Katrin Braunschweig, Maik Thiele, Wolfgang Lehner

A Flexible Graph-Based Data Model Supporting Incremental Schema Design and Evolution

Erstveröffentlichung in / First published in:

Current Trends in Web Engineering: Workshops, Doctoral Symposium, and Tutorials. Paphos, 20.-21.06.2011. Springer, S. 302–306. ISBN 978-3-642-27997-3.

DOI: http://dx.doi.org/10.1007/978-3-642-27997-3_29

Final edited form was published in "Current Trends in Web Engineering: Workshops, Doctoral Symposium, and Tutorials. Paphos 2011", S. 302–306, ISBN 978-3-642-27997-3
http://dx.doi.org/10.1007/978-3-642-27997-3_29

A Flexible Graph-Based Data Model Supporting Incremental Schema Design and Evolution

Katrin Braunschweig, Maik Thiele, and Wolfgang Lehner

Database Technology Group, Faculty of Computer Science,
Technische Universität Dresden,
01062 Dresden, Germany
{katrin.braunschweig,maik.thiele,wolfgang.lehner}@tu-dresden.de

Abstract. Web data is characterized by a great structural diversity as well as frequent changes, which poses a great challenge for web applications based on that data. We want to address this problem by developing a schema-optional and flexible data model that supports the integration of heterogenous and volatile web data. Therefore, we want to rely on graph-based models that allow to incrementally extend the schema by various information and constraints. Inspired by the on-going web 2.0 trend, we want users to participate in the design and management of the schema. By incrementally adding structural information, users can enhance the schema to meet their very specific requirements.

Keywords: data integration, schema flexibility, schema evolution, web data, graph theory.

1 Introduction

Recent years have seen a rise in data-driven technologies and applications on the web. Data on a wide range of topics is made publicly available following the trend towards open data. This data is inherently heterogenous in its structure and subject to frequent change. Due to these characteristics, it is a very complex task for application developers to handle web data efficiently. This is particularly true for so-called situational analytics and mashups which are developed by users with very different skill levels. To leverage the heterogenous resources on the web and to provide a uniform interface for applications, it is necessary for the data to be integrated into a queryable and consistent, but also flexible data model.

This challenge of integrating data from a number of diverse sources bears resemblance to ETL (extract, transform and load) processes common in data warehouse scenarios. Data from different sources, conforming to different schemas, is integrated using a mediated schema. This schema remains unchanged for long periods of processing and is only rebuild when it is required due to significant changes in the original schemas. In the context of the web, we need to integrate not only data with structural diversity, but also data that is schema-free. The main challenge, however, is the volatility of the resources. In contrast to the resources of a data warehouse, resources on the web change frequently and erratically. Traditional data models, such as the relational data model, do not provide the flexibility required to efficiently deal with these characteristics. Instead, we take a graph-based approach towards a flexible data model supporting the integration as well as the continuous evolution of web data. The model is meant to form the basis of a data repository for web applications. Inspired by the Web 2.0 trend of user participation, we plan to provide users with tools to collaboratively and incrementally enhance the integration of the data.

2 Research Problems and Objectives

The main problem we want to address in our research is the heterogeneity of web data and the resulting issues for applications regarding the integration and management of the data. In this scenario we have identified the following challenges, which we will address in our research. First of all, schema management should be flexible enough to handle both, structured and unstructured data. To achieve this, schema information should be optional so that different levels of structure can appear simultaneously in the system. This will, for example, enable unstructured data to be imported without extensive transformations. Inconsistencies in the schemas of different sources of structured data need to be addressed through mapping techniques. Furthermore, the query functionality should be determined by the amount and quality of metadata available. Due to the volatility of resources on the web, we need to take the evolution of the schema into consideration as well. However, extending and changing schema information should be non-destructive, which means it should not require the re-building of application processes. This requires a balance between flexibility and consistency regarding the schema. Additionally, we need to incorporate schema versioning, to ensure that schema changes do not invalidate previously existing applications. Apart from these features, which are closely related to the data integration and schema design challenge, there are further related topics that need to be studied in this context, but which are not the primary focus of our research. They include amongst others transaction support, permission and privacy issues, efficient data storage and distribution.

