Deriving Initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems

Michael Boehnlein
University of Bamberg, Germany
Feldkirchenstr. 21
96045 Bamberg, Germany
+49-951-863-2514
michael.boehnlein@seda.sowi.uni-bamberg.de

Achim Ulbrich-vom Ende
University of Bamberg, Germany
Feldkirchenstr. 21
96045 Bamberg, Germany
+49-951-863-2514
achim.ulbrich@seda.sowi.uni-bamberg.de

ABSTRACT

In recent years the construction of large scale data schemes for operational systems has been the major problem of conceptual data modeling for business needs. Multidimensional data structures used for decision support applications in data warehouses have rather different requirements to data modeling techniques. In case of operational systems the data models are created from application specific requirements. The data models in data warehouses base on the analytical requirements of the users. Furthermore, the development of data warehouse structures implicates the consideration of user-defined information requirements as well as the underlying operational source systems. In this paper we show that the conceptual data models of the underlying operational information systems can support the construction of multidimensional structures. We would like to point out that the special features of the Structured Entity Relationship Model (SERM) are not only useful for the development of big operational systems but can also help with the derivation of data warehouse structures. The SERM is an extension of the conventional Entity Relationship Model (ERM) and the conceptual basis of the data modeling technique used by the SAP Corporation. To illustrate the usefulness of this approach we explain the derivation of the warehouse structures from the conceptual data model of a flight reservation system.

Keywords

Data Warehouse, Decision Support System, Conceptual Data Model, Entity Relationship Model (ERM), Structured Entity Relationship Model (SERM), Star Scheme, Snowflake Scheme

DOLAP 99 Kansas City Mo USA

1. INTRODUCTION

During the last decade, data warehouse systems have become an essential component of modern decision support systems. According to a survey done by the Meta Group, 4.5 billion US-Dollar were spent on data warehouse projects in 1996. The warehousing industry expects a turnover of 15 billion US-Dollar for the year 2000. Data warehouse systems offer efficient access to integrated and historical data from different, partly heterogeneous and autonomous information sources in order to help managers in planning and decision making [11]. The data within the data warehouse is cleaned, consolidated, aggregated and accumulated in multidimensional data structures to support direct querying and multidimensional analysis.

The development of data warehouse systems is rather different from the development of conventional operational systems. The latter has only to fulfil the application specific business needs. The design of data warehouse systems not only involves the information requirements of the user. Additionally the structure of the underlying source systems has to be considered. The information requirements and the source systems have static as well as dynamic influence, illustrated by possible changes in user requirements and variations in the structure of the underlying sources. This clearly illustrates the need for a comprehensive data warehouse modeling technique. We are pursuing this goal and through this paper we aim to describe the first part of our work: the derivation of initial logical data warehouse structures from the conceptual data models of the underlying operational source systems.

For the reason that most source systems are relational, first we had to find a suitable ERM extension for the identification in the source systems. In our opinion the Structured Entity Relationship Model (SERM) is the best alternative, because of the feature of the SERM to visualize existency dependencies between data object types explicitly. The SERM also forms the basis of the data modeling technique used by the SAP Corporation. For the derivation of data warehouse structures we assume that the conceptual data models of the operational source system is designed on the basis of SERM. This is, however, not very restrictive, as every consistent entity relationship diagram can be transformed into a structured entity relationship diagram (SER-diagram). If no ERM documentation of the source systems is available, the SERM offers a comparatively speaking easier redesign functionality [24].

On the subject of application development most authors distinguish between conceptual, logical and physical development phases (e.g. [25]). Currently, the main emphasis of data warehouse modeling is on the logical and physical phases [15]. Recently some modeling concepts have been extended to the conceptual, business oriented issues ([3],[7]). In the absence of a widely accepted standard, we only aim to describe the relational alternative for multidimensional data modeling - the star schema. Considering that this approach can also be extended to other (e.g. multidimensional) modeling alternatives, this is not a restriction.

This paper continues with the related work in data warehouse modeling in section 2. This section contains an overview of the basic elements of multidimensional data structures and their representation in the logical modeling diagrams of the star scheme. In section 3 we briefly mention the conceptual data modeling technique of the SERM and explain the differences to the ERM. The main focus of this paper is the derivation of initial data warehouse structures as presented in section 4. The derivation process is illustrated through a conceptual data model of a flight reservation system. The result of the derivation is shown by means of the star scheme. The paper closes with a summary and an outlook on future research topics.

