IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP5124

# Configurative Reference Model-Based Development of Data Warehouse Systems

Ralf Knackstedt and Karsten Klose
European Research Center for Information Systems (ERCIS), University of Münster,
Leonardo-Campus 3, 48149 Muenster, Germany, {ralf.knackstedt, karsten.klose@ercis.de}

## ABSTRACT

Developing Data Warehouse Systems requires specifications of the underlying business need in the form of information models. The development of information models is often both expensive and extensive. Against this background, reference models provide useful means to reduce the costs of information modelling, because they can be used as a starting point for the construction of project-specific information models. However, reference models only provide benefits if the reduced effort resulting from their application is not overlapped by the adaptation costs. In this context, configurable reference models comprise rules which allow modifications of the original reference model depending on company or project individual determinations of configuration parameters. This paper provides concepts for a reference model-based Data Warehouse System development. Extensions of multidimensional modelling techniques are proposed which allow for configuring reference models. Moreover, influences of these reference models on the Data Warehouse engineering process are discussed and an appropriate tool support is presented.

## MOTIVATION

From a business perspective, developing Data Warehouse Systems is an integrative task of *organisational* and *application system design*. Within the organisational scope, it is necessary to identify what kind of data has to be provided to whom (decision maker) for what kind of management decision (Holten 2003). Within the application system scope, it is necessary to implement the underlying data basis whereupon this task comprises selection and configuration of relevant Software and Hardware components for the Data Warehouse System architecture (Inmon, Hackathorn 1994). In this context, information models foster the structuring of the two design areas (Karami 1988; Kotterman, Konsynski 1984). Depending on their technical relationship, information models are e. g. divided into the phases *requirements definition*, *design specification*, and *implementation* (e. *g.* Scheer 1992). Requirements definitions specify original business demands as they describe "what" the system under consideration should do (Pohl 1996; McMenamin, Palmer 1984; Davis 1990). Based on these definitions, the design specification phase develops suitable data base schemas, selects adequate system components, determines partitions of data base tables and specifies extraction, transformation and loading (ETL) processes (Boehm 1981; Davis , Bersoff, Comer 1988; Codd 1990). Within the implementation phase the previously defined components are realized. Here, tasks comprise the development of algorithms, data structures, and coding (Boehm 1981).

The development of information models is often both expensive and extensive. Despite the fact that an appropriate specification of Data Warehouse Systems is notably necessary (e. g. for long-term maintenance reasons), at the beginning of a project the construction of information models is often neglected (Vassiliadis, Bouzeghoub, Quix 2000) as Data Warehouse engineers often attempt a fast realisation (Vassiliadis 2000). Consequently, approaches are required which increase the efficiency of information modelling. Against this background, reference models provide useful means to reduce the effort of information modelling, because they can be used as a starting point for the construction of project-specific models (Rosemann 2003). Thus, reference models provide best (or common) practice solutions for information modelling projects. They can also be referred to as a knowledge management utility.

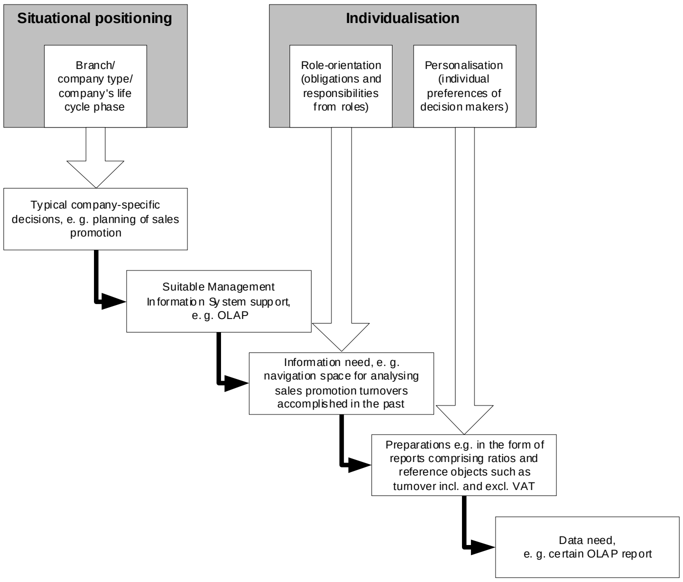
Information models perform the task of representing problems in their current processing state – or positively speaking – provide solution contributions (preliminary resp. final solutions) to a development problem (Newell, Simon 1972). The quality of a model is the better the more it complies with the subjective model user's perspective on the problem (Darke, Shanks 1996; Rosemann 1998) (*problem-oriented subjective abstraction*). However, a subject (here the model user) aligns his actions with his individual constructed perception of reality. As a result, in most cases it is not possible to adapt reference models on a 1:1 basis. It is rather required to align reference models to the particular application context.

In terms of Data Warehouse-based decision support, several factors – which are specified in the Data Warehouse requirements definition – impact the information need. They can be distinguished on basis of a theoretical model (cf. Figure 1, according to Mertens, Griese (2002), p. 67). Typical types of decisions (within a company) are determined by branch, company type, and current life cycle phase (*situational positioning*). Systems and methods for decision support determine the information need, which, in turn, implies a certain kind of preparation. The data need from external and internal sources can thereby be determined. In addition, for *individualisation* purposes, information has to be selected and represented suitable to the user's individual aspects. It is therefore necessary to consider on the one hand tasks that are linked to a certain job position within the company (*role-orientation*) and on the other hand personal preferences and aversions (*personalisation*). Consequently, individualisation can be further differentiated into role-orientation and personalisation. Thus, situational positioning and role-orientation determine a rather objective information need, whereas requirements based on personalisation hold a more subjective character (Meier 2003).

Reference models provide benefits only if the reduced effort resulting from their application is not overlapped by the adaptation effort. Adaptations are necessary since the reference model has to meet the particular business context. Therefore, reference models are either to be aligned to a specific user group or need to comprise concepts that facilitate an easy and efficient adaptation process. Furthermore, the development of reference models is often costly, risky, and extensive. This moreover underlines the demand for an easy-to-use adaptation approach. Like every other manufacturer, reference model developers need to identify their market potentials and their profits are subject to the product acceptance on the part of the customer (here companies and organisations). Thus, reference model developers face the following dilemma:

On the one hand, customers will choose a reference model that – alongside the quality of the transferred know-how – provides the best fit to their individual requirements and therefore implies the least need for changes. On the other hand, a restriction of the generality of the

*Figure 1. Impact Factors of a Data Warehouse-Based Decision Support*




model results in higher turnover risks because of smaller sales markets. Configurable reference models provide a solution to this problem. Configurable reference models comprise rules which allow modifications of the original reference model depending on company or project individual determinations of configuration parameters. By means of configuration mechanisms, it is possible to reduce the need for change which results in a higher customer acceptance.

