

TDWI World Conference Chicago  
19 May 2006  
SAP Business Information Warehouse

Dr. Michael Hahne

## Main Agenda

- Multidimensional Data Structures
- Conceptual Design
- Data Model of SAP Business Information Warehouse
- Modeling hierarchies in SAP BW
- Model Management
- Enterprise Data Warehousing with BW

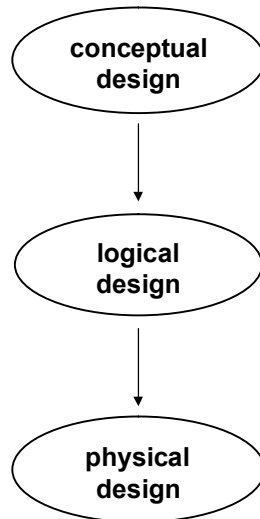
## Agenda: Multidimensional Data Structures

- Steps of data modeling
- Elements of multidimensional data structures
- Hierarchical dimension structures
- Temporal aspects
- Guidelines for modeling

## Agenda: Multidimensional Data Structures

- Steps of data modeling
- Elements of multidimensional data structures
- Hierarchical dimension structures
- Temporal aspects
- Guidelines for modeling

## Steps of data modeling

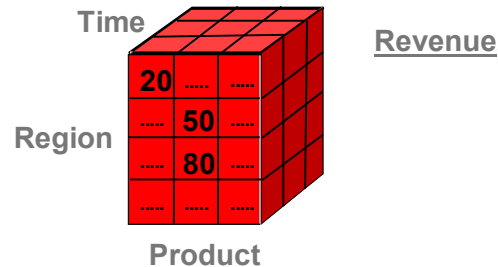


## Agenda: Multidimensional Data Structures

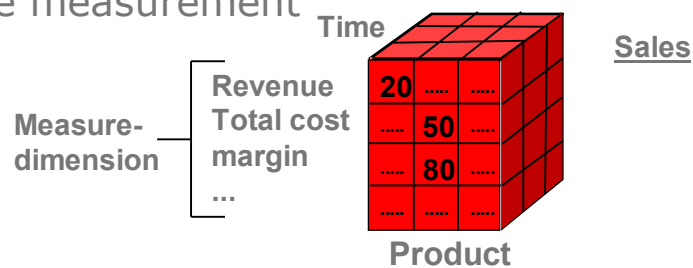
- Steps of data modeling
- Elements of multidimensional data structures
- Hierarchical dimension structures
- Temporal aspects
- Guidelines for modeling

## Key performance indicators in cubes

- One measure per cube



- Performance measurement system



7

TDWI World Conference Chicago '06, Michael Hahne

The increasing focus of Information Technology on data analysis and decision support has been accompanied by the development of innovative technical concepts in Data Warehousing, OLAP and Data Mining. Modeling possibilities determine the efficiency and success of such systems.

Multidimensional views of enterprise-internal and -external data deliver a useful managerial view of the enterprise, examining economically relevant circumstances in terms of numerous factors. The revenue for a product group in certain sales regions over a specific period is an example.

The arrangement of economic variables relating to indicators like revenue or gross sales is organized along various dimensions, such as customers, articles and regions. This structuring is considered to be a suitable decision-oriented view of economic facts. Figuratively speaking, quantitative characteristics are stored in multidimensional cubes, whose edges are marked and defined by individual dimensions.

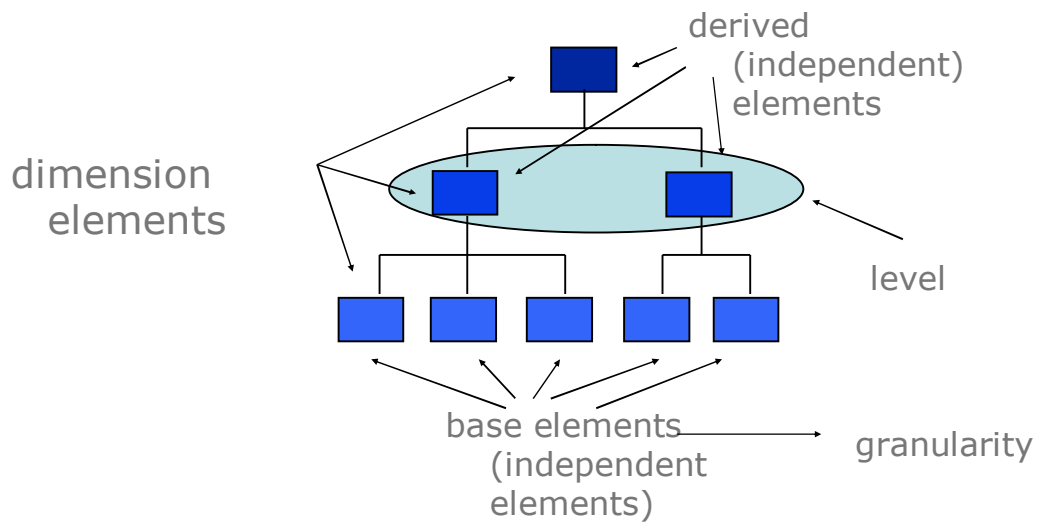
### References:

Torlone, R.: Conceptual Multidimensional Models. In : Rafanelli, M. (Ed.) : Multidimensional Databases : Problems and Solutions, 2003, pages 69-90.

## Dimensions

- Role:  
Concretizing and qualifying of measures
- Content:  
Similar objects relating to the same aspect of business
- Variants:
  - Untyped dimension (determined by elements)  
Few dimension elements (e.g. scenario or measure dimension)
  - Typed dimension (determined by levels)  
Huge dimensions (e.g. customer or material dimension)  
Characterization by set of objects

## Constituent components of dimensions



9

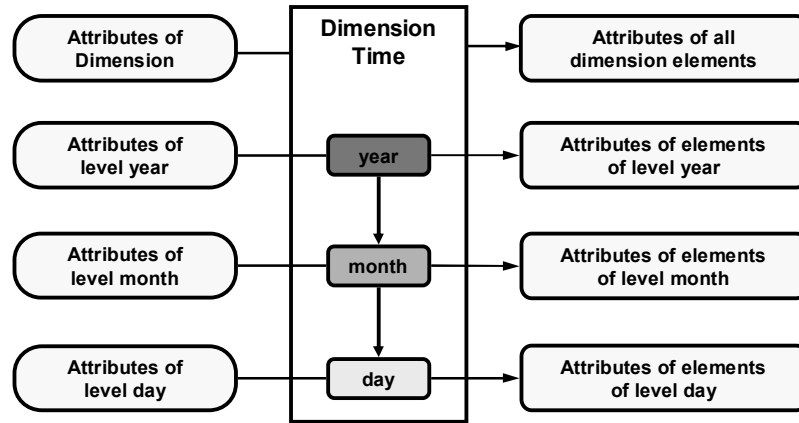
TDWI World Conference Chicago '06, Michael Hahne

There are two types of dimensions. On the one hand, there are small dimensions resulting from the listing of individual elements, and on the other there are the dimensions in which elements are grouped into classes or "levels". Thus we have *dimensions defined by elements* and *dimensions defined by levels*. The constituent components of a dimension structure for dimensions defined by levels are shown in figure 1.

References:

Pourabbas, E.; Rafanelli, M.: Hierarchies. In : Rafanelli, M. (Ed.) : Multidimensional Databases : Problems and Solutions, 2003, pages 91-115.

