A Design Methodology for Data Warehouses

Olaf Herden

Oldenburg Research and Development Institute
for Computer Science Tools and Systems (OFFIS),
Escherweg 2, 26121 Oldenburg, Germany
olaf.herden@offis.de

Abstract

The objective of this work is to develop a design methodology for data warehouses. It is based on the three level modeling approach with emphasis on conceptual modeling. Logical design to the relational model and physical tuning in this environment will also be treated.

1 Research Question

In recent years, data warehouses (DWs) [Inm92] as backbone of decision support systems caused a lively interest in research and practice. Typically, the DW is a database held separately from operational systems. Its data are integrated from the operational systems of an organization and often supplemented by data from external sources. The increasing popularity of DWs reflects the rising requirement to make strategic use of data integrated from heterogeneous sources. Some examples from economy for using the data stored in a DW are database marketing, controlling and (long–term) binding of customers.
But there are also application scenarios outside the economical context, e. g. in medical registries. To be able to make reliable predictions about e. g. local accumulation of a disease DWs can be used.
All these application scenarios are very important for the organizations and they all have in common that the underlying DW has to be reliable, maintainable and expandable. For ensuring these quality aspects a systematic engineering of DWs is necessary.
Data in a DW are modelled multidimensionally because this kind of modeling reflects the end user's understanding of the problem domain. The most important characteristic of the multidimensional model is dividing data into facts (also called measures or quantifying data) and dimensions (also called qualifying data).
To build reliable, durable DWs meeting the users' requirements a design methodology both falling back on the experiences from designing classical OLTP (online transaction processing) databases and considering the special aspects of DWs is necessary.
The aim of this work is to build such a design methodology. Lessons learned while

designing conventional OLTP databases should be considered, e. g. starting with conceptual modeling, followed by a transformation to the logical level and a physical design step. On the other hand, different requirements to both types of databases have also to be taken into account, particularly with regard to the following issues:

- In OLTP databases all data being relevant for the operational business have to be modeled whereas conceptual modeling for DWs should comprehend all information needed for decision support.
- When modeling a DW the multidimensional model should be considered.
- While the physical design of an OLTP database has to be optimized for high throughput of transactions, a DW should provide a good basis for OLAP (Online Analytical Processing)–tools querying complex amounts of data with minimal response time.
- Meta data play an important role in the context of DWs, e. g. describing the source of data.

2 Related Work

We can classify the related work by the four areas conceptual multidimensional modeling, transformation to the logical level, physical database design and meta data.

Conceptual Multidimensional Modeling

In the area of DWs some approaches for conceptual multidimensional modeling have been developed, namely MERM (*Multidimensional E/R Model*) [SBHD98], ADAPT (*Application Design for Analytical Processing Technologies*) [Bul96] and DFM (*Dimensional Fact Model*) [GMR98]. But they have some deficits: ADAPT and DFM have no formal foundation and both of them have no adequate expressiveness, especially for modeling sophisticated dimensional structures. Moreover, ADAPT does not distinguish between design levels and does not support a continuous development process, while DFM is not supported by tools. MERM has a well foundation with an extension of the relational calculus, distinguishes strictly between design levels and is embedded in a tool supported environment. But some aspects of multidimensional modeling, e. g. optional dimensional attributes or limiting aggregation to special operations can not be modeled. Furthermore, using an extension of the E/R model is oriented towards a relational implementation. Last of all, all approaches are not compatible among each other and none of them has object–oriented aspects.
On the other hand some commercial tools [Mic00b], [Inc00] for designing DWs are available but they are most often proprietary. Hence conceptual modeling is often left out in practice or the conceptual and logical levels coincide, e. g. applying Kimball's dimensional modeling [Kim96].

Transformation

There are some approaches [RBP$^+$93], [BPS97], [Cor98] to transform an object–oriented model into a relational schema. Usually they define transformation rules for each kind of connection. This results in a clear transformation but is inadequate in our context because we would loose expressivness e. g. distinction between different types of classes (facts and dimensions) made on the conceptual level.

