Towards the Integration Support for Machine Learning of Inter-Model Relations in Model Views

Conference Paper · April 2024

DOI: 10.1145/3605098.363614

4 authors, including:

James William Pontes Miranda
IMT Atlantique

Hugo Bruneliere
IMT Atlantique

Gerson Sunyé
University of Nantes

Towards the Integration Support for Machine Learning of Inter-Model Relations in Model Views

James Pontes Miranda
IMT Atlantique, LS2N (UMR CNRS 6004)
Nantes, France
james.pontes-miranda@imt-atlantique.fr

Hugo Bruneliere
IMT Atlantique, LS2N (UMR CNRS 6004)
Nantes, France
hugo.bruneliere@imt-atlantique.fr

Massimo Tisi
IMT Atlantique, LS2N (UMR CNRS 6004)
Nantes, France
massimo.tisi@imt-atlantique.fr

Gerson Sunyé
Nantes University, LS2N (UMR CNRS 6004)
Nantes, France
gerson.sunye@univ-nantes.fr

ABSTRACT

Model-driven engineering (MDE) supports the engineering of complex systems via multiple models representing various systems' aspects. These interrelated models are usually heterogeneous and often specified using complementary modeling languages. Whenever needed, model view solutions can be employed to federate these models in a more transparent way. To do so, the required inter-model links can sometimes be automatically computed via explicitly written matching rules. However, in some cases, matching rules would be too complex (or even impossible) to write. Thus, some inter-model links may be inferred by analyzing previous examples instead. In this paper, we introduce a Machine Learning (ML)-backed approach for expressing and computing such model views. Notably, we aim at making the use of ML as simple as possible in this context. To this end, we propose to refine and extend the ViewPoint Definition Language (VPDL) from the EMF Views model view solution to integrate the use of dedicated Heterogeneous Graph Neural Networks (HGNNs). These view-specific HGNNs can be trained with appropriate sets of contributing models before being used for inferring links to be added to the views.

CCS CONCEPTS

• **Software and its engineering** → *System modeling languages*; Integration frameworks; • **Computing methodologies** → *Neural networks*.

KEYWORDS

Model Driven Engineering, Modeling Languages, Model Views, Machine Learning, Graph Neural Networks.

ACM Reference Format:
James Pontes Miranda, Hugo Bruneliere, Massimo Tisi, and Gerson Sunyé. 2024. Towards the Integration Support for Machine Learning of Inter-Model Relations in Model Views. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3605098.3636143

SAC '24, April 8–12, 2024, Avila, Spain

1 INTRODUCTION

Complex systems engineering is challenging because of the various aspects to be tackled, including information fragmentation among stakeholders [1, 5]. To overcome this, Model-Driven Engineering (MDE) fosters using multiple models as fundamental artifacts to support analysts, engineers, etc., in performing their tasks more efficiently. However, dealing with heterogeneous models defined in different modeling languages at various abstraction levels (e. g., UML, SysML, BPMN, or DSLs for problem-specific tasks) requires appropriate model federation strategies.

Model View solutions are efficient ways to deal with model federation [3]. Model views are built over one or several existing models, called contributing models, that possibly conform to different metamodels. A model view allows users to access information from different contributing models in an integrated and transparent way. The elements of the contributing models are integrated by adding new *inter-model links* to the model view. Such links are concrete instances of an *inter-model relation* among the contributing models. Sometimes, inter-model links can be automatically computed via matching rules written by the engineers. However, specifying such rules requires a deep knowledge of the involved models and metamodels. Moreover, these rules may be too complex to write manually using a query language. As a consequence, automating the derivation of these rules has already been identified as an important challenge [6].

Machine Learning (ML) approaches have already been exploited to improve various model management operations [2, 9, 17, 18]. For example, Graph Neural Networks (GNNs), i. e., deep-learning models optimized for operation on graph-structured data, have already been successfully used in recommendation systems for model editing [10] or for the generation of structurally-realistic models [14]. Among other benefits, they have notably demonstrated their ability to capture interesting structural properties of model graphs from a limited set of examples.

In this paper, we introduce a ML-backed approach for computing model views that require the inference of inter-model links. Our objective is making the work of the view engineer easier by simplifying the use of ML as much as possible, so that she/he does not need to write ML code. To realize our approach, we rely on the link prediction capabilities of GNNs. Unlike previous research efforts, we propose to rely on Heterogeneous Graph Neural Networks (HGNNs), a particular class of GNNs that have native support for graphs whose nodes and edges have different types (which is the case in model views). In practice, we propose to extend the EMF Views[1] solution [4] and its ViewPoint Definition Language (VPDL) to integrate HGNNs. The engineer only needs to indicate 1) the relations to learn, 2) the parts of the models involved in the learning process, and 3) a relevant set of sample links. Then, a declarative description of the architecture and configuration for the corresponding HGNNs can be automatically generated and eventually manually updated by an ML specialist. These HGNNs can be transparently trained and finally used to infer inter-model links integrated into model views.