This paper provides methodical concepts for the requirement specification of Data Warehouse Systems that are based on reference models. The remainder of this paper is organised as follows. Section 2 describes the state-of-the-art of reference modelling within the context of Data Warehouse engineering. This section concludes that reference models are sporadically applied in Data Warehouse engineering projects. Comprehensive support for the adaptation of Data Warehouse reference models, however, is mostly neglected so far. Section 3 presents an extension of a multi-dimensional modelling method with configuration concepts. The implementation of this extended method is presented in Section 4. Section 5 discusses implications of the approach to the Data Warehouse engineering process. Finally, conclusions and future work are presented in Section 6.

## RELATED WORK

Discussions concerning Data Warehouse design guidelines are strongly affected by IT architecture models (Inmon 1996) which focus on Data Warehouse components and their relationships. Classical approaches classify Data Warehouse components in layers such as sources, import, data storage, presentation, and analysis. The diffusion of these conceptual IT-layer-architectures is particularly based on the work of Devlin, Inmon and Kimbal (Devlin 1997; Kimball 1996; Devlin, Murphy 1988; Inmon 1996; Inmon, Imhoff, Sousa 1998). Further developments (such as "Real-time" or "Active Data Warehouse") consider increased requirements on timeliness of the Data Warehouse (Brobst 2002) as it is demanded e. g. in the context of Customer Relationship Management. Other improvements address mutual demands of quantitative and qualitative data by integrating Data Warehouse Systems with the functionality of Knowledge- and Content-Management-Systems (Rieger, Kleber, von Maur 2000; Shilakes, Tylman 1998; Becker, Knackstedt, Serries 2003). Technical-orientated discussions on Data Warehousing also feature different approaches for the design of Data Warehouse data bases and schemas. Alternative solutions for relational OLTP-Systems are mainly represented in the classical Star-Schema, the Snow-Flake-Schema, and the Fact-Constellation-Schema (Golfarelli, Maio, Rizzi 1998; Golfarelli, Rizzi 1999).

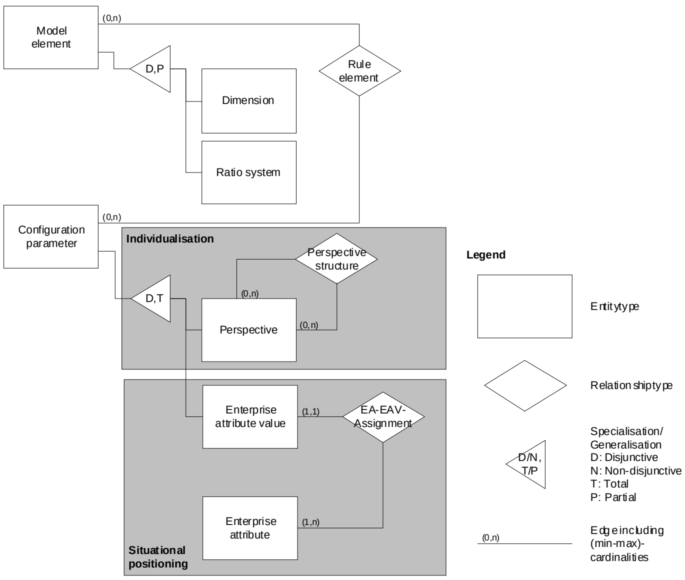
Guidelines – in terms of references – for modelling Data Warehouse Systems are particularly provided through the Common Warehouse Metamodell (CWM) by the Object Management Group (OMG) (Object Management Group 2000) and through the Open Information Modell (OIM) by the Meta Data Coalition (MDC) (Meta Data Coalition 1999). Both standards are specified in UML and define concepts for the description of Data Warehouse Systems. In contrast to OIM, CWM rather focuses on meta data management especially in the Data Warehouse context. An extensive comparison of the two standards is e. g. presented in (Jung, Rowohl 2000).

From a methodical perspective, the debate about design issues of Data Warehouse Systems is dominated by manifold modelling approaches. For the multi-dimensional specification of Data Warehouse requirements, a broad variety of modelling techniques exists (Böhnlein 2001). Some of them are closely related to Entity-Relationship Models (ERM) (Chen 1976) or provide Data Warehouse specific ERM extensions (Sapia, Blaschka, Höfling, Dinter 1998; Tryfona, Busborg, Christianson 1999). Others are derived from Scientific and Statistical Data Bases (Chan, Shoshani 1981; Rafanelli, Bezenchek, Tininini 1996; Rafanelli, Ricci 1983; Rafanelli, Shoshani 1990), are related to object-oriented modelling approaches (Harden, Herden 1999), or present a multi-dimensional modelling approach which is not based on an existing modelling technique (Bulos 1996; Thomsen 1997; Golfarelli, Maio, Rizzi 1998a; Holten 2003). These modelling approaches differ from the above-mentioned standards, especially in their strong orientation on business requirements. CWM and OIM instead focus rather on technical aspects of Data Warehouse Systems.

At the present time, technical-orientated debates regarding Data Warehouse design aspects are dominating discussions on the reuse of information models within the Data Warehouse requirements definition phase. The state-of-the-art of reference model application in the requirements specification phase of Data Warehouse projects mostly refers to an ad-hoc modification of existing information models (Adamson, Venerable 1998; Silverston, Inmon, Graziano 1997). As the analysis of various multi-dimensional modelling methods shows, the proposed modelling methods do not provide constructs for supporting model adaptation. For example, constructs for specifying parameters for model configurations are not provided. The reference models of Inmon solely represent approaches of specialisation (cf. http://www.billinmon.com). Libraries comprising reusable elements of Data Warehouse reference models are mostly specialised on particular model element types (Spitta 1997). Collections and definitions of ratios and ratio systems are widely-spread in business literature (Copeland, Koller, Murrin 1990; Eccles 1991; Lapsley, Mitchel 1996, Kaplan, Norton 1996). However, these collections neglect important aspects (mainly dimensions that have to be analysed for management tasks) of the Data Warehouse requirements specification (Holten 2003).

