[VNG+25] C. Verbruggen, L. Netz, P.- L. Glaser, M. Scholz, C. Huemer, M. Calamo, B. Rumpe, M. Snoeck, D. Bork:
Toward a Community-Curated Golden Dataset of UML Models.
In: 2025 ACM/IEEE 28th International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C),
pp. 43-50, DOI 10.1109/MODELS-C68889.2025.00012, IEEE, Oct. 2025.

Toward a Community-Curated Golden Dataset of UML Models

Charlotte Verbruggen[*], Lukas Netz[†], Philipp-Lorenz Glaser[*], Marion Scholz[*], Christian Huemer[*],
Marco Calamo[‡], Bernhard Rumpe[†], Monique Snoeck[§], Dominik Bork[*]

[*]*Business Informatics Group*, *TU Wien*, Vienna, Austria, Email: charlotte.verbruggen@tuwien.ac.at,
philipp-lorenz.glaser@tuwien.ac.at, marion.scholz@tuwien.ac.at, huemer@big.tuwien.ac.at, dominik.bork@tuwien.ac.at

[†]*Software Engineering*, *RWTH Aachen University*, Aachen, Germany, Email: netz@se-rwth.de, rumpe@se-rwth.de

[‡]*Dipartimento Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti,*
Sapienza Università di Roma, Rome, Italy, Email: calamo@diag.uniroma1.it

[§]*Research Center for Information System Engineering (LIRIS), KU Leuven*, Leuven, Belgium,
Email: monique.snoeck@kuleuven.be

***Abstract*—Datasets of Unified Modeling Language (UML) models are becoming increasingly valuable for education, empirical research, and tool development in model-driven engineering (MDE) and conceptual modeling. In recent years, several datasets have emerged – mostly compiled through automated crawling of open platforms such as GitHub and GenMyModel. While these efforts have improved access to real-world modeling artifacts, the resulting collections often suffer from serious quality issues: they include syntactically invalid models, semantically incorrect structures, and placeholder or dummy content. Moreover, most models are not accompanied by textual domain descriptions, which are essential for understanding the intent behind the model and assessing its semantic soundness. Therefore these model datasets are far from ideal as a source for modeling exercises or empirical MDE research. This paper presents an initial step toward a community-curated golden dataset of UML models, designed to address these limitations. Our contribution includes i) a curated set of UML models, each paired with a natural language description of the modeled domain requirements, ii) a publicly accessible web platform for exploring and querying the dataset, and iii) a structured process for community-based contribution and evaluation to support sustainable growth and quality assurance of the dataset. By fostering community involvement and providing high-quality, semantically grounded models, this work lays the foundation for a widely accepted benchmark dataset in UML-based research and education.**

***Index Terms*—Model repository, Dataset, UML, Open models, Curation, Community, Education, Machine learning**

I. Introduction

Teaching conceptual modeling or model-driven engineering (MDE) requires teaching students certain cognitive skills, such as abstraction. While part of a modeling course should focus on explaining the syntax and notation of the modeling language, students are often evaluated based on their model creation skills, rather than reproducing the notation of the modeling language or interpreting models [5]. As many modeling competencies are built by practice, the design of such courses requires a large number of exercises. While there are handbooks available that provide exercises (e.g., on BPM [15] or MDE [7]), teachers often need to curate their own set of modeling exercises to adequately support students to reach the learning goals of a specific course. Fostering community collaboration on the design of modeling exercises and teaching materials would strongly benefit the community in several ways. Educators would be able to reuse exercises developed by their colleagues, which could free up time for providing feedback to students. Additionally, the quality and quantity of exercises available to students would improve. Moreover, such a dataset would also enable self-paced learning.

In the past decades, several efforts have been made to create community-driven model repositories [13], [21], [33]. For example, the Open Model Initiative aimed to share reference models that could be reused across organizations [16]. Due to developments in conceptual modeling and AI research in recent years [6], [24], [31], there has also been increasing interest in research on machine learning for MDE (ML4MDE). This requires large enough datasets of models that new ML techniques can be applied to, amongst others, AI-supported model completion [1], [10], [42], model classification [22], [29], or model generation [3], [36].

To fill this gap, several model sets have been constructed in a (semi-)automated manner. However, these model sets have not been subjected to community-driven quality control. A recent study by Djelic et al. [14] revealed the questionable quality of automatically generated datasets, as they contain many dummy and duplicate models. Another barrier to using the existing model sets is that the models usually lack a textual description. Therefore, they are not suitable as an educational source for modeling exercises. Calamo et al. [9] provide an overview of existing datasets used for research on using LLMs to transform text to UML. These datasets are either very small (13 models or fewer) or do not contain the model solution for verification.

A recent community-driven effort to define a vision for the future of modeling also stressed the need for open and curated model repositories [25]. In this paper, we initiate such a community-curated dataset of UML models that ensures quality and maturity of the included models and allows their use in modeling education and ML4MDE scenarios.

979-8-3315-7990-6/25/$31.00 ©2025 IEEE
DOI 10.1109/MODELS-C68889.2025.00012

II. Related Work

A. Teaching Conceptual Modeling and MDE

Research on the topic of teaching conceptual modeling and model-driven engineering (MDE) addresses many different concerns, such as analyzing student behavior, developing automated assessment tools, and incorporating generative AI in course design.

Sedrakyan et al. [38] apply process mining to analyze the behavior of novice modelers. They conclude that different modeling behaviors (e.g., the frequency of testing the model via instantiation) result in different learning outcomes. Thiukhova et al. [41] investigate correlations between students' learning activities and their learning outcomes. They conclude that students attending in-person sessions where they have the opportunity to make exercises and receive real-time feedback, perform better in a conceptual modeling course.

