Information and Software Technology 164 (2023) 107317

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A reflection on the impact of model mining from GitHub

Gregorio Robles [a,*], Michel R.V. Chaudron [b], Rodi Jolak [c], Regina Hebig [d]

[a] *Universidad Rey Juan Carlos, Madrid, Spain*
[b] *TU Eindhoven, Eindhoven, The Netherlands*
[c] *RISE - Research Institutes of Sweden, Gothenburg, Sweden*
[d] *Universität Rostock, Rostock, Germany*

ARTICLE INFO

Dataset link: https://gsyc.urjc.es/grex/2023-ist-reflections-impact-model-mining

Keywords:
Modeling
UML
Mining software repositories
Empirical research

ABSTRACT

Context: Since 1998, the ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS) has been studying all aspects surrounding modeling in software engineering, from languages and methods to tools and applications. In order to enable empirical studies, the MODELS community developed a need for having examples of models, especially of models used in real software development projects. Such models may be used for a range of purposes, but mostly related to domain analysis and software design (at various levels of abstraction). However, finding such models was very difficult. The most used ones had their origin in academic books or student projects, which addressed "artificial" applications, i.e., were not base on real-case scenarios. To address this issue, the authors of this reflection paper, members of the modeling and of the mining software repositories fields, came together with the aim of creating a dataset with an abundance of modeling projects by mining GitHub. As a scoping of our effort we targeted models represented using the UML notation because this is the *lingua franca* in practice for software modeling. As a result, almost 100k models from 22k projects were made publicly available, known as the Lindholmen dataset.
Objective: In this paper, we analyze the impact of our research, and compare this to what we envisioned in 2016. We draw practical lessons gained from this effort, reflect on the perils and pitfalls of the dataset, and point out promising avenues of research.
Method: We base our reflection on the systematic analysis of recent research literature, and especially those papers citing our dataset and its associated publications.
Results: What we envisioned in the original research when making the dataset available has to a major extent not come true; however, fellow researchers have found alternative uses of the dataset.
Conclusions: By understanding the possibilities and shortcomings of the current dataset, we aim to offer the research community i) future research avenues of how the data can be used; and ii) raise awareness of the limitations, not only to point out threats to validity of research, but also to encourage fellow researchers to find ideas to overcome them. Our reflections can also be helpful to researchers who want to perform similar mining efforts.

1. Summary and main contributions of the original paper

Before 2016, having examples of models, especially of models used in real projects, was a difficult task. There were very few models that could be used as examples. The most used ones had their origin in academic books, which addressed "artificial" applications, i.e., were not base on real-case scenarios. The need for having datasets with models can be seen on several previous initiatives to collect datasets, which are often limited in the number of collected models [1]. The collections of models were usually fed from examples from software engineering books (including text books on UML), and diagrams gathered with the help of Google Images. However, the number of models was limited (e.g., in 2016 the largest dataset contained more than 800 class diagrams [2]) and, among them, those that came together with other elements of a software project (in particular, but not only, its source code) were even fewer.

To address this issue, members of the modeling and from the mining software repositories fields worked together with the aim of creating a dataset with an abundance of modeling projects. The expertise of the members of the modeling field lied in the fact of having searched for years for models. The expertise of the miners was in how to analyze GitHub projects and to obtain as many files as possible that could potentially contain models.

[*] Corresponding author.
E-mail address: gregorio.robles@urjc.es (G. Robles).

https://doi.org/10.1016/j.infsof.2023.107317
Received 29 September 2022; Received in revised form 26 May 2023; Accepted 1 August 2023
Available online 10 August 2023

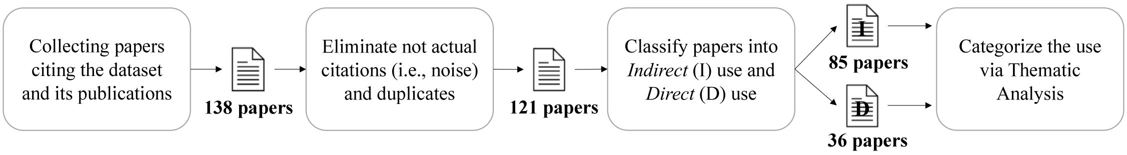

Collecting papers citing the dataset and its publications
138 papers
Eliminate not actual citations (i.e., noise) and duplicates
121 papers
Classify papers into *Indirect* (I) use and *Direct* (D) use
I
85 papers
D
36 papers
Categorize the use via Thematic Analysis

Fig. 1. Method.

In 2016, we published a paper at MODELS in which we presented "a method to systematically mine for UML models in GitHub [...] that [...] leads to an enormously promising set (much larger than any existing set of projects) for future analysis" [3]. This paper has been cited over 100 times, as of Google Scholar in July 2022. We had analyzed 10% of GitHub with over 21k files with models from 3k GitHub projects. This effort was augmented with a data paper presented at MSR 2017 where all GitHub projects had been analyzed for UML models [4]. The final dataset, named the *Lindholmen dataset*, which included the files and metadata on the projects where they belonged, counted with almost 100k files from over 20k projects. We considered the dataset to be relevant for researchers in the area of software design and modeling, because the whole community lacks good examples of not just models, but software systems that are built with the help of models as well. The data paper has been cited by almost 50 papers (Google Scholar, July 2022).

