Towards Data-Metadata Flexibility in Property Graph Data Management

Sepehr Sadoughi
Supervised by Prof. Dr. George Fletcher and Dr. Nikolay Yakovets
Eindhoven University of Technology
Eindhoven, Netherlands
s.sadoughi@tue.nl

ABSTRACT

Graph-based data management solutions are gaining widespread adoption, with many leading graph database vendors, such as Neo4j and Tigergraph, relying on the property graph data model to model, store, and query complex, interconnected data. A key strength of graph data models is that they can facilitate reasoning over different collections of data by providing a flexible and intuitive representation of entities and their relationships, enabling seamless traversal, pattern discovery, and contextual inference. Recently, ISO introduced the GQL/SQL-PGQ standard to formalize this model. However, when it comes to managing heterogeneous and conflicting data sources, this model falls short in several aspects. This PhD research investigates whether a data model and retrieval paradigm can be developed to support effective management and inference over heterogeneous collections of property graphs. A major source of heterogeneity in property graph collections arises from data-metadata misalignment, where metadata (e.g. labels, property keys) and data (e.g. property values) may shift roles across different graphs. This inconsistency challenges integration and inference. To address this, I aim to (1) explore how current property graph data model and query language handle such heterogeneity, (2) investigate extendeding current data models so that it can support flexible representation of heterogeneous structures and metadata, (3) and finally see how retrieval paradigms can be designed to support effective inferences over heterogeneous property graphs. Toward this end, we introduced Meta-Property Graph, a backward-compatible extension of the property graph model that enables flexibility in the treatment of data and metadata in property graphs and MetaGPML, as an extension of the Graph Pattern Matching Language (GPML), which underpins the ISO GQL standard, to enable querying these enhanced graphs. This foundational model and the pattern-matching language set the stage for more effective data management and more expressive inferences over heterogeneous graph collections.

VLDB Workshop Reference Format:
Sepehr Sadoughi
Supervised by Prof. Dr. George Fletcher and Dr. Nikolay Yakovets. Towards Data-Metadata Flexibility in Property Graph Data Management. VLDB 2025 Workshop: Ph.D. Workshop.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

1 INTRODUCTION

Motivation. Modern data engineering applications demand flexibility and agility more than ever before. For example, during exploratory analytics, one typically starts in a schema-less situation, where structure is discovered over time. Heterogeneity in what is a data value versus what is an attribute name is inherent during discovery, profiling, and exploration. As another example, during the integration of data sources, such as when building a knowledge graph, data and schema heterogeneity are the default. One source's data (e.g. attribute values) may correspond to another source's node labels, and in yet another data source, these correspond to node property names, which are considered as metadata; further, a combination of nodes and relationships, i.e. subgraph in one source corresponds to a node in a different data source. Modern data management solutions must fully embrace and support heterogeneity and diversity in modeling distinctions between data and metadata.

In data modelling, metadata is commonly understood in two distinct ways. The first and most common form is *attribute data*, which refers to the characteristics or details associated with a piece of information, such as a relational database table's attributes (i.e., column names) or the table's name itself. For instance, when storing a person's contact information, attributes such as name, email address, phone number, and address are essential metadata. In the property graph data model, this type of metadata corresponds to property keys or labels of a node or relationship. On the other hand, another common form of metadata is *reification*. This concept involves aggregating complex relationships or sets into a new entity or object, making it easier to manage and analyze. In the context of entity-relationship modeling, reification is often used to transform relationship sets into entity sets, allowing for more effective data modeling and manipulation. For example, in an online shopping platform, reifying the relationship between customers and orders could result in a new entity representing the order history or customer preferences.[1]

The ISO standard Property Graph (PGs) model has gained popularity in graph data management and is widely adopted, e.g., in graph DB systems such as Neo4j, Tigergraph, and Amazon Neptune, as well as relational DB systems such as DuckDB which implement the ISO extensions to SQL for PG querying. In a property graph nodes and edges are labeled and also have associated sets of property name/value pairs (e.g., a node labeled `Person` with property `Birthdate` having data value 11-11-2001; here `Person` and `Birthdate` are attribute metadata). While PGs offer a model

[1] Other more complex types of metadata, such as reflective or active data [5, 14], are beyond the scope of our current discussion.

closely aligned with conceptual domain representations, the model makes (1) a strict distinction between metadata and data, and (2) has no support for reification.