2. RELATED WORK

So far, there is no widely accepted standard for the conceptual modeling of data warehouses. Some of the basic methods which have been proposed include: Application Design for Analytical Processing Technologies (ADAPT) [3], "sichtenspezifische Modellierung" [6], Stars [17] and Dimensional Modeling [7]. There is also general consensus that classical ER modeling techniques are not suitable for warehouse design: "Entity relation models cannot be used as the basis for enterprise data warehouses." [13] and "...provide us with no good way of modeling hypercubes – the basic building blocks of OLAP databases." [3]. For this reason we use a logical and not a conceptual modeling technique for data warehouse structures in this paper.

The basic building blocks of multidimensional data structures as a central basis of data warehouses are briefly discussed in this section.

The basic idea of multidimensional data structures is the separation of quantitative and qualitative data [20] (figure 1). Quantitative measurable facts, called measures or measured facts, are analyzed from various viewpoints based on the qualitative

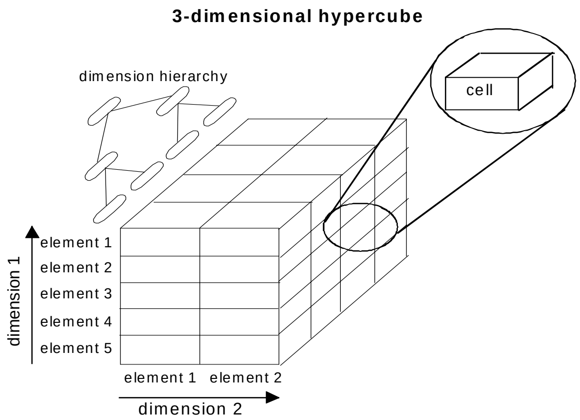

Figure 1: Basic elements of multidimensional data structures

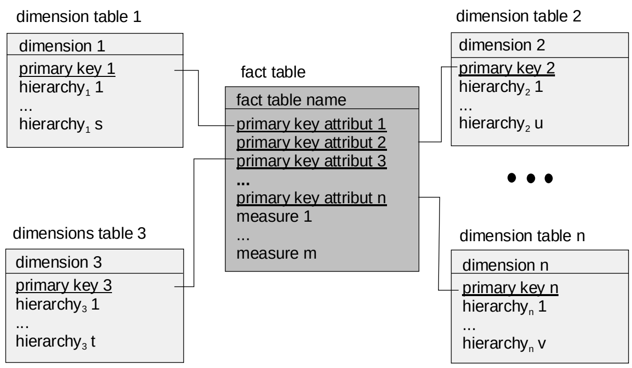

Figure 2: Star scheme

content of the data [12]. For example the turnover of a company could be analyzed by examining the product structure, sales structure and time. The combination of some qualitative aspects to a common consistence is called dimension. This classification leads to a n-dimensional data structure, called hypercube (multidimensional cube). Every dimension has a set of dimension elements, e.g. a product dimension of an automobile company could include different types of cars. The intersection of a dimension element for every dimension in the cube forms a cell with the quantitative measure. Dimension elements can be arranged hierarchically. The measures in the hypercube are summarized along the dimensional hierarchies according to mathematical rules. The most frequently applied consolidation rule is summarization, e.g. the turnover of different products has to be summarized to a single product group. But it is also possible to use more complex rules (e.g. the average). In this paper we do not mention any special case of dimensional modeling, like parallel hierarchies, proportional aggregation or unbalanced trees [10].

The techniques of star and snowflake models and their variants have become a well known representation on the logical layer of multidimensional data structures in relational based systems [9].

In the star scheme there are two different variants of tables: a fact table and at least one dimension table to describe a hypercube ([18], [19]). As the name indicates, the visualization of a star scheme looks like a star (figure 2). The measures are stored in the fact table and the dimension tables include the hierarchical structure of the qualitative attributes. Connecting the dimension tables and the fact table is made possible by storing the primary keys of the dimension tables in the fact table, where they become part of the primary key. The hierarchies of the dimensions are modeled in the corresponding dimension table. However, there is no information about the hierarchy levels stored in the basic structure of the star scheme. Additionally, the representation of a dimension by a single table leads to redundant, denormalised data.

A snowflake scheme is a transformation of the star scheme based on the third normal form. The redundancies are eliminated by the normalization of the dimension tables. For every dimension hierarchy there exists a separate table (dimension table).

