

Modelling Data Warehouses with Multiversion and Temporal Functionality

Waqas Ahmed⁺, Esteban Zimányi⁺, Robert Wrembel^{*}

⁺Department of Computer and Decision Engineering (CoDE), Université Libre De Bruxelles,

^{*}Institute of Computing Science, Poznań University of Technology

waqas.ahmed@ulb.ac.be

1. Introduction

- Data Warehouses (DWs) integrate data from heterogeneous external data sources (EDSs)
- EDSs evolve in content and structure
- The changes in EDSs must be propagated into the DWs

2. Available Approaches

1. **Slowly Changing Dimensions (SCDs)** – A technique to keep the content history of dimension members
 - Three basic types of SCD, each with a different method of handling the changes
2. **Temporal Data Warehouses** – Track the evolution history of attributes by associating orthogonal time dimension(s) to each record
3. **Multiversion Data Warehouses** – Consist of a sequence of DW versions
 - Every change creates a new DW version

3. Drawbacks of Available Approaches

1. **Slowly Changing Dimensions**
 - No or incomplete history
 - No support for schema changes
2. **Temporal Data Warehouses**
 - No support for schema changes
3. **Multiversion Data Warehouses**
 - Version management overhead
 - Resource requirements
 - Querying is not trivial

4. Motivations

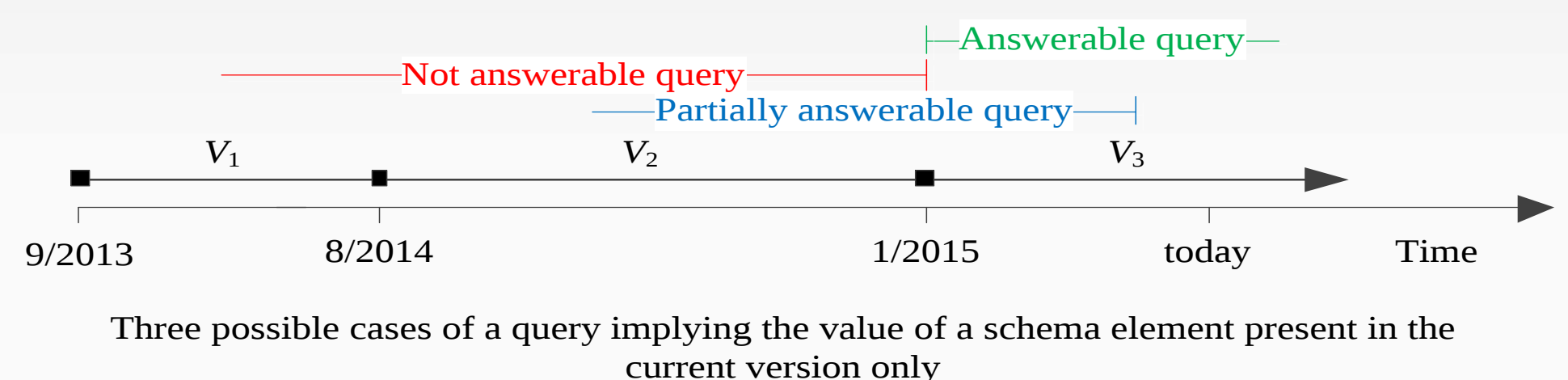
1. Querying data stored in multiple schema versions is not trivial
2. Partial results are better than no results at all
3. There is a need for a DW model that could support schema versioning and keep track of the history of data within each version

