M2M Mapping and Transformation Tool for Labeled Property Graphs

Pedro Guimarães[1,2,*], Ana León[2,3] and Maribel Yasmina Santos[2]

[1]CCG/ZGDV - ICT Innovation Institute, Campus de Azurém, 4800-058 Guimarães, Portugal
[2]ALGORITMI Research Centre, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
[3]Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València, Camí de Vera S/N, Valencia, Spain

Abstract

Class Diagrams and Labeled Property Graphs are essential concepts in software development and are often used together in Model Driven Development approaches. This paper follows up on our previous work [1] that presented a new approach to transform models based on the analytical requirements. Nevertheless, these transformations can be complex and time-consuming when not supported by any tool. This follow-up paper presents a tool that supports the systematic applicability of transformation patterns by integrating several components and technologies. The proposed solution uses a rule-based approach to detect patterns in the input Class Diagrams model, performs data transformations, and generates mappings between different data models to obtain one or more Labeled Property Graphs.

Keywords

Labeled Property Graphs, Class Diagrams, Model-to-Model Transformation, Conceptual Meta-model

1. Introduction

There has been a growing trend to automate Model-to-Model (M2M) transformations, although these are mainly focused on straightforward transformations from Class Diagram (CD) to specific domains [2, 3, 4, 5]. León et al. [6] propose a transformation process that follows a systematic approach based on patterns, such as aggregations, compositions, and specializations. The process of transforming a CD into a LPG (Labeled Property Graph) should include a systematic set of steps to ensure that the resulting Property Graph is the most appropriate for the given context. The previous work presented in Guimarães et al. [1] assessed different transformation tools considering five dimensions, namely the use of a meta-model with background information, the use of predefined patterns, the application of rules in the transformation process, the analysis of the analytical relevance of the domain concepts, and the compatibility with graph databases. Advancing this previous work, this paper proposes a tool that integrates all these dimensions and performs pattern mapping with the support of a patterns meta-model, and presents the defined architecture and the implemented solution, demonstrating its feasibility in automatically

ER2023: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, Project Exhibitions, Posters and Demos, and Doctoral Consortium, November 06-09, 2023, Lisbon, Portugal

[*]Corresponding author.

pedro.guimaraes@ccg.pt (P. Guimarães); aleon@vrain.upv.es (A. León); maribel@dsi.uminho.pt (M. Y. Santos)
0000-0003-3390-8528 (P. Guimarães); 0000-0003-3516-8893 (A. León); 0000-0002-3249-6229 (M. Y. Santos)

CEUR Workshop Proceedings (CEUR-WS.org)

transforming a given CD into one or more LPGs. This paper is organized as follows. Section 2 describes the tool technological components and Section 3 presents the conclusions and guidelines for future work.

2. Tool Technological Components

The M2M Mapping and Transformation Tool is a system designed to convert UML CD represented in an XML file into LPGs, represented as a set of Cypher scripts. The process that enables this transformation is more than a monolithic operation, it is an orchestration of various distinct components, each with its specific role, contributing to the system's overall functionality and effectiveness. In this section, we delve into the architecture of the M2M Mapping and Transformation Tool, exploring the role of each component. We examine their relevance in the operation of the system, the challenges faced during their design and implementation, and the strategies employed to overcome these challenges. This comprehensive overview, depicted in Figure 1, shows the tool's underlying logic, demonstrating how the integration of different components results in seamless data transformation.

