Schema-Evolution von Graphdatenbanken in ProSA

Antrittsvortrag Bachelorarbeit

Timo Hanöffner
Lehrstuhl für Medieninformatik
FAKULTÄT FÜR INFORMATIK UND DATA SCIENCE

Universität Regensburg

Vorstellung

- Timo Hanöffner
- 7. Semester Medieninformatik B.A., Medienwissenschaft (2. HF)

Betreuer & Gutachter

- Dr.-Ing. Tanja Auge; Dominique Hausler (Betreuerin)
- Prof. Dr. Niels Henze (Erstgutachter)
- Prof. Dr.-Ing. habil. Meike Klettke (Zweitgutachterin)

ProSA

- System zur Unterstützung der Reproduzierbarkeit von Datenbankauswertungen[1]
- Rekonstruierung von verlorengegangenen Duplikaten oder Dangling Tuples mit Hilfe von Provenance [1]
- Anfragen aktuell nur auf relationale Datenbanken

→ **Ziel: Benutzung von Graphdatenbanken in ProSA**

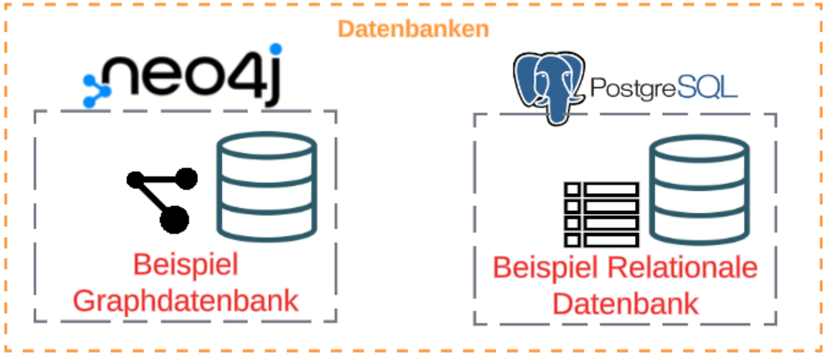
[1] Auge, T. (2023). Prosa: A provenance system for reproducing query results. In Companion proceedings of the acm webconference 2023 (S.1555–1558). NewYork,NY,USA: Association for Computing Machinery. Zugriff auf https://doi.org/10.1145/3543873.3587563 doi:10.1145/3543873.3587563

Eigener Ansatz

Anlegen von Beispieldatenbanken

- Erstellung einer Graphdatenbank in Neo4j
- Erstellung einer relationalen Datenbank in PostgreSQL

Schema-Extraktion

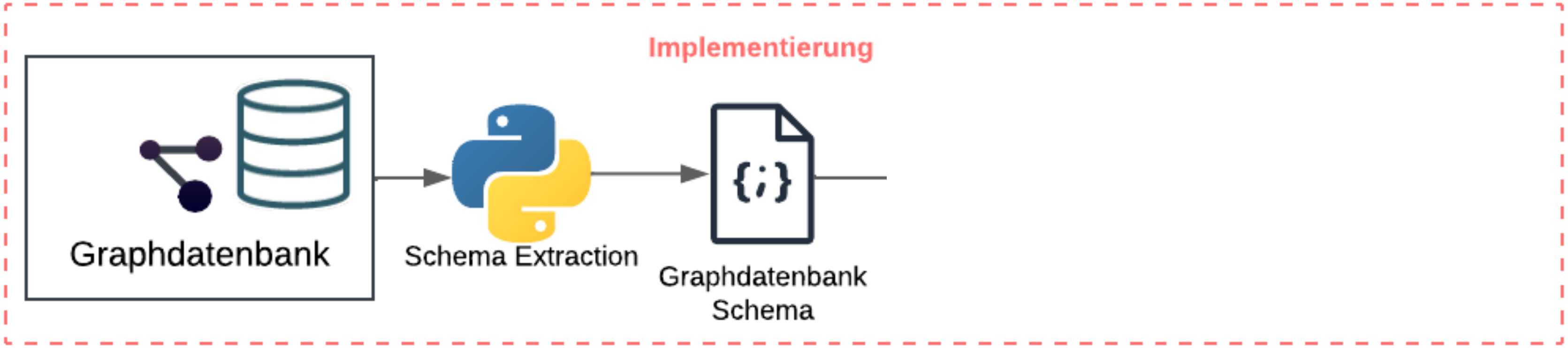


Eigener Ansatz

Anlegen von Beispieldatenbanken

Schema-Extraktion

- Extraktion des Schemas aus der Graphdatenbank mit Python
- Analyse der extrahierten Struktur für das Mapping



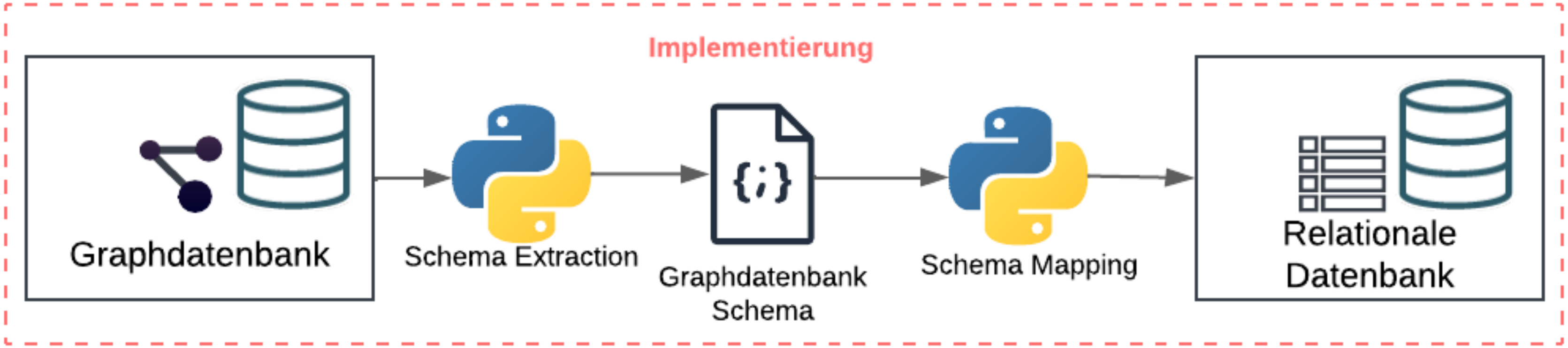
Timo Hanöffner
Lehrstuhl für Medieninformatik
FAKULTÄT FÜR INFORMATIK UND DATA SCIENCE

