Combining Provenance Management and Schema Evolution

Tanja Auge(✉) and Andreas Heuer(✉)

University of Rostock, Rostock, Germany
tanja.auge@uni-rostock.de, heuer@informatik.uni-rostock.de
https://dbis.informatik.uni-rostock.de

Abstract. The combination of provenance management and schema evolution using the CHASE algorithm is the focus of our research in the area of research data management. The aim is to combine the construction of a CHASE inverse mapping to calculate the minimal part of the original database — the *minimal sub-database* — with a CHASE-based schema mapping for schema evolution.

Keywords: CHASE algorithm · Data provenance
Schema evolution · Data evolution · Schema mapping
CHASE inverse

1 Introduction

Collecting, recording, storing, tracking, and archiving scientific data is the task of research data management, which is the basis for scientific evaluations on this data. In addition to the evaluation (i.e., a complex database query that we call *evaluation query*) and the result itself, the section of the original database used has also to be archived. Thus, to ensure reproducible and replicable research, the evaluation queries can be processed again at a later point in time in order to reproduce the result.

If the data or the schema of the research database changes frequently, the original database would now have to be *frozen* (permanently stored) after every evaluation carried out on the database. In order to avoid this and in order to avoid massively replicated databases, we want to use provenance management techniques to calculate the minimal part of the database that must be frozen in order to be able to generate the query result again. For this, we want to combine techniques of ***why*** and ***how*** provenance [3] with the theory of schema mappings for data integration and data exchange, especially the inverse schema mappings of Fagin [5,6].

In research data management, the path from data collection to publication should be kept comprehensible, reconstructable, and replicable [9]. Since the research database is constantly changing [8] and thus represents a bitemporal database [7], the evolution of data and schemata must interact with the management and archiving of results, the management of the evaluation queries, and

K. Belhajjame et al. (Eds.): IPAW 2018, LNCS 11017, pp. 222–225, 2018.
https://doi.org/10.1007/978-3-319-98379-0_24

the provenance management. Unfortunately, data provenance research has normally been carried out on a fixed database. Two research goals of the project are therefore (1) the calculation of the minimal part of the original research database (we call it *minimal sub-database*) that has to be stored permanently to achieve replicable research, and (2) the unification of the theories behind data provenance and schema (as well as data) evolution.

2 Problem and Poster Description

2.1 Calculation of a Minimal Sub-database

The calculated minimal sub-database should be able to reconstruct the results of the evaluation query under various constraints. The following constraints range from very strict preconditions to weaker constraints:

- The number of tuples of the original relation is retained.
- The sub-database can be homomorphically mapped to the original database.
- The sub-database is an intensional description of the original database.

One specific problem is to decide about the minimal (additional) information that is required for the reconstruction of the sub-database, provided that the query result and the evaluation query is archived. Is it sufficient to pick up a minimum amount of witnesses (***why***-Provenance, [4]) or to calculate the associated provenance polynomials (***how***-Provenance, [10])? Or is it necessary to freeze whole tuples or other parts of the database directly?

The calculation of an inverse query Q_{prov}, which is used to determine the required minimal sub-database, depends on the type of the original query Q and any additional information noted. Thus a *result equivalent CHASE inverse* can be used for the projection [2]. A projection without duplicate elimination can be specified by a *relaxed CHASE inverse* and a simple copy operation by an *exact CHASE inverse* [6]. The homomorphism as a required condition mentioned above is a quite strong constraint which has to be weakened in future investigations [1].

2.2 Unification of Provenance and Evolution

Previous provenance queries Q_{prov} (***where***-, ***why***- and ***how***-provenance) have usually been processed on a given fixed database S_1 and a query Q. The combination of data provenance with schema and data evolution should enable the evaluation of provenance queries with changing data and schemata (see Fig. 1). By means of an inverse evolution step $\mathcal{E}^{-1}$, the new database J can be transferred to the old schema, if possible. Formally, our evaluation result is calculated by an extended CHASE algorithm, based on ST-TGDs (see below), and an (inverse) provenance query Q'_{prov} should be added in a second step, the BACKCHASE phase. The minimal sub-database I^* (red dashed box) is then computed by chasing the provenance query Q_{prov} into the query result $K^* \subseteq K$ (green box), adding the necessary provenance annotation (such as provenance polynomials).

