

Holten, Roland

Working Paper

Conceptual models as basis for integrated information warehouse development

Arbeitsberichte des Instituts für Wirtschaftsinformatik, No. 81

Provided in Cooperation with:

University of Münster, Department of Information Systems

Suggested Citation: Holten, Roland (2001) : Conceptual models as basis for integrated information warehouse development, Arbeitsberichte des Instituts für Wirtschaftsinformatik, No. 81, Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik, Münster

This Version is available at:

<https://hdl.handle.net/10419/59356>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

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. 81

**Conceptual Models as Basis for Integrated
Information Warehouse Development**

Roland Holten

Contents

Contents	II
Figures	III
Tables	III
Summary.....	III
1 Introduction.....	1
2 Information Warehouse Development Framework	4
2.1 Organizing Model and Abstraction Levels	4
2.2 Organizing Development Phases	7
2.3 Combining the Dimensions: IWH Development Framework	9
3 Conceptual Specification of Management’s Information Requirements: Meta Information Creation	11
3.1 Scientific Background and Methodology	11
Background	11
Methodology	13
3.2 Language and Meta Model for IWH Concepts.....	14
3.3 Representation and Technique for IWH Concepts	18
3.4 Method for IWH Concepts: Creating IWH Meta Information.....	20
4 Horizontal Integration of Information Warehouse Development: Meta Information Use 23	
4.1 Enlargement of Language and Meta Model	23
4.2 Meta Information Repository and Development Tool.....	25
4.3 Using Meta Information	27
4.3.1 Data Warehouse Parameters.....	28
Definition of fact table attributes	29
Definition of the bottom level for every dimension.....	30
Definition of hierarchical levels and foreign key relations within the dimensions	31
4.3.2 Etl and Olap Parameters	32
5 Related Work and Discussion.....	38
6 Conclusion.....	40
References	41
Appendix	45

Figures

Figure 1: Model and Abstraction Levels	5
Figure 2: Information Warehouse Development Framework	10
Figure 3: Meta model of language for information warehouse concepts	18
Figure 4: Conceptual Information Warehouse Model – Dimension Objects.....	21
Figure 5: Conceptual Information Warehouse Model – Ratio Systems	22
Figure 6: Enlargement of modeling language	25
Figure 7: Example for relational schemes of repository.....	26
Figure 8: Data Warehouse Scheme	29
Figure 9: Information Warehouse Parameters	32
Figure 10: Information Warehouse Load	34
Figure 11: Information Warehouse Data	34
Figure 12: Information Warehouse Usage	35
Figure 13: Desired Report Structure	35
Figure 14: Report Results	37

Tables

Table 1: Language concepts, linguistic actions and meta model components – conceptual modelling.....	49
Table 2: Representation Formalism or Information Warehouse Concepts	50
Table 3: Language concepts, linguistic actions and meta model components – technical modelling	52

Summary

Research in the field of information warehousing mostly focuses technical aspects. Only recently some contributions are found dealing with methodical aspects of information warehouse development processes. With respect to the central role information warehouses play for the management a development method is presented which strictly concentrates on management views. Language concepts are developed which allow the specification of information warehouses out of the managements perspective and a representation formalism supporting this language is presented. Methodically the language construction is based on constructive philosophy. Conceptual models are used as meta information in later development phases and it is shown how meta data of etl and olap tools available on the market can be generated out of the conceptual models. The approach presented has been verified by means of a prototype.

1 Introduction

A data warehouse is generally accepted as central component of an information systems architecture storing materialized views on business processes in support of management's information requirements¹. It is located on a central layer of an idealized layer oriented architecture connecting online transaction processing (oltp) systems to 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 they are connected to the data warehouse by means of 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 today⁴. Contributions⁵ are reaching from technical discussions of data bases and algorithms enabling olap functionality⁶ information search behavior of managers⁷ and papers concentrating on methodologies of information systems development⁸. Research on development methodologies itself is a complex field since investigations refer to logically separate levels of abstraction. On the one hand there are processes developing methods.⁹ On the other hand information systems development processes 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 VASSILIADIS¹¹ the lack of an accepted methodology for data warehouse development is one central factor affecting failure of data warehouse projects. This paper deals with the development of conceptual models and their integration in the development process and thus tries to contribute to one of the many open research fields.¹²

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 cf. Vassiliadis (2000) for an overview of research topics

⁶ cf. Agarwal et al. (1996); Agarwal 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)

⁷ cf. Borgman (1998)

⁸ cf. Golfarelli et al. (Hawaii) (1998); 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)

¹² cf. Vassiliadis (2000)

In recent methodologically oriented contributions¹³ 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 Meta Data Framework¹⁴. 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 perspectives 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 directly 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.

Having understood the central role of the enterprise model for data warehouse development from a methodological point of view the main question is, which concepts are required to define the enterprise model, especially the conceptual perspective, in concrete development projects. This is regarded as the central obstacle to the methodical development of information warehouses and is the main issue of this work. The term information warehouse (instead of data warehouse) is used for the time being since the information system under consideration has to match management's information requirements in the first place. In accordance with the findings of JARKE ET AL.¹⁵ this matching should be investigated independently of any technological realization which is the essence of the methodical approach presented in this work.

Concerning the development of information warehouses *two main problems* each comprising a set of questions are addressed in this paper. The first problem refers to the conceptual specification of management's information requirements. The questions are:

1. Which concepts must be provided by a language in order to model management's information requirements independent of any technical aspect?
2. What is a suitable form of representation matching this language and being appropriate to support discussions with the management in order to model their information requirements adequately?

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

¹⁴ cf. Jarke et al. (1999); Jarke et al. (2000)

¹⁵ cf. Jarke et al. (1999); Jarke et al. (2000)

The second problem deals with the integration of this specification with technical meta data of information warehouse components. To support respective development decisions three additional questions are to answer:

3. How data warehouse schemes can be created based on conceptual specifications of management's information requirements and what parts of this task can be automated?
4. How meta data describing the business oriented logic of data warehouse data can be generated?
5. Which meta data required for olap functionality and the etl processes can be generated?

This paper presents a framework and a method intended to organize information warehouse development decisions based on a strict separation of management and technical views on a central enterprise model. The main contributions are the construction of a domain specific language providing concepts for the specification of management's information requirements and completing this language with a suitable representation formalism. The enterprise model is integrated with other meta data relevant to information warehouse development and usage by means of a central repository. A prototypical implementation of a tool supporting the development method proposed is discussed briefly and it is shown how the approach presented can be integrated with information warehouse components available on the market.

Discussions with tool vendors and project managers have shown that two main advantages can be expected if information warehouse development processes are strictly based on conceptual models specifying information requirements of the management: First of all the time necessary for development and maintenance of information warehouses can be reduced. Secondly the information warehouse implemented will match management's information requirements in a better way.

In section 2 a development framework for information warehouse projects is introduced. Section 3 develops a language and a corresponding representation formalism for the specification of management's information requirements as part of the conceptual enterprise model. The integration of conceptual models with technical meta data is discussed in section 4. The findings are confronted with related work in section 5 and it is shown how they can be integrated with other findings relevant to the topic. Section 6 provides summary and conclusions.

2 Information Warehouse Development Framework

The framework presented in this section is intended to organize information warehouse development tasks and relevant method development tasks as well. It enables the discussion of an information warehouse development approach based upon a differentiation of two orthogonal dimensions. The first dimension of the framework deals with different levels of abstraction required to understand the modeling process. The second dimension is characterized by the distinction of development phases arranged in a logical order.

2.1 Organizing Model and Abstraction Levels

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.

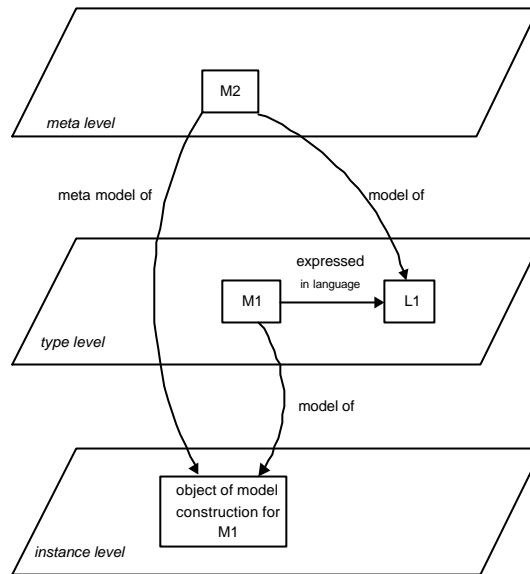
Different levels of abstraction can be derived from science theory¹⁶ and are well established in software engineering.¹⁷ Within the philosophical discipline of logic languages on different levels of abstraction are distinguished. A language which is subject to scientific analysis is called object language whereas the language used to carry out the analysis is called meta language¹⁸. The attribute “meta” thus describes the role a certain language fulfils in a scientific process. This understanding of language levels is applied to the process of modeling in the following way¹⁹: Languages, especially formalized languages, are used to create models. A certain language L1 used to create a model M1 can itself be expressed by a model M2. Following the intention of logic it can be said that M2 is a meta model with respect to the instance represented by model M1 (Figure 1).

¹⁶ cf. Holten (1999), pp. 11-17

¹⁷ cf. Pohl (1996); ISO (1990); Jacobs, Holten (1995)

¹⁸ cf. Kambartel (1996); Lorenz (A) (1995); Lorenz (B) (1995)

¹⁹ cf. Holten (1999), pp. 11



Cf. Nissen et al. (1996); Holten (1999)

Figure 1: Model and Abstraction Levels

Different levels of abstraction are discussed intensively for the purpose of information systems development in the literature. The IRDS framework²⁰ is characterized by four levels of abstraction. The lower three directly correspond to the levels discussed so far. 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 modeling”²¹. 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.

Compared to the repository structure of the DWQ approach²² the abstraction levels discussed here are shifted downwards. The DWQ approach intends to describe the method development processes using concepts provided by a so called process meta model. This model is located on a meta meta level with respect to the discussions made above. DWQ looks at development processes defined in so called process models which belong to the meta level defined in this paper. The lowest level of abstraction within DWQ’s repository structure contains business processes which are located on the type level in Figure 1.

²⁰ cf. ISO (1990); Pohl (1996); Jacobs, Holten (1995)

²¹ cf. Jacobs, Holten (1995)

²² cf. Jarke et al. (1999); Jarke et al. (2000)

The abstraction levels discussed in this paper are intended to describe information warehouse development processes and information usage processes in the enterprise. In contrast to the DWQ and the IRDS approaches it is not intended to look at method development processes in the first way. The levels relevant for this paper directly correspond to the abstraction levels discussed within the framework of the Architecture of Integrated Information Systems (ARIS)²³. Nevertheless the so called meta² level of the ARIS framework (corresponding to the meta model level of the DWQ approach and the IRD definition schema level of the IRDS approach) will not be discussed in this paper.

For demonstration purposes the following business example is used:

An example company (EXCOM) produces and sells cars and other goods and stores the following sales transaction in its oltp system: One sports car (product 4711) sold to client Westfalia Cars on November 8th 1999 for 28.000 EUROS. Additionally, EXCOM's management requires information concerning sales of cars in Europe within the year 1999. EXCOM's Information Warehouse calculates this excerpt of a sales report to 30 Billion EUROS. This can be assigned to three abstraction levels of interest here in the following way (see Figure 1):

1. The *instance level* comprises concrete data relevant for the domain, i.e. *sales of cars in region Europe in Year 1999 realized with reselling partners which amount to 30 Billion EUROS*. Documents on this level comprise the bookings made in the business process.
2. The next level is called *type level* and defines the type of instance level data in the sense of a database scheme or variables. In the example there must be a variable able to store a certain amount of EUROS for sales of cars in region Europe in Year 1999 realized with reselling partners.
3. The concepts and terms used to define the models on the type level can be modelled as well which is necessary e.g. for the development of repository based CASE-tools. This modelling task leads to documents and models on the next level of abstraction called *meta level*. Concerning the example the meta level model must show how a certain object related to business decisions can be combined with ratios used in accounting theory (like sales and cost).

²³ cf. Scheer (1999)

2.2 Organizing Development Phases

The data warehouse meta data framework and the respective repository meta model²⁴ are both characterized by the three perspectives “conceptual perspective”, “logical perspective” and “physical perspective”. Any perspective is defined by means of instances of the repository meta object class. The conceptual perspective is characterized by *models* which are composed of so called *concepts*. *Model* and *concept* are terms of the repository meta model. The logical perspective is characterized by *schemas* and *types* which both are meta terms. Finally the physical perspective is characterized by meta terms *agent* and *data store* which belong to the meta class *dw component*. The DWQ repository meta model shows how the meta concepts of the three perspectives are related to each other and gives clear definitions of the respective terms. The DWQ approach concentrates on the definition of the views characterizing the three perspectives on an abstract level. It provides the terms and concepts to describe the documents and models required in different phases of data warehouse development based on the assumption that these documents have to be integrated in one central repository. The DWQ approach implicitly looks at the development phases of data warehouse systems.