The selection of a suitable model is based on the trade-off between storage costs and query performance. Furthermore, there exists a lot of different variants for both models. For example the galaxy scheme is a star scheme with several fact tables [16]. The starflake scheme [1] or degenerated snowflake scheme is a combination of the star and the snowflake scheme in which a part of the dimension tables are denormalised. Within the fact constellation scheme [18] there are additional fact tables for specific dimensions including already summarized data.

In this paper we illustrate the derivation of the logical data warehouse structures from the conceptual data schemes of the operational source systems. A different approach to deriving data warehouse structures from ER-diagrams is discussed in [7]. The major disadvantage of this approach is having to find a first point of reference for the derivation in an ER-diagram. This difficulty is clearly illustrated if the conceptual data models of the underlying operational systems are very complex. SERM allows a better graphical representation of complex schemes and is more suitable for the derivation of complex schemes. The structure of SERM is described in the following section.

3. SERM

The most commonly applied modeling technique for data in operational business systems is the Entity Relationship Model (ERM) developed by Chen in 1976 [4]. The popularity and the acceptance of the ERM can be attributed to the natural expressions that resemble terms of objects in the real world. The entity type (E-type) and the relationship type (R-type) are the basic components of the ERM. In ER-diagrams E- and R-types are connected by edges. To indicate the complexity of the relationship between entities the (1,m,n)-notation is used in the basic design of the ERM. From a graph-theoretical point of view, the ER-diagram describes general bipartite graphs with nodes all with the same ranking.

A comprehensive extension of the ERM is the Structured Entity Relationship Model (SERM) ([21], [23], [24]). It eliminates some of the representational and analytical disadvantages found in the ERM [22]. The major advantages of the SERM [5] are:

- ***Designing extensive data models:*** Due to the fact, that in ER diagrams all nodes have the same ranking, it is very difficult to find a suitable starting point in the analysis of a data model. In SERM the nodes are arranged in such a way as to indicate their interdependencies in a hierarchical way. This leads, to a quasi-hierarchical (acyclic and directed) graph as opposed to the bipartite graph of the ERM.
- ***Visualization of the order of dependencies between data object types:*** In contrast to the ERM, where relations between E-types are modeled, modeling in SERM means constructing the data model based on the principle of dependency.
- ***Avoiding inconsistencies:*** The hierarchical structure of a SER-diagram prevents the modeling of a cycle, a special kind of closed loop. These cycles can be syntactically correct but lead semantically to inconsistent data models.
- ***Avoiding unnecessary relationships:*** In contrast to ERM, the creation of a relational database from a conceptual data model in SERM can be done with very little structural transformations.

Besides the E- and the R-type, the SERM also includes an entity-relationship-type (ER-type). This is a combination of an E-type and a R-type with a (1,1)-relationship. The ER-type has two faces: from the left side it is a R-type symbol and from the right side it is an E-type symbol. Different kinds of edges between the data object types correspond with the specification in (min,max)-notation (figure 3).

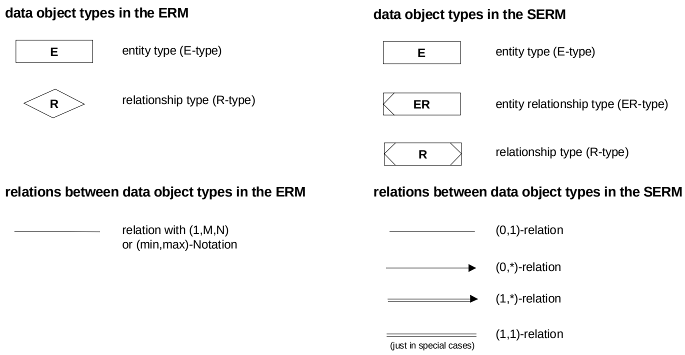

Figure 3: Modeling components in ERM and SERM

A given ERM without inconsistencies can be transformed into a SERM. A correct transformation from ERM to SERM is shown in [8]. A shortcut of the transformation rules is shown in Appendix 8.2.

The feature of the SERM to visualize existency dependencies between data object types is most important for the identification of initial data warehouse structures. The basic data object types (no existency dependencies) are located on the left hand side of the SER-diagram. The dependent data object types can be found to the right of the more independent data object types.