Eigener Ansatz

Mapping zwischen Datenbanken

- Überführung des extrahierten Schemas von der Graphdatenbank auf die relationale Datenbank

Evaluation und Vergleich



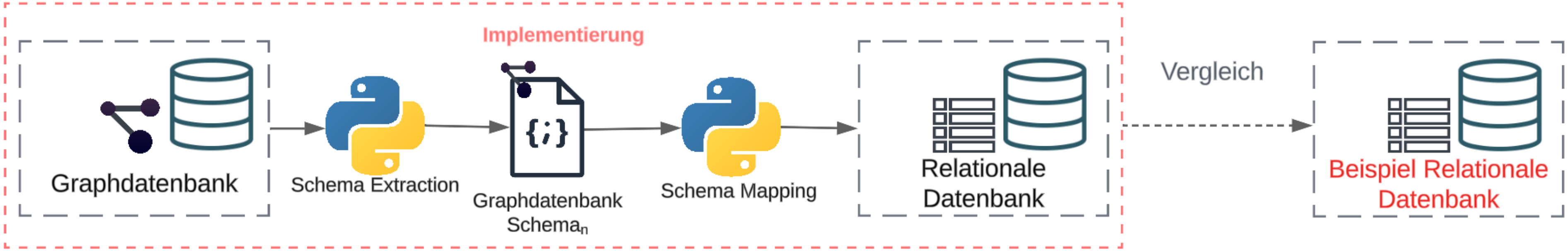
Timo Hanöffner
Lehrstuhl für Medieninformatik
FAKULTÄT FÜR INFORMATIK UND DATA SCIENCE