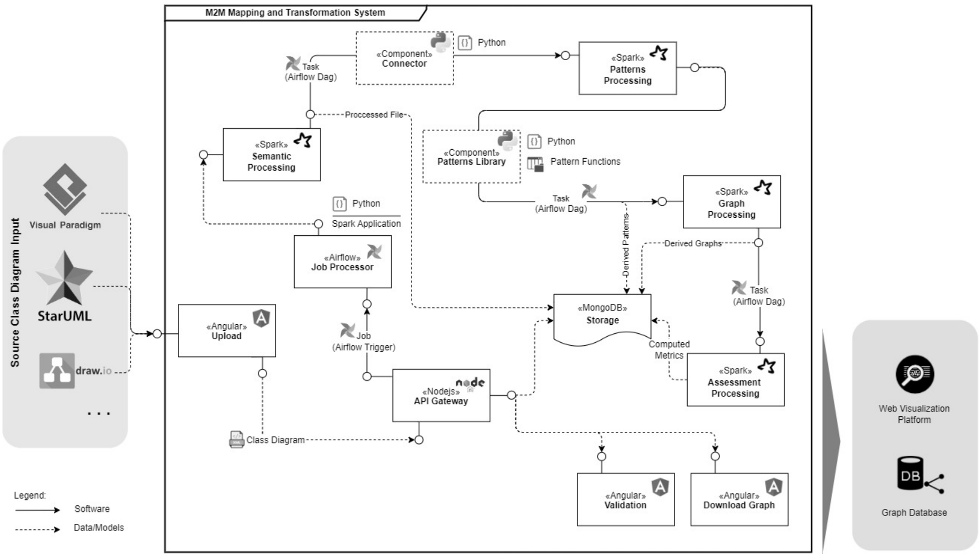

Figure 1: M2M Mapping and Transformation Tool Technological Components

The M2M Mapping and Transformation Tool leverages a range of software components to run its conceptual approach. The Web App is a user-friendly interface, allowing Domain Experts to input their CD and validate the output models. The API Gateway manages the communication between the various components, ensuring smooth data flow throughout the transformation pipeline. Apache Airflow orchestrates the sequence of the transformation process, ensuring the orderly execution of tasks. Apache Spark, used in every processing stage, offers scalability and efficient distributed data processing. MongoDB, the tool's primary storage system, ensures that the output of each stage is safely stored and can be traced back for auditing or troubleshooting purposes.

The tool relies on Connectors that perform the role of parsing and identifying relevant elements from the CD XML files that the Domain Expert uploads. They are responsible for translating the data from its original form into a structure that the M2M Mapping and Transformation Tool can process. This is required because different modeling tools represent CDs in XML files with different structures. Therefore, the connectors ensure that the tool is able to transform the CDs independently of the tool used to model them. Without them, the tool would be limited to a specific software or data structure, significantly limiting its versatility and applicability. Moreover, these connectors are designed to be extensible. Additionally, this approach empowers third-party developers to create their own connectors that interface with this technological solution, expanding its compatibility.

Also, the tool uses a Patterns Library that holds a set of functions designed to systematically identify and map patterns found within a UML CD file by following a set of rules [1]. Each pattern has its own unique way of being identified. The tool uses several elements, such as class type, cardinality, association type, and the analytical value of the classes, among others characteristics, to identify and apply patterns in a UML CD. The mapping functions focus on specific properties and relationships between classes, applying logic to manipulate these characteristics as needed. The transformation tool integrates Vis.js to visualize the models obtained at different stages of the transformation process (Figure 2). This is relevant in this tool as it does more than just rendering static images of models. Deployed as a Web Application, provides interactive and dynamic visual representations that allow Domain Experts to manipulate and inspect the models, promoting an active interaction between the user and the system. Domain Experts can navigate through these visual representations, zooming in and out, rearranging elements, and exploring different views and details of the model. As shown in Figure 2, for a given CD as input, one or more LPGs can be identified as output. The domain expert is responsible for both verifying and validating the output graphs, using the interactivity facilitated by Vis.js. This ensures that the resulting model aligns accurately with the user's requirements.