Physical Database Design

The three major tasks of physical database design are indexing, materialized views and partitioning. About all three aspects much work has been done in the past, also with respect to the special needs of data warehouses.
In the field of indexing the B–tree [BM72] or B*–tree [Wed74] are the classical one. It is also used in the context of DWs. But moreover, in DWs there is need for indexes as basis for efficient intersection– and union–operations. So the development of new kinds of indexes like bitmap indexes [OQ97b] has benn forced by data warehouses. Another appraoch are the UB–trees [Bay96], [MBB99a], [MBB99b], an index structure for the efficient processing of multidimensional range queries. For special solutions of indexing in the context of DWs the work of the Stanford database group is to be named [GHRU97] and is also be treated in [OQ97a], [Gra99].
Also in the field of materialized views this group has done a lot of work [Gup97], [GS97], [GM97], especially on the topics incremental maintenance and consistency on multi–source updates.
Original, partitioning has been developed for distributed [OV91], [BG92] and parallel databases [DG92], [AW98]. Of late, partitioning is also applied in centralized databases to handle very large database tables [KN99] and is already realized in commercial systems [DG98]. A related approach can be found in [MWM99].
But all the work listed former only considers one aspect of physical database design. A more holistic approach for the task of physical database design can be found in [RS91]. The authors propose a two-phase algorithm for physical database design. In phase one the algorithm, for each logical query, uses rules to determine characteristics of a physical design (such as indexes) that would be beneficial to the query, and selects a physical design that yields a low cost estimate for that query. In phase two a notion of compromise is used between physical database designs. Starting from the physical designs selected in phase one, the algorithm looks for a compromise physical design that minimizes the cost of a set of queries.

Meta Data

In the area of meta data a few standards have been proposed, namely:

- MDIS (Metadata Interchange Specification) by MDC (Meta Data Coalition) [Coa00]
- IRDS (Information Resource Dictionary System) [Gro00a]
- CWMI (Common Warehouse Metadata Interchange (OMG)) [Gro00b]
- OIM (Microsoft Open Information Model) [Mic00a]
- MDAPI (Multi-Dimensional API (Olap Council)) [Cou00]

In the meantime, MDIS is obsolete because Microsoft is also member of the MDC and so OIM is the common standard.
Furthermore, SMART (Supporting Metadata for Data Warehousing Systems) [Pro00] as a research project with practical regards is to be named.

3 Research Methodology

Within the scope of our research project ODAWA (OFFIS Tools for Data Warehousing) [Her99] our procedure to tackle the research problems described above consists of the following subtasks:

- *Definition of a framework for designing data warehouses:* We want to sketch the main actions of designing data warehouses.
- *Definition of a language for conceptual multidimensional modeling:* We want to design a language for multidimensional conceptual modeling. This language should allow the user to model in terms of the multidimensional model and provide constructs to build sophisticated multidimensional schemas.
- *Construct a transformation algorithm:* The algorithm should transform our multidimensional schema into a relational schema. During this process multidimensional properties should be preserved.
- *Physical database design:* A method for adequate physical database design should be applied. This method should consider different aspects of physical design (indexing, partitioning, materialized views) and should be configurable for different target database management systems.
- *Meta data:* During the process described in the subtasks above a lot of meta data is produced. They should be integrated into a meta data repository.
- *Prototypical implementation of selected software modules:* The design methodology should be tool–supported continously. One aim is to extend existing products. Those components where this is not possible are implemented prototypically in order to be able to demonstrate their basic functionality.
- *Evaluation by means of a real–world application:* To prove the soundness of the design methodology and its implementation, we want to apply it to a real–world application.

4 Basic Ideas and Preliminary Results

The framework for the design methodology is based on the three–level–modeling proved in designing conventional OLTP databases. Furthermore, on the conceptual level we are distinguishing between the language and the graphical representation. The framework is sketched in figure 1.

We also have developed a multidimensional meta language called MML (Multidimensional Modeling Language) [HH99a], [HH99b], having the following characteristics:

- MML is an object–oriented language and therefore provides a good basis for flexible, implementation–independent modeling.
- MML meets the needs of conceptual multidimensional models like e. g. distinguishing between dimensional– and fact–classes or providing the possibility to model sophisticated dimensional structures.
- MML enables schema evolution by assigning sets of time intervals to connection elements.

