Using SLE for creation of Data Warehouses

Yvette Teiken

OFFIS, Institute for Information Technology, Germany
teiken@offis.de

Abstract. This paper describes how software language engineering is applied to the process of data warehouse creation. The creation of a data warehouse is a complex process and therefore costly. My approach decomposes the data warehouse creation process into different aspects. These aspects are described with different languages which are integrated by a metamodel. Based on this metamodel, large parts of the data warehouse creation process can be generated. With my approach data warehouses are created more comfortable in less time.

1 Problem Description and Motivation

Health Reporting describes the preparation and presentation of health relevant issues relating to population. It is used to give information to stakeholders in the health care system, politicians and interested non-professionals. Furthermore, risks are identified and appropriate warnings issued. In the Federal State of North Rhine-Westphalia this task is carried out by the government agency for public health called LIGA (Landesinstitut für Gesundheit und Arbeit). LIGA provides a variety of different reports and is able to answer ad hoc questions. The reports are based on data from different sources and different systems with different formats. These sources are e.g. data from different public health departments or insurances. To fulfill these requirements, software support is needed. This means data has to be integrated into one dataset so that different sources can be related. Support is also needed for transforming data on regular basis into the integrated dataset. Also frontend and report generation has to be developed.

One software system providing this support is the MUSTANG platform developed at OFFIS. MUSTANG is used at LIGA. The base of each MUSTANG instance is an integrated dataset, this is also called data warehouse (DWH). In an integrated dataset all relevant organizational knowledge is stored for complex analysis. For fast data navigation and analysis, Online Analytical Processing (OLAP) [4] is often used. OLAP is an approach that allows navigation and querying data more comfortable than using exact queries like SQL. For OLAP a multidimensional data model is needed. The initial build-up of a DWH with a multidimensional integrated dataset is a complex task [8]. During the initial build-up of DWH the following analyzing and design steps have to be performed:

Analysis of organizational data: To find data that can be used in the resulting DWH, existing data sources have to be analyzed. This analysis includes the content, format and the accessibility of the data. This kind of data is

called fact data. In a DWH this fact data is extracted and integrated into the so called integrated dataset. **Define information demand:** Define what information should be provided by the DWH. This can be simple figures or complex computations. **Data source transformation:** Fact data has to be transformed in the data format of the integrated dataset. Therefore, for every data source a transformation has to be designed that translates data into the format of the integrated data set and stores it there. **Define multidimensional data model:** Defines how fact data can be described multidimensionally and grouped together in hierarchies. **Data quality:** Based on DWH, analysis decisions are made so it is important to define data quality standards and how to identify invalid data.

To perform these steps no standardized process exists. Documentation of these steps is usually done with a large number of documents. A problem with this kind of documentation is missing, distributed or inconsistent information. Another aspect is that during realization a lot schematic work has to be done. For example, a multidimensional schema has to be designed and realized in the OLAP system, the integrated dataset, and at the frontend software.

2 Related Work

Data Warehouse analysis and design as described in [1] consists of different phases. For these different phases of the DWH creation, languages and tools have been developed. In case of multidimensional modeling, languages like Application Design for Analytical Processing Technologies (ADAPT) by [3] exist. There are also languages that describe mapping for relational databases like R2D [2] or languages that describe data quality issues like InDaQu [15].

Another field of related work is automated creation of DWHs. The feasibility to connect MDA with the DWH process has been shown in [12]. They also developed a MDA framework for DWH. It covers data integration, data sources, and multidimensional models. The authors show the application of their approach through a case study. However, their main focus are models and not languages.

More related to SLE is the work of Rizzi. His group deals with modeling different aspects of DWHs. For example in [13] modeling technique for data cubes and data flows are suggested and in [9] a UML based approach for *what if*-analysis is provided. Another work that deals with SLE and aspects of DWH creation is [7]. They use modeling languages to generate multidimensional schemas.