2. Visions in the original paper

In our original work, we envisioned that our work may have impact in three areas:

(1) Use of UML: The dataset could be used to study how UML is used and to develop guidelines for UML beginners. The data could be used to learn what model layouts Open Source Software (OSS) developers use and which models are average sized. The study of the UML that appears in the images could also provide insights into the needs of OSS developers for visual highlighting strategies. For example, when observing images manually, we saw many UML images that use color for emphasis. Also, given the availability of the models (and the projects they belong to), the dataset allowed to analyze how the code and the models relate to each other. We wanted to know how much of a software system is typically covered by models and to what extent models abstract code.

(2) Advantages and disadvantages of UML: The dataset could be the basis for empirical studies on the advantages and disadvantages of UML (and modeling). Our dataset could contribute to enriching this research, which qualitatively examines individual cases, with quantitative studies. In addition, the dataset could help examine more generally how the use of UML modeling affects code structure and whether improvements in software quality and productivity can be observed with the adoption of UML.

(3) Evaluation of scientific approaches and modeling tools: Constructive research on software modeling often had the problem that there are not enough real cases of models to evaluate newly developed approaches and techniques. Before 2016, this constraint was solved using examples of artificially created toys or models. In rare cases, researchers may use obfuscated industry models or models created with the help of experts for assessment purposes. We envisioned that our dataset would provide real-world instances of UML models in machine-readable format. Professional tool providers providing case tools for modeling could use the dataset to test new features on real data, e.g., design generation.

3. Method

The method used to achieve the goal of this study is illustrated in Fig. 1. First, we created a script, based on Google Scholar, to collect the papers citing our dataset and its publications [3,4]. The collection of these papers was performed in December 2022. After collecting the papers, we eliminated 17 papers. Most of these eliminated papers were either not actual citations (i.e., noise due to a bug in Google Scholar's algorithm) or duplicates. Moreover, we left out papers that were written by the same author team as the original papers, so not as to inflate the impact from self-citations of papers that can be seen as extensions of the original set of papers. However, we did include papers that were co-authored by some of us in the context of new collaborations or as a result of supervising students. As a result, we ended up with 121 papers. Second, we created a spreadsheet and added all the remaining papers from the previous stage. We added the title of the papers, URL to the papers, and some meta-data such as the authors and their affiliation as well as country of employment, and year of publication of the papers. Third, we divided the papers into two parts: (i) indirect- and (ii) direct-use of the dataset. Here the indirect-use part contained papers referring to our research; including its used tools, methods, and findings. This stage assigned 36 of the papers to the direct-use category and 85 papers to the indirect-use category. Fourth, we created a short description of how papers used our dataset and cited our publications.

Finally, we conducted a thematic analysis [5] to identify, analyze, and categorize the indirect- and direct-use of the dataset. For this thematic analysis, we split the author team initially into two groups. The first group was concerned with analyzing papers that seemed to directly use our dataset, while the other group analyzed papers from the indirect-use category. For the analysis, we read all the papers in each category, with the objective to identify the purpose of the use. We performed *inductive coding*, i.e., we went in without predefined codes and allowed the codes to emerge from the analysis. We coded each paper with one code independently of the other. Later we met to discuss the codes and one author per group proposed how these codes could be merged in a meaningful way. The proposal was then improved iteratively in teamwork until we agreed on the result. For the direct-use papers, we quickly identified that they differed also in 'what' was used, e.g., the whole dataset vs. selected models. We, therefore, added this aspect to the analysis as we perceived this as an important difference to highlight. The complete thematic analysis was done using shared spreadsheets. In the end, both groups mutually reviewed the analysis of the other team. One author from the group initially analyzing the direct use joined the group analyzing the indirect-use to finalize the analysis, to ensure consistency between both analyses, and to enable us to detect potential overlaps between direct- and indirect-use.

4. Papers using our dataset (direct use)

We identified 36 publications that directly use (or plan to use) models from the Lindholmen dataset. This includes 9 journal publications, 16 conference and workshop publications, and 11 technical reports, bachelor's-, master's-, and Ph.D.-theses. It should be noted that there are 7 publications co-authored by the authors of this paper, that reflect their own research agenda. Table 1 summarizes these publications sorted by research groups (i.e., groups of authors).

Observation O1 multiple studies per research group We can observe that nearly half of the research groups (7) use the dataset in more than one

Table 1
Summary of research groups using the Lindholmen dataset directly. The group marked with * belongs to the authors of this paper. The number of theses and technical reports reported in the parentheses represents those works that have not also been published in a peer reviewed publication.