4. THE DERIVATION PROCESS

In this section we introduce a flight reservation system of an airline as an example of an application to explain the derivation of initial data warehouse structures. We assume that the major goal of the airline is to improve its turnover. Further objectives are maximization of profits and service performance. We assume that the turnover of the airline is mainly driven by flight bookings.

With a part of the data model of a central flight reservation system for several airlines we point out specific restrictions and assumptions of the application domain (figure 4)[1].

An airplane is a specific aircraft type and is assigned exactly to one airline and can be used for different flight. However, we assume that a specific flight is flown only by the same aircraft. A flight consists mostly of different flight intervals, which correspond to specific stages between two airports. A stage can occur in different flight intervals and therefore can be served by different aircrafts. Airports are characterized by their geographical location. A specific reservation is associated with a specific booking class. The number of available seats for a booking class depends on the aircraft. In this example we assume that the fares remain constant, related only to the flight interval and the corresponding booking class. Furthermore, each passenger reservation transaction corresponds explicitly to a specific flight interval and a specific booking class. In the illustration, PK and FK indicates the inheritance of primary or foreign keys respectively. The characteristic attributes and an extension of the data object types is shown in the appendix (figure 6).

[1] The corresponding ER-diagram is shown in Appendix 8.3.

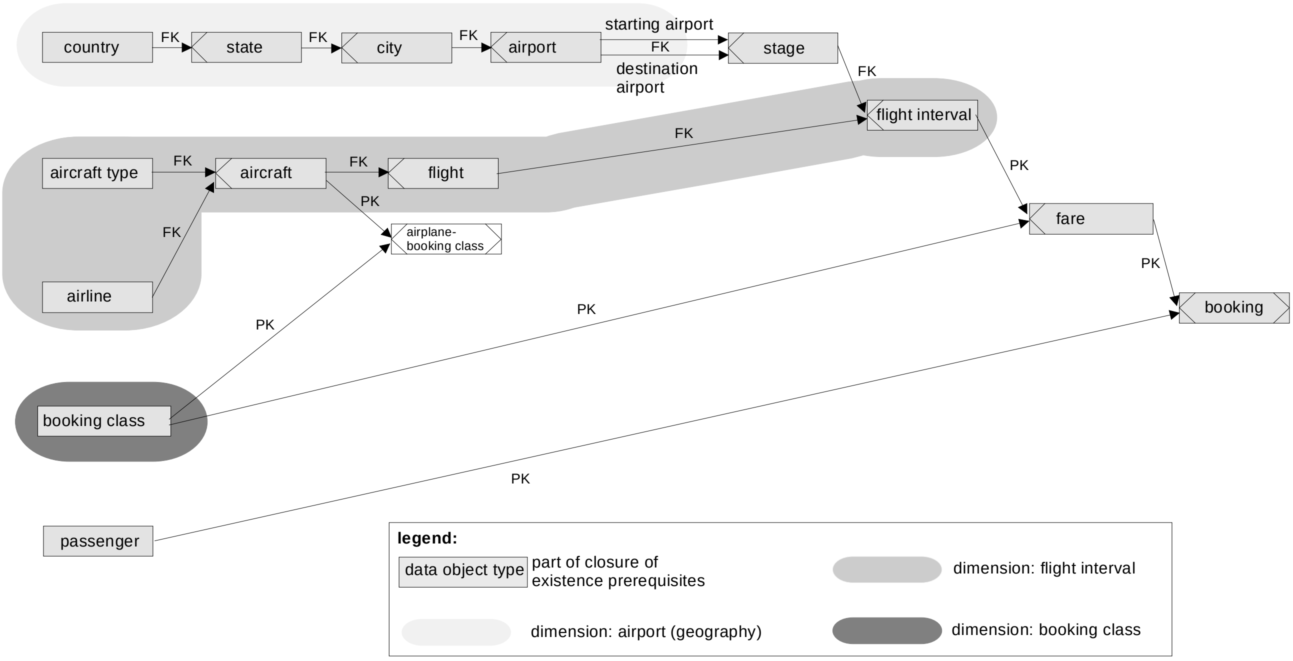

Figure 4: Identification of initial data warehouse structures in the SER-diagram

We now introduce a detailed description of the derivation of initial data warehouse structures from the conceptual scheme. We observe the following three stages:

a) Identification of business measures

b) Identification of dimensions and dimension hierarchies

c) Identification of integrity constraints along the dimension hierarchies

ad a) Identification of business measures

Business measures are determined by the argumentation chain: goals – services – measures.

