On the Replicability of Experimental Tool Evaluations in Model-based Development*

Lessons Learnt from a Systematic Literature Review Focusing on MATLAB/Simulink

Alexander Boll and Timo Kehrer

Department of Computer Science
Humboldt-Universität zu Berlin
Berlin, Germany
{boll,kehrer}@informatik.hu-berlin.de

Abstract. Research on novel tools for model-based development differs from a mere engineering task by providing some form of evidence that a tool 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. We investigate to which degree these basic prerequisites for the replicability of experimental results are met by recent research reporting on novel methods, techniques, or algorithms supporting model-based development using MATLAB/Simulink. Our results from a systematic literature review 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. 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 can only be achieved 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 · Tools · MATLAB/Simulink· Experimental evaluation · FAIR principles · Replicability.

1 Introduction

Model-based development [6,46] 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 [30]. It promotes the use of models in all stages of

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

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 [13]. 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 an experimental evaluation [40]. In turn, to ensure scientific progress in general, experimental results should be transparent and replicable. 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 [48,28].

In this paper, we investigate to which degree these principles of good scientific practice are actually adopted by current research on tools for model-based development of embedded systems. We focus on tools for MATLAB/Simulink, 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 MATLAB/Simulink replicable in principle?

RQ2: From where do researchers acquire MATLAB/Simulink models for the sake of experimentation?

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 MATLAB/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 Section 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 (RQ1). The models come from a variety of sources, e.g., from other research papers, industry partners of the paper's authors, open source projects, or examples provided along with MATLAB/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. Our findings are confirmed by other research papers which we investigated during our study. Our detailed results are presented in detail in Section 3.

While we do not claim our results to represent a complete image of how researchers adopt the FAIR principles of good scientific practice in our field of interest (see Section 4 for a discussion of major threats to validity), we see them 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 [47], 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 Section 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 Section 6, and Section 7 concludes the paper.

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 [24]. Details of our search process and research paper selection are described in Section 2.1. Section 2.2 is dedicated to our data extraction policy, structured along a refinement of our overall research questions RQ1 and RQ2, respectively.

2.1 Search Process and Research Paper Selection

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 MATLAB/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 [38] or model transformation [26], clone detection [43], test generation [31] and priorization [32], model checking [33] and validation [36], model slicing [15] and fault detection [23]. On the contrary, we ignore model-based solutions using MATLAB/Simulink for solving a specific problem in a particular domain, such as solar panel array positioning [35], motor control [34], or wind turbine design [14].

Databases and Search Strings As illustrated in Fig. 2, we used the digital libraries of *ACM*,[1] *IEEE*,[2] *ScienceDirect*,[3] and *dblp*[4] 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

[1] https://dl.acm.org
[2] https://ieeexplore.ieee.org/Xplore/home.jsp
[3] https://www.sciencedirect.com
[4] https://dblp.uni-trier.de

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))

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

model-based development like [37] or [12]. By using these four different digital libraries, we are confident to capture a good snapshot of relevant papers.

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 ($\star$). Since the wildcard symbol is not supported by the query engine of ScienceDirect[16], we slightly adapted these search strings for querying ScienceDirect. The same applies to dblp [16], 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 1st, 2015 and February 24th, 2020. With these settings, we found 625 papers on IEEE, 88 on ACM, 214 on ScienceDirect[5] and 15 on dblp.

Using the bibliography reference manager JabRef,[6] 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.

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

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

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

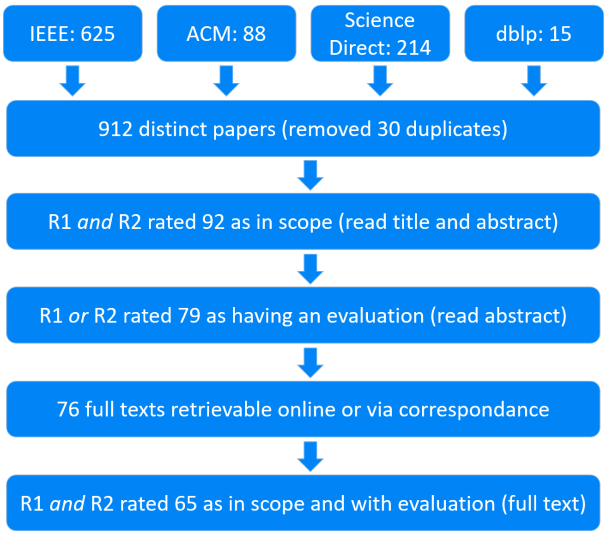


Fig. 2: Overview of the search process and research paper selection. Numbers of included research papers are shown for each step. After the initial query results obtained from the digital libraries of *ACM*, *IEEE*, *ScienceDirect* and *dblp*, the study has been performed by two researchers, referred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

coefficient [11][24], 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 and 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[7] 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 MATLAB/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