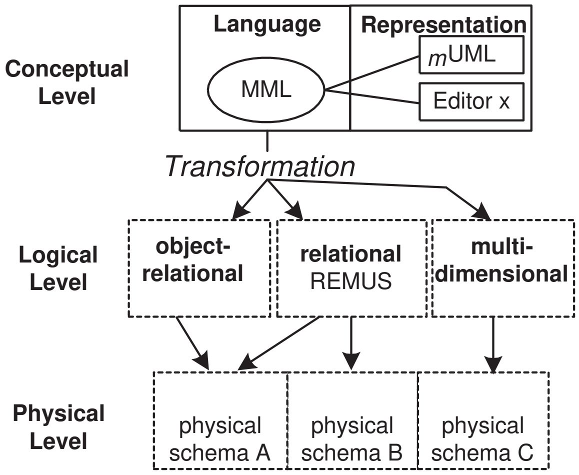

Figure 1: Three–Level–Modeling

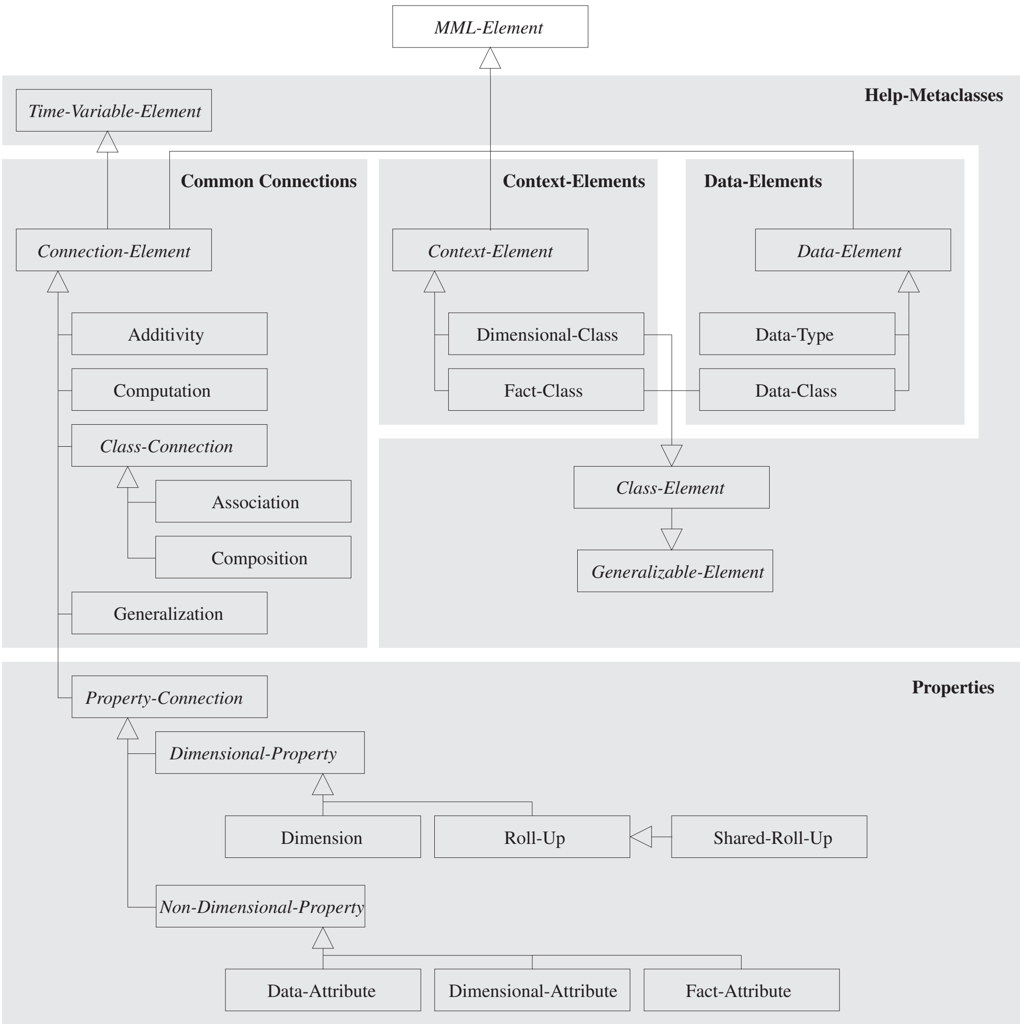

Figure 2: Inheritence hierarchy of the MML

The MML is specified semiformally by an UML–diagram, class descriptions and elucidating prose. The inherintence hierarchy is depicted in figure 2: The *Help-Meta-Classes* provide the basic object–oriented property of inheritence and by the metaclass *Time-Variable-Element* it is possible to assign valid time intervals to MML schema elements. *Data-Elements* provide basic and complex data types. The classes in the area *Context-Elements* introduce multidimensionality by distinguishing classes into *Fact-Classes* and *Dimensional-Classes*. The connections of a MML–diagram are falling into *Common Connections* as known from the object–oriented–world and *Properties* considering special types of connections of the multidimensional model.