Reference models for the specification of business processes are, in contrast to the area of Data Warehousing, widely perceived. Moreover, approaches for configurable business process modelling are also provided (Becker et al. 2004). The adaptation of business models based on configuration patterns is widely discussed (Nordstrom, Sztipanovits, Karsai, Ledeczi 1998; Nuseibeh 1994; Nissen, Jeusfeld, Jarke 1996; Hofstede, Verhoef 1996; Kotonya, Sommerville 1995). From a practical perspective, according approaches are particularly established in the context of Enterprise Ressource Planning (ERP)-System customizing (Rosemann, Shanks 2001, Rosemann 2003). However, ERP configuration parameters for report definitions are mainly restricted to a selection of pre-defined reports and organisational roles. But the documentation of underlying configuration rules is often inadequate since the configuration is conducted on a rather technical level. Thus, end users are only able to comprehend effects of the configuration in the form of eliminated reports or eliminated report parts. Furthermore, implementation oriented approaches for the reuse concept can be found in Enterprise Information Systems (EIS). EIS provide templates that

*Figure 2. Extensions for the Design of Multi-Dimensional Modelling Techniques*




can be modified with regard to certain project requirements. However, from a methodical perspective, a consistent separation between the phases requirements definition, design specification and implementation is not ensured in this context.

The transformation of Data Warehouse specification models into design schemas and implementations is addressed in a broad variety of approaches aiming at the tool support of Data Warehousing (Hahn, Sapia, Blaschka 2000; Golfarelli, Maio, Rizzi 1998b; Blaschka 2000). These approaches aim at (semi-)automatically transforming Data Warehouse requirements specifications into initial Data Warehouse implementations. Especially, modelling transformations with CWM are discussed with respect to MDA (Model Driven Architecture) (Frankel 2003). Thus, it seems reasonable for further developments on Data Warehouse reference modelling to address the requirement specification layer. As stated above, a reception and reflection of configurative reference modelling approaches in this research area is so far inadequate. In the following, we propose methodical extensions which facilitate an application of configurative reference models in the context of Data Warehousing.

## CONFIGURATIVE REFERENCE MODELLING TECHNIQUES

In order to use configurative reference modelling concepts, it is necessary to extend the language-based meta models (e. g. Nissen et al. 1996) of existing modelling methods for Data Warehouse specifications. Constructs are required that allow for the administration of model variants (cf. the ERM in Figure 2, according to Becker, Knackstedt (2004), p.44). These model variants are generated because it is necessary to determine (in advance) which model components are exclusively relevant in a given application context. Model element types of the modelling method that are designated for configuration are connected to configuration parameters by means of rules. Which model elements are selected depends on the underlying modelling method.

Dimensions and ratio systems, for example, are often proposed in multi-dimensional modelling approaches. They represent model elements that are often adequate for the adjustment of adaptation requirements. The need for certain model elements, depending on configuration parameters, is expressed via rules. Here, we propose enterprise attribute values and perspectives as specialisations of configuration parameters.

*Figure 3. Morphologic Box for Confining Retailer Classes*

| Enterprise attribute | Enterprise attribute value | | | |
|---|---|---|---|---|
| Business Level | Retailer | | Wholesaler | |
| Extent of trading | Within the country | | Outside the country | |
| Horizontal cooperation | Retailers | Wholesalers | Other cooperation | |
| Vertical cooperation | Retail and Wholesale | Wholesale and industrial companies | Retail and industrial companies | Retail, wholesale and industrial companies |
| Contact orientation | Stationary | Itinerant | Mail order | |
| Sales contact form | Sales person | Self-service | Catalog | Vending machine |
| Beneficiary | Investment goods trade | | Consumer goods trade | |
| Range extent | Wide and deep range | Wide and shallow range | Narrow and deep range | Narrow and shallow range |
| Price policy | Active | | Passive | |

Enterprise classes that are relevant for a certain model variant are described by means of enterprise attribute values. Enterprise attribute values are used as configuration parameter to cover aspects of situational positioning (cf. again Figure 1). Enterprise classes can be presented in the form of a morphologic framework. Rows in a morphologic framework assign enterprise attributes to possible values. Marked attribute values visualize a certain enterprise class (cf. Figure 3, Becker, Uhr, Vering (2001)).

Alongside enterprise classes, potential model users within enterprises have to be identified. Model users and their different subjective abstractions are represented by means of perspectives. In this manner, requirements on individualisation are covered, as stated in Section 1.

The attribute-based selection of enterprise classes abstracts from fine-granular differences between enterprises. Analogously, perspectives generate model user clusters based on examined differences in the subjective abstraction. At current, there is no universally valid position regarding which differentiation criteria determine the definition of perspectives. For a suitable determination of perspectives the following three dimensions seem to be reasonable:

- *Purposes:* Debates on multi-perspective modelling mostly refer to the two modelling purposes organisational and application system design (Becker et al. 2004). The application system architecture MEMO (Frank 2002) introduces a strategic perspective in addition to the perspectives organisation and application system. The purpose of application system design comprises aspects such as selection of ERP-Software, Software Development, Simulation, and Workflow Management. Organisational aspects are amongst others Benchmarking, Certification (e. g. ISO 9000), Knowledge Management and Business Process Management. By means of purposes, reference model application goals are specified (Rosemann 2003).
- *Roles:* Roles represent certain reference model users within several projects. The Zachman-Framework (Zachman 1987; Inmon, Zachman, Geiger 1997) distinguishes the roles planner, owner, designer, builder, and subcontractor. The layers of the ARIS framework (Scheer 2000) can also be regarded as roles as they consider the focal point of the model user's task (creation of requirements definition, design specification, or implementation). For example, within a Software Engineering project (here as a purpose) we can distinguish roles such as project manager, end user, data base expert, etc. But the role of a project manager is also filled in Knowledge Management projects. Thus, roles and purposes are principally relatable. Nevertheless, it is not ensured that every combination of roles and purposes is reasonable.
- *Miscellaneous influences:* Ideally, a 1:1 assignment between perspectives and subjective abstractions should be achieved. Thus, the model user has to distinguish a broad variety of additional aspects

such as method affinity and competence, colours or other layout-specific preferences, interests in certain working areas etc. Especially end users participating in application system projects, in which reference models are applied, tend to focus on model parts that represent their working domain (e. g. controlling, purchasing, marketing, sales, but also wholesaler or retailer). Enterprise-specific organisational concepts (such as function-orientation or object-orientation etc.) result in a multiplicity of variants. Due to the fact that a perspective determination is strongly related to organisational concepts, this aspect is difficult to take into account by reference model developers. However, in the context of model adaptation, the organisational aspect is more important. Due to the broad variety of possible aspects and the resulting complexity, only a few aspects of the dimension miscellaneous influences can be taken into account in reference modelling projects. However, in a specific application context, this dimension will be dominated by both dimension purposes and roles.