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

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 sub-questions which are supposed to be answered in a straightforward manner, typically by a classification into "yes", "no", or "unsure". After the first 20 papers have been investigated by researchers R1 and R2, they compared their different answers in order to clarify potential misunderstandings in the phrasing of the questions and/or the interpretation of the papers' contents.

RQ1: Are the experimental results of evaluating tools supporting model-based development with MATLAB/Simulink replicable in principle?

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 MATLAB/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 MATLAB/Simulink, sub-systems which are not shown in the paper, etc.

The aim of RQ1.1 is to assess if a paper comprises a hint on the accessibility of the models used for the sake of evaluation:

RQ1.1: Does the paper describe whether the models are accessible?

To answer this question for a given paper, we read the evaluation section of the paper, 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.". Please note that, for this question, we are only looking for a general statement on the accessibility of models. That is, if a paper states that the models used for evaluation cannot be provided due to, e.g., confidentiality agreements, we nonetheless answer the question with "yes".

On the contrary, a positive answer to RQ1.2 not only requires some statement about accessibility, but it requires that the models indeed *are* accessible:

RQ1.2: Are all models accessible?

The accessibility of models can only be checked if the paper includes a general statements on this. Thus, we did not inspect those papers for which RQ1.1 has been answered by "no" or "unsure". For all other papers, a positive answer to RQ1.2 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 MATLAB/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 MATLAB/Simulink. In this case, the concrete model name and version of the files or suite must also be mentioned.
- 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 RQ1.3:

RQ1.3: 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 RQ1.3 is answered by "yes", we continue with RQ1.4 and RQ1.5.

Similar to our investigation of the accessibility of models, we first assess if a paper comprises a hint on the accessibility of the presented tool:

RQ1.4: Does the paper describe whether the tool is 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.

The actual check for accessibility is addressed by RQ1.5:

RQ1.5: Is the tool accessible?

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. On the contrary, we considered MATLAB/Simulink or any of its toolboxes as pre-installed and thus accessible by default.

RQ2: From where do researchers acquire MATLAB/Simulink models for the sake of experimentation?

Next to the accessibility of models, we are interested in where the researchers acquire MATLAB/Simulink models for the sake of experimentation. 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 RQ2.1:

RQ2.1: Are all model creators mentioned in the paper?

By the term "creator" we 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 RQ2.1 with "yes".

Next to the model creator, RQ2.2 dives more deeply into investigating a model's origin:

RQ2.2: From where are the models obtained?

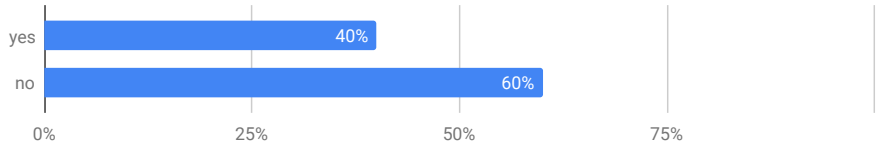
RQ2.2 is the only research question which cannot be answered by our usual "yes/no/unsure scheme". Possible answers were "researchers designed model themselves"*, "generator algorithm", "mutator algorithm", "industry partner"*, "open source", "other research paper"*, "MATLAB/Simulink-standard example"*, "multiple" and "unknown". The categories marked with a * were also used in [12]. 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 RQ2.1 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.

3 Results

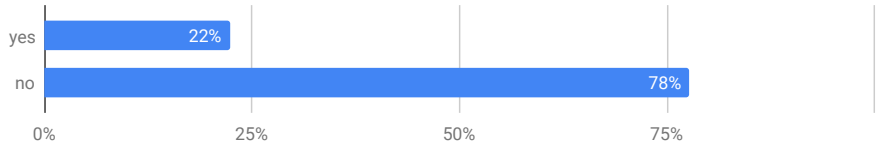
In this section, we synthesize the results of our study. All paper references found, raw data extracted and calculations of results synthesized can be found in the replication package of this paper [5]. The package includes all the MATLAB/Simulink models we found during our study.

