DATA WAREHOUSE SYSTEM FOR ENVIRONMENTAL INFORMATION

Sebastian Günther, Otto-von-Guericke-Universität Magdeburg, Germany
klguenth@iti.cs.uni-magdeburg.de
Jorge Marx Gómez, Otto-von-Guericke-Universität Magdeburg, Germany
rauten@iti.cs.uni-magdeburg.de
Claus Rautenstrauch, Otto-von-Guericke-Universität Magdeburg, Germany
gomez@iti.cs.uni-magdeburg.de

ABSTRACT

This paper describes how a data warehouse can be constructed step-by-step from a collection of relevant queries to a logical data model and its implementation. The method presented here is based on a combination of several well-known procedures, particularly use of index numbers and the object-type method. The background of this research is a case study in the environmental planning department of a larger-sized German automobile industry company.

KEYWORDS: Data Warehousing, On-line analytical processing, Semantic modeling of Data Warehouses, Languages for Data Warehouses

1. INTRODUCTION

Despite a literal flood of articles on the topic of data warehouses, previously only a few authors have devoted efforts to modeling such systems. Generally speaking reports on data warehouse modeling are limited to the discussion of star-, snowflake- and galaxy schemata, without any further treatment of methodical questions [1] [2]. Only more recently have there been contributions from basic research [3] [4] and practice [5] [6] which go beyond this. The results presented here stem from a successful project realized in the frame of cooperation between the department of business informatics of the Otto-von-Guericke-Universität Magdeburg, Germany and the company Volkswagen AG in Wolfsburg, Germany. In this contribution an overall method will be presented for the semantic and conceptual modeling of data warehouses based on the project results. Before going into the method itself, the most important foundations of data warehouse systems will be briefly presented.

2. DATA WAREHOUSE SYSTEMS

The term data warehouse (DW) will be used to refer to a secondary database created with the aid of suitable extraction mechanisms from one or more operative databases. The data of the secondary database are thereby to be edited and aggregated so that they will be arranged in the most suitable manner with regard to future evaluations [7]. The first step in the generation of a DW is data transformation. All data relevant for a DW is extracted and transformed from operative databases in accord with the applicable set of rules. Transformation includes cleaning up, integrating and aggregating data, consequently one also speaks of data cleansing or data scrubbing [8]. The DW itself consists of a database for secondary data and a meta-database. The meta-database contains data on the structure of the secondary data of the DW. This includes information on the physical storage of the data and the structure and relationships (aggregation rules) of the data stored in the data warehouse. The meta-database thereby in a certain sense contains the "blueprint" of the DW. For a DW as a rule conventional relational database systems are employed. However, today there are also so-called multi-dimensional database systems (DBMSs) for the special requirements of DW solutions. These permit the storage of data in the

form of multi-dimensional cubes (hypercubes) and thus form an adequate foundation for the evaluation of the data with the aid of so-called OLAP systems.

3. ON-LINE ANALYTICAL PROCESSING (OLAP)

The evaluation of information stored in a data warehouse can be done in a variety of ways. In principle all the tools suitable for data analysis can be employed here. Thus the possibility exists, for example, to load the data in an Excel worksheet and further process it there [9]. Another method is to employ statistical software, with which a multiplicity of standard statistical procedures can be applied to information from the data warehouse. However, there is a class of end-user tools which have been specially developed for the evaluation of data warehouses: so-called systems for On-line Analytical Processing (OLAP) [10]. The OLAP concept stems from [11]. The basic ideas of OLAP are:

- *Multi-dimensional evaluation*: Multi-dimensionality was already mentioned in the context of data warehouse databases. To illustrate the significance of multi-dimensionality in data warehouse environments one- and multi-dimensional queries will be contrasted. A one-dimensional query is, e.g.: "How high were emission of pollutants in October 1998?" This query has only time as a dimension. An example of a multi-dimensional query is: "How high were emission of pollutants for the manufacture of product x for the months of January and February 1998 at our Heaven and Hell subsidiary in Hanover and Wolfsburg compared with the previous year?" This query includes the dimensions emissions, time period, enterprise, space and type of data (targeted- or actual values). A special characteristic of OLAP systems is the ability to efficiently deal with such multi-dimensional queries.
- *What-if analyses*: With what-if analyses scenarios can be played through based on actual and target figures. Thus, for example, a query of the type: "How would our turnover in Europe change if we had the same growth rates as in the USA?" can be answered with the aid of OLAP systems.
- *Drill-down techniques*: Dimensions can have a hierarchical structure. Figure 1 shows, for example, hierarchical structures for the time and enterprise dimensions. With OLAP systems it is possible to represent data beyond their hierarchical structures more finely or more roughly. The user has the possibility to carry out evaluations hereby at the level of detail which is relevant for his respective task.