Eigener Ansatz

Mapping zwischen Datenbanken

Evaluation und Vergleich

- Abgleich der gemappten Datenbank mit der Beispieldatenbank

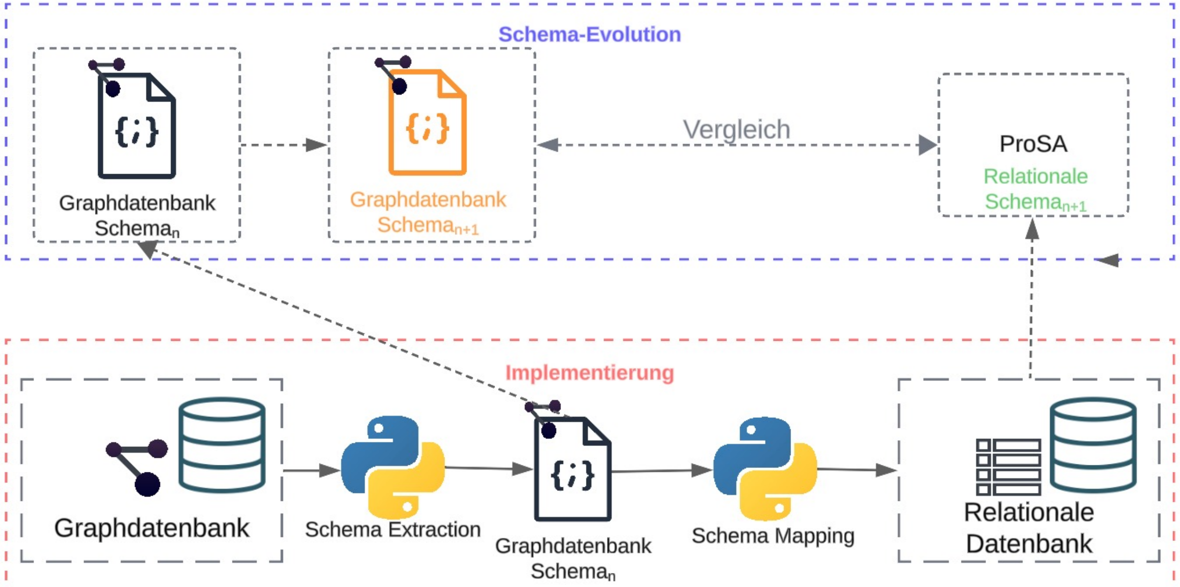


Eigener Ansatz

Schema-Evolution in ProSA

- Änderungen am Schema, ohne die Funktionalität der Datenbank zu beeinträchtigen

Rücktransformation auf Graphdatenbank

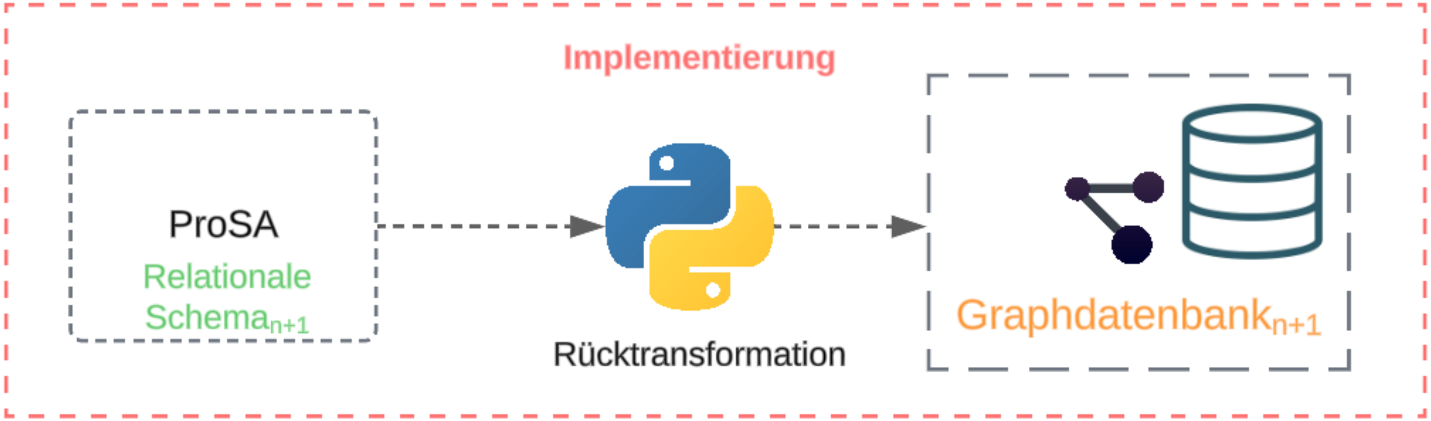


Eigener Ansatz

Schema-Evolution in ProSA

Rücktransformation auf Graphdatenbank

- Zurückübertragung der relationalen Datenbank auf die Graphdatenbank



Verwandte Arbeiten

An Approach for Schema Extraction of NoSQL Graph Databases

[2]

Angelo Augusto Frozza
Instituto Federal Catarinense (IFC), Santa Catarina, Brazil
E-mail: angelo.frozza@ifc.edu.br

Salomão Rodrigues Jacinto, Ronaldo dos Santos Mello
Universidade Federal de Santa Catarina (UFSC)
Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Santa Catarina, Brazil
E-mail: maorodriguesj@gmail.com, r.mello@ufsc.br

Abstract

Currently, a large volume of heterogeneous data is generated and consumed by several classes of applications, which raise a new family of database models called NoSQL. NoSQL graph databases is a member of this family. They provide high scalability and are schemaless, i.e., they do not require an implicit schema such as relational databases. However, the knowledge of how data is structured may be of great importance for data integration or data analysis processes. There are some works in the literature that extract the schema from graph structures or graph-based data sources. Different from them, this work proposes a comprehensive approach that consider all the common NoSQL database graph data model concepts, and generates a schema in the recent JSON Schema recommendation. Experimental evaluations show that our solution generates a suitable schema representation with a linear complexity.

1. Introduction

NoSQL databases is a relatively new family of database management system (DBMS) suitable to *Big Data* management as they present flexible data models as well as high availability and scalability [8, 12]. *Schemaless* is another usual feature of them. Most of the NoSQL products do not obligate a schema to be defined *a priori* to a database, as relational databases do. They argue that this feature avoids a rigid data structure control, which would impose a management overhead to the DBMS.

Nevertheless, to be schema-aware is important to several data management tasks, like data integration, data interoperability, as well as query processing and optimization. One example is the recent data lake management system [9]. A data lake is a huge repository of heterogeneous data sources that must frequently be integrated in order to answer a data analytics demand. This task is hard to be accomplished without a metadata management, which requires a schema extraction or a reverse engineering process [4].

978-1-7281-1054-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IRI49571.2020.00046

2017 IEEE International Conference on Big Data (BIGDATA)

Model Driven Reverse Engineering of NoSQL Property Graph Databases

The case of Neo4j

[3]

Isabelle Comyn-Wattiau
ESSEC Business School
Cergy, France
wattiau@essec.edu

JackyAkoka
CEDRIC-CNAM & IMT-TEM
Paris, France
jacky.akoka@lecnam.net

***Abstract*— Most NoSQL databases are schemaless. Although they offer some flexibility, they do not have any knowledge of the database schema, losing the benefits provided by these schemas. It is generally accepted that data modelling can have an impact**

Starting from a Neo4j code and using transformation rules, we derive a logical graph model and a conceptual Extended Entity-Relationship (EER) schema.

The rest of the paper is organized as follows. In Section 2 present a state of the art on modeling NoSQL databases, h forward and reverse engineering. Section 3 describes our del-driven approach (MDA) dedicated to reverse gineering of NoSQL property graph databases. The approach l the transformations rules are illustrated in Section 4. We o present and discuss some evaluation criteria related to the lity of the models obtained. Finally, Section 5 presents ne conclusions as well as some perspectives in terms of ure research.

II. STATE OF THE ART

Modeling NoSQL databases can be performed using either orward or a reverse engineering methodology. Modeling SQL databases using forward engineering consists in going m a conceptual to a physical data model. Several ntributions have been made in modeling document NoSQL abases [1,2]. Some authors propose transformations proach for column store databases [3,4]. As far as key-value abases is concerned, [5] discusses design issues related to -value data stores. Let us also mention the design process big data warehouses proposed by [6]. As for graph abases, [7] has presented an UMLtoGraphDB framework. present an approach transforming relational and RDF dels to a property graph. Let us mention some contributions XML database schema generation and extraction [9]. A guage for declaring schema evolution operations for ensible record stores and for NoSQL document stores can found in [10]. Finally, it is worth mentioning contributions ated to the transformation of multidimensional data rehouses [11,12].