With the MML as basis different front end tools can be used (distinction between language and graphical representation). Exemplarily, we have developed an extension of the UML (Unified Modeling Language), called $_{m}$UML (multidimensional UML). By using the concept of stereotypes for extending the UML [RU97] we have defined new stereotypes to model the different types of classes and to mark the connections for building hierarchies. Moreover, the UML extension mechanism of tagged values is used for modeling derived attributes. Figure 3 shows some of the new modeling constructs.

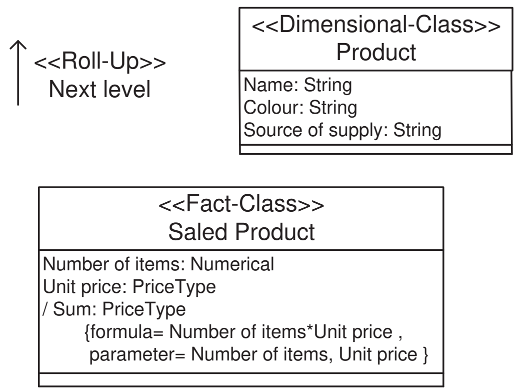

Figure 3: New modeling constructs in the $_{m}$UML

Figure 4 shows an example of an $_{m}$UML schema: in the middle of the picture there is the value of a sale as fact, aggregated by some single items. The dimensions 'product', 'time' and 'location' are placed all round the facts. In the dimensions 'product' and 'time' multiple hierarchies are defined. Furthermore, the edge from 'week' to 'year' has the stereotype shared–roll–up because not every week can be mapped to one year unambiguously. In the dimension 'location' inheritence is used to model the 'point of sale' as general concept and 'branches' and 'department stores' as specializations.

An architecture of the implementation is sketched in figure 5. The $_{m}$UML is realized as an extension of the commercial CASE tool Rational Rose (see figure 6). The MML is implemented as a class library in C++. For storing MML–diagrams persistently an ORACLE database is used at the moment, but we want to change to an extended version of the OIM as soon as possible.

Assuming a relational database, for satisfying the mapping from the conceptual to the logical level we have defined and implemented a mapping [Har99] trans-

