EA ModelSet – A FAIR Dataset for Machine Learning in Enterprise Modeling

Philipp-Lorenz Glaser, Emanuel Sallinger and Dominik Bork

To appear in:

16th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2023)

Final version available soon:

www.model-engineering.info

EA ModelSet – A FAIR Dataset for Machine Learning in Enterprise Modeling

Philipp-Lorenz Glaser[1 [0000−0002−0710−8052]], Emanuel Sallinger[2 [0000−0001−7441−129X]], and Dominik Bork[1 [0000−0001−8259−2297]]

[1] Business Informatics Group, TU Wien, Vienna, Austria
{philipp-lorenz.glaser,dominik.bork}@tuwien.ac.at
[2] Database and Artificial Intelligence Group, TU Wien, Vienna, Austria
emanuel.sallinger@tuwien.ac.at

Abstract. The conceptual modeling community and its subdivisions of enterprise modeling are increasingly investigating the potentials of applying artificial intelligence, in particular machine learning (ML), to tasks like model creation, model analysis, and model processing. A prerequisite—and currently a limiting factor for the community—to conduct research involving ML is the scarcity of openly available models of adequate quality and quantity. With the paper at hand, we aim to tackle this limitation by introducing an EA ModelSet, i.e., a curated and FAIR repository of enterprise architecture models that can be used by the community. We report on our efforts in building this data set and elaborate on the possibilities of conducting ML-based modeling research with it. We hope this paper sparks a community effort toward the development of a FAIR, large model set that enables ML research with conceptual models.

Keywords: Enterprise modeling · Machine learning · FAIR · Enterprise architecture · Data set.

1 Introduction

In recent years, the field of conceptual modeling, particularly enterprise modeling, has seen an increasing interest in exploring the promising applications of artificial intelligence, specifically machine learning (ML), to various tasks such as model creation, analysis, processing, and transformation [6,2,20]. Leveraging ML has the potential to revolutionize the way enterprise modeling is approached and implemented. However, a significant challenge hindering progress in this domain is the scarcity of readily available data, specifically high-quality and diverse models in sufficient quantities.

The success of ML approaches heavily relies on large and diverse datasets that capture the intricacies and complexities of real-world scenarios. For the conceptual modeling community, access to an extensive repository of models is crucial to enable robust and data-driven research. Unfortunately, the lack of publicly available, free-to-access datasets has emerged as a major bottleneck in advancing ML research in this domain. Without access to a substantial collection of models, researchers face significant challenges in developing and evaluating ML algorithms, hindering progress and innovation.

To address the challenges mentioned at the outset, researchers recognize the importance of adhering to the principles of Findable, Accessible, Interoperable, and Reusable (F.A.I.R.) [23] data management. A FAIR dataset ensures that data is discoverable and accessible to all interested researchers, fostering collaboration and enabling the reproducibility of results. Additionally, a FAIR dataset is designed to be interoperable, facilitating seamless integration with various ML tools and techniques. Moreover, by making the dataset reusable, researchers can build upon existing work and accelerate the development of innovative solutions.

A FAIR dataset of enterprise architecture (EA) models is essential for several reasons. Firstly, it addresses the issue of data scarcity by collating a comprehensive collection of diverse and high-quality EA models from various domains and industries. Secondly, adhering to the principles of FAIR ensures that the dataset is openly available to the conceptual modeling community, breaking down barriers and encouraging active engagement and contribution from researchers worldwide. The introduction of a FAIR EA ModelSet unlocks a plethora of possibilities for ML-based research in the domain of conceptual modeling. Researchers can now leverage this curated repository to train and validate ML models, enabling automated tasks such as generating new EA models, analyzing complex relationships within models, processing large volumes of data efficiently, and transforming models to adapt to evolving business requirements.

Furthermore, the availability of a FAIR dataset fosters the growth of a collaborative and innovative research community dedicated to exploring the potential applications of ML in EA management. By providing a common foundation for experiments and evaluations, the FAIR EA ModelSet empowers researchers to benchmark their methods against existing approaches, driving continuous improvement and development in the field. Eventually, this research holds immense significance for the conceptual modeling community by not only addressing data scarcity but also paving the way for a more collaborative and dynamic research landscape. Through this research, we aim to inspire and encourage a collective effort toward the development of a comprehensive and freely available dataset, sparking new avenues of exploration and innovation at the intersection of artificial intelligence and conceptual modeling (for an overview, see [6]).

FAIR datasets have garnered significant attention in various research domains. In the field of **conceptual and enterprise modeling**, researchers have focused on creating FAIR datasets that encompass various domain-specific models, such as data models, ontology models [3], and domain models. These datasets aim to enhance the accessibility and reusability of conceptual models for research and practical applications and to enable insights into the actual use of modeling languages. Additionally, efforts have been made to standardize metadata annotation and representation to improve the findability and interoperability of the datasets [4,21]. In **software engineering and software modeling** research, the development of FAIR datasets has been crucial for advancing the state of software development, testing, and maintenance. Researchers have built datasets that comprise software architecture models [17], UML diagrams, and source code representations [12,13,14,18]. These datasets enable software engineers to leverage ML and data-driven techniques to automate and/or improve software development tasks. Within the **process modeling** community, there have been efforts to curate datasets containing various types of process models [22,8,19]. The sub-discipline of **process mining**

is also heavily engaged in the creation and use of publicly available datasets (see [9]). These datasets facilitate the empirical analysis of business process management and the evaluation and comparison of process mining algorithms and tools.

In this paper, we report our efforts of creating an open, curated repository of EA models following the FAIR principles. In total, we were able to collect, harmonize, integrate, and publicize a total of 863 ArchiMate models. Moreover, we contribute means of efficiently exploiting the EA ModelSet by providing a Webpage, a Java Command Line Interface, and a Python Jupyter Notebook.