We differentiate between goals and objectives. We assume that goals describe the nature and purpose of the service production, while objectives comprise of the nature and extent of achieved goals. The airline in question pursues the goal of increasing its turnover. Objectives include the maximization of profits and service performance. Next, we examine the production program of the company. The airline offers a reservation service for transportation. How can the offered service be measured? Service performance and achievement of the set goals should be evaluated through adequate quantitative measures. Business events describe hints for discovering adequate measures. The number of reservations is a meaningful basic value to evaluate the offered transport services. This value corresponds to the business event "reserving flight by a passenger". With additional information about the fare we can determine the derived measure *reservation turnover*.

The identified measures, which form the core of the multidimensional data structure are assigned to one or more data object types of the conceptual model in SERM. The structure of the existency dependencies in the SERM is helpful for further considerations. It offers another dimension to the classical paradigm of dividing object types in structural and transactional data. From the aspect of existency dependencies this paradigm is generalized to a visualization of related object types into pairs [22]. The data object types on the left hand side are "more structural" and the data object types on the right hand side "more transactional". For this reason we have to assign measures to dependent data object types. The information about the measure "number of reservations" is stored in the *booking table* that corresponds to the data object type *booking* in the conceptual SERM scheme. The derived measure "reservation turnover" can be determined from the data object types *booking* and *fare*.

ad b) Identification of dimensions and dimension hierarchies

The clear visualization of existency dependencies in a conceptual SERM scheme is helpful for the identification of potential dimensions and dimension hierarchies. To identify candidates for dimensions and dimension hierarchies we first have to determine the closure of existency prerequisites C_{EX}. The starting point is the data object type assigned to the chosen measure. Then we have to pass through the edges in the opposite direction from right to left and gradually enclose all data object types with existency prerequisites to the data object types of our measures. In the example (figure 4) we determine the closure of existency dependencies for the *number of reservations* and the *reservation turnover* starting from the data object types *booking* and *fare*. The existency prerequisites of *fare* are *flight interval* and *booking class*, for *booking* it is *passenger*. The data object type *passenger* has no further existency prerequisites. Furthermore, *flight interval* depends on *stages* and *flights*. Existency prerequisites for *stages* are *airport*, *city*, *state* and *country*. The other existency prerequisites for *flight* (*aircraft*, *airline* and *aircraft type*) are added to the closure. The data object type *aircraft-booking class* which comprises the number of seats in an airplane related to a specific reservation class is not part of the closure. The resulting closure of existency prerequisites for the data object type *booking* is:

C_{EX} (booking, fare) = {booking, fare, passenger, flight interval, booking class, stage, flight, airport, aircraft, city, aircraft type, airline, state, country}

In figure 4, the closure of existency dependencies is illustrated through gray filling of the data object types. The emphasis is the sub graph that consists of data object types and the related edges from the closure of existency prerequisites.

Discovering dimensions and dimension hierarchies requires creativity and considerable knowledge of the application domain. Our approach supports the development of initial data warehouse structures. Starting with the data object types of the chosen measure we examine object types along the edges of the graph in the direction right to left for potential dimensions and dimension hierarchies. Data object types can form the base for building a dimension. It is decisive to find the data object types with important attributes for analytical purposes. The data object type *booking class* is associated with the information about the booking class of a passenger and the specific airline. This information is important for the evaluation of the number of reservations and the reservation turnover. Therefore, we create the dimension *booking class*. There is no clue for the identification of dimension hierarchies in a single data object type, when the conceptual data model is in third normal form. It is not possible to derive any information about hierarchies in the dimension *booking class* from our conceptual scheme.

We now examine the data object *type flight interval* and its existence prerequisites. It is not necessary that every data object type has information about potential dimensions and dimension hierarchies. (0,*)- and (1,*)-relations between two or more adjacent data object types give a formal hint at hierarchies within a dimension. The hierarchies must build a factual and consistent aggregation path for analytical purposes. The data object types *country*, *state*, *city* and *airport* form a chain of such (0,*)-relations. These data object types describe the geographical location of an airport with a meaningful hierarchy for analytical purposes. This leads to the construction of the dimension *airport (geography)*. The distinction between start and destination airport requires the double use of the dimension *airport (geography)* to build our cube. The data object type *stage* does not have to be considered as the essential information about the start and destination airports is already included.