Another well known approach structuring the development of information systems is the Architecture of Integrated Information Systems (ARIS)²⁵. 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, which are characterized by 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*. The three levels of the data view within the ARIS framework can be compared to the three perspectives of the DWQ framework mentioned above. ARIS as well intends to provide conceptual views on the requirements level, schema views on the design level and physical views on the implementation level.

To organize the development process of information warehouses the frameworks discussed so far provide relevant information. The assumption that there exists some opaque information system (referred to as information warehouse) matching management’s information requirements has to be verified by stepwise transforming these requirements into a logical information systems architecture in the sense of the logical perspective of the DWQ framework²⁶ and the design specification in the sense of the

²⁴ cf. Jarke et al. (1999); Jarke et al. (2000)

²⁵ cf. Scheer (1999); Scheer (2000); Scheer (1998)

²⁶ cf. Jarke et al. (1999); Jarke et al. (2000)

ARIS framework²⁷. This information systems architecture additionally has to match technical and organizational requirements characterizing the development situation under consideration. Finally the components of the logical architecture have to be implemented by means of physical components available on the market (physical perspective of the DWQ framework²⁸ and implementation description in the sense of the ARIS framework²⁹). With respect to the DWQ and ARIS frameworks and widely accepted development models in the literature the development phases of information warehouse development processes are structured logically by three phases called *conceptual specification*, *design* (or programming in the large) and *implementation*. Any of these phases are defined by means of development decisions characterizing the development state of the system under consideration. The phases depend logically upon each other. The development decisions characterizing the phases are sketched below.

It is widely accepted that any system development process has to start with the *conceptual specification* defining “what” the system under consideration should do.³⁰ The conceptual specification focuses 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 which enable this specification.

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 model³² and the user interface³³ 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 as well as documentation.

Of course these phases are integrated in an evolutionary development process with jumps back to (logically) previous phases if necessary.³⁴ Additionally, there could be a

²⁷ cf. Scheer (1998); Scheer (1999); Scheer (2000)

²⁸ cf. Jarke et al. (1999); Jarke et al. (2000)

²⁹ cf. Scheer (1998); Scheer (1999); Scheer (2000)

³⁰ cf. e.g. Pohl (1996); McMenamin, Palmer (1984); Davis (1990); IEEE (1984)

³¹ cf. e.g. Boehm (1981); Davis et al. (1988); Weske et al. (1999)

³² cf. Codd (1990); Date (1990); Embley (1998); Gupta, Horowitz (1991)

³³ cf. Balzert (1996)

³⁴ cf. Boehm (1981)

phase like maintenance. But since the phases are structured logically here and following NAGL³⁵ maintenance can be seen as another loop of the development process itself, and thus, another instantiation of the phases mentioned so far. For the purpose of this work the development process of information warehouses can be explained by the three phases *conceptual specification*, *design* (or programming in the large) and *implementation*.

2.3 Combining the Dimensions: IWH Development Framework

The framework developed in this section intends to clearly organize development tasks of information warehouse development processes. Development decisions on the type level are made in any of the three development phases. Decisions on the type level must be based on methodical preconditions which lead to models on the meta level. Since any development phase is required on any abstraction level the two dimensions discussed so far (abstraction levels and development phases) can be combined in a meaningful way to define the *information warehouse development framework*. Each of the two dimensions is characterized by three entities (three development phases and three levels of abstraction). Nevertheless the combination of the dimensions leads to a framework with eight entities only because there is no meaningful design phase on the instance level. Each of these eight entities of the IWH development framework is characterized by a specific combination of entities belonging to the respective dimensions (Figure 2).

Any of the eight boxes as components of the iwh development framework is characterized by a set of development tasks and respective input and output documents. The exchange of models and meta information between different boxes characterizes the iwh development process. The core idea of the concept presented in this paper is the interpretation of conceptual iwh models as meta information and their usage as parameters and meta data for components characterizing the iwh architecture. A detailed discussion of these topics is given in section 4.

³⁵ cf. Nagl (1990)

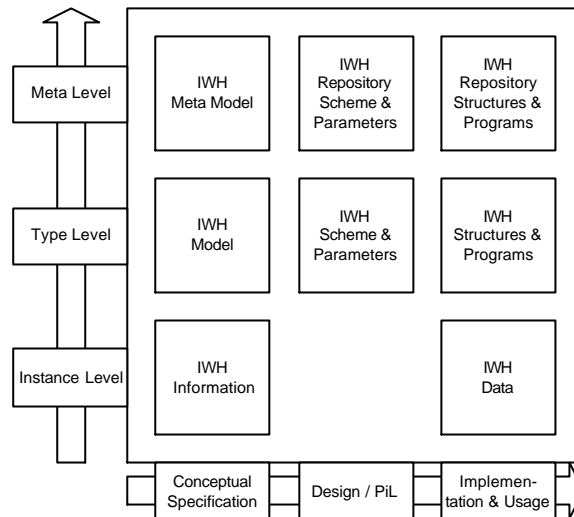


Figure 2: Information Warehouse Development Framework

Every abstraction level in the framework describes the development of certain information objects from the conceptual specification to their formalized and detailed implementation. The type level describes the phases required to develop the information warehouse itself. The meta level describes the development of the information warehouse repository which serves as means to store relevant meta information of the information warehouse. Finally the instance level describes the process of formalizing information coming up in a specific business process and the handling of this information by the information warehouse.

3 Conceptual Specification of Management's Information Requirements: Meta Information Creation

At the beginning of this chapter some introduction to the scientific background the rest of this paper is based on is given. Required information concerning a philosophy of science is presented to enable an understanding of the construction and modeling of language concepts. After the language for conceptually specifying management's information requirements is introduced a suitable and compatible representation formalism is presented. Finally the methodical usage of the modeling technique developed is demonstrated using a short example.

3.1 Scientific Background and Methodology

Background

The relationships between models, instances and languages are analyzed in this work based on the so called critical language approach.³⁶ The foundation of this approach is the constructive philosophy³⁷. Main ideas of this philosophy are that any scientific argumentation must be based on a so called critical language re-construction of the terms and concepts used in this argumentation. It is not allowed (since it would not make any sense) to use any concept that is not introduced explicitly in advance. This introduction of terms is called a *linguistic action* and is performed by statements. Only if respective definitions of concepts and terms are agreed on by a group of partners these persons form a so called language community able to participate in the argumentation.

The critical language approach was shown to make good use in the development of data base systems³⁸ and software systems in general³⁹. Recently it was shown how this approach can be used in the development of information warehouse systems.⁴⁰

The reason for the usage of the critical language approach in any of these works comprises two main arguments:

1. Any information systems' development has to focus the users views on the problem domain and to specify this views in a conceptual model.

³⁶ cf. Holten (1999); Ortner (1997); Lehmann, Jaszewski (1999); Wedekind (1981)

³⁷ cf. Kamlah, Lorenzen (1992); Lorenzen (1987); Lorenzen (2000)

³⁸ cf. Wedekind (1979); Wedekind (1981); Wedekind, Ortner (1977); Wedekind, Ortner (1980)

³⁹ cf. Ortner (1997); Ortner, Söllner (ObMe) (1989); Ortner, Söllner (Dict) (1989)

⁴⁰ cf. Becker, Holten (1998); Lehmann, Jaszewski (1999); Holten (1999)

2. This conceptual model has to be constructed stepwise based on critical language reconstructions of the concepts and terms used in the domain under consideration.⁴¹

For data oriented information systems the so called object type method was introduced to support the critical language construction of conceptual models.⁴² In works of SINZIG⁴³ and HOLTEN⁴⁴ it was shown how the respective construction operations can be used to develop entity relation ship models (The respective construction operations for ER models are introduced in the next section of this paper).

The importance of the defined levels of abstraction (see section 2.1) for the topic under consideration and their relationships to the critical language approach can be demonstrated with the following example: Specifying a management view on business processes is seen as a creative task here. It comprises two main linguistic actions which show that this modeling task cannot be automated: First a model component is created as an instance by the modeler (e.g. the model component called *Region*). Second this (new) object is declared explicitly as belonging to a certain class of objects (e.g. *Region* is declared to be a member of the class *DIMENSION*). The first linguistic action is called a *nomination* (creating an object characterized by an unique identifier) the second one is called a *predication* (declaring explicitly with a statement that this object belongs to a certain class). In terms of information systems modeling nominations (as linguistic actions) create so called *instance of-relations*.

The second kind of linguistic actions relevant for the current purpose (predications) also has its counterpart in information systems modeling. Predications (as linguistic actions) create so called *is a-relations*. By means of the modeler's statement "*Region* is a *DIMENSION*" the unique model component *Region* is assigned to the language class *DIMENSION* (predication). This of course means that *Region* is intended to have all characteristics members of the class *DIMENSIONS* have based on its definition (e.g. that they are structured hierarchically, see the following section).

Following the critical language approach there must be an agreement on the meaning of the term *DIMENSION* before it is used in the statement "*Region* is a *DIMENSION*". Otherwise this statement does not make any sense since nobody would know what is meant by *DIMENSION*. Defining the term *DIMENSION* as class of the language is a linguistic action (nomination) as well. This action defines the language construct which is given the unique name *DIMENSION*. It creates a new instance called *DIMENSION* of

⁴¹ cf. Ortner (1997); Lehmann, Jaszewski (1999)

⁴² cf. Ortner, Söllner (ObMe) (1989); Wedekind (1979); Wedekind (1981); Wedekind, Ortner (1977)

⁴³ cf. Sinzig (1990)

⁴⁴ cf. Holten (1999)

a class called *object type*. *Object type* of course is a class on a higher level of abstraction than *DIMENSION* and *Region*. The set of object types defining the language under consideration can itself be represented by another model (e.g. M2). Components of model M2 thus represent language concepts of L1 (see Figure 1).

Methodology

The conceptual specification of management's information requirements for information warehouses is seen as a specific requirements engineering task in this paper. Nevertheless as in any requirements engineering project a domain specific language is needed, which provides the concepts for the requirements specification.⁴⁵ In the DWQ approach development processes are argued to belong to the main determinants of data warehouse quality. Concerning the conceptual model and its integration with other perspectives (conceptual, logical, physical; see section 2.2) language concepts and respective relationships are presented in this approach⁴⁶.

The goal of the next section is to give an overview of an elaborate language and meta model intended to support the conceptual specification of management's information requirements.⁴⁷ The language is the core component of a method for information warehouse development. The language itself comprises two aspects⁴⁸: The *conceptual aspect* defines the language concepts with their meaning and relationships. The second aspect is called the *representational aspect* and relates representation formalisms to the language concepts. This intention of two language aspects can be found in the information systems literature as well. E.g. RUMBAUGH ET AL.⁴⁹ define the language concept *class* as a central term of the static view of the unified modeling language (UML) by means of the statement: "A class is the description of a concept from the application domain or the application solution"⁵⁰. A respective representation formalism is presented after defining the concept class.

In the following section the conceptual aspect of a language intended to specify management's information requirements and the respective meta model are constructed. A suitable and compatible representation formalism is presented in the section 3.3. With respect to Figure 1 (see section 2.1) language L1 on the type level is the matter of interest here. This language will be constructed methodically based on the critical language approach sketched above. It is important to remark that the idea of developing models

⁴⁵ cf. Pohl (1996); Jarke et al. (1993)

⁴⁶ cf. Jarke et al. (1999)

⁴⁷ cf. Becker, Holten (1998); Holten (1999); Holten (DW2000) (2000)

⁴⁸ cf. Holten (MobIS2000) (2000)

⁴⁹ cf. Rumbaugh et al. (1999)

⁵⁰ Rumbaugh et al. (1999), p. 25

based on the critical language approach normally is applied to construct information systems models like model M1 on the type level of Figure 1 in section 2.1. Nevertheless as shown by HOLTEN⁵¹ the language used to create models (e.g. language L1 on the type level in Figure 1) itself has to be developed based on this methodical approach. The goal of developing a so called *domain specific language* is to provide an instrument which is more suitable for the domain under consideration (in our case the conceptual specification of management's information requirements in the development of information warehouses) as a so called *domain neutral language* like e.g. UML.⁵² A domain specific language is expected to better fit the requirements in the modeling process since it provides language constructs which are more powerful in the current domain compared to domain neutral languages. It thus supports the creative task of constructing the conceptual model specifying information requirements of the management in the way that required model components can be expressed more easily.

