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. 67

A Framework for Information Warehouse Development Processes

Roland Holten

Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster,
Steinfurter Str. 107, 48149 Münster, Tel. (0251) 83-38100, Fax (0251) 83-38109
Mai 1999

Contents

Summary

Since the terms Data Warehouse and On-Line Analytical Processing were proposed by Inmon and Codd, Codd, Sally respectively the traditional ideas of creating information systems in support of management's decision became interesting again in theory and practice. Today information warehousing is a strategic market for any data base systems vendor. Nevertheless the theoretical discussions of this topic go back to the early years of the 20th century as far as management science and accounting theory are concerned and to the late 50s and early 60s when focussing information systems aspects. Although today efficient technology is available to de-

velop information systems for management's purposes information warehouse projects are still very risky and most of them are not finished successfully. The main reason for this situation lies in the lack of suitable languages to develop the conceptual specification of information warehouses. Based on this drawback methods for the development of information warehouses cannot be specified clearly. This paper proposes a language for the required conceptual specification of information warehouses based on an thorough analysis of management and accounting approaches to this topic and shows how this language can be integrated in a two dimensional development framework for information warehouse development as fundamental component.

1 Introduction and overview

Data Warehouse (DWH) and On-Line Analytical Processing (OLAP) are today's main buzz-words dominating the discussion concerned wirth information systems in support of managers' work. The term Data Warehouse was proposed and defined by INMON as "subject-oriented, integrated, nonvolatile and time-variant collection of data in support of management's decisions."[1] The core idea of data warehousing is to collect and aggregate required data with respect to the needs of management as intended group of users and independent of function oriented information systems architectures specialised in the support of operational business processes. DWH data are to be extracted and integrated when meaningful changes of operational data are recognised by monitors to become useful for the DWH. These transformation of operational data to DWH data is provided by components called extractors and integrators based on the work of monitoring components.[2] OLAP is intended to create views relevant to management's work on DWH data. The term OLAP was "invented" by CODD ET AL. and is defined as follows: " OLAP ... is the name given to the dynamic enterprise analysis required to create, manipulate, animate, and synthesize information from exegetical, contemplative, and formulaic data analysis models."[3] OLAP systems provide easy to use generators which enable the creation and usage of multi dimensional data bases as well as creating the effect of manipulating so called multi dimensional data structures or hypercubes.[4] Typically such systems offer manipulations called slicing, ranging or dicing, rotation and drill-down and roll-up.[5] The latter two are known from systems called executive information systems (EIS) since the early 80s.[6] These systems nevertheless were realised since the mid 60s.[7]

Concerning the development of information warehouses little work is done comprising the development process as a whole. Contributions usually focus on theoretical aspects concerning database theory and technical problems belonging to the late development phases like the implementation phase.[8] In some cases the development process is subject to the discussion the separation of development phases and the activities the phases are composed of are not based on a clear separation of conceptual specification of the system and the design of the system in the sense of programming in the large. So sometimes Entity-Relationship-Modelling is equated

---

1 Inmon (1996), p. 33. Cf. Inmon et al. (1997); Inmon, Hackathorn (1994); Chaudhuri, Dayal (1997); Ramakrishnan (1998); Silverston et al (1997); Widom (1995).

2 Cf. Inmon (1996), p. 78; Widom (1995).

3 Codd et al. (1993), p. 12.

4 Cf. Colliat (1996); Agarwal et al. (1996); Agrawal et al. (1997); Gyssens, Lakshmanan (1997).

5 Cf. OLAP Council (1997).

6 Cf. Rockart, Treacy (1982); Rockart, DeLong (1988).

7 Cf. Paller, Laska (1990), pp. 23-29.

8 This is true for the most of the contributions cited so far.

with the normalisation of the database scheme.[9] To enable a clear specification of the information warehouse in the sense of requirements engineering a domain specific language providing the concepts for requirements specification is needed.[10] As any development process the information warehouse development process has to start with a clear requirements specification as learned from different approaches to software engineering within the last twenty years.[11] Based on the conceptual or requirements specification the components the system is composed of can be specified in the sense of designing a system in the programming in the large phase. Only after providing the resulting documents the implementation can be done.

This paper firstly defines a two dimensional framework for development processes of information warehouses based on a separation of the development process dimension and the modelling or abstraction dimension. The framework then is used to arrange required development activities, the documents produced by these activities and the languages the documents are formulated in. As a conclusion it is shown how the framework can be used to clearly define a method of information warehouse development.

2 A process and abstraction oriented framework for information warehouse development

2.1 Dimension of development phases

It is widely accepted that any system development process has to start with the conceptual specification defining "what" the system under consideration should do.[12] This specification focuses on the domain the system has to work in. It thus has to specify clearly the domain specific requirements in a language providing domain specific concepts which enable this specification.

As a general rule the design or programming in the large phase is seen to be the second development phase.[13] The design has to specify the components the system consists of and the resulting system architecture. Typically decisions concerning the database model[14] and the user interface[15] are made in this phase. The architecture of the system describes the components of

---

9 Cf. Kimbal (1996), pp. 8.

10 Cf. Pohl (1996), pp. 4, 20, 34.; Jarke et al. (1993).

11 Cf. Boehm (1981); Davis et al. (1988); Weske et al. (1999).

12 Cf. McMenamin, Palmer (1984); Davis (1990); IEEE-830 (1984).

13 Cf. Balzert (1996a), pp.631; Nagl (1990).

14 Cf. Codd (1990); Date (1990); Embley (1998); Gupta, Horowitz (1991).

15 Cf. Balzert (1996b).

the system, the functions every component provides and the relationships between the components. The definition of component interfaces and the separation of component definition and its realisation are core principles of this phase.

The third phase – the implementation phase – deals with the realisation of the before defined components. Tasks in the implementation phase comprise the coding, the development of e.g. algorithms and data structures and the respective documentation of the documents.

Of course these phases are integrated in an evolutionary development process with jumps back to earlier phases if necessary. Additionally there are other phases like maintenance. But maintenance can be seen as another loop of the development process itself and thus another instantiation of the so far mentioned phases.[16] To sum it up for the purpose relevant here the construction process of the system can be explained by the three phases *conceptual specification*, *design or programming in the large* and *implementation*.