In the remainder of this paper, we discuss the method we applied to collect, process, and manage the ModelSet in Section 2. Section 3 then introduces the characteristics of the EA ModelSet. An evaluation of the ModelSet according to the FAIR principles is presented in Section 4. A number of enabled usage scenarios by our EA ModelSet are discussed in Section 5 before we conclude this paper in Section 6.

2 Method for Creating the Model Set

Next, we describe the three stages of the method we followed while creating the EA ModelSet dataset.

2.1 Dataset Collection

The data collection process (see Fig. 1) revolves around retrieving and storing EA models from diverse data sources. These models serve as the raw data input for subsequent processing activities. In our process, we identified *GitHub* and *GenMyModel* as valuable data sources due to their extensive collections of ArchiMate models, which can also be retrieved with reasonable effort. GitHub, a popular platform for hosting and sharing code repositories, hosts numerous open-source projects and provides a Search API for searching code globally across all indexed repositories. Utilizing the provided search functionality, we formulated specific queries to retrieve ArchiMate models in different formats commonly used by the community.

We obtained models in: i) The Open Group Standard ArchiMate Model Exchange File Format - a standard XML format allowing for model exchange between tools[1], ii) Archi model storage format - used by the Archi modeling tool[2], and iii) Git Friendly Archi File Collection (GRAFICO) format - mostly used by the model collaboration Archi plugin coArchi[3], which is also in XML format. These formats were queried by including the respective file extensions (i.e., `*.xml` and `*.archimate`).

The collection process is partly automated, by downloading the individual files from the search results through a browser script. At a later stage, we used the Python library PyGithub[4] to automatically retrieve models from GitHub, associate them with their respective repositories, and if present, link the corresponding license information. Models in GRAFICO format were transformed into format ii) using the Archi Command-Line

[1] `https://www.opengroup.org/xsd/archimate/`

[2] `https://www.archimatetool.com/`

[3] `https://github.com/archimatetool/archi-modelrepository-plugin`

[4] `https://github.com/PyGithub/PyGithub`

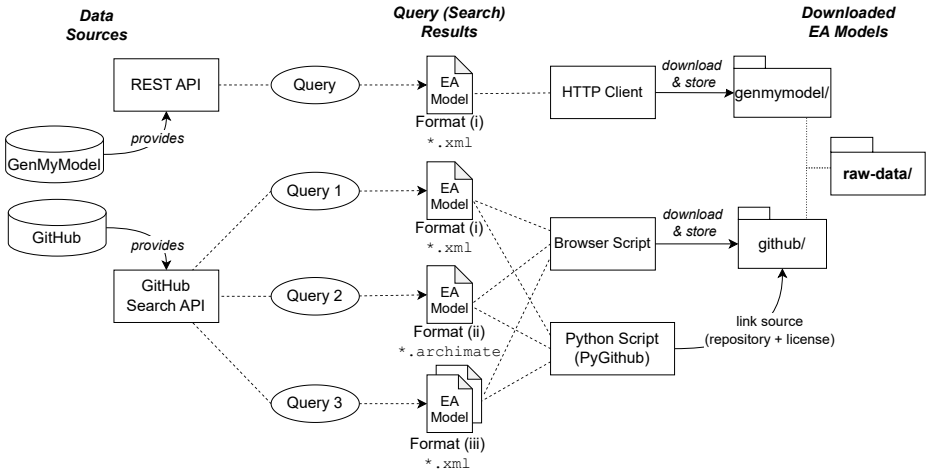


Fig. 1. Data collection workflow

Interface (CLI) tool to not introduce any additional complexity for later activities (e.g., not requiring an additional parser). In total, we collected 922 models from GitHub, stored in the `raw-data/github/` directory.

GenMyModel, an online modeling platform supporting a variety of modeling languages, serves as another data source. Through its REST API[5], we filtered for public ArchiMate projects and retrieved the models in the standard model exchange XML format (format i) from above). We collected 287 models from GenMyModel, stored in the `raw-data/genmymodel/` directory.

In addition to GitHub and GenMyModel, we manually collected models from other sources, including forums, publications, and project/company websites. These models were obtained through targeted web searches. We collected 15 models from other sources, stored in the `raw-data/other/` directory.

2.2 Dataset Processing

With a substantial collection of almost 1,000 ArchiMate models in different formats, the subsequent step in our method involves processing these models to transform them into a standardized format suitable for advanced analysis and ML tasks. The data processing phase is initiated by receiving the collected models from the `raw-data/` directory as input, with file duplicates discarded beforehand by comparing their MD5 file hashes. Each raw ArchiMate model is processed as follows:

Parsing: The file is parsed to extract relevant information and to create an intermediate `ParsedModel` representation. Since all our input files are either in format i) or ii), two separate XML parsers are used. Although the formats differ in their hierarchical structure and naming schemes, they contain the same information and, therefore, can be parsed into a unified representation (i.e., a `ParsedModel`). If any major errors occur