Reverse engineering is the process of translating the source le of an application into a conceptual model. It allows the citation of the missing semantic schemas, recovering the nplete specification of a database. Many publications have wn how to generate rich conceptual schemas from instances]. But most of them are not related to NoSQL databases]. Unlike NoSQL forward engineering, there is little earch on NoSQL reverse engineering. [15] present a model-

[4]

Artifacts Available acm V1.1

DiscoPG: Property Graph Schema Discovery and Exploration

Angela Bonifati
Lyon 1 University & LIRIS CNRS
angela.bonifati@univ-lyon1.fr

Stefania Dumbrava
ENSIIE & Inst. Polytech. de Paris
stefania.dumbrava@ensiie.fr

Emile Martinez
ENS Lyon
emile.martinez@ens-lyon.fr

Fatemeh Ghasemi
ENS Lyon
fatemeh.ghasemi@ens-lyon.fr

Malo Jaffré
ENS Lyon
malo.jaffre@ens-lyon.fr

Pacôme Luton
ENS Lyon
pacome.luton@ens-lyon.fr

Thomas Pickles
ENS Lyon
thomas.pickles@ens-lyon.fr

ABSTRACT

Property graphs are becoming pervasive in a variety of graph processing applications using interconnected data. They allow to encode multi-labeled nodes and edges, as well as their properties, represented as key/value pairs. Although property graphs are widely used in several open-source and commercial graph databases, they lack a schema definition, unlike their relational counterparts. The property graph schema discovery problem consists of extracting the underlying schema concepts and types from such graph datasets. We showcase DiscoPG, a system for efficiently and accurately discovering and exploring property graph schemas. To this end, it leverages hierarchical clustering using a Gaussian Mixture Model, which accounts for both node labels and properties. DiscoPG allows users to perform schema discovery for both static and dynamic graph datasets. Suitable visualization layouts and dedicated dashboards enable the user perception of the static and dynamic inferred schema on the node clusters, as well as the differences in runtimes and clustering quality. To the best of our knowledge, DiscoPG is the *first system to tackle the property graph schema discovery problem*. As such, it supports the insightful exploration of the graph schema components and their evolving behavior, while revealing the underpinnings of the clustering-based discovery process.

PVLDB Reference Format:
Angela Bonifati, Stefania Dumbrava, Emile Martinez, Fatemeh Ghasemi, Malo Jaffré, Pacôme Luton, and Thomas Pickles. DiscoPG: Property Graph Schema Discovery and Exploration. PVLDB, 15(12): 3654–3657, 2022.
doi:10.14778/3554821.3554867

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/PI-Clustering/code.

1 INTRODUCTION

Graphs are natural abstractions for representing interconnected data. They have been adopted in a large array of applications, ranging from social networks to scientific datasets, fraud detection, recommendation systems, and the Semantic Web. Their most expressive underlying data model is the *property graph* one, along with its numerous variants [1, 3, 10], in which lists of properties are attached to both the nodes and the edges of a directed, labeled, multi-graph. Property graphs underlie several open-source and commercial systems, out of which graph databases are the most prominent. While these do not impose a priori rigid schema constraints, the rich property graph model enables them to compactly capture complex patterns, while remaining digestible and self-explainable for end users [9]. However, the lack of a schema hinders several applications, such as data exploration, query formulation and optimization, meta-data management, and the extraction of type-based graph features for various machine learning pipelines. Moreover, property graph schemas are considered as relevant abstractions needed in future graph processing systems [10].

To address this, in DiscoPG, we tackle the discovery of property graph schemas, leveraging a *hierarchical clustering* algorithm, based on *Gaussian Mixture Models*. The hierarchical clustering method based on GMM has been shown [2] to be tailored to the complex property graph data model. Compared to previous schema discovery methods [7], DiscoPG allows to account for both labeling and property information and to build node clusters, reflecting the underlying typing hierarchy of the base dataset. Ours is a purely statistical approach (presented in detail in [2]), which we have experimentally shown to provide better accuracy and performance than previous analytical methods [7]. While schema discovery methods have been proposed for RDF datasets [4], notably based on distributed clustering using Spark, our method is centralized and supports the more expressive property graph model.

DiscoPG allows users to engage in the following scenarios:

- visualization and exploration of the discovered schema, with novel cluster-centered layouts that showcase the size of the discovered node types, their corresponding labels and properties, as well as those of their connecting edges.
- interactive inspection of a schema dashboard, displaying the performance of the underlying hierarchical clustering algorithms and the quality of the obtained schema, across each iteration of the discovery process; this allows analyzing the runtime behavior of the various methods and underpinnings of the graph clustering process.
- analysis of the property graph schema evolution, with custom rendering of the parts affected by modifications to the base graph instance. Both incremental and recomputation-based schema evolution scenarios are considered.

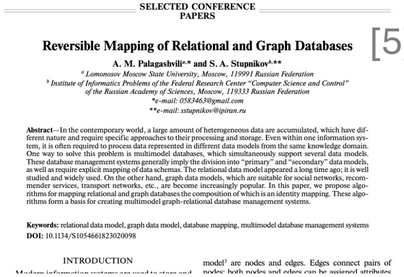
We showcase DiscoPG on both real-world and benchmarking datasets, e.g., CovidGraph [6] and LDBC [5]. These have different characteristics and range from simple to complex schemas. The implementation of DiscoPG is available on Github.