With respect to Figure 1 in section 2.1 the process of developing model M1 on the type level is supported indirectly by providing more powerful language concepts of language L1 on the same level. Additionally section 3.2 develops meta model M2 on the meta level. Referring Figure 1 and the iwh development framework (Figure 2) the development of a language on a certain level (e.g. the type level) which is intended to support the process of constructing models on the same level (e.g. model M1) and providing a model of this language on the next higher level of abstraction (e.g. model M2 on the meta level) - both based on the critical language approach - is given the name *vertical integration of information systems development* here. Using a meta model like M2 on the meta level of Figure 1 to integrate the phases of the development process is referring the iwh development framework (Figure 2) given the name *horizontal integration of information systems development* here. Aspects of the horizontal integration are the matter of interest in chapter 4 of this paper.

3.2 Language and Meta Model for IWH Concepts

In this section the concepts required to specify management's information needs are introduced based on the critical language approach sketched above. The core concepts are constructed based on a thorough analysis of management and accounting literature⁵³. The following linguistic actions are used⁵⁴:

⁵¹ cf. Holten (MobIS2000) (2000)

⁵² cf. Holten (MobIS2000) (2000)

⁵³ see Holten (1999) for a detailed discussion

⁵⁴ cf. Holten (1999); Wedekind (1981)

- **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 modeled with the entity type symbol.
- **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 modeled 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 modeled 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 modeled by surrounding the respective relationship type symbol by an entity type symbol.

A summary of the concepts defining the conceptual aspect of the language together with the required linguistic actions is given in Table 1 (see Appendix) and the resulting meta model is shown in Figure 3. The first concept to be defined is *Dimension*. It is used to create and organize the space the management's view is composed of. There are compulsory dimensions (like time and scenario) because any management view must have a relation to time and to (optimistic or pessimistic) planning scenarios or the real business. All other dimensions are non compulsory. To be able to cluster dimensions the concept of *Dimension-Grouping* is introduced. E.g. in retailing companies it is necessary to look at different aspects of locations. It could e.g. be of interest which competitors have locations in the same area in order to classify the own locations according to this situation. Other important aspects are location sites (like down town, outskirts or island) or age and degree of modernization. From a logical point of view it is possible to combine any of these aspects or classifications with another if these combinations are of interest for a certain manager. It follows that these aspects are orthogonal and thus lead to different dimensions in the sense defined above. Nevertheless a grouping of dimensions is useful since there are different aspects (like site, age or competitor situation) of one object type (e.g. location) are concerned.

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 a-

rangements or examinations of the management.⁵⁵ The enterprise theory provided by RIEBEL is centered around the decision as the fundamental element.⁵⁶ Any activity in an enterprise is produced and maintained by certain decisions which therefore are the real sources of costs, outcome and liquidity. Based on RIEBELS findings the concept *Dimension-Object* is introduced. From the management's point of view some dimension objects may be characterized by strong relationships to each other in the sense that they define a certain dimension of the management's view on the enterprise. These strong relationships only exist because of domain specific reasons and are created by a modeler. In this sense dimension objects are said to be part of a dimension's definition. Dimension objects are organized in hierarchies (concept *DO-Hierarchy*). The concept of DO-Hierarchy allows the construction of e.g. product hierarchies or hierarchies of regions. Since dimensions are organized hierarchically it is useful to distinguish hierarchy levels (concept *Hierarchy-Level*). Any dimension object is associated to one hierarchical level. To be able to define reports according to individual information needs it must be possible to extract the required scopes from every hierarchy (concept *Dimension-Scope*) and create individual combinations of these dimension scopes (concept *Dimension-Scope-Combination*).

The last concept required to define the structure of a management view is *Reference-Object*. 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”⁵⁷. Based on the definition of dimension hierarchies a reference object can formally be understood as a vector with dimension objects as its coordinates. Using the formalized relations of the meta model components the whole space of reference objects is defined implicitly this way. From the business point of view a reference object is everything related to a decision in a business process. An example for an instance (*rol*) of the concept Reference-Object is the *set of cars which are sold to reselling partners in region Europe in 1999*. This reference object comprises four dimensions (products, clients, region, time) and its four coordinates are given by four dimension objects (cars, reselling partners, Europe, 1999). Since reference objects are vectors they can also be called combined reference objects (concept *Combined-Reference-Object*). The concept Reference-Object is a generalization of the concepts Dimension-Object and Combined-Reference-Object. Any of these concepts is comprised by RIEBELS definition given above. Nevertheless the concept Dimension-Object indicates that some reference objects fulfill a special role in defining dimensions. While using combined reference objects for the definition of one unequivocal dimension any other dimension of the combined reference object is not of

⁵⁵ cf. Riebel (1979)

⁵⁶ cf. Riebel (1992); Holten (1999)

⁵⁷ cf. Riebel (1979), p. 869

interest and will be ignored. E.g. while using the dimension object *cars* to define the product hierarchy of the enterprise (dimension *product*) it is not of interest that cars as decision objects can be characterized by many other attributes defining other dimensions relevant for the management. These dimensions are ignored while defining the hierarchical order of the dimension called product.

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 companies⁵⁸, the performance of the business⁵⁹ and to analyze the financial situation of an enterprise⁶⁰. 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 reference objects. Their economic meaning is clearly specified and their calculation is defined by means of algebraic expressions (e.g. “profit = contribution margin – fixed costs”). The entire Du Pont-Pyramid with its main ratio “return on investment” can be expressed based on algebraic expressions. Another form of organizing ratios is used by so called ratio systems (concept *Ratio-System*). Ratio systems are organized hierarchically 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 NORTON⁶¹ is a set of ratio systems supporting this top down analysis of reference objects in the strategic performance measurement process. Since the balanced scorecard organizes ratio systems hierarchically further concepts are required to be able to express this situation.

Finally the concepts defining the space of reference objects and the concepts concerning ratios as aspects of reference objects must be connected. Doing this a type of information relevant to the management is defined. Only if a clear definition of required ratios is given and this set of ratios is connected to a set of reference objects the definition of information relevant to the management is complete. For this purpose the new concept *Fact* which goes beyond the characteristics of the concepts defined so far is introduced. This concept is the core component required for the specification of management’s information needs. Any other concept introduced is required only to define clearly what is meant by the concept *Fact*. Defining a fact therefore requires the definition of a reference object (e.g. the former mentioned *set of cars which are sold to reselling partners in region Europe in 1999*) and connecting this to a given ratio (e.g. *profit*).

⁵⁸ cf. Copeland et al. (1990)

⁵⁹ cf. Johnson, Kaplan (1987); Eccles (1991); Lapsley, Mitchel (1996); Kaplan, Norton (1997); Kaplan, Norton (1996); Kaplan, Norton (1992)

⁶⁰ cf. Brealey, Myers (1996)

⁶¹ cf. Kaplan, Norton (1997); Kaplan, Norton (1996); Kaplan, Norton (1992)

The conceptual aspect of the language constructed so far is shown as a meta model in Figure 3.⁶² An extension of this language supporting the integration with technical aspects is given in section 4 of this paper. The meta model shown (and its extensions) are the basis for the development of tools supporting the conceptual definition of an information warehouse out of a management's perspective. It is located in the box called *IWH Meta Model* of the framework presented in section 2.3 (see Figure 2).

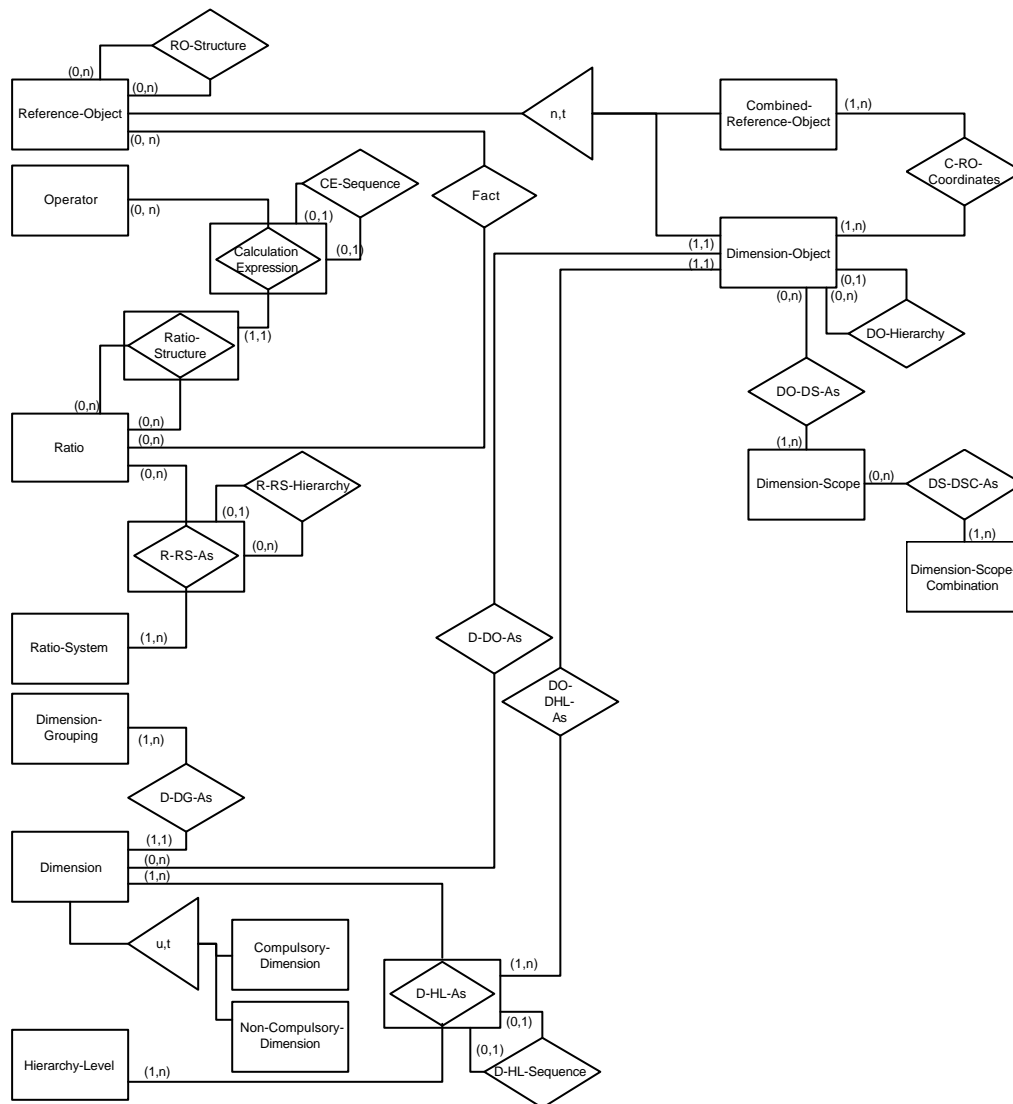


Figure 3: Meta model of language for information warehouse concepts

3.3 Representation and Technique for IWH Concepts

To be able to model the conceptual information warehouse model out of the management's perspective representation formalisms are required which consistently instantiate

⁶² see Table 1 in Appendix for construction details

the meta model developed in the previous section. A second requirement concerning a representation formalism is that it must be able to support discussions with the management. It thus should be easy to read and understandable for management users which are not interested in technical details like e.g. data warehouse database schemes. The modeling technique presented here is based on works of HOLTEN and KNACKSTEDT⁶³. The representation elements required are summarized in Table 2 (see Appendix).

The first representation elements to introduce concern dimensions and groupings of dimensions which are represented as boxes similar to entity types in the ERM approach. Symbols with sets of little boxes inside represent dimension groupings the other boxes represent dimensions. Identifiers are presented above the respective symbols. The association between dimensions and dimension groupings is represented by the line between the respective boxes. If all dimensions of a certain grouping of dimensions are selected no dimension is shown in the model to avoid information overflow. Dimension objects and their hierarchies are represented as trees. The associations of dimension objects to dimensions and their hierarchical levels are instantiated based on the dimensions identifiers given as their names. The levels of dimension hierarchies and their order are instantiated implicitly. Dimension scope combinations are represented by an unequivocal identifier and the respective scopes of any dimension involved.

The set of reference objects is represented using a relationship type symbol surrounded by an entity type symbol. This set is defined implicitly since it can be calculated as cartesian product out of the hierarchies of dimension objects and the dimension scope combinations. The role of dimension objects as coordinates for reference objects is represented by the line connecting the dimension symbols with the reference object symbol. Using this convention the space of reference objects relevant for the report under consideration is specified.

Ratios are represented as algebraic expressions and ratio systems as trees. The association of a ratio with a ratio system is represented by simply using it to make up the respective ratio system's tree. The tree itself also represents the hierarchical order within the respective ratio system. Ratio systems are identified by their names and additionally symbolized by boxes similar to entity type symbols with numbers inside.

Finally facts are represented as relationship type symbols connecting sets of reference objects (e.g. sales) and Ratio-Systems (e.g. DuPont ROI). A concrete set of facts defining a report relevant for management's decisions is defined implicitly this way.