2.2 Dimension of model and abstraction levels

In every development phase discussed so far certain documents or models are produced as output and certain documents or models are required as input respectively. The second dimension of the framework is characterised by different levels of abstraction derived from science theory[17] and well established in software engineering.[18] Within the philosophical discipline of logic different levels of language are distinguished. A language which is subject to scientific analysis is called the *object language* where as the language used to carry out the analysis is called the *meta language*.[19] The attribute "meta" thus describes the role a certain language fulfils in a scientific process. This intention of levels of languages is applied to the process of modelling.[20] Languages, especially formalised languages, are used to create models. A certain language L1 used to create a model M1 can itself be expressed (concerning the syntax of the language) by model M2. Following the intention of logic mentioned above it can be said that M2 is a meta model with respect to the object modelled by model M1 (cf. Figure 1).[21] The same idea of defining different levels of abstraction resulting in a type-instance-relationship is

---

16 Cf. Nagl (1990), pp. 17-23.

17 Cf. Holten (1999), pp. 10-17.

18 Cf. Pohl (1996), pp.78-90; ISO/IEC 10027; ISO/IEC (1990); Jacobs, Holten (1995), pp. 98.

19 Cf. Kambartel (1996); Lorenz (Meta Language) (1996); Lorenz (Object Language) (1996).

20 Cf. Holten (1996), pp. 11.

21 Cf. Nissen et al. (1996), pp. 37; Holten (1996), pp. 11.

used in the ISO/IEC IRDS framework where these relationships lead to interlocking level pairs (cf. Figure 2).[22]

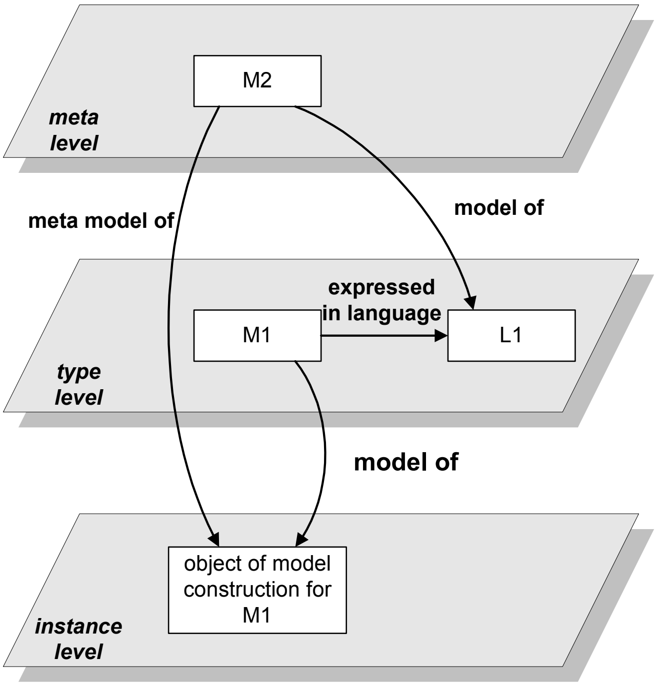

Figure 1: Levels of abstraction in the modelling process

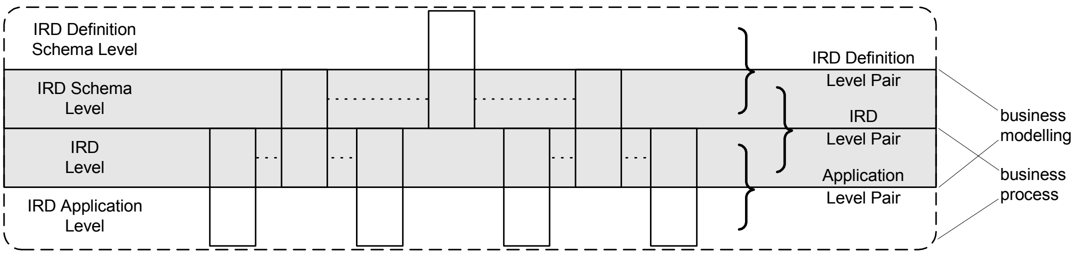

Figure 2: Levels of abstraction of the IRDS framework

Based on these considerations three levels of abstractions in the modelling process are derived for the framework (cf. Figure 1): The instance level comprises concrete data relevant for the domain, e.g. *sales of cars in region Europe in June 1999 realised with reselling partners which amount to 3 Million EUROs*. The next level 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 EURO for sales of cars in region Europe in June 1999 realised with reselling partners*. The concepts and terms used to define the models on the type level are defined on the next level of abstraction called the meta level. Concerning the example the mata level must de-

22 Cf. Jacobs, Holten (1995), pp. 98; Pohl (1996), pp. 80; ISO/IEC 10027; ISO/IEC (1990).

fine that there are *certain objects related to business decisions which can be combined with ratios* used in accounting theory (like sales and cost) to create new terms e.g. the concept of fact with a new semantics. Following the development methodology proposed by WEDEKIND the terms on the meta and the type levels have to be constructed using predefined operators.[23] Additionally the terms used to define the model on the type level must be instances of the terms provided on the meta level. The same holds for the relation between objects on the instance level and objects on the type level. It should be mentioned that the IRDS framework consists of a fourth abstraction level which is not of interest here.

2.3 Combining the dimensions: Information Warehouse Development Framework

Since any development phase produces certain documents and any of these documents can be characterised as a model it is possible to combine the above derived dimensions to a framework of information warehouse development processes. Each of the two dimensions is characterised by three entities. So the combination of the dimensions leads to a framework with nine entities each characterised by a specific combination of entities belonging to the respective dimensions (cf. Figure 3).