Perspectives can be combined by other perspectives (cf. again Figure 2). This fact is represented in perspective structures. Perspectives structures allow e. g. the differentiation of different project managers (e. g. from purchase and sales).

In the following, our extension concept will be applied to a modelling technique for the specification of management views. This technique is developed and elaborated on in detail in HOLTEN (Holten 2003). Its applicability is described in a series of business cases (e. g. Holten, Dreiling, Schmid 2002). From a modelling perspective, a task and role-orientated data specification presented on basis of a Data Warehouse can be interpreted as the task of constructing a navigation space for data. From a business perspective, following the work of RIEBEL (Riebel 1979; Riebel 1992) and SCHMALENBACH (Schmalenbach 1956), this navigation space– called information object – is spanned by reference objects and ratios.

Reference objects are defined as all “measures, processes and states of affairs which can be object to arrangements or examinations on their own.” (Riebel 1979). A reference object is everything that is related to a decision in a business process (such as products, sales promotions, customers, regions, sales channels, performance, and days). Reference objects can be assigned – with respect to the modelling purpose – to a specific analysis dimension. By means of dimensions we can for example analyse products according to countries of origin, or products according to typical product lines (such as a distinction of food and non-food). Usually, elements within dimensions, so-called dimension objects, are arranged hierarchically (for example to represent customer hierarchies or product groups). Dimensions that comprise identical reference objects as leaf elements are combined to dimension groups. Reference objects can be composed of other (non-combined) reference objects. Sets of reference objects can be defined by building a set of dimensions or scopes of dimensions. This set of (combined) reference objects can be regarded as a navigation room that is used for management analyses. These analyses are based on the operations aggregation and dis-aggregation with respect to the hierarchies of the dimensions.

Ratios are of fundamental importance for specifying information in management processes (Kaplan, Norton 1996). They define important aspects of reference objects such as invoice and payment amount, gross margin, product profitability, etc. They are clearly defined in the sense of a management view. Typically ratios are organised in hierarchies to enable top down analyses of unique reference objects according to different ratios. Business information can not be expressed exclusively based on ratios or reference objects (e. g. a statement such as ‘turnover is 400’ is senseless). Thus, ratios and reference objects have to be combined.

The assignment of ratios systems to a set of specified reference objects leads to a set of (business) facts. Facts represent tuples of ratios and combined reference objects (e. g. number of products sold in a certain region by a specific customer representative, turnover achieved with a certain product). Specifying a set of facts is appropriate to describe the navigation space relevant to management decisions.

Against the background of Category Management, for example, we can specify an information object that provides an analysis of alternative aggregations of planned and actual values (cf. dimension valuation) for products, sales promotions and days by means of the ratios turnover and product profitability. This specification facilitates the development of application systems by providing systematic guidelines for their upcoming implementation steps (e. g. initialising data base tables in a Data Warehouse, defining reports of an OLAP-System).

Figure 4 (according to Becker, Knackstedt (2004), p. 44) illustrates the application of the reference model configuration. Here, the specification of a navigation space for product group analysis is provided. By selecting different dimensions resp. dimension scopes and ratio systems, the navigation space (as previously described) is determined. Within our example, configuration parameters are enterprise attribute values ‘transaction type’, ‘purchase area’ and ‘report frequency’. An analysis of product turnovers according to sales promotion types seems to be reasonable only if the retailer makes use of a sales promotion business instead of a permanent low price strategy. Moreover, a consideration of products according to countries of origin only makes sense in case of an ‘international’ purchase area. The report frequency affects selection possibilities of analysis hierarchies with respect to the reference object ‘time’.

Rules can be differentiated in alternative representation forms. So-called build-time operators provide means to define configuration rules and to assign them to (a group of) specific model elements. Build-time operators should be applied for those model elements that enable enclosing (e. g. opening and shutting connectors in Extend Process Chains) or concatenation of specific model elements to each other (e. g. relationship types of Entity Relationship Models). In case of multi-dimensional models, build-time operators suit model elements defining the navigation space. The underlying rule basis can be presented in the form of decision tables, for example. The decision table depicted in Figure 4 assigns the stated conditions in the form of combinations of enterprise attribute values with specific actions. Actions consist of removing resp. adding model elements. The crosses used in Figure 4 illustrate which model element is a component of the derived enterprise-specific model. By means of analogous extensions we are able to create models that include perspectives and ratio systems as well.

An alternative representation form is the use of parameterisations that can be added to certain model elements. Parameterisations determine, depending on configuration parameter values, which model elements are part of the derived project-specific model (Schwegmann 1999). Figure 4 illustrates the application of parameterisations. Here, the rule ‘purchase area (international)’ is annotated to the dimension ‘product group ordered by countries of origin’. This rule defines that the dimension ‘product group order by countries of origin’ is to be dropped out in case of an enterprise exclusively ‘nationally’ purchasing. The syntax of parameterisations can be defined in the form of a context-free grammar (cf. Figure 5). The Extended-Backus-Naur-Form (EBNF) (Hopcroft, Motwani, Ullman 2000) can be used (on meta meta level) defining the grammar itself.

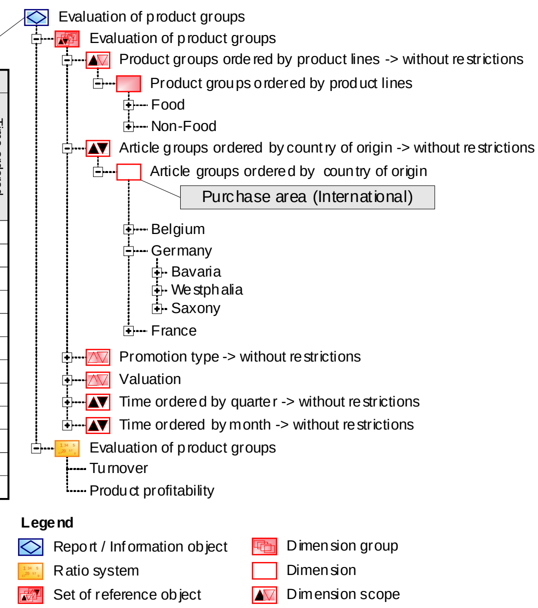
## IMPLEMENTATION

To a great part the *configuration* of the models themselves can be conducted automatically. The specification of configurative Data Warehouse models can principally be implemented through standard data bases. In the following, we suggest an alternative implementation approach in Prolog. At the starting point of the implementation, configuration parameters have to be identified. Figure 6 (according to Knackstedt (2004), p. 124) depicts an accordant dialogue for the enterprise parameter values ‘transaction type’, ‘purchase area’ and ‘report frequency’ (cf. for the example again Figure 4). Here, we suppose that the dialogue is implemented within a procedural programming language. Figure 6 (4) illustrates a possible realisation for the interface between the dialogue module and the inference machine (Prolog server). After identifying the configuration parameters, the inference machine is retrieved by means of the predicate ‘DeliverModelElements’. In doing