## Attributes



## Types of dimensions

- “Standard” (e.g. Material, Customer, Region)
- Measure Dimension
- Scenario dimension (Actual, Budget, Forecast, ...)
- Time dimension

The temporal aspect of data in multidimensional models is fundamental. This is reflected in the special meaning of the time dimension, since fact and transaction data are inherently timestamped by this dimension.

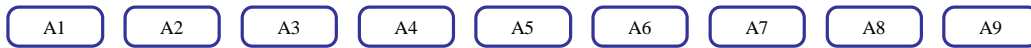
References:

Stock, S.; Beekmann, F.: Temporal aspects of data warehousing. In (Jardim-Gonçalves; Cha, J.; Steiger-Garção Hrsg.): Concurrent Engineering – The Vision for the Future Generation in Research and Applications, Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications, 26-30 July 2003, Madeira, Portugal, A.A. Balkema Publishers, Lisse et al 2003, 53-59.

## Agenda: Multidimensional Data Structures

- Steps of data modeling
- Elements of multidimensional data structures
- Hierarchical dimension structures
- Temporal aspects
- Guidelines for modeling

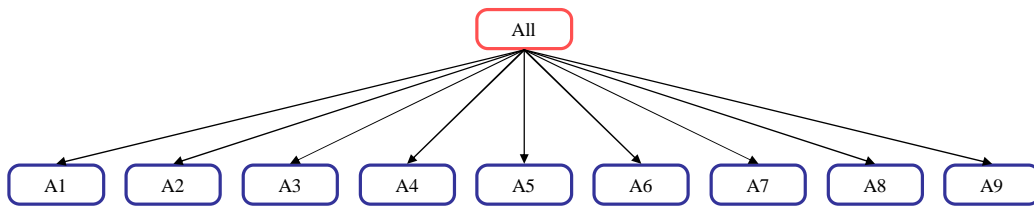
# Flat structure



## Typed flat structure

A

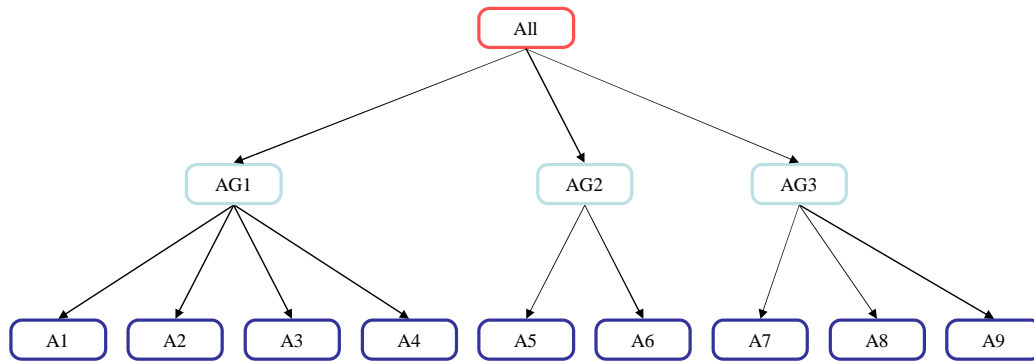
# Flat structure with "total"



# Typed flat structure with "total"



## Tree structure



17

TDWI World Conference Chicago '06, Michael Hahne

The depth of the nodes correspond to their type.

Definitions:

Depth: starting at root with 0

Generation: starting at root with 1

Level: starting at leaves with 0

Multidimens. Conceptional BW Model BW Hierarchy Management EDW Contact

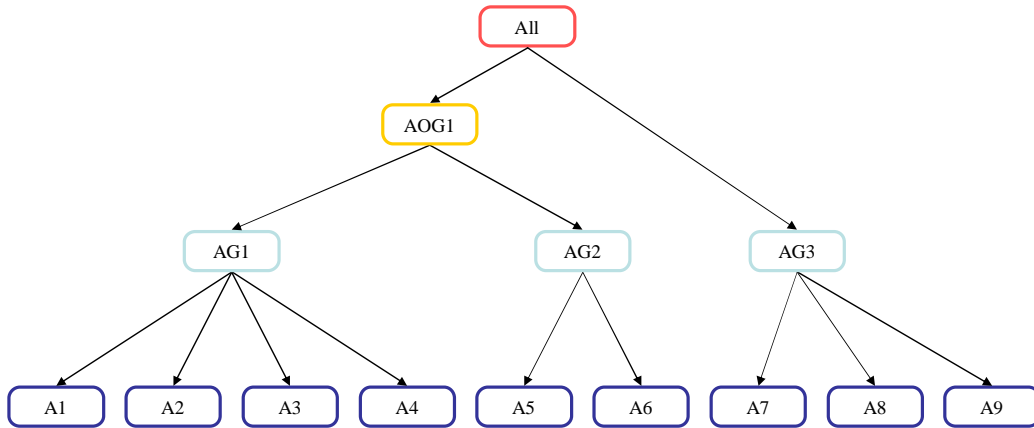
## Typed tree structure

```
graph TD; All[All] --> AG[AG]; AG --> A[A];
```

18 TDWI World Conference Chicago '06, Michael Hahne

Linear list with three nodes.

## Unbalanced tree structure - variant I

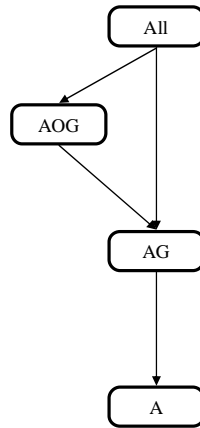


19

TDWI World Conference Chicago '06, Michael Hahne

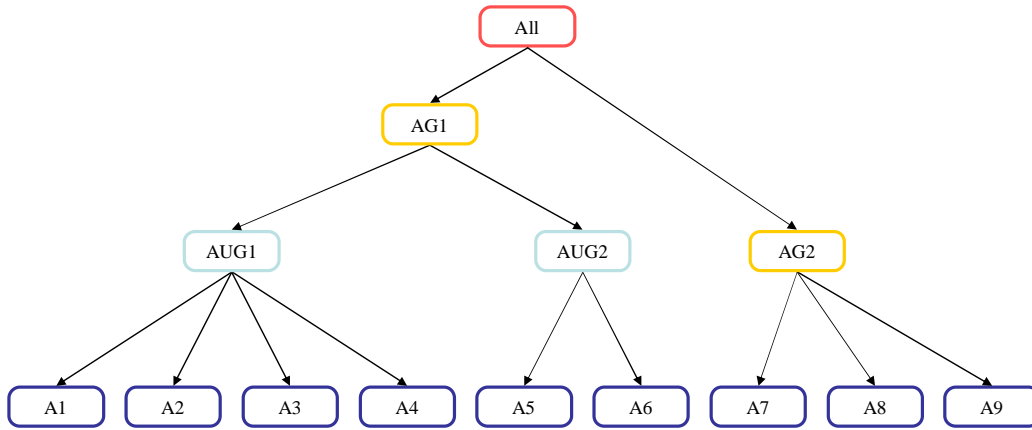
The various colors correspond to the node classes (types).