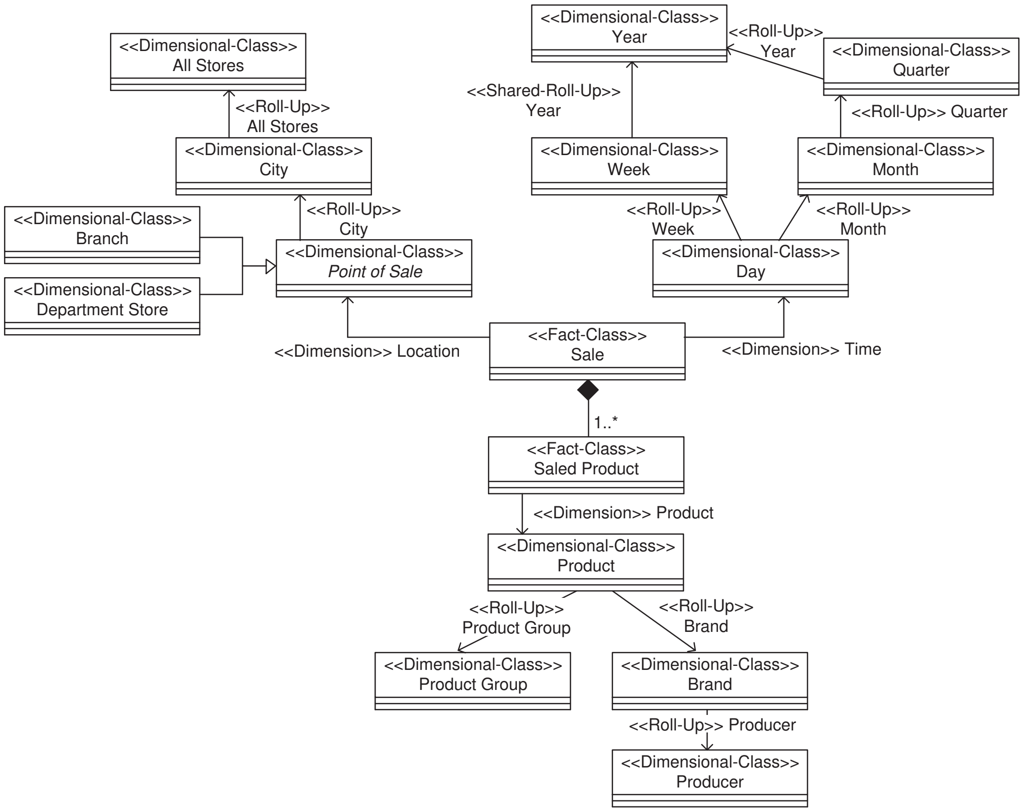

Figure 4: Example of $_m$UML

forming MML–diagrams into a special kind of relational schema, called REMUS (Relational Model for Multidimensional Purpose). Beside relations and attributes, a REMUS schema consists of multifarious meta data. These meta data carry the information of the multidimensional aspects of the MML–diagram which can not be mapped to tables and attributes directly.

The work on the physical design step is just in the beginning. As basic idea a three–step approach should be realized, consisting of the following steps:

- An algorithm transforming a REMUS schema into a basic working database schema. This schema should be independent of the DBMS and OLAP–tools to be used.
- An algorithm transforming the basic working database schema to a schema already considering some special needs of the DBMS and OLAP–tools (e. g. denormalization of hierarchies). But all issues of tuning are left out at this step.
- In the last step of physical database design the schema is extended by applying tuning actions. Again this is a two–stage step. First, under consideration of parameters about the extension (e. g. number of rows in a table), the physical model (e. g. organization of hard disks) and the user behaviour (e. g. query patterns) a set of actions being beneficial to the schema is selected. In the second step a global optimization is done by selecting some of these tuning actions.

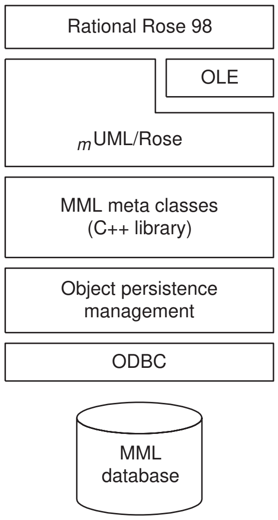

Figure 5: Implementation

Later on in the lifecycle of the DW, the last step of physical design can be repeated when the parameters have changed.

The metadata model is based on the OIM, extended by specific aspects for e. g. storing MML–schemas. Later on, we will extend the model to be able to store details about the physical design process.

5 Rating

In the scope of this work we are developing a design methodology for DWs. The main focus is on conceptual modeling whereby our language MML provides both many constructs to model sophisticated multidimensional structures and object–oriented constructs to describe the world of discourse in a natural way. By distinguishing between language and (graphical) representation on the conceptual layer we are providing freedom to the user in choosing his modeling tool.
The transformation algorithm considers the multidimensional model and leads to a relational schema specialized for DWs. Moreover, it delivers a set of meta data which can be used later on in the design cycle.
The ideas about physical design can not rated at the moment because the work is just in progress.
By using existing tools (Rational Rose) and standards (OIM) we save development effort, get runable prototypes rapidly and can prove our concepts short–term.
As a conclusion, we believe that our design methodology represents a novelty in current research on DW design and we are expecting that the planned evaluation will show the applicability of the work.