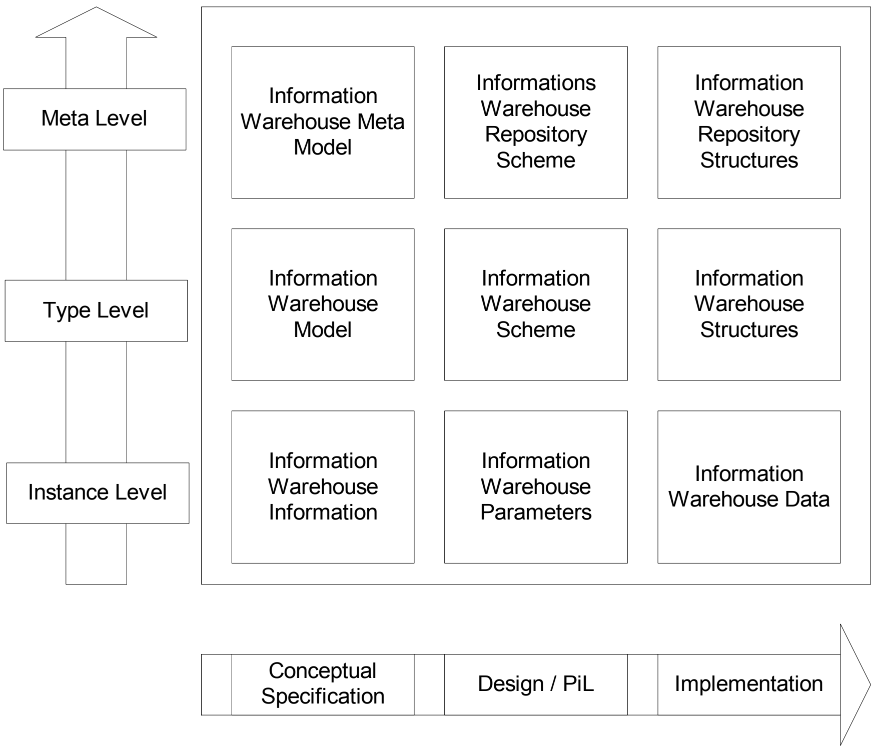

Figure 3: Information Warehouse Development Framework

[23] Cf. Wedekind (1981); Wedekind (1979); Wedekind, Ortner (1977); Holten (1999), pp. 19-27.

Every abstraction level in the framework describes the development of certain information objects from the conceptual specification to their formalised 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. Finally the implementation level describes the process of formalising information produced in the business process and handled by the information warehouse. In relation to the IRDS framework the levels concerned with business processes and business modelling are of interest here (cf. Figure 2).[24] The respective level pairs of the IRDS framework were characterised as “application environment” by Pohl.[25]

Every box of the information warehouse development framework is characterised by a unique combination of documents and activities to be performed. There are dependencies of development activities in one box to other boxes’ documents. The framework is intended to clearly organise these dependencies and thus serving as a means to clearly describe the process of information warehouse development. The contents of every box is outlined in the next session.

3 Content of framework and discussion

Information Warehouse Meta Model

To construct the meta model which defines the required concepts to model the information warehouse from a domain perspective in the conceptual specification phase we follow the methodology developed by WEDEKIND.[26] The meta model is represented as ERM where cardinalities are defined in (min, max)-notation. The construction of the required language concepts is based on thorough analyses of management and accounting theory since these fields have a long tradition in dealing with information warehouse problems from a domain perspective. In fact contributions can be found in which the ideas of OLAP and DWH were developed in the 50s to 70s of the 20th century with basic research work reaching back even to the beginning of the century. The integration of information in a separate information basis for management’s purposes and the analogy to multidimensional cubes (nowadays called hyper cubes) combining management’s views on this information which are characteristics of the Data Warehouse and the OLAP concepts were developed by the German researchers RIEBEL and SCHMALENBACH as basic calculation (Grundrechnung) and evaluation calculation (Auswertungsrechnung) in the middle of the 20th century.[27] This research was based on ideas discussed by RIEGER in the 20s

---

24 Cf. Jacobs, Holten (1995), pp. 98.

25 Cf. Pohl (1996), pp. 80-83.

26 Cf. Wedekind (1981); Wedekind (1979); Wedekind, Ortner (1977); Holten (1999), pp. 19-27.

27 Cf. Riebel (1979a); Riebel (1979b); Schmalenbach (1963); Schmalenbach (1948); Riebel (1992).

of the 20th century in the German literature and by the American researcher GOETZ in the first half of the century.[28] These theories provide an adequate basis to derive the language constructs required to model information warehouses from the domain perspective. This language is the core of the box “information warehouse meta model” of the respective development framework. Its construction is outlined below.[29]

The first concept required is *Reference Object* leading to the respective entity type (cf. Figure 4). Reference Objects are defined as all “measures, processes and states of affairs which can be object to arrangements or examinations on their own.”[30] A Reference Object is everything related to a decision in a business process. An example for an instance (*RO1*) of a Reference Object is the earlier mentioned *set of cars in region Europe in June 1999 which are sold to reselling partners*. Reference Objects are related to each other within networks leading to the relationship type *RO-Structure*. This relationship type is recursively related to the entity type Reference Object. Its cardinalities (0,n):(0,n) are characteristic for any structure or network since they define that any node may have zero or many higher nodes and zero or many subordinate nodes. In the above example the Reference Object (*RO2*) *set of cars in region World in June 1999 which are sold to reselling partners* is logically higher than RO1 according to the dimension *region* and the Reference Object (*RO3*) *set of cars in region Europe in Year 1999 which are sold to reselling partners* is logically higher than RO1 according to the dimension *time-month*. Reference Objects can be identified by elements defining the instances of their dimensions. They are specialised to *Combined Reference Objects* and *Dimension Objects*. This specialisation is not unequivocal but total (symbols n,t) since every Reference Object is a Combined Reference Object and some Reference Object are characterised as Dimension Objects. Dimension Objects serve as coordinates for Combined Reference Objects indicating that Reference Objects can be thought of as vectors. Every Reference Object thus can be identified by specifying one Dimension Object for every of its dimensions. This is expressed by the relationship type *C-RO-Coordinates*. Every Dimension Object depends existentially on one unequivocal dimension. This is modelled by the respective entity types *Dimension* and *Dimension Object* and the relationship type *D-DO-Ass* indicating the association between the respective entity types. The relationship type is characterised by cardinalities (1,1):(0,n) read from Dimension Object to Dimension defining that every Dimension Object needs at least and at most one related Dimension (existential dependence).

---

28 CF. Rieger (1928); Goetz (1949). A detailed discussion of these approaches and their connections to OLAP and Data Warehouse can be found in Holten (1999), pp.73-115.

29 A detailed discussion of this construction process is provided by Holten (1999), pp. 73-115 and Becker, Holten (1998), pp.483-488.

