AN OLAP CUBE BASED DEVELOPMENT METHOD OF INFORMATION SYSTEMS

Lemmik, R.

Abstract: The hypothesis of this work is that information systems development is possible with lower TCO (Total Cost of Ownership), if OLAP (On-Line Analytical Processing) model is developed together with OLTP (On-Line Transaction Processing) model and if those two models are in relation to each other. That kind of combined model makes possible to develop virtual data warehouse solution, with automatically generated ETL (Extract, Transform, Load) business logic. The result of this work is the customized development method of information systems, where are joined together OLAP and OLTP conceptions to create the combined data model. Composed method is base for the virtual data warehouse to provide same kind of functionality like classic and static OLAP dimensional database does.

Key words: virtual data warehouse, data modeling, software development, data analysis, model driven architecture

1. INTRODUCTION

Classical development method of Information Systems contains always transactional data model which it is very well structured and normalized without plenty data redundancy. This kind of data model covers all requirements for entering data and also there is easily possible to create reporting over the objects in the system. However in common case there is also necessary to create analytical reporting over the actions in the system and for those requirements the transactional data model isn’t applicable because it doesn’t represent actions. Therefore it is necessary to develop additionally for analytical reporting a denormalized data model with appropriate data views, commonly called as OLAP cubes [1].

1.1. Ineffective data warehouse

Classical development method of data warehouse contains following:

- Relevant actions are determined from processes
- Relations between actions and objects are re-engineered from transactional data model
- Reasonable amount of OLAP cubes are created
- Denormalized data model and database is created
- ETL business logic is created to load data warehouse with the data from transactional database

Data warehouses created this way will encounter often similar problems:

- Complete system development is separated into two parallel threads and this can cause data quality problems in data warehouse
- It takes lot of work to develop ETL business logic between those two threads with acceptable quality
- Re-engineering the transactional data model can cause mistakes and data quality problems in data warehouse

For those reasons the classical methods are ineffective under contemporary dynamic environment [2].
2. OLAP CUBE BASED DEVELOPMENT METHOD

Purpose of the combined development method is to include one abstraction level higher meta-model as a technical part of the information system. This model contains transactional and analytical data models combined together with reliable relations.

OLAP cube based development method presented in Fig.1 contains following:
- Process model is transcribed as combined data model which concurrently represents transactional and analytical point of view
- ETL business logic is generated automatically from combined data model

Described meta-model represents two models which have practically very different structure and that is the reason why there is necessary to have described both of those models and the combined meta-model contains relations between those [3].

2.1. Combined Meta-Model and Virtual Data Warehouse

Compared to classical data warehouse the virtual data warehouse needs to have meta-model as a mandatory part of the system. This makes possible to have data stored in one abstract repository and create model-driven representation of the data. In virtual data warehouse system as well as data representation also ETL business logic is generated automatically and dynamically with model-driven concepts. Classical data warehouse is presented in Fig.2 and virtual data warehouse is presented in Fig.3.
With this kind of MDA (Model Driven Architecture) the virtual data warehouse is only one possible realization. Similarly it is possible to create dynamic data warehouse where OLTP and OLAP physical databases are both automatically generated and as well the ETL business logic between them. This kind of system with dynamic data warehouse can guarantee sufficient performance for very complex systems with big amount of data and at the same time the data quality in both databases will be assured.

4. COMBINED META-MODEL

Described meta-model combines together transactional and dimensional data models.

Transactional data models are commonly represented as ER models where Entity, Relation and Attribute exist. Compared to ER model the transactional model is a bit more complex where Table corresponds to Entity, Data Field corresponds to Attribute. Meta-model of the transactional data model is presented in Fig.4 [1].

Analytical data models are commonly represented as star or snowflake models where Fact, Dimension and Measure exist. Meta-model of the analytical star data model is presented in Fig.5.

To resolve time dimension problem, described above is created ancillary data structure to hold time dimension values as classificatory. Time classificatory is filled automatically with data to cover whole analytical data periods. Dim_Days table contains all dates, Dim_Months contains initial dates of all months, Dim_Quarters contains initial dates of all quarters and Dim_Years contains initial dates of all years within chosen period. Time dimension classificatory structure is presented in Fig.6.
Combined meta-model presented in Fig.7 contains relations between data models described above.

**Entity DB_Tables** describes tables from transactional data model which are linked with both, facts and dimensions in analytical data model. Also above described time classificatory tables are defined here as transactional tables.

**Entity DB_Columns** describes data columns from transactional data model which are linked with facts as dimension values and measures. Also necessary data fields to describe dimensions classificatory are defined here.

**Entity DB_DataTypes** describes possible data field types in transactional data model.

**Entity DB_Relationships** describes one-to-many relationships between data fields in transactional data model. It is also possible to define relationships here if key consists from more than one data fields.

**Entity OLAP_Fact** describes facts from analytical data model.

**Entity OLAP_Measure** describes measures from analytical data model which are linked with facts. Measures are used in OLAP data views as summarized data.

**Entity OLAP_Fact_Table_rel** describes relationships for each fact and measure combination in analytical data model with data column in transactional data model. It is also possible to reference one fact and measure combination with more than one data column from more than one table in transactional data model.

Fig.7. Combined Meta-Model [1]
**Entity OLAP_Dimension**
describes dimensions in analytical data model. One dimension can be in relation with more than one fact so it makes possible to define multiple-star type analytical views.

**Entity OLAP_Dimension_Level**
describes dimension hierarchy in analytical data model. This makes possible to define snowflake type analytical views. Also here are defined relations with data columns in relational data model where are located dimension value descriptions which ones are used in OLAP data views.

**Entity OLAP_Fact_Dim_rel**
describes relations between facts and dimensions in analytical data model for OLAP data views. Also here are defined relations between analytical fact and transactional tables as well between analytical dimension and transactional data columns [1].

5. CONCLUSION

Current paper describes data modeling techniques where the main focus is on OLTP and OLAP combined meta-model. By using conceptions of MDA the virtual data warehouse deployment will be fully automated and the created meta-model is used as an inseparable part of the systems technical architecture.

The prototyped result of that work is virtual data warehouse for ERP software Microsoft Dynamics NAV where end-user can define OLAP cubes by using intuitively understandable OLAP data structure and the universal application generates data view as Excel pivot table. As well there is possible to use any kind of OLAP data analysis tools instead of Excel.

To continue evolvement of that idea, the plans contain to improve the meta-model and universal application and realize dynamic data warehouse together with automatically generated ETL business logic. That kind of automated deployment of the dynamic data warehouse can exclude database structure accordance and data quality problems, like most manually developed data warehouses have.

6. ACKNOWLEDGEMENTS

This research was supported by Innovative Manufacturing Engineering Systems Competence Centre IMECC co-financed by Enterprise Estonia (EAS) and European Union Regional Development Fund (project EU30006).

7. REFERENCES

1. Lemmik, R. (2008); An OLAP Cube Based Development Method of Information Systems, Tallinn University, Master Thesis
2. IBM International Technical Support Organization (1998); Data Modeling Techniques for Data Warehousing
3. IBM International Technical Support Organization (2006); Dimensional Modeling: In a Business Intelligence Environment

8. ADDITIONAL DATA ABOUT AUTHORS

Rivo Lemmik, MSc, Innovative Manufacturing Engineering Systems Competence Centre (IMECC), Akadeemia tee 19, 12618 Tallinn, Estonia
E-mail: rivo@raadi.net
Mobile: +372 515 0554