Chen et al. [11] developed an approach for automated grading of models in textual domain-specific languages based on text embedding. Hamann et al. [18] addressed the issue of interoperability between model formats and automated assessment tools by proposing a platform-independent tool called "Assisted Assessment". Sedrakyan et al. [39] developed a code generation tool that allows students to simulate their models with incorporated feedback. This led to significant improvements in the students' learning outcomes.

Ardimento et al. [2] incorporate AI in MDE education by developing a scaffolding approach for providing feedback to students.

B. Publicly Available Teaching Materials

Recent research has focused on supporting MDE educators with the appropriate tools. Kienzle et al. [19] identify key requirements for modeling tools in educational contexts, emphasizing support for standard notations, ease of use in the classroom, and minimal setup effort. While their focus is on tool capabilities, their findings highlight the importance of didactic usability–an aspect our model set directly addresses through lightweight, ready-to-use examples tailored for teaching.

Building on this, Zschaler et al. [43] present the MDENet Education Platform, a zero-install, web-based environment offering guided modeling activities. It restricts functionality to essential features, lowering the entry barrier while enabling smooth transitions to full IDEs. Bucchiarone et al. [8] provide an interactive platform that collects a large selection of open-source teaching materials in many domains. They support educators to use generative AI to develop lesson plans using these open-source materials. Our work complements such platforms by providing a structured and reusable model set that integrates well into these teaching-oriented environments.

There is a great deal of literature on teaching modeling and model-based software engineering. "Fundamentals of Business Process Management" by Dumas et al. [15] is widely used in BPM education and provides a structured introduction to the BPM lifecycle, with a strong emphasis on modeling using BPMN. While it includes examples and exercises, it lacks a dedicated, reusable model set explicitly designed to support teaching and learning of process modeling. The book "Model-Driven Software Engineering in Practice" by Brambilla, Cabot, and Wimmer [7] and "Agile Modeling with UML: Code Generation, Testing, Refactoring" by Rumpe [34] also offer a practical introduction to the concepts and techniques of MDE. They emphasize the use of models throughout the software engineering process, highlighting real-world tools and industrial applications. Similar to [15], they do not provide a focused or structured model set for teaching modeling itself.

C. Model Sets

The notion to collect and publish a model set has been discussed and realized before. There have been several approaches. Frank and Strecker [16] propose the concept of Open Reference Models, arguing that the development and reuse of reference models for enterprise architecture can benefit significantly from community-driven, open-source-inspired collaboration. They explore how openness enhances model quality, fosters broader adoption, and outline strategic approaches to initiating sustainable, shared modeling initiatives, paralleling open-source software practices.

ModelSet [21] is a publicly available model set for UML and Ecore models that were scraped from GitHub and GenMyModel. In contrast, the OntoUML dataset [4], [37] is a FAIR-compliant curated collection of community-created conceptual models based on OntoUML and UFO. It includes diverse, machine-readable models with rich metadata, hosted on GitHub and accessible via a FAIR Data Point. The dataset supports research and tool development, with libraries enabling easy access and analysis.

Glaser et al. [17] present an EA ModelSet, a curated, FAIR-compliant dataset of 977 ArchiMate-based enterprise architecture (EA) models designed to support research in model quality, reasoning, and AI-assisted model engineering. Noting the historic scarcity of openly available, high-quality EA model collections, the authors aggregate models from diverse sources, develop labeling and annotation tools, and provide web, CLI, and Python interfaces to facilitate empirical and machine-learning-based analysis. Their work aims to lower the barrier for data-driven research into EA modeling practices.

III. The Golden Dataset of UML Models

The long-term goal of this project is to expand the initial dataset into a truly community-curated dataset. To encourage educators and researchers to use and contribute to the dataset, we aim to host the dataset on an inclusive, easily accessible and user-friendly platform. Similarly, contributing to the dataset should require minimal effort. Therefore, the primary platform for the dataset is a dedicated website that is designed to allow users to intuitively inspect the dataset. The underlying GitHub repository is publicly accessible so that contributors can propose updates directly. There are several options for researchers who want to access the dataset: the live version can be downloaded from the website[1] or accessed via GitHub[2] and snapshots can be downloaded from Zenodo[3] or Hugging Face[4]. Contributors can submit new cases via a form on the website[5] or GitHub, based on their preference.

A. Dataset Structure

The golden UML dataset contains UML models where (at least) the description and the UML Class Diagram are given. Each case is stored in a folder with three files. The `metadata.txt` file contains basic information about the case, such as its name, domain, language, and source. Optionally, the case can be labeled with tags. The `description.md` file contains the case description, and the `plantuml.txt` file contains the UML class diagram in PlantUML format. Besides these mandatory files, for some cases, there might be additional material available, such as alternative formats for the UML model, an explanation of how the model is constructed based on the description, alternative or incorrect models (with explanations), etc. These files are stored in an additional folder called `extramaterial`.

The initial dataset consists of 45 cases from five different sources. It contains (exam) exercises from RWTH Aachen (7), exercises from the book "Principles of Database Management" by Lemahieu, Baesens, and vanden Broucke [20] (6) , course exercises from McGill University (10), and exercises from courses at KU Leuven (12) and TU Wien (10). The selected cases have been used (repeatedly) in courses and have been improved based on student feedback. We also consider the model element types that are present in each exercise. Almost all models have attributes (43/45). Many models also contain inheritance (28/45) and extra material (20/45), followed by association classes (17/45) and enumeration (17/45). Other properties occur less frequently: composition (15/45), abstract classes (15/45), methods (8/45), and aggregation (5/45).