2 PROPOSED APPROACH

2.1 Overview

In the standard EMF Views approach (cf. the lower part of Figure 1), the view engineer has to provide two artifacts: a *Viewpoint definition* at the metamodel level and a *View definition* at the model level. In EMF Views, these two artifacts can be partially generated from a specification in VPDL, as described in the next subsection. Then, the *View Builder* takes these two artifacts as inputs and builds a virtual model that materializes the specified *Model View*.

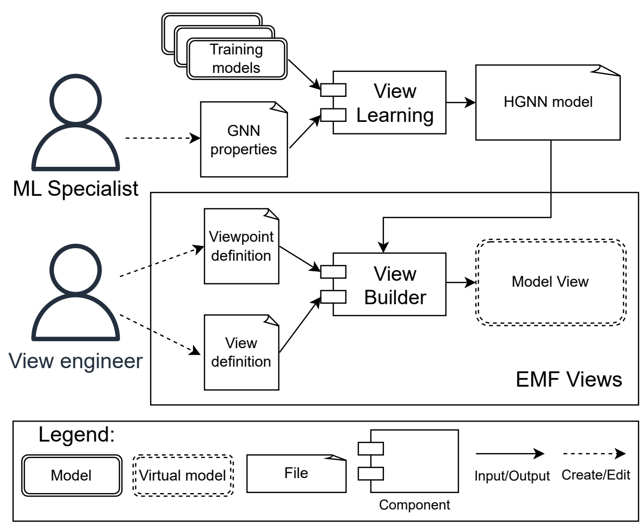

Figure 1: Overview of the proposed approach

In our approach (cf. the upper part of Figure 1), we propose to complement EMF Views with a new *View Learning* component to support the *View Builder* base component. A set of assignments for *GNN properties* is computed from the *Viewpoint definition*. They describe the architecture of the GNN and the hyperparameters for link prediction, including training and embedding. A *ML Specialist* can possibly edit the value of these properties, e. g. to fine-tune the learning step. *Training models* are also required, including existing links used as examples for learning. Such existing models can come from different sources, e. g. legacy projects. Then, the *View Learning* component takes these two artifacts as inputs and generates a trained HGNN model. The set of inter-model links are computed by the *View Builder* component using the HGNN model, before constructing the corresponding view.

Note that the EMF Views solution already supports delegating the computation of inter-model links to external tools. Hence, our approach can reuse the standard structure of the *Viewpoint definition* and the standard *View Builder* component from EMF Views with no modifications. Moreover, the approach aims at decoupling the contributions of the *View engineer* and the *ML specialist*. Thus, the *ML specialist* can support the engineer by working on improving the accuracy and relevance of the inferred links without affecting the original *Viewpoint definition* and *View definition* made by the *View engineer*. Overall, we intend to make the use of ML as transparent as possible from the *View engineer* perspective by delegating the ML integration and execution to our approach (and possible ML-specific optimizations to the ML specialist).

2.2 Extended ViewPoint Definition Language

We propose to rely on the standard EMF Views for partially generating the *Viewpoint definition* and *View definition* from a specification in VPDL. Then, these artifacts are manually completed to point to the actual resources, i. e., the contributing metamodels and models, respectively. Additionally, our approach exploits our VPDL extension for generating default GNN architectures and hyperparameters (based on previous experiments) for the learning process.

To illustrate this, we refer in what follows to a small example where a Users model contains personal information extracted from a social network, while a Movies model contains information from a film database. The goal is to automatically compute a model view containing users, movies, and links connecting each user with the *movies they probably watched*.

