METHOD FOR USER ORIENTED MODELLING OF DATA WAREHOUSE SYSTEMS

Lars Burmester
Institute of Management Information Systems, Philipps-University of Marburg, 35032 Marburg, Germany

Matthias Goeken
Dep. of Management Information Systems, HfB – Business School of Finance and Management, 60314 Frankfurt, Germany

Keywords: Data Warehouse, Multidimensional Modelling, Process Model, Modelling Framework.

Abstract: The paper describes a method for data warehouse development. One critical success factor of data warehouse development is determining information requirements. Hence, the method focuses on gathering of requirements and information needs of the users first. An extended data warehouse architecture and a technique for decomposition of the system serve as a developing framework. On the one hand this framework is used to define releases (builds) of the system, which is indispensable for an incremental development process. On the other hand it defines intermediate and final work products (artifacts) that are produced and used during further development stages. Starting with information requirements elicitation, each increment is realized through a series of data models which successively are transformed from conceptual to logical level. These logical data models are then used for implementation as well as for the modelling of ETL processes.

1 INTRODUCTION

During the past years, data warehousing has been an object of intense research. As one result, quite a few development methodologies were introduced, which roughly could be categorized as data-driven, goal-driven and user-driven (List et al., 2002). At data-driven approaches, the corporate data model or at least the data models of OLTP-systems are the starting point of the development process. ETL-processes, data staging and OLAP-cubes are built on top of that data models. User requirements are considered last, if considered at all (Golfarelli et al., 1998; Golfarelli and Rizzi, 1998). Goal-driven approaches derive the information requirements to be satisfied by the data warehouse from business-processes (Kimball and Ross, 2002) or overall company goals (Prakash and Gosain, 2003). In user-driven approaches the information requirements are gathered directly from the user, while the development of data models and ETL-processes follows.

Data- and goal-driven approaches bear the latent risk of not meeting the user requirements, leading to a lack of system-acceptance and in worst case to a complete failure of the developing project. Therefore user-driven developing approaches seem most promising for successful completion of data warehouse projects.

Following the described user-driven approaches, the elicitation of the information requirements from users regularly turns out to be a key problem (Holten, 1999). These problems are issued by Valusek and Fryback, who categorize them as “obstacles… *within* an individual user, *among* users, and *between* the user and those responsible for system development” (Valusek and Fryback, 1985; Browne and Rogich, 2001). Between-obstacles result from the fact that system developers and users speak different languages. This “user-developer cultural gap” leads to a “linguistic gap” which has to be bridged by applying suitable methods and techniques during requirements elicitation (Ortner, 1995). In addition, users often cannot explicit their requirements, which is because they are not instantly aware of them (within-obstacle) and requirements are still unstable during the development process due to business dynamics.

Despite the described challenges, the central technical challenge in data warehousing consists in building up an enterprise-wide integrated and consistent database, which turns out to be a complex, time consuming and expensive effort (Hackney, 1998). To tackle these problems, it is often recom-

Burmester L. and Goeken M. (2006).
METHOD FOR USER ORIENTED MODELLING OF DATA WAREHOUSE SYSTEMS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 366-374
DOI: 10.5220/0002456303660374

mended to start with the construction of data marts (e.g. for small organizational units like departments or regional units) and incrementally improve them towards the final data warehouse system ("think big, start small"). Though, support for comprehensive planning and development of enterprise-wide data warehouse systems is seldom found (see Hackney, 1998 and Ong, 1999 for older approaches of evolving data warehouse architecture).

Taking a look at existing user-driven methodologies reveals that some promising approaches exist, which tackle the described problems of requirements elicitation, especially the *within* and *between* obstacles (Bonifati et al, 2001; Winter and Strauch, 2003). Though, there are some deficits, regarding support of comprehensive system planning and modelling. Strauch clearly states the need for representation and verification of requirements, using multidimensional modelling languages, but gives no instructions how this should be accomplished. Bonifati et al. use idealised star schemas to represent user-requirements, consisting of basic multidimensional modelling elements, but lack constructs for dimension hierarchies, which in particular are crucial for OLAP cubes (see section 3.2.2 for details). While Strauch completely neglects implementation aspects, Bonifati et al. consider the reconciliation of user requirements and information supply as a trial and error process, finding the best match between the ideal star schemas and candidate star schemas derived from operative data schemas.

In the following parts of the paper a development method is introduced which considers the above mentioned challenges. Section 2 introduces an advanced data warehouse architecture which serves as a reference framework for the method. Section 3 treats the development method itself, starting off with systems planning and the process model (Section 3.1). The elicitation of information requirements, conceptual modelling and validation of the resulting structures is treated in section 3.2. Section 3.3 describes the successive transformation of data models towards a physical implementation and the implementation of ETL processes. Conclusions and discussion are found in section 4.