Proceedings of the VLDB Endowment, Vol. 15, No. 12 ISSN 2150-8097.
doi:10.14778/3554821.3554867

- Ansätze zur Schema Extraktion von Graphdatenbanken [2][3][4]
- Wichtig: Berücksichtigung von Multilabeling (Knoten mit mehr als einem Label) [2]

[2]Frozza,A.A.,Jacinto,S.&Mello,R. (2020,08). An approach for schema extraction of nosql graph databases. In(S. 271-278). doi: 10.1109/IRI49571.2020.00046
[3]Comyn-Wattiau, I. & Akoka, J. (2017). Model driven reverse engineering of nosql property graph databases:
The case of neo4j. In 2017 ieee international conferenceon big data(big data)(S. 453-458).
doi: 10.1109/BigData.2017.8257957
[4] Angela Bonifati, Stefania Dumbrava, Emile Martinez, Fatemeh Ghasemi, Malo Jaffré, Pacôme Luton, and Thomas Pickles. 2022. DiscoPG: property graph schema discovery and exploration. Proc. VLDB Endow. 15, 12 (August 2022), 3654–3657. https://doi.org/10.14778/3554821.3554867

Verwandte Arbeiten



SELECTED CONFERENCE
PAPERS

Reversible Mapping of Relational and Graph Databases

[5]

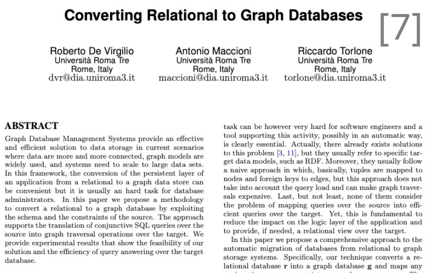
A. M. Palagashvili[a],* and S. A. Stupnikov[b],**
[a] Lomonosov Moscow State University, Moscow, 119991 Russian Federation
[b] Institute of Informatics Problems of the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119333 Russian Federation
**e-mail: 0583463@gmail.com*
***e-mail: sstupnikov@ipiran.ru*

Abstract—In the contemporary world, a large amount of heterogeneous data are accumulated, which have different nature and require specific approaches to their processing and storage. Even within one information system, it is often required to process data represented in different data models from the same knowledge domain. One way to solve this problem is multimodel databases, which simultaneously support several data models. These database management systems generally imply the division into "primary" and "secondary" data models, as well as require explicit mapping of data schemas. The relational data model appeared a long time ago; it is well studied and widely used. On the other hand, graph data models, which are suitable for social networks, recommender services, transport networks, etc., are become increasingly popular. In this paper, we propose algorithms for mapping relational and graph databases the composition of which is an identity mapping. These algorithms form a basis for creating multimodel graph-relational database management systems.

Keywords: relational data model, graph data model, database mapping, multimodel database management systems
DOI: 10.1134/S1054661823020098

INTRODUCTION

model[3] are nodes and edges. Edges connect pairs of



Converting Relational to Graph Databases

[7]

Roberto De Virgilio
Università Roma Tre
Rome, Italy
dvr@dia.uniroma3.it

Antonio Maccioni
Università Roma Tre
Rome, Italy
maccioni@dia.uniroma3.it

Riccardo Torlone
Università Roma Tre
Rome, Italy
torlone@dia.uniroma3.it

ABSTRACT

Graph Database Management Systems provide an effective and efficient solution to data storage in current scenarios where data are more and more connected, graph models are widely used, and systems need to scale to large data sets. In this framework, the conversion of the persistent layer of an application from a relational to a graph data store can be convenient but it is usually an hard task for database administrators. In this paper we propose a methodology to convert a relational to a graph database by exploiting the schema and the constraints of the source. The approach supports the translation of conjunctive SQL queries over the source into graph traversal operations over the target. We provide experimental results that show the feasibility of our solution and the efficiency of query answering over the target database.

task can be however very hard for software engineers and a tool supporting this activity, possibly in an automatic way, is clearly essential. Actually, there already exists solutions to this problem [3, 11], but they usually refer to specific target data models, such as RDF. Moreover, they usually follow a naive approach in which, basically, tuples are mapped to nodes and foreign keys to edges, but this approach does not take into account the query load and can make graph traversals expensive. Last, but not least, none of them consider the problem of mapping queries over the source into efficient queries over the target. Yet, this is fundamental to reduce the impact on the logic layer of the application and to provide, if needed, a relational view over the target.

In this paper we propose a comprehensive approach to the automatic migration of databases from relational to graph storage systems. Specifically, our technique converts a relational database **r** into a graph database **g** and maps any

Wojciech MUELLER, Przemysław IDZIASZEK, Sebastian KUJAWA, Mateusz ŁUKOMSKI, Przemysław NOWAK
Poznań University of Life Sciences, Institute of Biosystems Engineering, Poznań, Poland
e-mail: muellerw@up.poznan.pl

Received: 2018-04-05 ; Accepted: 2018-11-22

MAPPING OF RELATIONAL STRUCTURES IN GRAPH DATABASE NEO4J

[6]

Summary

Extension of functionality of most applications including the ones supporting agriculture, as a general rule requires an in-depth knowledge of relational structures creating databases, which can be sometimes difficult to achieve. It can result from the lack of complete technical documentation as well as relatively huge complexity of relational structures. The given publication is a continuation of the author's actions, aimed at creating a moderately universal application allowing to reproduce the existing relational structures created with the use of different relational database management systems (RDBMS), namely SQL Server, MySQL or Oracle into graph form on the level of Neo4j graph database. This form makes it possible to thoroughly recognize complex relational structures with the use of queries prepared in Cypher language in native client, which is made available from the level of the created application. During the construction process of the presented tool, technologies such as ADO.NET, graph database Neo4j together with available programming interface as well proper tables containing metadata were utilized.

