Arbeitsberichte des Instituts für Wirtschaftsinformatik
Herausgeber: Prof. Dr. J. Becker, Prof. Dr. H. L. Grob, Prof. Dr. S. Klein,
Prof. Dr. H. Kuchen, Prof. Dr. U. Müller-Funk, Prof. Dr. G. Vossen

Arbeitsbericht Nr. 87

Specification of Management Views in Information Warehouse Projects

Roland Holten

ISSN 1438-3985

Westfälische Wilhelms-Universität Münster · Institut für Wirtschaftsinformatik
Leonardo-Campus 3, 48149 Münster, Tel. (0251) 83-38100
April 2002

Contents

Figures

Tables

Summary

With respect to the major role information warehouses play for the management an approach for specifying management views within the requirements specification phase is presented. Based on a framework relating development phases and abstraction layers the roles of documents within development processes are organised. The importance of using management views as metadata and parameters in later development phases is elaborated. Formally the transformation of management views into logical data mart schemes and report queries is shown by means of algorithms. Development phases are integrated based on meta level relationships.

1 Introduction

A data warehouse stores materialized views on business processes in support of management's information requirements.[1] It is located on a central layer of an idealized layer oriented architecture connecting online transaction processing (OLTP) systems and components enabling online analytical processing (OLAP).[2] The latter components are intended to support navigations adequate for management users through so called multi dimensional information spaces. OLTP systems directly support the business processes and are the sources of data used by OLAP systems. Typically the integration of OLTP systems and the data warehouse is based on tools performing so called extraction, transformation and loading tasks (ETL).[3]

The spectrum of contributions on this general topic indicates that theory is far away from a clear understanding of all aspects relevant today.[4] Contributions[5] are reaching from technical discussions of data bases and algorithms enabling OLAP functionality[6] information search behaviour of managers[7] and papers concentrating on methodologies of information systems development[8]. Research on development methodologies itself is a complex field since investigations refer to logically separate levels of abstraction. On the one hand there are method development processes.[9] On the other hand processes of information systems development are the matter of interest.[10] The latter have to show how concepts of the former can be put in concrete terms. With respect to VASSILIADIS[11] the lack of an accepted methodology for data warehouse development is a central factor affecting failure of data warehouse projects.

This paper deals with the development and usage of management views as part of the requirements specification phase of information warehouse projects. The main problems addressed are the specification of management views and the integration of the requirements specification phase with later development phases.

1 cf. Inmon (1996); Inmon, Hackathorn (1994); Inmon et al. (1997)
2 cf. Becker, Holten (1998); Chaudhuri, Dayal (1997)
3 cf. Inmon (1996); Widom (1995)
4 cf. Vassiliadis (2000)
5 See Vassiliadis (2000) for an overview of research topics.
6 cf. Agarwal et al. (1996); Agrawal et al. (1997); Bauer et al. (2000); Codd et al. (1993) ; Colliat (1996) ; Gyssens, Lakshmanan (1997); OLAP Council (1997); Vassiliadis, Sellis (1999); Wedekind et al. (1999); Cabibbo, Torlone (2001)
7 cf. Borgman (1998)
8 cf. Golfarelli et al. (1998B); Jarke et al. (1999); Jarke et al. (2000)
9 cf. Nissen et al. (1996); Pohl (1996); Wedekind (1981); Wedekind (1992)
10 cf. Boehm (1981); Davis (1990); Davis et al. (1988); Weske et al. (1999)
11 cf. Vassiliadis (2000)

1. An approach for the specification of management's information requirements based on management theory, using concepts and terms managers are familiar with, is presented. The specification of management views is based on libraries of semantic concepts to solve semantic conflicts and is independent of technological constraints.

2. The integration of the requirements specification phase and later development phases is addressed by a development framework. Development decisions are organised in two dimensions separating three development phases and three levels of abstraction. Information warehouse development phases are integrated by meta level relations. Management views are used as metadata and parameters to generate documents in later development phases. Respective algorithms generating logical data mart schemes and data mart report queries based on management views are presented.

The paper is organised as follows. Section 2 discusses related work. The development framework organising development decisions is presented in section 3. Using an elaborate example an approach for specifying master data of management views in libraries (section 4.1) and defining management views (section 4.2) is presented. Section 5 shows by means of algorithms how master data of management views can be formally transformed into logical data mart schemes (section 5.1) and how management views themselves are transformed into report queries (section 5.2). Discussions and an outlook are given in section 6.

2 Related Work

In recent methodologically oriented contributions JARKE ET AL.[12] propose a quality oriented framework for data warehouse development. It is demanded that all views relevant to data warehouse development are understood as views on a central enterprise model. Even information sources schemes (OLTP systems schemes) are interpreted this way. These views are arranged into so called perspectives by the DWQ Metadata Framework.[13] The conceptual perspective is relevant for business analysts and business departments. It enables models independent from the physical organization of data and comprises views on the central enterprise model. The enterprise model itself gives an integrated overview of the conceptual objects of an enterprise. The logical perspective conceives a data warehouse from actual data models given by the corresponding physical components used to implement the logical scheme, e.g. relational database systems. The physical perspective is related to the physical components used to implement the data warehouse, e.g. commercial tools available on the market. The enterprise model thus not just plays a minor but the central role in the process of data warehouse development.

Another well known approach structuring the development of information systems stressing the fundamental role of enterprise models is the Architecture of Integrated Information Systems (ARIS) presented by SCHEER[14]. The ARIS framework is characterized by different views on the development of information systems. The development process itself is structured by three so called levels reflecting their proximity to information technology. These levels are directly related to well known development phases of information systems. They are called Requirements definition, Design specification and Implementation description. ARIS is not applied to the domain of data warehouse development so far.

An approach to conceptually model the data warehouse is presented by GOLFARELLI ET AL.[15]. A representation formalism for data warehouses called dimensional fact model is formalized in this approach. Additionally it is shown how the fact model can be developed based on given data base schemes of OLTP systems and required algorithms are presented. The focus is on the formal descriptions of the conceptual model and the integration with the OLTP systems. The approach presented by GOLFARELLI ET AL. conceptually describes the structures required to design the data warehouse. Another approach

12 cf. Jarke et al. (1999); Jarke et al. (2000)

13 cf. Jarke et al. (1999); Jarke et al. (2000)

14 cf. Scheer (1999); Scheer (2000); Scheer (1998)

15 cf. Golfarelli, Rizzi (1999); Golfarelli, Rizzi (1998); Golfarelli et al. (1998A); Golfarelli et al. (1998B)

focussing the transformation of conceptual data warehouse schemes into logical schemes is presented by LECHTENBÖRGER[16]. Formalisms for the conceptual specification of data warehouses are developed. Based on a conceptual data model the so called fact schema is introduced. It is shown how data warehouse requirements have to be integrated with the analysis of operational database schemes. Transformation formalisms from the conceptual fact schema into a logical (relational) database scheme are presented.

Other approaches focusing the analysis of OLTP systems schemes to generate or suggest data warehouse or data mart schemes are presented by HÜSEMANN ET AL.[17] and MOODY/KORTINK[18]. The approach presented by HÜSEMANN ET AL.[19] starts by identifying reasonable measures in the OLTP systems schemes whereas MOODY/KORTINK[20] start by classifying the OLTP systems entities. Both approaches concentrate on development decisions in the design phase. A novel approach integrating relational OLAP schemes and multidimensional approaches is proposed by CABIBBO/TORLONE[21]. The authors introduce a multi dimensional data model leading to a new layer in the data warehouse architecture. Another work focusing the integration of OLTP and data warehouse schemes is presented by CALVANESE ET AL.[22]. Views on OLTP and data warehouse schemes are integrated to solve conflicts between these schemes. A metadata based approach to generic graphical model design is presented by SAPIA ET AL.[23]. Another ER-oriented approach for the conceptual specification of data warehouses is presented by TRYFONA ET AL.[24]. The so called starER is intended to combine the star structure discussed in data warehousing environments and constructs of the ER model. This combination intends to better support user requirements and technical constraints as well. The work of BÖHNLEIN[25] concentrates on the specification of data warehouse requirements from the management's perspective. The so called semantic data warehouse model is one result of the requirements specification phase. A meta model is presented and it is shown how the approach can be integrated with the analysis of business process models.

---

16 cf. Lechtenbörger (2001)
17 cf. Hüsemann et al. (2000)
18 cf. Moody, Kortink (2000)
19 cf. Hüsemann et al. (2000)
20 cf. Moody, Kortink (2000)
21 cf. Cabibbo, Torlone (2001)
22 cf. Calvanese et al. (2001)
23 cf. Sapia et al. (2000)
24 cf. Tryfona et al. (1999)
25 cf. Böhnlein (2001)

3 Information Warehouse Development Framework

The framework presented in this section is intended to organize information warehouse development tasks and method development tasks as well. It comprises two dimensions. The first dimension of the framework deals with different levels of abstraction required to understand the modelling process. The second dimension is characterized by the distinction of development phases arranged in a logical order.

Different levels of abstraction are introduced as means to structure in a logical way the relationships between

1. the parts of the business processes which are the instances represented in models defining management relevant views on these business processes,

2. the languages used to create models defining the views on the business processes and

3. the representations of these languages themselves as models.

Different levels of abstraction can be derived from science theory[26] and are well established in software engineering. E.g. the IRDS framework[27] is characterized by four levels of abstraction. The lower three levels organize instance, type and meta information. A fourth level in the IRDS framework provides concepts relevant to develop methods on a meta meta level. Compared to the IRDS framework only the lower three levels forming two overlapping level pairs are of interest here. The interlocking level pairs can be understood as “business process” and “business modelling”[28]. The former describes the business processes on the type level while the process instances themselves are performed on the instance level. The latter explains the process of business development on the meta level while business development processes are performed on the type level. The levels relevant for this paper directly correspond to the abstraction levels discussed within the framework of the Architecture of Integrated Information Systems (ARIS)[29].

The second dimension of the framework concerns information systems development processes. For the purpose of this work development processes are structured in three phases: *Requirements specification, design* and *implementation*. It is widely accepted that any system development process has to start with the *requirements specification*

---

26 see Holten (1999) for a discussion
27 cf. ISO (1990); Pohl (1996); Jacobs, Holten (1995)
28 cf. Jacobs, Holten (1995)
29 cf. Scheer (1999)

defining “what” the system under consideration should do.[30] The focus is on the domain the system has to work in. Thus, it has to specify clearly the domain specific requirements in a language providing domain specific concepts. The *design* or programming in the large phase is generally accepted as the second development phase.[31] The design has to specify the system’s components and the resulting system’s architecture. Typically decisions concerning the logical database model[32] and the user interface[33] are made in this phase. The architecture of the system describes every component, the functions it provides and its relationships to other components. The definition of component interfaces and the separation of a component’s definition and its realization are core principles of this phase.

The third phase – the so called *implementation* phase – deals with the realization of the previously defined components. Tasks in the implementation phase comprise coding, development of algorithms and data structures. Of course these phases are integrated in an evolutionary development process with jumps back to (logically) previous phases if necessary.[34] Maintenance can be seen as another loop of the development process.[35] It is further denoted that the database design process generally is separated in three phases called conceptual, logical and physical design.[36] The intention of these phases corresponds to the phases separated in this paper. Nevertheless requirements engineering is sometimes seen as a sub phase of the conceptual design or as separated initial design phase.[37]

The relationship of the two dimensions (abstraction levels and development phases) is as follows. Development decisions on the type level are made in every of the three development phases. Every type level decision produces certain documents and the document types are specified as models on the meta level. Since every development phase is required on every abstraction level a combination of the two dimensions is meaningful. The resulting *information warehouse development framework* is a relation between the two dimensions comprising nine elements (Figure 1).

---

30 cf. Pohl (1996); McMenamin, Palmer (1984); Davis (1990); IEEE (1984)
31 cf. Boehm (1981); Davis et al. (1988)
32 cf. Codd (1990); Date (1990); Embley (1998); Gupta, Horowitz (1991)
33 cf. Balzert (1996)
34 cf. Boehm (1981)
35 cf. Nagl (1990)
36 cf. Batini et al. (1992); Elmasri, Navathe (2000)
37 cf. Elmasri, Navathe (2000); Vossen (2000)

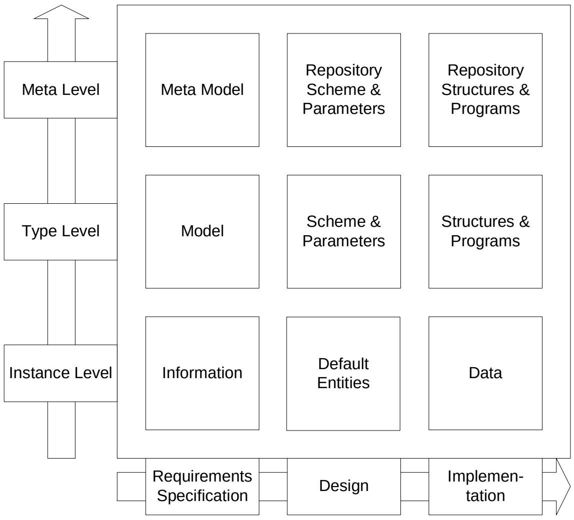


Figure 1: Information Warehouse Development Framework

Every box of the development framework is characterized by a unique set of development tasks and documents (Table 1).

Box / Level	**Decisions concerning**	**Documents e.g.**
Instance Level	*Business Process*	
Information	Business processes and business objects (like markets, products, people)	Contracts leading to sales amounting to 2 million € realised in 3-digit merchandise category "369 Ski & Walking Boots" in Quarter "2001-1" in all towns of region "Europe North"
Default Entities	Allocated entities of a certain type or class filled by default values	Template to replace the object's default values; entity tuple (xxx, yyy, zzz, $$$)
Data	Data tuple of a given type	Insert statement concerning tuple (369 Ski & Walking Boots, 2001-1, Europe North, 2.000.000)
Type Level	*Development Process*	
Model	Specification of management's information requirements	Semantic model specifying libraries of components for management views and information objects relevant for a management user (section 4)
Scheme & Parameters	Specification of information systems design, data base schemes, metadata and parameters of system's components	Data dictionary and repository data, logical data mart scheme (Table 17, Table 18 in the Appendix and remarks in section 5.1)
Structures & Programs	Algorithms and data structures in formal or programming languages defining information systems in support of the business processes	Algorithms realising insert statements and report queries in OLAP and data mart environments (section 5.2, Table 13)
Meta Level	*Method Engineering*	
Meta Model	Specification and modelling of concepts required on the type level for purposes of modelling management views and business processes	Meta model of modelling approach discussed in section 4 (Figure 16 and Table 16 in the Appendix)

Box / Level	Decisions concerning	Documents e.g.
Repository Scheme & Parameters	Specification of logical repository and data dictionary schemes and design of development tools	Repository and data dictionary scheme (section 5.1, Table 4)
Repository Structures & Programs	Algorithms and data structures in formal (or programming) languages in support of information systems development processes	Algorithms in pseudo code and relational expressions to generate logical data mart schemes (section 5.1, Table 5, Table 6, Table 7, Table 8) and report queries (section 5.2, Table 9, Table 10, Table 11, Table 12)

Table 1: Development Framework Boxes – Explanation and Examples

4 Conceptual Specification of Management Views

This section presents concepts and representation formalisms for the specification of management views on business processes. For this purpose the MetaMIS Approach is used.[38] The first subsection introduces master data relevant for the specification of task specific management views. This master data is stored in centralized libraries to solve semantic conflicts within the project. Every component used to specify management views has a unequivocal meaning defined and stored in the libraries. From a technical point of view these libraries are transformed into a repository relevant for development decisions in later project phases (section 5). The second subsection deals with the specification of task specific management views on the business. It is noted that further problems of semantic conflicts, especially concerning views on OLTP, are not the matter here.[39]

The modeling approach is introduced based on the following running example: The CEO of the retailing company EXCOM requires information for short term management tasks. Aspects of relevance are the time structured in months, the articles and the stores of the company. Additionally there exist a lot of ratios used in EXCOM's management, a subset of which is relevant for the CEO. With respect to the development framework introduced in section 3 the box "Model" (requirements specification phase on the type level, Figure 1 and Table 1) is in the focus of this section. The language concepts of the modeling approach are defined additionally in Table 16 and are shown as integrated meta model in Figure 16 (both in the Appendix). This meta model and the respective definitions belong to the box "Meta Model" (requirements specification phase on the meta level) of the framework.

4.1 Libraries of management view master data

The first concept of the modeling approach is *Dimension*. It is used to create and organize the space the management's views are composed of. Dimensions are orthogonal from the management's point of view. There are compulsory dimensions because every management view must have a relation to e.g. time and (optimistic or pessimistic) planning scenarios concerning the business. Concerning the orthogonality of dimensions it is required that dimensions are explicitly compatible when used to define a management view. E.g. there could be a dimension characterising the set of clients with respect to the branch they are working in (dimension "client branch") and a second dimension classi-

38 Holten (2001).
39 These problems are discussed in e.g. Jarke et al. (2000), Calvanese et al. (2001)

fying clients with respect to the gender (dimension “client gender”). From the management’s point of view both classification are useful but can not be combined in a meaningful way. Therefore the first dimension should comprise a sub class “no branch” consisting of all end consumers without a meaningful branch classification and the second dimension should comprise a sub class “no gender” consisting of all business clients (companies) without a meaningful gender classification. Other incompatibilities exist concerning the classification of date of time with respect to weeks versus months. Since weeks do not correctly overlap with months or even years these dimensions are incompatible from the management’s point of view.

In the literature incompatibilities of dimensions often lead to parallel branches within dimensions.[40] It is denoted here that orthogonality and compatibility are two different aspects relevant to the modelling of management views. If aspects concerning business objects are independent (and thus can be combined in principle) from the management’s point of view, this leads to different dimensions with respect to the orthogonality of dimensions. Nevertheless, if the combination of these dimensions is not meaningful from the management’s point of view, this leads to prohibited combinations with respect to the compatibility of dimensions. This means that parallel branches in dimensions are not possible. If they occur the aspects of orthogonality and compatibility are not taken account of.