5. Objectives

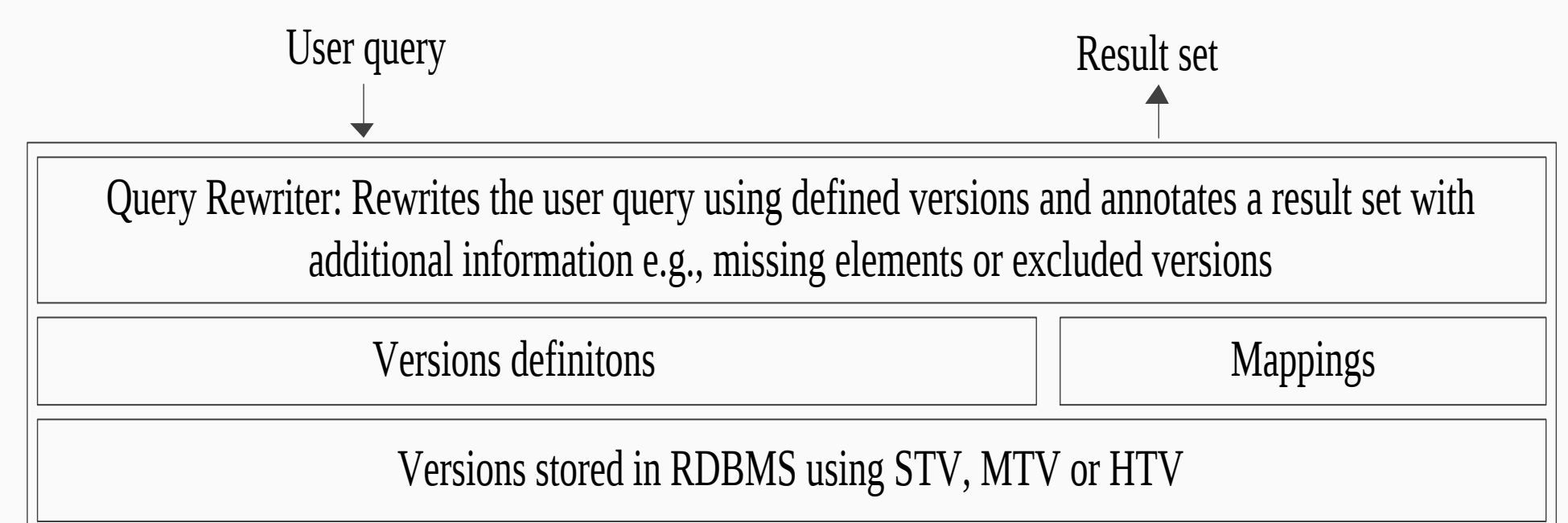
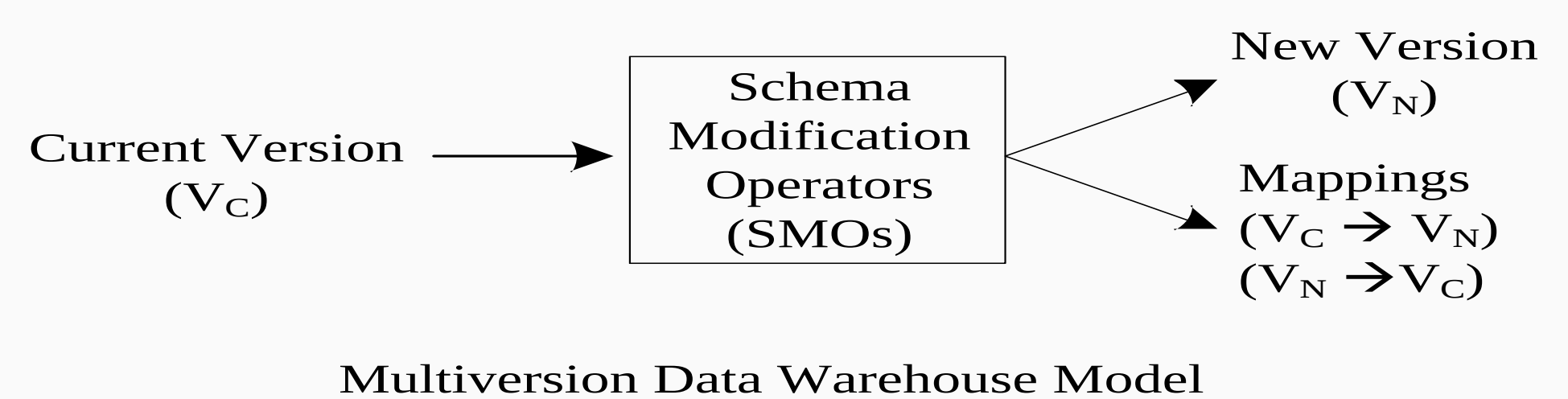
Design, implement, and evaluate a data warehouse model that can support multiple schema versions and track the temporal evolution of data within its versions.

6. Cross Version Queries

- Imply a time interval during which more than one schema versions are valid
- Not trivial to answer because data stored across multiple versions may have different structure



7. Proposed Solution



8. Conclusions

1. Combining the multiversion and temporal approaches is the natural solution to the problem of managing content and structure evolution in DWs
2. A model is needed that could support both the schema versioning and management of temporal evolution of content within the schema version, and
3. A query mechanism is required to answer the cross-version queries

References

- M. Golfarelli and S. Rizzi. *A survey on temporal data warehousing*. International Journal of Data Warehousing and Mining, 5(1):1-17, 2009.
- W.Ahmed, E. Zimányi and R. Wrembel. *A logical model for multiversion data warehouses*. DaWak 2014, Munnich, Germany.
- W. Ahmed and E. Zimányi, *On querying multiversion data warehouses*, ADBIS 2015, Poitiers, France