⁶³ cf. Holten (1999); Holten, Knackstedt (1999)

3.4 Method for IWH Concepts: Creating IWH Meta Information

After having provided a language and a representation formalism for conceptually modeling an information warehouse the methodical usage of these instruments can be demonstrated with an example. Creating the conceptual information warehouse model as a development task is located in the box *IWH Model* (see Figure 2) of the development framework presented in section 2.3. As discussed in section 3.1 the development of the conceptual model should be based on the critical language approach. This approach thus is followed twice in this paper: First we constructed the language to model conceptual information warehouse models (language L1 on the type level of Figure 1) based on this approach in section 3.2, second we demand that the methodical usage of this language and the respective modeling technique provided in the last section is based on the critical language approach as well. This second point deals with the construction of models on the type level, e.g. M1 in Figure 1. The methodical creation of a conceptual model of the information warehouse comprises the following steps:

1. The dimensions and the hierarchies of dimension objects they are composed of must be defined and stored in the respective library. In Figure 4 this is shown on the left hand side for the dimension called “Sales Organization Hierarchy” in the upper window.
2. After having defined the dimension groups the dimensions belong to, combined reference objects required for a certain report can be constructed. The dimensions and dimension groups required are selected and combined to the reference object needed (right hand side of Figure 4).
3. To enable individual and management task oriented supply with information, dimension scopes are created and combined to the respective dimension scope combinations for the report under consideration. In the example of Figure 4 this is shown for a dimension scope combination called “Car Sales Parts of Europe” where the needed parts of the hierarchy defining the “Sales Organization” are selected and copied to the bottom window. Eg. the manager responsible for EXCOM’s car sales in Germany only requires the subset comprising the regional sub-hierarchy of all subordinated nodes of the node “Germany”. On the other hand there may exist another subset “Europe-South” for the management colleague responsible for car sales in the whole southern region of Europe. Since both managers are responsible for (different parts of) EXCOM’s car sales their different regional subsets are combined with the same (unique) subset “Cars” of the dimension hierarchy that defines the logical structure of EXCOM’s product portfolio. In this way any possible combina-

tion of dimension objects can easily be defined according to individual information requirements.

4. Ratios are defined from the business point of view and by the respective algebraic expressions in the case that a ratio must be calculated based on other ratios. Ratios are stored in libraries. These libraries guarantee that any ratio is used with one defined meaning and are helpful in overcoming the problems of homonyms and synonyms which is of fundamental importance in real world information warehouse projects. Figure 5 shows an algebraic expression for the ratio “Profit” (left hand side).
5. Ratio systems relevant for the management tasks the report under consideration is intended to support are created based on the ratios available in the respective library (Figure 5 left hand side).
6. The ratio system required for a certain report is connected to the required reference object which leads to the creation of the facts needed for this report. Fig. 5 shows this for the report “Sales”.

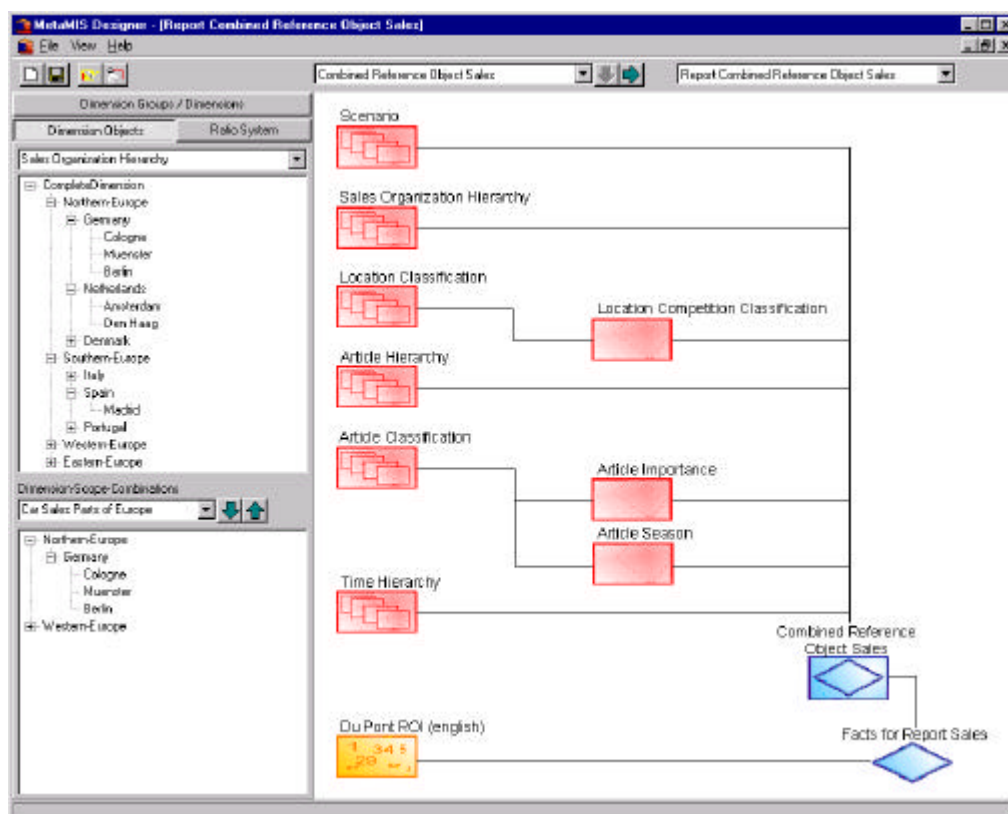


Figure 4: Conceptual Information Warehouse Model – Dimension Objects

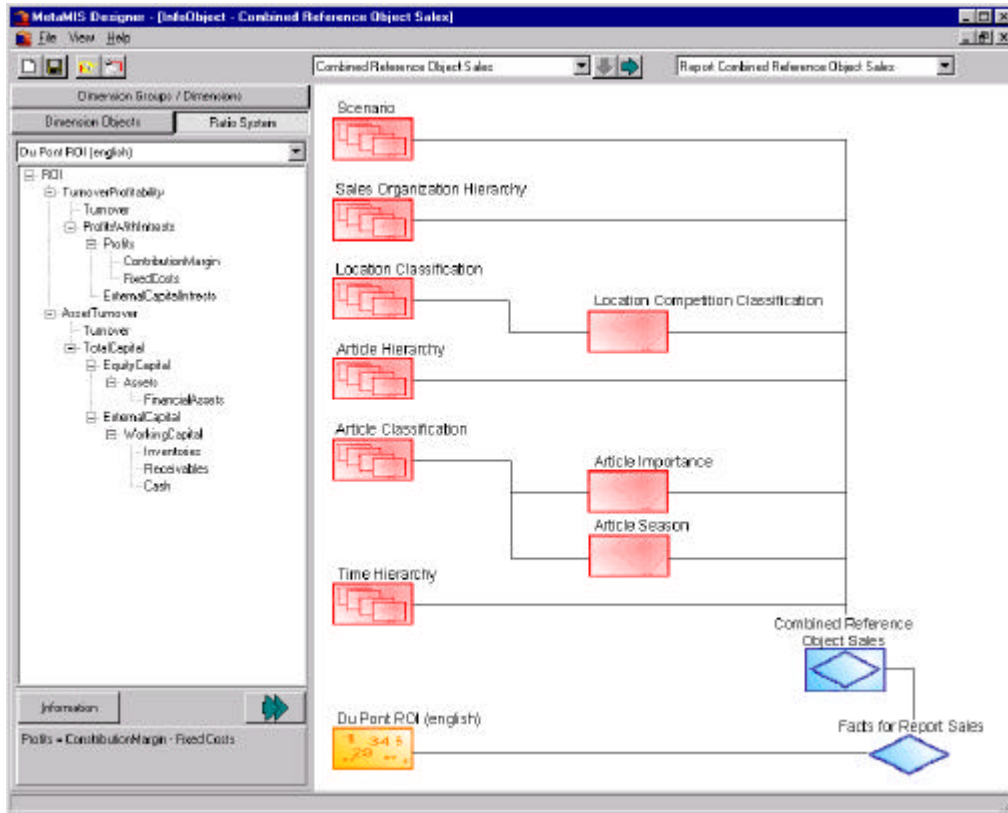


Figure 5: Conceptual Information Warehouse Model – Ratio Systems

4 Horizontal Integration of Information Warehouse Development: Meta Information Use

This section shows how conceptual models of information warehouses created based on the approach presented in section 4 can be used as meta information in the ongoing process of development, maintenance and usage of the information warehouse. The term horizontal integration indicates a re-usage of the conceptual model in later development phases with respect to the development framework discussed in section 2.3 (see Figure 2). For demonstration purposes the remarks in this chapter refer to the relational database model as far as the data warehouse scheme is concerned. This database model just serves as an example and was used as platform for an information system prototype. The ideas presented can be transferred to any other technical environment relevant to the purpose of information warehousing. Concerning the development phases the general design decisions characterizing these phases are of interest here (see section 2.2).

4.1 Enlargement of Language and Meta Model

With respect to the layer oriented information warehouse architecture discussed in section 1 it is assumed here that all components required to make up an information warehouse (oltp system, etl tool, data warehouse database management system, olap tool) are available on the market. For demonstration and prototyping purposes SAP R/3 was used as operational ERP system, Informatica PowerMart as etl tool, the data warehouse database was implemented using the Oracle 8 database management system and tools from MicroStrategy where used to implement the relational olap environment. The language is enlarged to integrate concepts introduced in section 3.2 with parts of the meta databases of the components used to realize the information warehouse. Concepts to express the input side of the etl tool and the transformation rules of the upload process are not the matter of interest here. Additionally it is assumed that queries required to get data out of the data warehouse are generated by the olap tool. An elaborate meta data approach discussing these technical aspects of the information warehouse environment in more detail is presented by STÖHR ET AL.⁶⁴

The enlargement of the language concerns concepts to describe simple data structures and standard queries. The respective concepts are summarized in Table 3 (see Appendix) and the resulting meta model is shown in Figure 6. The first concept introduced for the technical aspects of the information warehouse is *Data Attribute*. Data attributes are used in any data model to characterize data. There are two specializations of data attrib-

⁶⁴ cf. Stöhr et al. (1999)

utes relevant here concerning ratios (concept *Ratio Attribute*) and keys to identify data (concept *Key Attribute*). Ratios introduced as concepts of the conceptual model in section 3.2 are used to identify management oriented aspects of an information object within the database. Correspondingly dimensions as conceptual terms are interpreted as keys on the technical layer. The next step is to interpret dimension objects of the conceptual model as parts of the keys used to identify data on the technical layer (concept *Partial Key*). These partial keys become members of key instances (concept *Key Instance*) which in relation with ratio attributes are used to identify instances of facts (concept *Fact Instance*) in the database.

The connection of conceptual model and technical aspects of the information warehouse leads to the concept *Fact-Query*. This concept describes the relation between concept *Fact* defining a type of information for the conceptual model and concept *Fact Instance* specifying values stored in the database. Based on the construction presented any navigation discussed within the context of data warehousing and olap (drill down, slicing, ranging etc.) can be explained as movement between sets of facts on the conceptual level by means of the concepts RO-Structure and Ratio. The concept RO-Structure (see Table 1 in Appendix) allows the specification of movements from one reference object to another whereas the concept Ratio specifies aspects (ratios) required to complete the definition of a view for the management. If a set of facts is specified as result of a navigation operation on the conceptual level the corresponding values of these facts (defined by means of the concept Fact Instance on the technical level) are queried in the database (concept Fact-Query). Fact queries are combined to reports (concept *Report*). The conceptual structure of a report thus is explained by a set of facts relevant for a certain management task on the conceptual level while its instantiation is done by means of fact queries.

The integrated meta model of the language enlargement is shown in Figure 6. The concepts Ratio, Fact, Dimension and Dimension-Object show the integration with the meta model of Figure 3.