Dimensions are defined by means of dimension objects. Based on the enterprise theory of RIEBEL dimension objects can be understood as entities which are objects to arrangements or examinations of the management.[41] The enterprise theory provided by RIEBEL is centred around the decision as the fundamental element.[42] Any activity in an enterprise is produced and maintained by certain decisions which therefore are the real sources of cost, outcome and liquidity. Based on RIEBEL’s findings the language concept *Dimension Object* is introduced. Dimension objects are organized in hierarchies (concept *DO-Hierarchy*) and are part of a dimensions’ definition. The concept of DO-Hierarchy allows the construction of e.g. product hierarchies or hierarchies of regions. Every dimension object is associated to an unequivocal hierarchy level (concept *Hierarchy Level*). Hierarchies defining dimensions are always balanced. That is, the number of hierarchy levels in every branch of the hierarchy is the same within one dimension. This is a consequence of the aspect of orthogonality of dimensions. If an aspect is unequivocally defined, the hierarchy must be balanced. If a dimension’s hierarchy is not

40 cf. Bulos (1996); Golfarelli, Rizzi (1999); Golfarelli, Rizzi (1998); Golfarelli et al. (A) (1998); Golfarelli et al. (B) (1998); Sapia et al. (1998); Blaschka et al. (1998); Böhnlein (2001); Lechtenbörger (2001)

41 cf. Riebel (1979)

42 cf. Riebel (1992); Holten (1999)

balanced the aspect of orthogonality, the dimension's definition should be based on, is not taken account of. Contrary to these findings, some authors argue that unbalanced hierarchies are possible.[43] In the approach presented here unbalanced hierarchies will lead to additional dimensions with respect to the orthogonality of dimensions. Dimension objects on the lowest hierarchical level are called *Leaves*, all other dimension objects are called *Non Leaves.* The concepts Dimension and DO-Hierarchy are intended to hierarchically organize certain attributes of business objects which are matter of management decisions. Additionally these attributes are orthogonal from the management's point of view since they can (in principle) be combined with each other to demarcate the objects management decisions are dealing with.

Dimensions are represented by means of (red) rectangles. The distinction between compulsory and non compulsory dimensions and the compatibility of dimensions are not represented here since this is a matter of tools supporting the approach. Dimension objects and the respective hierarchies are represented by means of hierarchical structures. Squares represent hierarchical levels of non leaves. Dimension objects on the lowest level have no square as prefix. Every dimension object is associated with an identifier. Hierarchy levels identifiers are associated to the respective levels which is visualized by means of a doted line and an indentation. Lower level objects are placed rightwards of higher level objects. The squares with the "+" sign indicate that there are more subordinate dimension objects which are not shown to enhance clearness. For squares with the "-" sign all dimension objects of the succeeding hierarchy level are visible.

The models from Figure 2 to Figure 5 show (excerpts of) the four dimensions relevant for the management of articles. For every dimension the hierarchy levels are shown. The leaves of the hierarchies are identical for all four dimensions (see also explication of dimension groupings below). The first dimension "Store Assortment CCG" (Figure 2) is relevant for benchmarks since the CCG structure is an accepted standard in the retailing sector in Europe.[44] E.g. the article "36904711 Powder Power -S11- R" is member of the 4-digit merchandise category "3690 Men Ski Boots Alpine". The second dimension "Quality / Price Level" (Figure 3) is relevant for the segmentation of the assortment according to quality levels. The same article "36904711 Powder Power -S11- R" is associated to the quality level "high". The third dimension "Category Management" (Figure 4) is relevant to manage categories according to customer needs. E.g. the article "36904711 Powder Power -S11- R" belongs to category department "Seasonal Sports". Categories are required to structure layouts. Finally there is a dimension relevant for fashion managers called "Colour" (Figure 5). The article "36904711 Powder

[43] cf. e.g. Böhnlein (2001)
[44] cf. CCG (1997)

Power -S11- R" belongs to colour group "Red". There exists a blue version too ("36904711 Powder Power -S11- B").

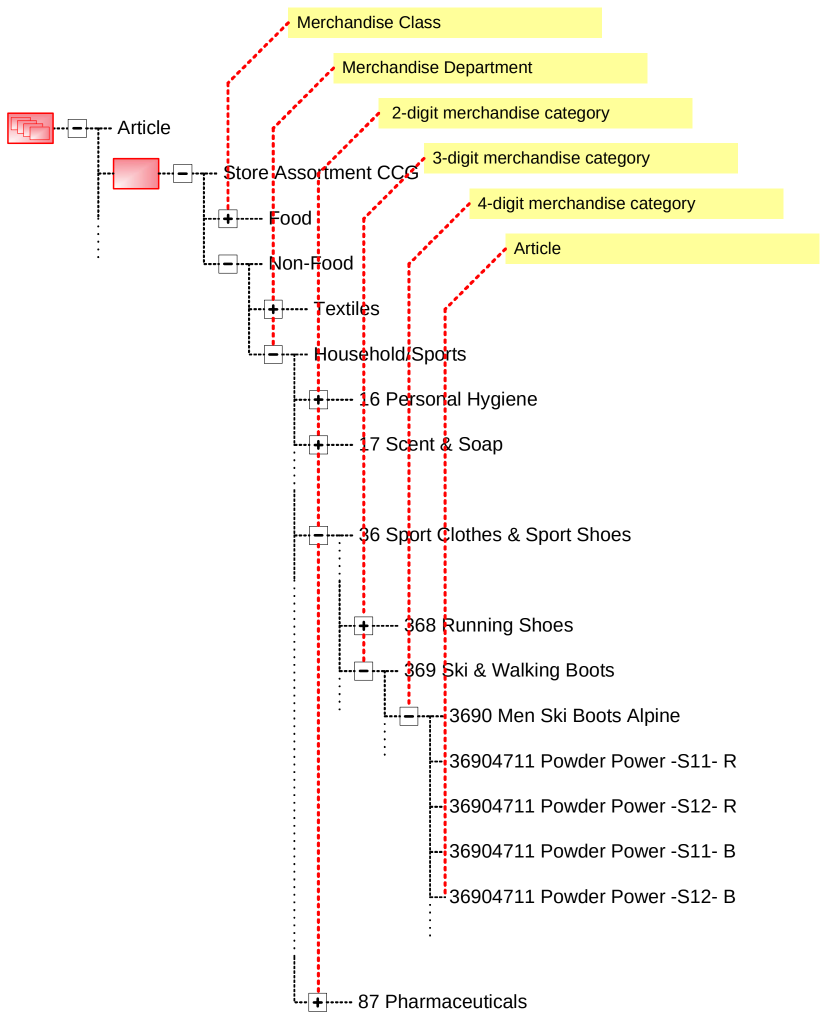


Figure 2: Dimension Grouping Article-Part 1

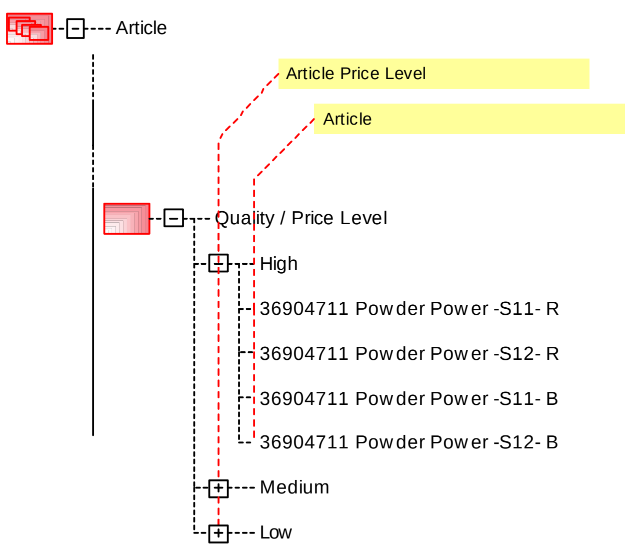


Figure 3: Dimension Grouping Article-Part 2

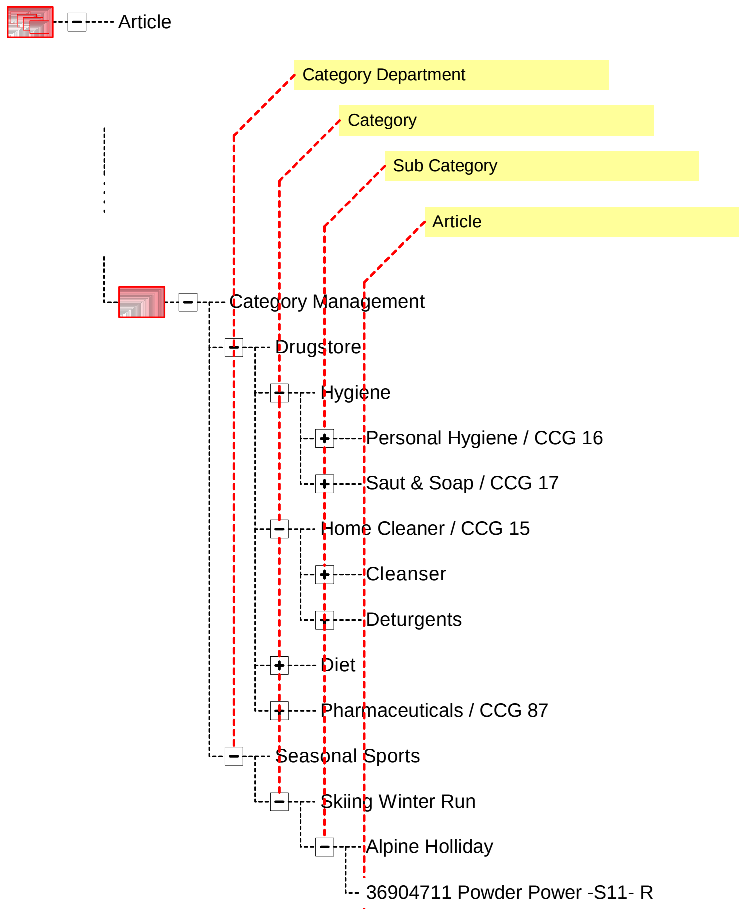


Figure 4: Dimension Grouping Article-Part 3

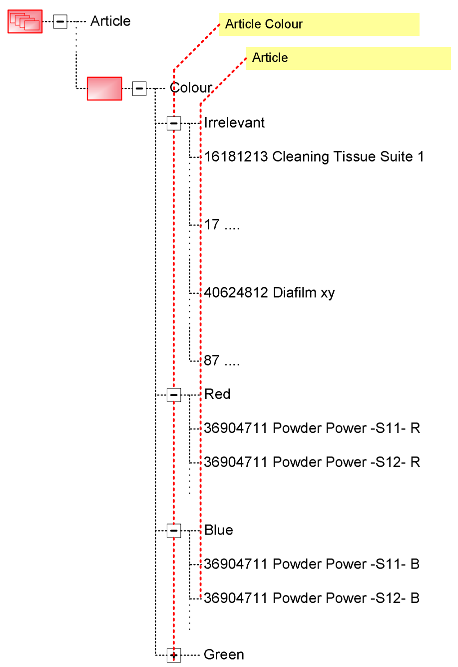


Figure 5: Dimension Grouping Article-Part 4

The leaves of the DO-Hierarchies introduced above are in fact the main objects management decisions are dealing with. This implies two important things:

1. Leave elements can appear in many dimensions (as they do in the examples above). These appearances are based on the existence of different views on identical objects. Since every dimension organizes certain attributes of these objects in a hierarchical manner the objects on the bottom level of the respective hierarchies are identical. This is expressed by means of the concept *Leave*.

2. To consistently integrate all the hierarchical views on identical objects another concept is required. For this purpose the concept *Dimension Grouping* is introduced.

Dimensions which define views on identical objects are subsumed in one dimension grouping (see dimension grouping “Article” in the example given above). All dimensions belonging to one dimension grouping have the *same set of leaves* in their hierarchies. E.g. in retailing companies it is necessary to look at different aspects of stores as fundamental objects (see Figure 7 and Figure 8 below). It is e.g. of interest which competitors have stores in the same area in order to classify the own stores according to this situation. Other important aspects are concerned with the sites of the stores (e.g. down town, outskirts or village) or the age and degree of modernization. All these attributes of the stores are relevant for the management and relate to the same identical set of stores as objects. Additionally it makes sense to combine any of these aspects or classifications with each other to create complex management views. It follows that these aspects are orthogonal from the management’s point of view and thus lead to different dimensions. Nevertheless a grouping of all these dimensions makes sense since they all relate to one unequivocal set of objects, e.g. the set of stores belonging to the retailing company under consideration.

Dimension groupings are represented by means of (red) rectangles containing a set of smaller rectangles. Dimensions belonging to a dimension grouping are hierarchically subordinated to this dimension grouping. The association of identifiers and dimension groupings corresponds to the one of dimensions.

Concerning the dimension grouping Time-Calendar (Figure 6) there exist two dimensions. The first one structures the time according to months, quarters and years, the second one according to weekdays. Every date is associated to every dimension. The first dimension is the compulsory time dimension required for the specification of management views. In the example there is no scenario (e.g. as is values versus plan values) for simplicity reasons.

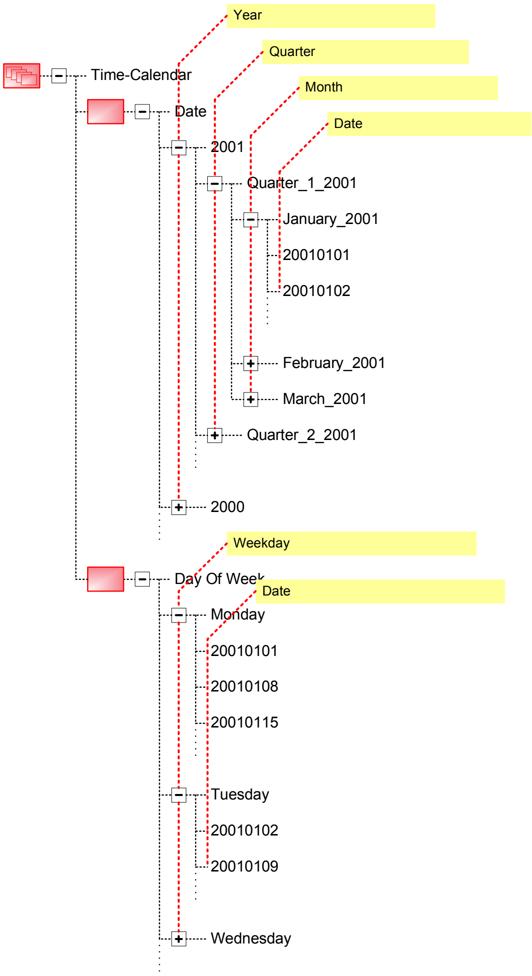


Figure 6: Dimension Grouping Time-Calendar

The dimension grouping "Store" comprises four dimensions (Figure 7 and Figure 8). The dimension "Region" shows the regional structure of EXCOM's business. The dimension "Competition" classifies the stores according to the degree of competition the management senses suitable. "Area / Location" classifies the stores according to the spending power of the respective areas. Finally "Modernization" is a classification with respect to the appearance of a store.

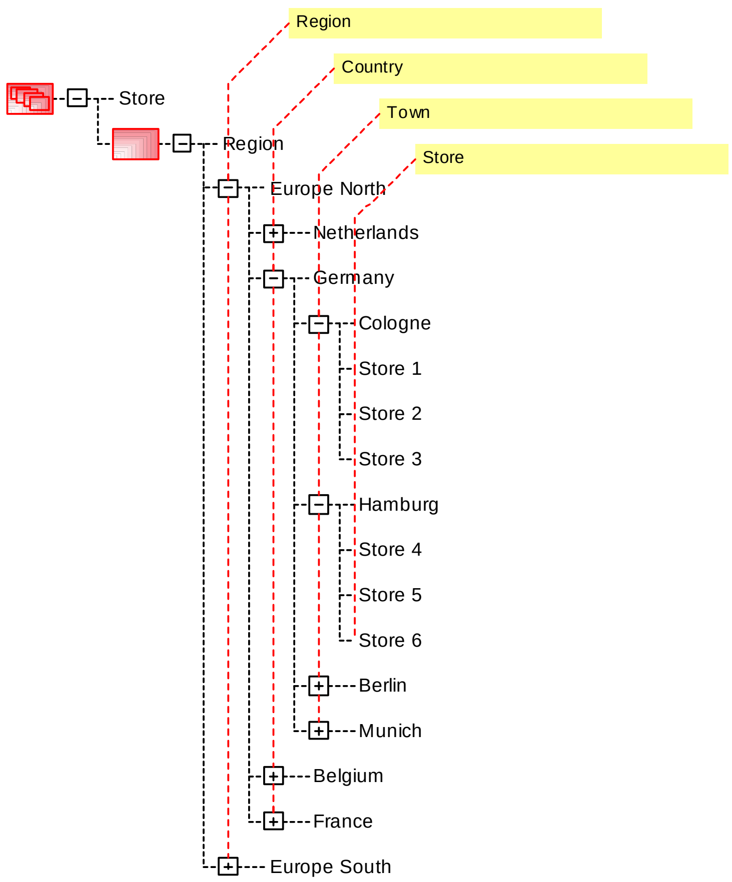


Figure 7: Dimension Grouping Store Part 1

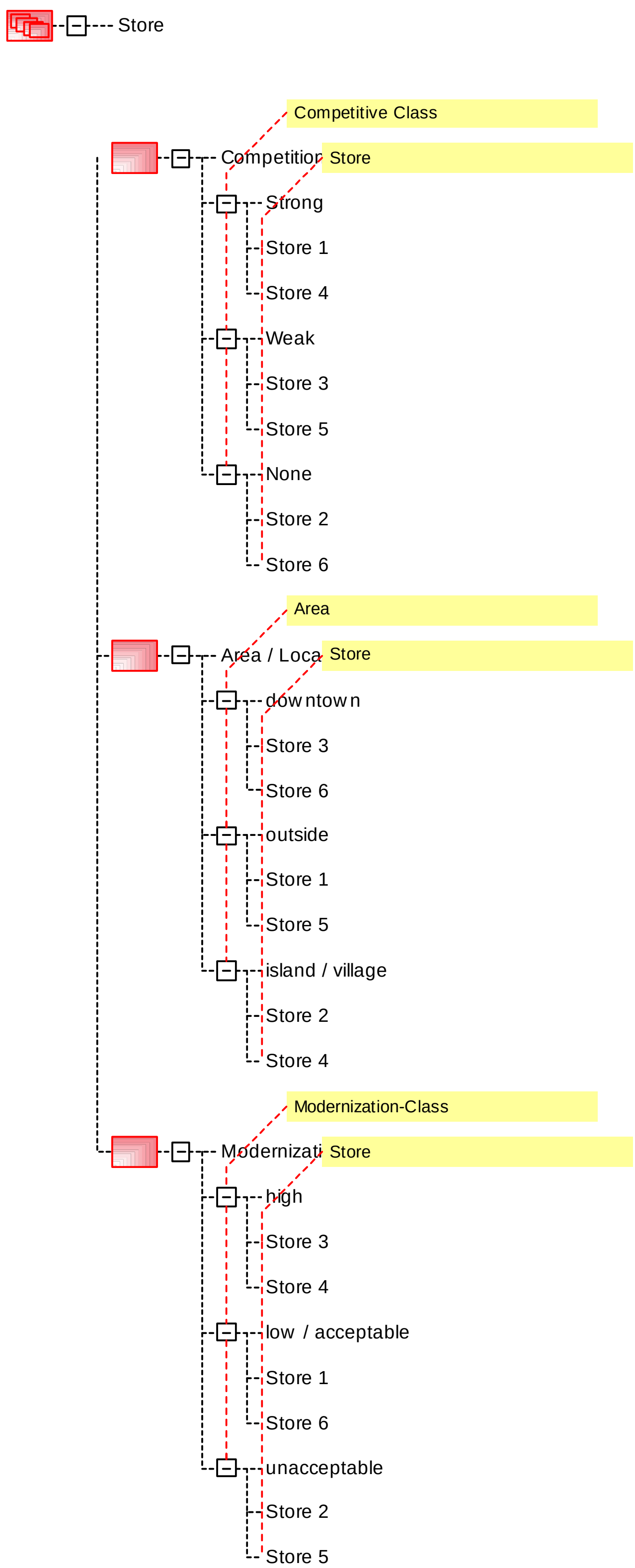


Figure 8: Dimension Grouping Store Part 2

The next concept required is *Ratio* which is of fundamental importance for specifying information in management processes. Ratios are core instruments to measure the value of companies[45], the performance of the business[46] and to analyse the financial situation of an enterprise[47]. Synonyms found in the management accounting literature are operating ratio, operating figure or measure of performance. Ratios like e.g. "gross margin" define important aspects of business objects. Their economic meaning is clearly specified and their calculation is defined by means of algebraic expressions (e.g. "profit = contribution margin – fixed costs").