The vision. We present a vision for overcoming barriers to flexible management of data–metadata heterogeneity in property graph data management applications. A key challenge in this space is the misalignment between data and metadata—where elements such as labels, property keys, and values may shift roles across different graphs. This misalignment is a significant source of heterogeneity in property graph collections, complicating integration, querying, and inference. Existing property graph models, including those formalized in the ISO GQL standard, enforce a strict separation between data and metadata and lack mechanisms for treating metadata as queryable, first-class entities. To address this, the research investigates whether a data model and retrieval paradigm can be developed to support effective management and inference over heterogeneous collections of property graphs. This vision is structured around three core research questions:

RQ1: To what extent do existing property graph models handle heterogeneity and misalignment in data and metadata when integrating collections of property graphs, and where do they fall short?

RQ2: What extensions to the property graph data model are necessary to support flexible representation of heterogeneous graph structures and metadata?

RQ3: How can retrieval paradigms be designed to support effective inferences over heterogeneous collections of property graphs?

As a first step toward this, we introduced Meta-Property Graphs (MPG) [13], a fully backward-compatible extension of the PG model that addresses limitations in representing and querying metadata. Our approach enables first-class treatment of labels and properties as queryable objects, as well as reification of subgraphs. On this foundation, we further propose MetaGPML, a fully backward-compatible extension of the Graph Pattern Matching Language (GPML), the core language at the heart of the ISO standard GQL for PG querying. The next steps in this research will then focus on evaluating the expressiveness and practical utility of MPG and MetaGPML in real-world scenarios, such as knowledge graph integration and metadata-driven querying. Additionally, we will explore the design of retrieval paradigms that can leverage the enriched structure of MetaPG to support flexible inference across heterogeneous graph collections. These efforts aim to bridge the gap between theoretical foundations and practical deployment in modern graph data ecosystems.

2 RELATED WORK

There is a long-standing debate regarding the treatment of data and metadata in data management and modeling. In the research literature, dealing with the challenges of data-metadata heterogeneity was studied in the context of relational data integration and data integration on the web, leading to solutions for relational and XML data-metadata mapping and exchange (e.g., [4, 7, 11]). When it comes to graph data management, metadata can be interpreted into different types and classes. The most common and probably the first-type of metadata is particularly properties and labels in graph data models. At one extreme, we have data models such as RDF and RDF-star[2], along with tuple normal forms, that treat this type of metadata like any other form of data. This methodology proves advantageous in scenarios where uniformity and consistency in data representation are paramount. Conversely, graph normal forms and Labeled Property Graphs (LPG) [3] adopt a different perspective, treating properties and labels as metadata alongside the actual data. This approach helps to create a representation that more closely mirrors the real-world structure of the data. It also enhances the flexibility of the data model, especially in identifying different instances of relationships. Unprincipled solutions, such as manipulating database catalogs, offer ad-hoc ways to achieve similar flexibility but often lack the theoretical foundation and consistency of more formal approaches.

Addressing the second type of meta-properties, namely reification, which is not inherently accommodated within the property graph data model, presents several alternative approaches. One option is to map the data to an alternative data model inherently equipped to support reification, such as through a direct or customized mapping of the property graph to RDF or a relational data model. However, this approach introduces complexity, and there is always the risk of losing some metadata during the mapping process. Another solution involves elevating the model to a more generalized framework, such as [2, 10, 12], which presents another way of managing this type of metadata. Nonetheless, the most seemingly effective approach involves addressing the reification issue directly within the property graph data model, focusing on resolving the reification challenge internally.

3 EXTENDING PROPERTY GRAPH

As the first step of my PhD, I introduced the foundation of Meta-Property Graph, in collaboration with my supervisors [13], with the goal of breaking down the barrier between data and metadata in the property graph model while maintaining full backward compatibility with the GQL standard. Meta-Property Graph enables first-class treatment of label sets and properties and introduces the ability to reify graph sub-structure (combination of nodes, edges, properties, and label sets), supporting more expressive and complex data analytics.