Dimension *time*	Dimension *enterprise*
Year ▪ Quarter ▪ Month ▪ Week ▪ Day	Enterprise ▪ Central department ▪ Department ▪ Group ▪ Workplace

Figure 1: Hierarchical dimensions

The data structure underlying OLAP systems is the hypercube, on which the dimensions on the axes of an n-dimensional space are portrayed; each variable, also called a fact, forms a cell of the cube. Figure 2 shows a three-dimensional cube, in which each cell represents a sales (or turnover) value. Depending on the rotation of the cube, one receives the sales values for the relevant market (upper surface), quarter (front) or product (side view).

Figure 2: Data Cube

If one wants to do more with the model, one can also take a cross-section of the cube. Depending on the direction in which sections are taken through the cube, we can generate the different views of sales relevant to a product manager, financial manager or branch office manager (this technique is called "slicing"). Furthermore, a sub-cube can also be cut out as a so-called ad hoc view (this technique is called "dicing").

4. SEMANTIC MODELING OF A DATA WAREHOUSE

The modeling of DW is based on the expected evaluations, which are specified by the number of all (multi-dimensional) queries. Each query defines a fact, and the dimensions to which the fact is related. A statement collection consisting of (in the ideal case all) the queries formulated in natural languages forms the starting point for the modeling of DW systems. From the administrative viewpoint each fact describes an administrative index number. We understand an index number to be a number with concentrated information value for the diagnosis, planning, monitoring and guidance of a system. Index numbers can be calculated from other index numbers according to an index number formula, whereby hierarchically structured index number systems are created. Index numbers not used for the further calculation of index numbers are called peak index numbers, and index numbers which are not calculated from other index numbers are basic index numbers. An index number system forms the starting point for the formal modeling of the DW [4]. The facts defined in the statement collection are, however, as a rule only to a limited degree suitable for the definition of index numbers. This is caused by linguistic defects such as synonyms, homonyms, equipollences, vaguenesses or false designators. Before statements can be drawn on for the definition of an index number system, the linguistic defects of the statement system must be cleaned up through the creation of an obligatory terminological frame in a cooperation among the participating technical departments. This also holds for the terminology underlying the dimensions. The facts defined in the cleaned-up statement collection then form the basis for the definition of index number systems. As a rule facts define peak index numbers, but index numbers of lower levels can be relevant for evaluations. Index number systems should be refined sufficiently that the basic index numbers reference data elements of the operative databases. Dimensions are classes of descriptors which describe facts and are more precisely described through attributes [2]. The dimensions which are described in the cleaned-up statement collection are to be modeled as hierarchies, insofar as a hierarchical structure underlies them.

Statement collections, index number systems and dimensional hierarchies form the foundation for the creation of conceptional data models. Up to now there has been no agreement in the literature on which data modeling method is most suitable for meeting the requirements of conceptual DW data models [1]. In this project the suitability of the object type method (OTM) for the modeling of OLAP applications has being studied [12]. It was shown that only a few syntactic constructs and an enlargement of the OTM are needed for this. The applied constructs are the following:

- object types which are to be equated with entity types,
- the connection which summarizes the relationships between independent and differently designated objects to a new object domain with its own concept,
- the aggregation with which objects of a concept are grouped (summarized) with objects having a different concept.