Suppose we have a data set describing *other users* with information about watched movies, coming from the MovieLens data set [12] for instance. We aim to learn a mathematical relation between each user and their watched movies from this data set. This relation is then applied in the view to compute new inter-model links between our users and movies they probably watched. For this purpose, we use HGNNs in VPDL.

```
create view watched as

select Users.User[id, name],
      Movies.Movie.*,
      Users.User join Movies.Movie
      as watched

from 'http://paper/movies' as Movies,
    'http://paper/users' as Users

where  "{s.id}<~s.movies~>{t.id, t.genres.value}"
       for watched
```

Figure 2: Extended VPDL for the recommendation example

Figure 2 shows a snippet of our viewpoint specification in Extended VPDL for computing recommendations on this example. The *create* and *from* parts are standard VDPL. However, the *where* part contains a specific expression indicating, for each inter-model relation, the properties of the two models and the training relation to be considered for learning. It contains:

[1] https://github.com/atlanmod/emfviews/

- A set of navigation paths starting from the source of the relation `s`, indicating the properties that should be considered for characterizing the source element. In our case, `{s.id}` indicates that the learning system will only use the `id` of the user (and not the name, age, etc.).
- A set of navigation paths starting from the target of the relation `t`, indicating the properties that should be considered for characterizing the target element. In our case, `{t.id, t.genres.value}` indicates that the learning system will use the `id` of the movie and the list of its `genres`. Note that the navigation expression can navigate the model to access attributes of other model elements, e. g. `Genre`.
- A navigation path indicating an existing relation used as the source of examples. This path is always represented between the two previous sets, with a specific arrow notation. In our case, `<~s.movies~>` indicates that the learning system will consider the `movies` relation as the set of examples to learn from (in the direction starting from `s`).

3 CONCLUSION AND FUTURE WORK

In this paper, we introduced an approach for automatically inferring inter-model links in the context of model views. To this end, we proposed to refine and extend the existing VPDL model view specification language from EMF Views to properly integrate the automated generation and use of view-dedicated HGNNs. In practice, we implemented an initial prototype of our approach by combining the Eclipse-based EMF Views solution with two Python libraries, PyEcore[2] and PyTorch Geometric[3], dedicated to model handling and HGNNs, respectively.

The related work on learning constraints and model transformations is particularly interesting in our context, as model views can be possibly implemented by model transformations. On the constraints side, Dang & Cabot [8] proposed the InferOCL tool for the automatic inference of OCL constraints (based on examples) on conceptual models and metamodels. On the model transformation side, Model Transformations by Example (MTBE) [13, 16], a user-friendly approach that aims at generating transformation rules from example of inter-model mappings, have already been quite studied. More recently, Burgueno et al. proposed a generic neural network architecture to support the automated inference of model-to-model and model-to-text transformations [7]. Compared to these research efforts, we focus more on computing relevant possibly existing inter-model links while limiting the number of ML code to be written by engineers.

Future work includes continuing the development and evaluation of the initial prototype of our proposed approach. We also plan to work on a dedicated language for GNN Properties, more smoothly integrated with VPDL. To this end, we can study the potential reuse of existing work on DSLs for supporting ML activities [11, 15].

ACKNOWLEDGMENT

This work was co-funded by the AIDOaRt ECSEL Joint Undertaking (JU) project under grant agreement No. 101007350, and by the French region Pays de la Loire.

[2] https://github.com/pyecore
[3] https://pytorch-geometric.readthedocs.io/

REFERENCES

[1] Wasif Afzal, Hugo Bruneliere, Davide Di Ruscio, Andrey Sadovykh, Silvia Mazzini, Eric Cariou, Dragos Truscan, Jordi Cabot, Abel Gómez, Jesús Gorroñogoitia, Luigi Pomante, and Pavel Smrz. 2018. The MegaM@Rt2 ECSEL project: MegaModelling at Runtime – Scalable model-based framework for continuous development and runtime validation of complex systems. *Microprocessors and Microsystems* 61 (Sept. 2018), 86–95. https://doi.org/10.1016/j.micpro.2018.05.010

[2] Angela Barriga, Rogardt Heldal, Adrian Rutle, and Ludovico Iovino. 2022. PARMOREL: a framework for customizable model repair. *Software and Systems Modeling* 21, 5 (Oct. 2022), 1739–1762. https://doi.org/10.1007/s10270-022-01005-0

[3] Hugo Bruneliere, Erik Burger, Jordi Cabot, and Manuel Wimmer. 2019. A feature-based survey of model view approaches. *Software & Systems Modeling* 18, 3 (June 2019), 1931–1952. https://doi.org/10.1007/s10270-017-0622-9

[4] Hugo Bruneliere, Florent Marchand de Kerchove, Gwendal Daniel, Sina Madani, Dimitris Kolovos, and Jordi Cabot. 2020. Scalable model views over heterogeneous modeling technologies and resources. *Software and Systems Modeling* 19, 4 (July 2020), 827–851. https://doi.org/10.1007/s10270-020-00794-6