***Key words**: mapping, relational structure, graph structure, database, neo4j*

MAPOWANIE STRUKTUR RELACYJNYCH W BAZIE GRAFOWEJ NEO4J

Streszczenie

Rozbudowa funkcjonalności większości aplikacji, w tym również wspomagających rolnictwo z reguły wymaga pełnej znajomości struktur relacyjnych tworzących bazy danych, co czasami może być trudne do osiągnięcia. Powodem może być brak pełnej dokumentacji technicznej oraz względnie duża złożoność struktur relacyjnych. Prezentowana publikacja, to kontynuacja działań autorów, zmierzająca do wytworzenia w miarę uniwersalnej aplikacji, pozwalającej na odwzorowanie istniejących struktur relacyjnych, powstałych przy wykorzystaniu różnych systemów bazodanowych SQL Server, MySQL oraz Oracle, do postaci grafowej na poziomie Neo4j. Ta postać umożliwia wygodne, dogłębne rozpoznawanie złożonej struktury relacyjnej za pomocą pytań konstruowanych w języku Cypher w natywnym programie klienckim udostępnianym z poziomu prezentowanej aplikacji. W procesie budowy prezentowanego narzędzia wykorzystano technologie ADO.NET, bazę grafową Neo4j wraz z dostępnym interfejsem programistycznym oraz odpowiednie tabele zawierające metadane.

***Słowa kluczowe**: mapowanie, struktura relacyjna, struktura grafowa, baza danych, neo4j*

1. Introduction and available technologies

Extension of functionality of applications supporting broadly defined agriculture, independent of the way of realization, as a general rule involves recognition and modification of relational structure of database, with which the given application works in tandem. The recognition of those structures without possessing documentations and with a large number of defined tables and relations between them, is often not an easy case. The existing tool, dedicated to specific systems of databases do make it possible to search relations between tables in a quick way. Making an endeavor to solve this problem was previously a subject matter of authors inquiries, which was expressed in publication [7]. Mentioned publication presents the mechanism of reproducing relational structures created on the level SQL Server together with defined relations into the graph form on the level of Neo4j [5]. This in turn made it possible to make use of Cypher language paying special attention to searching relations between tables, which are represented on the level of graph database with the use of directed edges [1]. The limitation of the proposed solution was narrowing to single SQL Server environment, resulting from the use of defined classes within the namespace *Microsoft.SqlServer.Management.Smo* [2]. The objects created on the above basis made it possible for us to recognize database structure managed by the given SQL Server, while delivering necessary metadata at the same time. They were related with database selected by user.

A new, alternative approach, proposed by us, having more universal character, utilizes the technology called ADO.NET by Microsoft, which contains in itself numerous techniques of access to databases controlled by different database management systems (DBMS) [3]. Depending on the offer by specific producer of DBMS, in total we can make use of ODBC driver, OLEDB technology or native classes for the given database environment during the process. In addition to the above, the three mentioned solutions were presented in chronological order. From the point of view of efficiency, the last solution is preferable to the two remaining ones on condition that it is available. In general within each of the signaled techniques we can realize both connection-oriented and non-connection-oriented model. *Connection* is the key object in both models, enabling communication with DBMS, created openly by programmer or generated without programmer [8]. It offers a number of methods, but the function from the point of view of receiving information about metadata is function called *GetSchema* [12].

Information about relational structures that has been obtained as well as openly defined relations in the form of foreign keys will constitute the basis in the process of creating graphs on the level Neo4j [6]. Querying the existing base mapping only relational structure can be solely realized in client application offered by the producer of Neo4j or through additional functionality, which is yet to be implemented in the information system being under construc-

Semantic Mapping Relational to Graph Model

[8]

[1,2]Dewi W. Wardani, [2]Josef Küng

[1]Informatics Department
Sebelas Maret University, Indonesia
[2]Institute for Application Oriented Knowledge Processing
Johannes Kepler University, Austria

Abstract— **Making data to be more connected is one of the goals of Semantic Technology. Therefore, relational data model as one of important data resource type, is needed to be mapped and converted to graph model. In this paper we focus in mapping and converting without semantically loss, by considering semantic abstraction of the real world, which has been ignored in some previous researches. As a graph schema model, it can be implemented in graph database or linked data in RDF/OWL format. This approach studies that relationship should be paid more attention in mapping and converting because, often be found a gap semantic abstraction during those processes. In our small experiment shows that our idea can map and convert relational model to graph model without semantically loss.**

***Keywords*— mapping; converting; relational model; graph model; graph schema; semantic; big data.**

I. INTRODUCTION

Semantic technology has been being discussed in many topics, in improving and supporting the current technologies since the latest decade. Some standards of this technology have been being implemented and progressed both in theory and practice. Since the researchers started the discussion, they have defined different point of view in this technology. In area of knowledge management research, recently this technology takes a big part not only practically in some domain [6-11], but also contributing in the core subject such in big data and new direction of data management [12-15].

In this paper we refer to the inventor of Web's idea, that one goal of Semantic technology is to create the web of data [16]. The data will be more equivalent with a thing, sometimes it's called the network of thing [17]. The idea is making data more connected, meaningful and understand able on the machine side. In both theory and practice, preparing data which ready to be used in semantic technology is the bottom block in this technology's stack. Basically, semantic technology uses graph graph-based model data [18-19] . The common and famous implementation is formed the data as linked data in RDF/XML [20-21]. Since 2003, W3C announced it as one of the standard of semantic technology.