The cases in our dataset have also been classified into domains. To label the cases, we adopted the list of domains developed by Nikoo et al. [30]. They curated a list of domains for classifying BPMN models. The list of domains and their definitions is presented in Table I, as well as the number of models from the dataset for each domain. The labeling was done by two authors of this paper independently, based on the textual descriptions of the models, with an agreement percentage of 88% (40/45). The cases differing domain labels were discussed until an agreement was reached, and domain definitions were updated to reduce the ambiguity. A final review was performed by the other authors.

B. Submitting new Cases

We encourage contributions from the modeling community to continuously extend the dataset. To facilitate this, we provide two options for submitting new cases. The first option is a submission form on the website where contributors can enter the details of their case and upload the model file. Optionally, contributors can also add extra material and provide their contact details. In case the submitted case has already been published elsewhere, a file containing the explicit permission from the author(s) and publisher should be included in the submission, as well as a citation that can be used on the website. The second option is submitting a pull request to the GitHub repository. We provide a template to ensure all the required files are submitted.

C. Reviewing Protocol

The submitted cases are reviewed before they are processed and published as part of the dataset. The reviewing protocol consists of three main steps. Each submitted case is reviewed by at least one reviewer. The authors of this paper will take the responsibility of acting as reviewer or finding a suitable reviewer.

1) Initial manual completeness check: First, the reviewer will do a quick manual check to verify if the submission is complete and the submitted model file uses the correct format and naming convention. In case there are any technical issues with the submission, they will alert the authors and request an updated submission. The reviewer will also check if the case is already listed in the dataset. While similar cases are allowed, exact duplicates will be rejected. For example, the dataset could contain two "Library" cases from different sources that each contain different requirements for a library management system. The similarity of a submitted case will be judged based on the submitted description and model, not on metadata (e.g., name and source of the case).

2) Manual Quality Check: The main task of the reviewer is to perform a manual quality check. The reviewer will check the quality of the metadata, the description, the syntactic and semantic quality of the model, and the quality of any extra material that is provided. In case the reviewer finds significant issues with the metadata, description, model, or extra material, they will contact the author of the case to ask for clarifications or corrections before approving the model for publication.

Since we aim to involve the community in the review of the cases, we provide a review checklist to ensure consistency in the quality control of the submitted models. The initial checklist is listed in Table II. As the dataset evolves, the reviewing checklist will be revised as needed. In future work, we also aim to automate part of the reviewing process with automatic quality and formatting checks.

Cases submitted directly on GitHub additionally need to be validated to ensure that the metadata and file names conform to the dataset structure as explained in Section III-D.

D. Model Processing and Publishing

Once the case is approved by the reviewer, it can be processed and included in the next release of the dataset. An overview of this process is visualized in Fig. 1.

[1] https://golden-uml-modelset.vercel.app/

[2] https://github.com/Charlotte-Verbruggen/GoldenUMLmodelset

[3] https://zenodo.org/records/16985872

[4] https://huggingface.co/datasets/CharlotteVerbruggen/GoldenUMLmodelset

[5] https://forms.gle/3R2rMeNvq7Jqko8e7

TABLE I
DOMAIN CLASSIFICATION OF THE MODELS. DOMAINS WHERE THE DEFINITION HAS BEEN ADJUSTED ARE INDICATED WITH '*', AND NEW DOMAINS ARE INDICATED WITH '**'.

Domain	Definition	count
Social Networks**	Any model related to social media and social networks.	1
Business Services*	Any model related to human resources management, building management, tender management, maintenance and repair services, and translation and interpretation services.	1
Sales*	Any model related to sales activities, ticketing and shopping.	7
Insurance	Any model related to health, travel, and property insurance services.	1
Logistics*	Any model related to logistics operations such as scheduling and warehousing.	6
Healthcare	Any model related to healthcare systems and the provision of healthcare services.	4
Education	Any model related to educational systems and education related activities.	4
Research	Any model related to research activities.	2
Agriculture	Any model related to agricultural activities.	0
Leisure and Recreation*	Any model related to taking a vacation, planning trips, dining at restaurants, and food delivery, as well as hobbies and exercise.	6
Media and Publishing*	Any model related to publishing articles and books, as well as activities related to photography, game development, etc.	6
Financial Services	Any model related to banking services, accounting, credit agencies, and asset management.	1
Government Services	Any model related to government services like emergency management, such as firefighting and police services, as well as models related to visas.	0
Environment	Any model related to environmental services, including weather forecasting.	0
Manufacturing*	Any model related to product lines and the manufacturing of products, as well as construction of buildings.	4
Personal Activities	Any model related to planned activities undertaken by an individual.	2

TABLE II
REVIEWING CHECKLIST

Initial Manual Completeness Check

- Does the submission have all the required files (description, metadata, and model)?
- Are the submitted files in the correct format and naming convention?
- Is there already a similar case in the dataset? If so, is this submission sufficiently different?

Manual Quality Check

Metadata Quality

- Is the listed language correct? (check in description and model)
- Is the listed domain accurate? (check in description)
- If any tags are included, are they representative?
- If a citation is provided, check the accessibility.

Description Quality

- Is the description clear and well written?

Syntactic Quality of the Model

- Is the UML class diagram syntax correctly applied?

Semantic Quality of the Model

- Does the model accurately represent the given case description?