30 Riebel (1979b), p. 869.

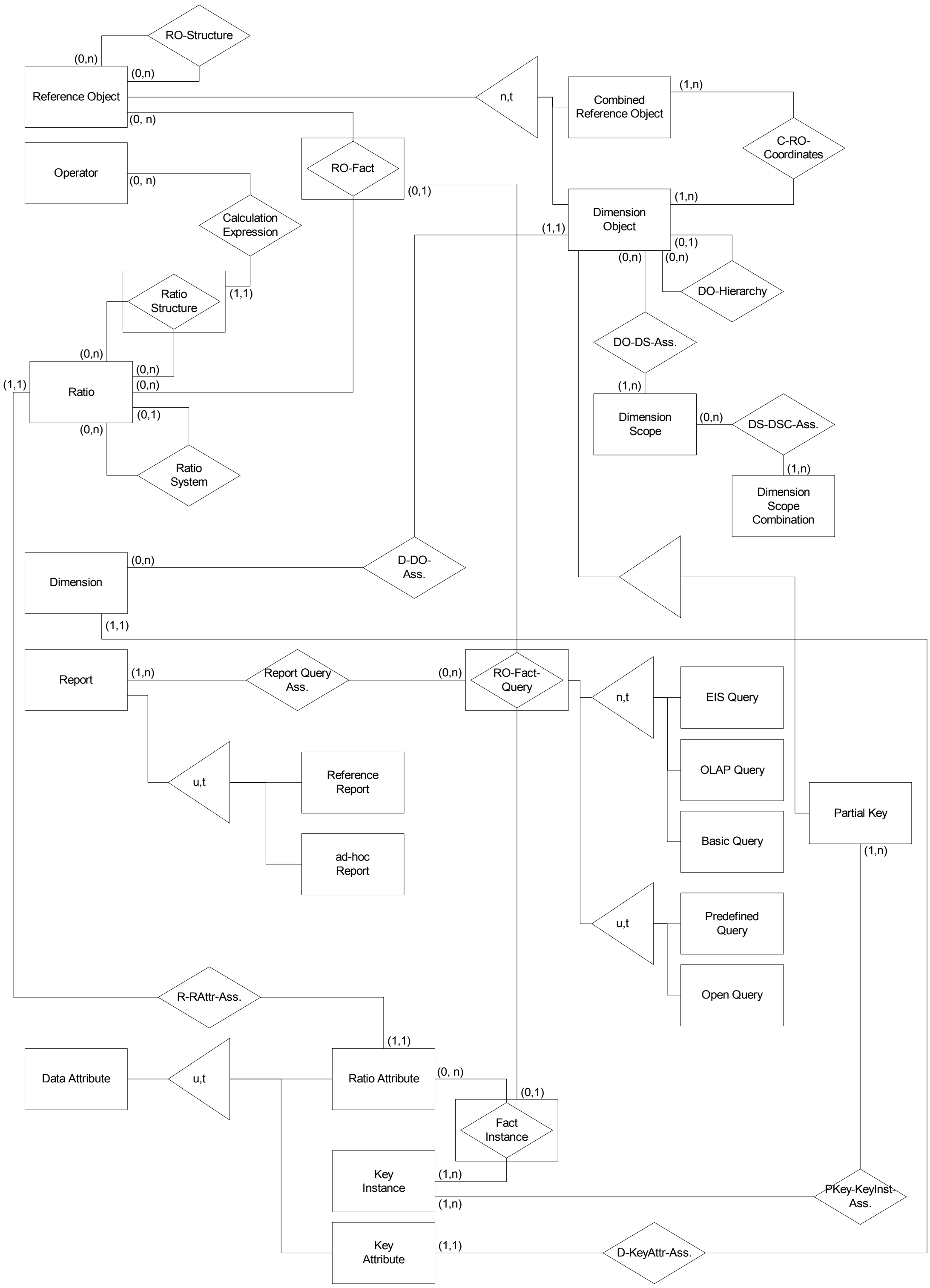

Figure 4: Information Warehouse Meta Model

Dimension Objects of one Dimension are characterised by stronger relationships between each other compared to other Reference Objects. These strong relationships only exist because of domain specific reasons and are created by a modeller. They lead to hierarchies of Dimension Objects like product hierarchies and hierarchies of regions. This concept is modelled by the relationship type *DO-Hierarchy* which defines a recursion on the entity type Dimension Object. The relationship type is defined by cardinalities (0,1):(0,n) read from the subordinate node to the higher node since in a hierarchy every node has at most one related higher node or none and can have zero or many subordinated nodes. Every element of DO-Hierarchy thus expresses one unique father son relationship (e.g. (regions Europe, region World)). The relationship type C-RO-Coordinates allows the specification of the space of all Reference Objects. The specification of scopes within this space of Reference Objects, which is required for the modelling of any hyper cube in the sense of OLAP, can be expressed using the entity types *Dimension Scope* and *Dimension Scope Combination* with the respective relationship types *DO-DS-Ass* and *DS-DSC-Ass* which allow the combination of subsets of the set of Dimension Objects.

The next concept is *Ratio* which is of fundamental importance for specifying information in management processes. Ratios define important aspects of Reference Objects like gross margin. They are clearly defined in the sense of a management view. Typically Ratios are organised in hierarchies to enable the top down analysis of one unique Reference Object according to different aspects. This is expressed by the recursive relationship type *Ratio System* which is characterised by cardinalities (0,1):(0,n) defining a hierarchy. To be able to express calculation schemes for Ratios (like: gross margin = sales revenues – direct cost) the concepts *Ratio Structure* and *Operator* connected by the relationship type *Calculation Expression* are introduced. These constructs allow the definition of post order expressions for the calculation of Ratios (e.g. the entire DuPont-Pyramid can be expressed this way). The relationship between Ratio and Reference Object leads to a new concept which goes beyond the characteristics of the two former defined concepts. This concept is called *RO-Fact* indicating that aspects defined by Ratios define Facts of Reference Objects which are relevant to management's decisions.

The concepts constructed so far allow to express relevant aspects (Ratios) of business (Reference Objects) and their combination (Facts). Facts clearly define a type of information. In the above example *the gross margin of the set of cars sold in region Europe in June 1999 to reselling partners* clearly is an instance of the concept Fact and exactly defines a type of information (and thus a section of a respective report) a sales manager could be interested in. But this Fact does not define the concrete value (the *Fact Instance*) realised in the real business process. Thus additional concepts are required to bring together the definition of Facts from the domain perspective and the Fact Instances stored in a technical device like the information base of an information or data warehouse. To describe the information base the concepts *Data Attribute*, *Ratio Attribute*, *Key Attribute* and *Key Instance* are introduced. They allow the description of a table oriented structure following the idea of a relational structure. Data Attributes are unequivocally and totally specialised in Ratio Attributes and Key Attributes (symbols u,t). Key Attributes are unequivocally related (associated) to Dimensions (relationship type *D-Key-Attr-Ass* with cardinalities (1,1):(1,1)) and Ratio Attributes are respectively associated to Ratios (relationship type *R-RAttr-Ass* with cardinalities (1,1):(1,1)). Every Dimension Object is interpreted as *Partial Key* (see respective specialisation) and Partial Keys are combined to *Key In-*