Ratios are represented by means of rows in tables. The association of identifiers and ratios leads to a respective table entry (Table 2 and Table 3). Basis ratios are defined from the management's point of view by means of statements (linguistic actions) (Table 2). Additionally there are synonyms listed. Calculated ratios require in addition calculation expressions for their definition (Table 3). Every definition of a calculated ratio requires that ratios used in the calculation expression are defined beforehand. The tables serve as libraries for ratios to prevent semantic conflicts.

Ratio	**Description / Unit**	**Synonym**
average annual sales	average sales per year in the local currency valued in planned sales standard prices	*annual sales*
average daily sales	average sales per day in the local currency valued in planned sales standard prices	
average annual inventory	average value of goods in stock per year in average purchase standard price	*average annual inventory level*
average inventory	average value of goods in stock in average purchase standard price	*average inventory level*
contact distance	in m^2 of shelf space	
current inventory	value of goods in stock in average purchase standard price	*stock*
inventory adjustment	adjustment of current inventory to physical inventory valued in average purchase standard prices	*(physical) inventory difference*
net purchase price	purchase price after discount, rebate or other reductions	*net purchasing price*
net sales price	planned sales standard price after discount, rebate or other reductions and without sales tax	
net sales value	value of sales in planned net sales prices	*net sales*
number of employees	number of employees	
order volume	value of orders in the local currency valued in purchase standard prices	*order value*
presentation area	in m^2 of floor space	

45 cf. Copeland et al. (1990)

46 cf. Johnson, Kaplan (1987); Eccles (1991); Lapsley, Mitchel (1996); Kaplan, Norton (1997); Kaplan, Norton (1996); Kaplan, Norton (1992)

47 cf. Brealey, Myers (1996)

Ratio	Description / Unit	Synonym
promotion sales price	sales price referring to a promotion	*promotion price*
promotion purchase price	purchase price referring to a promotion	
promotion purchase value	value in the local currency valued in purchase prices referring to a promotion	*promotion purchase*
promotion sales value	sales value in the local currency valued in planned promotion sales prices	*promotion sales*
purchase price	purchase standard price	
purchase value	purchase value in the local currency valued in standard purchase prices	*purchase, goods usage*
returns	returned goods valued in planned sales prices	
sales price	sales standard price	
sales quantity	sales quantity in quantity units	*asset sale*
sales value	sales value in the local currency valued in planned sales standard prices	*sales, turnover*
…	…	…

Table 2: Table of basis ratios

Ratio	Calculation Expression	Description / Unit	Synonym
area intensity	= average inventory level / presentation area	measure for usage of presentation area as stock in stock value / m^2	
area productivity	= sales value / presentation area	productivity of floor space in use measured in sales value / m^2	
employee productivity	= sales value / number of employees	productivity of staff measured in sales value / person	
gross yield	= net sales – goods usage	margin of goods sold in absolute value based on planned sales prices	*profit margin I*
discount gross margin	= gross yield * 100 / sales	margin in percent based on sales values (in %)	*sales margin*
markup gross margin	= gross yield * 100 / purchase	margin in percent based on net purchase prices (in %)	*goods receipt margin*
inventory turnover	= sales value / average inventory level	productivity measure for the goods usage and level of inventory	
annual inventory turnover	= annual sales / average annual inventory	productivity measure for the goods usage and level of inventory annual periods	
order gross yield	= net sales – order volume	margin of goods sold in absolute value based on planned sales prices	

Ratio	Calculation Expression	Description / Unit	Synonym
markup order margin	= order gross yield * 100 / order volume	markup gross margin of orders based on purchase prices in percent (in %)	*order margin*
promotion gross yield	= promotion sales – promotion purchase	margin of goods sold in absolute value based on promotion sales prices	
discount promotion gross margin	= promotion gross yield * 100 / promotion sales	margin in percent based on sales values referring to a promotion (in %)	*promotion sales margin*
range of coverage	= current inventory / average daily sales	time period daily sales are covered by current inventory (in days)	
shelf productivity	= sales value / contact distance	productivity of shelf space in use measured in sales value / m^2	
stock gross yield	= net sales – stock	margin of stock in absolute value based on planned sales prices	
stock gross margin	= stock gross yield * 100 / stock	markup gross margin of stock (in %)	*stock margin*
inventory shrinkage rate	= inventory difference * 100/ sales	rate of inventory difference in relation to sales (in %)	
…	…		…

Table 3: Table of calculated ratios

4.2 Task specific management views

To prevent information overflow individual excerpts out of dimension hierarchies are required and are combined to task specific views for every management user. For this purpose the concepts *Dimension Scope* and *Dimension Scope Combination* are introduced. Dimension scopes are sub trees of dimensions. Their combination defines a space of multi dimensional objects relevant for a management user. The type of vectors within this space is termed by means of the concept *Reference Object* with respect to RIEBEL's enterprise theory. Reference objects are defined as all "measures, processes and states of affairs which can be object to arrangements or examinations on their own"[48]. Dimension scopes are represented by means of (white) rectangles with (red) triangles inside. In the example case the first dimension scope defined is "Month -> current Month" which is shown for January 2001 (Figure 9). This dimension scope is defined based on the dimension "Date" and comprises the hierarchical levels "Month" and "Date" of the sub tree "January_2001". Based on this definition the second dimen-

[48] Riebel (1979), p. 869

sion scope “Month -> pervious Month” is defined in relation to “January 2001” (Figure 10). The relevance of hierarchical levels for the definition of dimension scopes is shown for the definition of dimension scope “Town” which is an excerpt of dimension “Region” of the dimension grouping “Store” (Figure 11). The hierarchy levels “Region”, “Country” and “Town” are relevant. The Hierarchy level “Store” is ignored. The squares with the “-“ sign indicate that “Town” is the lowest hierarchy level of this dimension scope. Finally dimension scope „CCG Merchandise Department“ is defined based on dimension “Store Assortment CCG” (Figure 12).

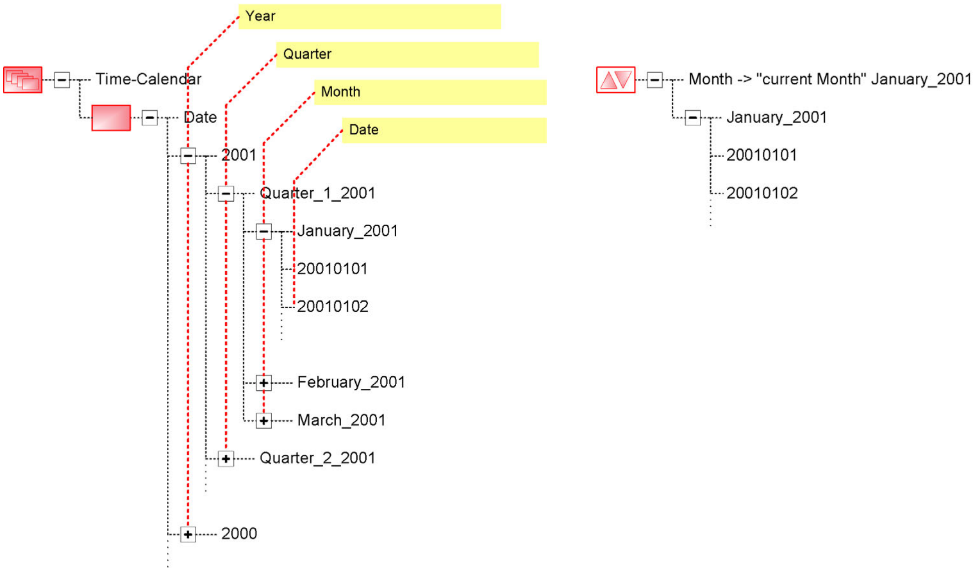


Figure 9: Dimension Scope Month (January 2001)

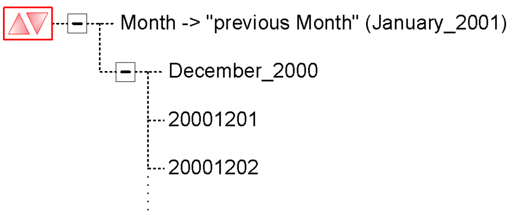


Figure 10: Dimension Scope Month (previous Month in relation to January 2001)

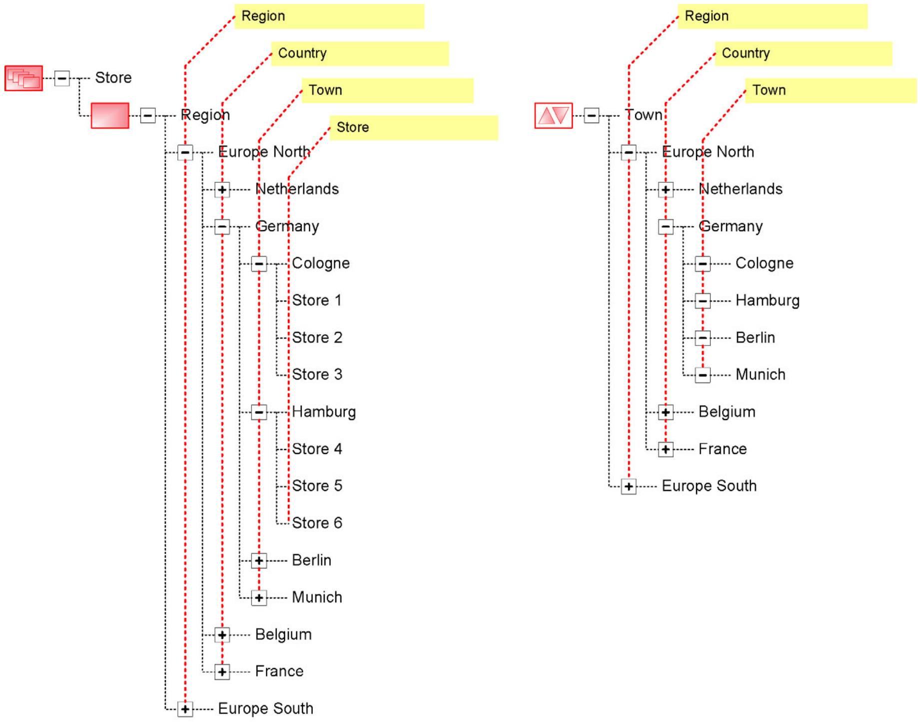


Figure 11: Dimension Scope Town

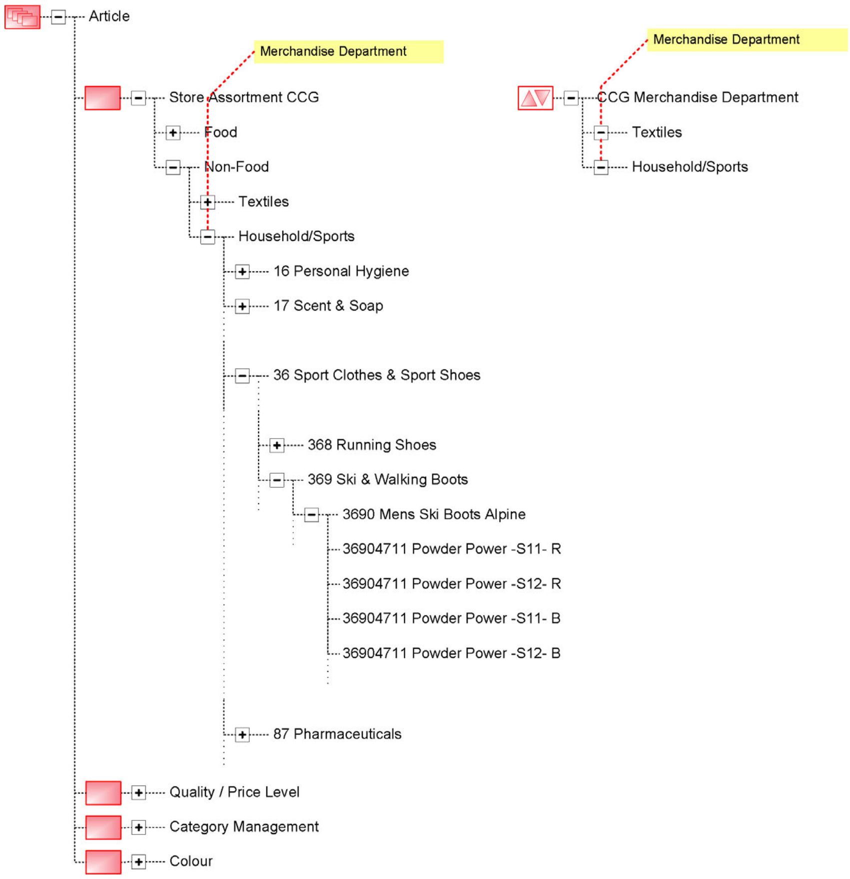


Figure 12: Dimension Scope CCG Merchandise Department

Dimension scope combinations are represented by means of (red) rectangles with small dimension scope symbols inside. The associated dimension scopes are related to dimension scope combinations by means of hierarchy constructs. Having defined the dimension scopes the dimension scope combination can be modelled (Figure 13).

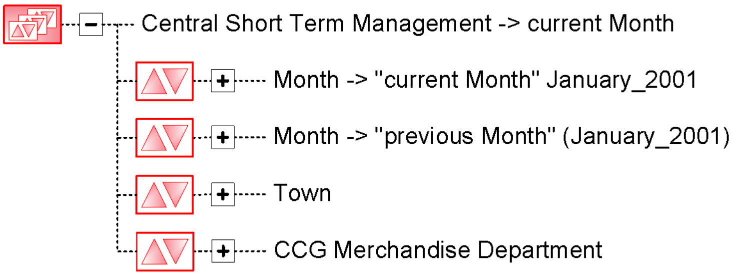


Figure 13: Dimension Scope Combination Central Short Term Management -> current Month

For the definition of management views ratios are assembled to so called ratio systems (concept *Ratio System*). Ratio systems are organized hierarchically (Figure 14) and enable the top down analysis of one unequivocal reference object according to different economical aspects relevant to the management. E.g. the balanced scorecard presented by KAPLAN and NORTON[49] is a set of ratio systems supporting this top down analysis of reference objects in the strategic performance measurement process.

Ratio systems are represented by means of (yellow) boxes with numbers inside and hierarchical structures. The meaning and calculation of every ratio are defined in the library tables (Table 2, Table 3 in section 4.1). Ratio system "Profitability and Store Management" shows the hierarchical structure of its ratios according to their importance for the management user (Figure 14). This means that e.g. ratio "sales" is on a higher level than ratio "returns". "Sales margin" respectively is of the same importance than "sales" but is more important than "asset sales", "promotion sales" etc. This structure implies a certain drill down logic for ratios from the management's point of view beyond an algebraic meaning or definition. It thus supplements the definition of ratios by means of expressions in the library tables.