2 ADVANCED DATA WAREHOUSE ARCHITECTURE

2.1 Modelling Framework

In this section an advanced data warehouse architecture is introduced, which serves as a modelling framework for the method. A closer look on the data warehouse layers reveals the stepwise data flow from data sources towards analytical applications situated in the presentation layer (see figure 1). This physical perspective only covers a part of the tasks a data warehouse has to cover and is insufficient to capture the users information needs. In fact, a conceptual perspective for representation of the business tasks of a data warehouse is missing (Vassiliou et al., 2000; Jarke et al., 1999; Quix, 2003). To utilize data from operational Systems (OLTP) for information supply and decision support not only from a physical but also from a conceptual point of view, it is necessary to explicitly describe each layer with an own conceptual (data) model. This integrates the so far isolated data flow perspective with an information-use perspective. The conceptual perspective respectively the conceptual (data) models support the user/analyst interaction and can also be seen as documentation of development steps. In terms of language they can be seen as "close-to-user", because they consist of modelling constructs that are easy to understand. In addition they are a decisive input for following stages of the development process (Wand and Weber, 2002).

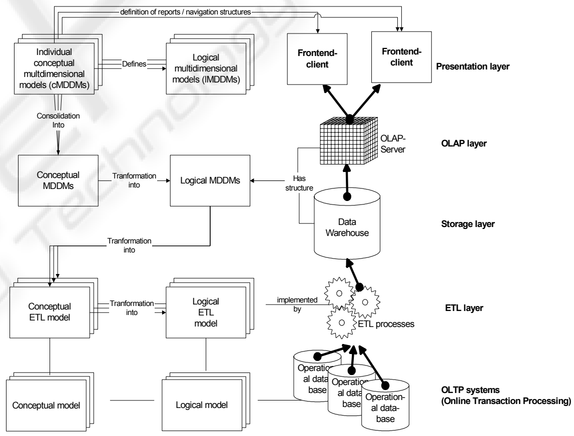

Figure 1: Extended data warehouse architecture.

To simplify the presentation it is assumed that the data models of the storage layer and the OLAP-layer don't differ. This means the storage layer already consists of the data structures that are necessary for relational OLAP. If a reconciled data layer exists, the ETL processes described in section 3.4 will access this instead of the operational source systems.

The described architecture is similar to the "meta data framework" developed in the DWQ project on (Vassiliou et al., 2000; Jarke et al., 1999; Quix, 2003). It differs in two points: on the one hand an enterprise data model is assumed as given. Such enterprise data models often don't exist in reality and are assumed as problematic, so these should not be considered in this paper (Stahlknecht and Hasenkamp, 2005; Schelp, 2001). Instead, individual user requirements should be the starting point of the method. On the other hand a different integration approach is followed. While the DWQ project follows the “local as view” approach, our approach considers the content of the data warehouse as views on the operational data sources. This “global as view” approach can also be used as a description of the data flow from data sources into the integrating system.

2.2 Developing in the Framework

Conceptual multidimensional data models (cMDDM) act as the starting point of the method (see figure 1, left uppermost corner) and are used for elicitation of information requirements and can be seen as the work products of this stage. During this development stage, single user (or user group) requirements and final specifications are treated separately. This separation corresponds to the often demanded addressee-orientation for management information systems and allows the traceability of requirements to their origin. By using individual cMDDM, addressee-appropriate reports, navigation and alternatives for analyzing the information space are provided. In addition these cMDDM could be consolidated and transformed into logical multidimensional data models (lMDDM; see figure 1, centre column). Furthermore lMDDM provide input for conceptual ETL models, which can be used for communication with the administrators of operative data sources. Conceptual ETL models can be seen as requirements of the data warehouse systems, which should be fulfilled by OLTP systems. During the development process, conceptual ETL models are transformed into logical ETL models, which are then physically implemented (see figure 1, 3rd row from above).

This architecture allows a structured process for data warehouse development and also secures documentation on each semantic level. Furthermore the architecture provides a framework for integration of different modelling approaches existing in data warehouse development. The goals pursued by (conceptual) models, like documentation, input for later development stages or communication with the user, could be hardly realized by using just one modelling language during the whole development process (Wand and Weber, 2002). In further description of our method, we exemplarily use ADAPT (Bulos and Forsman, 2002) for conceptual multidimensional modelling and the concept of Vassiliadis et al. (Vassiliadis et al., 2002a) for ETL modelling. Of course, other modelling approaches fit as well, so that preferred or more suitable ones could be chosen by the developer.