3.1 Meta-Property Graph data model

The formal formulation of the Meta-Property Graph (MPG) data model is as follows. This formulation follows the foundational principles of the property graph data model. For $\mathcal{I}$, $\mathcal{L}$, $\mathcal{K}$, and $\mathcal{V}$ as pairwise disjoint sets of object identifiers, labels, property keys, and property values, respectively we have:

Definition 1. *A meta-property graph is a directed and undirected vertex- and edge-labeled graph $G = (N, E, P, L, \lambda, \mu, \sigma, \upsilon, \eta, \rho)$, where:*

- *$N, E, P, L \subseteq \mathcal{I}$ are finite, pairwise disjoint sets,*
- *$\mu : L \to 2^{\mathcal{L}}$ assigns a finite set of labels to each label set identifier,*
- *$\lambda : N \cup E \to L$ is a bijective labeling function assigning a label set identifier to each node and edge,*

[2] https://www.w3.org/groups/wg/rdf-star

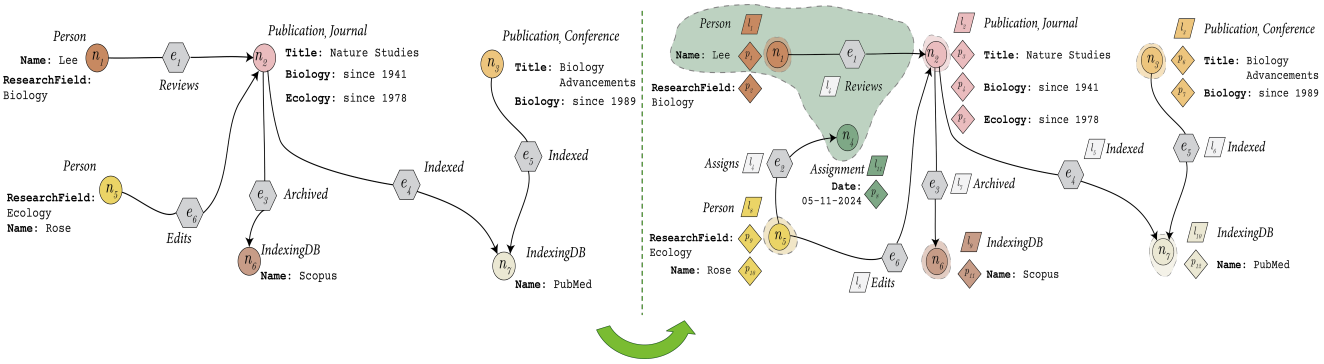

Figure 1: Property Graph vs Meta-Property Graph data modeling in an example

- $\upsilon : P \longrightarrow \mathcal{K} \times \mathcal{V}$ *assigns key-value pairs to properties,*
- $\sigma : N \cup E \longrightarrow \mathbb{C}$ *assigns compatible property sets to nodes and edges, such that for each pair of distinct objects* $o_1, o_2 \in N \cup E$*, it holds that* $\sigma(o_1) \cap \sigma(o_2) = \varnothing$ *and* $\bigcup_{o \in N \cup E} \sigma(o) = P$*, i.e., every and each property* $p \in P$ *is assigned to exactly one node or edge.*
- $\eta = (\eta_s, \eta_t, \eta_u)$ *where:*
  - $\eta_s, \eta_t : E_d \longrightarrow N$ *assign source and target nodes to directed edges,*
  - $\eta_u : E_u \longrightarrow \{\{u, v\} \mid u, v \in N\}$ *assigns node pairs to undirected edges,*
- $\rho : N \longrightarrow 2^{N \cup E \cup P \cup L}$ *assigns finite sets of objects to nodes, such that for each* $n \in N$*, it holds that* $n \notin \rho^*(n)$*, where* $\rho^*(n)$ *is the closure of* $\rho(n)$*,*[3] *ensuring that the sub-structure associated with each node is well-founded.*

where $E = E_d \cup E_u$*, and the set of compatible property sets* $\mathbb{C} = \{C \subseteq P \mid \forall p_1, p_2 \in C, p_1 \neq p_2 \Rightarrow Key(p_1) \neq Key(p_2)\}$*,* $Key(p) = \pi_1(\upsilon(p))$*,* $Val(p) = \pi_2(\upsilon(p))$*.*

For better understanding of the data model, figure 1 illustrates a sample MPG database of publications, indexing databases, and persons. Unlike standard PGs, our model treats four types of data objects as first-class citizens: edges (E_G), nodes (N_G), properties (P_G), and label sets (L_G). This is enabled by assigning identifiers not only to the nodes and edges but also to properties and label sets.