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

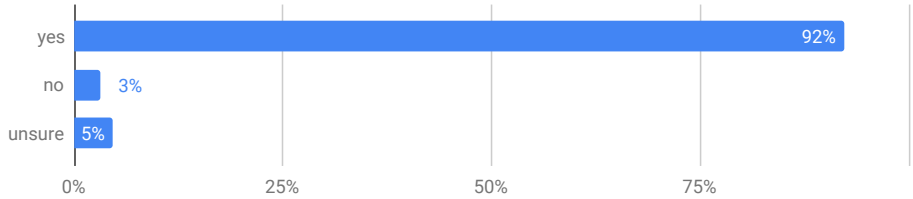
In this section, we first summarize the results for the research questions RQ1.1 through RQ1.5 (see Fig. 3 for an overview), before we draw our conclusions for answering the overall research question RQ1.



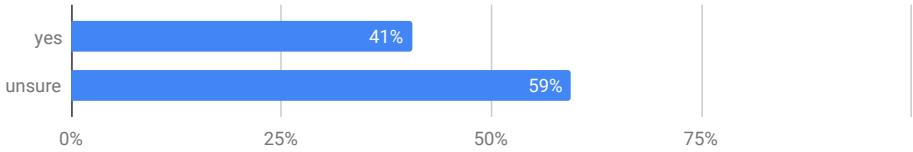
(a) RQ1.1: Does the paper describe whether the models are accessible?



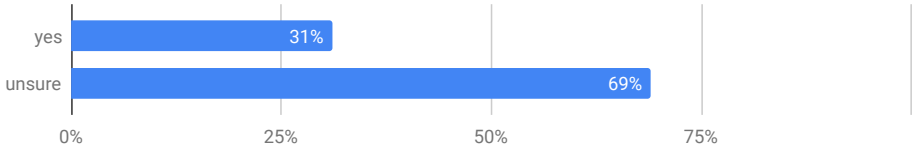
(b) RQ1.2: Are all models accessible?



(c) RQ1.3: Is the tool needed for the evaluation?



(d) RQ1.4: Does the paper describe whether the tool is accessible?



(e) RQ1.5: Is the tool accessible?

Fig. 3: Overview of the results for research questions RQ1.1 through RQ1.5. Percentage values are representing the average of the answers of researchers R1 and R2.

RQ1.1: *Does the paper describe whether the models are accessible?* On average, 26 (R1: 25, R2: 27) of the 65 papers for which we did a detailed analysis include a hint on whether the models used as experimental subjects are accessible (see Fig. 3a). For this question, researchers R1 and R2 achieve an inter-rater reliability of 0.62, measured in terms of Cohen's kappa coefficient.

RQ1.2: *Are all models accessible?* As for the actual accessibility, on average, 14.5 (R1: 13, R2: 16) of the papers including descriptions on the availability of models indeed gave access to all models (see Fig. 3b). The results have been achieved with an inter-rater reliability of 0.78. One paper includes a broken link [50], and we were not able to find the claimed web resources of 4 papers.

RQ1.3: *Is the tool needed for the evaluation?* Of all the 65 papers, on average, 60 (R1: 62, R2: 58) require their developed tool to be executed for the sake of evaluation. While the tool evaluations for 2 of the papers clearly do not rely on an actual execution of the tool (R1: 2, R2: 2), we were unsure in some of the cases (R1: 1, R2: 5), see Fig. 3c. Cohen's kappa is at 0.58 for this question. As for RQ1.4 and RQ1.5, we analyze the 58 papers for which both R1 and R2 gave a positive answer to this question.

RQ1.4: *Does the paper describe whether the tool is accessible?* We initially answered this question with average values of 23.5 (R1: 26, R2: 21), 11.5 (R1: 19, R2: 4) and 23 (R1: 13, R2: 33) regarding the possible answers of "yes", "no" and "unsure", respectively. This reflects a rather poor inter-rater reliablity of 0.38. However, our observation was that it is almost impossible to be sure that there is *no* description of a tool's accessibility, since it could be "hidden" in multiple places. We thus revised our answers and merged the "no" and "unsure" categories to become "unsure". With the merged categories, 34.5 (R1: 32, R2: 37) of the papers were listed as "unsure" (see Fig. 3d), and a kappa of 0.61.