3 METHOD

3.1 Systems Planning and Process Model

3.1.1 Systems Planning

To reduce the complexity of developing enterprise-wide data warehouse systems, it is often suggested to start off with data marts (“start small”) and incrementally improve them towards an overall target system. Incremental development means to focus a well defined goal, which should be realized by stepwise enhancement of incomplete parts (Gilb, 1988; Larman and Basili, 2003). This allows handling the dynamics oft the business environment as well as fine-grain-planning the delivery of increments (builds). Misunderstandings can be resolved during the development process and early testing also helps finding out whether the available information supply matches the actual information demand. Following an incremental approach requires that a target system must be decomposable. If this is given, build stages can be appointed into a build plan (Goguen and Linde, 2003). The resulting loops in the development process can then be seen as planned iterations.

In the introduced development method, the overall target system is decomposed using two decomposition criteria. Information and decision objects are used as a business criterion on one side and the layers of the data warehouse architecture are used as technical criterion on the other side. Information or decision objects represent relevant aspects of the business, which can be derived e.g. from the organizational structure or a performance measurement system. They define the data warehouse subjects on an abstract level. Combining this business perspective with the layers of the data warehouse architecture leads to development objects, which represent modules of the overall target system (see figure 2).

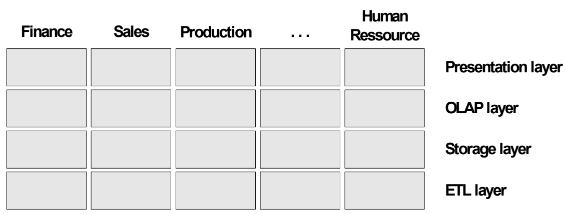

Figure 2: Developing objects as a result of the system decomposition.

These development objects are central subjects for planning builds and prototypes. The build plan shows which objects are realized in which order. For a more precise planning of the development stages of a development object, it is viewed in form of the introduced advanced data warehouse architecture. In the build plan, milestones could be defined at which a certain development process has to be completed. Figure 3 shows a build timetable for an example project.

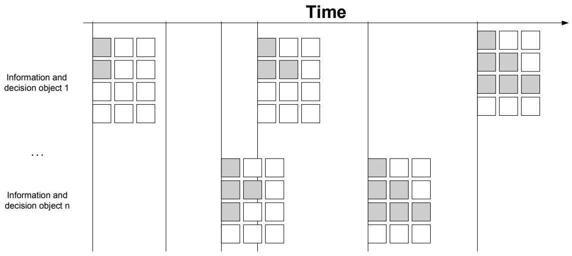

Figure 3: Exemplary build timetable of multiple subsystems.

3.1.2 Process Model

The process for developing a build consists of several steps. The starting point of the process is the requirements elicitation stage in which the information requirements of each individual user towards the target system are acquired. At the beginning of the design stage these requirements are converted into individual cMDDM and are consolidated into a conceptual overall view during the stage. At the end of the conceptual design stage the overall cMDDM is transformed into a logical data model (lMDDM) which forms the basis for the construction of a validation prototype. During the validation stage the so far generated conceptual structures are validated by the future users. The constructed prototype supports this step. If the prototype is rejected, the process will loop back to earlier stages, depending on the reasons for rejection. A positive validation triggers the specification stage. The generated lMDDM specify the formalized information demand, which has to be reconciled with the information supply in the ETL layer. The transformation of the lMDDM into conceptual ETL models as well as logical ETL models based on that, are the foundation for their final physical implementation. An overview of the process model for the realisation of each development object is provided in figure 4.

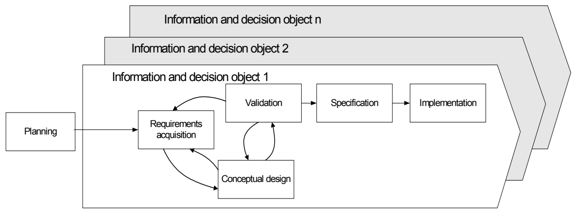

Figure 4: Process model for the realisation of developing objects.

3.2 Requirements Acquisition and Conceptual Design

3.2.1 Requirements Acquisition

The requirements acquisition stage represents the starting point for the cycle in which the information need is acquired and formalized. First, the information demand of each individual user (or user group) is acquired. On the one hand it should be found out, which facts and measures quantify an information and decision object. On the other hand qualifying perspectives on these measures or facts are created by identifying user-defined dimensions and hierarchies (Lehner, 2003).