Group	Country	# Theses/ Tech Reports (not peer-reviewed)	# Conferences/ Workshops	# Journals	Use of What	Usage Category
Kretschmer et al.	Austria, France	1 (0)	2	2	Models (HP)	Evaluating Algorithms
Arora et al.	India	0	1	1	Models (SC)	Evaluating Algorithms
Ott et al.	USA	0	0	2	Models (SC)	Diagram Classification
Shcherban et al.	China	0	1	1	Models (SC)	Diagram Classification
Ahmar et al.	France, Sweden	1 (0)	2	0	Models (SC)	Phenomenon Investigation
Chen et al.	China, USA	0	1	1	Models (SC), Projects (HP)	Phenomenon Investigation, Evaluating Algorithms
Chaudron et al.*	Sweden, The Netherlands, Malaysia	4 (3)	2	1	Models (SC), Models (RS), Projects (SC)	Phenomenon Investigation, Diagram Classification, Evaluating Algorithms
Schulze et al.	Germany	2 (0)	1	0	Models (RS)	Evaluating Algorithms
Leigh et al.	UK	1 (0)	0	1	Projects (HP)	Study of Usefulness
Oliveira Barbosa et al.	Brazil	1 (0)	1	0	Projects (HP)	Phenomenon Investigation
Torre et al.	Canada, Luxembourg, Spain	1 (0)	1	0	Models (SC)	Phenomenon Investigation
Tavares et al.	Brazil	0	1	0	Models (SC)	Diagram Classification
Vega et al.	Spain, UK	0	1	0	Models (SC)	Evaluating Algorithms
Stephan	USA	0	1	0	Models (SC)	Knowledge-base
Mangaroliya et al.	India	0	1	0	Models (SC)	Diagram Classification

study. In one research group (the authors group), the dataset was used in three technical reports (bachelor, master's, or PhD thesis) that are still not published as peer-reviewed publications. In the other half of the research groups (8) the dataset was used for a single peer-reviewed publication each. In half of these cases, there are also theses or technical reports covering the same studies as well.

4.1. What part of the dataset is used?

We can distinguish studies that use models from studies that use complete projects for their research.

The majority of papers (23) (≈82%) uses only *models* from the dataset. This can take different forms. In 4 of these cases only models from a few and often *handpicked (HP)* set of projects are used. For example, Kretschmer et al. [6] use 18 models, of which only 2 are from our dataset. 17 papers define *selection criteria (SC)* to collect a subset of the models from the dataset. These criteria can be diagram type, e.g., papers working with a single diagram type (observed for activity diagrams, sequence diagrams, and class diagrams), or file format, e.g., image files. Finally, 2 papers use a *random selection (RS)* of models from the dataset (often after first applying a selection criterion). Zooming out, 6 of the papers use also models from other sources in addition to models from our dataset.

In 5 papers (≈ 17%) not just models, but also the corresponding *projects*, including the source code, are used for the studies. In 3 of these papers a set of single projects are *handpicked (HP)*. For example, Oliveira Barbosa et al. [7] use 3 projects, of which 2 are from our dataset. In the 2 papers where projects are not handpicked, *selection criteria (SC)* are used such as the number of issues in the issue tracking system. Just as for papers using models, also two of the papers using projects, utilize in addition other projects that do not stem from our dataset.

There are 6 publications that are co-authored by the authors of this paper. These publications include 3 technical reports that are not peer reviewed publications. In 4 papers, the authors define *selection criteria (SC)* to collect a subset of the models from the dataset. In 2 papers, the models and the corresponding projects are used for the studies.

Observation O2 little combined use of models and source code When first creating our dataset, there were already datasets out there that offered collections of models to be studied by researchers. However, there was a lack of available models that could be studied in the context of their projects, e.g., together with the corresponding source code. This is something that we wanted to change. Thus, it is surprising for us to see that only a small share of the researchers make use of the fact that the dataset allows studying models *together with* the source code. One reason for such an outcome is that the code of those projects is not part of the dataset, as only links to the models and the repositories are given. Interested researchers need to retrieve those from GitHub, and, depending on their needs, maybe access an old version of the code, e.g., the one that corresponds to the moment when the UML diagram was introduced. Although multiple tools exist that allow to mine and analyze GitHub repositories, we acknowledge that we might have assumed that researchers are familiar with these (as in the case of the mining software repositories field) — this has not to be the case in general in software engineering research.

Observation O3 labeling efforts happen Two research groups manually labeled the data from our dataset further. One group did that by labeling what type of diagram a model belonged to. In the other case, trace links were created by the researchers.

Observation O4 combination with other sources 5 papers handpick models or projects from our dataset and combine that selection with models and projects from other sources. For example, Khelladi et al. [8] use our dataset to select three projects from which they take models for evaluating their algorithm for repairing model inconsistencies. In addition, they select models from one academic project and ten industrial projects (presumable from industry collaborators). Chen et al. [9] use selected projects from our dataset and the dataset of Karasneh et al. [10] to evaluate their algorithm on trace-link detection. Another example comes from Barbosa et al. [7] who study the evolution of source code and state machines using 2 projects selected from our dataset, as well as one project directly selected from GitHub. In 3 papers the models from our dataset are mixed with models found in other sources to form a bigger dataset. For example, the group of Shcherban et al. [11] train an ML/AI classifier, working with images originating from our dataset, from the datasets in [12,13], and from the Kaggle Graphs dataset [14]. Tavares et al. [15] use, in addition to images from our dataset, images they found using web scrapping scripts on google.com and bing.com.