[49] cf. Kaplan, Norton (1997); Kaplan, Norton (1996); Kaplan, Norton (1992)

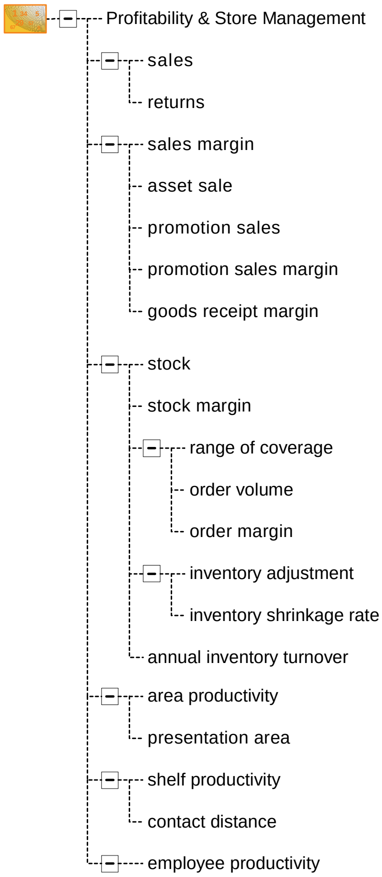


Figure 14: Ratio System Profitability and Store Management

To define information spaces relevant for a management user the set of reference objects (specified by means of a dimension scope combination, e.g. Figure 13) and the set of ratios (specified by means of a ratio system, e.g. Figure 14) must be integrated. For this purpose the concept *Information Object* is introduced. An information object is a relation between a set of reference objects (defined by means of a dimension scope combination) and a set of ratios (defined by means of a ratio system). The type of elements of this relation is termed *Fact*. A fact is a relation of one reference object and one ratio. Having provided dimension scope combinations and ratio systems, information objects for the management users can be modelled. This is shown for the information object “CEO Retailing Company -> current Month” (Figure 15). Information objects are represented by means of (blue) rectangles with a (blue) rhomb inside. The association of dimension scope combinations and ratio systems to information objects is represented by means of the hierarchical constructs. It is noted that concepts introduced in this subsection are also stored in libraries to prevent semantic conflicts. Especially dimension scopes, dimension scope combinations and ratio systems have to be used based unequivocal meaning.

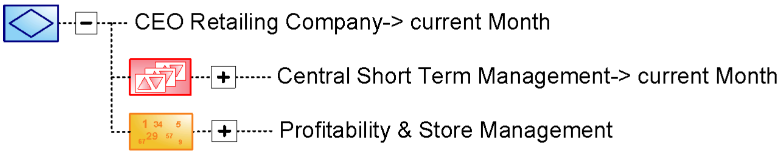


Figure 15: Information Object CEO Retailing Company -> current Month

5 Design and Implementation of Data Marts based on Management Views

In this section the concept of metadata is used to generate logical data mart schemes based on the libraries of management view components (4.1) and report queries based management view definitions (4.2). Suppose a repository scheme for management view components *RS* and a data dictionary scheme *DD*, it is shown that management view components defined by a relation r $\in$ Rel(*RS*) can formally be transformed into a logical data mart scheme *DMS*, with *DMS* $\in$ Rel(*DD*) (section 5.2). Suppose *RS* $\cup$ *DD* = *MDS*, a metadata scheme, it is shown that *MDS* is sufficient to generate report queries against *DMS* (5.2). These queries implement management views defined by means of information objects in section 4.2.

5.1 Data Mart Schema Generation

The specification of management views and the implementation of data marts as phases of the development process are related by means of the design phase. These type level development phases (Figure 1 and Table 1 in section 3) are integrated by means of meta level documents. Therefore the meta model (Figure 16, Table 16 in the Appendix)) is transformed into relational schemes of a repository storing the type level models developed in section 4. The repository relation schemes in Table 4 are used for demonstration purposes. Attributes defining keys are underlined. The development of the repository scheme on the meta level is not in the focus of this paper. The matter of interest is to show in a formalised way how data mart schemes and reports can be generated based on management view components and definitions. For that purpose repository data is used as parameters. With respect to the framework the repository scheme (Table 4) is located in the box “Repository Scheme & Parameters” (meta level, design phase; see Figure 1 and Table 1). The models specifying the management views as content of the repository are located in the box “Scheme & Parameters” (type level, design phase). The algorithms generating data mart schemes (Table 5, Table 6, Table 7, Table 8) are located in framework box “Repository Structures & Programs” (meta level, implementation phase).

Data mart schema generation leads to technical metadata stored in a data dictionary. An approach to model technical metadata for data warehousing environments is proposed by STÖHR ET AL.[50]. According to this approach the bottom of Table 4 shows an excerpt of a data dictionary scheme required for the example.

Repository schemes for data mart schema generation
HierarchyLevel (HL-ID, HL-Name, DG-ID, D-ID, HL-ID-Father)
DimensionObject (DO-ID, DO-Name, DO-Father-ID, D-ID, HL-ID, Leave)
Dimension (D-ID, D-Name, DG-ID)
Ratio (R-ID, R-Name, CalculationExpression, Description)
RatioSynonym (RSynonym-ID, R-ID)
RatioSystem (RS-ID, RS-Name)
R-RS-As (R-ID, RS-ID, R-ID-Father, RS-ID-Father)
DimensionScope (DS-ID, DS-Name, SelectionCondition, D-ID)
DS-HL-As (DS-ID, HL-ID)
DO-DS-As (DO-ID, DS-ID)
DimensionScopeCombination (DSC-ID, DSC-Name)
DS-DSC-As (DS-ID, DSC-ID)
InformationObject (IO-ID,IO-Name DSC-ID, RS-ID)
DimensionGrouping (DG-ID, DG-Name)
DimensionCompatibility (D-ID, D-ID)
Data dictionary scheme (Excerpt)
RelationType (Relation, DW-Role)
ForeignKeyAssociactions (Attribute, Relation, KeyType)

Table 4: Repository and Data Dictionary Scheme

To demonstrate the rules for data mart schema generation based on management views we develop a normalized snowflake scheme (see Table 17 in the Appendix). This scheme must be understood as a *suggestion* for a data mart scheme. A completely automated generation of database schemes is not useful since the data warehouse engineer responsible for the whole data warehouse must be able to react on technical and project specific requirements which are not specified in the management views. On the other hand the warehouse engineer must have the possibility to maintain the warehouse scheme e.g. to speed up query processing if the database has grown over time. Third, the warehouse engineer has to match this suggested scheme with limitations of OLTP systems as data sources.[51] All tasks of the data warehouse engineer belong to the framework box "Scheme & Parameters" (design phase, type level).

The rules for data mart schema generation are formalised by means of (pseudo code) algorithms and relational expressions. To identify names and attributes of data mart relations repository relation schemes "Dimension", "DimensionObject", "HierarchyLevel", "Ratio" (Table 4) are used. The prefix "LK" (look up) is used for every rela-

[50] cf. Stöhr et al. (1999)
[51] cf. Golfarelli, Rizzi (1999); Golfarelli et al. (A) (1998)

tion taking part in the definition of the dimensional structure of the snowflake scheme. The tuples contained in the relations are determined by means of the relation schemes “HierarchyLevel”, “Dimension” and “DimensionObject”. Relations and attributes are inserted into the data dictionary (Table 4 and Table 18 in the Appendix). Concerning the generation of a snowflake scheme the following steps are required:

1. A fact relation is generated (Table 5). The set of key attributes is created based on bottom hierarchy levels. These sets of leaves are identical for every dimension of a dimension grouping (section 4, Table 16 in the Appendix). Every partial key attribute has suffix “ID”. Bottom hierarchy levels are determined by means of the boolean attribute “Leave” of repository scheme “DimensionObject” (Table 4). The set of ratio attributes (measures) is determined by means of repository scheme “Ratio”. The fact relation is given the name of the data mart under consideration with prefix “Facts-DataMart“. The filling of the fact table is not in the focus of this paper. Complex extraction, transformation and loading (ETL) routines are required for this purpose.

Fact relation
Naming of relation
`rel := Facts-DataMart [Name]"`
Insert relation into data dictionary
`Insert (rel, FactTable) into RelationType.`
Set of key attributes given by
$\pi_{\text{HL-ID}}\ \sigma(_{(\text{Leave}=\mathit{TRUE})}(\text{DimensionObject})).$
Naming of key attributes (Suffix ID) and insert into data dictionary
$\forall\ \overline{\text{hl}} \in \pi_{\text{HL-ID}}\ \sigma(_{(\text{Leave}=\mathit{TRUE})}(\text{DimensionObject}))$ `attr :=` $\pi_{\text{HL-Name}}\ \sigma(_{(\text{HL-ID}=\overline{hl})}(\text{HierarchyLevel})) \cup$ `"-ID".`
`Insert (attr, rel, Foreign) into ForeignKeyAssociactions.`
Set of attributes for ratios (measures)
$\pi_{\text{R-ID}}\ (\text{Ratio}).$
Naming of ratio attributes
$\forall \bar{r} \in \pi_{\text{R-ID}}\ (\text{Ratio}):$ $\pi_{\text{R-Name}}\ (\sigma_{(\text{R-ID}=\bar{\text{r}})}(\text{Ratio})).$

Table 5: Generation of fact relation

2. For every dimension defined as master data for management views a relation is generated and filled by tuples for the attributes with suffixes “ID” and “Description” (Table 6). Attributes with suffix “ID” are primary keys. These relations have no for-

eign key attributes. They are the top level relations of the snowflake scheme's branches. Repository relation "Dimension" contains the relevant information.

Top level relations without foreign keys
Set of relations given by
$\pi_{\text{D-ID, D-Name}}$ (Dimension).
Naming of relations and insert into data dictionary
$\forall \bar{d} \in \pi_{\text{D-ID}}$ (Dimension) `rel := "LK-"` $\cup$ $\pi_{\text{D-Name, D-ID}}(\sigma_{(\text{D-ID}=\bar{d})}(\text{Dimension}))$
`Insert (rel, LookUpTable) into RelationType.`
Attributes of relations
Suffix ID (primary key) and insert into data dictionary
`attr :=` $\pi_{\text{D-Name}}(\sigma_{(\text{D-ID}=\bar{d})}(\text{Dimension})) \cup$ `"-ID".`
`Insert (attr, rel, Primary) into ForeignKeyAssociactions.`
Suffix Description
$\pi_{\text{D-Name}}(\sigma_{(\text{D-ID}=\bar{d})}(\text{Dimension})) \cup$ `"-Description"`
Filling of top level relations
$\forall \bar{d} \in \pi_{\text{D-ID}}$ (Dimension) `tuple :=` $\pi_{\text{D-ID, D-Name}}(\sigma_{(\text{D-ID}=\bar{d})}(\text{Dimension}))$.
`Insert (tuple) into rel.`

Table 6: Generation and filling of top level relations of snowflake scheme

3. For every top hierarchy level of dimensions in the management view master data libraries a relation is generated and filled by tuples for one primary key and one foreign key attribute (suffixes "ID") and another attribute "Description" (Table 7). The foreign key attributes relate to the respective relations generated in step 2. Repository relations "HierarchyLevel", "Dimension" and "DimensionObject" are relevant for this step.

Relations for top hierarchy levels
Set of relations given by
$\pi_{\text{HL-ID, HL-Name}}\ (\sigma_{(\text{HL-Father-ID=NIL})}(\text{HierarchyLevel})).$
Naming of relations and insert into data dictionary
$\forall\, \overline{\text{hl}} \in \pi_{\text{HL-ID}}\ (\sigma_{(\text{HL-Father-ID=NIL})}(\text{HierarchyLevel}))$ `rel := "LK-"` $\cup\ \pi_{\text{HL-Name, HL-ID}}\ (\sigma_{(\text{HL-ID}=\overline{hl})}(\text{HierarchyLevel})).$
`Insert (rel, LookUpTable) into RelationType.`
Attributes of relations
Suffix ID (primary key) and insert into data dictionary
`attr :=` $\pi_{\text{HL-Name}}(\sigma_{(\text{HL-ID}=\overline{\text{hl}})}\ (\text{HierarchyLevel}))\ \cup$ `"-ID"`
`Insert (attr, rel, Primary) into ForeignKeyAssociactions.`
Suffix Description
$\pi_{\text{HL-Name}}(\sigma_{(\text{HL-ID}=\overline{\text{hl}})}\ (\text{HierarchyLevel}))\ \cup$ `"-Description"`
Foreign key attribute and insert into data dictionary
`attr :=` $\pi_{\text{D-Name}}\ (\text{Dimension} >< (\sigma_{(\text{HL-ID}=\overline{\text{hl}})}(\text{HierarchyLevel})))\ \cup$ `"-ID"`.
`Insert (attr, rel, Foreign) into ForeignKeyAssociactions.`
Filling of relations for top hierarchy levels
$\forall\, \overline{\text{hl}} \in \pi_{\text{HL-ID}}\ (\sigma_{(\text{HL-Father-ID=NIL})}(\text{HierarchyLevel})):$ `tuple :=` $\pi_{\text{DO-ID, DO-Name, D-ID}}(\sigma_{(\text{HL-ID}=\overline{\text{hl}})}\ (\text{DimensionObject})).$
`Insert (tuple) into rel.`

Table 7: Generation and filling of snowflake scheme relations for top hierarchy levels

4. For every other hierarchy level of the management view's master data relations are generated and filled by tuples for one primary key attribute, an attribute "Description" and a set of foreign key attributes. Primary and foreign key attributes have the suffix "ID" (Table 8). Sets of foreign keys are required for the bottom hierarchy levels since leave sets of all dimensions belonging to one dimension grouping in the management views are identical (section 4, Table 16 in the Appendix). These leaves are transformed into tuples of bottom relations in the snowflake scheme. Management view leave sets of different dimensions belonging to one dimension grouping are different representations (in the sense of views) of the same set of business objects. E.g. article "36904711 Powder Power -S11- R" belongs to dimension grouping "Article" and is an element of bottom hierarchy levels of dimensions "Store Assortment CCG" (Figure 2, section 4) and "Colour" (Figure 5, section 4) as well. "Store 3" respectively belongs to dimension grouping "Store" and is an element of

bottom hierarchy levels of dimensions “Region” (Figure 7, section 4) and “Area / Location” (Figure 8, section 4) as well. From an algebraic point of view this leads to isomorphisms relating the leave sets within the same dimension grouping. With respect to these isomorphisms there exist equivalence classes (e.g. the leave sets of all articles and stores). Every of these equivalence classes is transformed into one relation. E.g. relations for bottom hierarchy levels “LK-Article” and “LK-Store” (Table 17, Table 18 in the Appendix). The resulting relations comprise a set of foreign key attributes since they are related to a set of branches of the snowflake scheme. E.g. “LK-Store” is related by foreign key “Town-ID” to “LK-Town” belonging to dimension “Region” and by foreign key “Area-ID” to “LK-Area” belonging to dimension “Area / Location” (Table 17, Table 18 in the Appendix). For every hierarchy level superior to the bottom level and inferior to the top hierarchy level the same relational expressions are used, nevertheless leading to exactly one foreign key attribute for the respective relations only.

Relations for hierarchy levels with superior levels
Set of relations given by
$\pi_{\text{HL-ID, HL-Name}}\,(\sigma_{(\text{HL-Father-ID}\neq\text{NIL})}(\text{HierarchyLevel})).$
Naming of relations and insert into data dictionary
$\forall\,\overline{\text{hl}}\in\pi_{\text{HL-ID}}\,(\sigma_{(\text{HL-Father-ID}\neq\text{NIL})}(\text{HierarchyLevel}))$ $\texttt{rel := “LK-”} \cup \pi_{\text{HL-Name, HL-ID}}\,(\sigma_{(\text{HL-ID}=\overline{hl})}(\text{HierarchyLevel})).$
`Insert (rel, LookUpTable) into RelationType.`
Attributes of relations
Suffix ID (primary key) and insert into data dictionary
$\texttt{attr :=}\ \pi_{\text{HL-Name}}(\sigma_{(\text{HL-ID}=\overline{\text{hl}})}\,(\text{HierarchyLevel})) \cup \texttt{“-ID”}$
`Insert (attr, rel, Primary) into ForeignKeyAssociactions.`
Suffix Description
$\pi_{\text{HL-Name}}(\sigma_{(\text{HL-ID}=\overline{\text{hl}})}\,(\text{HierarchyLevel})) \cup \texttt{“-Description”}$
List of foreign key attributes) and insert into data dictionary
$\forall\,\overline{\text{fk}}\in\pi_{\text{HL-Name}}(\text{HierarchyLevel} >< (\rho_{HL-ID\leftarrow HL-ID-Father}(\pi_{\text{HL-ID-Father}}(\sigma_{(\text{HL-ID}=\overline{hl})}\,(\text{HierarchyLevel}))))):$ $\texttt{attr :=}\ \overline{\text{fk}} \cup \texttt{“-ID”}.$
`Insert (attr, rel, Foreign) into ForeignKeyAssociactions.`
Filling of relations for hierarchy levels with superior levels
Leave elements of dimension hierarchies can have a set of fathers in different dimensions. This set is given by
$\forall\,\overline{\text{hl}}\in\pi_{\text{HL-ID}}\,(\sigma_{(\text{HL-Father-ID}\neq\text{NIL})}(\text{HierarchyLevel}))$ $\forall\,\overline{\text{do}}\in\pi_{\text{DO-ID}}\,(\sigma_{(\text{HL-ID}=\overline{hl})}(\text{DimensionObject}))$ $\texttt{[List of Fathers]:=}\ \pi_{\text{DO-Father-ID}}(\sigma_{(\text{DO-ID}=\overline{\text{do}})}\,(\text{DimensionObject})).$
Using [List of Fathers] the set of tuples for every relation is given by
$\texttt{tuple :=}\ \pi_{\text{DO-ID, DO-Name, [List of Fathers]}}(\sigma_{(\text{HL-ID}=\overline{\text{hl}})}\,(\text{DimensionObject})).$
`Insert (tuple) into rel.`

Table 8: Generation and filling of snowflake scheme relations for hierarchy levels with superior levels

5.2 Report Generation

Using technical data mart metadata stored in a data dictionary, report queries implementing the management views (defined in section 4.2) can be generated. This section shows the formal transformation of repository information into report queries. With respect to the development framework (Figure 1, Table 1) the algorithms generating report queries (Table 9, Table 10, Table 11, Table 12) belong to the implementation phase on the meta level (box “Repository Structures & Programs”). Resulting queries implementing reports (Table 13) belong to the implementation phase on the type level (box “Structures & Programs”). Repository data and data dictionary data are used as parameters for report query generation algorithms. This data therefore belongs to the design phase on the type level (box “Scheme & Parameters”).

In the following we use information object *CEO Retailing Company* -> *current Month* (Figure 15) specified in section 4.2 as an example. The algorithms presented intend to show that the specification of management views based on the approach introduced in section 4 is formally enough to generate data base reports. Any kind of algorithmic or database efficiency is not the matter here.