[5] `https://app.genmymodel.com/api/projects/public`

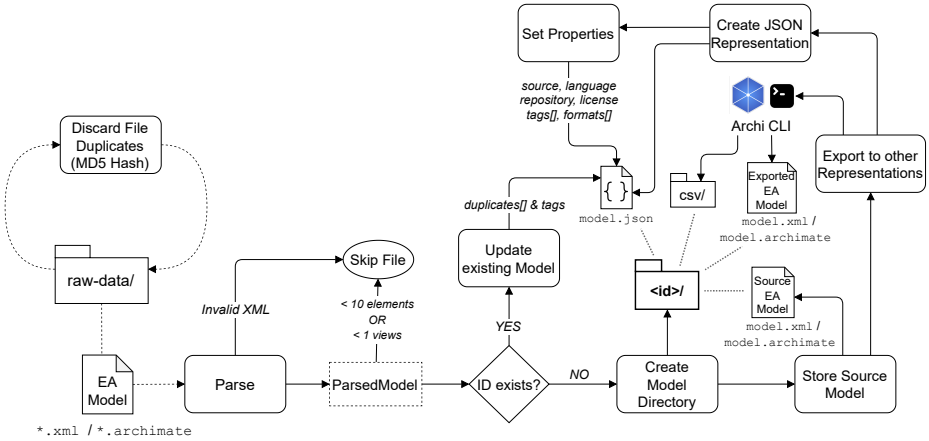


Fig. 2. Data processing workflow

during parsing or the number of elements in the parsed model is less than 10 (indicating a model with insufficient complexity), the file is skipped.

Duplicate Detection: The parsed model's ID is checked against existing IDs of models that have already been processed. If a duplicate is found, the existing JSON representation of the model is updated by adding the duplicate model's file path to the list of detected duplicates, and the model is tagged with a `DUPLICATE` label. The processing workflow then continues with the next file.

Directory Creation: For each unique model, a new directory is created with the ID as its name. This directory is used to store and locate the model in various formats.

Storage & Export of Formats: The source file from which the model was parsed from is stored first, either as `model.xml` or `model.archimate`. To improve interoperability, the model is additionally exported into the respective other ArchiMate model format (i.e., as `model.xml` or `model.archimate`) using the Archi CLI tool[6]. Elements, relations, and properties of the model are exported as separate CSV files (`elements.csv`, `relations.csv`, and `properties.csv`, respectively) within a directory named `csv/`.

JSON Representation: The last file that is created in the model's directory is a JSON representation of the model, named `model.json` and conforming to a defined JSON schema in `ea-model.schema.json`. The JSON representation includes additional properties to further classify certain characteristics of the model in the dataset, in addition to common ArchiMate model properties already present in the parsed source file (see Section 3.1 for more information regarding the JSON schema). For the first release of the dataset, we relied on simple mechanisms to set the properties: The `source` property is set to the path of the parsed source file, for warnings during the parsing process (e.g., a relationship could not be parsed due to invalid source/target ID) we added a `WARNING` label to the list of `tags`, a corresponding `repository` URL and `license` is linked, the list of `formats` is based on the successfully exported formats of the previous step, and at last we set the `language` property by merging the names of

[6] `https://github.com/archimatetool/archi/wiki/Archi-Command-Line-Interface`

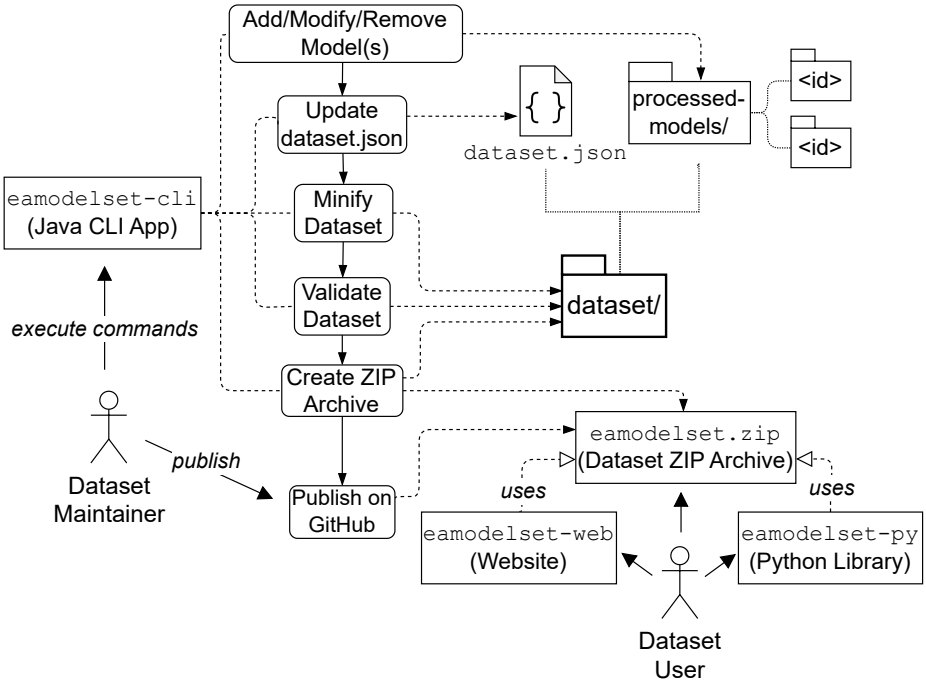


Fig. 3. Dataset management and publishing workflow

a model's elements into a single textual representation to serve as input for the language detection Java library Lingua[7] that provides us an estimate of a suitable language.

After data processing, a total of 863 unique models remained, which are stored in the `processed-models/` directory. Each model has its own subdirectory, denoted by the model's ID, and contains the different representations of the model created during the processing stage (i.e. JSON, XML, ARCHIMATE, CSV).

2.3 Dataset Management & Publishing

The final stage of our method focuses on managing and publishing the EA ModelSet dataset with its accompanying services. The dataset is stored within the `EAModelSet` GitHub repository[9] in a central directory called `dataset/`. This directory includes the `processed-models/` directory from the previous stage and a `dataset.json` file which adheres to the JSON schema specified in `ea-dataset.schema.json` (see Section 3.1) containing metadata and computed data about the dataset itself. It also includes brief information about each model and a subset of its characteristics, facilitating model search. The dataset.json is further used by the website for model search and the Python library for searching within the pandas dataframe.