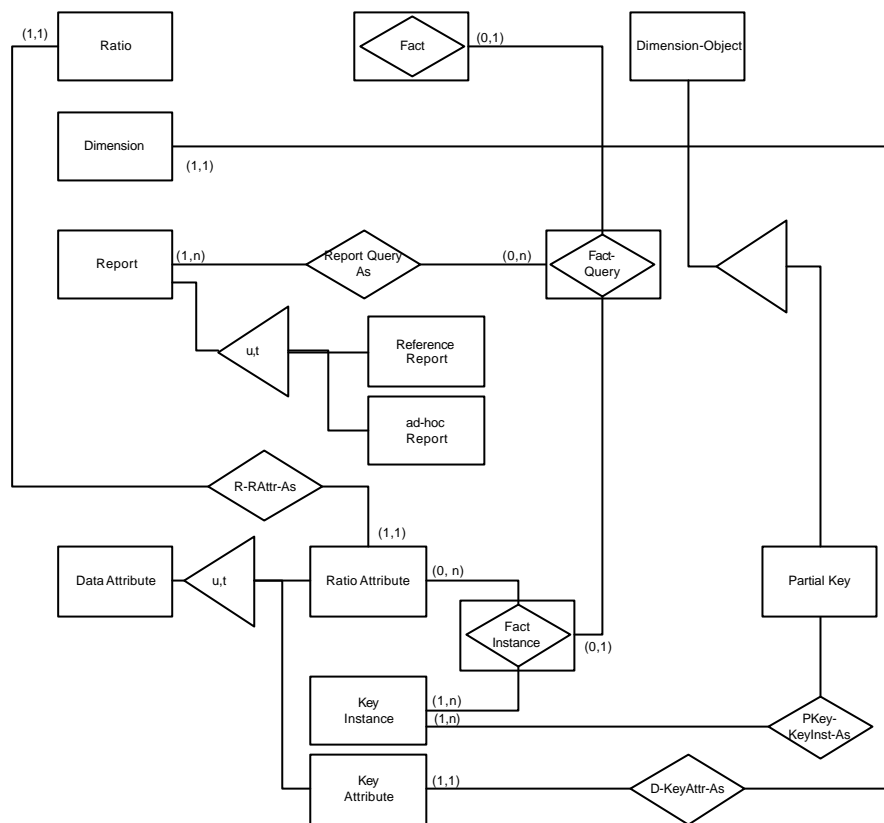


Figure 6: Enlargement of modeling language

4.2 Meta Information Repository and Development Tool

Based on the meta model constructed in section 3.2 a repository intended to store conceptual models of information warehouses can be developed. The enlargement of the language concerning technical aspects of the information warehouse in section 4.1 allows the integration of conceptual information warehouse models with meta data used as parameters by components the information warehouse is implemented with. This meta data integration requires a comparison of models describing the meta data of information warehouse components (e.g. etl tool, olap tool) with the meta model developed in this paper. This could lead to serious problems if the meta models of the components are not documented in the required form. It then would be necessary to reengineer these meta models by analyzing the respective repositories. The development of the repository storing information warehouse concepts itself is a standard software engineering task. The meta model has to be transformed into a repository scheme. This tool development process itself is not the matter of interest here. Obviously the main concepts discussed so far (e.g. dimensions, dimension groupings, dimension objects etc.) lead to relational tables in the repository scheme (provided the repository is realized on a relational database management system as platform).

The following relation schemes are used for demonstration purposes in the next sections. They are excerpts of a (possible) repository scheme. Attributes defining keys are underlined.

```

Dimension (D-ID, D-Name, DG-ID)
DimensionGrouping (DG-ID, DG-Name)
DimensionObject (DO-ID, DO-Name, DO-Father-ID, D-ID, HL-ID)
DimensionScope (DS-ID, DS-Name)
DO-DS-As (DO-ID, DS-ID)
DimensionScopeCombination (DSC-ID, DSC-Name)
DS-DSC-As (DS-ID, DSC-ID)
Ratio (R-ID, R-Name, CalculationExpression)
RatioSystem (RS-ID, RS-Name)
R-RS-As (R-ID, RS-ID, R-ID-Father, RS-ID-Father)
Facts-Report (FR-ID, Report-Name, DSC-ID, RS-ID)

```

Figure 7: Example for relational schemes of repository

With respect to the development framework proposed in section 2.3 the resulting repository scheme is located in the framework box *IWH Repository Scheme & Parameters*. This framework box is characterized by models and documents on the meta level (see Figure 1) and development decisions concerned with the design of an information system. Three classes of parameters are of interest with respect to the information warehouse architecture discussed in section 1:

- DWH parameters serve for the creation of the database scheme of the data warehouse database.
- Etl parameters are used to specify the upload process for the corresponding operational data.
- Olap parameters serve as basis for the generation of queries to navigate within the data warehouse.

Additionally this framework box contains the software architecture describing the components required to support the process of information warehouse development based on the approach presented here. All components are realized prototypically and operate on the basis of a central repository. The following components are required:

- The *IWH Designer* provides a graphical interface to develop conceptual information warehouse models based on the language and representation formalism presented in sections 3.2 and 3.3. The designer is used to fill the libraries of dimensions and their

hierarchies, to define dimension scope combinations, to create combined reference objects, to define ratios and ratio systems and finally to define reports as sets of facts by combining ratio systems with reference objects. Section 3.4 briefly sketched these tasks and Figure 4 and Figure 5 showed screen shots of the IWH Designer.

- Another tool specifies the so called data warehouse parameters and generates (suggestions of) the data warehouse database scheme (see section 4.3.1). Dimensions and dimension groups of the conceptual model are transformed into relation schemes for the data warehouse database scheme and respective create table statements are generated. Finally the resulting dimension tables can be populated.
- Based on the data warehouse database scheme the target structures the etl tool has to write in can be derived and the respective meta data can be generated (etl parameters) and be written into the etl tools meta database (see section 4.3.2). The mappings completing the definition of the etl process must be added by a technical engineer.
- Finally the meta data required for the olap tool (olap parameters) can be generated based on the conceptual information warehouse model. Components of the conceptual model (like e.g. dimension objects, dimension scopes, dimension scope combinations, ratios and ratio systems) must be transformed into components provided by the olap tool (e.g. templates, filters and metrics in the case of MicroStrategy) and be written into the respective meta database (see section 4.3.2). Meta data describing the data warehouse's database scheme also must be written into the olap tool's meta database.

Development activities concerned with the implementation of the repository storing information warehouse concepts and the tools discussed above and sketched in section 3.4 are located in the box *IWH Repository Structures & Programs* of the development framework (see Figure 2). These activities create and maintain models and documents (e.g. program structures and program code) on the meta level (cf. Figure 1) within the implementation and usage phase of the information system intended to support the development of information warehouses.

4.3 Using Meta Information

Having provided the language, technique, methodical steps and the respective tool support for information warehouse development processes in the preceding sections the methodical usage of meta information within information warehouse development processes can be discussed. With respect to the development framework shown in Figure 2

(see section 2.3) this usage of meta information is called horizontal integration since documents and models developed in the conceptual specification phase are (re-)used as meta information and parameters in the later development phases. The horizontal integration of information warehouse development refers to the boxes on the type level of the development framework.

Development decisions located in the framework box *IWH Scheme & Parameters* are concerned with database scheme design and the definition of meta data used by information warehouse components (like etl tools and olap tools) as parameters. The following remarks use the relational model as example to explain the re-usage of conceptual models as meta information.

4.3.1 Data Warehouse Parameters

These parameters are specified within the design phase of the information warehouse development process. Suggestions for data warehouse parameters can be derived out of the conceptual models. A completely automated generation of database schemes is not useful since the data warehouse engineer must be able to react on technical and project specific requirements which are not specified in the conceptual model. 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 in its usage phase.

Decisions concerning DWH parameters and their horizontal integration with the conceptual specification phase are demonstrated in the following using the example of a denormalized snowflake scheme. Respective design decisions could be demonstrated based on any other database model as well (e.g. a multi array oriented implementation of so called multi cubes based on flat file systems). The intention here is to show in which way development decisions belonging to different phases of the information warehouse development process (and thus are located in different boxes of the development framework shown in Figure 2 on the type level) can horizontally be integrated based on meta information.

Figure 8 shows an example of a denormalized snowflake scheme. Fact instances relevant to management are stored in a central fact table. Dimension tables structure data of business relevant dimensions with respect to aggregation needs. E.g. products are structured hierarchically in product groups. There is a dimension table for any aggregation level (e.g. product level, product group level). The scheme is denormalized since information describing e.g. product groups is stored both on the product group and the product level (see e.g. tables LK-Product and LK-Product Group in Figure 8). By using for-

each key relations the fact table is related to the root of every dimension defined as the bottom level table of the respective dimension (LK-Town, LK-Product, LK-Date, LK-Client). The scheme shown in Figure 8 is characterized as snowflake since the root table defining dimension time (LK-Date) has foreign key relations to three logically independent branches within the dimension time. Thus, a given date is related to one week (LK-Week), to one month (LK-Month) and finally to one weekday (LK-DayOfWeek).

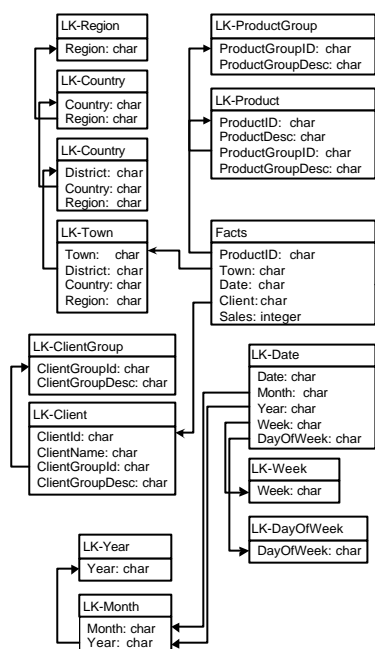


Figure 8: Data Warehouse Scheme

Decisions concerning the development of a data warehouse database scheme in the case of a snowflake scheme comprise the following tasks:

- Definition of the fact table attributes.
- Definition of the bottom level for every dimension.
- Definition of branches within every dimension.
- Definition of hierarchical levels of every branch of every dimension.
- Definition of foreign key relations within every branch of every dimension

Since the language (and the meta model) introduced in section 3 guarantee a properly structured conceptual model of the information warehouse (which could e.g. be stored based on the repository scheme shown in Figure 7) it can be used to generate suggestions for the database scheme in the following way (cf. concepts defined in Table 1):

Definition of fact table attributes

In the following it is assumed, that the dimensions of the database scheme are generated based on dimension groups (concept Dimension-Grouping). The dimensions of the conceptual model (concept Dimension) will then lead to different branches of the dimension of the database scheme which corresponds to the respective dimension group of the conceptual model. This makes sense since the fact table under construction should be as small as possible. The set of required dimensions for the database scheme can be calculated based on the repository scheme given in Figure 7 by means of the following simple relational expression.

p DG - ID, DG - Name (Dimension Grouping).

These dimensions define the key attributes required for the fact table. The attributes required for measures can be calculated by means of the following expression where only base ratios are used. The calculation of complex expressions has to be performed by the ROLAP tool.

s R - ID, R - Name _(CalculationExpression = NIL) (Ratio).

Definition of the bottom level for every dimension

Since dimension groups of the conceptual model are transformed into dimensions of the data warehouse scheme the dimensions of the conceptual model become branches within the resulting snowflake scheme. For this reason the relationships of the hierarchies' leaves defining dimensions of the conceptual model must be analyzed. The concept of Dimension-Grouping (see Table 1) was defined to characterize different aspects of one unequivocal object type. Leaves of the hierarchies of dimension objects defining dimensions of one dimension groups are thus different representations of the same set of objects. From an algebraic point of view there exist isomorphisms relating every object on the bottom level of one dimension of a dimension group to every other dimensions bottom levels of the same dimension group. What follows is that there exist equivalence classes with respect to these isomorphisms. These equivalence classes define the bottom level of the dimensions in the data warehouse scheme since they are related to the different branches of the snowflake scheme characterizing the respective dimension. Formally this can be stated as follows:

Given a $\overline{dg_m} \in DimensionGrouping$

The set of dimensions belonging to this dimension group is given by

$A : s_{(DG=\overline{dg_m})}$ (Dimension)

The set of all dimension objects which are leaves in any dimension are given by the next expression provided the lowest hierarchical level is indicated as “1”.

$$B : s_{(HL=1)} \text{ (Dimension Object)}$$

Joining sets A and B leads to the set of all dimension objects which are on the bottom level of the dimension group under consideration

$$C : A \triangleright \triangleleft_{D-ID} B$$

Now it is important to note that the relation scheme called DimensionObject (see Figure 7) is used to store the equivalence classes of dimension objects in the way mentioned above. What follows is that the unequivocal key for tuples of this relation scheme is given by combining the three attributes DO-ID, D-ID, HL. By means of this key the different representations of objects used to define the respective dimensions are identified. These dimension objects are related on the bottom level (HL = 1) by isomorphisms. To identify the respective equivalence classes the (partial) key DO-ID can be used. Using this the required set of equivalence classes used to define the bottom level of the respective dimension “dg_m” of the data warehouse database scheme is given by

$$D : p_{DO-ID} (C)$$

Definition of hierarchical levels and foreign key relations within the dimensions

Using statement

$$A : s_{(DG=dg_m)} \text{ (Dimension)}$$

again the required branches of the database scheme dimension “dg_m” can be obtained. Joining this set with relation DimensionObject leads to the set of all dimension objects of the database scheme dimension under consideration:

$$E : A \triangleright \triangleleft_{D-ID} \text{DimensionObject}$$

To calculate the hierarchical levels within this dimension and the required foreign key relations only a sub set of information is required given by the following expression:

$$P_{(DO-ID, D-ID, HL-ID, DO-Father-ID)} (E)$$

As shown the logic of the management view on the business which is specified by means of the conceptual model can be transformed into suggestions for the database scheme based on the rules discussed above. Any technical specification of data types and attributes relevant for the implementation of this scheme is independent of the logic

defined within the conceptual model. Database engineers have to make their decisions concerning these aspects in the implementation phase.