Using relation algebra data mart queries generated are of the form

$$\pi_{\text{[ListOfDimensions], [ListOfRatios]}} (\sigma_{\text{([SelectionExpression])}}((\text{JoinExpression}))) .$$

The algorithms relate to the repository and the data dictionary scheme shown in Table 4 (section 5.1), the snowflake example and the respective data dictionary examples (Table 17, Table 18 in the Appendix). Based on some basic settings the form of the queries leads to three main generation steps:

1. The basic settings comprise the identification of an information object and its defining components (dimension scope combination and ratio system). Additionally the set of top level ratios is selected from the ratio system and the dimensions relevant for the report are identified (Table 9).

Information object identification and basic settings
Identification of information object
$\overline{io} := \pi_{\text{IO-ID}} \, (\sigma_{(\text{IO-Name} = \text{"CEO Retailing Company -> current Month"})}(\text{InformationObject})$
$\overline{dsc} := \pi_{\text{DSC-ID}} \, (\sigma_{(\text{IO-ID} = \overline{\text{io}})}(\text{InformationObject})$
$\overline{rs} := \pi_{\text{RS-ID}} \, (\sigma_{(\text{IO-ID} = \overline{\text{io}})}(\text{InformationObject})$
Top level ratios of report and naming of ratio columns

`ReprortRatios := (RR-ID, RR-Name).`
`ReportRatios :=`
$\rho_{RR-ID \leftarrow R-ID, RR-Name \leftarrow R-Name}(\pi_{\text{R-ID-Father, R-Name}}((\text{Ratio}) \bowtie (\sigma_{(\text{RS-ID}=\overline{rs}, \text{ R-ID}=\bar{\text{r}})}(\text{R - RS - As}))))$

Dimensions of report
`ReportDimensions := (RD-ID, RD-ColumnName).` $\pi_{\text{RD-ID}}(\text{ReportDimensions}) :=$ $\pi_{\text{D-ID}} \, (\sigma_{(\text{DSC-ID} = \overline{\text{dsc}})}(\text{DS - DSC - As}) \bowtie (\text{DimensionScope}))$

Table 9: Report query generation: Basic settings

2. The join expression of a report query is constructed (Table 10) based on joining look up relations of every dimension scope top down (loop-1). The algorithm starts by identifying the dimension scope’s top hierarchy levels based on repository relations (“HierarchyLevels”, “DS-HL-As”) in loop-2. The top hierarchy level of a dimension scope ($\overline{\text{hl - father}}$) is used as starting point to identify join operations (loop-3). This proceeds using foreign key relations stored in the data dictionary (relation “ForeignKeyAssociation”; see Table 4, section 5.2 and Table 18 in the Appendix). The loop terminates if the fact table is reached (attribute “DW-Role” of relation “RelationType” stored in the data dictionary is checked for this purpose). The join expression itself is constructed based on the set of join elements (stored in structure “DS-Joins”). The sets of joins resulting out of one dimension scope are put into brackets and are themselves joined in loop-4. The resulting join expression is shown as part of the report query in Table 13.

Generation of join expression

[JoinExpression] := NIL.

DS-Joins := (JoinElement, DS-ID).

LOOP-1

$\forall\, \overline{ds} \in (\sigma_{(DSC\text{-}ID = \overline{dsc})}\ (DS\text{-}DSC\text{-}As))$

Calculation of top hierarchy level and naming of report dimension columns

$(DS\text{-}HierarchyLevels) := \pi_{HL\text{-}ID,\, HL\text{-}ID\text{-}Father}((DS\text{-}HL\text{-}As) \bowtie (HierarchyLevel))$

LOOP-2

$\forall\, \overline{hl} \in \pi_{HL\text{-}ID}(DS\text{-}HL\text{-}As)$:

$\overline{hl\text{-}father} := \pi_{HL\text{-}ID\text{-}Father}(\sigma_{HL-ID=\overline{hl}}(DS\text{-}HierarchyLevels))$

IF $(\overline{ds}, \overline{hl\text{-}father}) \notin (DS\text{-}HL\text{-}As)$: ($\overline{hl\text{-}father}$ is top hierarchy level of $\overline{ds}$)

ColumnDimension:= $\pi_{HD\text{-}ID}(\sigma_{HL-ID=\overline{hl-father}}(HierarchyLevel))$.

ColumnName:= $\pi_{HL\text{-}Name}(\sigma_{HL-ID=\overline{hl-father}}(HierarchyLevel))$.

$\pi_{RD\text{-}ColumnName}(\sigma_{(RD-ID=ColumnDimension)}(ReportDimensions)$:= ColumnName.

END OF LOOP-2

Finding look up relation with respect to hierarchy level in data dictionary

RelationName := "LK-" $\cup\ \pi_{HL\text{-}Name}\ (\sigma_{(HL\text{-}ID = \overline{hl\text{-}father})}(HierarchyLevel)$

father := $\pi_{Relation}\ (\sigma_{(Relation = RelationName)}(ForeignKeyAssociation)$

Find primary key of relation

key := $\pi_{Attribute}\ (\sigma_{(Relation = father\ \wedge\ KeyType = Primary)} ForeignKeyAssociation)$

Identification of join operations for dimension scope

LOOP-3

son := $\pi_{Relation}\ (\sigma_{(Attribute = key\ \wedge\ KeyType = Foreign)} ForeignKeyAssociation)$

(Since the snowflake scheme is normalised, there is only one relation possible)

Add ("(" father " $\bowtie$ " son ")", $\overline{ds}$) to DS-Joins.

IF FactTable = $\pi_{DW\text{-}Role}\ (\sigma_{(Relation = son)} RelationType)$:

END OF LOOP-3

ELSE:

key := $\pi_{Attribute}\ (\sigma_{(Relation = son\ \wedge\ KeyType = Primary)} ForeignKeyAssociation)$.

father := son.

Calculate join expression for report

[ListOfJoins] := $\pi_{JoinElement}\ (\sigma_{(DS\text{-}ID = \overline{ds})}(DS\text{-}Joins))$.

operand := FirstElement [ListOfJoins].

```
IF operand <> NIL:
       IF [JoinExpression] NotEmty:
       Add " ><  (" to [JoinExpression].
       Else:
       Add "(" to [JoinExpression].
```

	Generation of join expression - continued
	`Remove FirstElement [ListOfJoins].`
	`Add operand to [JoinExpression].`
	`LOOP-4` `operand := FirstElement [ListOfJoins].`

	`IF` `operand <> NIL:`
	`Add " >< " to JoinExpression.`
	`Add operand to [JoinExpression].`
	`Remove FirstElement [ListOfJoins].`

	`ELSE:` `Add ")" to [JoinExpression].` **`END of LOOP-4`**`.`
`ELSE:` **`END OF LOOP-1`**	

Table 10: Report query generation: Join expression

3. Selections on certain hierarchy levels like in e.g. dimension scope *Month -> "current Month" January_2001* (Figure 9, section 4.2) or *Month -> "previous Month" (January_2001)* (Figure 10, section 4.2) are transformed into selection expressions of the report query (Table 11). Selections related to the same hierarchy level (in the example case to hierarchy level "Month") are unified by operator " ∨ " (loop-6). The unified sets of selections belonging to different hierarchy levels are related by intersection operator " ∧ " (loop-7; not in the example). The resulting selection expression is shown as part of the report query in Table 13.

Generation of selection expression
Consider selection condition (e.g. "current Month" -> January_2001) of dimension scopes
Union of selections on identical hierarchy levels
HL-SelectionConditions := (HL-ID, SelectionCondition). HL-SelectionConditions := $\pi_{\text{HL-ID, SelectionCondition}}((\text{DS-HL-As}) \bowtie (\text{DimensionScope}))$.
Loop-5 HL-Selections := (HL-ID, HL-Selection-Union). $\forall\, \overline{\text{hl}} \in \text{HL-SelectionConditions}:$ [ListOfSelections] := $\sigma_{(\text{HL-ID}=\overline{\text{hl}})}(\text{HL-SelectionConditions})$. HL-Selection-Union := NIL.
IF NIL <> [ListOfSelections]: **LOOP-6** selection := FirstElement [ListOfSelections]. Remove FirstElement [ListOfSelections]. Add selection to HL-Selection-Union. **IF** NIL <> [ListOfSelections]: Add " $\vee$ "to HL-Selection-Union. **Else:** **END of LOOP-6.**
Add ($\overline{\text{hl}}$, HL-Selection-Union) to HL-Selections. **END of LOOP-5.**
Intersection of selections on different hierarchy levels
[SelectionExpression] := NIL.
[ListOfArguments] := $\pi_{\text{HL-Selection-Union}}(\text{HL-Selections}):$
argument := FirstElement [ListOfArguments] **IF** argument <> NIL: **LOOP-7** Add "(" argument ")" to [SelectionExpression] Remove FirstElement [ListOfArguments] argument := FirstElement [ListOfArguments] **IF** argument <> NIL: Add " $\wedge$ " to [SelectionExpression] **Else:** **END of LOOP-7.**

Table 11: Report query generation: Selection expression

4. The projection expression is constructed based on the set of top level ratios (structure "ReportRatios") and the dimensions relevant for the report (Table 12). These dimensions were calculated in step 2 (join expression). The report columns are given the names of the top hierarchy levels of the dimension scopes under consideration (see the resulting projection expression as part of the report query in Table 13). For this purpose the structure $\overline{\text{hl-father}}$ and loop-2 in the calculation of the join expression (see Table 10 in step 2) are used. The respective information is stored in structure "ReportDimensions".

Projection expression and report query assembly
`[ListOfDimensions] :=` $\pi_{\text{RD-ID}}(\text{ReportDimensions})$.
`[ListOfRatios] :=` $\pi_{\text{RR-ID}}(\text{ReportRatios})$.
`Projection expression and query assembly`
`report query :=` $\pi_{\text{[ListOfDimensions], [ListOfRatios]}}\ (\sigma_{\text{([SelectionExpression])}}((\text{JoinExpression})))$.

Table 12: Report query generation: Projection expression and query assembly

The resulting report query for information object "CEO Retailing Company -> current Month" (Figure 15, section 4.2) is shown in Table 13. This query is a basic query for the information object under consideration. This means that information is presented with respect to the top hierarchy levels of the information object's dimension scopes. In the example case these are the hierarchy levels "Month", "Region" and "Merchandise Department". The report columns are given these names (shown as "NAME" within the projection expression). The joins respectively start at these top hierarchy levels for every dimension scope (e.g. relations LK-Month, LK-Region [HierarchyLevel xxx] and LK-Merchandise Department). Since the algorithm constructing the join expression contains a basic loop (loop-1) for every dimension scope of the dimension scope combination under consideration (Table 10) and there are two dimension scopes related to the same dimension "Date" (these are dimension scopes *Month -> "current Month" January_2001* (Figure 9, section 4.2) and *Month -> "previous Month" (January_2001)* (Figure 10, section 4.2)) having the same top hierarchy level ("Month"), the first two parts of the resulting join expression are identical. Solving this redundancy will be part of query optimisation tasks which is not the matter of interest here.

Report query information object “CEO Retailing Company -> current Month”

$$\pi_{\text{Date (“Month”), Region [Dimension xxx] (“Region”), Store Assortment CCG (“Merchandise Department”) sales, sales margin, stock, area productivity, shelf productivity, employee productivit}}$$

$$(\sigma_{\text{(Month-ID = January\_2001} \vee \text{Month-ID = December\_2000)}}$$

```
(
       (LK-Month ⋈  LK-Date [HierarchyLevel xxx])
        ⋈ (LK-Date [HierarchyLevel xxx]  ⋈  Facts-DataMart "Excom")
)
 ⋈ (
       (LK-Month ⋈  LK-Date [HierarchyLevel xxx])
        ⋈ (LK-Date [HierarchyLevel xxx]  ⋈  Facts-DataMart "Excom")
)
 ⋈ (
       (LK-Region [HierarchyLevel xxx]  ⋈  LK-Country)
        ⋈ (LK-Country ⋈  LK-Town)
        ⋈ (LK-Town ⋈  LK-Store)
        ⋈ (LK-Store ⋈  Facts-DataMart "Excom")
)
 ⋈ (
        ⋈ (LK-Merchandise Department ⋈  LK-2-digit merchandise category)
        ⋈ (LK-2-digit merchandise category ⋈  LK-3-digit merchandise category)
        ⋈ (LK-3-digit merchandise category ⋈  LK-4-digit merchandise category)
        ⋈ (LK-4-digit merchandise category ⋈  LK-Article)
        ⋈ (LK-Article ⋈  Facts-DataMart "Excom")
)
```

Table 13: Basic report query for information object “CEO Retailing Company -> current month” (Figure 15, section 4.2)

With respect to the basic query generated, drill operations lead to the following changes of the algorithms:

1. Drill down or drill up operations change the top hierarchy level of a dimension scope. Drill down will shift the top to the next lower hierarchy level and drill up to the next higher level (Table 14). Report column names are changed with respect to these operations. For every drilling operation it must be checked if the drill under consideration is allowed with respect to the definition of the management view. For this purpose the changed top hierarchy level is matched against the dimension scope definition. To prevent information overflow only drill operations consistent to the information object definition are transformed into report queries. These operations change loop-2 in the algorithm generating the join expression. Loops 3 and 4 remain unchanged. For all other dimension scopes the join expressions are generated in the way defined in Table 10.

Drilling operations within dimension scope combination
Changes in join expression
LOOP-1 Not used for $\overline{\mathrm{ds}}$ but for all other dimension scopes.
Given $\overline{\mathrm{ds}}$ and $\overline{\mathrm{hl\text{-}father}}$ as top hierarchy level.
Drill down
newFather := $\pi_{\text{HD-ID}}(\sigma_{HL-ID-Father=\overline{hl-father}}(\text{HierarchyLevel}))$.
IF $(\overline{\mathrm{ds}}, \text{newFather}) \in (\text{DS-HL-As})$: (drill down is allowed) ColumnDimension:= $\pi_{\text{HD-ID}}(\sigma_{HL-ID=newFather}(\text{HierarchyLevel}))$. ColumnName:= $\pi_{\text{HL-Name}}(\sigma_{HL-ID=newFather}(\text{HierarchyLevel}))$. $\pi_{\text{RD-ColumnName}}(\sigma_{(RD-ID=ColumnDimension)}(\text{ReportDimensions})$:= ColumnName.
Drill up
newFather := $\pi_{\text{HD-ID-Father}}(\sigma_{HL-ID=\overline{hl-father}}(\text{HierarchyLevel}))$.
IF $(\overline{\mathrm{ds}}, \text{newFather}) \in (\text{DS-HL-As})$: (drill down is allowed) ColumnDimension:= $\pi_{\text{HD-ID}}(\sigma_{HL-ID=newFather}(\text{HierarchyLevel}))$. ColumnName:= $\pi_{\text{HL-Name}}(\sigma_{HL-ID=newFather}(\text{HierarchyLevel}))$. $\pi_{\text{RD-ColumnName}}(\sigma_{(RD-ID=ColumnDimension)}(\text{ReportDimensions})$:= ColumnName.
LOOP-3 ...
LOOP-4 ...

Table 14: Report generation: Drilling operations

2. Since only top level ratios are used to generate the projection expression of the basic report query drills with respect to the definition of the ratio system are possible. In the example case of ratio system “Profitability & Store Management” (Figure 14, section 4.2) drills will lead to changes in the projection expression and the basic settings (Table 9, Table 12). The changes are related to structure “ReportRatios” and are based on repository relation “R-RS-As” (Table 4, section 5.2) storing father son relationships for ratios (Table 15).

Drill within ratio system
$\overline{rs} := \pi_{\text{RS-ID}} \ (\sigma_{(\text{IO-ID} = \overline{\text{io}})}(\text{InformationObject})$.
$\overline{\text{r}} \in \overline{rs}$.
`ReprortRatios := (RR-ID, RR-Name).`
Drill down
`ReportRatios :=` $\rho_{RR-ID \leftarrow R-ID, RR-Name \leftarrow R-Name}(\pi_{\text{R-ID, R-Name}}((\text{Ratio})$ $>< (\sigma_{(\text{RS-ID}=\overline{rs}, \text{ R-ID-Father} = \overline{\text{r}})}(\text{R - RS - As}))))$
Drill up
`ReportRatios :=` $\rho_{RR-ID \leftarrow R-ID, RR-Name \leftarrow R-Name}(\pi_{\text{R-ID-Father, R-Name}}((\text{Ratio})$ $>< (\sigma_{(\text{RS-ID}=\overline{rs}, \text{ R-ID} = \overline{\text{r}})}(\text{R - RS - As}))))$

Table 15: Report generation: Drilling within ratio system

6 Discussion and Outlook

The development framework presented in section 3 is based on the separation of three development phases which are combined with three abstraction layers. Compared to the DWQ framework[52] the same phases are separated and called perspectives. DWQ presents a metadata framework and a repository meta model which are characterized by the three perspectives "conceptual perspective", "logical perspective" and "physical perspective".[53] Terms and concepts to describe documents and models required in different phases of data warehouse development processes are provided. These documents are integrated by means of a central repository. This corresponds to the central role of repository and metadata belonging to the development framework defined in this paper. Referring to this framework the box "Scheme & Parameters" (type level, design phase; Figure 1 and Table 1 in section 3) fulfils a central coordination function for the integration of phases in development processes (see section 5).

The three abstraction layers characterising the DWQ framework are not identical to the ones of the framework presented in this paper. Compared to the development framework of Figure 1 DWQ is characterised by the type level, the meta level and a meta meta level. With respect to the IRDS framework[54] (see section 3) the DWQ framework deals with the upper three layers whereas the framework presented in this paper deals with the lower three layers. The intersection thus are the type and the meta level with respect to the development framework discussed here (Figure 1). Since the DWQ approach concentrates on the development of methodologies in general, the process of *process development* is analysed in more detail than in the approach presented in this paper. DWQ concentrates on quality oriented development processes.[55] The quality oriented aspect of usefulness, dealing which the data warehouse access according to users' work[56], has strong relations to the approach presented in this paper.