RQ1.5: *Is the tool accessible?* Similarly to RQ1.4, we initially ended up in a poor inter-rater reliability of 0.42 for classifying the accessibility of 18 (R1: 19, R2: 17), 17 (R1 26, R2 8) and 23 (R1 13, R2 33) as "yes", "no" and "unsure", respectively. Again, we revised the answers, merging the categories "no" and "unsure", as in RQ1.4. Finally, we got average values of 18 (R1: 19, R2: 17) "yes" and 40 (R1: 39, R2: 41) "unsure" (see Fig. 3e), with a kappa of 0.68.

Aggregation of the results. To answer RQ1, we combine RQ1.2 and the revised answers of RQ1.5. For those papers where there was no tool needed (RQ1.3), RQ1.5 was classified as "yes". The formula we used is "If RQ1.2 = 'no' then 'no' else RQ1.5". This way, on average, 6 (R1: 5, R2: 7) papers have been classified as replicable, 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 by at least one of the researchers.

However, we have to stress here that being replicable *in principle* does not imply that the results of the paper are in fact replicable. In fact, the accessibility

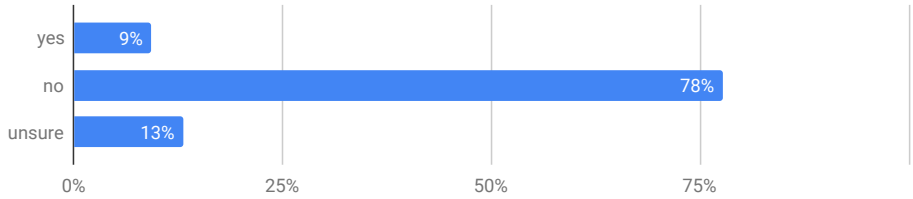


Fig. 4: RQ 1: Are studies of model-based development replicable in principle?

of models and tools used in the evaluation is only a necessary but not a sufficient condition for replicability. We did not try to install or run the tools according to the experimental setups described in the papers.

RQ 1: Are studies of model-based development replicable in principle?
We found 9% of the examined papers to be replicable in principle. For 78%, either the tool or the models used as experimental subjects were not accessible, and were not able to determine the principle replicability of 13% as we were unsure about the accessibility of tools.

3.2 From where do researchers acquire MATLAB/Simulink models for the sake of experimentation? (RQ2)

RQ 2.1 *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 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.

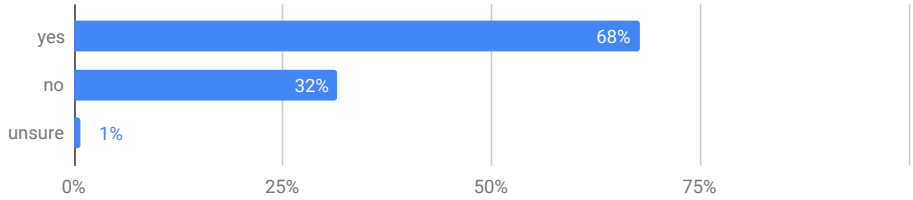


Fig. 5: RQ 2.1: Are all model creators mentioned in the paper?

RQ 2.2 *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 MATLAB/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 RQ2.1. For some papers, we were able to classify RQ2.2, even though we answered RQ2.1 with "no". E.g. we classified the origin of a model of [18] 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. RQ2.2 was answered with Cohen's kappa of 0.68.

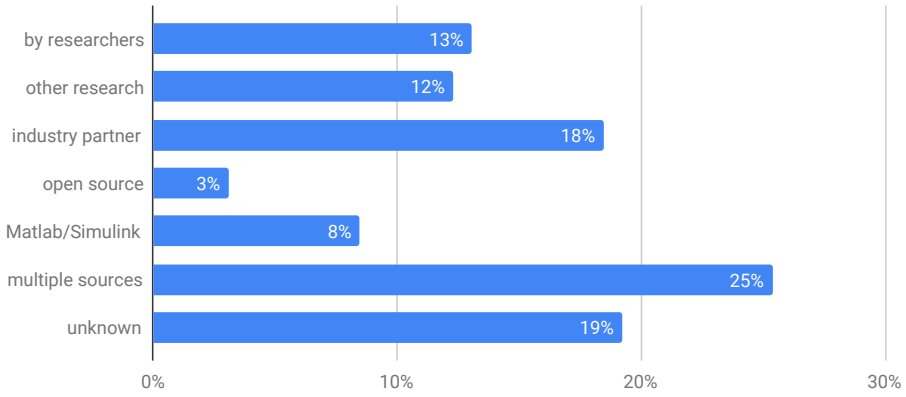


Fig. 6: RQ 2.2: From where are the models obtained?