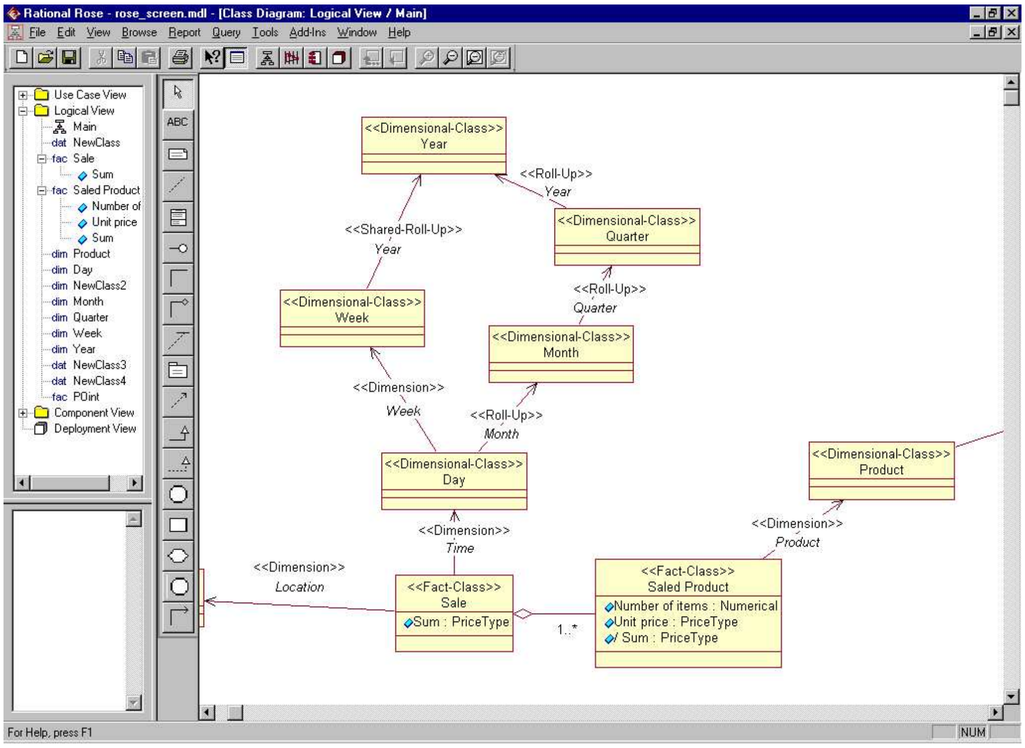

Figure 6: Rational Rose extended for $_m$UML modeling

References

[AW98] M. Abdelguerfi and K.-F. Wong, editors. *Parallel Database Techniques*. IEEE Computer Society, 1998.

[Bay96] R. Bayer. The Universal B-Tree for multidimensional Indexing. *Technical University of Munich (Germany), Technical Report TUM-I9637*, November 1996.

[BG92] D. Bell and J. Grimson, editors. *Distributed Database Systems (International Computer Science Series)*. Addison-Wesley, 1992.

[BM72] R. Bayer and E.M. McCraight. Organization and Maintenance of Large Ordered Indexes. *Acta Informatica*, 1:173–189, 1972.

[BPS97] M. Blaha, W. Premerlani, and H. Shen. Converting OO Models into RDBMS Schema. *IEEE Software*, 11(3):28–39, May 1997.

[Bul96] D. Bulos. A New Dimension. *Database Programming & Design 6/1996*, pages 33–37, 1996.

[Coa00] Meta Data Coalition. *Homepage MDC*. http://www.MDCinfo.com, 2000.

[Cor98] Rational Software Corporation. *Rational Rose 98*. , 1998.

[Cou00] OLAP Council. *MDAPI OLAP Council*. http://www.olapcouncil.com, 2000.

[DG92] D.J. DeWitt and J. Gray. Parallel database systems: The future of high performance database systems. *Communications of the ACM*, 35(6):85–98, 1992.

[DG98] G. Dodge and T. Gorman, editors. *Oracle8 Data Warehousing*. John Wiley & Sons, 1998.

[GHRU97] H. Gupta, V. Harinarayan, A. Rajaraman, and J.D. Ullman. Index Selection for OLAP. In *Proceedings of the Internatl. Conference on Data Engineering*, Binghampton, UK, 1997.

[GM97] H. Gupta and I.S. Mumick. Selection of Views to Materialize Under a Maintenance Cost Constraint. Technical report, Stanford University, 1997.

[GMR98] M. Golfarelli, D. Maio, and S. Rizzi. Conceptual Design of Data Warehouses from E/R Schemes. *Proc. of Hawaii International Conference On System Sciences*, 1998.

