding requirements violations in Simulink models. In: Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software

engineering. ESEC/FSE 2019, Association for computing machinery, New York, NY, USA, pp. 1015–1025. https://doi.org/10.1145/3338906.3340444
24. Rao AC, Raouf A, Dhadyalla G, Pasupuleti V (2017) Mutation testing based evaluation of formal verification tools. In: 2017 international conference on dependable systems and their applications (DSA), pp. 1–7. https://doi.org/10.1109/DSA.2017.10
25. Gerlitz T, Kowalewski S (2016) Flow sensitive slicing for MATLAB/Simulink models. In: 2016 13th working IEEE/IFIP conference on software architecture (WICSA), pp. 81–90. https://doi.org/10.1109/WICSA.2016.23
26. Khelifi A, Ben Lakhal NM, Gharsallaoui H, Nasri O (2018) Artificial neural network-based fault detection. In: 2018 5th international conference on control, decision and information technologies (CoDIT), pp. 1017–1022. https://doi.org/10.1109/CoDIT.2018.8394963
27. Oussalem O, Kourchi M, Rachdy A, Ajaamoum M, Idadoub H, Jenkal S (2019) A low cost controller of PV system based on Arduino board and INC algorithm. Mater Today Proc. https://doi.org/10.1016/j.matpr.2019.07.689
28. Norouzi P, Kıvanç ÖC, Üstün Ö (2017) High performance position control of double sided air core linear brushless DC motor. In: 2017 10th international conference on electrical and electronics engineering (ELECO), pp. 233–238
29. Gallego-Calderon J, Natarajan A (2015) Assessment of wind turbine drive-train fatigue loads under torsional excitation. Eng Struct 103:189–202. https://doi.org/10.1016/j.engstruct.2015.09.008
30. Rashid M, Anwar MW, Khan AM (2015) Toward the tools selection in model based system engineering for embedded systems—a systematic literature review. J Syst Softw 106:150–163
31. Elberzhager F, Rosbach A, Bauer T (2013) Analysis and testing of Matlab simulink models: a systematic mapping study. In: Proceedings of the 2013 international workshop on joining AcadeMiA and industry contributions to testing automation. JAMAICA 2013, association for computing machinery, New York, NY, USA, pp. 29–34. https://doi.org/10.1145/2489280.2489285
32. Gusenbauer M, Haddaway NR (2019) Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed and 26 other resources. Res Synth Methods 11:181–217
33. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46
34. Yakimenko OA (2019) Engineering computations and modeling in MATLAB®/Simulink®. American Institute of Aeronautics and Astronautics, Inc 12700 Sunrise Valley Drive, Suite 200 Reston, VA 20191-5807 ISBN (print): 978-1-62410-515-9 https://doi.org/10.2514/4.105159
35. Waltman L, van Eck NJ (2013) A systematic empirical comparison of different approaches for normalizing citation impact indicators. J Informetr 7(4):833–849
36. Boll A, Vieregg N, Kehrer T, The download link of digital artifacts of this paper, for reuse and replication. https://doi.org/10.6084/m9.figshare.13633928
37. Bourke T, Carcenac F, Colaço JL, Pagano B, Pasteur C, Pouzet M (2017) A synchronous look at the simulink standard library. ACM Trans Embed Comput Syst. https://doi.org/10.1145/3126516
38. Chowdhury SA, Mohian S, Mehra S, Gawsane S, Johnson TT, Csallner C (2018) Automatically finding bugs in a commercial cyber-physical system development tool chain with SLforge. In: 2018 IEEE/ACM 40th international conference on software engineering (ICSE), pp. 981–992. https://doi.org/10.1145/3180155.3180231
39. Boström P, Wiik J (2015) Contract-based verification of discrete-time multi-rate simulink models. Softw Syst Model. https://doi.org/10.1007/s10270-015-0477-x