An interesting yet partially expected perspective arises from combining RQ2.2 and RQ1.2. None of the models obtained from an "industry partner" are accessible. Three papers which we classified as "multiple" in RQ2.2 did provide industrial models though: [33] provides models from a "major aerospace and defense company" (name not revealed due to a non-disclosure agreement), while [1] and [2] use an open source model of electro-mechanical braking provided by Bosch in [44]. Finally, [3] and [27] use models for an advanced driver assistance system by Daimler [4], that can be inspected on the project website.[8]

RQ 2: From where do researchers acquire MATLAB/Simulink models for the sake of experimentation?
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 MATLAB/Simulink or a toolbox and only 3% used open source models. We could not determine 19% of the models' origins.

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

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 [25] and [41] and using wildcarded 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 MATLAB/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 Section 2.2. We also merged our answers of "no" and "unsure" for RQ1.4 and RQ1.5 in reflection of this problem.

More generally, the data collection process was done by two researchers, each of which may have overlooked important information or misinterpreted a concrete research question. 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.

Our methodology section does not present a separate quality assessment which is typical in systematic literature review studies [42]. 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" [40]) was done in our inclusion/exclusion process.

In this study, we focused on the accessibility of models and tools for replicating findings. Another critical part of replicating results, is the concrete experimentation setup. We did not analyze this aspect, here. Thus the number of studies deemed replicable may even be overestimated.

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 MATLAB/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.

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 MATLAB/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."[45]; "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)."[3]; "[...] most public domain Stateflows are small examples created for the purpose of training and are not representative of the models developed in industry."[19]; or "Such benchmarking suites are currently unavailable [...] and do not adequately relate to real world models of interest in industrial applications."[36] 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."[20] or "[...] no study of existing Simulink models is available [...]."[8,7] discuss the lack of accessible MATLAB/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 [22,21]. 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 can not 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 and had to rate them as "unsure" or "no" in RQ1.2 and RQ1.4. 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 [49] is in German. Four papers referenced pages that did not exist anymore, e.g., a private Dropbox[9] account. These issues can be easily addressed by a more thorough archiving of a paper's replication package.

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 MATLAB/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. 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 [5]. Altogether we downloaded 517 MATLAB/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 MATLAB/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. [12] in 2013. They conducted a systematic mapping study in which they analyzed papers about quality assurance of MATLAB/Simulink models. This is a sub-scope of our inclusion criteria, see Section 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 Section 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

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

RQ1: "In addition, examples from industry are sometimes only described, but not shown in detail for reasons of confidentiality."

The lack of publicly available MATLAB/Simulink models inspired the SLforge project to build the only large-scale collection of public MATLAB/Simulink models [8,10] 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 MATLAB/Simulink tool environment itself (see, e.g., [9]). Another interesting approach was used by [39]. They used Google BigQuery[10] to find a sample of the largest available MATLAB/Simulink models on GitHub.

In a different context focusing on UML models only, Hebig et al. [17] 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. [29].

7 Conclusion and Future Work

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 supporting MATLAB/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. Moreover, only a minor fraction of the models is obtained from open source projects. 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 MATLAB/Simulink. 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 a much larger corpus of MATLAB/Simulink models. However, besides some basic metrics, such as the number of blocks and connections comprised by these models, little is known about the methods and processes being adopted in the development of these models.

One potential data source which, to the best of our knowledge, has not yet been investigated in detail with a particular focus on MATLAB/Simulink and which we want to consider in our future work are open development platforms such as GitHub. They may not only host further models which are not yet included in any of the existing model collections, but also provide plenty of meta-data which can be exploited for a much more detailed characterization of the extracted models and projects.