All these approaches only deal with a certain aspect of DWH creation, not with the whole process with language support. In my thesis, I will develop an approach with languages that cover the whole process of DWH creation. These languages are integrated through a common metamodel and can deal with multidimensional structures. Based on the metamodel I will create transformations that allow generating large parts of the resulting DWH. With these transformations, schematic work in the step of realization is reduced. Furthermore, I will create a process model that orders the steps described in combination with the developed languages to improve documentation. With the process model, the different aspects are connected and refined.

3 Proposed Solution

Data warehouses, as describes here, are very complex systems with different views, aspects, and levels of detail. To create a single language, these systems are difficult and not easy to use and maintain. Therefore, it is necessary to decompose a DWH creation into different aspects and create languages for each. The different languages will be used by different roles and provide a different level of detail. A first result is that the process can be decomposed in six aspects:

Data Sources Schemas: Describes all relevant or available data stored in its operational system of an organization. This aspect contains the subject, the representation, and technical accessibility. The advantage is that all relevant sources are described together with their formats and accessibility at one place. For this aspect the development of an own language may not be necessary but a meta model will be sufficient.

Data Source Transformation: Describes how data from the sources have to be transformed to match the analysis schema. Such a description makes it possible to abstract from the concrete target system and some automatic matching process can be applied.

Analysis Schema: This aspect describes the multidimensional schema of the resulting DWH. The multidimensional schema consists of cubes and dimensions. In a cube, fact data is stored. Each cell represents fact data that is numerical and can be aggregated. They are described by dimensional metadata. In a cube with a time dimension, fact data is stored for dates so monthly and yearly values are computed by aggregation. The language is based on ADAPT. With such a language it is possible to reduce schematic work in the realization phase, as shown in [17], and it can be used to communicate with domain experts.

Measures: Describes what kind of information is intended to be stored in the DWH. This can be simple fact data like infections. It also includes the designated granularity of information. For example, to predict an epidemic infect, data should be available for every date. Measures link fact data with mathematical operations. In Health Reporting, these are mostly crude rates, interest, and average. Measures are usually defined by domain experts, in case of LIGA by epidemiologists. Measures are refined in the analysis schema. With a language for measures the definition can be used in the realization process and does not need to be reimplemented.

Hierarchy: The hierarchy aspect is a central one in my thesis because all other aspects use this directly or indirectly. This aspect describes how data is aggregated. In most cases hierarchies have many members and a complex structure and they are used in the multidimensional model. For example, imagine a geographical dimension that contains countries and cities. Using an own language, these structures can be modeled appropriate. It can help to build a repository that can be reused in a different DWH. The concrete syntax of the hierarchy language is based on ADAPT. However, it only allows to model hierarchies conceptually. To model hierarchies logically a tabular extension was created. To create parent-child relationships a query language was also integrated.

Data Quality: In a DWH it is important to ensure that certain quality issues are met. Naturally, data quality is an aspect of the data sources but when integrating different systems it would be very costly to deal with quality at the sources because many different systems have to be considered and changing these systems is rarely possible. Data quality is a large research field. In my approach, I want to integrate existing approaches to show how data quality issues can be integrated. InDaQu [15] is integrated to deal with data consistency.

The languages for the described aspects are developed independently. Each language is developed with SLE techniques [6] and based on tools like EMF [14] and MS DSL Tools [5]. To generate a DWH, the different languages and their metamodels have to be integrated into one metamodel because the different aspects are very strongly related. Furthermore, to be able to cover the whole process refering elements from other aspects is necessary.

The integrated metamodel covers all aspects of a DWH at one place. A common metamodel is a standardized documentation of the whole DWH system. Other possiblities of an integrated metamodel have been shown in [16]. To integrate a metamodel, [10] suggests two ways, via transformation and via common elements. I decided to use integration via unidirectional transformation for integration into metamodel. I created my DSLs with MS DSL Tools but for better analysis and transformation I move the models to EMF. The common metamodel consists of different separated metamodels. These are kept in different files, as suggested in [11]. The different metamodels are integrated via common elements and the instances via soft references.