## Typed unbalanced tree structure - variant I

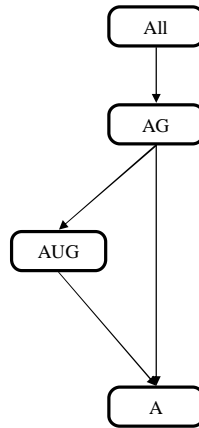


Typing the unbalanced structure results in a variant of this triangular arrangement.

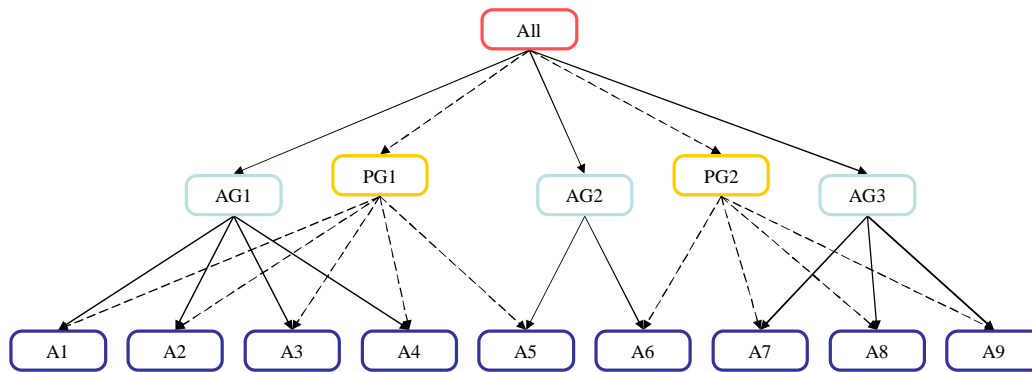
# Unbalanced tree structure - variant II



## Typed unbalanced tree structure - variant II



## Parallel hierarchy

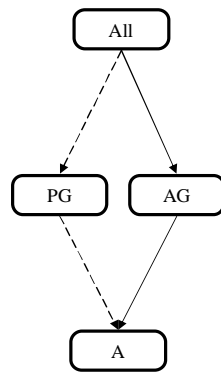


23

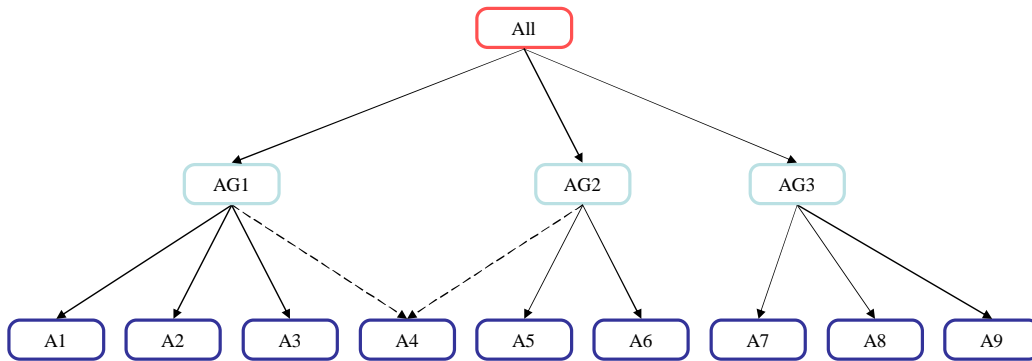
TDWI World Conference Chicago '06, Michael Hahne

In this structure, nodes with the same assigned value for level, depth or generation can belong to in different classes. In particular, the level attribute alone (as it is used in most database solutions) is not sufficient to determine the levels of the hierarchy.

## Typed parallel hierarchy



## Heterarchy (many:many relationship)



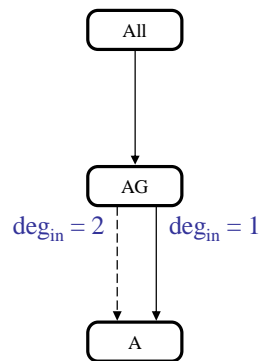
25

TDWI World Conference Chicago '06, Michael Hahne

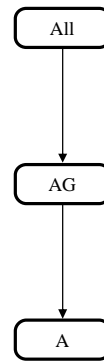
The partitioning of nodes is not affected by the input degree.

## Typed heterarchy

### Distinguishing the “cardinality” of edges

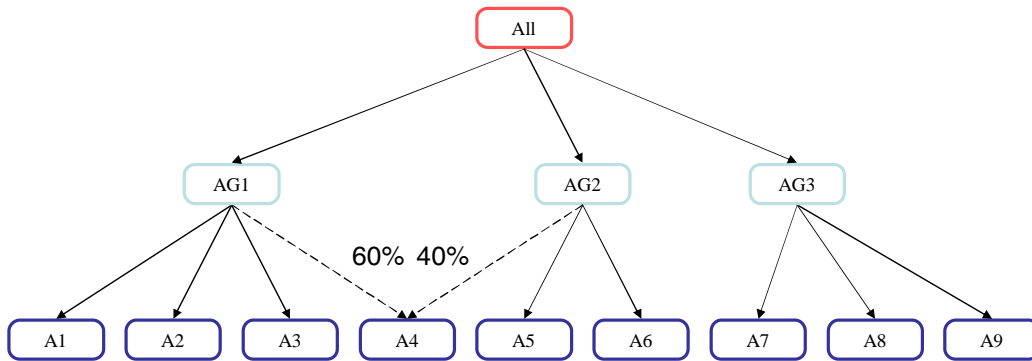


### Without distinguishing the “cardinality” of edges



The typing of a hierarchy can be done in two ways. If the incoming degree is recognized, this would result in two edges in the example provided. Otherwise, only one edge will result. This difference has no affect on navigation along the hierarchy.

# Pro rata consolidation



## Additivity

### Aggregation in the case of nonadditive facts

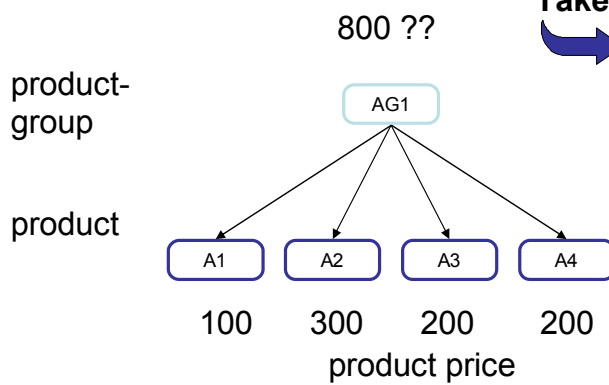
e.g. price, stock of inventory

**May be reasonable:**

→ calculation of average or last value

**Take care:**

→ Semiadditive facts like closing stocks can be added along the storage location dimension but not along the time dimension



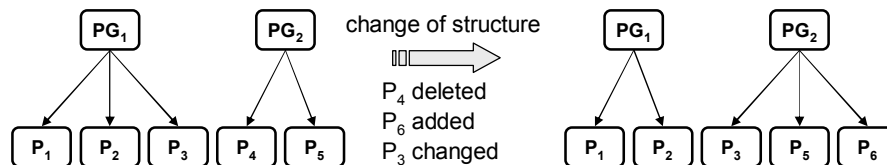
## Agenda: Multidimensional Data Structures

- Steps of data modeling
- Elements of multidimensional data structures
- Hierarchical dimension structures
- **Temporal aspects**
- Guidelines for modeling