Due to the high level of abstraction encountered in conceptual multidimensional modelling languages (e.g. the Dimensional Fact Model (DFM), ADAPT and others (Bulos and Forsman, 2002; Golfarelli et al., 1998; Abello et al., 2002), they are often not suitable for communicating with inexperienced users about their requirements. The overcoming of the above mentioned "linguistic gap" between analysts and users is aggravated by using too formalized modelling languages, especially when used for supporting the articulation of information needs. In particular, it often turns out to be difficult to explain the differences of multidimensional modelling constructs to users (for an empirical study which leads to similar results see Nordbotten and Crosby, 1999). Therefore analyzing and interpreting the users information needs remains a key task for the developer. For communication with the user, interrogatives or so called "w-questions" have turned out to be quite useful, because they describe the constructs of multidimensional modelling in colloquial speech (Browne and Rogich, 2001; Stahlknecht and Hasenkamp, 2005; Quigley and Debons, 1999). Interrogatives can be interpreted as a basic grammar (construction plan for sentences) as it has been suggested by Ortner for the method neutral conceptual design (Ortner, 1995). For graphical representation, simplified models could be used, which use only a subset of common multidimensional modelling elements (facts, measures, dimensions, some early hierarchies). In common performance measurement concepts, dimensionality is often disregarded, so that confronting "inexperienced" users with multidimensionality might lead to the above described obstacles. The confinement on few constructs of multidimensional modelling may lead to a better understanding and enables the user to concentrate on general aspects.

The results of the requirements acquisition stage are the information needs of individual users or user groups. A set of simplified conceptual models and interrogatives are the work products, which are passed to the next stage.

3.2.2 Conceptual Design

In the following section the design process is introduced which realizes the information requirements within the advanced data warehouse architecture.

Starting-point are the results of requirements acquisition, which represent the information need of the single users. The first step consists of the transforming the simplified conceptual models into common multidimensional models (e.g. in ADAPT) and their consolidation into a conceptual overall view. All objections against the use of common multidimensional modelling languages are omitted because these models are no longer used for user-analyst-interaction, but for documentation and as a central input for further transformations into lMDDM (see figure 1 again).

To illustrate the process, the development of parts of a sales information system should be taken as an example for the rest of the paper. During the requirements acquisition stage a couple of simplified conceptual models for single users were generated. Further the future system can be described by the interrogative "Which amounts of money (in €), of which kind (e.g. turnover, discounts, promotion costs) have flown doing business with which customer (e.g. wholesalers, retailers) in which region (e.g. Nielsen-area, state) at which point of time (e.g. fiscal year, calendar year) ?". The ERP-system as well as a not integrated marketing information system should be seen as the data sources of the future system.

The transformation of the simplified conceptual models into common multidimensional models here exemplarily should lead to ADAPT models, starting off with the dimensions (Bulos and Forsman, 2002). The formalisation of a dimension requires representing the elements of the hierarchical structure. In common multidimensional modelling languages dimensional hierarchies are often modelled collapsed because a full display of the hierarchy would soon become too hard to work with. Hence, dimensions are modelled as generalised, abstracted dimension levels. Simple balanced hierarchies can be transformed directly into a multidimensional model. Dimensions with more complex structures, like multiple or unbalanced hierarchies as well as non-additive roll-ups, couldn't be transformed in the described manner, which than requires an uncompressed view of the hierarchy again (Schelp, 2000).

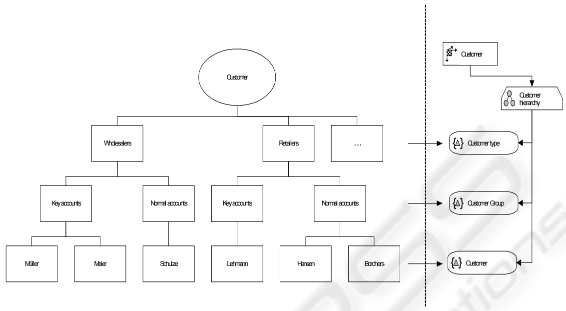

Figure 5: Transformation of a simple balanced hierarchy into the ADAPT notation.

After the transformation of the simplified multidimensional models of the single users into formalized cMDDM, these are consolidated into a multiple user cMDDM. The resulting model represents the cMDDM of the storage/OLAP layer. Consolidating the facts and measures usually does not impose bigger problems, because they usually origin from well known corporate performance measurement systems. The consolidation of the dimensions could be seen as a major challenge in the design process. Hereby it is crucial to recognize different views on a dimension and to consolidate these into one consistent dimension. In the following the consolidation process should be illustrated with an example, which shows the consolidation of different views upon a time dimension. Here, a finance officer needs a deeper time hierarchy than the sales officer. The latter uses the corporate fiscal hierarchy, while the analyst uses the calendar hierarchy. Figure 6 shows the consolidation of the described views upon the time dimension.