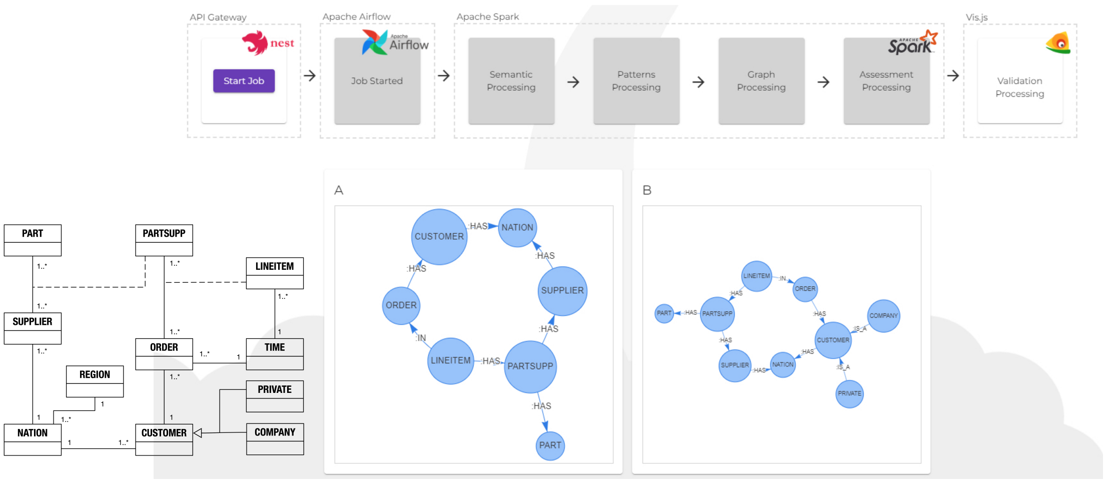

Figure 2: Visualization of the Mapping and Transformation Process Results

The integration between the conceptual and the technological layers is what enables the M2M

Mapping and Transformation Tool to function effectively. The conceptual approach sets the ‘what’ and the ‘why’, defining what needs to be done and why it is being done. The technological components provide the ’how’, making available the means by which the conceptual goals can be achieved. The formalized patterns and transformation rules defined by León et al. [6] are strictly followed by the tool, ensuring a rigorous, yet flexible transformation process. The technological components allow the M2M Mapping and Transformation Tool to remain responsive and adaptable, capable of meeting a diverse range of domain modeling needs while still ensuring a robust and efficient transformation process.

3. Conclusions

LPGs have become popular for representing complex data, but the manual transformation of CDs into LPGs can be complex and time-consuming. While related transformation processes mainly focused on transforming CDs to specific domains, this work is focused on pattern-based transformations from CDs to LPGs. Through the integration of a formalized meta-model, technological components, and a system implementation, we have presented a tool that further enhances this process identifying one or more LPGs for a given CD.

Acknowledgements

This work has been supported by *FCT – Fundação para a Ciência e Tecnologia* within the R&D Units Project Scope: UIDB/00319/2020, and by the Spanish Ministry of Universities and the Universitat Politècnica de València under the Margarita Salas Next Generation EU grant.

References

[1] P. Guimarães, A. León, M. Y. Santos, An automated patterns-based model-to-model mapping and transformation system for labeled property graphs, in: S. Nurcan, A. L. Opdahl, H. Mouratidis, A. Tsohou (Eds.), Research Challenges in Information Science: Information Science and the Connected World (Vol. 476), Springer Nature Switzerland, 2023, pp. 171–186. doi:10.1007/978-3-031-33080-3_11.

[2] F. Abdelhedi, A. A. Brahim, F. Atigui, G. Zurfluh, Mda-based approach for nosql databases, in: 19th International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2017), 2017, pp. 88–102.

[3] R. Billen, UML as a schema candidate for graph databases, Technical Report, University of Liège, 2014.

[4] M. J. Mior, K. Salem, A. Aboulnaga, R. Liu, Nose: Schema design for nosql applications, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2275–2289.

[5] T. Wang, S. Truptil, F. Benaben, An automatic model-to-model mapping and transformation methodology to serve model-based systems engineering, Inf Syst E-Bus Manage 15 (2017).

[6] A. León, M. Y. Santos, A. García, J. C. Casamayor, O. Pastor, Model-to-model transformation: From uml class diagrams to labeled property graphs, Business & Information Systems Engineering (2023). doi:10.1007/s12599-023-00824-9.