### References:

- Kimball, R.: The Data Warehouse Toolkit. John Wiley & Sons Inc., 1996.
- Bliujute, R.; Saltenis, S.; Slivinskas, G.; Jensen, G.: Systematic Change Management in Dimensional Data Warehousing, Technical Report TR-23, Time Center, Computer Science Department, University of Arizona, 1998.
- Mendelzon, A.O.; Vaisman, A.: Temporal queries in OLAP. Proceedings of the 26th International Conference on Very Large Databases (VLDB'00), Cairo, Egypt, 2000, 242-253.
- Yang, J.; Widom, J.: Maintaining temporal views over non-temporal information sources for data warehousing. Proceedings of the Sixth International Conference on Extending Database Technology, Valencia, Spain, 1998, 389-403.
- Yang, J.; Widom, J.: temporal view self-maintenance in a warehousing environment, Proceedings of the 7th International Conference on Extending Database Technology, Konstanz, Germany, 2000, 345-412.
- Hurtado, C.; Mendelzon, A.O.; Vaisman, A.: Maintaining data cubes under dimension updates. Proceedings of 1999 International Conference on Data Engineering (IEEE/ICDE'99), Sydney, Australia, 1999, 346-355.
- Hurtado, C.; Mendelzon, A.O.; Vaisman, A.: Updating OLAP dimensions. Proceedings of the 1999 Workshop on Data Warehousing and OLAP (DOLAP'99), Kansas City, 1999, 60-66.
- Pedersen, T.B.; Jensen, C.: Multidimensional data modeling for complex data. Proceedings of 1999 International Conference on Data Engineering (IEEE/ICDE'99), Sydney, Australia, 336-345.
- Stock, S.; Beekmann, F.: Temporal aspects of data warehousing. In (Jardim-Gonçalves; Cha, J.; Steiger-Garção Hrsg.): Concurrent Engineering – The Vision for the Future Generation in Research and Applications, Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications, 26-30 July 2003, Madeira, Portugal, A.A. Balkema Publishers, Lisse et al 2003, 53-59.

## Slowly changing dimensions



The question of how to handle structural changes in dimensions, which arises again and again during the modeling process, is crucial to the structuring of Data Warehouse and OLAP systems because it determines what kinds of analysis will be possible.

It is sufficient to consider dimensions and their hierarchies in order to justify time stamping in the conceptual model, since all facts in a data warehouse have a relation to time by virtue of the time dimension. This problem was first described by Kimball, who discussed it using the expression "slowly changing dimensions". He proposed a solution for handling changes in consolidation trees whereby dimensions are timestamped with their validity interval.

The problem of temporal OLAP has been addressed by Bliujute et al. at the Time Center at the University of Arizona. They compared the performance of various SQL queries with approaches using fact tables that represent time series, fact tables that are event-oriented, and the Kimball approach. Beyond that, Mendelzon and Vaisman responded to the need for a temporal query language for OLAP by introducing TOLAP. They also defined a temporal multidimensional data model.

Temporal views and their maintenance are the focus of some recent work of Yang & Widom. They developed a view definition language that operates with non-temporal data sources. A formal model of dimension updates resulting from modifications in hierarchical structures in a multidimensional model has been developed by Hurtado et al. They also deal with the question of an implementation of dimension updates in ROLAP storage for the data warehouse.

Pedersen & Jensen introduced a multidimensional model that is able to handle complex data and that also considers temporal aspects. These are considered as one of a number of data modeling problems.

The structural changes in consolidation hierarchies result from changing, deleting and adding nodes, as represented by fig. 2 for the case of a two-level hierarchy with product groups and products. A change of the mapping to higher and subordinated elements in the hierarchy occurs where there is a change to a node of the hierarchy. The consolidation hierarchy is also changed through addition of new nodes and deletion of existing nodes.

## Possible reporting requirements

- As is: report with current structure
- As of: report with the structure as it was on a fixed date (point in time)
- As posted: report showing the "historical truth" (transaction-oriented view)
- Report with comparable results (time slice)

The analysis of data comparing the current period with a historical period or to the "historical truth" is a very common reporting requirement. Another reporting requirement might be to only consider those elements of the structure that are valid for the entire reporting period ("not to compare apples with oranges").

References:

Hahne, M.: Time aspects in SAP Business Information Warehouse. In (Jardim-Gonçalves; Cha, J.; Steiger-Garção Hrsg.): Concurrent Engineering – The Vision for the Future Generation in Research and Applications, Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications, 26-30 July 2003, Madeira, Portugal, A.A. Balkema Publishers, Lisse et al 2003, 69-74.

## Slowly changing dimensions - example

### Product dimension as of 2005-04

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

### Product dimension as of 2005-05

Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

### fact table

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

To clarify the effects of these structural changes, they are explained along with possible reporting requirements using a simple numerical example. All three types of change to the hierarchy shown in this slide occurred at the transition from a fictitious time period 1 to a second fictitious time period 2. The fictitious values for the corresponding accounting periods are also shown in this slide.

## Reporting requirements - example

### Reporting current structure

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	300	400

### Reporting as of (2005-04)

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	200	200
PG Y	200	200

### Reporting as posted

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	200	100
PG Y	200	400

### Reporting comparable results

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	200	200

Listed here are the results of the computation for the product groups, representing different reporting results to respond to different requirements in handling structural changes for each of the four scenarios.

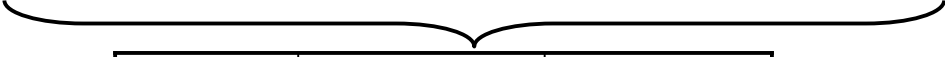
## As is: Report with the current structure

**Product dimension as of 2005-05**

Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

**fact table**

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100



Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	300	400

The valid structure differs not only in the new element Product E but also in the rearrangement of Product B in the second period.

## As Of: Report with the structure as it was on a fixed date

**fact table**

**Product dimension as of 2005-04**

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	200	200
PG Y	200	200

This slide presents computation on the basis of the dimension structure valid for the earlier time period 1. In this scenario, values for Product E cannot be considered.

## As posted: Report the "historical truth"

### Product dimension as of 2005-04

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

### Product dimension as of 2005-05

Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

### fact table

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	200	100
PG Y	200	400

Computation for the "historical truth" clearly differs from the two other points of view in that the transactional data view becomes significant. Computation is always performed on the basis of the structure valid for each case.

## Report with comparable results

### Product dimension as of 2005-04

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

### Product dimension as of 2005-05

Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

### fact table

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	200	200

All three variants have the disadvantage that computation does not lead to directly comparable values for time series analyses. For this reason, it is sometimes necessary to take up only elements that are valid for the entire query period.

## Solutions for structural changes in dimensions

- **Modifying historical data to match new structures**  
**Pros:** Small data sets; data structures remain clear  
**Cons:** Old structures are lost even if users want to analyze data with old structures
- **Save historical data separately in addition to all data matched to new structures**  
**Pros :** Old reports can be reproduced  
**Cons :** Higher volume of data; update processes are complicated and costly if new data needs to be analyzed according to old structures; confusing for end-users
- **Build parallel hierarchies with old and new structures**  
**Pros :** All data can be analyzed with any structure  
**Cons :** Dimension structure is very confusing
- **Temporal databases - timestamps**  
**Pros :** All data can be analyzed with any structure  
**Cons :** Poor performance; concepts not fully developed

The following four possibilities for the implementation of the different report requirements are available:

Adjustment of historical data to new structures.