4.3.2 Etl and Olap Parameters

Etl tools use (parts of the) meta data defining the data warehouse scheme as parameters for the data warehouse upload process. An approach to model technical meta data for data warehousing environments is proposed by STÖHR ET AL⁶⁵. According to this approach some examples of meta data are shown in Figure 9. Note that this kind of notation (Figure 9) is another way of representing *the same* information shown in Figure 8. Nevertheless Figure 8 is a representation which is used by human software designers, whereas the structure shown in Figure 9 is oriented at the relational database model. Both kinds of representation are located in the framework box called *IWH Scheme & Parameters* (see Figure 2) if development decisions are the focus of interest. From a technical point of view this information is located in the etl tool and olap tool repositories. Since both forms of representation show the same information concerning meta data describing database schemes the rules for the suggestion of the data warehouse scheme based on the conceptual model (see section 4.3.1) can be applied to generate suggestions for these meta data as well.

DS-Type		
DataStructure	DS-Type	DS-DWH-Function
Facts	relational table	fact table
LK-Town	relational table	dimension table
LK-District	relational table	dimension table
...

DS-Attributes			Dimensions	
DataStructure	Attributes	A-Type	Table	Dimension
Facts	Product	Char	LK-Product	Products
Facts	Town	Char	LK-ProductGroup	Products
Facts	Date	Char	LK-Town	Region
Facts	Client	Char	LK-District	Region
Facts	Sales	Integer	LK-Country	Region
...	LK-Region	Region
LK-Town	Town	Char	LK-Date	Time
LK-Town	District	Char	LK-Month	Time
LK-Town	Country	Char	LK-Year	Time
...	LK-Week	Time
			LK-DayOfWeek	Time

ForeignKey-Associations		
Attribute	Primary	Foreign
ProductID	LK-Product	Facts
ProductGroupID	LK-ProductGroup	LK-Product
...
Country	LK-Country	LK-District
Country	LK-Country	LK-Town
...
Year	LK-Year	LK-Date
Year	LK-Year	LK-Month
...

Figure 9: Information Warehouse Parameters

⁶⁵ cf. Stöhr et al. (1999)

Technical development tasks concerned e.g. with the definition of data types and other technical parameters depend on the components used to implement the information warehouse. They are the matter of interest within the implementation phase. These tasks are located in the framework box *IWH Structures & Programs* (see Figure 2) on the type level. While implementing an information warehouse the parameters discussed above are used to specify exactly the tasks of etl and olap tools. Based on a short example this is shown in the following.

The sales transaction introduced in section 2.1 is booked in EXCOM's oltp system (① Figure 10) and has to be transformed into the integrated data format of the data warehouse using the etl tool. For this purpose meta data describing the oltp systems' data structures are read by the etl tool. Providing these meta data is a technical task which is beyond the scope of this paper. The sales transaction could e.g. be stored in a set of SAP R/3 tables or any other data source (② Figure 10). The next step is to read the meta data describing the data warehouse scheme to determine the target structures and the required data formats for the data warehouse upload (③ Figure 10). In our example the etl tool will find that the required information must be stored in the fact table. This meta information can be provided based on the rules discussed in section 4.3.1. Finally etl tools need meta data to define required transformations (called mappings in the case of Informatica) of source data into data warehouse data (④ Figure 10). This is the third kind of meta data stored in the etl repository. Again, these meta information cannot be generated out of the conceptual model of the information warehouse. The source data may be stored in any data type in the oltp system and must e.g. be transformed into character strings for the key attributes of the fact table and into integer format for the sales attribute. In the example the row stored in the fact table will be composed of the following attribute values, where "xxx" indicates xxx to be a character string: (Product: "4711"; Region: "Muenster"; Date: "19991108"; Client: "Westfalia Cars"; Sales: 28.000).

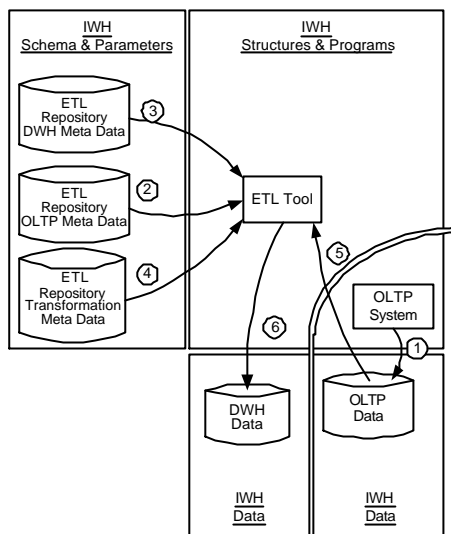


Figure 10: Information Warehouse Load

Reading oltp data, transforming them and writing them into the data warehouse in an integrated format are the next steps for the etl tool to perform (⑤, ⑥ Figure 10). These tasks of the etl process are performed periodically. Examples of fact data stored in the fact table are given in Figure 11. Data stored in the etl repository serve as functional parameters for transformation and integration routines of the etl tool (and thus are located in framework box *IWH Scheme & Parameters*). Data stored in the fact table of the data warehouse or in the oltp systems data store are both located in the framework box *IWH Data* on the instance level. The information generated within the business process is located in the framework box *IWH Information* on the instance level (see Figure 2).

Facts					
Product	Town	Date	Client	Sales	...
4711	Muenster	19991108	Westfalia Cars	28.000	
4711	Dortmund	19991107	Phoenix	28.000	
4712	Muenster	19991105	Westfalia Cars	31.000	
...	

LK-ProductGroup	
ProductGroup	ProductGroupDesc
1	Cars
2	Equipment
3	Sparks
4	Electronics

LK-Product			
Product	ProductDesc	ProductGroup	ProductGroupDesc
4711	FunRacer	1	Cars
4711	Compact	1	Cars
4813	Spoiler	2	Equipment
...

Figure 11: Information Warehouse Data

In order to instantiate reports and multidimensional views for the management the olap tool is used. Reports are defined based on concepts provided by the olap tool (e.g. filters, metrics and templates in the case of MicroStrategy) and these definitions are stored

in the repository of the olap tool. Using these concepts as parameters in connection with the meta data describing the data warehouse scheme the olap tool transforms structured report definitions into SQL statements and sends these queries to the database management system of the data warehouse (①, ② Figure 12). These SQL queries are located in the framework box *IWH Structures & Programs* of the development framework (see Figure 2).

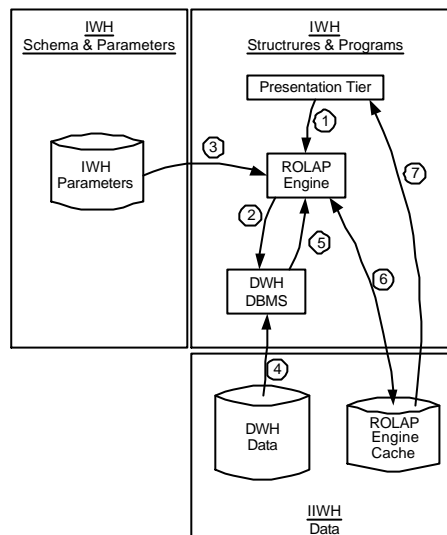


Figure 12: Information Warehouse Usage

1999 Sales Europe in Mil.EURO	
Products	Sales
...	...
...	...
...	...
...	...
...	...

Figure 13: Desired Report Structure

Figure 13 as a (very simple) report definition prescribes that any data is needed that belongs to the period year 1999 and gives information about sales in EURO for any product. This report definition can be specified with a conceptual model (cf. Table 1 for the concepts) by means of the dimensions “time”, “region” and “products” (concept Dimension), the dimension objects “1999”, “Europe” and “all” for these dimensions (concepts Dimension, Dimension-Object, D-DO-As) and the ratio “Sales” (concept Ratio). The required meta information can be generated out of the conceptual model of the information warehouse by means of the following rules (cf. the repository scheme in Figure 7):

A report is conceptually defined by means of a so called dimension scope combination and a respective ratio system (see section 3.2). Ratios (called metrics in the case of MicroStrategy) of a certain report can be determined by the expression

$$F : \text{Ratio} \triangleright \triangleleft_{(R-ID)} (R - RS - AS \triangleright \triangleleft_{(RS-ID)} \text{Facts} - \text{Report})$$

Required dimension objects together with their dimensions and hierarchy levels are given by the expression

$$G : \text{DimensionObject}_{(DO-ID)} \triangleright \triangleleft (DO - DS - AS \triangleright \triangleleft_{(DS-ID)} (DS - DSC - AS \triangleright \triangleleft_{(DSC-ID)} \text{Facts} - \text{Report}))$$

Figure 13 additionally defines the presentation structure of the report. This specification must be added to the conceptual model as a technical parameter.

Referring to the information warehouse parameters shown in Figure 9 the attribute “Sales” of the data structure (the table) “Facts” is of interest for the query generation. Additionally the foreign key relations of the attributes “Year” and “ProductID” between the data structures Facts, LK-Date and LK-Product respectively which are defined in table “ForeignKey-Association” are required for the query generation. Finally data of attribute “Date” of data structure “Facts” can be qualified against data of attribute value “Year = 1999” of data structure LK-Date using the foreign key relations defined in the table “ForeignKey-Associations” (Figure 9). This meta information is used by the olap tool (③ Figure 12). It then generates complex SQL queries which read from both fact and dimension tables of the data warehouse as well.

For the example the generation of the SQL statement proceeds as follows. Based on the report definition the required attributes are identified (`select` and `from` clauses) and the required join operations are generated based on foreign key associations (table ForeignKey-Associations, Figure 9) leading to the main part of the `where` clause. The value of the attribute “Year” in table LK-Date is qualified to “1999” (`where` clause). Finally the presentation structure of the desired report is used to generate the required `group by` clause. To guarantee that there is only one description for one product in the result table the `max` operator is used. This is necessary since it could be possible that during the etl process some integration tasks like transforming different product descriptions for the same product from different oltp systems into one data warehouse representation are omitted (which could be reasonable to speed up the etl process). For that purpose the `max` operator is applied to attribute `Product_Desc` (which is of type `char`) in the `select` clause in order to choose one (arbitrary) product description for all products with the same product id. This operator is required since it prevents the same

product to be listed several times in the result table if there does exist more than one product description for one product. The second reason for this operator is that every argument of the `Select` clause must become an argument of the `group by` clause unless an aggregation operator is used. In our example SQL generated by the olap tool could e.g. look like the following statement:

```
Select      a2.[Product_ID] as [Product_ID],
           max(a2.[Product_Desc]) as [Product],
           SUM(a1.[Sales])) as [Sales]
from        [Facts].a1,
           [LK-Product].a2,
           [LK-Date].a3
where       a2.[Product_ID] = a1.[Product_ID]
and         a3.[Date] = a1.[Date]
and         a3.[Year] = (1999)
group by   a2.[Product_ID]
```

The result sets are calculated by the database management system of the data warehouse, formatted according to the report definition and usually cached by the olap tool (④ to ⑥ Figure 12). Tools on the presentation tier create formats presented to the management user (⑦ Figure 12), e.g. the report shown in Figure 14 (where product ids are omitted). Any olap functionality (like e.g. rotation, slicing, dicing, drill-down or roll-up) is realized generating SQL queries by the olap tool based on meta information as shown above.

1999 Sales Europe in Mil.EURO	
Products	Sales
Cars	30.000
Equipment	11.580
Sports	19.758
Electronics	23.423

Figure 14: Report Results

5 Related Work and Discussion

The information warehouse development framework presented in section 2 is build on the separation of three development phases which are combined with three model and abstraction layers to organize development decisions characterizing information warehouse development processes. Compared to the framework of the DWQ approach⁶⁶ the same phases are used. Nevertheless the abstraction layers analyzed in JARKE ET AL⁶⁷ and VASSILIADIS ET AL⁶⁸ are shifted upwards compared to the framework presented in this paper. DWQ focuses the questions of quality oriented development and integration of data warehouse components within development and change processes (see the meta model based approach in JARKE ET AL⁶⁹). The DWQ approach concentrates on the development of methodologies in general with a focus on quality of developed information systems and development processes as well. For that reason the process of process development is analyzed in more detail than in the approach presented in this paper.