After consolidating the individual cMDDM, the resulting multiple user cMDDM has to be transformed into the logical data model (lMDDM). Though independent from physical implementation, logical models of multidimensional data structures are aligned to the chosen database technology. Because of the wide spread of relational database technology, the logical model of the multidimensional data layer regularly is modelled as variants of the star schema.

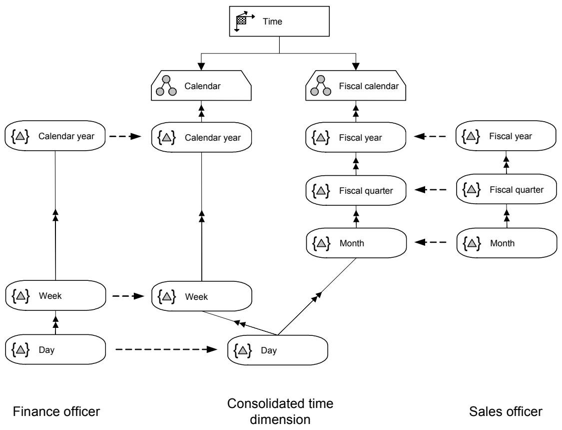

Figure 6: Consolidation of different views on a time dimension.

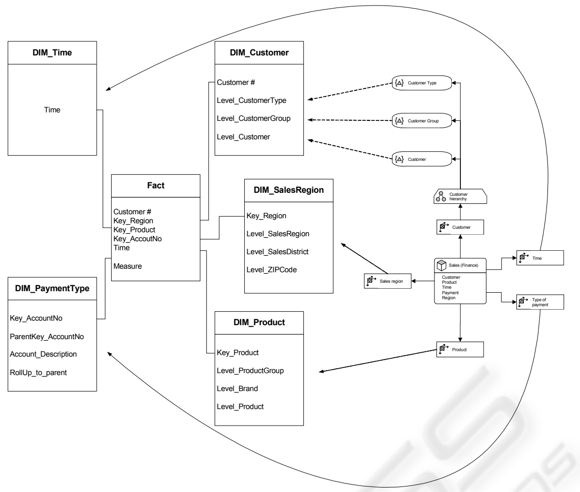

Figure 7: Mapping of an ADAPT model onto a simple star Schema (lMDDM).

Deriving logical data models from conceptual models constitutes a critical step in the development process, because this transformation always means the loss of semantics between the two models (Schelp, 2000; Blaschka, 2000; Hahne, 2002). For that reason the danger of mis-mapping is biggest in this stage of the process. A close look on the transformation of facts and measures shows that their transformation could be handled in some few steps, while the transformation of dimensions is a lot harder. Though most of multidimensional modelling languages are capable of representing complex hierarchical structures, like multiple or unbalanced hierarchies or non-additive roll-ups, just a few approaches exist that secure transformation into logical multidimensional data models without the loss of too much semantics (Herden, 2001).

For an example of mapping cMDDM to logical ROLAP-models see (Blaschka, 2000). There this is done by mapping the elements of the meta-models of conceptual multidimensional modelling languages onto the elements of the relational model. Doing so, transformation rules, like “For each fact or dimension exactly one table should be created”, could be derived. By applying these rules, a conceptual multidimensional model could be formally transformed into a simple star schema. Figure 7 shows the example of mapping an ADAPT model onto a star schema (logical MDDM).

To optimize a ROLAP-solution in terms of performance, maintainability and storage, methods like indexing, partitioning tables as well as materialized views can be used (Herden, 2001; Peralta and Ruggia, 2003; Lehner, 2003). The discussion on the mentioned methods should not take place in this paper, because just the realization of informational requirements should be demonstrated.

The lMDDM represents the informational requirements of a data warehouse system using table structures. To validate these generated structures, they have to be filled with data. Because of the incremental approach and the usage of prototyping, for now, simulation data should be used during early iterations, because the modelling and implementation of ETL processes result in major effort (Vassiliadis et al, 2002a for example estimate the share of total development at 80%). By further advancement of the development process, real data should by used to validate prototypes and secure a smooth transition to operational use.

3.3 Validation

The validation of the conceptual model and the logical structures is realized in one single step. The future users could validate, if the generated structures correctly represent the universe of discourse, regarding measures and dimensions. If the prototype should be fully or partly rejected, a loop-back to earlier stages takes place. To determine the stage which should be looped back to, a verification of the artifacts of passed stages is necessary. Figure 8 shows the procedure during the validation stage. If a prototype is accepted, the development process advances to the specification stage.