Separate storage of the historical data in addition to the complete new structures.

Parallel hierarchies with old and/or new structures.

Temporal databases and timestamps.

Adjusting the historical data has the advantage that the volume of data is not increased and the data dimension structures remain visible. Not all report requirements can be covered because the old structures are lost.

Old evaluations can be retrieved by employing separate storage. This involves larger volumes of data and more complex update processes.

The parallel hierarchies alternative offers the advantage that all numbers with arbitrary structures can be indicated. The developing dimension structures are rather unclear.

Versioning timestamps provide maximum flexibility for reporting systems with arbitrary historical variants of structures. However, performance tends to be worse, and the concepts are still not fully developed.

Temporal databases store versions of objects in order to track the evolution of these objects over time, introducing temporal attributes of versions called timestamps. These represent either a point in time or an interval, and the time axis is composed of a series of discrete time units. The smallest relevant time unit determines the granularity of time measurement and is called a chronon.

In information systems that recognize temporal aspects, we can distinguish between "valid time" and "transaction time" for timestamping. Whereas the valid time of an object or a fact is defined as the interval during which this object has a constant state, the transaction time is the time when a fact is current and retrievable in the database. Databases that recognize either valid time or transaction time are called temporal databases.

References:

Chamoni, P.; Stock, S.: Temporal structures in data warehousing, Proceedings on the 1st international conference on data warehousing and knowledge discovery, Florence, 1999, 353-358.

Jensen, C. et al.: A Consensus Glossary of Temporal Database Concepts. In: Sigmod Record, 23 (1994) 1, 52-64.

Snodgrass, R.; Ahn, I.: A Taxonomy of Time in Databases. In: Sigmod Record, 14 (1985) 4, 236-246.

## Agenda: Multidimensional Data Structures

- Steps of data modeling
- Elements of multidimensional data structures
- Hierarchical dimension structures
- Temporal aspects
- **Guidelines for modeling**

## Guidelines for conceptual design of dimensions

- Number of dimensions in a cube should be between four and ten (best is between six and eight)
- Number of levels should be small (a maximum of seven is good)
- Not too many children per consolidation element (a maximum of fifteen to twenty)
- Objects that define dimensions:
  - 1:1 relationship unsuitable
  - 1:many relationship most suitable for dimension hierarchy
  - many:many relationship should be modeled as two dimensions

## Agenda: Conceptual Design

- Methods for conceptual design
- Conceptual modeling with ADAPT
- Tool support by Microsoft Visio™
- Tool support by SemTalk™

## Agenda: Conceptual Design

- **Methods for conceptual design**
- Conceptual modelling with ADAPT
- Tool support by Microsoft Visio™
- Tool support by SemTalk™

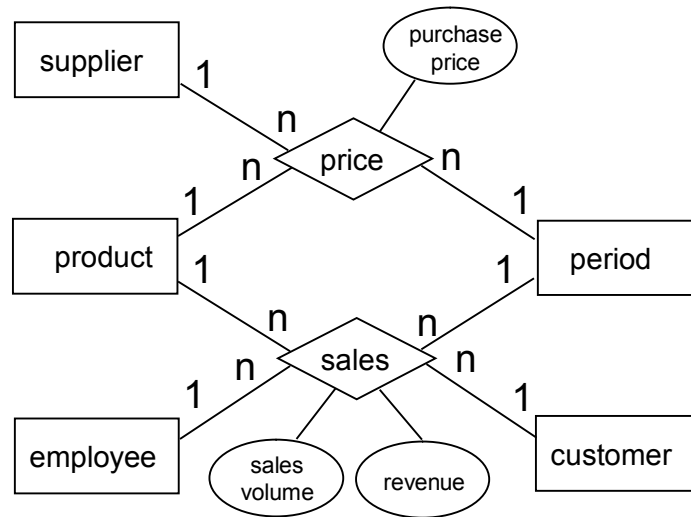
## Conceptual modeling approaches

- Entity Relationship Model
- Multidimensional Entity Relationship Model (ME/R-Model)
- Dimensional Fact Modeling (Golfarelli)
- Dimensional Modeling (Kimball) / Star Schema
- Multidimensional Data Model (Cabibbo / Torlone)

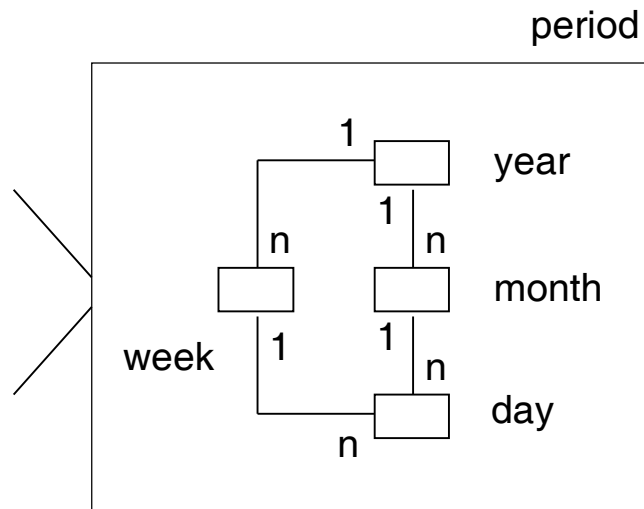
Different approaches to the conceptual modeling of multidimensional information systems have been discussed in the literature. Apart from modifications of entity relationship models and object-oriented approaches, also suggestions with an own methodology arise.

Modeling of analysis-oriented information systems requires special care because this will determine subsequent analytical possibilities. Therefore, a suitable description of possibilities is required in the conceptual phase.

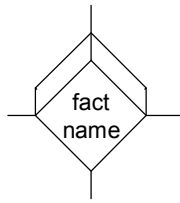
## Entity Relationship Model



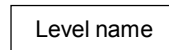
## Extension eER for modeling hierarchies



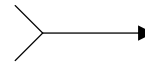
## Multidimensional Entity Relationship Model (ME/R-Model)



**Fact  
Relationship Type**



**Dimension Level  
Entity Type**

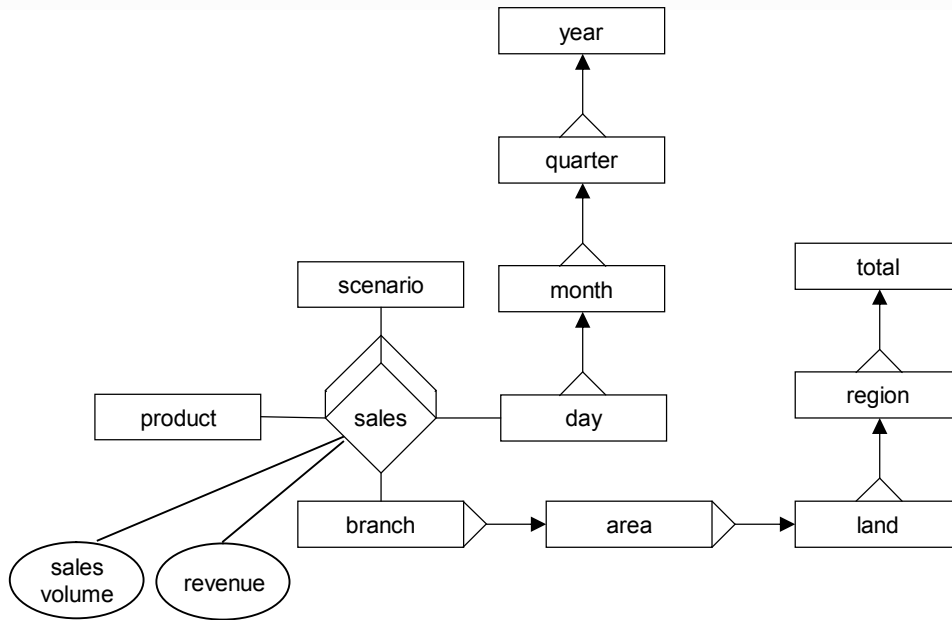


**Roll-Up  
Relationship Type**

reference: Sapia, Carsten; Blaschka, Markus; Höfling, Gabriele.; Dinter, Barbara: Extending the E/R Model for the Multidimensional Paradigm.