Dataset management activities are primarily performed using the accompanying *Java CLI application*[9], enabling maintainers to add, modify, or remove models from

[7] `https://github.com/pemistahl/lingua`

the dataset. When preparing for a new release, the `dataset.json` file is updated to reflect the changes made to the dataset. Following the update, the processed models undergo a *minification* process to reduce their file size and optimize storage efficiency. Minification involves removing unnecessary white spaces, comments, and other non-essential elements from the model files, further improving their compactness. The dataset then undergoes a validation process to ensure its quality and consistency. Validation includes checking the JSON schema to ensure conformity and verifying the file structure and presence of all required files. Any models that do not adhere to the defined schema or have missing files are flagged for further manual investigation or correction, ensuring the overall integrity of the data.

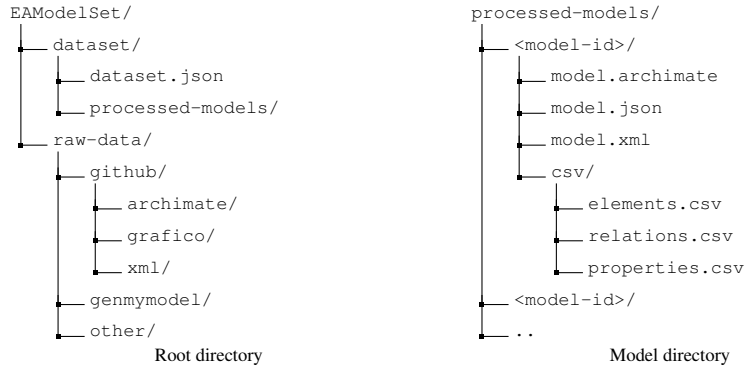
Once the dataset is prepared and validated, it is compressed into a single file archive, named `ea-modelset.zip`. Compression further reduces the overall file size, making it easier to distribute and download the dataset while preserving its content and structure. After following the described stages, the EA ModelSet dataset is effectively organized, summarized, validated, compressed, and ready to be made publicly accessible. It is published as a new GitHub release to ensure availability, version control, and visibility of the dataset to the wider community. The prepared ZIP archive is also utilized by the accompanying applications, such as the website and Python library (see Section 3.3), enabling consistent use across various services and interfaces.

3 EA ModelSet

We now introduce the curated and FAIR EA ModelSet—a dataset of ArchiMate models.

3.1 Dataset Structure & Schema

EA ModelSet follows a well-defined structure and leverages JSON schemas [15] to facilitate efficient data management and to provide a FAIR dataset of EA models. The relevant directories and files within the dataset are structured as follows:



The `raw-data/` directory holds the collected raw data models that were used for data processing. It includes subdirectories for different data sources, such as `github/` (i.e., from GitHub), `genmymodel/` (i.e., from GenMyModel), and `other/` (i.e., from

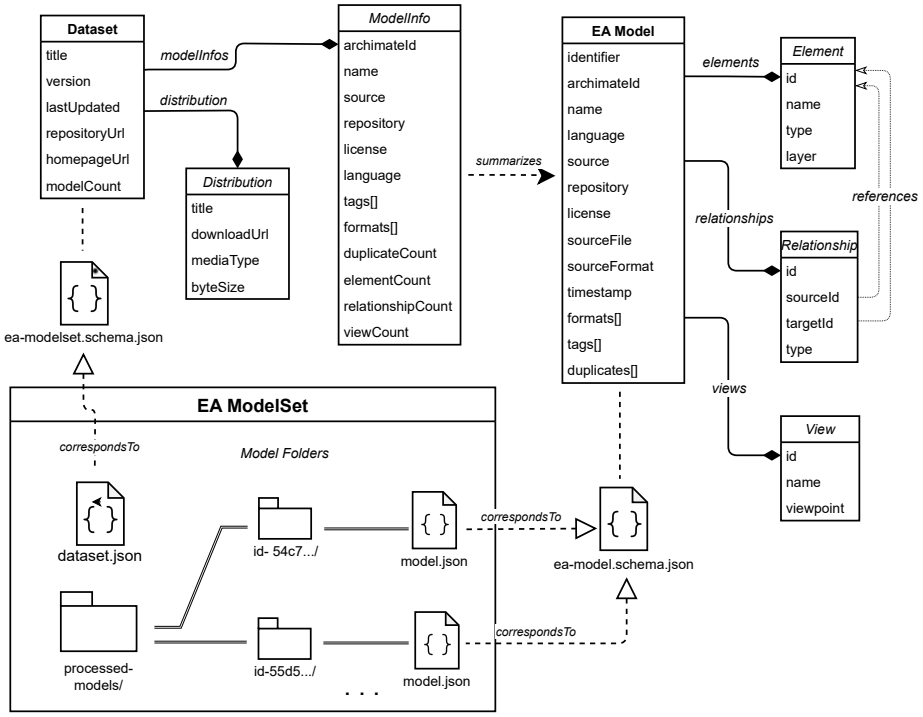


Fig. 4. JSON Schema

miscellaneous sources). The models from GitHub are further organized in three subdirectories `archimate/`, `grafico/`, and `xml/` based on their respective file format.

The main directory for the dataset is the `dataset/` directory, which contains the `dataset.json` file. Within the `processed-models/` directory, each processed model has its own subdirectory and follows a consistent format. A model directory contains the primary JSON model file (`model.json`) and two ArchiMate XML model files (`model.archimate` or `model.xml`). Additionally, models and their contents are stored in separate CSV files within the `csv/` directory.