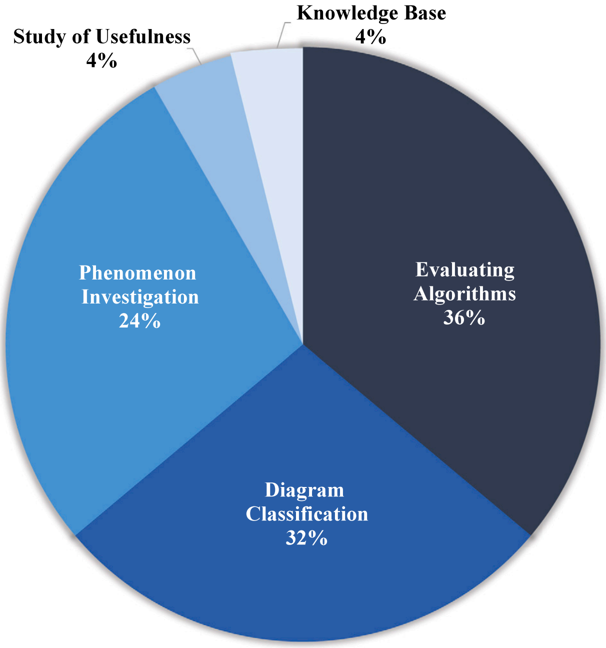


Fig. 2. What is the data used for?.

Observation O5 dataset used in vs. outside of our network Generally, we did not observe any divergence in what part of the dataset is used. In particular, around 60% of the publications in both our and others' network define selection criteria to use models form the datasets. 30% of the publications in our network use the complete projects compared to 8% in publications outside of our network.

4.2. What is the data used for?

The data is used for multiple purposes as it can be seen from Fig. 2.

In 9 papers the data is used to *evaluate algorithms* developed by the researchers. These are, for example, algorithms for detecting and repairing inconsistencies between models, algorithms for synthesizing test scenarios from activity diagrams or algorithms for automatic layout management of diagrams, and algorithms for the generation of trace links.

A second big group of 6 papers use the dataset for training and testing of ML/AI models for the *Classification of UML diagrams*. This classification targets for example the question of what diagram type a model belongs to. Another purpose is to classify diagrams based on other criteria, such as the quality of the layout, or the role types of shown classes in the diagram.

5 papers present studies that use the dataset for empirical *investigations of different phenomena*. These phenomena are, for example, visual variables used in models, the use of design models in Open Source systems, the evolution of state machines through a project's life cycle, or the defect proneness and quality of projects with models.

One research group *investigates the usefulness* of different diagrams for supporting architects in a risk prediction analysis (1 paper).

One paper plans to use the models in the dataset as a *knowledge-base* for the development of a modeling assistant.

Finally, regarding the publications that are co-authored by the authors of this paper and including their technical reports that are not peer reviewed publications, 3 papers use the dataset for training and testing of ML/AI models for the *Classification of UML diagrams*, 2 papers for empirical *investigations of different phenomena*, and one paper to *evaluate algorithms* developed by the researchers.

Observation O6 uniform usage within research groups If a group writes multiple papers using the dataset, the group tends to use the dataset the same way, in every paper. For example, the group around Kretschmer et al. utilizes the same handpicked set of models throughout 4 papers to evaluate their algorithms.

An exception to that rule is the group around Chen et al. who use selected models from the dataset in one paper to investigate the phenomenon of model use in open-source systems. In their second paper, however, they use selected projects to evaluate an algorithm for trace-link detection. The other exception is the group around our co-author Chaudron et al. which is not surprising as we, as authors, are more motivated to explore different ways the dataset can be used in a targeted way.

Observation O7 ML/AI-based diagram classification is much more prevalent than expected We are surprised to see that many publications that are concerned with training ML/AI models for the classification of diagrams. This can be probably explained by the increased interest and by the higher ease of use of ML frameworks. On the one hand, it is surprising how much the dataset is seen as a source for big data and, thus, an opportunity to train ML algorithms. Specifically, we expected researchers to be more interested in models stored in machine-readable formats, such as XMI. However, it seems that in the end models stored in image formats have been much more attractive for researchers.

Observation O8 diversity of phenomena investigated One very positive observation is the large diversity of phenomena studied based on our dataset. We did not predict most of these uses, but of course, had hoped that there is a wide range of possible usages for our data in various empirical studies.

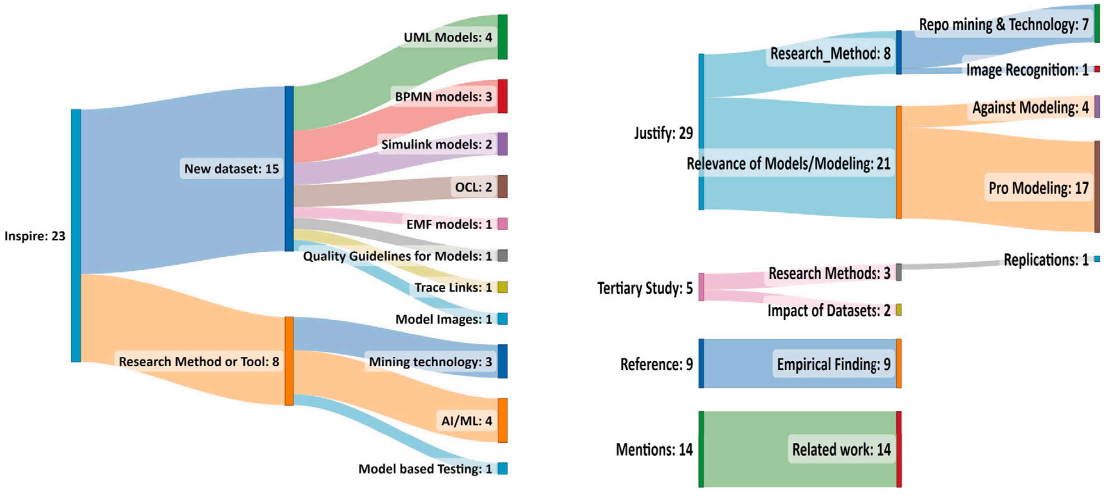
Observation O9 use of dataset in vs. outside of our network We observe that around 50% of the publications in our network use the dataset for classification of UML diagrams. In contrast, around 50% of the publications outside of our network use the dataset for evaluating algorithms developed by the researchers.