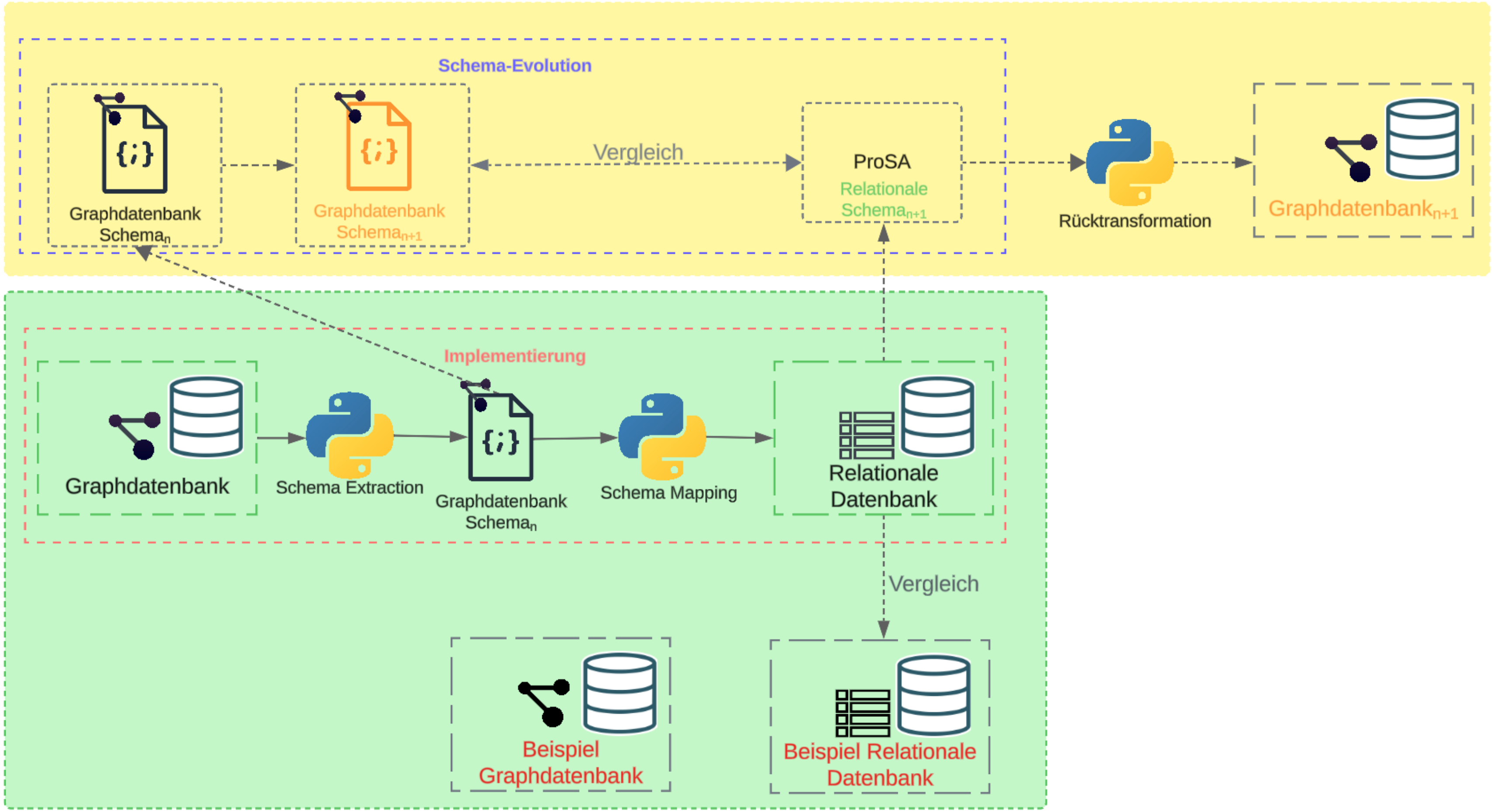
In reality, a lot of data exists in many domains, in structured or unstructured type, but the model is not semantic technology friendly. Specially, the structured data in relational model schema, it has huge and broad-area user and has been used successfully for long time [1]. It usually contains high quality information, in term that contains main information in almost all application. This type of data can't be ignored to be part as data source of semantic technology. Concerning to its importance, it's needed to map and convert this relational model to graph model, which is semantic technology friendly. Therefore, in reaching the goal web of data which is more connected and meaningful, some previous works tried to map relational model directly to RDF linked data [2-5]. Even, W3C released the features of direct mapping relational model to RDF data model. It's really good way and framework, but in our opinion, one problem of direct mapping is the process often loss the semantic of the real world problem from relational model. Some other works, convert relational database model to graph database model. This approach usually uses naive converting process, which put all tuple in each relation as a node and the foreign key as an edge between two nodes, but not so many scientific papers are written for this approach. Almost all ended their work until relational database is converted in graph database and do not concern possibility to be extended as semantic technology's need. As far as our knowledge our work is the first which create property relationship in the result of the mapping and converting process and there were no previous works which consider relationship as an important in graph schema. The opposite, in our work we think relationship will be an important part in the network of data, and mining the relationship will be important in the near future.

Here, we propose mapping relational model to graph model which try to keep the semantic abstraction from the real world of relational model and also satisfied for the need of semantic technology. This approach can be said as the process to create semantic schema in graph model, which can be implemented both in graph database or linked data in RDF format. In this work we implemented it in graph database, but the schema will be easier to be mapped to linked data in RDF format because it uses property graph model. Once the data is