In: Proceedings of the Advances in Databases Technologies Conference, November 19-20, 1998, Singapore

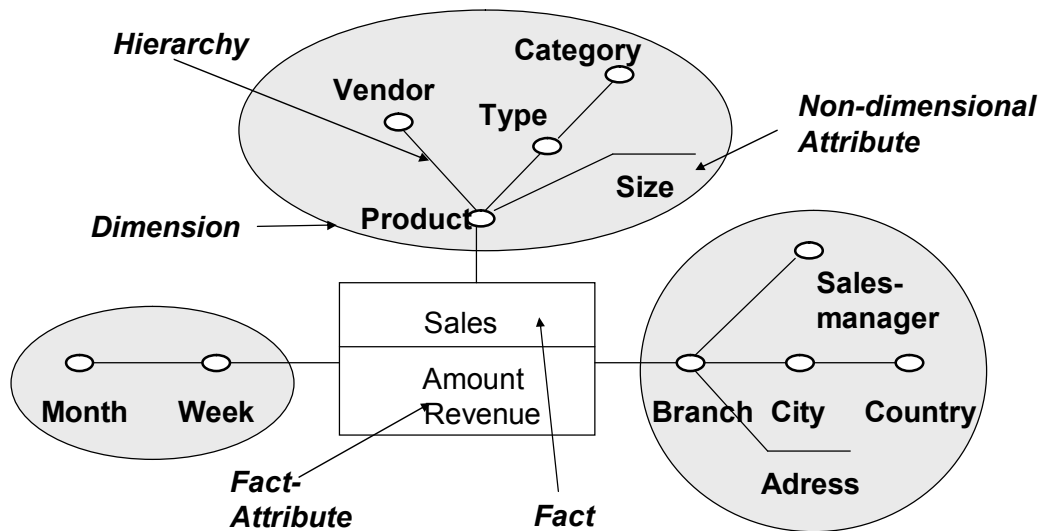
## Example of a ME/R-Model



## ERM: useful to a certain extent

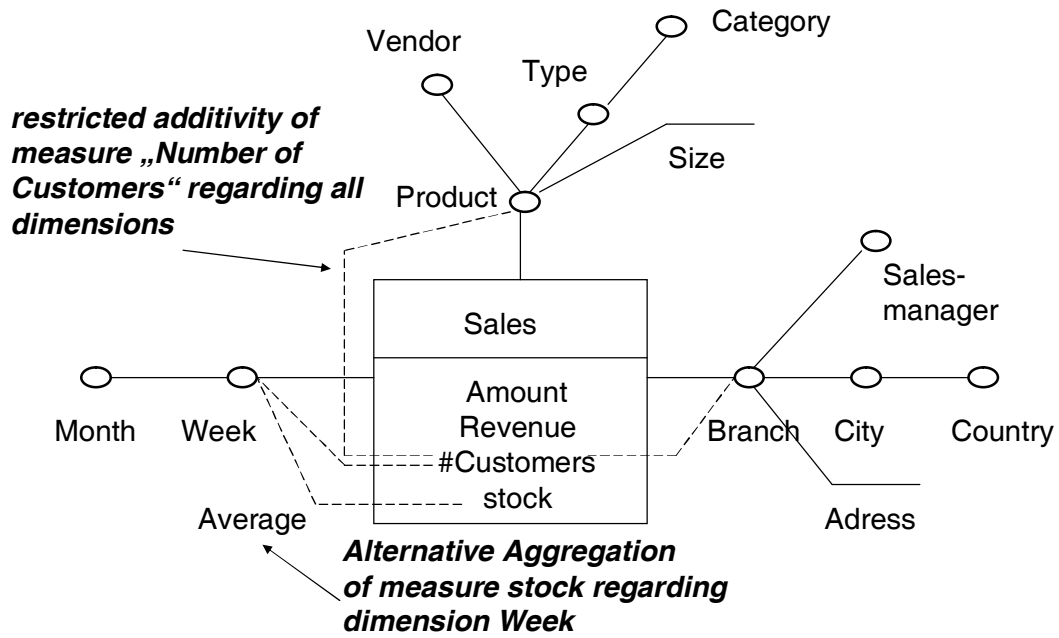
- Insufficient ability of handling single objects (Measures, actual/budget/forecast)
- Focus on classes of objects
- No support of connections between single objects (relative measures, calculation of measures)

## Dimensional Fact Modeling

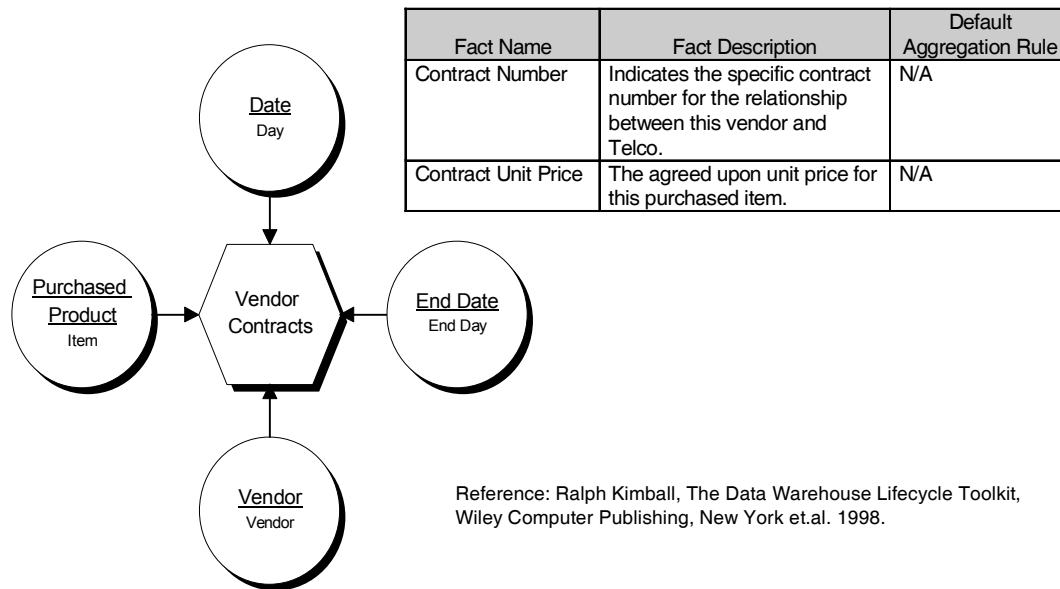


Reference: Golfarelli, Matteo; Maio, Dario; Rizzi, Stefano: Conceptual Design of Data Warehouses from E/R Schemas. In: Proceedings of the Hawaii International Conference On System Sciences, January 6-9, 1998, Kona, Hawaii.