The data object types *aircraft type*, *aircraft*, *flight*, *flight interval* and *airline*, *aircraft*, *flight*, *flight interval* also form two chains of (0,*)-relations. These data object types fulfil the formal criterion of a dimension hierarchy and build a meaningful aggregation path. *Flight interval* is chosen as the title of the dimension. The intersection of the data object types *aircraft*, *flight* and *flight interval* in both chains indicates a potential parallel hierarchy within the dimension. It is possible to build aggregations with *flight interval*, *flight*, *aircraft*, *airline* and *totals* as well as aggregations with *flight interval*, *flight*, *aircraft*, *aircraft type* and *totals*. In general, several edges leading to the same data object type could give a hint towards the existence of parallel hierarchies within a dimension.

An important indication for appropriate candidates for dimensions from specific data object types of a conceptual model is the recurrent economic standard dimensions with the according consolidation paths [2]. For example, these are company structure (sphere of activity, organisational structure and juristical units), product structure (product families, product groups, products), regional structure (country, state, city) and customer structure (customer groups).

There is an additional dimension in our example which is an essential part of almost every data warehouse structure: the *time dimension*. Due to the fact that operational systems normally do not store any historical sales transaction for longer time periods, the time dimension cannot be found in the operational data model. In some instances time is modeled as an attribute in the relation of the corresponding measure. In the case of the flight reservation system the data object type *booking* includes the attribute *booking date* (figure 6). Normally, the hierarchical structure of the time dimension is given by the analytical actuality requirements. We assume the following hierarchy: *day* - *month* - *quarter* - *financial year*.

Our example is restricted to the following four dimensions: *time* - *airport (start and departure)* - *flight interval* - *booking class*. An additional dimension could be formed by the demographic aspects of the passengers.

ad c) Identification of integrity constraints along the dimension hierarchies

The consolidation of data along the dimension hierarchies depends on specific integrity constraints. The data elements of a lower hierarchical level are summarized to the higher hierarchical level. Other more complex computation rules, e.g. the median, could also be used. These consolidation rules can not be represented in standard star and snowflake schemes. The correctness of the consolidation rules can only be proven by specific analysement tools.

In figure 5 we show the transformation of the identified multidimensional data structures into the star scheme. The fact table of the star scheme contains both selected measures "number of reservations" and "reservation turnover" as attributes. For each of the four dimensions a separate dimension table is build. A hierarchy level of a dimension is realized by a corresponding attribute. The attributes *flight interval*, *flight*, *aircraft*, *aircraft type* and *airline* in the dimension table *flight interval* are the hierarchy levels. The information about the order of hierarchy levels cannot be illustrated. The dimensions *time*, *booking class*, *airport (geography*) are designed analogous. The primary keys of the dimensions are included in the fact table. The primary key of the dimension *airport* is assigned twice to the fact table to distinguish between the start and destination airports. A star scheme is transformed to a snowflake scheme through denormalization of the dimension tables. A dimension hierarchy is

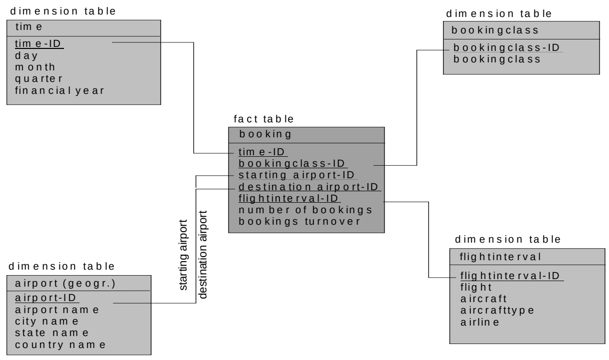

Figure 5: Star scheme of the flight reservation system

build through separate tables. The primary key of the higher hierarchical table is inherited to the hierarchical lower table as a foreign key.

5. SUMMARY AND OUTLOOK

In this paper we have presented a new approach to derive initial data warehouse structures from the conceptual schemes of operational sources. We have shown that the data modeling technique of SERM is highly suitable for this purpose because of the explicit visualization of existency dependencies. The efficiency of our approach was illustrated by an example of a flight reservation system. The automatic creation of multidimensional tables is not practical, because of varying requirements and creative scope. Currently, we are developing a design tool that supports semiautomatic derivation of multidimensional data structures from conceptual source schemes.