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

References

1. Arrieta, A., Wang, S., Arruabarrena, A., Markiegi, U., Sagardui, G., Etxeberria, L.: Multi-objective black-box test case selection for cost-effectively testing simulation models. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 1411 – 1418. GECCO '18, Association for Computing Machinery, New York, NY, USA (2018)
2. Arrieta, A., Wang, S., Markiegi, U., Arruabarrena, A., Etxeberria, L., Sagardui, G.: Pareto efficient multi-objective black-box test case selection for simulation-based testing. Information and Software Technology **114**, 137 – 154 (2019)
3. Bertram, V., Maoz, S., Ringert, J.O., Rumpe, B., von Wenckstern, M.: 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 (2017)
4. Bertram, V., Maoz, S., Ringert, J.O., Rumpe, B., von Wenckstern, M.: Component and connector views in practice: an experience report. In: 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS). pp. 167–177. IEEE (2017)
5. Boll, A., Kehrer, T.: The download link of all digital artifacts of this paper, https://www.informatik.hu-berlin.de/de/forschung/gebiete/mse/ICSMM2020_replication_package/at_download/file
6. Brambilla, M., Cabot, J., Wimmer, M.: Model-driven software engineering in practice. Synthesis lectures on software engineering **3**(1), 1–207 (2017)
7. Chowdhury, S.A.: 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 (May 2018)
8. Chowdhury, S.A., Mohian, S., Mehra, S., Gawsane, S., Johnson, T.T., Csallner, C.: 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 (May 2018)
9. Chowdhury, S.A., Shrestha, S.L., Johnson, T.T., Csallner, C.: SLEMI: Equivalence modulo input (EMI) based mutation of CPS models for finding compiler bugs in Simulink. In: Proc. 42nd ACM/IEEE International Conference on Software Engineering (ICSE). ACM. To appear (2020)
10. Chowdhury, S.A., Varghese, L.S., Mohian, S., Johnson, T.T., Csallner, C.: 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). pp. 45–48. IEEE (2018)
11. Cohen, J.: A coefficient of agreement for nominal scales. Educational and psychological measurement **20**(1), 37–46 (1960)
12. Elberzhager, F., Rosbach, A., Bauer, T.: 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. p. 29–34. JAMAICA 2013, Association for Computing Machinery, New York, NY, USA (2013)
13. France, R., Rumpe, B.: Model-driven development of complex software: A research roadmap. In: Future of Software Engineering (FOSE'07). pp. 37–54. IEEE (2007)
14. Gallego-Calderon, J., Natarajan, A.: Assessment of wind turbine drive-train fatigue loads under torsional excitation. Engineering Structures **103**, 189 – 202 (2015)

15. Gerlitz, T., Kowalewski, S.: Flow sensitive slicing for MATLAB/Simulink models. In: 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA). pp. 81–90 (April 2016)
16. Gusenbauer, M., Haddaway, N.R.: Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed and 26 other resources. Research Synthesis Methods (2019)
17. Hebig, R., Quang, T.H., Chaudron, M.R., Robles, G., Fernandez, M.A.: 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 (2016)
18. Holling, D., Hofbauer, A., Pretschner, A., Gemmar, M.: Profiting from unit tests for integration testing. In: 2016 IEEE International Conference on Software Testing, Verification and Validation (ICST). pp. 353–363 (April 2016)
19. Hussain, A., Sher, H.A., Murtaza, A.F., Al-Haddad, K.: Improved restricted control set model predictive control (iRCS-MPC) based maximum power point tracking of photovoltaic module. IEEE Access **7**, 149422–149432 (2019)
20. 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 (July 2017)
21. Kehrer, T., Kelter, U., Ohrndorf, M., Sollbach, T.: Understanding model evolution through semantically lifting model differences with SiLift. In: 28th IEEE International Conference on Software Maintenance (ICSM). pp. 638–641. IEEE (2012)
22. Kehrer, T., Kelter, U., Pietsch, P., Schmidt, M.: Adaptability of model comparison tools. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. pp. 306–309. IEEE (2012)
23. Khelifi, A., Ben Lakhal, N.M., Gharsallaoui, H., Nasri, O.: Artificial neural network-based fault detection. In: 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT). pp. 1017–1022 (April 2018)
24. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
25. Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O.P., Turner, M., Niazi, M., Linkman, S.: Systematic literature reviews in software engineering–a tertiary study. Information and software technology **52**(8), 792–805 (2010)
26. Kuroki, Y., Yoo, M., Yokoyama, T.: A Simulink to UML model transformation tool for embedded control software development. In: IEEE International Conference on Industrial Technology, ICIT 2016, Taipei, Taiwan, March 14-17, 2016. pp. 700–706. IEEE (2016)
27. Kusmenko, E., Shumeiko, I., Rumpe, B., von Wenckstern, M.: Fast simulation preorder algorithm. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development. p. 256–267. MODELSWARD 2018, SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT (2018)
28. Lamprecht, A.L., Garcia, L., Kuzak, M., Martinez, C., Arcila, R., Martin Del Pico, E., Dominguez Del Angel, V., van de Sandt, S., Ison, J., Martinez, P.A., et al.: Towards fair principles for research software. Data Science (Preprint), 1–23 (2019)
29. Langer, P., Mayerhofer, T., Wimmer, M., Kappel, G.: On the usage of UML: Initial results of analyzing open UML models. Modellierung 2014 (2014)
30. Liggesmeyer, P., Trapp, M.: Trends in embedded software engineering. IEEE software **26**(3), 19–25 (2009)
31. Matinnejad, R., Nejati, S., Briand, L.C., Bruckmann, T.: Automated test suite generation for time-continuous Simulink models. In: 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE). pp. 595–606 (May 2016)