Quality of the Extra Material (if applicable)

- Is the submitted extra material relevant for the case and useful for end users of the dataset?
- Is the context and purpose of the submitted extra material clear?
- If the extra material contains alternative model visualizations, are they the same as the plantUML file?
- If the extra material contains alternative solutions, check the syntactic and semantic quality.
- If the extra material contains incorrect solutions or feedback, are the explanations clear?

1) Processing of cases: The cases submitted by form need to be processed before they can be added to the GitHub repository. For each submitted and approved case, a folder with a unique name is created in the dataset. This folder contains at least the following three files:

- The information entered in the submission form is stored in a file called `metadata.txt`. This file contains the name of the case, the language, the domain, and optionally, the user-entered tags, the source of the case, and contact information of the author. In the second step, the reviewers execute some transformations: the metadata entered in the form is stored as a `metadata.txt` file.
- The submitted case description is stored as a `description.md` file.
- The PlantUML model is rendered as an image using the PlantUML parser and stored in three formats (the original `.txt` file, a `.png` file, and a `.svg` file).

If additional material was submitted, this is stored in a subfolder called `extramaterial`.

2) Parsing: For each new release of the dataset, the files from the dataset are parsed to facilitate the visualization of the website. A `models.json` file is generated that contains all information about the cases, with the structure as shown in Fig. 2. For each case, the `models.json` file contains the provided metadata, the provided description, and the following properties that are parsed from the `plantuml.txt` file:

- *classCount*: the number of classes
- *associationCount*: the number of associations
- *hasExtraMaterial*: TRUE if the submission contains extra material
- *hasComposition*: TRUE if the model contains composition
- *hasAggregation*: TRUE if the model contains aggregation
- *hasInheritance*: TRUE if the model contains inheritance
- *hasAbstract*: TRUE if the model contains an abstract class
- *hasEnumeration*: TRUE if the model contains enumeration
- *hasAttribute*: TRUE if the model contains an attribute

- *hasMethod*: TRUE if the model contains a method
- *hasAssociationClass*: TRUE if the model contains an association class

3) Publishing: When new models are added to the dataset, they are immediately available on the dedicated website[1]. A snapshot of the dataset is also available on Zenodo[3] and Hugging Face[4]. New releases of the model dataset will be published as the dataset grows. Older releases will remain available for the sake of traceability. When the dataset is used in research projects, users will be able to refer to the exact version of the dataset they worked on via Zenodo[3] or Hugging Face[4].

The website contains three types of pages:

- The homepage provides general information about the dataset, and a button to download the latest version of the entire dataset as a .zip archive.
- The Search page (Fig. 3) allows the user to search through the dataset using keywords or filters (on model properties, language, domain, tags, or source). Cases can also be sorted on their name, language, source, number of classes or number of associations.
- For each case, there is a dedicated page with all the metadata, the description, and a visualization of the UML class diagram (an example is given in Fig. 4). If provided, the source, original citation, and/or contact info of the contributor are given. This page also contains a "Download" button that will download all the files of this case as a .zip archive.

The website provides a link to the submission form for new cases via the button "Submit", and to the GitHub repository.

E. Long-term Support, Traceability, and Maintainability

Since the dataset is open source and we aim to further extend it in the future via community contributions, it is important to ensure long-term support, traceability, and maintainability of the dataset.

1) Long-term Support: The team of authors is committed to providing long-term support for the dataset by taking on the reviewing, processing, and publishing responsibilities. As the quality of the overall dataset is driven by knowledge sharing by and for the modeling community, interested community members are welcomed to contribute to the reviewing process.

2) Long-term Traceability: The website shows the most recent state of the Golden UML Dataset. However, to ensure transparency and traceability, versions of the dataset will be made available for downloading on Zenodo and Hugging Face, which enables referencing a dedicated version.

3) Long-term Maintainability: The submission process is currently automated by the submission form, while reviewing, processing, and publishing are mostly manual processes. To ensure the long-term maintainability of the dataset, we aim to provide automation to support these processes, such as automated quality checks and file transformations.

IV. Applications

A. Use of the Golden UML Dataset in Education

Educators searching for exercises can use the dataset to use the cases as-is or to derive their own exercises from it. Afterwards, they can upload the new exercises as variants of existing cases. Given the (public) availability of the solutions, teachers won't be able to use the cases "as is" for summative evaluations. With LLMs developing into increasingly competent assistants for many tasks, one of the key future skills of modelers is evaluating the accuracy of a given solution against requirements [40]. Having multiple variants of the same cases could turn the dataset into a useful source of material to train these skills: having multiple variants of a case facilitates creating exercises on requirements and solution analysis, and evaluating the justification of model variants. Educators can also point their students to the dataset as a source of extra learning material.

B. Use of the Golden UML Dataset in ML/AI

Generative AI is increasingly applied to modeling tasks, where it is used to automatically generate models from textual descriptions, structured data, or other inputs [27], [28]. These AI systems aim to support or replace manual modeling activities by producing models faster and potentially at a lower cost. This opens new possibilities for model-driven development and analysis, but also introduces challenges in verifying the correctness and quality of the generated models.

Generative AI will almost always produce an output for any modeling task [23]. Consequently, research questions shift from verifying the feasibility of generating models to assessing the quality and accuracy of the generated results. This makes it essential to evaluate AI-based modeling approaches.