40. Arrieta A, Wang S, Arruabarrena A, Markiegi U, Sagardui G, Etxeberria L (2018) Multi-objective black-box test case selection for cost-effectively testing simulation models. In: Proceedings of the genetic and evolutionary computation conference. GECCO '18, association for computing machinery, New York, NY, USA, p. 1411 – 1418. https://doi.org/10.1145/3205455.3205490
41. Arrieta A, Wang S, Markiegi U, Arruabarrena A, Etxeberria L, Sagardui G (2019) Pareto efficient multi-objective black-box test case selection for simulation-based testing. Inf Softw Technol 114:137–154. https://doi.org/10.1016/j.infsof.2019.06.009
42. Mancini T, Mari F, Massini A, Melatti I, Tronci E (2015) Sylvaas: system level formal verification as a service. In: 2015 23rd euromicro international conference on parallel, distributed, and network-based processing, pp. 476–483. https://doi.org/10.1109/PDP.2015.119
43. Morozov A, Ding K, Chen T, Janschek K (2017) Test suite prioritization for efficient regression testing of model-based automotive software. In: 2017 international conference on software analysis, testing and evolution (SATE), pp. 20–29. https://doi.org/10.1109/SATE.2017.11
44. Holling D, Hofbauer A, Pretschner A, Gemmar M (2016) Profiting from unit tests for integration testing. In: 2016 IEEE international conference on software testing, verification and validation (ICST), pp. 353–363. https://doi.org/10.1109/ICST.2016.28
45. Strathmann T, Oehlerking J (2015) Verifying properties of an electro-mechanical braking system. In: 2nd workshop on applied verification of continuous and hybrid systems (ARCH 2015)
46. Bertram V, Maoz S, Ringert JO, Rumpe B, von Wenckstern M (2017) Component and connector views in practice: An experience report. In: Proceedings of the ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems. p. 167–177. MODELS '17, IEEE Press. https://doi.org/10.1109/MODELS.2017.29
47. Kusmenko E, Shumeiko I, Rumpe B, von Wenckstern M (2018) Fast simulation preorder algorithm. In: Proceedings of the 6th international conference on model-driven engineering and software development. MODELSWARD 2018, SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT. https://doi.org/10.5220/0006722102560267
48. Bertram V, Maoz S, Ringert JO, Rumpe B, von Wenckstern M (2017) Component and connector views in practice: an experience report. In: 2017 ACM/IEEE 20th International conference on model driven engineering languages and systems (MODELS), IEEE, pp. 167–177
49. Kitchenham B, Pretorius R, Budgen D, Brereton OP, Turner M, Niazi M, Linkman S (2010) Systematic literature reviews in software engineering-a tertiary study. Inf Softw Technol 52(8):792–805
50. Frâncila Weidt and Rodrigo Silva, Systematic literature review in computer science-a practical guide, Technical Report, Federal University of Juiz de Fora, November 2016, https://doi.org/10.13140/RG.2.2.35453.87524
51. Stapić Z, López EG, Cabot AG, de Marcos Ortega L, Strahonja V (2012) Performing systematic literature review in software engineering. In: CECIIS 2012-23rd international conference
52. Masuzzo P, Martens L (2017) Do you speak open science? resources and tips to learn the language. Tech. rep, PeerJ Preprints
53. Tomita T, Ishii D, Murakami T, Takeuchi S, Aoki T (2019) A scalable Monte-Carlo test-case generation tool for large and complex simulink models. In: 2019 IEEE/ACM 11th International Workshop on Modelling in Software Engineering (MiSE). pp. 39–46. https://doi.org/10.1109/MiSE.2019.00014
54. Hussain A, Sher HA, Murtaza AF, Al-Haddad K (2019) Improved restricted control set model predictive control (iRCS-MPC) based maximum power point tracking of photovoltaic mod-