32. Matinnejad, R., Nejati, S., Briand, L.C., Bruckmann, T.: Test generation and test prioritization for Simulink models with dynamic behavior. IEEE Transactions on Software Engineering **45**(9), 919–944 (Sep 2019)
33. Nejati, S., Gaaloul, K., Menghi, C., Briand, L.C., Foster, S., Wolfe, D.: 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. p. 1015–1025. ESEC/FSE 2019, Association for Computing Machinery, New York, NY, USA (2019)
34. Norouzi, P., Kıvanç, Ö.C., Üstün, Ö.: 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 (Nov 2017)
35. Oussalem, O., Kourchi, M., Rachdy, A., Ajaamoum, M., Idadoub, H., Jenkal, S.: A low cost controller of PV system based on Arduino board and INC algorithm. Materials Today: Proceedings (2019)
36. Rao, A.C., Raouf, A., Dhadyalla, G., Pasupuleti, V.: Mutation testing based evaluation of formal verification tools. In: 2017 International Conference on Dependable Systems and Their Applications (DSA). pp. 1–7 (Oct 2017)
37. Rashid, M., Anwar, M.W., Khan, A.M.: Toward the tools selection in model based system engineering for embedded systems—a systematic literature review. Journal of Systems and Software **106**, 150–163 (2015)
38. Rebaya, A., Gasmi, K., Hasnaoui, S.: 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 (2018)
39. Sanchez, B., Zolotas, A., Rodriguez, H.H., Kolovos, D., Paige, R.: 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). pp. 205–215. IEEE (2019)
40. Shaw, M.: What makes good research in software engineering? International Journal on Software Tools for Technology Transfer **4**(1), 1–7 (2002)
41. Silva, R., Neiva, F.: Systematic literature review in computer science - a practical guide (11 2016)
42. Stapić, Z., López, E.G., Cabot, A.G., de Marcos Ortega, L., Strahonja, V.: Performing systematic literature review in software engineering. In: CECIIS 2012-23rd International Conference (2012)
43. Stephan, M., Cordy, J.R.: Identifying instances of model design patterns and antipatterns using model clone detection. In: Proceedings of the Seventh International Workshop on Modeling in Software Engineering. p. 48–53. MiSE '15, IEEE Press (2015)
44. Strathmann, T., Oehlerking, J.: Verifying properties of an electro-mechanical braking system. In: 2nd Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH 2015) (4 2015)
45. Tomita, T., Ishii, D., Murakami, T., Takeuchi, S., Aoki, T.: 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 (May 2019)
46. Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S.: Model-driven software development: technology, engineering, management. John Wiley & Sons (2013)

47. Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R.: 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 (2013)
48. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. Scientific data **3** (2016)
49. Wille, D., Babur, Ö., Cleophas, L., Seidl, C., van den Brand, M., Schaefer, I.: Improving custom-tailored variability mining using outlier and cluster detection. Science of Computer Programming **163**, 62 – 84 (2018)
50. Yang, Y., Jiang, Y., Gu, M., Sun, J.: Verifying Simulink Stateflow model: Timed automata approach. In: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. p. 852–857. ASE 2016, Association for Computing Machinery, New York, NY, USA (2016)