Such evaluations require a point of reference. These references should be curated, high-quality datasets [32], enabling comparisons to ideal solutions. A mediocre dataset would only verify whether an approach can produce a model at all–a task that could be performed comparably well by a simple parser and does not require a benchmarking set. The goal is not merely to check whether an AI system can produce a model, but whether it can produce a meaningful, high-quality model.

A dataset that maps each model to its corresponding task is ideal for benchmarking semantic differences [26], [35] between AI-generated models and reference solutions. Such a dataset serves as a standardized benchmark set for evaluating AI-based modeling approaches.

V. Future Work and Call To Action

In this paper, we presented an initial dataset of curated UML class diagrams that are enhanced with textual descriptions and, in some cases, with additional material. Educators who want to find a certain type of exercise in the dataset can use the filters provided on the website. However, our aim is to continue improving the dataset and developing it into an interactive hub for modeling educators. To achieve this goal, several steps can be taken.

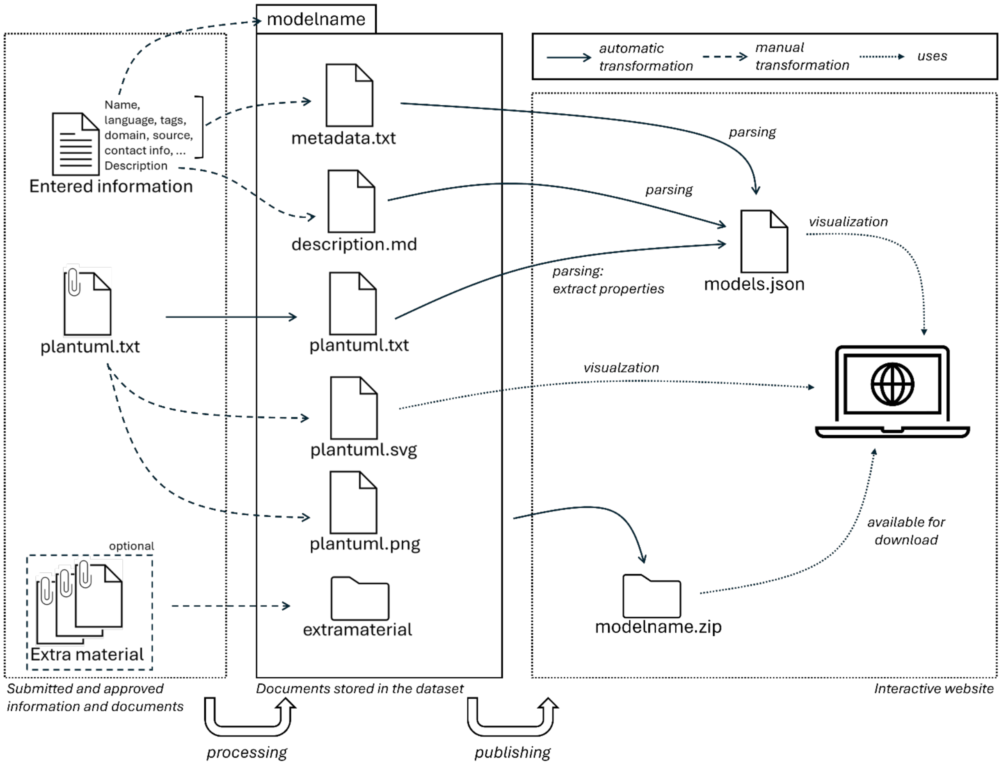
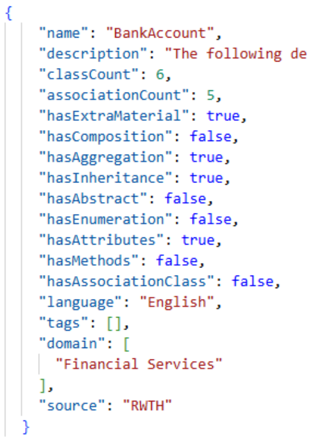


Fig. 1. Overview of the processing of the submitted files



```
{
    "name": "BankAccount",
    "description": "The following de
    "classCount": 6,
    "associationCount": 5,
    "hasExtraMaterial": true,
    "hasComposition": false,
    "hasAggregation": true,
    "hasInheritance": true,
    "hasAbstract": false,
    "hasEnumeration": false,
    "hasAttributes": true,
    "hasMethods": false,
    "hasAssociationClass": false,
    "language": "English",
    "tags": [],
    "domain": [
      "Financial Services"
    ],
    "source": "RWTH"
  }
```

Fig. 2. Example of a case as represented in the models.json file

First and foremost, we are launching a call to action, encouraging the modeling community to contribute their UML modeling cases to further extend the dataset.

Additionally, we aim to add components to the current website. For example, cases with “extra material” are currently labeled. However, no further information is given; the extra material can simply be downloaded with the case. This “extra material” can range from alternative file formats and visualizations (without additional content) to detailed feedback, incorrect solutions (with explanations), and model interpretation/evaluation questions. We plan to make the type of extra material more visible on the website.