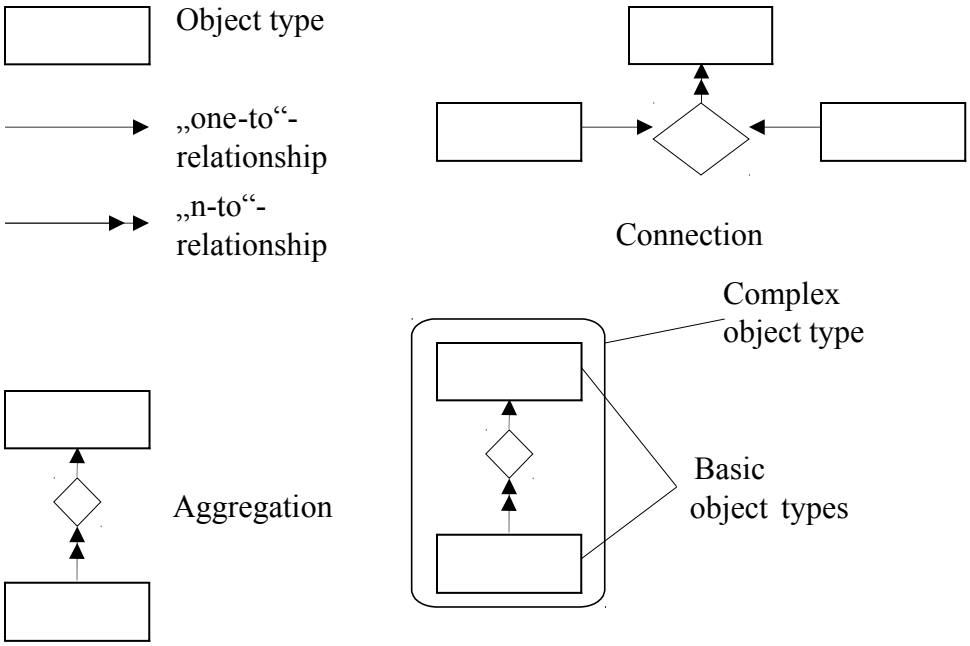

Figure 3: Employed constructs of OTM

The enlargement concerns the introduction of complex object types [13]. A complex object type summarizes the basic object types which have a hierarchical relationship to one another in a dimension. Basic object types are object types which relate on a deeper hierarchical level to a complex object type. Of course the OTM as a linguistic construct offers the so-called super object type, however, it is hereby a matter of a logical object type which is not implemented with the later realization, while the complex object type does have a physical realization. Figure 3 shows the linguistic constructs used by the OTM.

The next step is to come from statement collection, index number system and dimensional hierarchies to logical data models. Every statement contains a fact with the dimensions in the context of which it is evaluated and thereby exactly specifies an OLAP hypercube. A data model with the star schema is obtained if the fact and the dimensions are represented as respectively their own object type, and the dimensions are related in a 1:n connection to the fact.