Fig. 4 illustrates how the JSON schemas are positioned in relation to the dataset to ensure consistency of metadata and data. The `ea-modelset.schema.json` and `ea-model.schema.json` schema files define the structure and validation rules for content in the `dataset.json` and `model.json` files, respectively.

The *Dataset* object contains the dataset metadata and includes information such as the title, version, lastUpdated date, repository URL, homepage URL, distribution details (including distribution title, download URL, media type, and byte size), model count, and an array of *ModelInfo* objects that provide a reduced subset of metadata and computed properties of each individual model. The *EA Model* object provides comprehensive information about each model including its elements, relationships, and views.

3.2 Dataset Description & Statistics

The final EA ModelSet dataset is composed of 863 unique ArchiMate models. The table in Fig. 5 (left) provides some descriptive statistics of the dataset including the sum, average, minimal, and maximum number of elements, relationships, and views. Fig. 5 (right) further shows the distribution of the models by means of relating the number of model elements on the x-axis to the number of model views on the y-axis. It can be derived from these statistics, that the dataset features models of varying size (from 10 up to 4,003 elements, from zero to 5,773 relationships) and the number of views (from one to 357). Further statistics are provided at the EA ModelSet homepage[8].

	Elements	Relationships	Views
Sum	91275	121937	6471
Avg	105,76	141,29	7,50
Min	10	0	1
Max	4003	5773	357
Mode	16	20	1
Median	50	58	2
Stdev	244,01	404,79	23,93

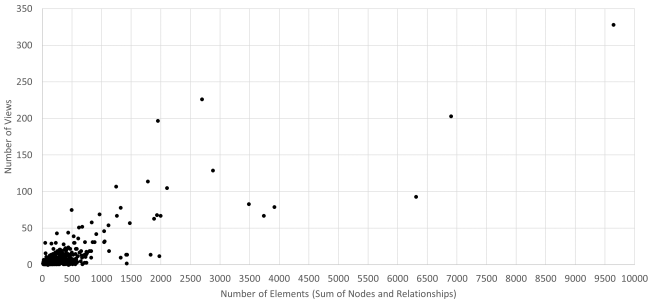


Fig. 5. Descriptive statistics of the EA ModelSet models (left) and distribution of the models with respect to number of model elements and views (right).

3.3 Dataset Usage

The EA ModelSet facilitates various usage scenarios by providing accompanying services and applications. In this section, we describe these different scenarios and the support we provided to efficiently access and utilize the dataset. The dataset and all its related services and applications can be found in the central EA ModelSet GitHub repository, accessible through the assigned pURL[9].

Download Dataset: The dataset can be downloaded as a compressed ZIP file from the GitHub repository's release section[9] with a Git tag introduced for each new version. The ZIP file contains all the necessary files and directories to access and explore the dataset locally (except `raw-data/` files). It serves as the primary method for obtaining the dataset and forms the basis for the accompanying services and applications.

Website: The EA ModelSet has a dedicated website[8] (Fig. 6) that also serves as the landing page for the dataset, offering a user-friendly interface for easy exploration of the models. The website is divided into four sections:

i) Home: The home section serves as the dataset's landing page and as a starting point for users to get acquainted with the dataset. It lists the dataset's metadata, which is read from the `dataset.json` file, ensuring that the information can be easily updated in subsequent releases. The home section also includes a button to download the dataset as a ZIP file (also linked through the JSON file to the released distribution on GitHub).

[8] `https://me-big-tuwien-ac-at.github.io/EAModelSet/home`

[9] `https://purl.org/eamodelset`

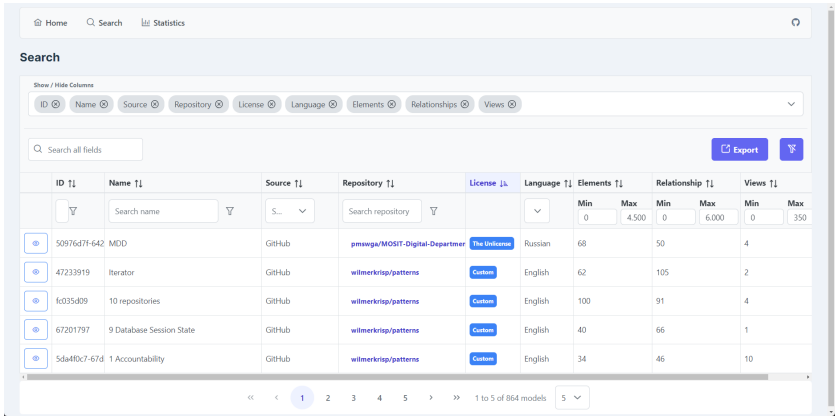


Fig. 6. Search tab of the EA ModelSet homepage

ii) Search: The search interface enables efficient exploration and retrieval of relevant models in the dataset (see Fig. 6). Users can search for specific models based on various criteria, such as model ID, name, tags, language, source, license, or the minimum/maximum number of elements, relationships, or views. The search functionality supports arbitrary combinations of filtering criteria, sorting columns, and a "global search" feature to filter all fields.

iii) Model Details: This page allows in-depth analysis of each model. It can be accessed by navigating from the search section or by following the URL of the model's identifier (`https://me-big-tuwien-ac-at.github.io/EAModelSet/model/<id>`). The details page lists all information related to a specific model, including its metadata, elements, relationships, and views, which are extracted from the respective `model.json` file. Additionally, the associated files of a model (e.g., .json, .xml, .archimate, .csv) can be downloaded directly from this page.

iv) Statistics: The statistics page provides insights into the dataset's composition, complexity, and characteristics through the presentation of key statistics and metrics. Users can explore charts showing the usage of specific languages, layers, element/relationship types, or concrete values for the total number of models, as well as the total, minimum, maximum, and average number of elements, relationships, and views.