The advantage of using SLE for the creation of DWHs is that experts can design and analyze all aspects of the DWH independently in adequate domain specific languages. Using the integrated metamodel the generation of a DWH is easier because all information and transformations are in that single model.

4 Research Method

My hypothesis will be validated via implementation. I will implement the described languages, metamodels, and transformations on basis of the MUSTANG platform. My prototype will be able to generate a configuration for a DWH. I will regenerate parts of the LIGA DWH that was developed by OFFIS. I will compare the steps taken. When using my approach these steps will be reduced.

5 Conclusion

Currently I have developed three languages for hierarchies, analysis schema, and data quality. I integrated them in a common metamodel. Based on this, transformations for generating multidimensional schemas in databases and integration interface with consistency were built. The next action is to develop languages for data sources and data integration as well as the extension of the common metamodel. The current state of my thesis shows that modeling and generation of data warehouses can be possible and reasonable with SLE.

References

1. Bauer, A., Günzel, H.: Data-Warehouse-Systeme. Architektur, Entwicklung, Anwendung. Dpunkt Verlag (2008)
2. Bizer, C.: D2R MAP - a database to rdf mapping language. In: WWW (Posters) (2003)
3. Bulos, D.: OLAP database design: A new dimension. Database Programming&Design Vol. 9(6) (1996)
4. Codd, E.F., Codd, S.B., Salley, C.T.: Providing OLAP to User-Analysts: An IT mandate. White paper, E.F. Codd Associates (1993)
5. Cook, S., Jones, G., Kent, S.: Domain Specific Development with Visual Studio DSL Tools (Microsoft .net Development). Addison-Wesley Longman, Amsterdam (2007)
6. Favre, J.M., Gasevic, D., Lämmel, R., Winter, A.: Editorial - software language engineering. IET Software 2(3), 161–164 (2008)
7. Gluchowski, P., Kurze, C., Schieder, C.: A modeling tool for multidimensional data using the adapt notation. In: HICSS. pp. 1–10. IEEE Computer Society (2009)
8. Golfarelli, M.: Data Warehousing Design and Advanced Engineering Applications: Methods for Complex Construction, chap. From User Requirements to Conceptual Design in Data Warehouse Design - a Survey. Information Science Reference (2009)
9. Golfarelli, M., Rizzi, S.: UML-Based modeling for What-If Analysis. In: DaWaK '08: Proceedings of the 10th international conference on Data Warehousing and Knowledge DiscoveryMazon. pp. 1–12. Springer-Verlag, Berlin, Heidelberg (2008)
10. Kelly, S., Tolvanen, J.P.: Domain-Specific Modeling: Enabling Full Code Generation. John Wiley & Sons (2008)
11. Kleppe, A.: Software Language Engineering: Creating Domain-Specific Languages Using Metamodels. Addison-Wesley Longman (2008)
12. Mazon, J.N., Trujillo, J., Serrano, M., Piattini, M.: Applying MDA to the development of data warehouses. In: DOLAP '05: Proceedings of the 8th ACM international workshop on Data warehousing and OLAP. pp. 57–66. ACM, New York, NY, USA (2005)
13. Pardillo, J., Golfarelli, M., Rizzi, S., Trujillo, J.: Visual modelling of Data Warehousing Flows with UML profiles. In: Pedersen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK. Lecture Notes in Computer Science, vol. 5691, pp. 36–47. Springer (2009)
14. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework. Addison-Wesley Longman (2008)
15. Teiken, Y., Brüggemann, S., Appelrath, H.J.: Interchangeable consistency constraints for public health care systems. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.C. (eds.) SAC. pp. 1411–1416. ACM (2010)
16. Teiken, Y., Flöring, S.: A common meta-model for data analysis based on dsm. In: The 8th OOPSLA workshop on domain-specific modeling (2008)
17. Teiken, Y., Rohde, M., Appelrath, H.J.: Model-driven ad hoc data integration in the context of a Population-based Cancer Registry. In: ICSOFT 2010 (2010)