The quality oriented aspect of usefulness, which concentrates on the data warehouse access according to users' work⁷⁰, has strong relations to the approach presented in this paper. Both approaches, DWQ and the one presented here, stress the central role of the conceptual model of the information warehouse. The main idea presented in this paper is to create a conceptual model of the information warehouse that fits the information requirements of management before making any design or implementation decisions and using this conceptual model as meta information for development decisions in later development phases. For that purpose a method is proposed and integrated into the framework presented. The focus is on language (comprising concepts and representation formalism), modeling technique and tool supported method. The DWQ approach on the other hand motivates the central role of the conceptual model to assure quality to documents and processes but does not discuss languages, techniques and methods to create conceptual models in concrete terms.

Another approach to conceptually modeling the data warehouses is presented by GOLFARELLI, RIZZI and GOLFARELLI ET AL⁷¹. A graphical 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 for-

⁶⁶ cf. Jarke et al. (2000); Jarke et al. (1999); Vassiliadis et al. (2000)

⁶⁷ cf. Jarke et al. (2000); Jarke et al. (1999)

⁶⁸ cf. Vassiliadis et al. (2000)

⁶⁹ cf. Jarke et al. (2000), pp. 128, 135

⁷⁰ cf. Jarke et al. (2000)

⁷¹ cf. Golfarelli, Rizzi (1999); Golfarelli, Rizzi (1998); Golfarelli et al. (JCIS) (1998); Golfarelli et al. (Hawaii) (1998)

mal description 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. Compared to the approach presented in this paper the fact model approach complements the suggestions for the data warehouse scheme generated based on the conceptual model. The approach presented in this paper can be seen as top down approach since it starts with the analysis of managements information requirements. The language concepts and representation formalism presented are intended to formalize these requirements by means of concepts and terms used by the management without any technical restrictions. In later development phases, especially in the design phase, this conceptual model must be transformed into a data base scheme for the data warehouse. Technical requirements, especially limitations and possibilities concerning the operational data sources, are core parameters of these development decisions. The approach presented by GOLFARELLI, RIZZI and GOLFARELLI ET AL.⁷² logically completes the approach presented in this paper since it is a bottom up approach allowing the formalized analysis of oltp systems data structures and the transformation of this analysis into a conceptual description. The integration of the two approaches 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.

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.⁷³ and MOODY, KORTINK⁷⁴. The approach presented by HÜSEMANN ET AL.⁷⁵ starts by identifying reasonable measures in the oltp systems schemes whereas MOODY, KORTINK⁷⁶ start by classifying the oltp systems entities. Both approaches concentrate on development decisions in the design phase. A meta data based approach to generic graphical model design is presented by SAPIA ET AL.⁷⁷. Compared to the approach presented in this paper there is no domain specific knowledge which can be used to create graphical notations. On the opposite the notation discussed in this paper is developed based on the domain specific analysis of concepts required.

⁷² cf. Golfarelli, Rizzi (1998); Golfarelli et al. (JCIS) (1998); Golfarelli et al. (Hawaii) (1998)

⁷³ cf. Hüsemann et al. (2000)

⁷⁴ cf. Moody, Kortink (2000)

⁷⁵ cf. Hüsemann et al. (2000)

⁷⁶ cf. Moody, Kortink (2000)

⁷⁷ cf. Sapia et al. (2000)

6 Conclusion

This paper deals with methodical aspects of information warehouse development processes. The focus is on the development and usage of conceptual models. Development processes are analysed based on a framework separating three development phases and three model and abstraction levels. This framework organizes development decisions with respect to their logical order and shows the relationships of decisions within the development processes to decisions of the method development process. Based on this framework the paper discusses development and usage of methods for conceptually modelling the information warehouse. Additionally it is shown how conceptual models of the information warehouse can be used as meta information and parameters in the later development phases.

Main contributions of the paper are the discussion of fundamentals of method creation for conceptually modelling information warehouses based on the constructive philosophy presented by LORENZEN and KAMLAH⁷⁸ and the methodological integration of conceptual models as meta information in the information warehouse development process. Based on the constructive philosophy of LORENZEN and KAMLAH⁷⁹ and the constructive language approach⁸⁰ the construction operators of the so called object type method are applied to construct language concepts required for the conceptual specification of information warehouses. The respective concepts are developed based upon the domain specific management and accounting literature. The language is intended to specify information warehouses without any technical aspects. The resulting language is modelled on a meta level and a representation formalism for the language is presented. The method for conceptually modelling information warehouses is completed by defining modelling tasks and their logical order. Central libraries are core components of this method. The method is supported by a tool which is developed based on the meta model of the language constructed. A repository storing libraries and conceptual models is the core component of the tool.

Conceptual models developed based on the approach presented play central roles in the development processes since they are (re-)used as meta information and parameters in later development phases. The paper shows how conceptual models stored in a central repository can be transformed into meta data used by etl and olap tools available on the market.

⁷⁸ cf. Lorenzen (1987); Lorenzen (2000); Kamlah, Lorenzen (1992)

⁷⁹ cf. Lorenzen (1987); Lorenzen (2000); Kamlah, Lorenzen (1992)

⁸⁰ cf. Wedekind (1981)

References

- Agarwal, S.; Agrawal, R.; Deshpande, P. M.; Gupta, A.; Naughton, J. F.; Ramakrishnan, R., Sarawagi, S.: On the computation of multidimensional 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 Alamos, 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.
- 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.
- Boehm, B.: Software-Engineering Economics. Prentice Hall 1981.
- 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.
- 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 & Associates, 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.
- 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.
- 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. (JCIS): 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. (Hawaii): 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.
- Holten, R. (DW2000): 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. (MobIS2000): Entwicklung einer Modellierungstechnik für Data Warehouse-Fachkonzepte. In: H. Schmidt (Hrsg.): *Modellierung betrieblicher Informationssysteme*, *Proceedings der MobIS-Fachtagung 2000*, 11.-12.10.2000, Siegen; *Rundbrief der GI-Fachgruppe 5.10*, 7 (1) Oktober 2000, S. 3-21.
- Holten, R.: *Entwicklung von Führungsinformationssystemen. Ein methodenorientierter Ansatz*. Wiesbaden 1999.
- Holten, R.; Knackstedt, R.: Fachkonzeptuelle Modellierung von Führungsinformationssystemen am Beispiel eines filialisierenden Einzelhandelsunternehmens. In: E.J. Sinz et al. (Hrsg.): *Tagungsband MobIS '99 Rundbrief Informationssystem-Architekturen des FA 5.2*. Oktober 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/>
- IEEE-830: *Guide to Software Requirements Specification*. ANSI/IEEE Std. 830, 1984.
- 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.

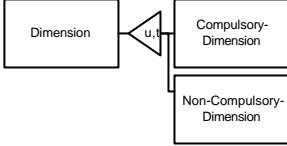

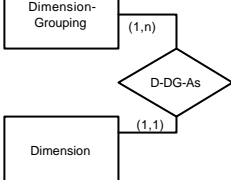
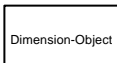
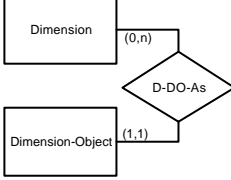
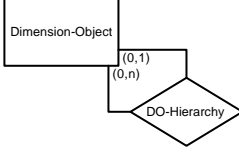
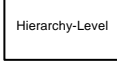
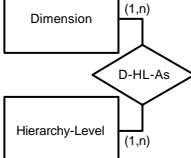
- Inmon, W. H.; Welch, J. D.; Glasey, K. L.: *Managing the Data Warehouse*. New York et al. 1997.
- ISO: *ISO/IEC 10027: Information Technology. Information Resource Dictionary Systems (IRDS)-Framework*. ISO/IEC Intl. Standard, 1990.
- Jacobs, S.; Holten, R.: *Goal Driven Business Modeling. Supporting 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 Press 1995, S. 96-105.
- Jarke, M.; Pohl, K.; Jacobs, S.; Bubenko, J.; Asenova, P.; Holm, P.; Wangler, B.; Roland, C.; Plihon, V.; Schmitt, R.; Sutcliffe, A.; Jones, S.; Maiden, N, Till, D.; Vasiliou, Y.; Constantopoulos, P.; Spanoudakis, G.: *Requirements Engineering. An Integrated View of Representation, Process and Domain*. Proc. of the Fourth European Software Engineering Conference, Garmisch-Partenkirchen, Germany, Springer-Verlag 1993, pp. 100-114.
- 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.
- Johnson, H. T.; Kaplan, R. S.: *Relevance Lost. The Rise and Fall of Management Accounting*. Boston 1987.
- Kamlah, W.; Lorenzen, P.: *Logische Propädeutik: Vorschule des vernünftigen Redens*. Stuttgart 1992
- 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.
- Kimbal, R.: *The Data Warehouse Toolkit*. Wiley, New York et al. 1996.
- Lapsley, I.; Mitchel, F.: *The Accounting Challenge: Accounting and Performance Measurement*. London 1996.
- Lehmann, P.; Jaszewski, J.: *Business Terms as a Critical Success Factor for Data Warehousing*. In: S. Gatzju, M. Jeusfeld, M. Staudt, Y. Vassiliou (eds.), *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> (2000-11-20).
- Lorenzen, P.: *Constructive Philosophy*. Amherst, University of Massachusetts Press 1987.
- Lorenzen, P.: *Lehrbuch der konstruktiven Wissenschaftstheorie*. Stuttgart 2000.

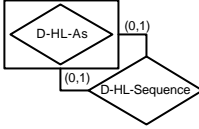
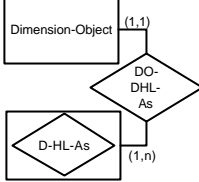
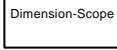
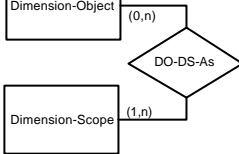
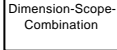
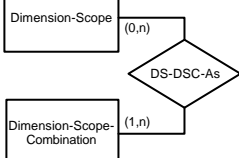
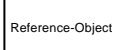
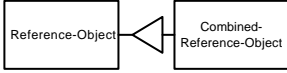
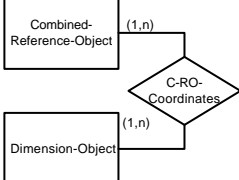
- McMenamin, S. M.; Palmer, J. F.: *Essential System Analysis*. Yourdon Press, Prentice Hall. Englewood Cliffs, NY, 1984.
- Moody, D. L.; Kortink, M. A. R.: *From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design*. Proceedings of the International Workshop 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.
- Ortner, E.: *Methodenneutraler Fachentwurf*. Stuttgart, Leipzig 1997.
- Ortner, E.; Söllner, B. (Dict): *Konzept und Einsatz eines Data Dictionary bei DATEV*. *Informatik Spektrum*, 12 (1989) 6, S. 82-92.
- Ortner, E.; Söllner, B. (ObMe): *Semantische Datenmodellierung nach der Objekttypen-Methode*. *Informatik Spektrum*, 12 (1989) 12, S. 31-42.
- Pohl, K.: *Process-Centered Requirements Engineering*. Research Studies Press, Taunton, Somerset 1996.
- Riebel, P.: *Einzelkosten- und Deckungsbeitragsrechnung. Grundlagen einer markt- und entscheidungsorientierten Unternehmensrechnung*. 7. Auflage, Wiesbaden 1994.
- Riebel, P.: *Einzelerslös-, Einzelkosten- und Deckungsbeitragsrechnung als Kern einer ganzheitlichen Führungsrechnung*. 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.: *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*. 3rd ed., Springer, Berlin 1999.
- Scheer, A.-W.: *ARIS – Business Process Modeling*. 3rd ed., Springer, Berlin 2000.
- Scheer, A.-W.: *Business Process Engineering*. 3rd ed., Springer, Berlin 1998.
- Sinzig, W.: *Datenbankorientiertes Rechnungswesen. Grundzüge einer EDV-gestützten Realisierung der Einzelkosten- und Deckungsbeitragsrechnung*. 3. Aufl., Berlin et al. 1990
- 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.
- 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.
- Wedekind, H.: Die Objekttypen-Methode beim Datenbankentwurf - dargestellt am Beispiel von Buchungs- und Abrechnungssystemen. ZfB, 49 (1979) 5, S. 367-387.
- 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.; Ortner, E.: Der Aufbau einer Datenbank für die Kostenrechnung. Die Betriebswirtschaft, 37 (1977) 4, S. 533-542.
- Wedekind, H.; Ortner, E.: Systematisches Konstruieren von Datenbankanwendungen. Zur Methodologie der Angewandten Informatik. München, Wien 1980.
- 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.