Python Library: A dedicated Python library is provided to facilitate programmatic access and analysis of the dataset within a Python environment. The library offers convenient methods to interact with the dataset using a pandas dataframe representation. Users can display the data in a tabular format and use additional filtering functionality to filter models based on various attributes such as source, language, or the minimum/maximum number of elements, relationships, or views. The complete JSON or CSV representation of a model (with all its elements/relationships/views) can then be accessed by passing the model's `ID` property obtained from the dataframe to a provided method. An example showcasing the functionality of the EA ModelSet Python library can be found in the provided Jupyter Notebook[10] in the repository.

[10] `https://github.com/me-big-tuwien-ac-at/EAModelSet/blob/main/python-lib/examples/python-example.ipynb`

Java CLI: For managing and maintaining the dataset, a Java Command-Line Interface (CLI) was realized. The CLI enables users to issue command line commands to perform operations on the dataset like adding or removing models, updating metadata, generating statistics, or validating the dataset's integrity (cf. Section 2.3). The Java CLI also provides the option to connect and load the data into a MongoDB document database or a Neo4j Graph Database for advanced querying and analysis. The use of the functionality of the Java CLI is demonstrated in the Github repository[11].

4 Evaluation Against the FAIR Principles

The FAIR principles provide guidelines to improve the **F**indability, **A**ccessibility, **I**nteroperability, and **R**euse of digital assets [23]. The FAIR principles further emphasize machine-actionability in scientific data management to support dealing with increased volume, complexity, and creation speed of data. In the following, we evaluate the compliance of `EAModelSet` in regard to each FAIR principle.

Findability *F1: "(meta)data are assigned a globally unique and persistent identifier"* The EAModelSet meets this requirement by assigning a Persistent Uniform Resource Locator (pURL) to access the (meta)data stored in the GitHub repository[9]. Furthermore, the dataset is accessible via a globally unique DOI ($10.5281/zenodo.8192011$) and uses ORCID for author identification. Within the dataset, each model has a unique URI, in the form of `https://me-big-tuwien-ac-at.github.io/EAModelSet/model/<id>`, where `<id>` represents a tool-generated Universally Unique Identifier (UUID) or a similar type of identifier for the model. The unique identifier allows direct access to each model and guarantees global uniqueness and unambiguous identification.

F2: "data are described with rich metadata" The dataset provides comprehensive information about each model, capturing e.g., its name, description, source, license, language, and various other attributes (see Fig. 4). The metadata defined in the JSON schema richly describes the data through additional characteristics.

F3: "metadata clearly and explicitly include the identifier of the data it describes" In the JSON representation of the EA Model, the metadata explicitly includes the identifier of the data it describes. Each model is associated with a unique URI identifier that incorporates its ID, providing a clear reference to a model. The ID is based on the `archimateId` property which is also included in the metadata and is an auto-generated UUID already present in the collected data, which is reused.

F4: "(meta)data are registered or indexed in a searchable resource" The EAModelSet is hosted in a public GitHub repository, providing e.g., search functionality and version control to locate and access the dataset. The dedicated website and Python library offer additional functionalities, including search- and filter capabilities to find models based on certain characteristics (e.g., language, views, number of elements).

Accessibility *A1: "(meta)data are retrievable by their identifier using a standardized communications protocol"* Metadata and data are retrievable on GitHub, and also using

[11] `https://github.com/me-big-tuwien-ac-at/EAModelSet/tree/main/cli-app`

the identifier URI leading to the website, which is accessible using an open, free, and universally implementable communications protocol (*A1.1*), e.g., through the HTTP(S) protocol by using a common web browser. The protocol thereby enables free access for use but requires an authentication and authorization procedure for updating the dataset (*A1.2*) (i.e., a GitHub account with the required permissions on the repository.

A2: "metadata are accessible, even when the data are no longer available" The dataset includes an additional JSON file for each model, providing descriptive metadata for each model. This metadata remains accessible even if the actual data associated with the model are no longer available. We further publish the repository releases on persistent data storage via Zenodo [10] to ensure accessibility even if the GitHub repository would not be available anymore.

Interoperability *I1: "(meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation"* The metadata and data are stored in JSON files as the main method for knowledge representation. JSON is a widely adopted format for structuring data in a human-readable and machine-readable manner, and the files correspond to a JSON Schema that provides a formal and standardized syntax. Furthermore, we enable additional data formats, including XML and CSV.

I2: "(meta)data use vocabularies that follow FAIR principles" The dataset employs a customized and adapted meta(data) description that partly reuse subsets of FAIR vocabularies. The dataset reuses vocabularies, e.g., from Data Catalog Vocabulary (DCAT)[12] or Dublin Core Terms (DCT)[13], by translating relevant properties into JSON schema[14]. Relevant datatypes are also translated, e.g., dates are formatted according to the provided datatypes in the JSON schema language (i.e. `date` and `date-time`), and for language codes, the two-letter ISO-6391-1 format is used.

I3: "(meta)data include qualified references to other (meta)data" The dataset itself includes `ModelInfo` objects, which are a lightweight representation of models and include an explicit reference to the actual model. Also, the metadata of each model contains explicit references to related models (e.g. duplicates) or internal (e.g. source file) and external resources (e.g. repository).

Reusability *R1: "(meta)data are richly described with a plurality of accurate and relevant attributes"* Each JSON file contains the (meta)data derived from the source model, together with other relevant properties to richly describe a model (e.g. source, timestamp, language, tags). While the dataset already includes many relevant attributes, there is still room for improvement in terms of enriching the metadata. For instance, additional properties such as categories or more descriptive tags could be incorporated to enhance the richness of the metadata, precise filtering, and analysis.