[Gra99] C. Grandy. The Art of Indexing. Technical report, DISC - Dynamic Information Systems Corporation, 1999.

[Gro00a] ISO/IEC JTC1/SC32/WG2 IRDS Rapporteur Group. *Homepage IRDS*. http://www.irds.org, 2000.

[Gro00b] Object Management Group. *Homepage OMG*. http://www.omg.org, 2000.

[GS97] H. Gupta and D. Srivastava. Selecting and Maintaining Materialized Views for Message Management. Technical report, Stanford University, 1997.

[Gup97] H. Gupta. Selection of Views to Materialize in a Data Warehouse. In *Proceedings of the Internatl. Conference on Database Theory*, Athens, Greece, 1997.

[Har99] A. Harren. *Konzeptionelles Data Warehouse-Design (In German)*. Diploma Thesis, University of Oldenburg, Germany, 1999.

[Her99] O. Herden. *Homepage ODAWA*. ODAWA Homepage http://odawa.offis.uni-oldenburg.de, 1999.

[HH99a] A. Harren and O. Herden. Conceptual Modeling of Data Warehouses. In *Proc. of Demonstration and Posters E/R99*, Paris (France), November 1999.

[HH99b] A. Harren and O. Herden. MML und $_m$UML – Sprache und Werkzeug zur Unterstuetzung des konzeptionellen Data Warehouse–Designs (In German). In *Proceedings DMDW99*, Magdeburg (Germany), September 1999.

[Inc00] Sybase Inc. *Sybase Warehouse Architect*. http://www.sybase.com, 2000.

[Inm92] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, Inc., 1992.

[Kim96] R. Kimball. *The Data Warehouse Toolkit*. John Wiley & Sons, Inc., 1996.

[KN99] K. Kuespert and J. Nowitzky. Partitionierung von Datenbanktabellen (In German). *Informatik Spektrum*, 22(2):146, 1999.

[MBB99a] V. Markl, M. Bauer, and R. Bayer. Improving OLAP Performance by Multidimensional Hierarchical Clustering. In *Proceedings of IDEAS Conf.*, Montreal (Canada), August 1999.

[MBB99b] V. Markl, M. Bauer, and R. Bayer. Variable UB-Trees: an efficient way to accelerate OLAP queries. In *Proceedings DMDW99*, Magdeburg (Germany), September 1999.

[Mic00a] Microsoft. *Description OIM*. http://www.microsoft.com/technet, 2000.

[Mic00b] Microstrategy. *Microstrategy DSS Architect*. http://www.microstrategy.com/-products/Architect/index.htm, 2000.

[MWM99] D. Munneke, K. Wahlstrom, and M. K. Mohania. Fragmentation of Multidimensional Databases. In *Proceedings Australasian Database Conference 1999*, Auckland (New Zealand), January 1999.

[OQ97a] P. O'Neil and D. Quass. Improved Query Performance with Variant Indexes. In *SIGMOD'97*, Tucson, Arizona, USA, 1997.

[OQ97b] P. O'Neill and D. Quass. Improved Query Performance with Variant Indices. In *Proceedings of SIGMOD Conference*, 1997.

[OV91] M.T. Oeszu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[Pro00] SMART Project. *Homepage SMART (Supporting Metadata for Data Warehousing Systems)*. http://www.ifi.unizh.ch/dbtg/Projects/SMART, 2000.

[RBP+93] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object–oriented Modeling and Design*. Prentice Hall, 1993.

[RS91] S. Rozen and D. Shasha. A framework for automating physical database design. In Guy M. Lohman, Amílcar Sernadas, and Rafael Camps, editors, *17th International Conference on Very Large Data Bases, September 3-6, 1991, Barcelona, Catalonia, Spain, Proceedings*. Morgan Kaufmann, 1991.

[RU97] Rational Software Corporation and UML partners. *UML Semantics. Version 1.1*. Object Management Group. OMG Document ad/97-08-04, September 1997.

[SBHD98] C. Sapia, M. Blaschka, G. Hoefling, and B. Dinter. Extending the E/R Model for the Multidimensional Paradigm. *Proc. International Workshop on Data Warehouse and Data Mining*, November 1998.

[Wed74] H. Wedekind. *On the Selection of Access Paths in a Data Base System. Database Management.* North–Holland, 1974.