3.2 MetaGPML in practice

To facilitated the use of the new capabilities Meta-Property Graph, a proposal is provided to extend the property graph pattern matching language (GPML) [6, 8, 9] that we call MetaGPML. MetaGPML ensure backward compatibility in the sense that every GPML query is a MetaGPML query. Table 1 introduces some of the basic pattern notations of the MetaGPML for nodes and edges in MPG.

[3]Formally, $\rho^0(n) = \rho(n)$, $\rho^i(n) = \rho^{i-1}(n) \cup \bigcup_{n' \in N \cap \rho^{i-1}(n)} \rho(n')$, and $\rho^*(n) = \bigcup_{i=0}^{\infty} \rho^i(n)$.

Table 1: Data objects pattern notation

Nodes	
`(x:l)`	node variable (nv) x with label l
`(x:l).z`	nv x with label l and property variable z
`(x::`π`)`	nv x with a pattern π in its reified sub-structure
Edges	
`-[x:l]->`	edge variable x with label l
`-[x:?y]->`	edge variable x with label set variable y
`-[x].z->`	edge variable x with property variable z

The following two queries are presented to show case how querying a meta-property graph using MetaGPML works in practice. Q_1 aims to match a publication's research fields, which is being stored as metadata or property key in this graph structure of the meta-property graph graph in Figure 1, with potential reviewers' expertise through their `ResearchField` property.

Q_1: Finding reviewers based on research fields

```
MATCH (x:Person), (y:Publication).z
WHERE x.ResearchField = KEY(z)
RETURN x.Name AS "Reviewer candidate",
       y.Name AS "Publication venue",
       KEY(z) AS "Research field"
```

Result table of Q_1

Reviewer candidate	Publication venue	Research field
Lee	Nature Studies	Biology
Lee	Biology advancement	Biology
Rose	Nature Studies	Ecology

In Figure 1, the meta-property graph represents the statement "Rose assigned Lee as a reviewer on 5th November 2024" by reifying the review relationship into a node. Query Q_2 retrieves the assigning editor's name through this reified sub-structure.

Q_2: Who assigned Lee as a reviewer and when?

```
MATCH   (x:Person)-[:assigns]->
        (y::(z:Person)-[:reviews]->())
WHERE   z.Name = "Lee"
RETURN   z.Name AS "reviewer name",
         y.Date AS "Date",
         x.Name AS "Assigning editor"
```

Note that in the MATCH clause we have a graph pattern embedded in a node pattern, to denote a query to be executed on the sub-structure reified by the node which is bound to variable y.

Result table of Q_2

Reviewer name	Date	Assigning editor
Lee	05-11-2024	Rose

4 FUTURE RESEARCH VISION

In this PhD research the focus is on underscoring the critical need to dissolve the rigid boundary between data and metadata in property graph management. As an initial step toward this vision, we introduced a fully specified data model and query language for meta-property graphs, enabling seamless modeling and integration of data and metadata. While this work establishes a solid foundation for more flexible and expressive property graph management in contemporary applications, several key research directions remain to be explored in order to fully realize this vision.

(1) Physical implementation and technical challenges. A key challenge is implementing the MPG data model efficiently. Since label sets and properties are treated as data objects with identifiers, they may need dedicated storage and indexing strategies. Alternatively, approaches like concatenated IDs could maintain existing storage structures but may impact query evaluation. Research is needed on physical representations and indexing strategies that optimize MPG performance.

(2) Meta-Property Graphs in practice. It's important to further investigate how Meta-Property Graph can enhance knowledge engineering and management in practice. Understanding how metadata awareness and sub-structure reification can contribute to improving tasks such as auditing and human-in-the-loop validation of knowledge graphs or data cleaning, wrangling, integration, and exchange is crucial. Furthermore, studies should be conducted on how the capabilities that MPG introduce, such as subgraph annotation and querying different forms of metadata, can enhance knowledge reasoning and facilitate advanced analysis within knowledge graphs. Additionally, developing effective educational approaches and training resources for students and professionals working with MPGs and MetaGPML requires further study.

(3) Improvements and integration. Meta-Property Graph and MetaGPML can be enhanced through: (1) extending MetaGPML with additional functions to leverage better metadata awareness and also including other currently existing abilities such as paths and repetition which we did not include in this proposed vision of MetaGPML for the sake of simplicity, (2) developing schema and constraint languages for MPG building on PG-SCHEMA [1], and (3) incorporating other forms of metadata such as reflection to expand the metadata awareness in MPG.