3 Research Methodology and Approach

To achieve the outlined objectives we will build on existing data models and query languages. Instead of enforcing a static mediated schema during the data load process, we plan to enable incremental extraction and enhancement of schema information. Leveraging the current web trend, we want to encourage users to participate in the management and integration of their data by collaboratively building schemas as they are required for querying. Automated extraction and integration techniques will be incorporated to support users through, for example, recommendations. An overview of our approach is depicted in Figure 1.

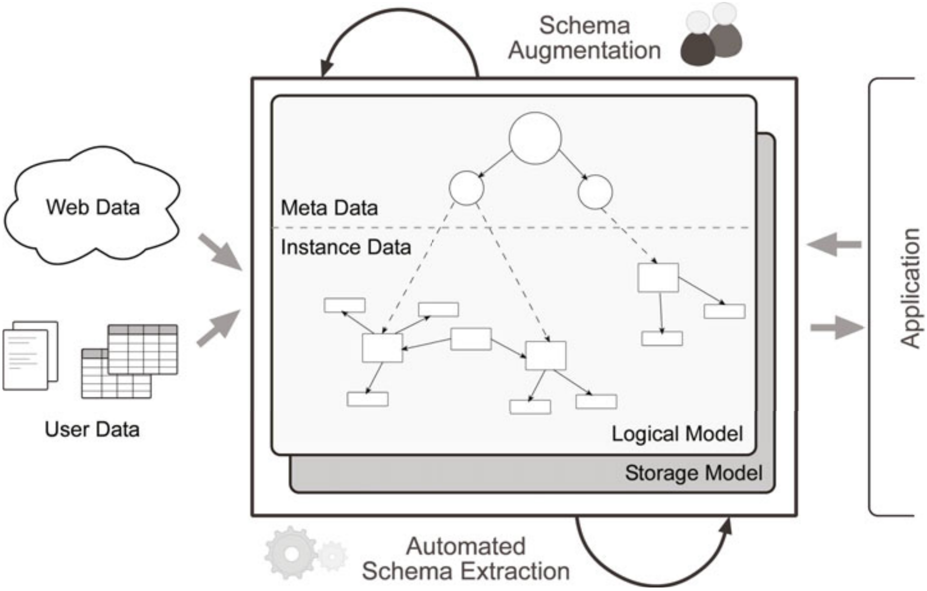

Fig. 1. Overview: Graph-based data model supporting incremental and collaborative schema evolution

3.1 Graph-Based Data Model

The traditional relational data model is a well established choice for structured data. However, it is not considering the heterogeneity and volatility of web resources. Instead, we use a directed, labeled multi-relational graph, a common model in graph databases, as a basis for our approach. Vertices and edges in the graph represent entities and relationships respectively. Both, vertices and edged, can be labeled with name/value pairs which represent properties. Schema information can be stored in a schema graph that defines entity types, primitive types and relationship types. Instance data is stored in an instance graph which contains concrete entities, primitive values as well as concrete relationships. We take a graph based approach due to a number of characteristic features that support our proposition. First of all, the graph structure supports different levels of complexity within the same graph. Not all instances are required to have the same amount and type of properties and relationships. Additionally, a tight coupling of data and meta data is achieved by representing both as graphs that are connected. This leads to a natural integration of metadata into the system and facilitates meta data querying. Finally, graph structures are easy to interpret, which is beneficial for our plan to enable strong user participation.

3.2 Tentative Research Plan

The tentative plan for our research consists of the following steps:

Final edited form was published in "Current Trends in Web Engineering: Workshops, Doctoral Symposium, and Tutorials. Paphos 2011", S. 302–306, ISBN 978-3-642-27997-3
http://dx.doi.org/10.1007/978-3-642-27997-3_29