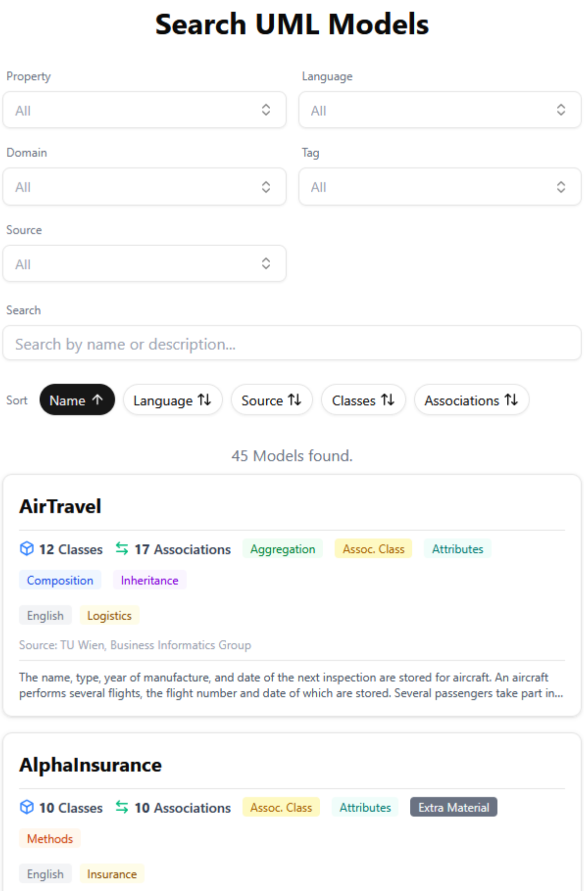
The dataset can also be extended with different types of cases. This can range from different modeling languages (e.g., ER models) to different types of exercises (e.g., multiple choice questions about model interpretations). Another potential extension is the addition of learning goals for each case.

Eventually, we aim to establish a traceability functionality that interactively links parts of the exercise solution with the corresponding natural language text passage in the requirements specification. This traceability can also be used to link comprehensibility or multiple-choice questions to the corresponding parts in the model that would serve as a response.

To evaluate the usability of the website and the additional needs of the community, our goal is to conduct a usability study with educators.

VI. Conclusion

The growing reliance on UML model datasets for education and research in model-driven engineering underscores the need for high-quality, semantically meaningful modeling artifacts. Existing datasets, while valuable, often fall short in terms of quality and contextual understanding. This paper addresses these shortcomings by introducing a curated dataset of UML



Search UML Models

Property: All

Language: All

Domain: All

Tag: All

Source: All

Search: Search by name or description...

Sort: Name ↑ | Language ⇅ | Source ⇅ | Classes ⇅ | Associations ⇅

45 Models found.

AirTravel

12 Classes 17 Associations Aggregation Assoc. Class Attributes Composition Inheritance

English Logistics

Source: TU Wien, Business Informatics Group

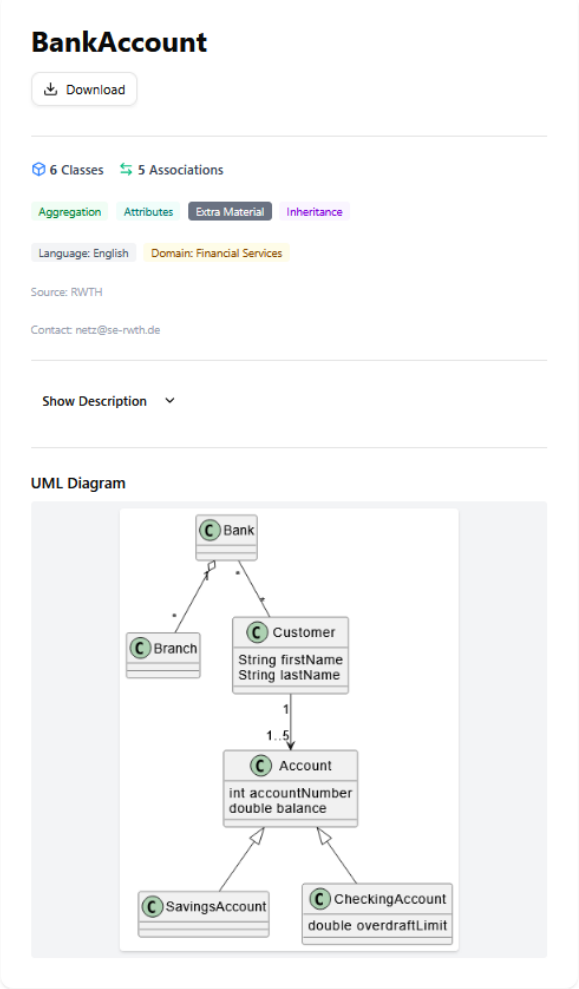
The name, type, year of manufacture, and date of the next inspection are stored for aircraft. An aircraft performs several flights, the flight number and date of which are stored. Several passengers take part in...

AlphaInsurance

10 Classes 10 Associations Assoc. Class Attributes Extra Material Methods

English Insurance

Fig. 3. The search page for the Golden UML ModelSet website



BankAccount

Download

6 Classes 5 Associations

Aggregation Attributes Extra Material Inheritance

Language: English Domain: Financial Services

Source: RWTH

Contact: netz@se-rwth.de

Show Description

UML Diagram

Fig. 4. The page for a single case in the Golden UML ModelSet website, in this case BankAccount

models, each paired with a natural language domain description, and made openly accessible through an interactive web platform. To ensure ongoing quality and relevance, we propose a community-driven contribution and evaluation process that encourages broad participation and sustainable curation. By laying the groundwork for a golden dataset of UML models, this initiative aims to foster a shared resource for the modeling community that supports reproducibility, benchmarking, and pedagogical excellence. The dataset can be accessed via: https://golden-uml-modelset.vercel.app/

Acknowledgment

We thank Wilfried Lemahieu, Bart Baesens, Seppe vanden Broucke and Cambridge University Press for allowing the use of the UML modeling exercises from their book [20]. We thank Kua Chen, Yujing Yang, Boqi Chen, José Antonio Hernández López, Gunter Mussbacher and Dániel Varró for allowing the use of their cases published in [12].