5. Papers using our publications (indirect use)

In the set of citing papers, a collection of 85 papers do not use our dataset, but base their research or research method on our work. We call this 'indirect use'. The indirect use itself, can be classified into papers that (i) are inspired in their research by our paper, or (ii) papers that justify their direction of research based on our paper. Of these 85 papers, 5 have been authored or co-authored by the authors, so we have left them out of this analysis to avoid bias. Two graphs depicting a thematic grouping of these papers are shown in Figs. 3(a) and 3(b).

In the 'inspiration set', we recognize a few sub-groups: one group of 15 papers mimics our idea, to build datasets of models, but mostly (11 out of 15) for different types of models than UML as in our paper. In particular, datasets have been created for BPMN models, EMF models, Simulink models, clones of models, and UML models with OCL expressions. One dataset defines a set of quality guidelines for models. Also 3 papers present an alternative: they construct, using their own mining methods, alternative datasets of UML models. A second group of 8 papers use our paper as inspiration for developing new methods and tools. More specifically, these include new mining technologies, AI and ML algorithms for UML models and one paper describes a tool for testing models.

The other class of papers (29) are papers that draw from our paper to justify some aspect of their own research. The majority (22) of these papers use our paper to justify that models and modeling are relevant topics for research (17 out of 21), or conversely that models and modeling are not relevant for research/practice (4 out of 21). Another part of these papers (8) justify some aspect of their research method based on our paper, for example: feasibility of mining GitHub (for models), focusing on diagrams as representation of models, reporting



(a) Inspiration for Research

(b) Justification for Research

Fig. 3. Sankey Diagrams showing the number of papers by types and subtypes of *indirect use* of the Lindholmen dataset. 5 papers authored or co-authored by the Lindholmen dataset have been left out.

links to projects, the need for curating projects on GitHub, or the use of file-types for finding particular content.

Our papers are mentioned in 5 tertiary studies. Three of these studies address Research Methods (on Repository Mining, or on Replicability). Two studies look at the impact or use of large datasets.

7 papers referr to particular empirical findings in our paper, either as context or as a reference for comparison. Particular findings that are referenced include: Timing-characteristics of creation and frequency of updating of models. More general empirical claims refer to: the popularity of GitHub and Open Source projects, and GitHub API restrictions as challenge (a step in mining-toolchains).

We can observe in Figs. 3(a) and 3(b) 14 papers that mention/relate to our paper in a more loose or casual manner. These types of mentions consist of: (i) as example of Open Source repositories, as example of a (large scale) mining approach, and as example of an open dataset; and (ii) as a work that is relevant background, or just as a mention in related work.

Observation O10 miss-citations 4 papers cite our work to justify that software models are not relevant. For that, they refer to the fairly low percentage of projects we identified to have models in GitHub. This happens, despite the fact that we discussed in the threats to validity section of our original paper that this number is an underestimate and that it, therefore, cannot be used to make statements about the frequency of model usage. Thus, one lesson learned is that we should, in the future, place such disclaimers right next to the presentation of the observations themselves, to make it easier for other researchers to find that information in the paper.

6. Research network

One of the aims of our reflection is to observe how our papers have impacted the software engineering community. We want to see how far our ideas have gone beyond our own research groups and our collaborators. Therefore, we have taken all citing publications found in the previous step, and have looked for them in DBLP. We found a total of 93 publications in DBLP, as master's and bachelor's theses are not included in that database. However, given that DBLP is very complete for relevant computer science publications, we argue that those 93 publications are thus the most important ones among the total population of citing publications.

For those publications that show up in DBLP, we took the authors and stored their co-authors. Some authors, esp. students, are often not found in DBLP. But, again, we find this does not introduce a big bias in our results, as computer science researchers appear in DBLP when they start having a relevant publication. Then, we divided the authors into two sets depending on if they have a common publication in their record with any of the six authors of the original papers. The two sets are: (a) a set A with co-authors (which could be considered as the research network of the authors), and (b) a set B with non-co-authors. The size of set A is composed of 36 researchers, while the size of set B is 181.