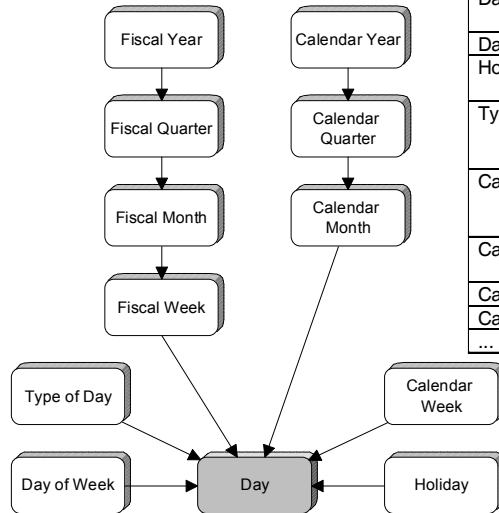
## Visualize aggregation with Dimensional Fact Modeling



## Dimensional Modeling (Facts)

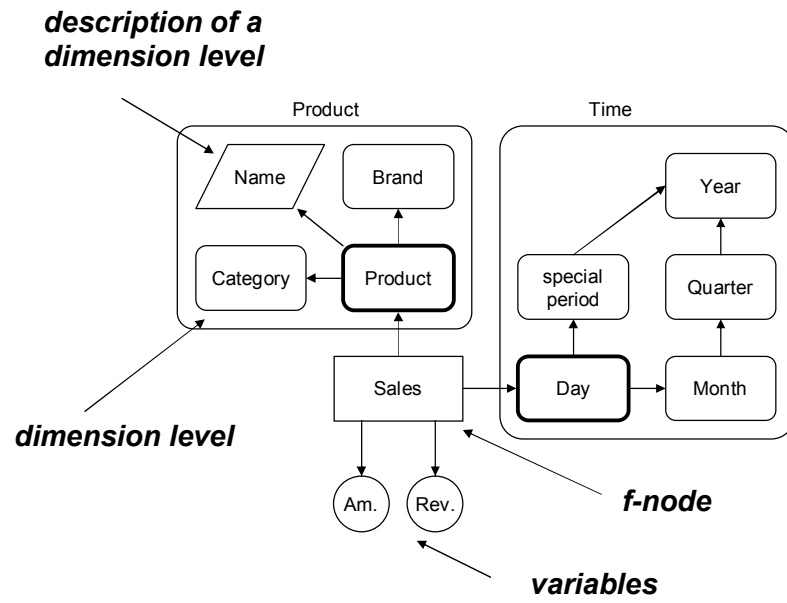


## Dimensional Modeling (Dimensions)



Attribute Name	Attribute Description	Sample Values
Day	The specific day that an activity took place.	06/04/1998; 06/05/1998
Day of Week	The specific name of the day.	Monday; Tuesday
Holiday	Identifies that this day is a holiday.	Easter; Thanksgiving
Type of Day	Indicates whether or not this day is a weekday or a weekend day.	Weekend; Weekday
Calendar Week	The week ending date, always a Saturday. Note that WE denotes week ending.	WE 06/06/1998; WE 06/13/1998
Calendar Month	The calendar month.	January, 1998; February, 1998
Calendar Quarter	The calendar quarter.	1998Q1; 1998Q4
Calendar Year	The calendar year.	1998
...	...	...

## Multidimensional Data Model (MD)



## Agenda: Conceptual Design

- Methods for conceptual design
- **Conceptual modeling with ADAPT**
- Tool support by Microsoft Visio™
- Tool support by SemTalk™

## Objects in ADAPT

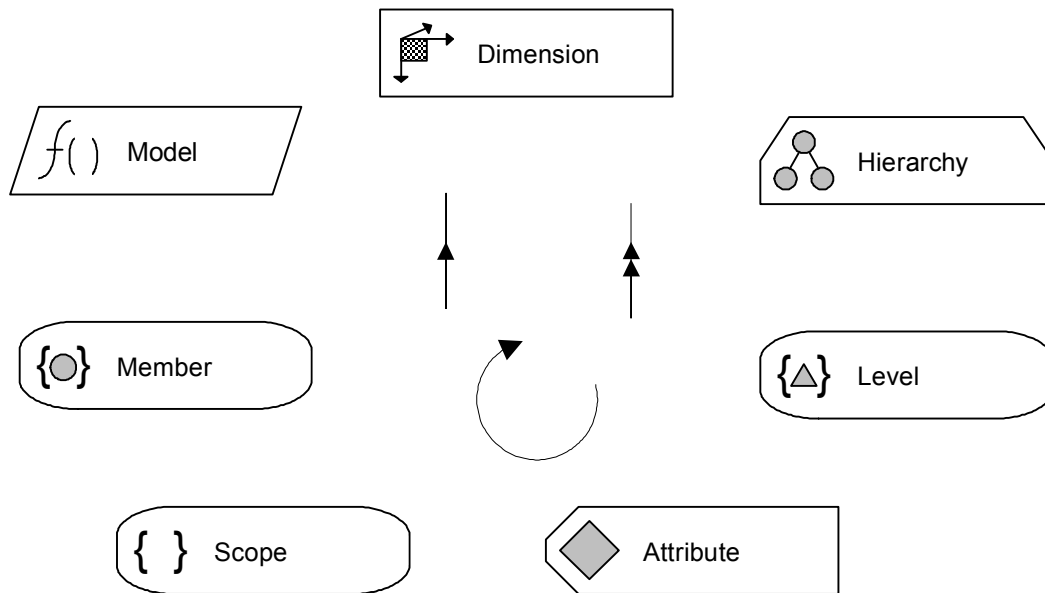
- ADAPT: Application Design for Analytical Processing Technologies
- Objects for modeling of dimensions
- Objects for modeling of cubes
- Objects representing relationships

The modeling of analysis-oriented information systems requires special care, because it will determine subsequent analytical possibilities. For this reason, suitable descriptions of possibilities are required even in the conceptual phase. ADAPT (Application Design for Analytical Processing Technologies) is an acknowledged method for achieving this. ADAPT was introduced by Bulos in 1996. ADAPT is a modeling approach which explicitly focuses on multidimensional data structures for OLAP applications.

References:

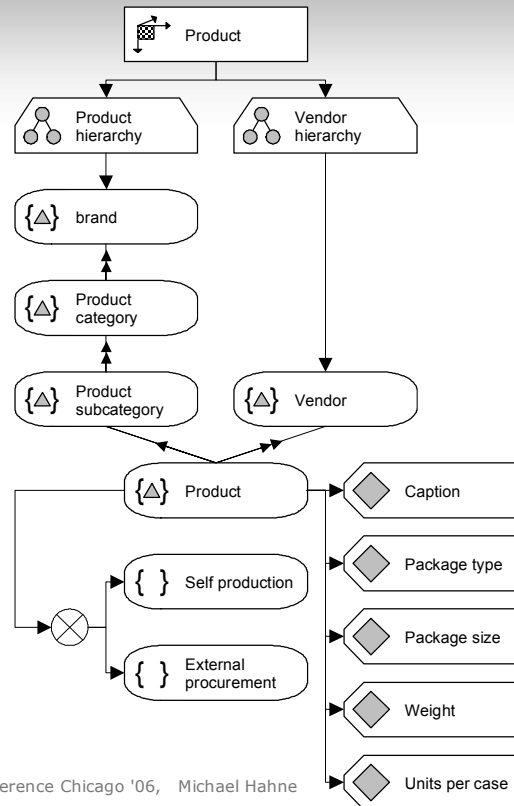
Bulos, D.: A New Dimension. In: Database Programming & Design 9 (1996) 6, 33-38.