*Figure 4. Configurable Multi-Dimensional Model*

| Conditions | | | Activities | | | | | |
|---|---|---|---|---|---|---|---|---|
| Transaction type | Purchase area | Report frequency | Product groups ordered by product lines | Product groups ordered by country of origin | Promotion type | Valuation | Time ordered by quarter | Time ordered by month |
| [WB \| WB + TPD] | [International \| International + Regional] | [Week] | X | X | | X | X | |
| | | [month] | X | X | | X | | X |
| | | [Week + Month] | X | X | | X | X | X |
| | [Regional] | [Week] | X | | | X | X | |
| | | [Month] | X | | | X | | X |
| | | [Week + Month] | X | | | X | X | X |
| [SPB \| WB + SPB \| WB + SPB + TPD \| SPB + TPD] | [International \| International + Regional] | [Week] | X | X | X | X | X | |
| | | [Month] | X | X | X | X | | X |
| | | [Week + Month] | X | X | X | X | X | X |
| | [Regional] | [Week] | X | | X | X | X | |
| | | [Month] | X | | X | X | | X |
| | | [Week + Month] | X | | X | X | X | X |

**Legend**
WB := Warehousing business
TPD := Third-party deal
SPB:= Sales promotion business




so, parameters of the previous dialogue are forwarded. Depending on the forwarded configuration parameters, the predicate creates a list of all existing model elements (variable X) that are to be transferred into the configured model. Therefore the predicate 'DeliverModElelements' adds further predicates into the rule basis ("asserta"). Thus, the validity of respective parameters is specified (cf. Figure 6 (3)). Thereupon, the predicate 'CreateModelElementList' identifies a valid model variant applying the meta predicate 'findall'. This predicate verifies successively, if model elements are contained in the resulting parameter-specific model.

The predicate 'ModelElement' is used to evaluate if certain model elements are included in the configured model (cf. Figure 6 (1)). If the predicate value returns 'true' the model element name is added to the model element list that is bound to the variable X. Constraints of the predicate 'ModelElement' ensure that model elements can only once be added to the list. The decision about adding model elements to the list is conducted by computing the predicates of the configuration rules (cf. Figure 6 (2)).

After analysing the rule basis, a model element list that is bound to the variable X in the form of character strings is available (cf. again Figure 6 (4)). This list comprises the relevant model elements and allows for identifying the relevant application-specific reference model variant. For (graphical) presentation purposes this list can be transferred to an appropriate editor.

*Figure 5. Context-Free Grammar in EBNF for the Specification of Parameterisations*

```
<Term>                          ::= <Expression> {<Operator> <Expression>}
<Expression>                    ::= <Prefix> "Perspective"
                                    <Perspective_Value_List>
<Expression>                    ::= <Prefix> "Enterprise_Attribute"
                                    <Enterprise_Attribute_Value_List>
<Perspective_Value_List>        ::= "(" <Prefix> <Perspective_Value_List>
                                    { <Operator> <Prefix> <Perspective_Value_List> } ")"
<Perspective_Value_List>        ::= <Perspective>
<Enterprise_Attribute_Value_List> ::= "(" <Prefix> <Attribute_Value_Value_List>
                                    { <Operator> <Prefix> <Attribute_Value_Value_List> } ")"
<Enterprise_Attribute_Value_List> ::= <Enterprise_Attribute>
<Operator>                      ::= "|" | "+"
<Prefix>                        ::= "NOT" | <empty>
```

## IMPLICATIONS FOR THE DATA WAREHOUSE DEVELOPING PROCESS

The application of configurable reference models alters the usual Data Warehouse engineering process. By means of suitable tool support, the following tasks basically evolve (cf. Figure 7, according to Becker, Knackstedt (2004), p. 47):

- *Construction:* At first, reference models have to be constructed. A capable reference model construction needs appropriate tool support. Thus, a modelling editor must support not only standard multi-dimensional constructs but also the definition of configuration rules. Moreover, enterprise attribute values and perspectives defining the configuration rules could be used to classify the constructed reference model. As a result, constructed reference models can easily be retrieved and selected in reference model catalogues (Fettke, Loos 2003).
- *Configuration:* After retrieving a configurative reference model, identification of project-specific configuration parameters values is required. Querying these values can be supported through appropriate dialogues, as presented in Section 4. Based on received configuration parameters relevant model variants have to be created. The generation of model variants can be supported through configuration components. For implementation aspects, classical data base solutions, or – as again stated in Section 4 – Prolog allows for specifying and applying configurative multi-dimensional models.
- *Adaptation:* Configurative reference models are not able to support all supposable project specifics. Configuration parameters cluster certain project varieties in order to reduce complexity. Therefore, from further specialities is abstracted. Consequently, it is necessary to evaluate the configured Data Warehouse reference model with respect to the correctness of the determined information need. During this evaluation well-established methods for the specification of information needs can be applied (Carter 1983; Watson, Frolick 1993). Configurative reference models provide initial solutions that prompt critical reflection and improvements already at the stage of interviews and document analyses. In many cases, the provision of initial solutions is a basic requirement for an effective interrogation of management executives. For adapting configured models modelling editors should be provided as well.