stances (relationship type *Pkey-KeyInst-Ass* with cardinalities (1,n):(1,n) indicating that every Key instance must consist of one or more Partial Keys and every Partial Key becomes at least part of one Key Instance). Now *Fact Instances* can be identified by combining Ratio Attributes with Key Instances with the respective relationship type characterised by cardinalities (1,n):(0,n). That is there can be no Key Instance without a Fact Instance. Now it can be expressed that RO-Facts (as a type of information) are related to their instances (Fact Instances) by a new concept given the name *RO-Fact-Query* and modelled by the respective relationship type. The cardinalities (0,1):(0,1) of this relationship type define that there is one unequivocal Fact Instance for every RO-Fact if it does exist. Fact Queries are specialised in several classes. Finally Fact Queries are used to define *Reports*. Every Report consist of at least one Fact Query (relationship type *Report Query Ass*) and is specialised in the classes *Reference Report* and *ad-hoc Report*.

To clearly define the usage of the meta model algorithmic specifications are required. The respective algorithms must be understandable in the conceptual phase and on the meta level and thus in the information warehouse meta model box of the framework. To become clear on this abstraction level an algorithm must be formulated using the concepts defined so far and using some instructions with a meaning in everyday usage (e.g. successor, mark, sequence of a certain type of objects). That is algorithms must be clearly defined as action requests.[31] To define these conceptual algorithms a notation is used which is similar to pseudo code an known from the definition of mini specifications.[32] It contains notation concepts used as well in structured programming.[33] Its basic notational elements are *sequence of instructions, comment, for all loop* and *if condition* (cf. Figure 5). The conceptual algorithms are defined as follows:

```
Procedure Name(List of Parameters)
{Comment}
    For all loop:
        Instruction.
        Instruction.
    If condition:
        Instruction.
        Instruction.
```

Figure 5: Notational Conventions for Conceptual Algorithms

31 Cf. Wedekind (1981), pp. 140-144.
32 This notation is proposed in Holten (1999), pp. 232.
33 Cf. Nassi, Schneiderman (1973); Dijkstra (1967).

As an example it is shown how the algorithm which calculates the set of all *Combined Reference Objects* based on the set of *Dimensions* and *Dimension Objects* can be formulated using some assistant procedures (cf. Figure 6).

Assistant Procedures:

```
Write Coordinate(Coordinate Position, Coordinate Value, Combined Reference Object)
{writes value of Coordinate to Coordinate Position of the Combined Reference Object }.

Write Combined Reference Object(Sequence of Dimension, Dimension Objects, Combined Reference Object)
{writes all Coordinates of the Combined Reference Object}
    For all Dimensions:
        Search marked Dimension Object.
        Write Coordinate(Dimension, marked Dimension Object, Combined Reference Object).

Create Combined Reference Object for last Dimension (Sequence of Dimensions, Dimension)
{creates all possible instances of concept Combined Reference Object for all Dimension Objects of last Dimension of Sequence of Dimensions with marked Dimension Objects in other Dimensions}
    For all Dimension Objects of Dimension:
        Mark Dimension Object.
        Create new instance of concept Combined Reference Object.
        Write Combined Reference Object (Sequence of Dimensions, marked Dimension Object, Combined Reference Object).
```

Algorithm / Main Procedure:

```
Create all Combined Reference Objects (Sequence of Dimensions, Dimension)
{creates as an recursive procedure all possible instances of concept Combined Reference Object }
    If last Dimension of sequence:
```

Create Combined Reference Object for last Dimension (Sequence of Dimensions, Dimension)

If not last Dimension of sequence:

  Create all Combined Reference Objects (Sequence of Dimensions, successor of marked Dimension).

  If not last Dimension Object of marked Dimension:

    Mark successor of marked Dimension Object of marked Dimension.

    Create all Combined Reference Objects (Sequence of Dimensions, marked Dimension).

<u>Integration of Algorithm in Main Program:</u>

Create (any but stable) sequence of Dimensions.

Mark first Dimension.

For all Dimensions:

  Create (any but stable) sequence of Dimension Objects.

  Mark first Dimension Object.

Create all Combined Reference Objects (Sequence of Dimensions, first Dimension).

Figure 6: Recursive Algorithm for Calculation of set of Combined Reference Objects

A runtime example of the above algorithm is shown in Figure 7.

A1: **A**BC, a1b1c1 (Start)
      A1: A**B**C, a1b1c1
            A1: AB**C**, a1b1c1
                        -> a1b1c1 (created Combined Reference Object)
                        -> a1b1c2
      A1: A**B**C, a1b2c1
            A1: AB**C**, a1b2c1
                        -> a1b2c1
                        -> a1b2c2
A1: **A**BC, a2b1c1
      A1: A**B**C, a2b1c1
            A1: AB**C**, a2b1c1
                        -> a2b1c1
                        -> a2b1c2

```
A1: ABC, a2b2c1
     A1: ABC, a2b2c1
               -> a2b2c1
               -> a2b2c2.
```

Figure 7: Runtime Example

Information Warehouse Repository Scheme

This box of the information warehouse development framework contains an architecture of the information warehouse repository (cf. the simplified architecture in Figure 8) since it defines the documents of the design phase on the meta level. The repository additionally is based on a database scheme defined in this development phase too. This database scheme is derived from the former constructed meta model. Examples for required relations (provided the repository is realised on a relational database as platform) are e.g. *Relation "Dimensions"(Dims: Id; Dim-Name: Char)* defining the respective relation with required attributes identifyer and name and *"Dimension Objects" (DO: Id; DOName: Char; DO-Father: Id)* defining the recursive structure according to the relationship type DO-Hierarchy (cf. Figure 4).