R1.1: "(meta)data are released with a clear and accessible data usage license" The majority of models in the dataset have their source repository attached as an entry in the JSON file, including information about the license. The repository and license

[12] `www.w3.org/TR/vocab-dcat-2/`

[13] `www.dublincore.org/specifications/dublin-core/dcmi-terms/`

[14] `https://json-schema.org/specification.html`

were automatically retrieved during data collection (see Section 1) and the results were manually re-checked to ensure accuracy. The data usage license is clearly associated with each model, providing information on how to legally use the data.

R1.2: "(meta)data are associated with detailed provenance" The JSON files in each model's folder include properties to present the original source and associated information. The properties provide a level of provenance and include, e.g., source, repository, license, or the parsed source file, allowing to trace back the origin of a model. While the current provenance information offers valuable insights, there is potential for more detailed provenance to be included. For example, associating publications or providing diagrams (e.g, as PNG files) could further enhance the dataset's provenance.

R1.3: "(meta)data meet domain-relevant community standards" The EAModelSet provides models in domain-relevant formats such as ArchiMate XML (two different formats) and CSV. The formats are widely accepted and align with the community's standards, promoting interoperability with existing tools. Furthermore, the newly introduced JSON schema maintains well-established structures and adheres to recognized naming conventions. The introduction of the JSON schema does not add unnecessary complexity, but rather provides clarity and consistency to ensure the metadata is understandable within the EA domain. The CSV formats further ease the execution of ML techniques on the EA ModelSet.

5 EA ModelSet Applications, Reflection, and Future Work

The EA ModelSet dataset provides a rich collection of EA models, unlocking new possibilities for research and practical applications. In this section, we critically reflect on our efforts to establish a FAIR dataset of EA models and discuss potential applications and directions for future research.

Researchers can explore the dataset to gain insights into different modeling approaches, applications, and patterns. By analyzing the models within the dataset, researchers can identify best practices and discover common modeling patterns, which can significantly contribute to advancing the field of EA.

The dataset's availability in different formats, including JSON, XML, and CSV, makes it applicable for a range of ML tasks that extract valuable insights from the data. Some potential applications of ML using EAModelSet include *Natural Language Processing* (NLP) and *Recommender Systems*. The dataset's textual information, e.g., names, documentation, languages, or tags, can be used to develop NLP models that extract meaningful information from unstructured text. This can support tasks such as *automatic model annotation* [1] and *semantic search*. The EA ModelSet dataset can also assist in building recommender systems tailored to EA [16,24]. By analyzing patterns and similarities among EA models, ML algorithms can provide context-based recommendations for specific modeling scenarios. These recommendations can guide architects by suggesting architectural decisions based on historical data, which can enhance productivity and support informed decision-making [5].

While the current EA ModelSet dataset is valuable, there are some limitations and areas for future improvement. One current limitation is its ability to process ArchiMate models in the .archimate and .xml formats. To broaden its applicability in the future,

we aim to incorporate EA models that $i)$ conform to other EA modeling languages, and which were $ii)$ created with different EA modeling tools. Of course, such an extension will require additional research with respect to data harmonization and integration. Even transforming images of models created with other tools to our format is an interesting research challenge.

An additional current shortcoming we aim to address in the future is the fully automated data collection process and to ensure correct record linkage of the source. The current process involving GitHub downloads poses challenges due to authentication, rate limits, and API constraints (e.g., limited file size).

Maintaining data quality and integrity is essential for the EA ModelSet's adoption. Aside from our initial efforts to detect and flag duplicates (based on identifiers and MD5 file hashes) we plan to research and develop more advanced similarity metrics [7] that would help to further clean the data. In terms of data maintenance and publishing, we aim to enhance the dataset's interoperability and operationalizability using an RDF ontology (e.g. [11]). We also aspire to enrich the classification of models by incorporating semantic domain classification (e.g., hotel, banking, insurance). However, such a classification process requires significant manual effort and thus necessitates community engagement.

For all future considerations, we invite and hope to actively engage the enterprise modeling research community. The EA ModelSet is open source, and we plan to realize functionalities that enable efficient contributions from the community, especially with respect to curating the existing dataset and extending the dataset with new models. In conclusion, the EA ModelSet dataset not only empowers current research but also presents a promising platform for future endeavors. With continuous community engagement and improvement efforts, we aspire to make the EA ModelSet a valuable and comprehensive resource for researchers in the enterprise modeling domain.

6 Conclusion

In the paper at hand, we proposed the EA ModelSet, the first FAIR dataset that allows machine learning research in enterprise modeling. The EA ModelSet is a curated dataset that currently contains 863 enterprise architecture models represented by ArchiMate. We believe the EA ModelSet can be an important asset for sparking research at the intersection of machine learning and enterprise modeling. We invite the modeling research community to help further curate and extend the dataset, and also tool vendors to explore their interest in sharing some of their models. The scarcity of models in adequate quantity and quality is a huge barrier to conducting cutting-edge machine learning research in modeling. We hope that the EA ModelSet becomes the central point for FAIR model data which can be freely used to conduct this kind of research.

Acknowledgements

This work has been partially funded through the Erasmus+ KA220-HED project "Digital Platform Enterprise" (project no. 2021-1-RO01-KA220-HED-000027576) and the Vienna Science and Technology Fund (WWTF) (10.47379/VRG18013).