Concerning the development and role of conceptual models the approach presented in this paper is related to the work of GOLFARELLI ET AL. and the work of LECHTENBÖRGER. The approach presented by GOLFARELLI and RIZZI[57] is a bottom up approach allowing the formalized analysis of OLTP system's data structures and the transformation of this analysis into a conceptual description. The limitations of OLTP systems are taken care of. Nevertheless there is no specification of management views

[52] cf. Jarke et al. (2000); Jarke et al. (1999); Vassiliadis et al. (2000)
[53] cf. Jarke et al. (1999); Jarke et al. (2000)
[54] cf. ISO (1990); Pohl (1996); Jacobs, Holten (1995)
[55] See the meta model based approach in Jarke et al. (2000), pp. 128, 135
[56] cf. Jarke et al. (2000)
[57] cf. Golfarelli, Rizzi (1998); Golfarelli et al. (A) (1998); Golfarelli et al. (B) (1998); Golfarelli, Rizzi (1999)

in the sense presented in this paper. The fact model is a technically motivated data model which is formally integrated with database scheme analysis and development. The findings of GOLFARELLI ET AL. can be integrated with the approach presented in this paper. For this purpose the algorithms generating a logical data mart scheme discussed in section 5.1 have to check the limitations of the OLTP systems analysed with the fact model approach. Future work has to show the potential of an integration of both approaches. An integration could be fruitful to come even closer to a completed method of information warehouse development. For that purpose the language concepts characterizing the both approaches have to be integrated by means of a meta model.

The approach of data warehouse schema design presented by LECHTENBÖRGER[58] can be supplemented by the approach presented in this paper. LECHTENBÖRGER[59] explicitly neglects methods concerning the specification of management views. Nevertheless the work of LECHTENBÖRGER shows formally how management requirements (which are given as examples) are transformed into conceptual data warehouse schemes. For that purpose quality criteria are introduced. LECHTENBÖRGER's work and the approach presented in this paper could be integrated to generate logical data mart schemes with respect to the formal criteria proposed by LECHTENBÖRGER. For this purpose the algorithms in section 5 must be integrated with the formalisms proposed by LECHTENBÖRGER. This integration is matter of future work.

The approach presented here is based on prototypical implementations. HOLTEN[60] sketches by means of a scenario how management view specifications can be transformed into metadata of ROLAP and ETL tools available on the market. Based on a central repository serving as library for management view components and definitions CRISANDT[61] and HILBERS[62] develop a prototype generating data mart schemes and query reports in a SQL environment. Future work will concentrate on the integration of the approaches presented by GOLFARELLI ET AL. and LECHTENBÖRGER. Additionally there are problems concerning the integration of management theory and data warehouse loading to solve. From a management theoretical point of view bookings on any combination of dimension objects, especially dimension objects on higher hierarchy levels, are required. This leads to further requirements concerning logical data mart schemes, especially concerning different fact relations and their integration. Furthermore questions concerning report query generation occur. The generation of join expression will

58 cf. Lechtenbörger (2001)
59 cf. Lechtenbörger (2001), p. 110
60 cf. Holten (2000)
61 cf. Crisandt (2000)
62 cf. Hilbers (2000)

become by far more complex if management theoretical conditions are taken into consideration.

References

Agarwal, S.; Agrawal, R.; Deshpande, P. M.; Gupta, A.; Naughton, J. F.; Ramakrishnan, R., Sarawagi; S.: On the computation of multidimesnional aggregates. Proc. 22nd International Conference on Very Large Data Bases, San Francisco, CA, Morgan Kaufmann, 1996, pp. 506-521.

Agrawal, R.; Gupta, A.; Sarawagi, S.: Modeling multidimensional databases. Proc. 13th International Conference on Data Engineering, Los Alimatos, CA, IEEE Computer Society Press, 1997, pp. 232-243.

Balzert, H.: From OOA to GUIs: The JANUS system. Journal of Object-Oriented Programming, (1996) 2, pp. 43-47.

Batini, C.; Ceri, S.; Navathe, S. B.: Conceptual Database Design. An Entity-Relationship Approach. Benjamin, Cummings. Redwood City 1992.

Bauer, A.; Huemmer, W.; Lehner, W.: An Alternative Relational OLAP Modeling Approach. Second International Conference on Data Warehousing and Knowledge Discovery (DAWAK 2000, London, U.K., September 4-6), pp. 189-198.

Becker, J.; Holten, R.: Fachkonzeptuelle Spezifikation von Führungsinformationssystemen. WIRTSCHAFTSINFORMATIK 40 (1998) 6, pp. 483-492.

Blaschka, M.; Sapia, C.; Höfling, G.; Dinter, B.: Finding Your Way through Multidimensional Data Models. Proceedings of the International Workshop on Data Warehouse Design and OAP Technology (DWDOT 1998), Vieanna, Austria, August 28, 1998, pp. 198-203.

Boehm, B.: Software-Engineering Economics. Pretnice Hall 1981.

Böhnlein, M.: Konstruktion semantischer Data-Warehouse-Schemata. Deutscher Universitäts-Verlag, Wiesbaden 2001.

Borgman, H. P.: Manager's information search behaviour using Management Support Systems. Wirtschaftsinformatik 40 (1998), 6, pp. 527 – 535.

Brealey, R. A.; Myers, S. C.: Principles of Corporate Finance. 5th ed. McGraw-Hill Companies, Inc. New York et al. 1996.

Bulos, D.: OLAP Database Design. A New Dimension. Database Programming & Design (DBPD), Vol. 9, No. 6, June 1996, pp. 33-37.

Cabibbo, L.; Torlone, R.: An Architecture for Data Warehousing Supporting Data Independence and Interoperability. International Journal of Cooperative Information Systems, Vol. 10, No. 3 (2001), pp. 377-397.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Nardi, D.; Rosati, R.: Data Integration in Data Warehousing. International Journal of Cooperative Information Systems, Vol. 10, No. 3 (2001), pp. 237-271.

CCG 1997: Standard-Warngruppenklassifikation für Verbrauchsgüter, Gebrauchsgüter, Investitionsgüter und Rohstoffe. 2nd Ed. Centrale für Coorganisation GmbH, Cologne 1997.

Chaudhuri, S.; Dayal, U.: An overview of data warehousing and olap technology. ACM SIGMOD Record, 26 (1997) 1, pp. 65-74.

Codd, E. F.: The Relational Model for Database Management. Version 2. Reading, MA, Addison-Wesley, 1990.

Codd, E. F.; Codd, S. B.; Salley, C. T.: Providing olap (On-Line Analytical Processing) to User Analysts: An IT Mandate. White Paper, E. F. Codd & Asociates, 1993. http://www.arborsoft.com/essbase/wht_ppr/coddTOC.html, downloaded 1998-06-05.

Colliat, G.: olap, relational, and multidimensional database systems. ACM SIGMOD Record 25 (3), pp.64-69.

Copeland T.; Koller, T.; Murrin, J.: Valuation: Measuring and Managing the Value of Companies. John Wiley&Sons, New York et al. 1990.

Crisandt, J.: Integration eines SQL-Generators für ROLAP-Anfragen in eine Data-Warehouse-Entwicklungsumgebung. Diplomarbeit Universität Münster 2000.

Date, C. J.: An Introduction to Database Systems. Volume 1. 5th ed. Reading, MA, Addison-Wesley, 1990.

Davis, A. M.: The Analysis and Specification of Systems and Software Requirements. In: R. H. Thayer, M. Dorfman (Eds.): Systems and Software Requirements Engineering, IEEE Computer Society Press-Tutorial 1990, pp.119-134.

Davis, A. M.; Bersoff, E. H.; Comer, E. R.: A Strategy for Comparing Alternative Software Development Life Cycle Models. IEEE Transactions on Software Engineering 10 (14) 1988, pp. 1453-1461.

Eccles, R. G.: The performance measurement manifesto. Harvard Business Review, 69 (1991) January-February, pp. 131-137.

Elmasri, R.; Navate, S. B.: Fundamentals of Database Systems. 3. ed. Addison-Wesley, Redwood City 2000.

Embley, D. W.: Object Database Development. Concepts and Principles. Reading, MA. Addison-Wesley, 1998.

Golfarelli, M.; Rizzi, S: Designing the data warehouse: Key steps and crucial issues. Journal of Computer Science and Information Management, vol. 2, n. 3, 1999.

Golfarelli, M.; Rizzi, S.: A Methodological Framework for Data Warehouse Design. In Proceedings ACM First International Workshop on Data Warehousing and OLAP (DOLAP), Washington, 1998.

Golfarelli, M.; Maio, D.; Rizzi, S. (A): The Dimensional Fact Model: A Conceptual Model for Data Warehouses. International Journal of Cooperative Information Systems, vol. 7, n. 2&3, 1998.

Golfarelli, M.; Maio, D.; Rizzi, S. (B): Conceptual Design of Data Warehouses from E/R Schemes. Proceedings of the Hawaii International Conference On Systems Science, January 6-9, IEEE 1998.

Gupta, A.; Horowitz, E.: Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD. Englewood-Cliffs, NJ. Prentice-Hall 1991.

Gyssens, M.; Lakshmanan, L. V. S.: A foundation for multi-dimensional databases. Proc. 23th International Conference on Very Large Data Bases, San Francisco, CA, Morgan Kaufmann, 1997, pp. 106-115.

Hilbers, J.: Integration eines Schemagenerators in eine Data-Warehouse-Entwicklungsumgebung. Diplomarbeit Universität Münster 2000.

Holten, R.: The MetaMIS Approach for the Specification of Management Views on Business Processes. In: J. Becker; H. L. Grob; U. Müller-Funk; Klein, S.; Kuchen, H.; G. Vossen: Arbeitsberichte des Instituts für Wirtschaftsinformatik Nr. 84, Münster 2001.

Holten, R.: Framework and Method for Information Warehouse Development Processes. In: R. Jung, R. Winter (Hrsg.): Data Warehousing 2000 - Methoden, Anwendungen, Strategien. Heidelberg 2000, S. 135-163.

Holten, R.: Entwicklung von Führungsinformationssystemen. Ein methodenorientierter Ansatz. Wiesbaden 1999.

Hüsemann, B.; Lechtenbörger, J.; Vossen, G.: Conceptual Data Warehouse Design. Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000), Stockholm, Sweden, June 5-6, 2000, http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-28/

Inmon, W. H.: Building the Data Warehouse. 2nd ed., Wiley, New York et al. 1996.

Inmon, W. H.; Hackathorn, R. D.: Using the Data Warehouse. Wiley, New York et al. 1994.

Inmon, W. H.; Welch, J. D.; Glasey, K. L.: Managing the Data Warehouse. New York et al. 1997.

ISO: ISO/IEC 10027: Information Technology. Information Resource Dictionary Systems (IRDS)-Framework. ISO/IEC Intl. Standard, 1990.

Jacobs, S.; Holten, R.: Goal Driven Business Modeling. Supporting Decision Making within Information Systems Development. In: N. Comstock, C. Ellis, R. Kling, J. Mylopoulos, S. Kaplan (Hrsg.): Conference on Organizational Computing Systems, August 13-16, 1995, Milipatis, California, U.S.A. acm Press1995, S. 96-105.

Jarke, M.; Jeusfeld, M. A.; Quix, C.; Vassiliadis, P.: Architecture and Quality in Data Warehouses: An extended Repository Approach. Information Systems 24 (1999) 3, pp. 229 – 253.

Jarke, M.; Lenzerini, M.; Vassiliou Y.; Vassiliadis, P.: Fundamentals of Data Warehouses. Berlin et al. 2000.

Kaplan, R. S.; Norton, D. P.: Why Does Business Need a Balanced Scorecard? Journal of Cost Management, 11 (1997) 3, pp. 5-10.

Kaplan, R. S.; Norton, D. P.: Translating Strategy into Action. The Balanced Scorecard. Boston, Mass. 1996.

Kaplan, R. S.; Norton, D. P.: The Balanced Score Card. Measures that Drive Business Performance. Harvard Business Review, 70 (1992) January-February, pp. 71-79.

Lapsley, I.; Mitchel, F.: The Accounting Challenge: Accounting and Performance Measurement. London 1996.

Lechtenbörger, J.: Data Warehouse Schema Design. Akademische Verlagsgesellschaft Aka. Berlin 2001.

McMenamin, S. M.; Palmer, J. F.: Essential System Analysis. Yourdon Press, Prentice Hall. Englewood Cliffs, NY, 1984.

Moody, D. L.; Kortink, M. A. R.: From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000), Stockholm, Sweden, June 5-6, 2000, http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-28/.

Nagl, M.: Softwaretechnik. Methodisches Programmieren im Großen. Berlin, Heidelberg 1990.

Nissen, H. W.; Jeusfeld, M.; Jarke, M.; Zemanek, G. V.; Huber, H.: Managing Multiple Requirements Perspectives with Metamodels. IEEE Software, 13 (1996) 3, pp. 37-48.

olap Council: olap and olap Server Definitions. http://www.olapcouncil.org/research/glossaryly.htm 1997, downloaded 1998-06-03.

Pohl, K.: Process-Centered Requirements Engineering. Research Studies Press, Taunton, Somerset 1996.

Riebel, P.: Einzelerlös-, Einzelkosten- und Deckungsbeitragsrechung als Kern einer ganzheitlichen Führungsrechung. In: W. Männel (Hrsg.), Handbuch Kostenrechnung. Wiesbaden 1992, S. 247-299.

Riebel, P.: Gestaltungsprobleme einer zweckneutralen Grundrechnung. In: ZfbF 31 (1979), S. 863-893.

Sapia, C.; Blaschka, M.; Höfling, G.; Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. Proceedings of the International Workshop on Data Warehouse and Data Mining (DWDM 1998) Singapore, November 19-20 1998, Springer 1998, pp. 105-116.

Sapia, C.; Blaschka, M.; Höfling, G.: GraMMi: Using a Standard Repository Management System to Build a Generic Graphical Modeling Tool. Proceedings 33rd Hawaii International Conference on System Sciences (HICSS-33), Maui, Hawaii, USA 2000.

Scheer, A.-W.: ARIS – Business Process Frameworks. 3nd ed., Springer, Berlin 1999.

Scheer, A.-W.: ARIS – Business Process Modeling. 3nd ed., Springer, Berlin 2000.

Scheer, A.-W.: Business Process Engineering. 3rd ed., Springer, Berlin 1998.

Stöhr, T.; Müller, R.; Rahm, E.: An Integrative and Uniform Model for Metadata Management in Data Warehousing Environments. In.: S. Gatztiu, M. Jeusfeld, M. Staudt, Y-VAsiliou (Hrsg.): Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99). http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-19/, downloaded 1999-07-12.

Tryfona, N.; Busborg, F.; Borch Christiansen, J. G.: starER. A Conceptual Model for Data Warehouse Design. Proceedings of the DOLAP '99, ACM Second International Workshop on Data Warehousing and OLAP. November 6, 1999, Kansas City, Missouri, USA, Proceedings. ACM 1999, pp. 3-8.

Vassiliadis, P.: Gulliver in the land of data warehousing : practical experiences and observations of a researcher. Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000), Stockholm, Sweden, June 5-6, 2000, http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-28/.

Vassiliadis, P.; Bouzeghoub, M.; Quix, C.: Towards Quality-Oriented Data Warehouse Usage and Evolution. Information Systems 25 (2000) 2, pp. 89-115.

Vassiliadis, P.; Sellis, T.: A Survey of Logical Models for olap Databases. Sigmod Record 28 (4), S. 64-69, 1999.

Vossen, G.: Data Models, Database Languages and Database Management Systems. 4th ed. Oldenbourgh, München, Wien 2000 (in German).

Wedekind, H.: Datenbanksysteme I. Eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung. 2. Aufl., Mannheim et al. 1981.

Wedekind, H.: Objektorientierte Schemaentwicklung. Ein kategorialer Ansatz für Datenbanken und Programmierung. Mannheim et al. 1992.

Wedekind, H.; Albrecht, J.; Günzel, H.; Lehner, W.: Repositories for Data Warehouse Systems in a Middleware Environment. Fifth International Conference on Information Systems Analysis and Synthesis (ISAS'99, Orlando, Florida, July 31 – August 4 1999).

Weske, M.; Goesmann, T.; Holten, R.; Striemer, R.: A Reference Model for Workflow Application Development Processes. Proc. of International Joint Conference on Work Activities Coordination and Collaboration (WACC), San Francisco, CA 1999. ACM, pp. 1-10.

Widom, J.: Research problems in data warehousing. Proc. 4th International Conference on Information and Knowledge Management. New York: ACM 1995.