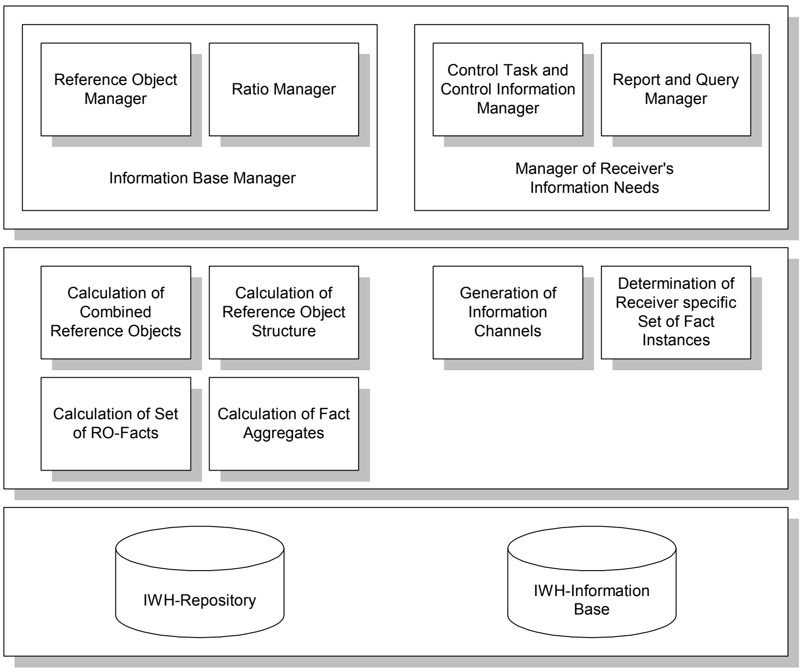

Figure 8: Three Tier Architecture of Information Warehouse Repository

Information Warehouse Repository Structures

The implementation on the meta level is concerned with defining and programming e.g. the required data structures for the repository. In our case SQL could be used to create the needed tables. Additionally the modules defined in the architecture must be implemented (e.g. as

JAVA applets). The repository defined so far can now be used to model and implement an information warehouse based on the language defined above.

Information Warehouse Model

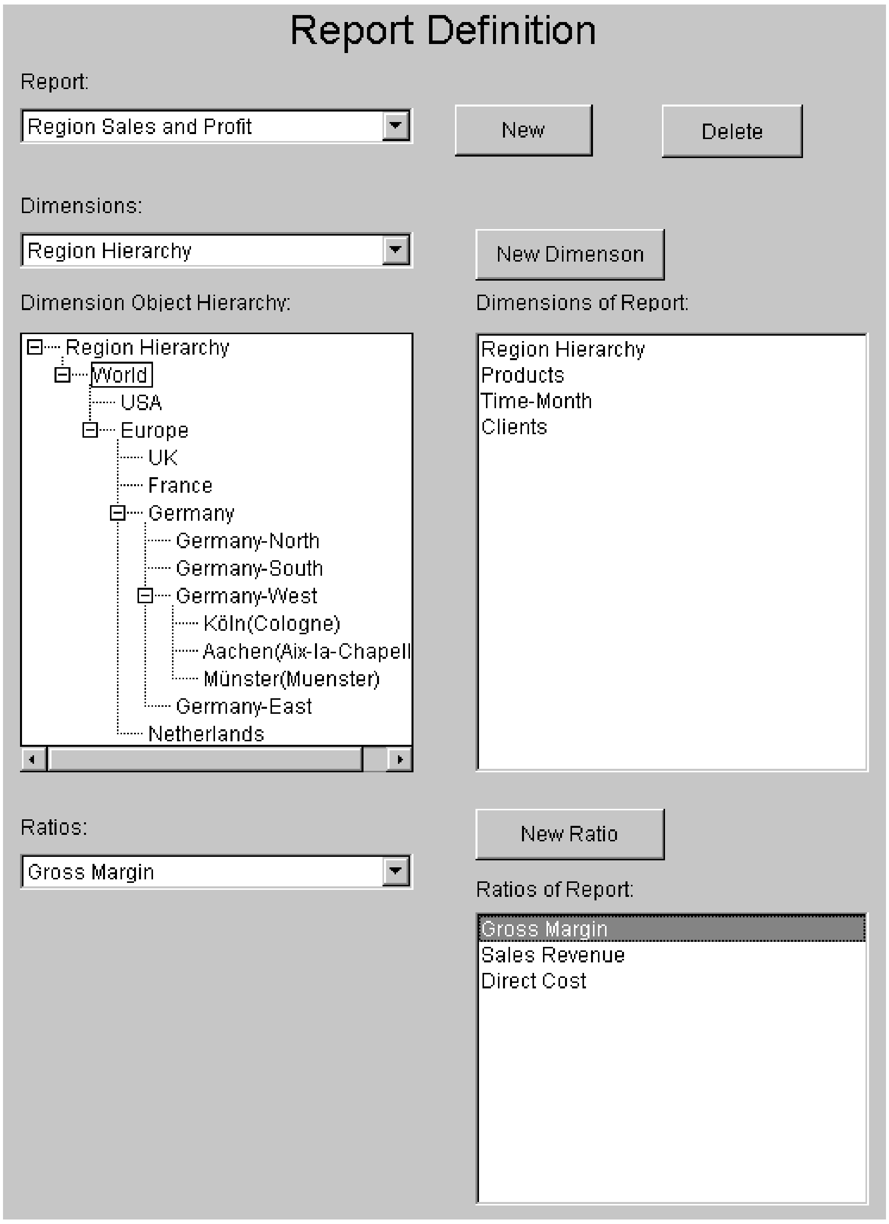

Figure 9: Tool based definition of Information Warehouse Concept

The first task of developing the information warehouse itself is to model conceptually the requirements and to structure these requirements using a domain specific language. The language defined above and the repository developed so far provide adequate tools to support these activities (cf. Figure 9). In the example the required conceptual information to define a report is specified. Using the information warehouse repository tool the concepts *Dimension, Dimension Object, Dimension Object Hierarchy, Ratio* and *Report* (cf. Figure 4) are instantiated. Based on the conceptual algorithms (e.g. the example to calculate all required Combined Reference Objects in Figure 6) this information is sufficient to create the required *Facts, Fact Queries* and *Data Attributes* for the implementation of the information warehouse scheme and parts of the information base.

Information Warehouse Scheme

Provided the architecture of the information warehouse should be based on a relational database system the required relations can be derived from the conceptual specification of the information warehouse. In the example a scheme similar to the star scheme should be implemented. So there are relations required for *Fact Instances* and e.g. dimension tables for *Regions, Products* and *Clients*. The number of *Key Attributes* of the fact table can be calculated directly based on the conceptual information defining the required *Dimensions*. This calculation which is clearly part of the design phase of the information warehouse development process can be done automatically since the meta model defines the (1,1):(1,1)-relationship between Dimensions of the conceptual scheme and Key Attributes of the database scheme (cf. Figure 4). Thus (as is shown by this example) the conceptual model of the information warehouse in conjunction with the meta model it is an instance of implicitly defines parts of the information warehouse scheme which must be made explicit in the respective design phase.