ACKNOWLEDGMENTS

This work was supported by the MATTER-TKI-HTSM/22.0024 Research grant.

REFERENCES

[1] Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savkovic, Michael Schmidt, Juan Sequeda, Slawek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoc, Mingxi Wu, and Dusan Zivkovic. 2023. PG-Schema: Schemas for Property Graphs. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 198:1–198:25.

[2] Renzo Angles, Aidan Hogan, Ora Lassila, Carlos Rojas, Daniel Schwabe, Pedro Szekely, and Domagoj Vrgoč. 2022. Multilayer Graphs: A Unified Data Model for Graph Databases. In *Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. Association for Computing Machinery, Article 11, 6 pages.

[3] Renzo Angles, Harsh Thakkar, and Dominik Tomaszuk. 2019. RDF and property graphs interoperability: Status and issues. In *Proceedings of the 13th Alberto Mendelzon International Workshop on Foundations of Data Management*. Asunción, Paraguay.

[4] Angela Bonifati, Elaine Chang, Terence Ho, Laks V. S. Lakshmanan, Rachel Pottinger, and Yongik Chung. 2010. Schema mapping and query translation in heterogeneous P2P XML databases. *The VLDB Journal* 19, 2 (2010), 231–256. https://doi.org/10.1007/s00778-009-0159-9

[5] Jan Van den Bussche, Dirk Van Gucht, and Stijn Vansummeren. 2007. A crash course on database queries. In *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM, 143–154.

[6] Alin Deutsch, Nadime Francis, Alastair Green, Keith Hare, Bei Li, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Wim Martens, Jan Michels, Filip Murlak, Stefan Plantikow, Petra Selmer, Oskar van Rest, Hannes Voigt, Domagoj Vrgoč, Mingxi Wu, and Fred Zemke. 2022. Graph Pattern Matching in GQL and SQL/PGQ. In *Proceedings of the 2022 International Conference on Management of Data* (New York, NY, USA) *(SIGMOD '22)*. Association for Computing Machinery, 2246–2258. https://doi.org/10.1145/3514221.3526057

[7] George Fletcher and Catharine Wyss. 2009. Towards a General Framework for Effective Solutions to the Data Mapping Problem. *Journal on Data Semantics XIV* (2009), 37–73.

[8] Nadime Francis, Amelie Gheerbrant, Paolo Guagliardo, Libkin Leonid, Victor Marsault, Wim Martens, Filip Murlak, Liat Peterfreund, Alexandra Rogova, and Domagoj Vrgoč. 2023. A Researcher's Digest of GQL. In *26th International Conference on Database Theory (ICDT 2023)* (Ioannina, Greece), Vol. 255. https://doi.org/10.4230/LIPIcs.ICDT.2023.1

[9] Nadime Francis, Amélie Gheerbrant, Paolo Guagliardo, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Liat Peterfreund, Alexandra Rogova, and Domagoj Vrgoc. 2023. GPC: A Pattern Calculus for Property Graphs. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (New York, NY, USA) *(PODS '23)*. Association for Computing Machinery, 241–250. https://doi.org/10.1145/3584372.3588662

[10] Ewout Gelling, George Fletcher, and Michael Schmidt. 2024. Statement Graphs: Unifying the Graph Data Model Landscape. In *Database Systems for Advanced Applications - 29th International Conference, DASFAA 2024*. 364–376.

[11] Mauricio A. Hernández, Paolo Papotti, and Wang-Chiew Tan. 2008. Data exchange with data-metadata translations. *Proceedings of the VLDB Endowment* 1, 1 (2008), 260–273. https://doi.org/10.14778/1453856.1453888

[12] O. Lassila, M. Schmidt, B. Bebee, D. Bechberger, W. Broekema, A. Khandelwal, K. Lawrence, R. Sharda, and B. Thompson. 2023. The OneGraph Vision: Challenges of Breaking the Graph Model Lock-In. *Semantic Web Journal* 14 (2023). Issue 1.

[13] Sepehr Sadoughi, Nikolay Yakovets, and George HL Fletcher. 2025. Breaking Down the Data-Metadata Barrier for Effective Property Graph Data Management. In *Proceedings of the 28th International Conference on Extending Database Technology, EDBT 2025*. 978.

[14] Divesh Srivastava and Yannis Velegrakis. 2007. Intensional associations between data and metadata. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 401–412.