[5] Hugo Bruneliere, Vittoriano Muttillo, Romina Eramo, Luca Berardinelli, Abel Gómez, Alessandra Bagnato, Andrey Sadovykh, and Antonio Cicchetti. 2022. AIDOaRt: AI-augmented Automation for DevOps, a model-based framework for continuous development in Cyber–Physical Systems. *Microprocessors and Microsystems* 94 (Oct. 2022), 104672. https://doi.org/10.1016/j.micpro.2022.104672

[6] Antonio Bucchiarone, Jordi Cabot, Richard F. Paige, and Alfonso Pierantonio. 2020. Grand challenges in model-driven engineering: an analysis of the state of the research. *Software and Systems Modeling* 19, 1 (Jan. 2020), 5–13. https://doi.org/10.1007/s10270-019-00773-6

[7] Loli Burgueño, Jordi Cabot, Shuai Li, and Sébastien Gérard. 2022. A generic LSTM neural network architecture to infer heterogeneous model transformations. *Software and Systems Modeling* 21, 1 (Feb. 2022), 139–156. https://doi.org/10.1007/s10270-021-00893-y

[8] Duc-Hanh Dang and Jordi Cabot. 2015. On Automating Inference of OCL Constraints from Counterexamples and Examples. In *Knowledge and Systems Engineering (Advances in Intelligent Systems and Computing)*, Viet-Ha Nguyen, Anh-Cuong Le, and Van-Nam Huynh (Eds.). Springer International Publishing, Cham, 219–231. https://doi.org/10.1007/978-3-319-11680-8_18

[9] MohammadHadi Dehghani, Shekoufeh Kolahdouz-Rahimi, Massimo Tisi, and Dalila Tamzalit. 2022. Facilitating the migration to the microservice architecture via model-driven reverse engineering and reinforcement learning. *Software and Systems Modeling* 21, 3 (June 2022), 1115–1133. https://doi.org/10.1007/s10270-022-00977-3

[10] Juri Di Rocco, Claudio Di Sipio, Davide Di Ruscio, and Phuong T. Nguyen. 2021. A GNN-based Recommender System to Assist the Specification of Metamodels and Models. In *2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*. 70–81. https://doi.org/10.1109/MODELS50736.2021.00016

[11] Joan Giner-Miguelez, Abel Gómez, and Jordi Cabot. 2023. A domain-specific language for describing machine learning datasets. *Journal of Computer Languages* 76 (Aug. 2023), 101209. https://doi.org/10.1016/j.cola.2023.101209

[12] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (Jan. 2016), 1–19. https://doi.org/10.1145/2827872

[13] Gerti Kappel, Philip Langer, Werner Retschitzegger, Wieland Schwinger, and Manuel Wimmer. 2012. Model Transformation By-Example: A Survey of the First Wave. In *Conceptual Modelling and Its Theoretical Foundations: Essays Dedicated to Bernhard Thalheim on the Occasion of His 60th Birthday*, Antje Düsterhöft, Meike Klettke, and Klaus-Dieter Schewe (Eds.). Springer, Berlin, Heidelberg, 197–215. https://doi.org/10.1007/978-3-642-28279-9_15

[14] José Antonio Hernández López and Jesús Sánchez Cuadrado. 2023. Generating Structurally Realistic Models With Deep Autoregressive Networks. *IEEE Transactions on Software Engineering* 49, 4 (April 2023), 2661–2676. https://doi.org/10.1109/TSE.2022.3228630 Conference Name: IEEE Transactions on Software Engineering.

[15] Zahra Rajaei, Shekoufeh Kolahdouz-Rahimi, Massimo Tisi, and Frédéric Jouault. 2021. A DSL for Encoding Models for Graph-Learning Processes. https://hal.science/hal-03252919

[16] Michael Strommer. 2008. *Model Transformation By-Example.* Ph. D. Dissertation. Vienna University of Technology, Vienna. https://publik.tuwien.ac.at/files/PubDat_166165.pdf

[17] Xiangru Tang, Zhihao Wang, Jiyang Qi, and Zengyang Li. 2019. Improving code generation from descriptive text by combining deep learning and syntax rules. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE* 2019-July (2019), 385–390. https://doi.org/10.18293/SEKE2019-170 ISBN: 1891706489.

[18] Martin Weyssow, Houari Sahraoui, and Eugene Syriani. 2022. Recommending metamodel concepts during modeling activities with pre-trained language models. *Software and Systems Modeling* 21, 3 (June 2022), 1071–1089. https://doi.org/10.1007/s10270-022-00975-5