ule. IEEE Access 7:149422–149432. https://doi.org/10.1109/ACCESS.2019.2946747
55. Jiang Z, Wu X, Dong Z, Mu M Optimal test case generation for Simulink models using slicing. In: 2017 IEEE international conference on software quality, reliability and security companion (QRS-C), pp. 363–369. https://doi.org/10.1109/QRS-C.2017.67
56. Chowdhury SA (2018) Understanding and improving cyber-physical system models and development tools. In: 2018 IEEE/ACM 40th international conference on software engineering: companion (ICSE-Companion), pp. 452–453
57. Kehrer T, Kelter U, Pietsch P, Schmidt M (2012) Adaptability of model comparison tools. In: Proceedings of the 27th IEEE/ACM international conference on automated software engineering. IEEE, pp. 306–309
58. Kehrer T, Kelter U, Ohrndorf M, Sollbach T (2012) Understanding model evolution through semantically lifting model differences with SiLift. In: 28th IEEE international conference on software maintenance (ICSM). IEEE, pp. 638–641
59. Wille D, Babur Ö, Cleophas L, Seidl C, van den Brand M, Schaefer I (2018) Improving custom-tailored variability mining using outlier and cluster detection. Sci Comput Program 163:62–84. https://doi.org/10.1016/j.scico.2018.04.002
60. Chowdhury SA, Varghese LS, Mohian S, Johnson TT, Csallner C (2018) A curated corpus of Simulink models for model-based empirical studies. In: 2018 IEEE/ACM 4th international workshop on software engineering for smart cyber-physical systems (SEsCPS). IEEE, pp. 45–48
61. Chowdhury SA, Shrestha SL, Johnson TT, Csallner C (2020) SLEMI: Equivalence modulo input (EMI) based mutation of CPS models for finding compiler bugs in Simulink. In: Proceeding 42nd ACM/IEEE international conference on software engineering (ICSE), ACM. To appear
62. Sanchez B, Zolotas A, Rodriguez HH, Kolovos D, Paige R (2019) On-the-fly translation and execution of OCL-like queries on simulink models. In: 2019 ACM/IEEE 22nd international conference on model driven engineering languages and systems (MODELS). IEEE, pp. 205–215
63. Bourbouh H, Garoche PL, Garion C, Gurfinkel A, Kahsai T, Thirioux X (2017) Automated analysis of stateflow models. In: 21st International conference on logic for programming, artificial intelligence and reasoning (LPAR 2017), pp. 144–161
64. Ernst G, Arcaini P, Donze A, Fainekos G, Mathesen L, Pedrielli G, Yaghoubi S, Yamagata Y, Zhang Z (2019) Arch-comp 2019 category report: falsification. In: ARCH@ CPSIoTWeek, pp. 129–140
65. Dajsuren Y, van den Brand MG, Serebrenik A, Roubtsov S (2013) Simulink models are also software: modularity assessment. In: 9th international ACM sigsoft conference on quality of software architectures (QoSA), pp. 99–106
66. Dajsuren Y (2015) On the design of an architecture framework and quality evaluation for automotive software systems. Ph.D. thesis, Department of mathematics and computer science, Technische Universiteit Eindhoven
67. Hebig R, Quang TH, Chaudron MR, Robles G, Fernandez MA (2016) The quest for open source projects that use UML: mining GitHub. In: Proceedings of the ACM/IEEE 19th international conference on model driven engineering languages and systems, pp. 173–183
68. Philip Langer, Tanja Mayerhofer, Manuel Wimmer, Gerti Kappel, On the Usage of UML: Initial Results of Analyzing Open UML Models, Modellierung 2014, Editors: Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer, Gesellschaft für Informatik e.V., Bonn, 2014 ISBN 978-388579-619-0 http://dl.gi.de/handle/20.500.12116/20950
69. Boll A, Brokhausen F, Amorim T, Kehrer T, Vogelsang A, Characteristics, potentials, and limitations of open-source simulink projects for empirical research. Softw Syst Model pp. 1–20 (2021)