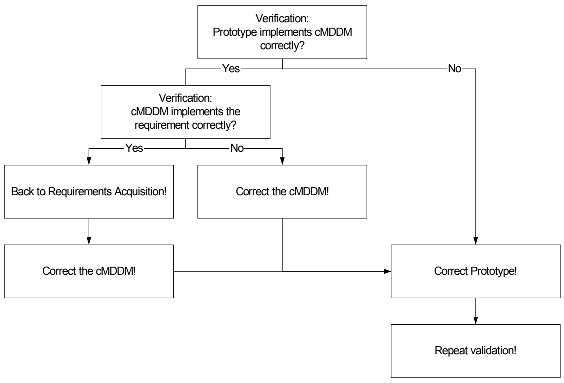

Figure 8: Procedure when prototype is rejected.

3.4 Specification and Implementation

If the prototype is accepted during the validation stage, the generated logical structures represent the formalized information demand. The information supply consists mainly of data from internal OLTP systems and is enriched by additional external data. The reconciliation of information demand and information supply in data warehouse systems takes place in the ETL layer as ETL processes.

As before in the OLAP and storage layer, the data acquisition layer could also be represented by conceptual, logical and physical models (Vassiliadis et al., 2002a, Trujillo and Lorán-Mora, 2003). Equal to the simplified multidimensional data models used during the requirements elicitation stage, conceptual ETL models could be used for communication with the administrators of OLTP systems.

The approach for conceptual modelling of ETL processes from Vassiliadis et al. proposes the analysis of user requirements as well as an analysis of the structure and contents of the data sources, before starting the modelling process (Vassiliadis et al., 2002a). The user requirements as target of the ETL process already exist as table definitions (lMDDM) (in the following referred to as data consumer). On the opposite internal and external data sources represent the information supply (in the following referred to as data provider). The creation of a conceptual ETL model takes place in a three step process (Simitsis and Vassiliadis, 2003). In the first step adequate data providers have to be chosen. Following that, delivery and mapping relations between potential data providers and data consumers have to be concretizised. This second step is crucial for the success of the modelling process due to the heterogeneity of source systems. It requires intense interaction with the OLTP systems administrators as well as comprehensive testing, to secure the adequate filling of the target tables. After modelling the mapping relations, the process can be annotated with runtime constraints, like execution plan, supervision and logging settings, as well as exceptions and error handling. Additional requirements can be mentioned (e.g. the level of data freshness) which have to be considered during the physical implementation.

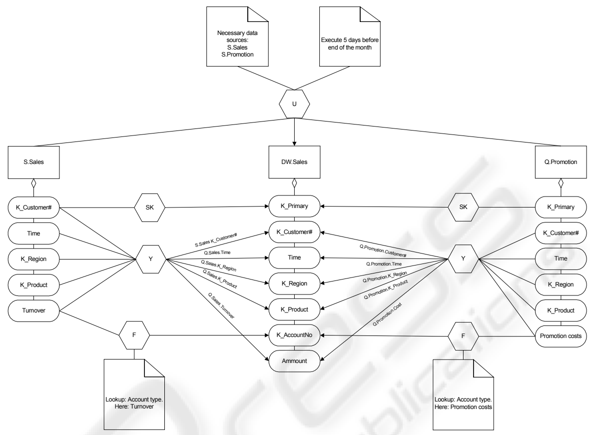

Figure 9: Example of a conceptual ETL model.

An example of a conceptual ETL model is shown in figure 9 (using the notation of Vassiliadis et al., 2002a). The data consumer in this model is a fact table (DW.Sales) of the sales information system, which is supplied from two sources (S.Sales and S.Promotion). In the example further transformations, besides simple mapping processes (Y (Aggregations)) can be found. On the one hand a unique surrogate key is assigned (SK), to avoid contradictory values for the primary key in the fact table. On the other hand a transformation function (F) is applied, which assigns a qualifying account-type to a measure. For example, the account type 'promotion costs' is assigned to records originating from the data source 'promotion' and the account type 'turnover' is assigned to records from the data source 'sales'. Further annotations were made regarding the implementation of the process, e.g. execution restraints or the necessary data sources (U).

Vassiliadis et al. state that a logical ETL model consists of the data flow process from the data sources towards the data warehouse and several ETL activities, which can be seen as logical abstract of physical code (Vassiliadis et al, 2002b; Simitsis, 2003). Basic elements of the logical ETL model are ETL activities, which can be described on multiple levels. From the perspective of the meta model, ETL activities consist of several components, like name, input and output and the relationship between them. Instances of these meta-activities are called template activities (e.g. push, join, not-null-check etc.). Standard activities could be customized to fit the requirements of a concrete ETL process. These adapted ETL processes can be seen as an instance of the template.