Within our research network (set A), 13 out of the 36 researchers have made direct use of the dataset, while 23 have used it only indirectly. However, interestingly enough, 11 out of those 13 used the dataset only in direct collaboration with some of the authors of the original papers, i.e., they have co-authored at least one of the citing papers with them. This means, that only 2 researchers from our research network have used the dataset directly without co-authoring with the original authors. Both belong to two different research groups. One is Djamel Eddine Khelladi, whose research group with 4 collaborators (all from set B) has authored 5 publications using the dataset. The second one is Marcela Genero, who has collaborated as well with other 4 researchers (all from set B) in 2 publications using the dataset. The other 11 researchers belong to three different research groups, where each of them had a direct collaboration with at least one of the authors of the dataset.

From our previous analysis (Section 4), we obtained that in total 14 research groups (excluding the group of Chaudron) have made direct use of the database. Of these, 12 research groups are not part of the collaboration network of the original authors have been the ones using the database for their research.

Observation O9 main use of the dataset outside of our network The majority (12 out of 14) of research groups using our dataset do not include the original authors as co-authors. The more innovative, and for us unexpected uses of the dataset, come from research groups that are not directly part of our network.

7. Reflections

In this section, we revisit and reflect on our visions as well as further aspects that arise from this investigation.

7.1. Reflections on our original visions

First, we look at our original visions (as described in Section 2):

Vision #1 focused on the use of UML, how a vast amount of examples could foster its understanding, and how it is used in practice. Very few groups address this (mainly subsequent works by the authors – together or with other groups – and another group of researchers). We have witnessed some few enrichments to the dataset, although not many and, in any case, not of fundamental nature. We have learned that the dataset format makes it difficult to enhance the data, by further annotating and enriching the dataset. Some publications have even noted that up to 30% of the links provided do not work anymore (as files or even repositories have been removed from GitHub) [15,16]. This is a known problem in the research literature [17]. A possible solution that would have avoided this problem is to provide the actual data rather than links to it. Although that would make the dataset much larger in size, it would benefit from perpetuating consistency. The data with the repositories (probably in the range of hundreds of gigabytes) could be distributed in a separate file to avoid having to download them if not required. All things considered, this brings the question of quality assurance for this type of dataset. Maintenance is not easy to perform for such datasets, and, thus, there is no feedback loop as many development models (the Open Source and the agile technologies) have shown to be beneficial.

All in all, we question if mining software repositories is the best method for the type of research we envisioned. Much of the information can be obtained by alternative methods, such as interviews, which are more prone to offer the type of lessons learned that we intended. Our material from the dataset can complement these findings but is not that suited to be the main driver of these advancements.

Vision #2 dealt with the advantages and disadvantages of UML. This aspect has been pointed out in several papers that cite our work. However, they directly use the findings but not the dataset. Our reflection is that there is still a somewhat *religious* debate on the benefits and limitations of UML. We hoped to have this debate based on data, but very few papers really base their arguments on those. We have seen how our research has been both used in favor of and against UML. We think that, in particular, those papers that point out that UML is seldom used do not make a *fair* interpretation of our research. Considering our results in absolute terms (i.e., the number of repositories where we have found models) is misleading. This is because we do not capture the whole picture of modeling because of using a mining software repositories approach.

Software repositories, and in particular code repositories, do not contain a lot of relevant information that is of importance in modeling: we do not know the goal of the model and we do not see how models are used, among others. For this kind of task, a mixed methods approach would be more appropriate. Our conclusion is that we were very optimistic thinking that we had a lot of data (actually we do), but beyond doubt this data is partial. Versioning systems have been designed with (textual) code in mind, mostly with the granularity of line of code (although some even do it at the token level). Models are of a different nature. They are generally visual, so the results are usually images, and versioning systems do not handle them very well as they are in binary format. Even the text-based UML formats do not make sense at the line level; their changes are somewhere in between the line and the file level. This consideration results in fewer models being included in the versioning system, and in very few being updated.

Vision #3 proposed the use of the dataset for evaluating scientific approaches and tools. Although we did not foresee it, our dataset has been used widely for refining machine-learning techniques on diagrams. It has also been used for refinement tools for analyzing the consistency of models. Still, we have found that our dataset contains too much noise: We still do not know how many real cases (actual serious software engineering projects) we have. We have tried to filter out student projects — but even if filtering out those, we do not know how close they are to reality, because of the reluctance to include models in versioning systems.

The question how software engineering practices in Open Source relate to practices in industrial software projects is an open questions to the software engineering community at large, and we do not have the definitive answer to that question. Indeed, our research carries this same threat to validity as many empirical studies with Open Source projects. Some proponents of studying Open Source claim that many projects are also of industrial nature (and a fair number of projects led by companies are known). In addition, some scholars have proposed a way to identify "engineered" projects on GitHub [18], which could be closer to industrial practices.

We know that in industry, practices for software modeling vary widely across projects [19]. This is reflected as well in the large variety of types and styles of models that are found in the Lindholmen dataset for Open Source projects as a corpus at large. For many models it holds that they are typical of the type of models that are also found in industry — in terms of size, complexity, and layout [20,21]. This assessment is currently mostly based on experience, and not performed using rigorous comparison techniques.

A possible source of differences is the number of models that are used in concert to model a larger software system. For our dataset we do not have data on whether models complement each other. For any given model, it is difficult to say whether it is similar or different from some industrial practice. Also, we selected models that were clearly recognizable as UML models. One source of variation in industry is the use of more informal modeling styles that may deviate from formal UML syntax.