References

1. Ali, S.J., Guizzardi, G., Bork, D.: Enabling representation learning in ontology-driven conceptual modeling using graph neural networks. In: Indulska, M., Reinhartz-Berger, I., Cetina, C., Pastor, O. (eds.) Advanced Information Systems Engineering - 35th International Conference, CAiSE 2023, Zaragoza, Spain, June 12-16, 2023, Proceedings. Lecture Notes in Computer Science, vol. 13901, pp. 278–294. Springer (2023). https://doi.org/10.1007/978-3-031-34560-9_17
2. Barbosa, A.O., Santana, A., Hacks, S., von Stein, N.: A taxonomy for enterprise architecture analysis research. In: 21st International Conference on Enterprise Information Systems, ICEIS 2019. pp. 493–504. SciTePress (2019). https://doi.org/10.5220/0007692304930504
3. Barcelos, P.P.F., Sales, T.P., Fumagalli, M., et al.: A FAIR model catalog for ontology-driven conceptual modeling research. In: 41st International Conference on Conceptual Modeling, ER 2022. pp. 3–17. Springer (2022). https://doi.org/10.1007/978-3-031-17995-2_1
4. Bernabé, C., Sales, T.P., Schultes, E., et al.: A goal-oriented method for fairification planning (2023). https://doi.org/10.21203/rs.3.rs-3092538/v1
5. Bork, D., Ali, S.J., Dinev, G.M.: Ai-enhanced hybrid decision management. Bus. Inf. Syst. Eng. **65**(2), 179–199 (2023). https://doi.org/10.1007/s12599-023-00790-2
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. Borozanov, V., Hacks, S., Silva, N.: Using machine learning techniques for evaluating the similarity of enterprise architecture models - technical paper. In: Advanced Information Systems Engineering - 31st International Conference. pp. 563–578 (2019)
8. Corradini, F., Fornari, F., Polini, A., et al.: Reprository: a repository platform for sharing business process models and logs. In: Proceedings of the 1st Italian Forum on Business Process Management. pp. 13–18. CEUR-WS.org (2021)
9. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of bpm: Model collections, `http://fundamentals-of-bpm.org/process-model-collections/`, last accessed: 24.07.2023
10. Glaser, P.L., Sallinger, E., Bork, D.: EA ModelSet (Jul 2023). https://doi.org/10.5281/zenodo.8192011
11. Hinkelmann, K., Laurenzi, E., Martin, A., et al.: Archimeo: A standardized enterprise ontology based on the archimate conceptual model. In: Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2020. pp. 417–424. SCITEPRESS (2020). https://doi.org/10.5220/0009000204170424
12. López, J.A.H., Cuadrado, J.S.: An efficient and scalable search engine for models. Softw. Syst. Model. **21**(5), 1715–1737 (2022). https://doi.org/10.1007/s10270-021-00960-4
13. López, J.A.H., Izquierdo, J.L.C., Cuadrado, J.S.: Modelset: a dataset for machine learning in model-driven engineering. Softw. Syst. Model. **21**(3), 967–986 (2022). https://doi.org/10.1007/s10270-021-00929-3
14. López, J.A.H., Izquierdo, J.L.C., Cuadrado, J.S.: Using the modelset dataset to support machine learning in model-driven engineering. In: Kühn, T., Sousa, V. (eds.) 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS 2022. pp. 66–70. ACM (2022). https://doi.org/10.1145/3550356.3559096
15. Pezoa, F., Reutter, J.L., Suárez, F., et al.: Foundations of JSON schema. In: 25th International Conference on World Wide Web, WWW 2016. pp. 263–273. ACM (2016)
16. Raavikanti, S., Hacks, S., Katsikeas, S.: A recommender plug-in for enterprise architecture models. In: 25th International Conference on Enterprise Information Systems, ICEIS 2023. pp. 474–480. SCITEPRESS (2023). https://doi.org/10.5220/0011709000003467

17. Rahman, M.I., Panichella, S., Taibi, D.: A curated dataset of microservices-based systems. CoRR **abs/1909.03249** (2019), http://arxiv.org/abs/1909.03249
18. Robles, G., Ho-Quang, T., Hebig, R., et al.: An extensive dataset of UML models in github. In: 14th International Conference on Mining Software Repositories, MSR 2017. pp. 519–522. IEEE Computer Society (2017). https://doi.org/10.1109/MSR.2017.48
19. Schäfer, B., van der Aa, H., Leopold, H., Stuckenschmidt, H.: Sketch2bpmn: Automatic recognition of hand-drawn BPMN models. In: 33rd International Conference Advanced Information Systems Engineering. pp. 344–360. Springer (2021)
20. Shilov, N., Othman, W., Fellmann, M., Sandkuhl, K.: Machine learning for enterprise modeling assistance: an investigation of the potential and proof of concept. Softw. Syst. Model. **22**(2), 619–646 (2023). https://doi.org/10.1007/s10270-022-01077-y
21. da Silva Santos, L.O.B., Sales, T.P., Fonseca, C.M., Guizzardi, G.: Towards a conceptual model for the FAIR digital object framework. CoRR **abs/2302.11894** (2023). https://doi.org/10.48550/arXiv.2302.11894
22. Sola, D., Warmuth, C., Schäfer, B., et al.: SAP signavio academic models: A large process model dataset. In: Process Mining Workshops - ICPM 2022 International Workshops. pp. 453–465. Springer (2022). https://doi.org/10.1007/978-3-031-27815-0_33
23. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data **3**(1), 160018 (2016). https://doi.org/10.1038/sdata.2016.18
24. Zhi, Q., Zhou, Z.: Empirically modeling enterprise architecture using archimate. Comput. Syst. Sci. Eng. **40**(1), 357–374 (2022). https://doi.org/10.32604/csse.2022.018759