Modelling Data Warehouses with Multiversion and Temporal Functionality

Waqas Ahmed, Esteban Zimányi, Robert Wrembel*

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

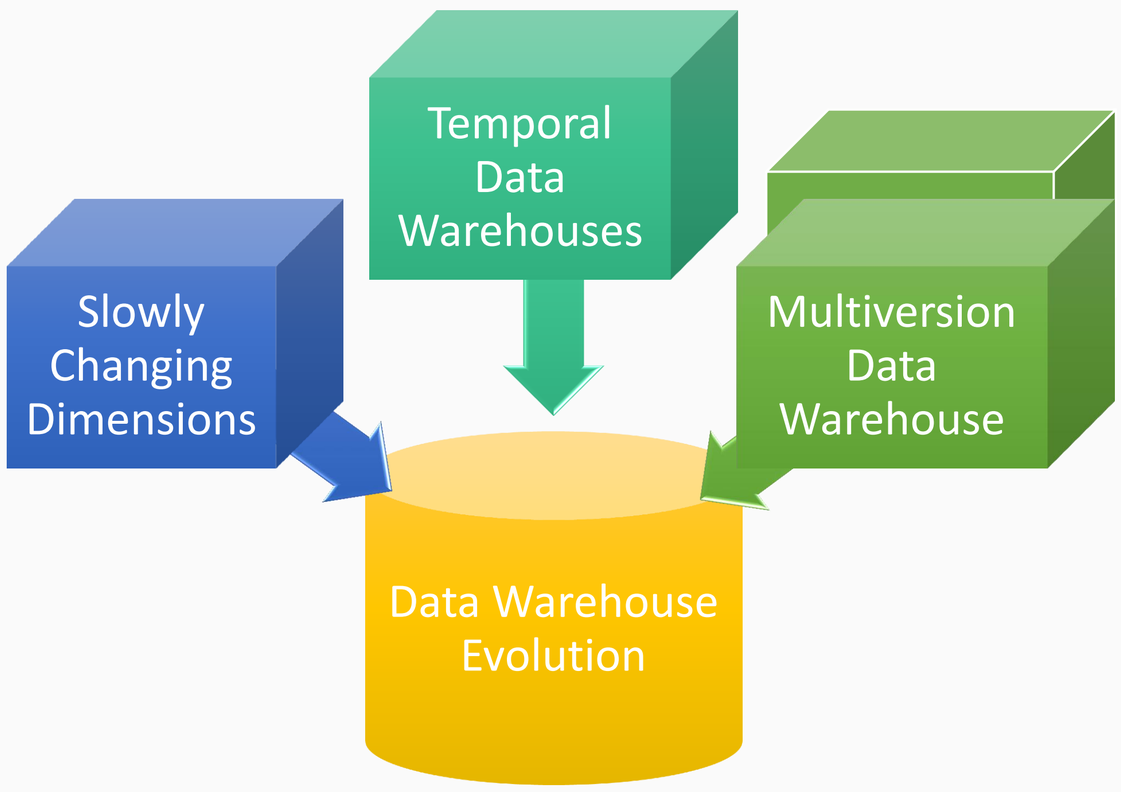
*Institute of Computing Science, Poznań University of Technology

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 and indexing issues

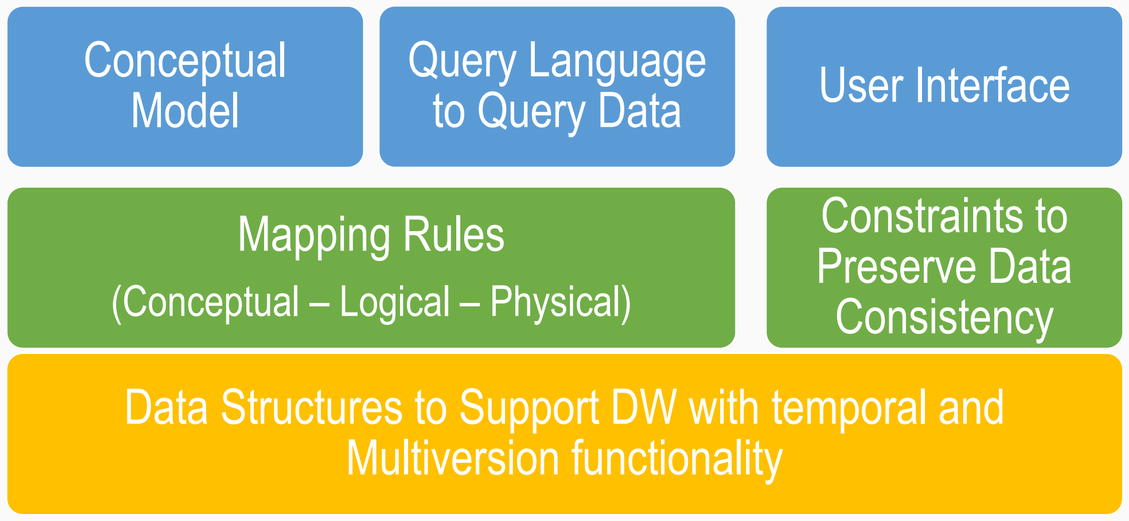
4. Motivations

1. Existing approaches to handling DW evolution support
   - Data Evolution
   - Schema Evolution
   - Schema and Data Versioning
2. User cannot "turn off" schema versioning leaving data versioning "on", and vice versa
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. Proposed Solution



6. Conclusions

1. Combining the multiversion and temporal approaches is the natural solution to the problem of managing content and structure evolution in DWs
2. It requires a model that could support both the schema versioning and management of temporal evolution of content within the schema version
3. Data structures are needed to store and support efficient querying of data

References

- M. Golfarelli and S. Rizzi. *A survey on temporal data warehousing*. International Journal of Data Warehousing and Mining, 5(1):1-17, 2009.
- R. Wrembel and B. Bebel. *Metadata management in a multiversion data warehouse*. Journal on Data Semantics VIII, pages 118-157. Springer, 2007.
- W.Ahmed, Esteban Zimanyi and R. Wrembel. *A logical model for multiversion data warehouses*. DaWak 2014, Munnich, Germany.