ETL processes are described in so called ETL scenarios, depicting a sequence of ETL activities, which sketch the data flow between source and target records. Figure 10 shows the ETL scenario of the introduced conceptual ETL model. The relevant records are loaded from the source databases (S.Sales, S.Promotion) into the tables of the data staging area (DS.Sales, DS.Promotion) via FTP. This is followed by two ETL activities, the assigning of a unique surrogate key and the lookup of the account type, including the logging of failed transactions. The ending activity unites the records from both data sources and loads them into the data warehouse table (DW.Sales).

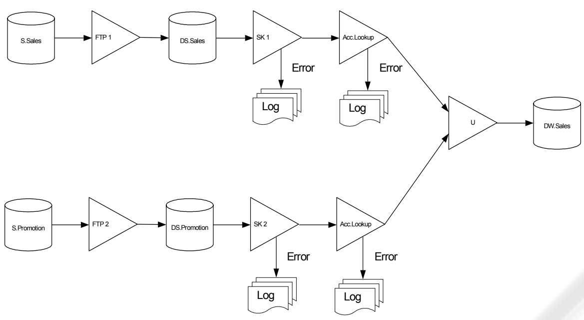

Figure 10: Logical ETL Scenario.

4 CONCLUSIONS

This paper introduced a method for user oriented development of enterprise-wide data warehouse systems. The focus lay on the decomposition of the system, which is used for planning builds for the incremental design. Additionally a process model is introduced, which defines stages and working products of the development process.

The method proved to be useful and handy in the project EiSFach, which the authors worked in. The projects subject is the design and implementation of data warehouse based performance measurement and management support systems for the Philipps University of Marburg (Germany). Due to uncertainty in laws and novelty of the field of performance measurement in universities, the specification and dynamic of requirements were the key success. The introduced method helped to tackle these problems, while the potential usefulness of the method in a more complex business environment still has to be proved.

Concrete techniques for elicitating the user requirements were just mentioned. Possible techniques should concretizised and assigned to the several stages during further improvement of the method. Therefore contingency approaches for situation-specific choice of methods and techniques of requirements elicitation could be useful. This would allow a rational choice between available techniques. In addition, techniques for inter-model transformation should be more formalized, whereas the possibility of tool support would arise.

Though the information requirements are the key success factor of data warehouse development, other requirements, like performance, usability and maintainability, were not regarded. The enhanced data warehouse architecture from section 2 of this paper can be used as a framework for modelling and implementation this requirements. For example performance requirements can be realized alternatively through optimization of the logical data model (partitioning of tables or fractional design) or through physical techniques (indexing). These advanced requirements should be more regarded on further improvement of the method.

REFERENCES

Abello, A., Samos, J. and Saltor F., 2000. *A Data Warehouse Multidimensional Data Models Classification*. Grenada: Technical Report LSI-2000-6, Dept. Llenguages y Sistemas Informáticos.

Blaschka, M., 2000. *FIESTA: A Framework for Schema Evolution in Multidimensional Databases*. Dr. Technical University Munich.

Bonifati, A. et al., 2001. Designing Data Marts for Data Warehouses. *ACM Transactions on Software Engineering and Methodology*, 10 (4), pp. 452483.

Browne, G. and Rogich, M., 2001. An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. *Journal of Management Information Systems*, 17 (4), pp. 223–249.

Bulos, D. and Forsman, S, 2002. *Getting started ADAPT – OLAP Database Design*. San Rafael: Symetry Corp, White Paper.

Gilb, T., 1988. *Principles of Software Engineering Management*. Reading et al.: Addison-Wesley.

Goguen, J. A. and Linde, C., 1993. Techniques for Requirements Elicitation. *Proceedings of IEEE International Symposium on Requirements Engineering 1993 in San Diego*.

Golfarelli, M. and Rizzi, S., 1998. A Methodological Framework for Data Warehouse Design. *Proceedings of the ACM first international Workshop on Data Warehousing and OLAP 1998 in Washington*.

Golfarelli, M., Maio, D. and Rizzi, S., 1998. Conceptual Design of Data Warehouses from E/R Schemes. *Proceedings of the Hawaii International Conference On Systems Science 1998 in Kona*.

Hackney, D., 1998. *Architectures and Approaches for Successful Data Warehouses*. [Online]. http://datawarehouse.ittoolbox.com/documents/document.asp?i=815 [Cited 19 January 2006].

Hahne, M., 2002. Transformation mehrdimensionaler Modelle. In E von Maur and R Winter, eds. *Vom Data Warehouse zum Corporate Knowledge Center*. Heidelberg: Physica-Verlag, pp. 399–420.