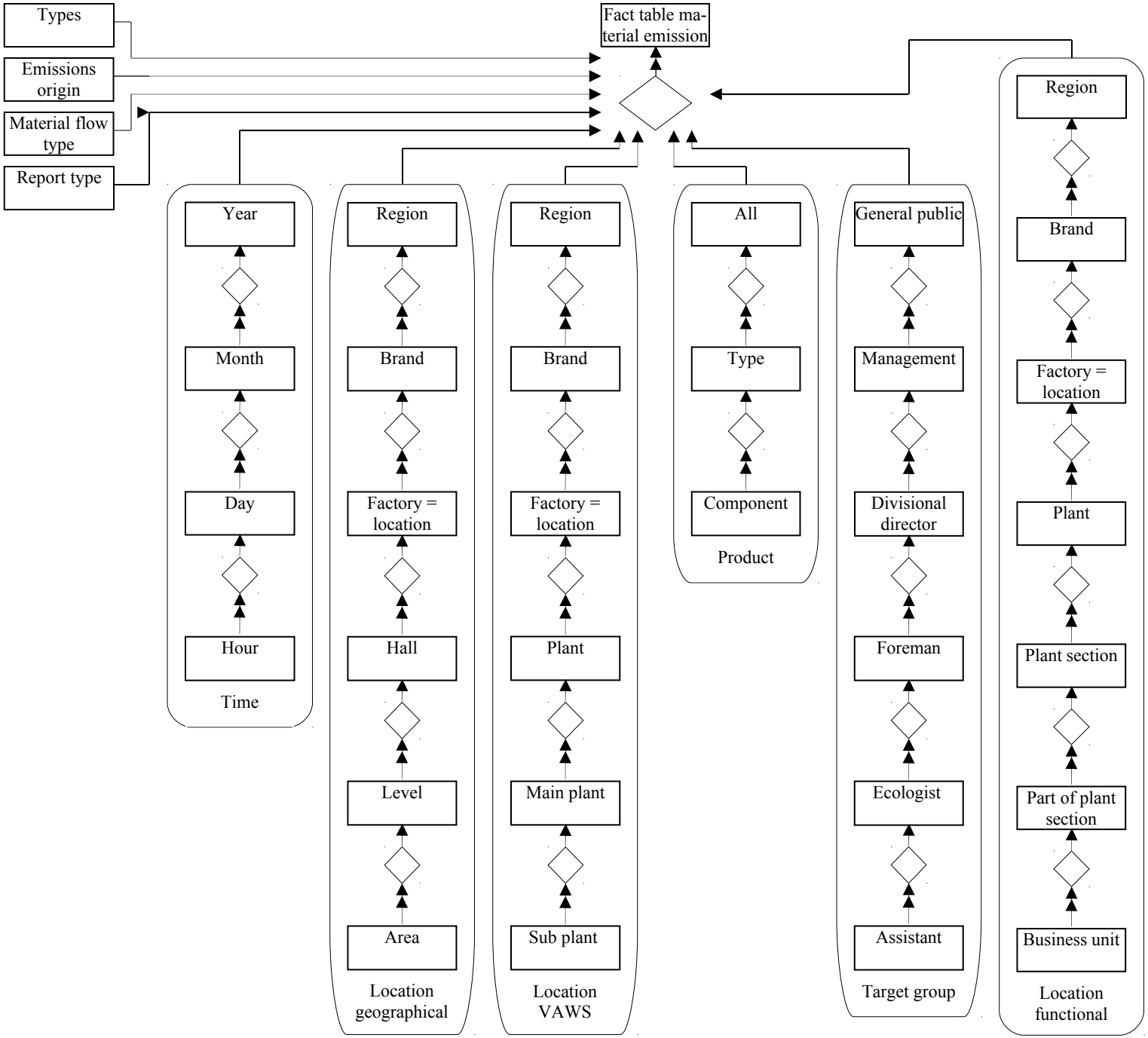

Figure 4: Employed constructs of OTM

If one opens up the dimensions according to the modeled dimensional hierarchies, one obtains the snowflake schema. Each dimensional hierarchy is modeled as a complex object type whose basic object types are connected with each other through aggregation. Figure 8 shows in OTM notation the snowflake schema for the fact "Emissions" from the case study.

An important domain which, however, receives too little attention in the context of modeling DW, is the modeling of integrity rules. Integrity rules specify the permissible changes in the state of the database. A DW database experiences changes in state during routine operation only when operative data are aggregated and added to the DW database, since OLAP tools can only read the database. Integrity rules thereby define the semantics of data extraction for DW systems. A popular method of modeling integrity conditions is the use of ECA rules (Event, Condition, Activity). An ECA rule must be interpreted so that a specified activity will be carried out precisely when the event occurs and the condition is fulfilled [14]. Events depend with DW systems on the transformation strategy [15]. If data propagation is applied as a strategy, then a database event, i.e., a change in a basic table of the operative database, leads to an immediate transformation in the DW. If a periodic strategy is employed, a temporal event, i.e., reaching a specific point in time or respectively the expiration of a specific period, causes a data transformation. Which strategy will be used to carry out the transformation must be respectively established for different domains of the DW, the so-called data marts. Conditions are given in the form of predicates whose truth value is checked as soon as the specified result occurs. Here, e.g., exceptions can be made ("Monday no data will be transformed"). Conditions are optional. The activity describes the transformation function to be carried out, i.e., which operative data will be aggregated in what manner into secondary data (i.e., facts). The aggregation rules are documented in the index number system as index number formulas. In addition, with the application of the data-update transformation strategy it is to be noted that only adjusted basic data are transformed in DW.