7.2. Further reflections

In addition to the reflection on our original visions, we also have some further reflections on such datasets and the modeling and mining community.

There is a need for accessibility, quality, maintenance, and metadata. By understanding the possibilities and shortcomings of the current dataset, we aim to offer the research community (i) future research avenues of how the data can be used; and (ii) raise awareness of the limitations, not only to point out threats to validity of research, but also to encourage fellow researchers to find ideas to overcome them.

We have learned lessons about how a dataset of software engineering data should be. A major drawback that we have found in our dataset is the fact of its accessibility. We had in mind a comprehensive, complete dataset with all the models to be found in GitHub (including metadata of the repositories where these files have been found). However, although size matters, the result is a dataset that contains too much noise, and the amount of meta-data is scarce. In addition, because many models are images, the quality of the dataset is lower than expected. Thus, regarding accessibility, it is difficult to search and find based on keywords or content.

Thus, when creating a new dataset, researchers should consider the following aspects: the quality of the dataset (signal-to-noise ratio), the accessibility of the dataset (e.g., enabling online access (and querying to) the dataset), the maintenance/evolution of the dataset (i.e., is it likely that projects will go missing?), and what meta-data researchers would like to be embedded into the dataset.

Separate communities might be limiting the potential for synergies. Finally, we would like to stress the fact that we come from two different software engineering research fields: mining software repositories, which is very close to developers, and modeling, which is related to software architects. We have scanned DBLP for the names of the authors in the proceedings of the main two conferences of the modeling and mining software repositories research fields: the International Conference on

Model-Driven Engineering Languages and Systems (MODELS) and the International Conference on Mining Software Repositories from 2016 to 2022 (both years included). In total, we found that, in the last 7 years, 2,002 different authors have published in those conferences (1,351 in MSR and 685 in MODELS). Of those, only 34 have authored papers in *both* conferences (1.70%), and 6 of the 34 are the authors of the papers we are reflecting on. The intersection between those two research communities is, thus, very shallow. In addition, we have learned that these communities do not communicate much, both in industry and academia. As a matter of fact, they use different tooling, different textbooks, and different practices — in short, they follow a different culture.

This raises also some challenges for our dataset in comparison to other, similar efforts in the past. One such example is the *crowdsourcing the annotation* for datasets. In the area of bug research, for instance, developers started to link the bug fixes (when committed to a repository) with the bug issues (as found on the bug-tracking system), in a synergistic move where both the research community and the development community benefited. The idea is simple and requires not much labor if included in the development process: it requires just to include in the commit message something like "fixes bug #324", where #324 is the id of the bug in the bug-tracking system. Having a similar practice for models and code, e.g., creating links between them by specifying in the commits when or why a model is followed or updated, would be a major step forward with many benefits for developers and architects. We, however, do not see it happening in the near future, not because developers and architects do not think that would be a good idea (which in informal talks, they agree with), but because of the fact that their way of working is completely different and there are hardly meeting points.

Better tool-support for modeling and design in project repositories Versioning remains a challenge. So is lack of a proper standard for textual representation of UML models (XMI does not define enough to act as a standard). Thus, while coding is very tightly integrated in repositories/platforms for collaborative online software development, the lack of a good serialization standard for models prohibits the same type of support for modeling.

8. Threats to validity

Wohlin et al. discuss four main types of validity threats in empirical software engineering research: conclusion, internal, construct and external [22]:

Conclusion validity, being how sure we can be that the treatment we used in an investigation is related to the actual outcome we observed, does not affect our reflection.

Construct validity is the degree to which an investigation measures what it claims to be measuring. One of the sources of our analysis are the papers that cite the original papers where the dataset is described. This is an approximation of the impact of the dataset, which might be limited. It may well be that it has been used in other, non-academic contexts (i.e., in an industrial or educational context where a publication has not been produced). Researchers might have used the dataset, but may not have found it useful or conveniently organized for their purposes. Finally, the dataset might have been referenced in publications just with its URL and not with a citation; if so, we have not been able to identify it. The other source of analysis in the paper uses DBLP as a data source. While DBLP contains many publications, it is not comprehensive, and we might have not been able to find research collaborations that have taken place in practice. Nonetheless, we can assume that the ones in DBLP are the most relevant ones, so this mitigates to some extent this threat.

Internal validity is the extent to which a causal conclusion based on a study is warranted, which is determined by the degree to which a study minimizes systematic errors. We have tried to follow a systematic approach by downloading all papers and categorizing them according to the use they make of the dataset. Bias might have been introduced in this categorization, as the purpose of the use of the dataset might have been misinterpreted. To avoid this threat, the coding has been done in pairs by the authors in a first phase, and checked by a third author afterwards for consistency.

External validity is the degree to which results can be generalized to other contexts. This paper reflects on a very specific dataset of models in GitHub. Although the lessons learned might go beyond this scenario, and be valuable for researchers creating software engineering datasets, or for researchers interested in how innovation expands, we are aware that it is specific to its context and many of our observations may hold only there.