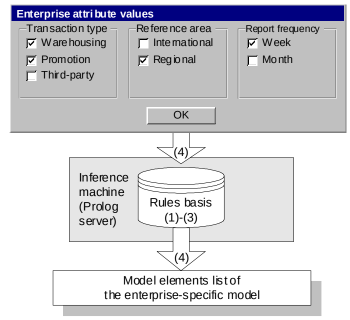
*Figure 6. Tool-Based Model Configuration*

**(1) Model elements (Prolog)**

```
ModelElement ('ProductGroupProductLines') :- ProductGroupProductLines.
ModelElement ('ProductGroupCountryofOrigin') :- ProductGroupCountryofOrigin.
ModelElement ('PromotionGroupPromotionType') :- PromotionGroupPromotionType.
ModelElement ('Valuation') :- valuation.
ModelElement ('DayGroupWeek') :- DayGroupWeek.
ModelElement ('DayGroupMonth') :- DayGroupMonth.
```

**(2) Build-time operator rules (Prolog)**

```
ProductGroupProductLines.
ProductGroupCountryofOrigin :- not (PurchaseArea ('Regional')).
PromotionGroupPromotionType :- TransactionType ('Promotion').
PromotionGroupPromotionType :- TransactionType ('WarehousingPromotion').
PromotionGroupPromotionType :- TransactionType ('WarehousingPromotionThird-partyDeal').
PromotionGroupPromotionType :- TransactionType ('PromotionThird-partyDeal').
Valuation.
TimeGroupWeek:- ReportFrequency ('Week').
TimeGroupMonth:- ReportFrequency('Month').
```




**(3) Evaluation of Rules basis (Prolog)**

```
DeliverModelElements (TranscationType, PurchaseArea, ReportFrequency, X) :-
        asserta (TranscationType (TranscationType)),
        asserta (PurchaseArea (PurchaseArea)),
        asserta (ReportFrequency (ReportFrequency)),
        CreateModelElementList (X).
CreateModelElementList (ModelElementList) :-
        findall (X, ModelElement (X), ModelElementList).
```

**(4) Dialogue and control component (procedural programming language)**

```
ConnectionEngine.ExecStr (t, 'DeliverModElelements (WarehousingPromotion, Regional, Week, X)');
ConnectionEngine.GetArg (t, 5, dTERM, @term);
```

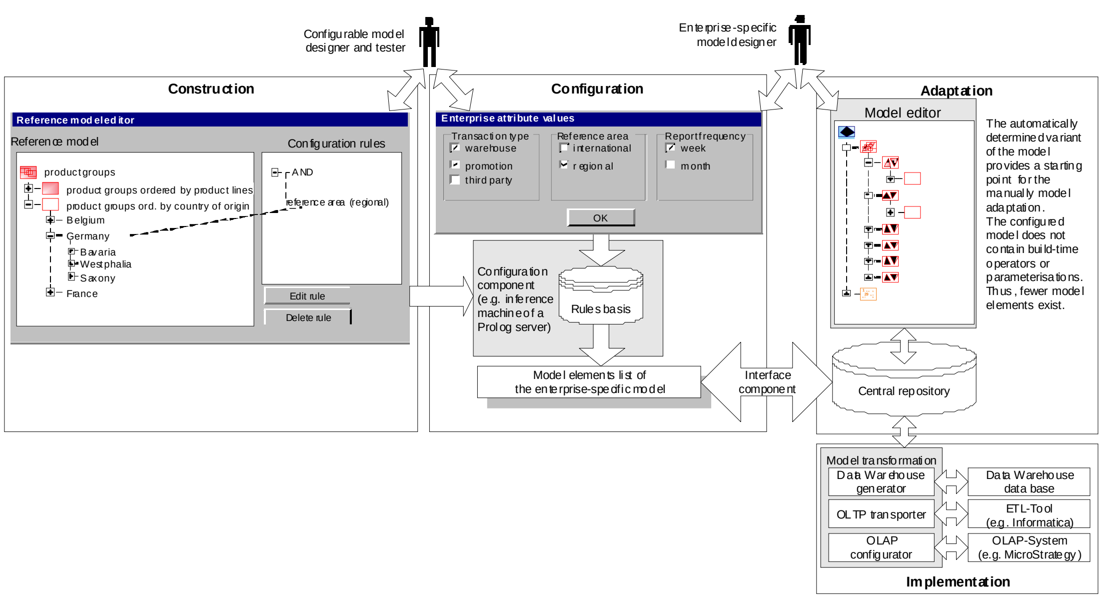
- *Implementation:* Following the adaptation of the reference models phase, the requirements definition model needs to be transformed into a technical implementation. Therefore, further Software modules can be applied for (semi-) automatically generating underlying data base or Data Warehouse schemas. Moreover, tools are needed that provide ETL processes with matching rules for data as well as tools that allow for the report definition of OLAP-Systems. Here, former research results need to be integrated. Detailed algorithms for model transformations of the modelling methods applied in this paper are presented in Holten (Holten 2003). Similar approaches exist for other modelling techniques as well.

## CONCLUSIONS AND FUTURE WORK

In this paper we presented process-oriented modifications for the development of Data Warehouse Systems by means of configurative reference models. Configurative reference models enhance the efficiency of information modelling since they present on the one hand a basis for the evaluation of individual solutions. On the other hand, they provide adaptable model parts and business vocabularies. Moreover, they offer guidelines for the methodical design of own modelling systems. Alongside these benefits, however, effort incurs for selecting and acquiring applied initial models as well as for adapting this model to project-specific requirements. From a producer's perspective, a novel and self-dependend (or bundled with existing modelling tools and services) product is created. From an economical research perspective, the effects of reference model applications should be investigated in detail.

From a methodical perspective, future research needs to address the combination of reference model configurations with further approaches for the support of model adaptations. Thus, an integration of the

*Figure 7. Configurative Reference Model Based Development Process*



configuration and adaptation phase within the requirements definition phase of Data Warehouse Systems is aspired. Against this background, the aggregation of reference building blocks is of high relevance. To ensure the consistence between certain model parts, model elements such as dimensions and ratio system definitions should be organised in libraries. In case of (initial) model extensions, according model elements should be extracted from these libraries (aggregation). Libraries can consist of two layers. The first layer comprises all model elements of the configured reference model. The second layer includes additionally model elements which are eliminated within the configuration phase as well. Furthermore, the creation of configurative reference models for specific branches (especially in the domain of retailing) is intended.

## REFERENCES


Adamson, C.; Venerable, M. (1998): Data Warehouse Design Solutions. New York.

Becker, J.; Delfmann, P.; Dreiling, A.; Knackstedt, R.; Kuropka, D. (2004): Configurative Process Modeling – Outlining an Approach to Increased Business Process Model Usability. In: Khosrow-Pour, M. (Eds.): Information Through Information Technology. 2004 Information Resources Management Association. International Conference. New Orleans, Louisiana, USA. May 23-36, 2004. Hershey et al. 2004, pp. 615-619.

Becker, J.; Knackstedt, R. (2001): Konfigurative Referenzmodelle als Instrumente des Wissensmanagements bei der Data-Warehouse-Entwicklung. In: Schnurr, H.-P.; Saab, S.; Studer. R.; Stumme, G.; Sure, Y. (Eds.): Professionelles Wissensmanagement. Erfahrungen und Visionen. Aachen 2001, pp. 113-128. [in german].