References

[1] Ali, S.J., Bork, D.: A graph language modeling framework for the ontological enrichment of conceptual models. In: International Conference on Advanced Information Systems Engineering. pp. 107–123. Springer (2024)

[2] Ardimento, P., Bernardi, M.L., Cimitile, M., Scalera, M.: Enhancing software modeling learning with ai-powered scaffolding. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. p. 103–106. MODELS Companion '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3687776

[3] Arora, C., Sabetzadeh, M., Nejati, S., Briand, L.: An active learning approach for improving the accuracy of automated domain model extraction. ACM Transactions on Software Engineering and Methodology (TOSEM) **28**(1), 1–34 (2019)

[4] Barcelos, P.P.F., Sales, T.P., Fumagalli, M., Fonseca, C.M., Sousa, I.V., Romanenko, E., Kritz, J., Guizzardi, G.: A fair model catalog for ontology-driven conceptual modeling research. In: International Conference on Conceptual Modeling. pp. 3–17. Springer (2022)

[5] Bogdanova, D., Snoeck, M.: Domain modelling in bloom: Deciphering how we teach it. In: IFIP working conference on the practice of enterprise modeling. pp. 3–17. Springer (2017)

[6] Bork, D., Ali, S.J., Roelens, B.: Conceptual modeling and artificial intelligence: A systematic mapping study. CoRR **abs/2303.06758** (2023). https://doi.org/10.48550/ARXIV.2303.06758

[7] Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice, Second Edition. Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers (2017). https://doi.org/10.2200/S00751ED2V01Y201701SWE004

[8] Bucchiarone, A., Cicchetti, A., Vázquez-Ingelmo, A., Adami, F., Schiavo, G., García-Holgado, A., García-Peñalvo, F.J.: Designing and generating lesson plans combining open educational content and generative ai. In: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. p. 78–86 (2024)

[9] Calamo, M., Mecella, M., Snoeck, M.: Assessing the suitability of large language models in generating uml class diagrams as conceptual models. In: Int. Conf. on Business Process Modeling, Development and Support, Int. Conf. on Evaluation and Modeling Methods for Systems Analysis and Development. pp. 211–226. Springer (2025)

[10] Chaaben, M.B., Burgueño, L., David, I., Sahraoui, H.A.: On the utility of domain modeling assistance with large language models. CoRR **abs/2410.12577** (2024). https://doi.org/10.48550/ARXIV.2410.12577

[11] Chen, K., Chen, B., Yang, Y., Mussbacher, G., Varró, D.: Embedding-based automated assessment of domain models. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. p. 87–94. MODELS Companion '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3687774

[12] Chen, K., Yang, Y., Chen, B., Hernández López, J.A., Mussbacher, G., Varró, D.: Automated domain modeling with large language models: A comparative study. In: 2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS). p. 162–172. IEEE Press (2023). https://doi.org/10.1109/MODELS58315.2023.00037, https://doi.org/10.1109/MODELS58315.2023.00037

[13] Di Rocco, J., Di Ruscio, D., Iovino, L., Pierantonio, A.: Collaborative repositories in model-driven engineering [software technology]. IEEE Software **32**(3), 28–34 (2015)

[14] Djelic, Ali, S.J., Verbruggen, C., Neidhardt, J., Bork, D.: A model cleansing pipeline for model-driven engineering: Mitigating the garbage in, garbage out problem for open model repositories. In: 2025 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS) (To Appear), https://model-engineering.info/publications/papers/MODELS25-ModelCleansing.pdf

[15] Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018). https://doi.org/10.1007/978-3-662-56509-4

[16] Frank, U., Strecker, S.: Open reference models - community-driven collaboration to promote development and dissemination of reference models. Enterprise Modelling and Information Systems Architectures (EMISAJ) **2**(2), 32–41 (2007)

[17] Glaser, P.L., Sallinger, E., Bork, D.: The extended ea modelset—a fair dataset for researching and reasoning enterprise architecture modeling practices. Software and Systems Modeling (2025). https://doi.org/10.1007/s10270-025-01278-1

[18] Hamann, M., Götz, S., Aßmann, U.: Towards an interoperable model-driven automated assessment system for computer science education. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. p. 95–102. MODELS Companion '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3687775

[19] Kienzle, J., Zschaler, S., Barnett, W., Sağlam, T., Bucchiarone, A., Abrahão, S., Syriani, E., Kolovos, D., Lethbridge, T., Mustafiz, S., et al.: Requirements for modelling tools for teaching. Software and Systems Modeling **23**(5), 1055–1073 (2024)

[20] Lemahieu, W., vanden Broucke, S., Baesens, B.: Principles of Database Management: The Practical Guide to Storing, Managing and Analyzing Big and Small Data. Cambridge University Press (2018)

[21] López, J.A.H., Cánovas Izquierdo, J.L., Cuadrado, J.S.: Modelset: a dataset for machine learning in model-driven engineering. Software and Systems Modeling pp. 1–20 (2022)

[22] López, J.A.H., Cuadrado, J.S., Rubei, R., Di Ruscio, D.: Modelxglue: a benchmarking framework for ml tools in mde. Software and Systems Modeling pp. 1–24 (2024)

[23] Mancoridis, M., Weeks, B., Vafa, K., Mullainathan, S.: Potemkin understanding in large language models. arXiv preprint arXiv:2506.21521 (2025)