Information Warehouse Structure

The last phase on the type level deals with the implementation of the required information or data structures of the information warehouse. Here the instances of the meta model specified in the conceptual model are used as input for schema generators which can create e.g. the required DDL SQL-statements for the warehouse implementation (provided the warehouse is implemented on a relational platform). In addition it is possible to generate parts of the information base itself since based on the meta model, the algorithms (e.g. the one in Figure 6) and the conceptual model (defining the instances of meta model concepts) the maximum set of Key Instances and the maximum set of Ratio Attributes are defined.

Information Warehouse Information

In this box of the framework *multidimensional* information is generated within the business processes of the enterprise. Multidimensional information can be thought of as instances of the concept *Fact Instance* defined in the meta model. In the above example *sales of cars in region Europe in June 1999 realised with reselling partners which amount to 3 Million EUROs* are “one piece of” multidimensional information with relevance to a certain sales manager.

Information Warehouse Parameters

On the instance level the design phase can be reduced to the association of information warehouse information to parts of the information warehouse architecture. In our example where the information base is realised using a relational platform the mentioned piece of multidimensional information (sales of cars in region Europe in June 1999 realised with reselling partners which amount to 3 Million EUROs) is e.g. associated to the fact table with the respective associations of Key Instance and Ratio Attributes.

Information Warehouse Data

Finally any information must be calculated and stored as structured data in a database. In this box of the framework the information warehouse is in use. Any instance of the concept Fact Instance is booked into the information base (for this purpose components of data warehouse like monitors, extractors and integrators are required) and aggregates are calculated using programs which are instances of the conceptual algorithms defined on the meta level. Additionally these programs use parameters which are defined on the type level during the conceptual specification phase of the information warehouse development process.

4 Derived development method for information warehouses and conclusion

Based on the defined meta model and the conceptual algorithms together with the prototypical implementation of the respective tool a method for the development of information warehouses can be derived. The intention of this approach is similar to the CASE-idea. The required development steps are as follows:[34]

- Model the Dimensions, Dimension Objects and Dimension Object Hierarchies which are required from the management's perspective. These structures may differ completely from structures used in other applications supporting the operational business processes. Thus an integrated model for the information warehouse concept is required.

- Generate the set of Combined Reference Objects using the defined algorithms automatically.

- Specify the required Ratios with their Calculation Expressions and the required Ratio Systems.

---

34 Cf. for a detailed discussion Holten (1999).

- Generate the maximum set of RO-Facts or the required scopes of this set based on the instances of the concept Dimension Scope Combination.

- Define the required Reports.

- Generate the required information base structure (instances of the concept Fact Instance).

- Generate the required instances of the concept Fact Query.

In real projects it has shown that a the communication with business managers can be facilitated by using direct modelling approaches like the one shown in Figure 9 in combination with ERM-like notations for the definition of Dimensions, Combined Reference Objects and Facts on a type level where the ERM is enriched by examples of instances of the respective entity and relationship types (cf. Figure 10).

For the development steps concerned with the generation of parts of the information warehouse (structure as well as information base) bridge programs are required which enable the integration of Data Warehouse and OLAP-Systems available on the market into the demonstrated approach. These programs are currently developed as prototypes for selected systems like SAP R/3 and MicroStrategy.

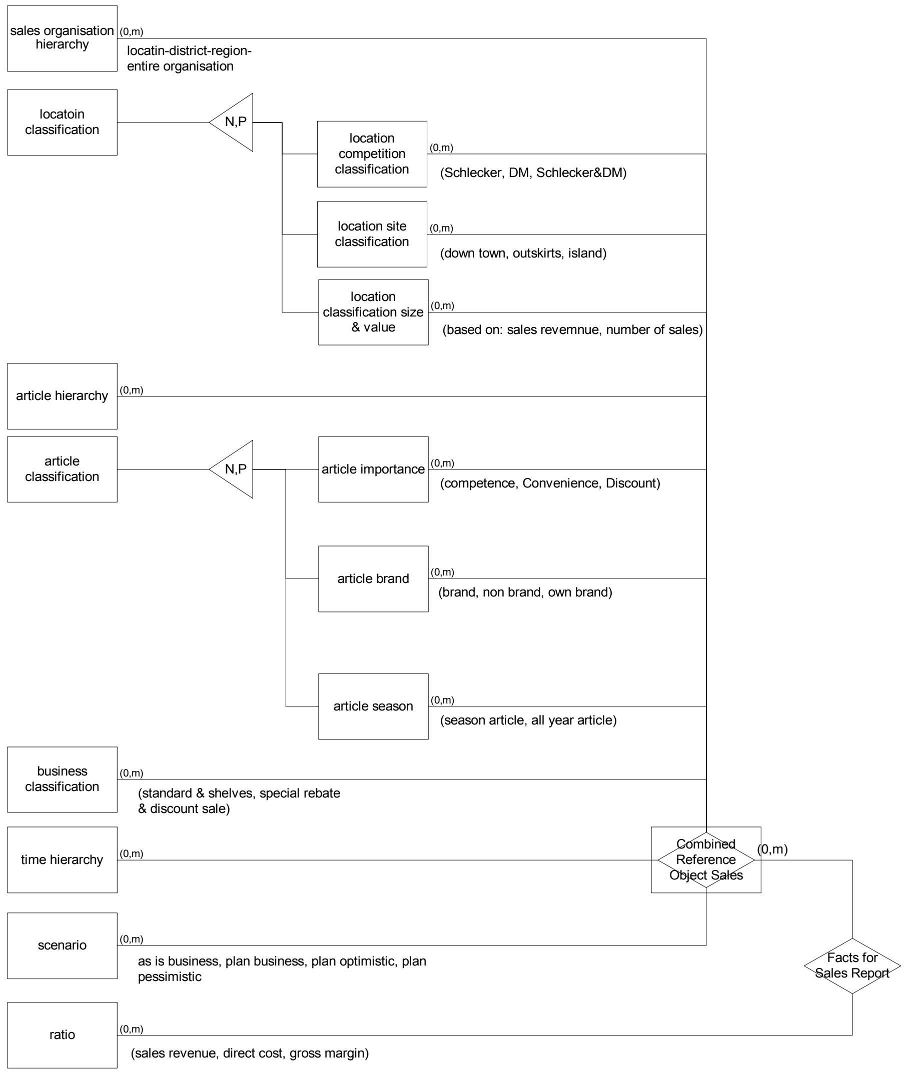

Figure 10: Example of enriched ERM as part of Information Warehouse Model

References

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

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

Balzert, H. (1996a): Textbook of Software Technique. Software Development. Heidelberg et al. 1996. (in German)

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

Becker, J.; Holten, R.: Conceptual Specification of Management Information Systems (Fachkonzeptuelle Spezifikation von Führungsinformationssystemen). WIRTSCHAFTSINFORMATIK 40 (1998) 6, pp. 483-492. (in German)

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

Budde, R.: Prototyping. An approach to evolutionary systems development. Springer 1992.