Becker, J.; Knackstedt, R. (2004): Referenzmodellierung im Data Warehousing – State-of-the-Art und konfigurative Ansätze für die Fachkonzeption. Wirtschaftsinformatik, 46 (2004) 1, pp. 39-49. [in german].

Becker, J.; Knackstedt, R.; Serries, T. (2003): Architectures for Enterprise Information Portals: An Approach to Integrate Data Warehousing and Content Management. In: Ciborra, C.; Mercurio, R.; De Marco, M.; Martinez, M.; Carignani, A. (Eds.): New Paradigms in Organizations. Markets and Society. Proceedings of the 11th European Conference on Information Systems (ECIS 2003). Naples, 19-21 June 2003.

Becker, J.; Uhr W.; Vering, O. (2001): Retail Information Systems Based on SAP Products. Berlin et al. 2001.

Blaschka, M. (2002): FIESTA: A Framework for Schema Evolution in Multidimensional Databases. Dissertation, University of Munich 2000.

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

Böhnlein, M. (2001): Konstruktion semantischer Data-Warehouse-Schemata. (Construction of Semantic Data Warehouse Schemata). Wiesbaden, Germany 2001. [in german].

Brobst, S. (2002): Delivering Extreme Data Freshness with Active Data Warehousing. In: Journal of Data Warehousing. The Data Warehousing Institute. Spring 2002.

Brobst, S. A. (2002): Enterprise Application Integration and Active Data Warehousing. In: Von Maur, E.; Winter, R. (Eds.): From Data Warehouse to Corporate Knowledge Center. Proceedings of Data Warehousing 2002. Heidelberg 2002, pp. 15-22.

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

Carter, M. P. (1983): Determining information needs. Management Decision, 21 (1983) 4, pp. 45-51.

Chan, P.; Shoshani, A. (1981): SUBJECT – A Directory Driven System for Organizing and Accessing Large Statistical Databases. In: IEEE Computer Society: Proceedings of the 7th International Conference on Very Large Data Bases (VLDB'81). Cannes, France, 9.-11. September 1981. Cannes 1981, pp. 553-563.

Chen, P. P.-S. (1976): The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems, 1 (1976) 1, pp. 9-36.

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

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

Darke, P.; Shanks, G. (1996): Stakeholder Viewpoints in Requirements Definition. Requirements Engineering, 1 (1996) 1, pp. 88-105.

Davis, A. M. (1990): 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. (1988): A Strategy for Comparing Alternative Software Development Life Cycle Models, IEEE Transactions on Software Engineering 14 (1998) 10, pp. 1453-1461.

Devlin, B. (1997): Data Warehouse. From Architecture to Implementation. Reading et al. 1997.

Devlin, B. A.; Murphy, P. P. (1988): An Architecture for a Business and Information System. IBM Systems Journal, 27 (1988) 1, pp. 60-80.

Eccles, R. G. (1991): The performance measurement manifesto. In: Harvard Business Review, 69 (1991) 1-2, pp. 131-137.

Fettke, P.; Loos, P. (2003): Classification of reference models: a methodology and its application. Information Systems and e-Business Management, 1 (2003) 1, pp. 35-53.

Frank, U. (2002): Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-35), Honolulu 2002.

Frankel, S. D. (2003): Model Driven Architecture TM. Apllying MDA to Enterprise Computing. Indianapolis/ Indiana 2003.

Golfarelli, M.; Maio, D.; Rizzi, S. (1998a): The Dimensional Fact Model – A Conceptual Model for Data Warehouse. International Journal of Cooperative Information Systems, 7 (1998) 2-3, pp. 215-246.

Golfarelli, M.; Maio, D.; Rizzi, S. (1998b): Conceptual Design of Data Warehouses from E/R-Schemes. In: H. El-Rewini (Hrsg.): Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98). 6.-9. Januar 1998. Los Alamitos et al. 1998. pp. 334-343.

Golfarelli, M.; Rizzi, S. (1999): Designing the data warehouse: Key steps and crucial issues. In: Journal of Computer Science and Information Management. Vol. 2, Nr. 3, 1999.

Hahn, K.; Sapia, C.; Blaschka, M. (2000): Automatically Generating OLAP Schemata from Conceptual Graphical Models. In: Proceedings of the ACM Third International Workshop on Data Warehousing and OLAP (DOLAP 2000). Washington D. C., USA, 10. November 2000.

Harren, A; Herden, O. (1999): MML and mUML – Language and Tool for Supporting Conceptual Data Warehouse Designs. In: Proceedings 2.GI-Workshop Data Mining und Data Warehousing (DMDW99), pp. 57-68, Magdeburg (Germany), September 1999. LWA99. Papers in edited volumes. [in german].

Hofstede, A.H.M., Verhoef, T.F. (1996): Meta-CASE: Is the game worth the candle? In: Info Systems Journal, (6), 1996, pp. 41-68.

Holten, R. (2003): Specification of management views in information warehouse projects. Information Systems, 28 (2003) 7, pp. 709-751.

Holten, R.; Dreiling, A.; Schmid, B. (2002): Management Report Engineering – A Swiss Re Business Case. In: E. von Maur, R. Winter (Eds.): From Data Warehouse to Corporate Knowledge Center. Proceedings of Data Warehousing 2002. Heidelberg 2002, pp. 421-437.

Hopcroft, J. E.; Motwani, R.; Ullman, J. D. (2000): Introduction to Automata Theory, Languages, and Computation. 2 ed. Boston, MA, USA: Addison-Wesley Longman Publishing.

Inmon, W. H. (1996): Building the Data Warehouse. 2. ed., New York.

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

Inmon, W. H.; Imhoff, C.; Sousa, R. (1998): Corporate Information Factory. New York et al. 1998.

Inmon, W. H.; Zachman, J. A.; Geiger, J. G. (1997): Data Stores, Data Warehousing, and the Zachman Framework. Managing Enterprise Knowledge. New York et al. 1997.

Jung, R.; Rowohl, F. (2003): Vergleichende Analyse der Standardisierungsbestrebungen im Metadatenmanagement. Neue Modelle der MDC und OMG. (Analysing Standards in the Context of Meta Data Management. New Models of MDC and OMG). In: R. Jung, R. Winter (Eds.): Data Warehousing 2000. Heidelberg 2000, pp. 113-133. [in german].

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

Karimi, J. (1998): Strategic Planning for Information Systems: Requirements and Information Engineering Methods. Journal of Management Information Systems, 4 (4), 1988, pp. 5-24.

Kimball, R. (1996): Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouses. New York 1996.