[24] Marcén, A.C., Iglesias, A., Lapeña, R., Pérez, F., Cetina, C.: A systematic literature review of model-driven engineering using machine learning. IEEE Trans. Software Eng. **50**(9), 2269–2293 (2024). https://doi.org/10.1109/TSE.2024.3430514

[25] Michael, J., Bork, D., Wimmer, M., Mayr, H.C.: Quo Vadis Modeling? Findings of a Community Survey, an Ad-hoc Bibliometric Analysis, and Expert Interviews on Data, Process, and Software Modeling. Journal Software and Systems Modeling (SoSyM) **23**(1), 7–28 (February 2024). https://doi.org/10.1007/s10270-023-01128-y

[26] Nachmann, I., Rumpe, B., Stachon, M., Stüber, S.: Open-World Loose Semantics of Class Diagrams as Basis for Semantic Differences. In: Modellierung 2022. pp. 111–127. Gesellschaft für Informatik e.V. (June 2022). https://doi.org/10.18420/modellierung2022-016

[27] Netz, L., Michael, J., Rumpe, B.: From Natural Language to Web Applications: Using Large Language Models for Model-Driven Software Engineering. In: Weske, M., Michael, J. (eds.) Modellierung 2024. pp. 179–195. LNI, GI (March 2024). https://doi.org/10.18420/modellierung2024_018

[28] Netz, L., Reimer, J., Rumpe, B.: Using Grammar Masking to Ensure Syntactic Validity in LLM-based Modeling Tasks. In: Workshop on Artificial Intelligence and Model-driven Engineering (MDE Intelligence), MODELS Companion '24. pp. 570–577. ACM (October 2024). https://doi.org/10.1145/3652620.3687805

[29] Nguyen, P.T., Di Ruscio, D., Pierantonio, A., Di Rocco, J., Iovino, L.: Convolutional neural networks for enhanced classification mechanisms of metamodels. Journal of Systems and Software **172**, 110860 (2021)

[30] Nikoo, M.S., Kochanthara, S., Babur, Ö., van den Brand, M.: An empirical study of business process models and model clones on github. Empirical Software Engineering **30**(2) (Mar 2025). https://doi.org/10.1007/s10664-024-10584-z

[31] Rädler, S., Berardinelli, L., Winter, K., Rahimi, A., Rinderle-Ma, S.: Bridging MDE and AI: a systematic review of domain-specific languages and model-driven practices in AI software systems engineering. Softw. Syst. Model. **24**(2), 445–469 (2025). https://doi.org/10.1007/S10270-024-01211-Y

[32] Rawat, D.B.: Secure and trustworthy machine learning/artificial intelligence for multi-domain operations. In: Pham, T., Solomon, L. (eds.) Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III. vol. 11746, p. 1174609. International Society for Optics and Photonics, SPIE (2021). https://doi.org/10.1117/12.2592860

[33] Robles, G., Ho-Quang, T., Hebig, R., Chaudron, M.R., Fernandez, M.A.: An extensive dataset of uml models in github. In: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR). pp. 519–522. IEEE (2017)

[34] Rumpe, B.: Agile Modeling with UML: Code Generation, Testing, Refactoring. Springer International (May 2017)

[35] Rumpe, B., Stachon, M., Stüber, S., Voufo, V.: Semantic Difference Analysis with Invariant Tracing for Class Diagrams Extended by OCL. In: Workshop on Model Driven Engineering, Verification and Validation (MoDeVVa), MODELS Companion '24. pp. 1066–1075. ACM (2024)

[36] Saini, R., Mussbacher, G., Guo, J.L., Kienzle, J.: Domobot: A modelling bot for automated and traceable domain modelling. In: 2021 IEEE 29th International Requirements Engineering Conference (RE). pp. 428–429. IEEE (2021)

[37] Sales, T.P., Barcelos, P.P.F., Fonseca, C.M., Souza, I.V., Romanenko, E., Bernabé, C.H., Bonino da Silva Santos, L.O., Fumagalli, M., Kritz, J., Almeida, J.P.A., Guizzardi, G.: A fair catalog of ontology-driven conceptual models. Data & Knowledge Engineering **147**, 102210 (2023). https://doi.org/10.1016/j.datak.2023.102210

[38] Sedrakyan, G., De Weerdt, J., Snoeck, M.: Process-mining enabled feedback: "tell me what i did wrong" vs. "tell me how to do it right". Computers in Human Behavior **57**, 352–376 (2016). https://doi.org/10.1016/j.chb.2015.12.040

[39] Sedrakyan, G., Snoeck, M., Poelmans, S.: Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. Computers & Education **78**, 367–382 (2014). https://doi.org/10.1016/j.compedu.2014.06.014

[40] Snoeck, M., Pastor, O.: Teaching conceptual modelling in the age of LLMs: shifting from model creation to model evaluation skills. Software and Systems Modeling (2025). https://doi.org/10.1007/s10270-025-01307-z

[41] Tiukhova, E., Verbruggen, C., Baesens, B., Snoeck, M.: Learning analytics tells: Know your basics and go to class. In: CEUR Workshop Proceedings. vol. 3618. CEUR Workshop Proceedings (2024)

[42] Weyssow, M., Sahraoui, H., Syriani, E.: Recommending metamodel concepts during modeling activities with pre-trained language models. Software and Systems Modeling **21**(3), 1071–1089 (2022)

[43] Zschaler, S., Barnett, W., Boronat, A., Garcia-Dominguez, A., Kolovos, D.: The mdenet education platform: Zero-install directed activities for learning mde. Software and Systems Modeling (2025)