Bulos, D.: A New Dimension. In: Database Programming & Design, 9 (1996) 6, S. 33-37.

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

Chen, M.: A Model-Driven Approach to Accessing Managerial Information: The Development of a Repository-Based Executive Information System. In: Journal of Management Information Systems, 11 (1995) 4, S. 33-63.

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.

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.

Dijkstra, E. W.: A Discipline of Programming. Englewood Cliffs 1967.

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

Goetz, B. E.: Management Planning and Control. A managerial approach to industrial accounting. 5. Aufl. New York et al. 1949.

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.: Development of Management Information Systems. A methodical Approach. Wiesbaden 1999. (to be published in June 1999) (in German)

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.; Glassey, K. L.: Managing the Data Warehouse. New York et al. 1997.

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

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

Jarke, M.; Pohl, K.; Jacobs, S.; Bubenko, J.; Assenova, P.; Holm, P.; Wangler, B.; Rolland, C.; Plihon, V.; Schmitt, R.; Sutcliffe, A.; Jones, S.; Maiden, N, Till, D.; Vassiliou, 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.

Kambartel, F.: Stufe (Level) In: Mittelstraß (Ed.): Enzyklopädie Philosophie und Wissenschaftstheorie, Band 4. (Encyclopedia Philosophy and Science Theory, Volume 4) Stuttgart, Weimar 1996, p. 116. (in German)

Kimbal, R.: The Data Warehouse Toolkit. Wiley, New York et al. 1996.

Lorenz, K. (Meta Language): Metasprache. In: J. Mittelstraß (Ed.): Enzyklopädie Philosophie und Wissenschaftstheorie, Band 2. (Encyclopedia Philosophy and Science Theory, Volume 2) Stuttgart, Weimar 1995, p. 875. (in German)

Lorenz, K. (Object Language): Objektsprache. In: J. Mittelstraß (Ed.): Enzyklopädie Philosophie und Wissenschaftstheorie, Band 2. (Encyclopedia Philosophy and Science Theory, Volume 2) Stuttgart, Weimar 1995, pp. 1054-1055. (in German)

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

Nagl, M.: Software Technique. Methodical Programming in the Large. Berlin, Heidelberg 1990. (in German)

Nassi, I.; Shneiderman, B.: Flowchart Techniques for Structured Programming. ACM SIGPLAN Notices, 8 (1973) 8, S. 12-26.

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.

Paller, A.; Laska, R.: The EIS-Book. Information Systems for Top-Managers, Homewood, Ill. 1990.

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

Ramakrishnan, R.: Database Management Systems. Boston, MA: WCB/McGraw-Hill 1998.

Riebel, P.: Direct Proceeds, Direct Cost and Coverage Accounting as core of Management Accounting (Einzelerlös-, Einzelkosten- und Deckungsbeitragsrechnung als Kern einer ganzheitlichen Führungsrechnung). In: Männel, W. (Ed.): Handbook of Cost Accounting (Handbuch Kostenrechnung). Gabler, Wiesbaden 1992, S.247-299. (in German)

Riebel, P.: On Basic Calculations (Zum Konzept einer zweckneutralen Grundrechnung). In: ZfbF 31 (1979), S. 785-798. (in German)

Riebel, P.: Designproblems of Basic Calculations (Gestaltungsprobleme einer zweckneutralen Grundrechnung). In: ZfbF 31 (1979), S. 863-893. (in German)

Rieger, W.: Introduction to Private Economy (Einführung in die Privatwirtschaftslehre). Nürnberg 1928 (reprint: Nürnberg 1959). (in German)

Rockart, J. F.; De Long, D. W.: Executive Support Systems: The Emergence of Top Management Computer Use, Homewood, Ill. 1988.

Rockart, J. F.; Treacy, M. E.: The CEO goes On-Line. HBR 60 (1982). reprint: H. J. Watson; R. K. Rainer; G. Houdeshel (Hrsg.), Executive Information Systems: Emergence Development Impact. New York et al. 1992. S. 3-12.

Schmalenbach, E.: Cost Accounting and Price Policy (Kostenrechnung und Preispolitik). 8th ed.., Westdeutscher Verlag, Köln, Opladen 1963. (in German)

Schmalenbach, E.: Financial Steering of Ecomomy. Volume 1 and 2 (Pretiale Wirtschaftslenkung. Band 1 und 2). Industrie- und Handelsverlag, Bremen 1948. (in German)

Silverston, L.; Inmon, W. H.; Graziano, K.: The Data Model Resource Book. New York: Jhon Wiley & Sons 1997.

Wedekind, H.: Database Systems I. A constructive Introduction to Data Processing in Commerce and Administration (Datenbanksysteme I. Eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung). 2nd Ed., BI Wissenschaftsverlag, Mannheim et al. 1981. (in German)

Wedekind, H.: Object Type Method to Database Design (Die Objekttypen-Methode beim Datenbankentwurf - dargestellt am Beispiel von Buchungs- und Abrechnungssystemen). ZfB (Zeitschrift für Betriebswirtschaft) 49 (1979) 5, pp. 367-387. (in German)

Wedekind, H.; Ortner, E.: The Desing of a Database for Accounting (Der Aufbau einer Datenbank für die Kostenrechnung). Die Betriebswirtschaft 37 (1977) 4, pp. 533-542. (in German)

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.

Arbeitsberichte des Instituts für Wirtschaftsinformatik

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 der 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, T.; Püttmann, M.: Konzeption und prototypische Realisierung eines Informationssystems für das Prozeßmonitoring und -controlling; September 1996.

Nr. 50 von Uthmann, C., Turowski, K. unter Mitarbeit von Rehfeldt, M.; Skall, M.: Workflow-basierte Geschäftsprozeßregelung als Konzept für das Management industrieller Produktentwicklungsprozesse; Oktober 1996.

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

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

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

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

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

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

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

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

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

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

Nr. 61 Speck, M.: In Vorbereitung.

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

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

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

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

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

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