Appendix

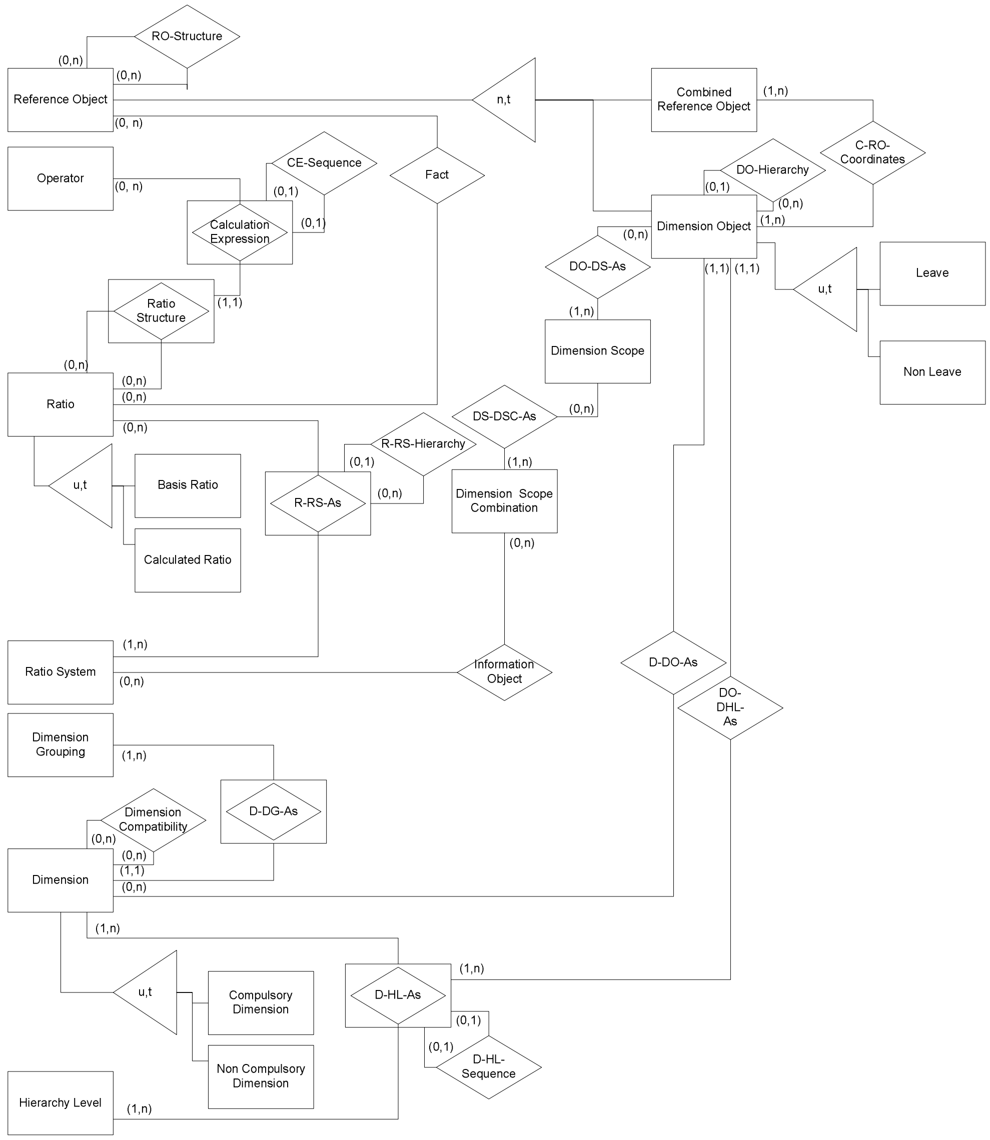


Figure 16: Meta model of language concepts for the specification of management views on business processes

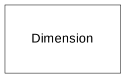
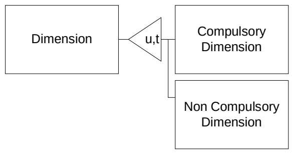
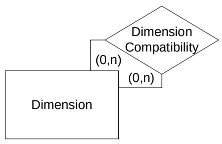
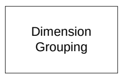
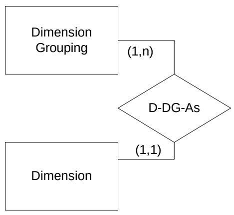
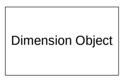
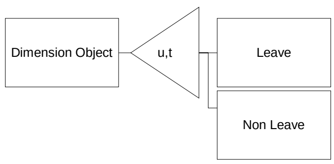
The language concepts are constructed based on a thorough analysis of management and accounting literature.[63] The following linguistic actions are used[64]:

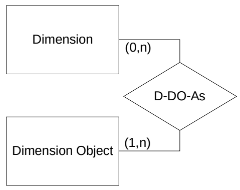
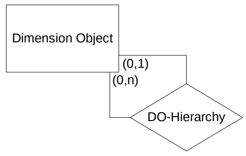
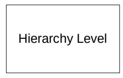
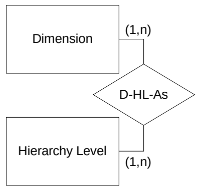
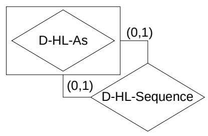
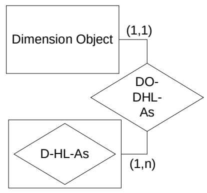
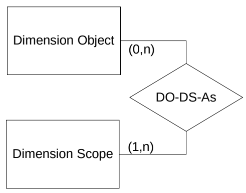
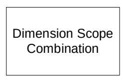
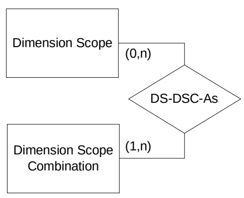
- Subsumption: A concept is created by statements. By means of subsumptions object types are created in the sense of an instance-of relation. An object type defines a set of objects. Concepts created by means of subsumptions are modelled with the entity type symbol.

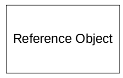
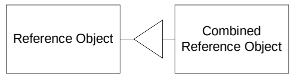
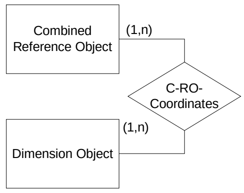
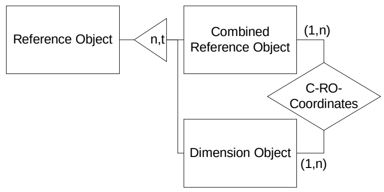
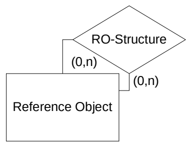
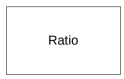
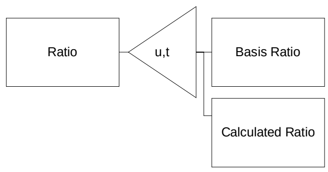
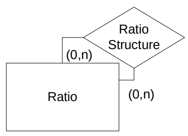
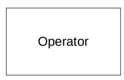
63 See Holten (1999) for a detailed discussion.
64 cf. Holten (1999); Wedekind (1981)

- Subordination: A set of concepts is subordinated to a higher concept by statements. By means of subordinations is-a relations are defined between object types. Is-a relations created by subordinations are modelled with a triangle.

- Composition: Two (or more) concepts are related by statements. By means of compositions relationship types are created. Concepts created by means of compositions are modelled with relationship type symbols and cardinalities in min-max notation. Cardinalities define the complexity of relationship types. For any concept used to define the meaning of the composition the complexity of minimum and maximum values of the respective elements are given as zero, one or many values. If composed concepts are required to compose further concepts this is modelled by surrounding the respective relationship type symbol by an entity type symbol.

Conceptual Language Aspect	**Linguistic action and statement**	**Meta Model Component** (cf. Figure 16)
Dimension	Subsumption: Used to create and organize the space the management's view is composed of.	
Compulsory Dimension, Non Compulsory Dimension	Subsumption and Subordination: Some dimensions like time and scenario are compulsory for any conceptual description of management views. Any other dimension is non compulsory. The specialization is unequivocal (symbol u) and total (symbol t).	
Dimension Compatibility	Composition: Recursive relationship of concept Dimension to itself. From the managements' point of view it may make sense to combine a dimension with none or many dimensions while defining management views on the business. The concept Dimension is used twice in this relationship (cardinalities (0,n) (0,n)).	
Dimension Grouping	Subsumption: A specific object type for which different dimensions can be used to characterize its aspects relevant for the management.	
D-DG-As (Dimension Dimension Grouping Assosiation)	Composition: Relationship between concepts Dimension and Dimension-Grouping. A certain dimension belongs to one unequivocal dimension grouping (cardinalities (1,1)). A certain dimension grouping comprises at least one dimension but may comprise many dimensions (cardinalities (1,n)).	
Dimension Object	Subsumption: Entities relevant for management's arrangements and examinations and part of the definition of dimensions in the sense that they have strong relationships to each other from the management's point of view.	
Leave, Non Leave	Subsumption and Subordination: The concept Dimension Object is unequivocally and totally (symbols u, t) specialised in the concepts Leave and Non Leave. Leaves are on the lowest level of the dimension hierarchies. Non Leaves are on all other levels. The set of leaves is the same for all dimensions belonging to the same dimension group-	

Conceptual Language Aspect	**Linguistic action and statement**	**Meta Model Component** (cf. Figure 16)
	ing.	
D-DO-As (Dimension Dimension Object Association)	Composition: Relationship between concepts Dimension and Dimension Object. A dimension requires a (possible empty) set of dimension objects for its definition (cardinalities (0,n)) and any dimension object requires a relationship to at least one dimension (cardinalities (1,n)). Leaves are related to all dimensions of a dimension grouping. All other dimension objects (non leaves) are related to exactly one dimension.	
DO-Hierarchy (Dimension Object Hierarchy)	Composition: Recursive relationship from concept Dimension Object to itself. For dimension objects a hierarchical order is required. Any dimension object may have zero or one higher dimension object (cardinalities (0,1)) and zero or many subordinated ones (cardinalities (0,n)).	
Hierarchy Level	Subsumption: Levels of hierarchy dimensions consist of and dimension objects are assigned to.	
D-HL-As (Dimension Hierarchy Level Association)	Composition: Relation between concepts Dimension and Hierarchy-Level. Any Dimension comprises one or many hierarchical levels (cardinalities (1,n)) and a hierarchical level as abstract object can be related to one or many dimensions (cardinalities (1,n)).	
D-HL-Sequence (Dimension Hierarchy Level Association Sequence)	Composition: There is a unequivocal order of he hierarchy levels associated to a dimension. Every hierarchical level of a dimension has zero or one predecessor and zero or one successor. (cardinalities (1,0) on either side).	
DO-DHL-As (Dimension Object Dimension Hierarchy Level Association Association)	Composition: Relationship between concepts Dimension-Object and D-HL-As. Every dimension object must unequivocally be associated to one hierarchical level of the dimension it belongs to (cardinalities (1,1)) and every hierarchical level of a dimension must contain at least one or many dimension objects (cardinalities (1,n)).	
Dimension Scope	Subsumption: Used to define scopes out of dimensions relevant for a management view.	
DO-DS-As (Dimension Object Dimension Scope Association)	Composition: Relationship between concepts Dimension-Object and Dimension-Scope. Any dimension object may or may not be member of a dimension scope (cardinalities (0,n)). Any dimension scope comprises one or more dimension objects (cardinalities (1,n)).	
Dimension-Scope-Combination	Subsumption: Used to identify combinations of dimension scopes while defining management views.	
DS-DSC-As (Dimension Scope Dimension Scope-Combination Association)	Composition: Relationship between concepts Dimension-Scope and Dimension-Scope-Combination. Any dimension scope combination may contain one or many dimension scopes (cardinalities (1,n)) whereas any dimension scope can be a member of zero or many dimension scope combinations (cardinalities (0,n)).	

Conceptual Language Aspect	**Linguistic action and statement**	**Meta Model Component** (cf. Figure 16)
Refrence Object	Subsumption: Reference objects are defined by Riebel as all “measures, processes and states of affairs which can be object to arrangements or examinations on their own”[65].	
Combined Reference Object	Subsumption and Subordination: A combined reference object is a reference object interpreted as a vector.	
C-RO-Coordinates (Combined Reference Object Coordinates)	Composition: Relationship between concepts Combined-Reference-Object and Dimension-Object. Dimension objects are used as coordinates to specify combined reference objects. Any dimension object can be used as a coordinate for one or many combined reference objects (cardinalities (1,n)) and any combined reference object has one or many coordinates (cardinalities (1,n)).	
Reference Object, Combined Reference-Object, Dimension-Object	Subordination: A reference object is a vector and then specialized as combined reference object. Additionally a reference object can have the role of an dimension object and then is used to define dimensions and as coordinates for combined reference objects. Nevertheless any dimension object is a reference object. The specialization of reference objects thus is not unequivocal (symbol n) but total (symbol t).	
RO-Structure (Reference Object Structure)	Composition: Recursive relationship from concept Reference-Object to itself. Logically this relationship defines the space of all reference objects management views can be composed of. Any reference object may have zero or many higher reference objects (cardinalities (0,n)) and zero or many subordinated ones (cardinalities (0,n)).	
Ratio	Subsumption: Ratios are the instruments to measure management relevant aspects of the value of an enterprise, the business performance and the financial situation.	
Basis Ratio, Calculated Ratio	Subsumption and Subordination: The concept Ratio is unequivocally and totally (symbols u and t) specialised in the concepts Basis Ratio and Calculated Ratio. Basis ratios are defined by means of statements. Calculated ratios are additionally defined by means of algebraic calculation expressions. Every ratio used to define a calculated ratio must be defined in advance.	
Ratio Structure	Composition: Recursive relationship from concept Ratio to itself. Any ratio can become part of an algebraic expression to calculate another ratio (cardinalities (0,n)) and any ratio can be explained algebraically based on a possible empty set of other ratios (cardinalities (0,n)).	
Operator	Subsumption: Operators are used in algebraic expressions to define ratios.	

65 Riebel (1979), p. 869

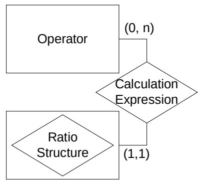
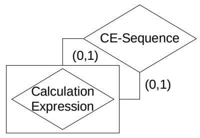
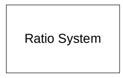
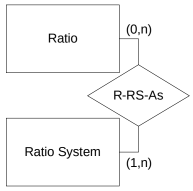
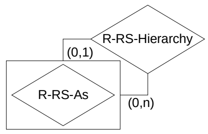
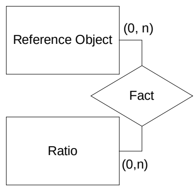
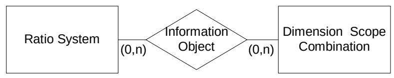
Conceptual Language Aspect	**Linguistic action and statement**	**Meta Model Component** (cf. Figure 16)
Calculation Expression	Composition: Relationship between concepts Operator and Ratio-Structure. Since ratio structures are parts of algebraic expressions there must be an unequivocal association of a given ratio structure to one operator (cardinalities (1,1)) whereas any operator can be used in zero or many calculation expressions (cardinalities (0,n)).	
CE-Sequence (Calculation Expression Sequence)	Composition: Recursive relationship from concept Calculation Expression to itself. To explain an algebraic expression an unequivocal sequence of calculation expressions is required. Any calculation expression must have zero or one predecessor and zero or one successor (cardinalities (0,1) on either side).	
Ratio System	Subsumption: A ratio system is a set of ratios which enables the analysis of different meaningful aspects of a business situation.	
R-RS-As (Ratio Ratio System Association)	Composition: Relationship between concepts Ratio and Ratio-System. A ratio system comprises one or many ratios (cardinalities (1,n)) and a ratio may be member of zero or many ratio systems (cardinalities (0,n)).	
R-RS-Hierarchy (Ratio Ratio System-Association Hierarchy)	Composition: Recursive relationship from concept R-RS-As to itself. Ratios which are part of a ratio system are organized hierarchically. Any ratio as member of a given ratio system may have zero or one higher ratio (cardinalities (0,1)) and zero or many subordinated ones (cardinalities (0,n)).	
Fact	Composition: Relationship between concepts Reference-Object and Ratio. Any reference object can be combined with zero or many ratios and vice versa (cardinalities (0,n) on either side).	
Information Object	Composition: Relationship between concepts Ratio System and Dimension Scope Combination. Set of facts relevant for a management user. One ratio system can be combined with none or many dimension scope combinations and vice versa (cardinalities (0,n) on either side).	

Table 16: Language concepts, linguistic actions and meta model components – conceptual modelling

The generated snowflake scheme is shown in Table 17. The tuples of the relations are not shown to keep the example simple. Additionally, identifiers with type information are given only if required to prevent misunderstandings. E.g., `Region-ID [Dimension xxx]` and `Date-ID [Dimension xxx]` indicate that the objects named "Region" and "Date" belong to the class "Dimension" and have an identifier "xxx" where as the objects `Region-ID [HierarchyLevel xxx]` and `Date-ID`

`[HierarchyLevel xxx]` are different objects (given the same name) with a different identifier and belong to the class “HierarchyLevel”.

Top level relations (Table 6)
LK-Store Assortment CCG (Store Assortment CCG-ID, Store Assortment CCG-Description)
LK-Quality / Price Level (Quality / Price Level-ID, Quality / Price Level-Description)
LK-Category Management (Category Management-ID, Category Management-Description)
LK-Colour (Colour-ID, Colour-Description)
LK-Date[Dimension xxx] (Date-ID [Dimension xxx], Date-Description [Dimension xxx])
LK-Day Of Week (Day Of Week-ID, Day Of Week-Description)
LK-Region [Dimension xxx] (Region-ID [Dimension xxx], Region-Description [Dimension xxx])
LK-Competition (Competition-ID, Competition-Description)
LK-Area / Location (Area / Location-ID, Area / Location-Description)
LK-Modernization (Modernization-ID, Modernization-Description)
Relations for top hierarchy levels (Table 7)
LK-Merchandise Class (Merchandise Class-ID, Merchandise Class-Description, Store Assortment CCG-ID)
LK-Article Price Level (Article Price Level-ID, Article Price Level-Description, Quality / Price Level-ID)
LK-Category Department (Category Department-ID, Category Department-Description, Category Management-ID)
LK-Article Colour (Article Colour-ID, Article Colour-Description, Colour-ID)
LK-Year (Year-ID, Year-Description, Date-ID [Dimension xxx])
LK-Weekday (Weekday-ID, Weekday-Description, Day Of Week-ID)
LK-Region [HierarchyLevel xxx] (Region-ID [HierarchyLevel xxx], Region-Description [HierarchyLevel xxx], Region-ID [Dimension xxx])
LK-Competitive Class (Competitive Class-ID, Competitive Class-Description, Competition-ID)
LK-Area (Area-ID, Area-Description, Area / Location-ID)
LK-Modernization-Class (Modernization-Class-ID, Modernization-Class-Description, Modernization-ID)

Relations for hierarchy levels with superior levels – without bottom level (Table 8)
LK-4-digit merchandise category (4-digit merchandise category-ID, 4-digit merchandise category-Description, 3-digit merchandise category-ID)
LK-3-digit merchandise category (3-digit merchandise category-ID, 3-digit merchandise category-Description, 2-digit merchandise category-ID)
LK-2-digit merchandise category (2-digit merchandise category-ID, 2-digit merchandise category-Description, Merchandise Department-ID)
LK-Merchandise Department (Merchandise Department-ID, Merchandise Department-Description, Merchandise Class-ID)
LK-Sub Category (Sub Category-ID, Sub Category-Description, Category-ID)
LK-Category (Category-ID, Category-Description, Category Department-ID)
LK-Month (Month-ID, Month-Description, Quarter-ID)
LK-Quarter (Quarter-ID, Quarter-Description, Year-ID)
LK-Town (Town-ID, Town-Description, Country-ID)
LK-Country (Country-ID, Country-Description, Region-ID [HierarchyLevel xxx])
Relations for bottom hierarchy levels (Table 8)
LK-Article (Article-ID, Article-Description, 4-digit merchandise category-ID, Article Price Level-ID, Sub Category-ID, Article Colour-ID)
LK-Date [HierarchyLevel xxx] (Date-ID [HierarchyLevel xxx], Date-Description [HierarchyLevel xxx], Month-ID, Weekday-ID)
LK-Store (Store-ID, Store-Description, Town-ID, Competitive Class-ID, Area-ID, Modernization-Class-ID)
Fact relation (Table 5)
Facts-DataMart “EXCOM” (Article-ID, Date-ID [HierarchyLevel xxx], Store-ID, average annual sales, average daily sales, average annual inventory, …, area intensity, area productivity, …)