1. First, we have to define a uniform data model that enables efficient integration of structurally diverse data and flexible evolution of the schema. So far, we have selected the graph model described above as a suitable basis for our data model.
2. Our next task is the development and incorporation of basic operators for the definition, manipulation and querying of both, data and metadata. Our goal is to exploit the natural features of the graph structure for the operators as much as possible. For example, we can utilize graph traversal techniques for the propagation of schema modifications. These operators include operators that enable users to incrementally enhance the schema.
3. Additionally, we will consider suitable techniques to ensure stability for applications despite schema changes. Therefore, schema inconsistencies with regard to the applications must be compensated to a certain degree in order to delay expensive reorganizations.
4. In connection with schema evolution we will study options for supporting schema versioning in our data model.
5. In addition to studying the flexible schema design and evolution on a theoretical level, we will implement our approach in a prototype to validate our assumptions. In order to evaluate the flexibility, scalability and efficiency of our approach, we will look for a suitable benchmark [4].

4 Related Work

In addition to the relational data model, which is often the standard when dealing with structured data, a number of alternative data models have emerged in the context of the web. Often associated with the term "NoSQL databases", these alternatives include basic key/value stores, column-oriented stores, document stores and graph databases. The main concerns of these systems are on the one hand scalability that meets the requirements of big web applications and on the other hand, relaxation of the tight schema requirements of relational systems towards a schema-free solution. In our research we focus on graph databases which offer the highest flexibility. Angeles et al. [1] provide an extensive survey of existing graph database models including various graph query languages. In [6], a hypergraph model is introduced. Based on this model, storage, querying as well as indexing techniques are described. A well established open source graph database is Neo4j[1]. In [2], Bollacker et al. present Freebase, a graph database for storing human knowledge in a structured manner. Freebase provides tools for users to collaboratively augment the data and schema. In contrast to our approach, data in Freebase has to conform to predefined types, which can be extended by the user, but do not provide the flexibility we aim to achieve.
The incremental extraction of metadata from unstructured data has been addressed in [3]. Chu et al. propose a relational approach based on the interpreted storage format using three basic operators (extract, integrate and cluster) to incrementally discover and extract structure. It maintains the flexibility of schemaless models, since keyword search can be applied at any time, but also enables

[1] http://neo4j.org/

the user to run complex queries as soon as more schema information has been extracted. A similar concept for structured data can be found in research regarding dataspace systems [5]. Data sources, that have not been integrated fully, can be queried through keyword search, for example. If more complex operations like data mining are requested, the data sources can be integrated incrementally. This approach is often referred to as ”pay as you go”, which also inspired the web-scale data integration architecture PayGo by Madhavan et al. [7]. PayGo aims at incrementally integrating structured data found on the Web by applying techniques for automated schema mapping and schema clustering as well as techniques for discovering additional relationships between data.

5 Conclusion

The heterogeneity of data resources on the web present a difficult challenge for web application developers. Many existing data integration solutions are not flexible enough to handle both structured and unstructured data and are not designed to address the volatility of data sources on the web. We have presented our concept for a graph-based solution, which utilizes a combination of automated techniques and user participation to achieve flexible data integration and evolution.

References

1. Angles, R., Gutierrez, C.: Survey of graph database models. ACM Comput. Surv. 40 (2008)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD 2008 (2008)
3. Chu, E., Baid, A., Chen, T., Doan, A., Naughton, J.: A relational approach to incrementally extracting and querying structure in unstructured data. In: VLDB 2007, pp. 1045–1056 (2007)
4. Curino, C.A., Tanca, L., Moon, H.J., Zaniolo, C.: Schema evolution in wikipedia: toward a web information system benchmark. In: Enterprise Information Systems (2009)
5. Franklin, M., Halevy, A., Maier, D.: From databases to dataspaces: a new abstraction for information management. SIGMOD Rec. 34 (2005)
6. Iordanov, B.: HyperGraphDB: A Generalized Graph Database. In: Shen, H.T., Pei, J., Özsu, M.T., Zou, L., Lu, J., Ling, T.-W., Yu, G., Zhuang, Y., Shao, J. (eds.) WAIM 2010. LNCS, vol. 6185, pp. 25–36. Springer, Heidelberg (2010)
7. Madhavan, J., Jeffery, S.R., Cohen, S., Dong, X.L., Ko, D., Yu, C., Halevy, A., Inc, G.: Web-scale data integration: You can only afford to pay as you go. In: CIDR 2007 (2007)