Herden, O., 2001. *Eine Entwurfsmethodik für Data Warehouses*. Dr. Universität Oldenburg.

Holten, R., 1999. *Entwicklung von Führungsinformationssystemen. Ein methodenorientierter Ansatz*. Dr. Universität Münster.

Jarke, M. et al., 1999. Architecture and Quality for Data Warehouses: An Extended Repository Approach. *Information Systems*, 24 (3), pp. 229–253.

Kimball, R. and Ross, M., 2002. *The Data Warehouse Toolkit*. New York: John Wiley.

Larman, C. and Basili, V. R., 2003. Iterative and Incremental Development: A Brief History. *IEEE Computer*, 36 (6), S. 4756.

Lehner, W., 2003. *Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden*. Heidelberg: dpunkt-Verlag.

List, B. et al., 2002. A Comparison of Data Warehouse Development Methodologies Case Study of the Process Warehouse. *Proceedings of the 13th international conference on Database and Expert Systems 2002 in Aix-en-Provence*.

Nordbotten, J. C. and Crosby, M. E., 1999. The effect of graphic style on data model interpretation. *Information Systems Journal*. 9 (2), pp. 139–155.

Ong, H., 1999. *The Evolution Of A Data Warehouse Architecture - One Size Fits All?* [Online]. http://www.aurora-consult.com.au/white.html [Cited 19 January 2006].

Ortner, E., 1995. Elemente einer methodenneutralen Konstruktionssprache für Informationssysteme. *Informatik - Forschung und Entwicklung*, 10 (3), pp. 148–160.

Peralta, V. and Ruggia, R., 2003. Using Design Guidlines to improve Data Warehouse logical Design. [Online]. http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos_act/2003/DAP_SistDW/Material/DesGuidelines-VPRR.pdf [Cited 19 January 2006].

Prakash, N. and Gosain, A., 2003. Requirements Driven Data Warehouse Development. *Short Paper Proceedings of The 15th Conference on Advanced Information Systems Engineering 2003 in Klagenfurt / Velden*.

Quigley, E. J. and Debons, A., 1999. Interrogative Theory of Information and Knowledge. *Proceedings of the ACM SIGCPR conference on Computer personnel research 1999 in San Diego*.

Quix, C. J., 2003. *Metadatenverwaltung zur qualitätsorientierten Informationslogistik in Data-Warehouse-Systemen*. Dr. Technische Hochschule Aachen.

Schelp, J., 2000. *Modellierung multidimensionaler Datenstrukturen analyseorientierter Informationssysteme*. Dr. Universität Bochum.

Simitsis, A. and Vassiliadis, P., 2003. A Methodology for the Conceptual Modeling of ETL Processes. *Decision Systems Engineering Workshop (DSE'03), in conjunction with the 15th Conference on Advanced Information Systems Engineering (CAiSE '03) 2003 in Klagenfurt / Velden*.

Simitsis, A., 2003. Modeling and managing ETL Processes. *Proceedings of the VLDB 2003 PHD Workshop 2003 in Berlin*.

Stahlknecht, P. and Hasenkamp, U., 2005. *Einführung in die Wirtschaftsinformatik*. 11. ed. Berlin et al.: Springer.

Trujillo, J. and Lorán-Mora, S., 2003. A UML Based Approach for Modeling ETL Processes in Data Warehouses. *Proceedings of the ER 2003 in Chicago*.

Valusek, J. R. and Fryback, D. G., 1985. Information requirements determination obstacles within, among and between participants. *Proceedings of the twenty-firstannual conference on computer personnel research 1985 in Minneapolis*.

Vassiliadis, P., Simitsis, A. and Skiadopoulos, S., 2002a. Conceptual Modeling for ETL Processes. *Proceedings of the 5th International Workshop on Data Warehousing and OLAP 2002 in McLean.*

Vassiliadis, P., Simitsis, A. and Skiadopoulos, S., 2002b. On the logical Modeling of ETL Processes. *Proceedings of the 14th conference on advanced information Systems Engineering 2002 in Toronto*.

Vassiliou, Y. et al., 2000. Data Warehouse Research: Issues and Projects. In M Jarke et al., eds. *Fundamentals of Data Warehouses*. Berlin et al.: Springer, pp. 15–21.

Wand, Y. and Weber, R., 2002. Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, 13 (4), pp. 363–376.

Winter, R. and Strauch, B., 2003. A Method for Demand-driven Information Requirements Analysis in Data Warehousing Projects. *Proceedings of the 36th Hawai International Conference on Systems Science 2003 in Hawaii*.