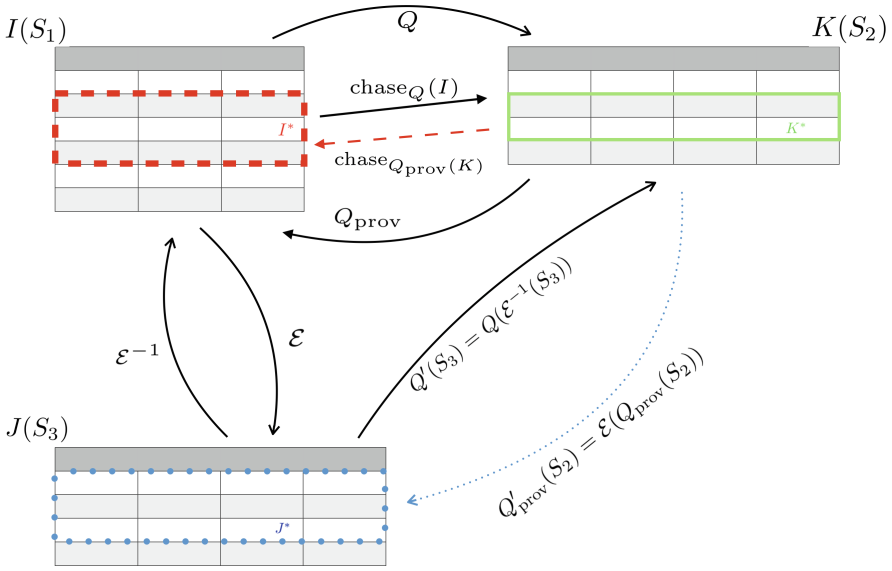

Fig. 1. Unification of provenance and evolution (Color figure online)

Under the schema evolution $\mathcal{E} : S_1 \to S_3$, the query Q' can be directly calculated as a composition of the original query Q and the inverse evolution $\mathcal{E}^{-1}$:

$$Q'(J(S_3)) = (\mathcal{E}^{-1} \circ Q)(J(S_3)) = Q(\mathcal{E}^{-1}(J(S_3))) = Q(I(S_1)).$$

The new provenance query Q'_{prov} results analogously as

$$Q'_{\text{prov}}(K^*(S_2)) = (Q_{\text{prov}} \circ \mathcal{E})(K^*(S_2)).$$

It is therefore sufficient to memorize one of the two minimal sub-databases $I^*(S_1)$ (red dashed box) or $J^*(S_3)$ (blue dotted box). The other can be calculated with the help of the inverse. In research data management, K^* always corresponds to the entire result database K, i.e. $K^* = K$, since the complete result of the scientific evaluation has to be reproducible. However, general provenance queries can also be processed on subsets of this result (or even on single tuples in the result).

2.3 Query Q

The representation of the evaluation query Q in the form of extended S-T TGDs (*source-to-target tuple-generating dependencies*) or EGDs (*equality-generating dependencies*) allows the application of the CHASE algorithm [5,6]. This incorporates a set of dependencies, here S-T TGDs and EGDs, into a given database instance. The calculation of a CHASE inverse Q_{prov} via the BACKCHASE involves the reconstruction of the minimal sub-database I^* of the original database $I(S_1)$.

2.4 Evolution $\mathcal{E}$

By using the inverse $\mathcal{E}^{-1}$, the old minimal sub-database I^* can be calculated from the current minimal sub-database J^*. For this, the evolution $\mathcal{E}$ and its (exact) inverse $\mathcal{E}^{-1}$ are formulated as S-T TGDs and EGDs and processed by the CHASE algorithm.

2.5 Data Provenance Q_{prov}

The result of the evaluation query Q described by extended S-T TGDs and EGDs can be calculated using the CHASE algorithm. The subsequent construction of the minimal sub-database I^* succeeds by inverting the query Q. This inverse Q_{prov} doesn't necessarily have to correspond to an inverse in the classical sense

$$Q \circ Q_{\text{prov}} = \text{Id},$$

since a *CHASE inverse* can't always be specified [5,6]. In most cases, however, a *result equivalent CHASE inverse* [1,2] can be specified that returns the same result after applying the CHASE algorithm to the original instance I and the minimal sub-database I^* calculated using the BACKCHASE.

References

1. Auge, T., Heuer, A.: Inverse im Forschungsdatenmanagement. In: Proceedings of 30th Workshop Grundlagen von Datenbanken (2018, accepted for publication, to appear). (in German)
2. Auge, T.: Umsetzung von Provenance-Anfragen in Big-Data-Analytics-Umgebungen. Master's thesis, University of Rostock (2017). (in German)
3. Cheney, J., Chiticariu, L., Tan, W.C.: Provenance in databases: why, how and where. Found. Trends Databases **1**(4), 379–474 (2009)
4. Buneman, P., Khanna, S., Tan, W.C.: Why and where: a characterization of data provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 316–330. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44503-X_20
5. Fagin, R.: Inverting schema mappings. ACM Trans. Database Syst. **32**(4), 25-1–25-53 (2007)
6. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Schema mapping evolution through composition and inversion. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) Schema Matching and Mapping. DCSA, pp. 191–222. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-16518-4_7
7. Johnston, T.: Bitemporal Data: Theory and Practice. Morgan Kaufmann, Burlington (2014)
8. Bruder, I., Klettke, M., Möller, M.L., Meyer, F., Heuer, A., Jürgensmann, S., Feistel, S.: Daten wie Sand am Meer – Datenerhebung, -strukturierung, -management und Data Provenance für die Ostseeforschung. Datenbank-Spektrum **17**(2), 183–196 (2017). (in German)
9. Heuer, A.: METIS in PArADISE: Provenance Management bei der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen. In: BTW Workshops. LNI, vol. 242, pp. 131–136. GI (2015). (in German)
10. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: PODS, pp. 31–40. ACM (2007)