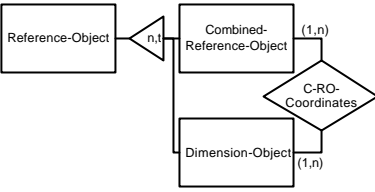
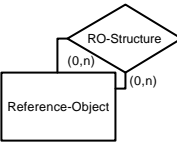

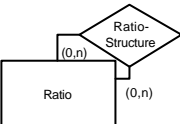
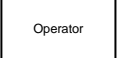
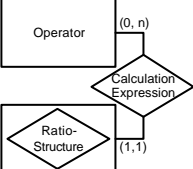
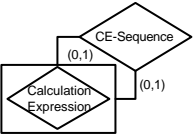
Appendix

Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
Dimension	Subsumption: Used to create and organize the space the management's view is composed of.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Dimension</div>

Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
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-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 Association)	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.	
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 unequivocal relationship to one unequivocal dimension (cardinalities (1,1)).	
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)).	

Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
D-HL-Sequence (Dimension Hierarchy-Level Association Sequence)	Composition: There is a unequivocal order of the 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)).	
Reference-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”, ⁸¹ .	
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	

⁸¹ Riebel (1979), p. 869

Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
	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 a 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.	
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.	
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).	

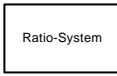
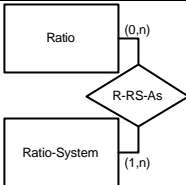
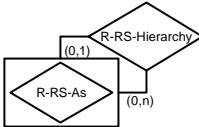
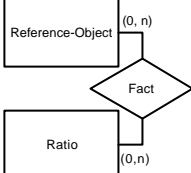
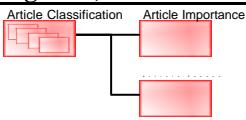
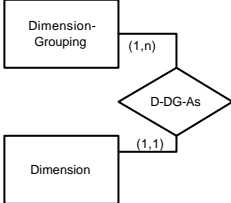
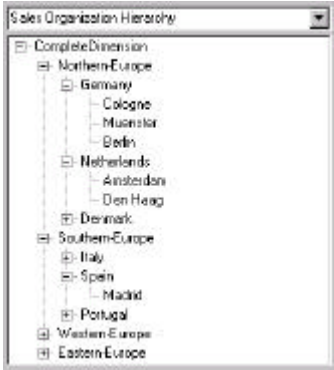
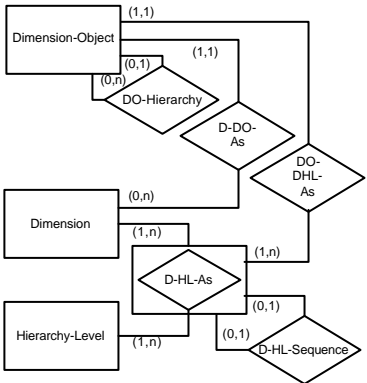
Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
Ratio-System	Subsumption: A ratio system is a set of ratios which enables the analysis of different meaningful aspects of a business 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).	

Table 1: Language concepts, linguistic actions and meta model components – conceptual modelling

Representation Element to create Model M1 (type level of Figure 1)	Language Concepts of language L1 used (type level of Figure 1; see Table 1)	Part of Meta Model M2 (meta level of Figure 1) instantiated while modeling (see Figure 3)
	Dimension, Dimension-Grouping, D-DG-As (Dimension Dimension-Grouping-Association)	
	Dimension-Object, DO-Hierarchy (Dimension-Object-Hierarchy), D-DO-As (Dimension Dimension-Object Association), Hierarchy Level, D-HL-As (Dimension Hierarchy-Level Association), D-HL-Sequence (Dimension Hierarchy-Level-Association Sequence), DO-DHL-As (Dimension-Object Dimension-Hierarchy-Level-Association Association)	

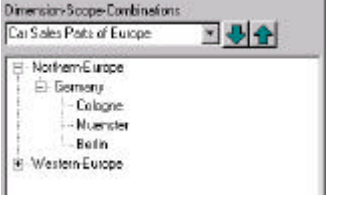
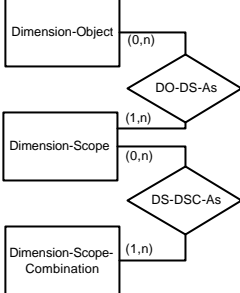
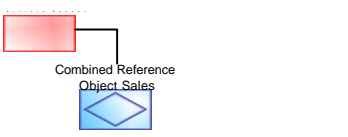
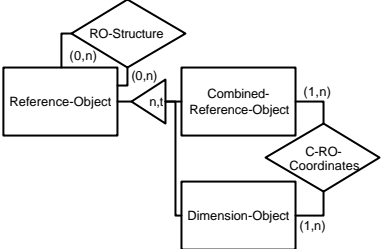
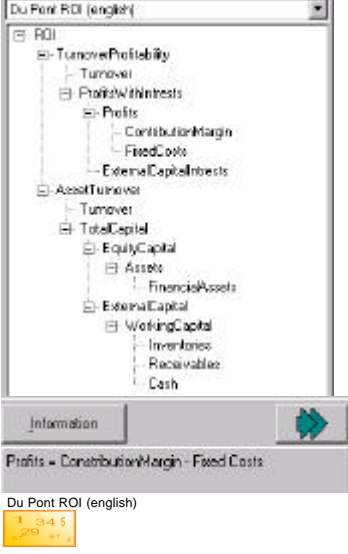
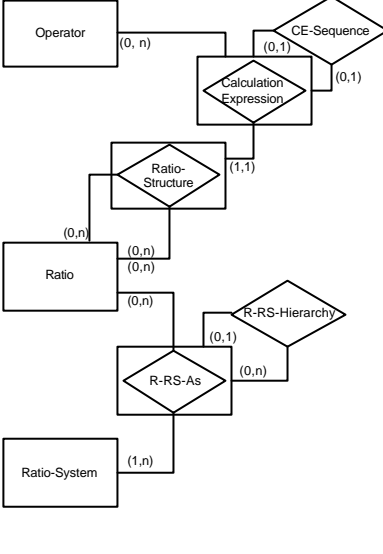
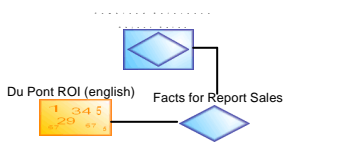
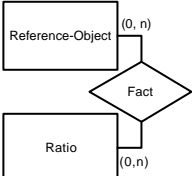
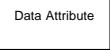
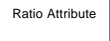
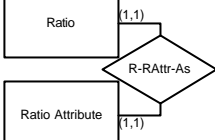
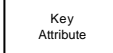
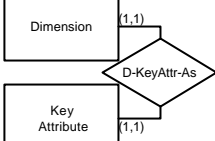
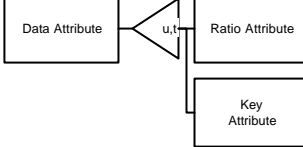
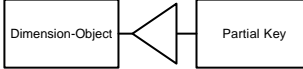
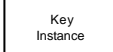
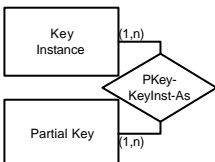
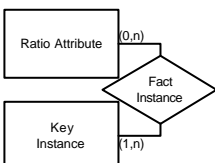
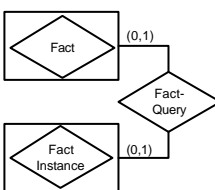
Representation Element to create Model M1 (type level of Figure 1)	Language Concepts of language L1 used (type level of Figure 1; see Table 1)	Part of Meta Model M2 (meta level of Figure 1) instantiated while modeling (see Figure 3)
	<p>Dimension-Scope, Dimension-Scope-Combination, DO-DS-As (Dimension-Object Dimension-Scope Association), DS-DSC-As (Dimension-Scope Dimension-Scope-Combination Association)</p>	
	<p>Reference-Object, Combined-Reference-Object, RO-Structure (Reference-Object Structure), C-RO-Coordinates (Combined Reference-Object Coordinates)</p>	
	<p>Ratio, Ratio-System, R-RS-As (Ratio Ratio-System Association), R-RS-Hierarchy (Ratio Ratio-System-Association Hierarchy), Ratio-Structure, Operator, Calculation Expression, CE-Sequence (Calculation-Expression Sequence)</p>	
	<p>Fact</p>	

Table 2: Representation Formalism or Information Warehouse Concepts

Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
Data Attribute	Subsumption: Used to structure data within a technical data base	
Ratio Attribute	Subsumption: Used to describe management relevant aspects of data objects stored in the	

Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
	data base.	
R-RAttr-As (Ratio Ratio-Attribute Association)	Composition: Relationship between concepts Ratio and Ratio Attribute. Every ratio leads to one unequivocal ratio attribute and every ratio attribute corresponds to exactly one ratio (cardinalities (1,1) on either side).	
Key Attribute	Subsumption: Used to identify data objects in the data base.	
D-KeyAttr-As (Dimension Key-Attribute Association)	Composition: Relationship between concepts Dimension and Key Attribute. Every dimension is interpreted as a key attribute and every key attribute is related to a dimension (cardinalities (1,1) on either side).	
Data Attribute, Ratio Attribute, Key Attribute	Subordination: Ratio attributes and key attributes are both data attributes but fulfill different functions.	
Partial Key	Subsumption and Subordination: The concept Partial Key is used to create keys identifying data objects in the database. Every identifier of a dimension object of the conceptual layer serves as partial key on the technical layer (specialization).	
Key Instance	Subsumption: Used to compose keys identifying data objects in a database.	
Pkey-KeyInst-As (Partial-Key Key-Instance Association)	Composition: Relationship between concepts Partial Key and Key Instance. Every partial key can be used to compose many key instances and a key instance is composed of one or more partial keys (cardinalities (1,n) on either side).	
Fact Instance	Composition: Relationship between concepts Key Instance and Ratio Attribute. One instance of a fact is identified within the database if one key and one ratio attribute is given. Every ratio attribute can be used to identify zero or many fact instances (cardinalities (0,n)). Every key is used to identify at least one or many fact instances (cardinalities (1,n)).	
Fact Query	Composition: Relationship between concepts Fact and Fact Instance. There is zero or one fact instances for every fact (cardinalities (0,1)) since there must not be a value for a fact but if a value is given it should be unequivocally given. On the other hand there is zero or one fact for any fact instance as well (cardinalities (0,1)) since there are fact instances possible which are not used to create reports to a certain time given.	


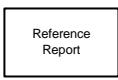
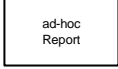
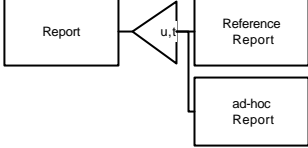
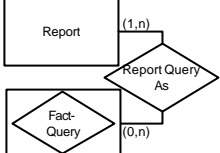
Conceptual Aspect of Language L1 (Type Level of Figure 1)	Linguistic action and meaning (Type Level of Figure 1)	Meta Model Component of Model M2 (Meta Level of Figure 1)
Report	Subsumption: Used to give information to a management user.	
Reference Report	Subsumption: Specialization of concept Report. Reference reports are predefined and have certain parameters to be specified when instantiated.	
ad-hoc Report	Subsumption: Specialization of concept Report. Ad-hoc reports are instantiated without having any parameters in advance..	
Report, Reference Report, ad-hoc Report	Subordination: Every reference report and every ad-hoc report is a member of concept Report (specialization symbol t). Every report belongs unequivocally to one of the two sub concepts (specialization symbol u).	
Report Query As (Report Query Association)	Composition: Relationship between concepts Report and Fact-Query. Every report is related to at least one fact query (cardinalities (1,n)). There might be fact queries not used in reports (cardinalities (0,n)).	

Table 3: Language concepts, linguistic actions and meta model components – technical modelling

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 Integrationschritte 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
- 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 Produktentwicklungsprozessen; 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.: 1997~~ 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 v. Uthmann, C.; Becker, J.; Brödner, P.; Maucher, I.; Rosemann, M.: PPS meets Workflow. Proceedings zum Workshop vom 9. Juni 1998.
- Nr. 65 Scheer, A.-W.; Rosemann, M.; Schütte, R. (Hrsg.): Integrationsmanagement, Januar 1999.
- Nr. 66 zur Mühlen, M.; Ehlers, L.: 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 – Instandhaltung eines FIS-Metamodells am Beispiel eines Einzelhandelsunternehmens, Mai 1999.
- Nr. 69 Holten, R.: Semantische Spezifikation Dispositiver Informationssysteme, Juli 1999.
- Nr. 70 zur Mühlen, M.: In Vorbereitung.
- 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.: In Vorbereitung.
- Nr. 75 Becker, J. (Ed.): Proceedings of the University Alliance Executive Directors Workshop – ECIS 2001.
- Nr. 76 Klein et. al. (Eds.): MOVE: Eine flexible Architektur zur Unterstützung des Außendienstes mit mobile devices. (In Vorbereitung.)
- Nr. 77 Knackstedt, R.; Holten, R.; Hansmann, H.; Neumann, S.: 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., Lechtenböcker, 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.