We have limited our approach to the data modeling perspective, that has to be realized by a conceptual data model. Our next step is the extension of our approach to business process models [14]. Furthermore, the adequate representation of maintenance and consistency requirements in the data warehouse has to be investigated for a comprehensive modeling technique. This is very important for incorporating changes to the structural data within dimensions and their effect on aggregations.

Further research should involve developing a suitable conceptual modeling technique, which will enable an adequate representation of the multidimensional world. The conceptual data warehouse modeling approach should be independent from the underlying database technology. Relational, multidimensional and hybrid database technologies should be served without changes to the conceptual models. Such a modeling technique forms a foundation for discussions between the business and the information technology staff.

6. ACKNOWLEDGMENTS

The empirical background of our approach is the CEUS-HB project (Computerbasiertes Entscheidungsunterstützungssystem für die Hochschulen in Bayern - Computerbased Decision Support System for the Universities of Bavaria), by which a hierarchical distributed data warehouse system for the Bavarian universities is being developed. The CEUS-HB project is initiated and supported by the Bavarian Ministry of Science, Research and Art. We are members of an interdisciplinary research group of the Bavarian State Institute For Higher Education Research (Prof. Dr. H.-U. Küpper) and the University of Bamberg (Prof. Dr. E. J. Sinz).

7. REFERENCES

[1] Anahory, S.; Murray, D.: Data Warehousing in the Real World, Addison-Wesley, Harlow, 1997.

[2] Behme, W.; Schimmelpfeng, K.: Führungs-informationssysteme: Geschichtliche Entwicklung, Aufgaben und Leistungsmerkmale, in: Behme, W.; Schimmelpfeng, K. (editors): Führungsinformations-systeme - Neue Entwicklungstendenzen im EDV-gestützten Berichtswesen, Gabler, Wiesbaden, 1993, p. 3-16.

[3] Bulos, D.: A New Dimension, in: Database Programming & Design, 6/1996.

[4] Chen, P. P.-S.: The Entity-Relationship Model – Toward a Unified View of Data, in: ACM Transactions on Database Systems, Vol. 1, No. 1, 1976, p. 9-36.

[5] Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik, third edition, Oldenbourg, 1998.

[6] Gabriel, R.; Gluchowski, P.: Semantische Modellierungstechniken für multidimensionale Datenstrukturen, in: HMD - Theorie und Praxis der Wirtschaftsinformatik 195, 1997, p. 18-37.

[7] Golfarelli, M.; Maio, D.; Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schemes, in: Proceedings of the Hawaii International Conference on System Sciences, January 6-9, Kona, Hawai, 1998.

[8] Gross, H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell; Bamberger Beiträge zur Wirtschaftsinformatik, no. 9, march 1992.

[9] Hahne, M.: Logische Datenmodellierung für das Data Warehouse, in: Chamoni, P.; Gluchowski, P. (editors): Analytische Informationssysteme, Springer, Berlin, 1998, p. 104-122.

[10] Holthuis, J.: Der Aufbau von Data Warehouse Systemen - Konzeption - Datenmodellierung - Vorgehen, DUV, Wiesbaden, 1998.

[11] Inmon, W. H.: Building the Data Warehouse, Second Edition, Wiley & Sons, New York, 1996.

[12] Kenan Technologies: An Introduction to Multidimensional Database Technology, White Paper,Kenan Technologies, 1995.

[13] Kimball, R: Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouse, Wiley & Sons, New York, 1996.

[14] Kimball, R.: A Dimensional Modeling Manifesto, in: DBMS online, http://www.dbmsmag.com/9708d15.html.

[15] Labio, W. J.; Quass, D.; Adelberg, B.: Physical Database Design for Data Warehousing, in: Proceedings of the 13th International Conference on Data Engineering, april 7-11, Binghamton, UK, 1997.

[16] McGuff, F.: Hitchhiker's Guide to Decision Support, http://members.aol.com/fmcguff/dwmodel/frtoc.htm.

[17] Peterson, S.: Stars: A Pattern Language for Query Optimized Scheme, http://c2.com/ppr/stars.html.

[18] Raden, N.: Star Scheme 101, http://members.aol.com/nraden.str.htm.

[19] Raden, N.: Modeling the Data Warehouse, http://techweb.cmp.com/iw/564/64oldat.htm.

[20] Shoshani, A.: Statistical Databases: Characteristics, Problems and some Solutions, in: Proceedings of the

8th International Conference on Very Large Data Bases (VLDB'82, Mexico City, Mexico, sept. 8-10), 1982, p. 208-222.