Table 17: Generated snowflake scheme for example case (section 4, section 5.1)

RelationType	
Relation	*DW-Role*
Facts-DataMart “Excom”	FactTable
LK-Store Assortment CCG	LookUpTable
LK-Merchandise Class	LookUpTable
LK-Merchandise Department	LookUpTable
LK-2-digit merchandise category	LookUpTable
LK-3-digit merchandise category	LookUpTable
LK-4-digit merchandise category	LookUpTable

LK-Article	LookUpTable
LK-Month	LookUpTable
LK-Date [HierarchyLevel xxx]	LookUpTable
LK-Region [HierarchyLevel xxx]	LookUpTable
LK-Country	LookUpTable
LK-Town	LookUpTable
LK-Store	LookUpTable

ForeignKeyAssociations		
Attribute	*Relation*	*KeyType*
Article-ID	Facts-DataMart "Excom"	Foreign
Date-ID [HierarchyLevel xxx]	Facts-DataMart "Excom"	Foreign
Store-ID	Facts-DataMart "Excom"	Foreign
Store Assortment CCG-ID	LK-Store Assortment CCG	Primary
Merchandise Class-ID	LK-Merchandise Class	Primary
Store Assortment CCG-ID	LK-Merchandise Class	Foreign
Merchandise Department-ID	LK-Merchandise Department	Primary
Merchandise Class-ID	LK-Merchandise Department	Foreign
2-digit merchandise category-ID	LK-2-digit merchandise category	Primary
Merchandise Department-ID	LK-2-digit merchandise category	Foreign
3-digit merchandise category-ID	LK-3-digit merchandise category	Primary
2-digit merchandise category-ID	LK-3-digit merchandise category	Foreign
4-digit merchandise category-ID	LK-4-digit merchandise category	Primary
3-digit merchandise category-ID	LK-4-digit merchandise category	Foreign
Article-ID	LK-Article	Primary
4-digit merchandise category-ID	LK-Article	Foreign
Article Price Level-ID	LK-Article	Foreign
Sub Category-ID	LK-Article	Foreign
Article Colour-ID	LK-Article	Foreign
Month-ID	LK-Month	Primary
Quarter-ID	LK-Month	Foreign
Date-ID [HierarchyLevel xxx]	LK-Date [HierarchyLevel xxx]	Primary
Month-ID	LK-Date [HierarchyLevel xxx]	Foreign
Weekday-ID	LK-Date [HierarchyLevel xxx]	Foreign
Region-ID [HierarchyLevel xxx]	LK-Region [HierarchyLevel xxx]	Primary
Region-ID [Dimension xxx]	LK-Region [HierarchyLevel xxx]	Foreign
Country-ID	LK-Country	Primary
Region-ID [HierarchyLevel xxx]	LK-Country	Foreign
Town-ID	LK-Town	Primary
Country-ID	LK-Town	Foreign
Store-ID	LK-Store	Primary
Town-ID	LK-Store	Foreign
Competitive Class-ID	LK-Store	Foreign
Area-ID	LK-Store	Foreign
Modernization-Class-ID	LK-Store	Foreign

Table 18: Data dictionary data (exceprt)

Arbeitsberichte des Instituts für Wirtschaftsinformatik

Nr. 1 Bolte, Ch.; Kurbel, K.; Moazzami, M.; Pietsch, W.: Erfahrungen bei der Entwicklung eines Informationssystems auf RDBMS- und 4GL-Basis; Februar 1991.

Nr. 2 Kurbel, K.: Das technologische Umfeld der Informationsverarbeitung - ein subjektiver 'State of the Art'-Report über Hardware, Software und Paradigmen; März 1991.

Nr. 3 Kurbel, K.: CA-Techniken und CIM; Mai 1991.

Nr. 4 Nietsch, M.; Nietsch, T.; Rautenstrauch, C.; Rinschede, M.; Siedentopf, J.: Anforderungen mittelständischer Industriebetriebe an einen elektronischen Leitstand - Ergebnisse einer Untersuchung bei zwölf Unternehmen; Juli 1991.

Nr. 5 Becker, J.; Prischmann, M.: Konnektionistische Modelle - Grundlagen und Konzepte; September 1991.

Nr. 6 Grob, H.L.: Ein produktivitätsorientierter Ansatz zur Evaluierung von Beratungserfolgen; September 1991.

Nr. 7 Becker, J.: CIM und Logistik; Oktober 1991.

Nr. 8 Burgholz, M.; Kurbel, K.; Nietsch, Th.; Rautenstrauch, C.: Erfahrungen bei der Entwicklung und Portierung eines elektronischen Leitstands; Januar 1992.

Nr. 9 Becker, J.; Prischmann, M.: Anwendung konnektionistischer Systeme; Februar 1992.

Nr. 10 Becker, J.: Computer Integrated Manufacturing aus Sicht der Betriebswirtschaftslehre und der Wirtschaftsinformatik; April 1992.

Nr. 11 Kurbel, K.; Dornhoff, P.: A System for Case-Based Effort Estimation for Software-Development Projects; Juli 1992.

Nr. 12 Dornhoff, P.: Aufwandsplanung zur Unterstützung des Managements von Softwareentwicklungsprojekten; August 1992.

Nr. 13 Eicker, S.; Schnieder, T.: Reengineering; August 1992.

Nr. 14 Erkelenz, F.: KVD2 - Ein integriertes wissensbasiertes Modul zur Bemessung von Krankenhausverweildauern - Problemstellung, Konzeption und Realisierung; Dezember 1992.

Nr. 15 Horster, B.; Schneider, B.; Siedentopf, J.: Kriterien zur Auswahl konnektionistischer Verfahren für betriebliche Probleme; März 1993.

Nr. 16 Jung, R.: Wirtschaftlichkeitsfaktoren beim integrationsorientierten Reengineering: Verteilungsarchitektur und Integrationsschritte aus ökonomischer Sicht; Juli 1993.

Nr. 17 Miller, C.; Weiland, R.: Der Übergang von proprietären zu offenen Systemen aus Sicht der Transaktionskostentheorie; Juli 1993.

Nr. 18 Becker, J., Rosemann, M.: Design for Logistics - Ein Beispiel für die logistikgerechte Gestaltung des Computer Integrated Manufacturing; Juli 1993.

Nr. 19 Becker, J.; Rosemann, M.: Informationswirtschaftliche Integrationsschwerpunkte innerhalb der logistischen Subsysteme - Ein Beitrag zu einem produktionsübergreifenden Verständnis von CIM; Juli 1993.

Nr. 20 Becker, J.: Neue Verfahren der entwurfs- und konstruktionsbegleitenden Kalkulation und ihre Grenzen in der praktischen Anwendung; Juli 1993.

Nr. 21 Becker, K.; Prischmann, M.: VESKONN - Prototypische Umsetzung eines modularen Konzepts zur Konstruktionsunterstützung mit konnektionistischen Methoden; November 1993

Nr. 22 Schneider, B.: Neuronale Netze für betriebliche Anwendungen: Anwendungspotentiale und existierende Systeme; November 1993.

Nr. 23 Nietsch, T.; Rautenstrauch, C.; Rehfeldt, M.; Rosemann, M.; Turowski, K.: Ansätze für die Verbesserung von PPS-Systemen durch Fuzzy-Logik; Dezember 1993.

Nr. 24 Nietsch, M.; Rinschede, M.; Rautenstrauch, C.: Werkzeuggestützte Individualisierung des objektorientierten Leitstands ooL; Dezember 1993.

Nr. 25 Meckenstock, A.; Unland, R.; Zimmer, D.: Flexible Unterstützung kooperativer Entwurfsumgebungen durch einen Transaktions-Baukasten; Dezember 1993.

Nr. 26 Grob, H. L.: Computer Assisted Learning (CAL) durch Berechnungsexperimente; Januar 1994.

Nr. 27 Kirn, St.; Unland, R. (Hrsg.): Tagungsband zum Workshop "Unterstützung Organisatorischer Prozesse durch CSCW". In Kooperation mit GI-Fachausschuß 5.5 "Betriebliche Kommunikations- und Informationssysteme" und Arbeitskreis 5.5.1 "Computer Supported Cooperative Work", Westfälische Wilhelms-Universität Münster, 4.-5. November 1993; November 1993.

Nr. 28 Kirn, St.; Unland, R.: Zur Verbundintelligenz integrierter Mensch-Computer-Teams: Ein organisationstheoretischer Ansatz; März 1994.

Nr. 29 Kirn, St.; Unland, R.: Workflow Management mit kooperativen Softwaresystemen: State of the Art und Problemabriß; März 1994.

Nr. 30 Unland, R.: Optimistic Concurrency Control Revisited; März 1994.

Nr. 31 Unland, R.: Semantics-Based Locking: From Isolation to Cooperation; März 1994.

Nr. 32 Meckenstock, A.; Unland, R.; Zimmer, D.: Controlling Cooperation and Recovery in Nested Transactions; März 1994.

Nr. 33 Kurbel, K.; Schnieder, T.: Integration Issues of Information Engineering Based I-CASE Tools; September 1994.

Nr. 34 Unland, R.: TOPAZ: A Tool Kit for the Construction of Application Specific Transaction; November 1994.

Nr. 35 Unland, R.: Organizational Intelligence and Negotiation Based DAI Systems - Theoretical Foundations and Experimental Results; November 1994.

Nr. 36 Unland, R.; Kirn, St.; Wanka, U.; O'Hare, G.M.P.; Abbas, S.: AEGIS: AGENT ORIENTED ORGANISATIONS; Februar 1995.

Nr. 37 Jung, R.; Rimpler, A.; Schnieder, T.; Teubner, A.: Eine empirische Untersuchung von Kosteneinflußfaktoren bei integrationsorientierten Reengineering-Projekten; März 1995.

Nr. 38 Kirn, St.: Organisatorische Flexibilität durch Workflow-Management-Systeme?; Juli 1995.

Nr. 39 Kirn, St.: Cooperative Knowledge Processing: The Key Technology for Future Organizations; Juli 1995.

Nr. 40 Kirn, St.: Organisational Intelligence and Distributed AI; Juli 1995.

Nr. 41 Fischer, K.; Kirn, St.; Weinhard, Ch. (Hrsg.): Organisationsaspekte in Multiagentensystemen; September 1995.

Nr. 42 Grob, H. L.; Lange, W.: Zum Wandel des Berufsbildes bei Wirtschaftsinformatikern, Eine empirische Analyse auf der Basis von Stellenanzeigen; Oktober 1995.

Nr. 43 Abu-Alwan, I.; Schlagheck, B.; Unland, R.: Evaluierung des objektorientierten Datenbankmanagementsystems ObjectStore; Dezember 1995.

Nr. 44 Winter, R.: Using Formalized Invariant Properties of an Extended Conceptual Model to Generate Reusable Consistency Control for Information Systems; Dezember 1995.

Nr. 45 Winter, R.: Design and Implementation of Derivation Rules in Information Systems; Februar 1996.

Nr. 46 Becker, J.: Eine Architektur für Handelsinformationssysteme; März 1996.

Nr. 47 Becker, J.; Rosemann, M. (Hrsg.): Workflowmanagement - State-of-the-Art aus Sicht von Theorie und Praxis, Proceedings zum Workshop vom 10. April 1996; April 1996.

Nr. 48 Rosemann, M.; zur Mühlen, M.: Der Lösungsbeitrag von Metadatenmodellen beim Vergleich von Workflowmanagementsystemen; Juni 1996.

Nr. 49 Rosemann, M.; Denecke, Th.; Püttmann, M.: Konzeption und prototypische Realisierung eines Informationssystems für das Prozeßmonitoring und -controlling; September 1996.

Nr. 50 v. Uthmann, C.; Turowski, K. unter Mitarbeit von Rehfeldt, M.; Skall, M.: Workflow-basierte Geschäftsprozeßregelung als Konzept für das Management von Produktentwicklungs-prozessen; November 1996.

Nr. 51 Eicker, S.; Jung, R.; Nietsch, M.; Winter, R.: Entwicklung eines Data Warehouse für das Produktionscontrolling: Konzepte und Erfahrungen; November 1996.

Nr. 52 Becker, J.; Rosemann, M.; Schütte, R. (Hrsg.): Entwicklungsstand und Entwicklungsperspektiven der Referenzmodellierung, Proceedings zur Veranstaltung vom 10. März 1997; März 1997.

Nr. 53 Loos, P.: Capture More Data Semantic Through The Expanded Entity-Relationship Model (PERM); Februar 1997.

Nr. 54 Becker, J.; Rosemann, M. (Hrsg.): Organisatorische und technische Aspekte beim Einsatz von Workflowmanagementsystemen. Proceedings zur Veranstaltung vom 10. April 1997; April 1997.

Nr. 55 Holten, R.; Knackstedt, R.: Führungsinformationssysteme - historische Entwicklung und Konzeption; April 1997.

Nr. 56 Holten, R.: Die drei Dimensionen des Inhaltsaspektes von Führungsinformationssystemen; April 1997.

Nr. 57 Holten, R.; Striemer, R.; Weske, M.: Ansätze zur Entwicklung von Workflow-basierten Anwendungssystemen - eine vergleichende Darstellung; April 1997.

Nr. 58 Kuchen, H.: Arbeitstagung Programmiersprachen, Tagungsband; Juli 1997.

Nr. 59 Vering, O.: Berücksichtigung von Unschärfe in betrieblichen Informationssystemen - Einsatzfelder und Nutzenpotentiale am Beispiel der PPS; September 1997.

Nr. 60 Schwegmann, A.; Schlagheck, B.: Integration der Prozeßorientierung in das objektorientierte Paradigma: Klassenzuordnungsansatz vs. Prozessklassenansatz; Dezember 1997.

Nr. 61 Speck, M.: In Vorbereitung.

Nr. 62 Wiese, J.: Ein Entscheidungsmodell für die Auswahl von Standardanwendungssoftware am Beispiel von Warenwirtschaftssystemen; März 1998.

Nr. 63 Kuchen, H.: Workshop on Functional and Logic Programming, Proceedings; Juni 1998.

Nr. 64 Uthmann, C. v.; Becker, J.; Brödner, P.; Maucher, I.; Rosemann, M.: PPS meets Workflow. Proceedings zum Workshop vom 9. Juni 1998; Juni 1998.

Nr. 65 Scheer, A.-W.; Rosemann, M.; Schütte, R. (Hrsg.): Integrationsmanagement; Januar 1999.

Nr. 66 zur Mühlen, M.: Internet - Technologie und Historie; Juni 1999.

Nr. 67 Holten R.: A Framework for Information Warehouse Development Processes; Mai 1999.

Nr. 68 Holten R.; Knackstedt, R.: Fachkonzeption von Führungsinformationssystemen - Instanziierung eines FIS-Metamodells am Beispiel eines Einzelhandelsunternehmens; Mai 1999.

Nr. 69 Holten, R.: Semantische Spezifikation Dispositiver Informationssysteme; Juli 1999.

Nr. 70 Becker, J.: Workflow Management Conference.Proceedings of the 1999 Workflow Management Conference on Workflow based Applications; 1999.

Nr. 71 Klein, S.; Schneider, B.; Vossen, G.; Weske, M.; Projektgruppe PESS: Eine XML-basierte Systemarchitektur zur Realisierung flexibler Web-Applikationen; Juli 2000.

Nr. 72 Klein, S.; Schneider; B. (Hrsg): Negotiations and Interactions in Electronic Markets, Proceedings of the Sixth Research Symposium on Emerging Electronic Markets, Muenster, Germany, September 19 - 21, 1999; August 2000.

Nr. 73 Becker, J.; Bergerfurth, J.; Hansmann, H.; Neumann, S.; Serries, T.: Methoden zur Einführung Workflow-gestützter Architekturen von PPS-Systemen; November 2000.

Nr. 74 Terveer, I.: Die asymptotische Verteilung der Spannweite bei Zufallsgrößen mit paarweise identischer Korrelation; Februar 2002.

Nr. 75 Becker, J. (Ed.): Research Reports, Proceedings of the University Alliance Executive Directors Workshop – ECIS 2001; Juni 2001.

Nr. 76, Klein, u.a. (Eds.): MOVE: Eine flexible Architektur zur Unterstützung des Außendienstes mit mobile devices. (In Vorbereitung)

Nr. 77 Knackstedt, R.; Holten, R.; Hansmann, H.; Neumann, St.: Konstruktion von Methodiken: Vorschläge für eine begriffliche Grundlegung und domänenspezifische Anwendungsbeispiele; Juli 2001.

Nr. 78 Holten, R.: Konstruktion domänenspezifischer Modellierungstechniken für die Modellierung von Fachkonzepten; August 2001.

Nr. 79 Vossen, G., Hüsemann, B., Lechtenbörger, J.: XLX – Eine Lernplattform für den universitären Übungsbetrieb; August 2001.

Nr. 80 Knackstedt, R., Serries, Th.: Gestaltung von Führungsinformationssystemen mittels Informationsportalen; Ansätze zur Integration von Data-Warehouse- und Content-Management-Systemen; November 2001

Nr. 81 Holten, R.: Conceptual Models as Basis for the Integrated Information Warehouse Development; Oktober 2001.

Nr. 82 Teubner, A.: Informationsmanagement: Historie, disziplinärer Kontext und Stand der Wissenschaft; Februar 2002.

Nr. 83 Vossen, G.: Vernetzte Hausinformationssysteme – Stand und Perspektive.

Nr. 84 Holten, R.: The MetaMIS Approach for the Specification of Management Views on Business Processes; November 2001.

Nr. 85 Becker, J.; Neumann, S.; Hansmann, H. : Workflow-integrierte Produktionsplanung und -steuerung: Ein Architekturmodell für die Koordination von Prozessen der industriellen Auftragsabwicklung Januar 2002.

Nr. 86 Teubner, R.A.; Klein, S.: Bestandsaufnahme aktueller deutschsprachiger Lehrbücher zum Informationsmanagement, April 2002.

Nr. 87 Holten, R.: Specification of Management Views in Information Warehouse Projects, April 2002.