Depending on its complexity, a DW is usually not constructed as a whole, but rather step-by-step. For this purpose it is divided up into individual, maximally discrete domains, so-called data marts (Demarest 1993). Viewed in themselves these are small, transparent DWs. The DW then arises through the integration of these individual data marts. For the division of a complex DW into data marts there are two strategies: Summarized in a data mart are all objects belonging to the same business procedure (in this case the division is oriented to the structuring of business processes of the company). Or all objects of a company organization structure domain are summarized in a data mart. Which of the two variants is finally selected depends on company-specific preferences. The division of the DW into data marts should, however, occur only after the cleaning up of the statement collection. This is because cleaned-up statements greatly facilitate later integration of the individual data marts.

5. SUMMARY: A PROCEDURAL MODEL FOR THE MODELING OF DW SYSTEMS

The aim of this project was the modeling, i.e. the preparation of a star- and snowflake schemata, as a data mart for a selected area within the environmental management of Volkswagen AG. The basic idea in the development of the approach for modeling DW systems chosen for this contribution is to rely on proven concepts wherever possible, instead of reinventing the wheel.
For the early phase of modeling use was made of the linguistic-critical approach [16] and of such simple structures as index number systems and dimension hierarchies. The conceptual model was then developed with a minimally expanded OTM, whereby models developed with OTM could be transferred into entity-relationship models with little loss [13]. Implementation can then be

largely realized through generation mechanisms using suitable tools. Figure 4 shows in summary the procedural model for DW modeling of this contribution.

Figure 3: Procedural model

8. REFERENCES

[1] Kimball, R.: A Dimensional Modeling Manifesto. DBMS 10 (1997) 9, http://www.dbmsmag.com/9708d15.html.

[2] Raden, N.: Modeling the Data Warehouse – Introduction, General Description. http://www.strategy.com/dwf/raden/iw01961.htm.

[3] Bulos, D.: A New Dimension. OLAP Database Design. Database Programming & Design 9 (1996) 6.

[4] Gabriel, R., Gluchkowski, P.: Semantische Datenmodellierungstechniken für multidimensionale Datenstrukturen. HMD 34 (1997) 195, S. 18-37.

[5] Lehmann, P., Ellerau, P.: Implementierung eines Data Warehouse für die Verpackungsindustrie. HMD 34 (1997) 195, S. 76-93.

[6] Altenpohl, U., Huhn, M., Schwab, W., Zeh, T.: Datenmodellierung Data Warehouse. Interner Bericht der UAG GSE Rhein-Main 1997.

[7] Rautenstrauch, C.: Effiziente Gestaltung von Arbeitsplatzsystemen. Bonn u.a. 1997.

[8] Widom, J.: Research Problems in Data Warehousing. In: Proceedings of the 4th International Conference on Information and Knowledge Engineering (CIKM) 1995, http://www-db.stanford.edu/warehousing/publications.html.

[9] Business Objects: Business Query for Excel. Product information 1995.

[10] Jahnke, B., Groffmann, H.-D., Kruppa, S.: On-Line Analytical Processing (OLAP). Wirtschaftsinformatik 38 (1996) 3, S. 321-324.

[11] Codd, E. F., Codd, S. B., Salley, C. T.: Beyond Decision Support. Computer World vom 26.09.1993.

[12] Ortner, E.: Aspekte einer Konstruktionssprache für den Datenbankentwurf. Darmstadt 1983.

[13] Inan, Y.: Semantische Modellierung komplexer OLAP-Anwendungen mit der Objekttypenmethode (OTM) – Grundlagen und Fallstudie. Diplomarbeit Universität Konstanz 1997.

[14] Dayal, A. P., Buchmann, A. P., McCarthy, D. R.: Rules are Objects Too: A Knowledge Model for an Active Object-Oriented Database Management System. In: Dittrich, K. R. (ed.): Advances in Object-Oriented Database Systems. Berlin et al. 1988, pp. 129-143.

[15] Kirchner, J.: Datenveredelung im Data Warehouse – Transformationsprogramme und Extraktionsprozesse von entscheidungsrelevanten Basisdaten. In: Mucksch, H., Behme, W. (Hrsg.): Das Data-Warehouse-Konzept. Wiesbaden 1996, S. 265-299.

[16] Ortner, E., Söllner, B.: Semantische Datenmodellierung nach der Objekttypenmethode. Informatik Spektrum 12 (1989) 1, S. 31-42.