9. Conclusion

We think that, even though we had many failures on the way, the research we performed in 2016 mining for models in GitHub is a success story. The purpose of this reflection has been to learn from our failures for us and other research groups, in case they would like to perform a similar task.

In addition, our experience can be considered *unconventional*, as it is not grounded in a single of our own narrow research bubbles. We came from two very distant areas (software modeling vs. repository mining) that had almost no intersection at the time. As a result, we find that these efforts were not just suitable to reach researchers in both areas, but – even further – researchers in other areas (e.g., on AI-based image recognition). It is exciting to see that many of the researchers using our dataset come from outside of our own network.

Last but not least, we are aware that our reflections are very focused on our experience, but there are some general lessons learned that could be of value for software engineering scholars sharing datasets. On the other hand, given that many software engineering conferences and journals ask these days to share the artifacts used, our work can as well serve as an inspiration for those researchers who would like to offer it in a way that is most convenient to fellow researchers, as it has been found that doing so increases the impact of the work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The documented reproduction package, including data, spreadsheets and scripts, can be found at https://gsyc.urjc.es/grex/2023-ist-reflections-impact-model-mining.

Acknowledgments

The work of G. Robles has been supported in part by the Spanish Ministry of Science and Innovation (PID2022-139551NB-I0).

References

[1] Harald Störrle, Regina Hebig, Alexander Knapp, An index for software engineering models, in: PSRC@ MoDELS, 2014, pp. 36–40.

[2] B.H.A. Karasneh, An Online Corpus of UML Design Models: Construction and Empirical Studies (Ph.D. thesis), Leiden University, 2016.

[3] Regina Hebig, Truong Ho Quang, Michel R.V. Chaudron, Gregorio Robles, Miguel Angel Fernandez, 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, 2016, pp. 173–183.

[4] Gregorio Robles, Truong Ho-Quang, Regina Hebig, Michel R.V. Chaudron, Miguel Angel Fernandez, An extensive dataset of UML models in GitHub, in: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories, MSR, IEEE, 2017, pp. 519–522.

[5] Richard E. Boyatzis, Transforming Qualitative Information: Thematic Analysis and Code Development, sage, 1998.

[6] Roland Kretschmer, Djamel Eddine Khelladi, Roberto Erick Lopez-Herrejon, Alexander Egyed, Consistent change propagation within models, Softw. Syst. Model. 20 (2021) 539–555.

[7] Matheus de Oliveira Barbosa, Franklin Ramalho, An approach to identify and classify state machine changes from code changes, in: Proceedings of the 14th Brazilian Symposium on Software Components, Architectures, and Reuse, 2020, pp. 111–120.

[8] Djamel Eddine Khelladi, Roland Kretschmer, Alexander Egyed, Detecting and exploring side effects when repairing model inconsistencies, in: Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering, 2019, pp. 113–126.

[9] Fangwei Chen, Li Zhang, Xiaoli Lian, Cdtc: Automatically establishing the trace links between class diagrams in design phase and source code, Available at SSRN 4111234.

[10] Bilal Karasneh, Michel R.V. Chaudron, Online Img2UML repository: An online repository for UML models, in: EESSMod@ MoDELS, 2013, pp. 61–66.

[11] Sergei Shcherban, Peng Liang, Zengyang Li, Chen Yang, Multiclass classification of UML diagrams from images using deep learning, Int. J. Softw. Eng. Knowl. Eng. 31 (11n12) (2021) 1683–1698.

[12] Truong Ho-Quang, Michel RV Chaudron, Ingimar Samúelsson, Jóel Hjaltason, Bilal Karasneh, Hafeez Osman, Automatic classification of UML class diagrams from images, in: 2014 21st Asia-Pacific Software Engineering Conference, Vol. 1, IEEE, 2014, pp. 399–406.

[13] Mohd Hafeez Osman, Truong Ho-Quang, Michel R.V. Chaudron, An automated approach for classifying reverse-engineered and forward-engineered UML class diagrams, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, pp. 396–399.

[14] Graphs Dataset, SunEdition, 2010, URL: https://www.kaggle.com/sunedition/graphs-dataset.

[15] José Fernando Tavares, Yandre M.G. Costa, Thelma Elita Colanzi, Classification of UML diagrams to support software engineering education, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW, IEEE, 2021, pp. 102–107.

[16] José Antonio Hernández López, Javier Luis Cánovas Izquierdo, Jesús Sánchez Cuadrado, Using the ModelSet dataset to support machine learning in model-driven engineering, in: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, 2022, pp. 66–70.

[17] Daniel M. German, Bram Adams, Ahmed E. Hassan, Continuously mining distributed version control systems: an empirical study of how Linux uses Git, Empir. Softw. Eng. 21 (2016) 260–299.

[18] Nuthan Munaiah, Steven Kroh, Craig Cabrey, Meiyappan Nagappan, Curating github for engineered software projects, Empir. Softw. Eng. 22 (2017) 3219–3253.

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

[20] Christian F.J. Lange, Michel R.V. Chaudron, Johan Muskens, In practice: UML software architecture and design description, IEEE Softw. 23 (2) (2006) 40–46.

[21] Werner Heijstek, Michel R.V. Chaudron, Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process, in: 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2009, pp. 113–120.

[22] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, 2012.