[21] Sinz, E.J.: Datenmodellierung betrieblicher Probleme und ihre Unterstützung durch ein wissensbasiertes Entwicklungssystem, habilitation paper, Regensburg, 1987.

[22] Sinz, E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM), in: Müller-Ettrich, G. (editor): Fachliche Analyse von Informationssystemen, Addison-Wesley, Bonn, 1992.

[23] Sinz, E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM); Bamberger Beiträge zur Wirtschaftsinformatik, Nr. 10, may 1992.

[24] Sinz, E.J.: Das Strukturierte Entity-Relationship Model (SER-Modell), in: Angewandte Informatik, volume 30, issue 5, 1988, p. 191-202.

[25] Vossen, G.: Datenbankmodelle, Datenbanksprachen und Datenbankmanagementsysteme, München, third edition, Oldenbourg, 1999.

8. APPENDIX

8.1 Extension of the flight reservation example

Primary key attributes are underlined. Foreign key attributes are explicitly part of the data object types. For each data object type a possible extension consisting of two data objects is shown.

country

country-ID	countryname
01	USA
02	Germany
...	...

state

state-ID	country-ID	statename
CA	01	California
KS	01	Kansas
...	...	...

city

city-ID	state-ID	cityname
01	CA	Los Angeles
02	CA	San Francisco
...	...	...

airport

airport-ID	city-ID	airportname
01	01	LA Airport
02	02	SF Airport
...	...	...

stage

stage-ID	starting airport-ID	destination airport-ID
1	02	01
2	01	02
...	...	...

aircraft type

aircraft type-ID	aircraft typename
01	Boing 747
02	Airbus
...	...

airline

airline-ID	airlinename
01	Lufthansa
02	USAir
...	...

airplane

airlplane-ID	aircrafttype-ID	airline-ID	airplanename
01	01	02	Hamburg
02	01	02	Dresden
...	...	...	...

flight

flight-ID	airplane-ID	flightname
CO51	02	Frankfurt-LA
LU43	01	Frankfurt-SF
...	...	...

flight interval

flight interval-ID	stage -ID	flight-ID	departure
01	1	CO51	25.03.99 15:30
02	1	CO51	26.04.99 07:00
...	...	...	...

booking class

booking class-ID	booking class
1	First Class
2	Standard Class
...	...

airplane - booking class

booking class-ID	airplane-ID	seats
1	02	30
1	01	20
...	...	...

passenger

passenger-ID	name
2300	Böhnlein
2301	Ulbrich-vom Ende
...	...

fare

flight interval-ID	booking class-ID	fare
01	02	300
02	01	400
...	...	...

booking

flight interval-ID	booking class-ID	passenger-ID	booking date
01	01	2300	20.02.1999
02	01	2301	02.03.1999
...	...	...	...

Figure 6: Attributes and extension of the flight booking example

8.2 Corresponding structures in ERM and SERM

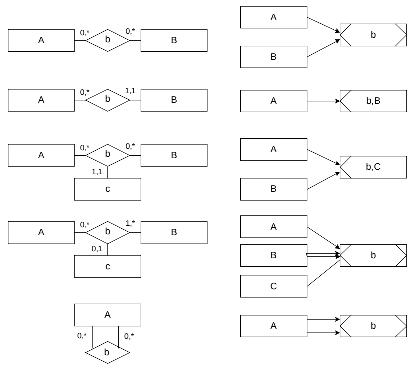

Figure 7: Corresponding structures in ERM and SERM

8.3 ER-diagram of the flight reservation example

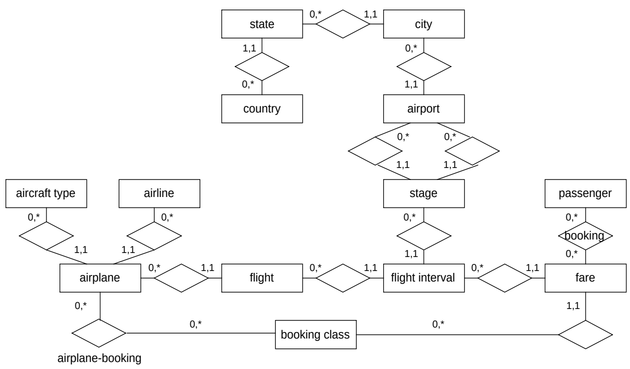

Figure 8: ER-diagram of a flight reservation system