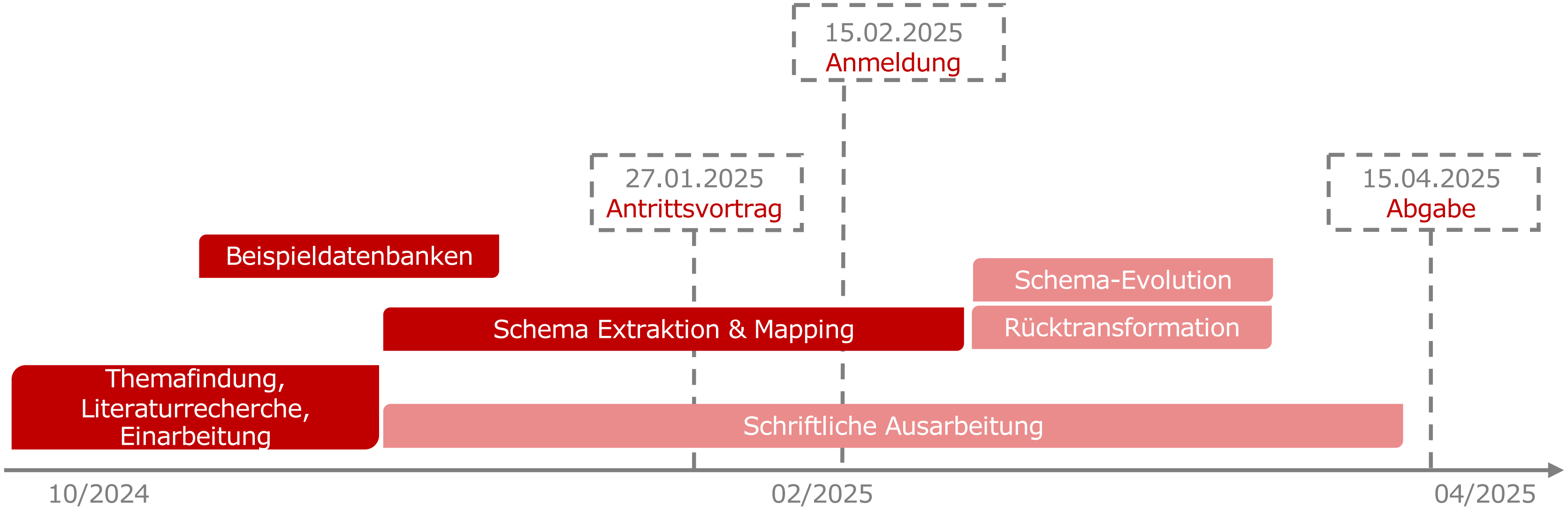
- Ansätze zum Mapping von Graphdatenbanken auf Relationalen Datenbanken [5][6][7][8]
- Keine Berücksichtigung von Multilabels

[5]A. M. Palagashvili and S. A. Stupnikov. 2023. Reversible Mapping of Relational and Graph Databases. Pattern Recognit. Image Anal. 33, 2 (Jun 2023), 113–121. https://doi.org/10.1134/S1054661823020098
[6] Mueller, W., Idziaszek, P., Kujawa, S., Łukomski, M., & Nowak, P. (2018). Mapping of relational structures in graph database Neo4j. Journal of Research and Applications in Agricultural Engineering, 63(4), 121-124
[7] Roberto De Virgilio, Antonio Maccioni, and Riccardo Torlone. 2013. Converting relational to graph databases. In First International Workshop on Graph Data Management Experiences and Systems (GRADES '13). Association for Computing Machinery, New York, NY, USA, Article 1, 1–6. https://doi.org/10.1145/2484425.2484426
[8] D. W. Wardani and J. Küng, "Semantic mapping relational to graph model," 2014 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Bandung, Indonesia, 2014, pp. 160-165, doi: 10.1109/IC3INA.2014.7042620.

Eigener Ansatz



Zeitplan u. Fragen



Zusammenfassung

Thema:

Benutzung von ProSA für Graphdatenbanken

Methode:

1. Extrahieren des Schemas
2. Mapping auf eine relationale Datenbank
3. Schema-Evolution in ProSA + Rücktransformation

Nächsten Schritte:

Anmeldung der Arbeit

Fine-Tuning und Testen von Extraktions- und Mapping-Algorithmen

Schema-Evolution

Rücktransformation

Schreiben der Arbeit

Quellen

[1] Auge, T. (2023). Prosa: A provenance system for reproducing query results. In Companion proceedings of the acm webconference 2023 (S.1555–1558). NewYork,NY,USA: Association for Computing Machinery. Zugriff auf https://doi.org/10.1145/3543873.3587563 doi:10.1145/3543873.3587563

[2]Frozza,A.A.,Jacinto,S.&Mello,R. (2020,08). An approach for schema extraction of nosql graph databases. In(S. 271-278). doi: 10.1109/IRI49571.2020.00046

[3]Comyn-Wattiau, I. & Akoka, J. (2017). Model driven reverse engineering of nosql property graph databases: The case of neo4j. In 2017 ieee international conferenceon big data(big data)(S. 453-458). doi: 10.1109/BigData.2017.8257957

[4] Angela Bonifati, Stefania Dumbrava, Emile Martinez, Fatemeh Ghasemi, Malo Jaffré, Pacôme Luton, and Thomas Pickles. 2022. DiscoPG: property graph schema discovery and exploration. Proc. VLDB Endow. 15, 12 (August 2022), 3654–3657. https://doi.org/10.14778/3554821.3554867

[5]A. M. Palagashvili and S. A. Stupnikov. 2023. Reversible Mapping of Relational and Graph Databases. Pattern Recognit. Image Anal. 33, 2 (Jun 2023), 113–121. https://doi.org/10.1134/S1054661823020098

[6] Mueller, W., Idziaszek, P., Kujawa, S., Łukomski, M., & Nowak, P. (2018). Mapping of relational structures in graph database Neo4j. Journal of Research and Applications in Agricultural Engineering, 63(4), 121-124

[7] Roberto De Virgilio, Antonio Maccioni, and Riccardo Torlone. 2013. Converting relational to graph databases. In First International Workshop on Graph Data Management Experiences and Systems (GRADES '13). Association for Computing Machinery, New York, NY, USA, Article 1, 1–6. https://doi.org/10.1145/2484425.2484426

[8] D. W. Wardani and J. Kiing, "Semantic mapping relational to graph model," 2014 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Bandung, Indonesia, 2014, pp. 160-165, doi: 10.1109/IC3INA.2014.7042620.