Kotonya, G.; Sommerville, I. (1995): Requirements Engineering With Viewpoints. Technical Report: CS EG/10/1995. Cooperative Systems Engineering Group. Computing Department. Lancaster 1995.

Kottemann, J. E.; Konsynski, B. R. (1984): Information Systems Planning and Development: Strategic Postures and Methodologies. Journal of Management Information Systems, 1 (2), 1984. pp. 45-63.

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

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

Meier, M. C. (2003): Situations- und benutzerorientierte Filterung von Führungsinformations-systemen. (Situation and user-orientated filtering of management information systems) In: Proceedings of the German Conference for Information Systems (Wirtschaftsinformatik 2003). Dresden 2003, pp. 475-494. [in german].

Mertens, P.; Griese, J. (2002): Integrierte Informationsverarbeitung. (Integrated Information Processing) 9. eds, Wiesbaden 2002. [in german].

Meta Data Coalition (1999): Meta Data Coalition - Open Information Model. Austin, Texas 1999.

Newell, A.; Simon, H. A. (1972): Human problem solving. 2. ed., Englewood Cliffs, N. J., Prentice Hall, 1972.

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

Nordstrom, G.; Sztipanovits, J; Karsai, G.; Ledeczi, A. (1998): Metamodeling – Rapid Design and Evolution of Domain-Specific Modeling Environments. In: Proceedings of the IEEE ECBS'98 Conference. Nashville, Tennessee, April 1998. Nashville 1998, pp. 68-74.

Nuseibeh, B. A. (1994): A Multi-Perspective Framework for Method Integration. Dissertation, University of London 1994.

Object Management Group (2000): Common Warehouse Metamodel (CWM) Specification, OMG document. Needham, MA 2000. Access date: 2000-01-01.

Object Management Group (2002): Object Management Group (OMG) Homepage – The CWM Specification. 2002, http://www.omg.org/cwm. Access date: 2002-08-10.

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

Rafanelli, M.; Bezenchek, A.; Tininini, L. (1996): The Aggregate Data Problem – A System for their Definition and Management. Sigmod Record, 25 (1996) 1, pp. 8-13.

Rafanelli, M.; Ricci, F. L. (1983): Proposal of a Logical Model for Statistical Data Base. In: Proceedings of the Second International Workshop on Statistical Database Management (SSDBM'83). Los Altos, USA, 27.-29. September 1983, pp. 264-272.

Rafanelli, M.; Shoshani, A. (1990): STORM – A Statistical Object Representation Model. In: Z. Michaelewicz (Eds.): Proceedings of the 5th International Conference on Statistical and Scientific Database Management (SSDBM'90). Charlotte, USA, 3.-5. April 1990, pp. 14-29.

Riebel, P. (1979): Gestaltungsprobleme einer zweckneutralen Grundrechnung, ZfbF 31 (1979), pp. 863-893. [in german].

Riebel, P. (1992): Einzelerlös-, Einzelkosten- und Deckungsbeitragsrechung als Kern einer ganzheitlichen Führungsrechung, In: W. Männel (Ed.), Handbuch Kostenrechnung, Wiesbaden, 1992, pp. 247-299. [in german].

Rieger, B.; Kleber, A.; von Maur, E. (2000): Metadata-Bases Integration of Qualitative and Quantitative Information Resources Approaching Knowledge Management. In: H. R. Hansen, M. Bichler, H. Mahrer (Eds.): Proceedings of the 8th European Conference on Information Systems (ECIS 2000). Vienna 2000, pp. 372-378.

Rosemann, M. (1998). Managing the Complexity of Multiperspective Information Models using the Guidelines of Modelling. Proceedings of the 3rd Australian Conference on Requirements Engineering, Geelong, Australia 1998.

Rosemann, M. (2003): Application Reference Models and Building Blocks for Management and Control. In: Handbook of Enterprise Architecture. Eds.: P. Bernus, L. Nemes, G. Schmidt. Berlin et al. 2003, pp. 595-615.

Rosemann, M. (2003): Preparation of Process Modeling. In Becker, J.; Kugeler, M.; Rosemann, M. (Eds.). Process Management. A Guide for the Design of Business Processes. Berlin et al. 2003, pp. 41-78.

Rosemann, M.; Shanks, G. (2001): Extension and Configuration of Reference Models for Enterprise Resource Planning Systems. Proceedings of the 12th Australasian Conference on Information Systems (ACIS 2001). G. Finnie, D. Cecez-Kecmanovic, B. Lo. Coffs (Eds.). Harbour, 4-7 December, pp. 537-546.

Sapia, C.; Blaschka, M.; Höfling, G.; Dinter, B. (1998): Extending the E/R Model for the Multidimensional Paradigm. In: Proceedings of the International Workshop on Data Warehouse and Data Mining (DWDM'98). Lecture Notes in Computer Science. Singapur, 19.-20. November 1998, pp. 105-116.

Scheer, A. W. (2000): ARIS – Business Process Modelling. 3. ed., Berlin et al. 2000.

Scheer, A.-W. (1992): Architecture of Integrated Information Systems. Foundations of Enterprise-Modelling. Berlin et al. Springer, 1992.

Schmalenbach, E. (1956): Kostenrechnung und Preispolitik. 6 ed. Opladen 1956. [in german].

Schwegmann, A. (1999): Objektorientierte Referenzmodellierung. Theoretische Grundlagen und praktische Anwendung. (Object-oriented Reference Modelling). Dissertation. University of Münster. Wiesbaden 1999. [in german].

Shilakes, C. C.; Tylman, J. (1998): Enterprise Information Portals. New York 1998. http://www.kellen.net/ ect580/ Merrill_Lynch_EIP.pdf, Access date: 2003-03-01.

Silverston, L.; Inmon, W. H.; Graziano, K. (1997): The Data Model Resource Book. New York 1997.

Spitta, T. (1997): Wiederverwendbare Attribute als Ausweg aus dem Datenchaos. In: HMD, 34 (1997) 195, pp. 38-55. [in german].

Thomsen, E. (1997): OLAP Solutions – Building Multidimensional Information Systems. New York 1997.

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

Vassiliadis, P. (2000): 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), June 5-6, Stockholm, Sweden.

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

Watson, H. J.; Frolick, M. N. (1993): Determining Information Requirements for an EIS. MIS Quarterly, 17 (1993) 3, pp. 255-269.

Zachman, J. A. (1987): A Framework for Information Systems Architecture. IBM Systems Journal, 26 (1987) 3, pp. 277-293.

## Related Content