## Objects for modeling of dimensions in ADAPT



The strong point of ADAPT is its representation of various dimension structures and their economic references. The individual components for modeling dimensions are summarized here along with the hierarchical structures defined on them.

## Typed dimension



57

TDWI World Conference Chicago '06, Michael Hahne

The illustration of the components used for representing dimensions in ADAPT is based on an example of a (partial) product dimension. Consolidation in the product hierarchy is done by brands, product categories and product sub-categories.

Aggregation to the different manufacturers of the products is also considered. Product caption, packing size and type as well as weight are also relevant along with the actual product number. They are attached to the ADAPT model as attributes of the product level. Self-production and external procurement of products are differentiated by means of the dimension scope object.

According to basic multidimensional principles, a dimension features one or more hierarchies. These are based on superordination and subordination conditions, and form the consolidation paths in dimensions. They are thus the basis for drill-down and roll-up operations. The illustration shows that two hierarchies are defined for products, and these are developed differently in each case.

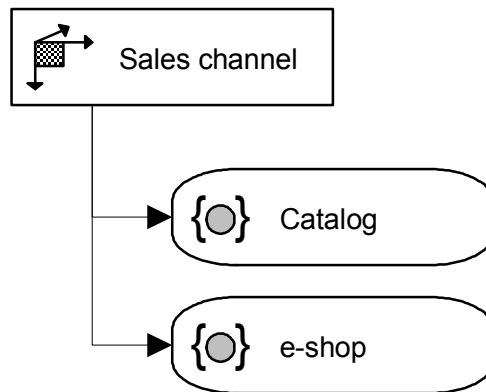
The illustration shows a dimension defined by levels, so that all hierarchies are modelled on dimension levels. These different hierarchies are often called parallel hierarchies. The common foundation level is represented by the object for the product level, which specifies the finest granularity in this dimension.

The vendor hierarchy is an initial simple case, with only an aggregation from products to vendors. Exactly one vendor is assigned to each product.

The double arrow used to show the connection of the levels clarifies this strict relationship; therefore, exactly one superordinate element is assigned to each product. In the other hierarchy, in which the level's brand, category and subcategory are differentiated, this is not the case. However, there can also be products which are assigned to no subcategory but are rather directly connected to a category. In ADAPT, this connection is represented by a single-headed connecting arrow.

Attributes of dimensions or dimension levels are another important object for the representation of multidimensional structures on the semantic level. The attributes are exclusively on the lowest level of all products in the example, and model further characteristics of products, such as weight, along with further packing-related information. In the modelling phase, it often must be decided whether various economic objects should be modeled as attributes or as dimensions in themselves.

## Non-typed dimension



Along with level-determined dimension types, it is also possible to have dimensions specially developed for the type of object. Typically, these are dimensions for scenarios and measures [the element-defined type of dimension]. Here the example shows consideration of the different ways of selling for a dimension.

## Types of relationships in ADAPT



Fully exclusive



Fully overlapping



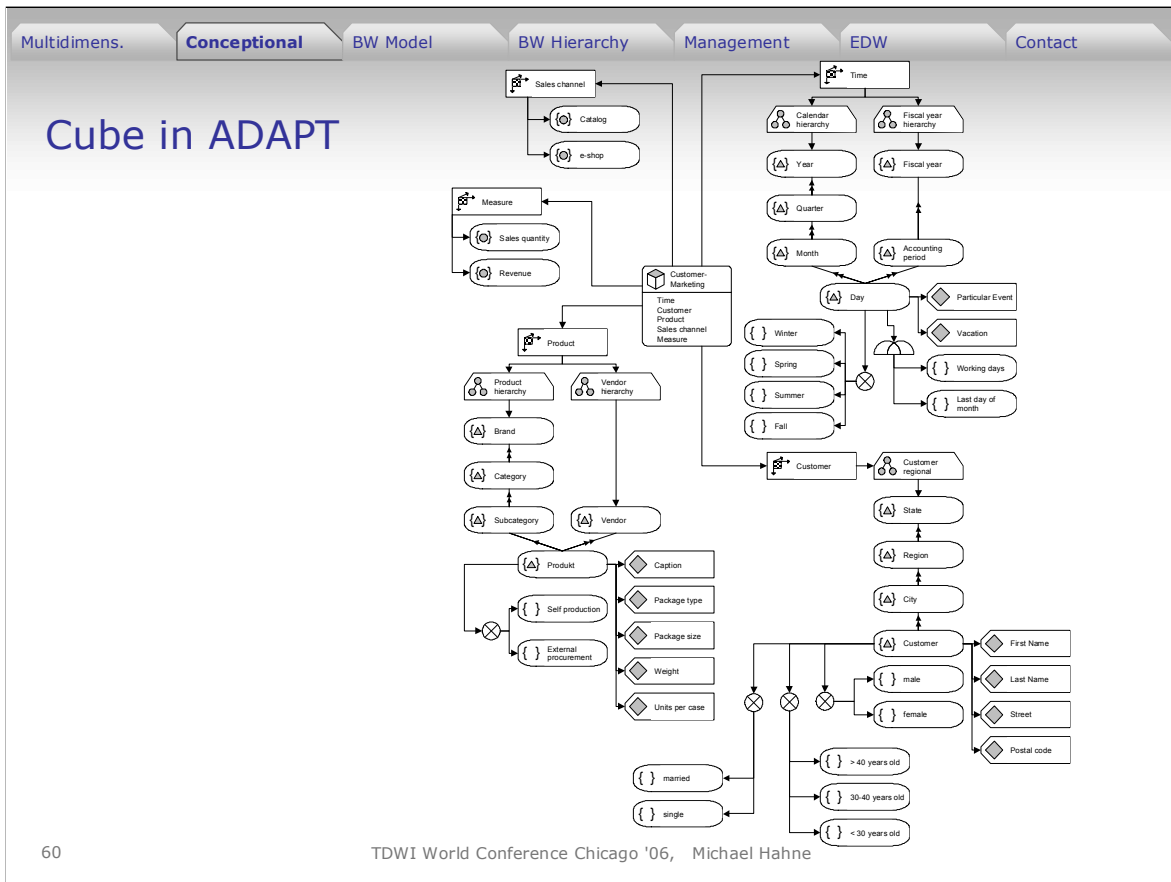
Partially exclusive



Partially overlapping

ADAPT's strength is in modeling dimension scopes representing logically connected subsets of a dimension. For example, subsets for self-production and external procurement might be used for products. These dimension scopes are related to each other; a product is either produced or purchased from an external source, for example. This type of relation is described as "fully exclusive". This slide shows an overview of the different types of relation between dimension scopes.

When describing the relations between subsets, the term "exclusive" means they are disjoint. However, subsets may partially overlap. When the combination of the subsets constitutes the entire set, it is described as "full"; otherwise, it is "partial".



60

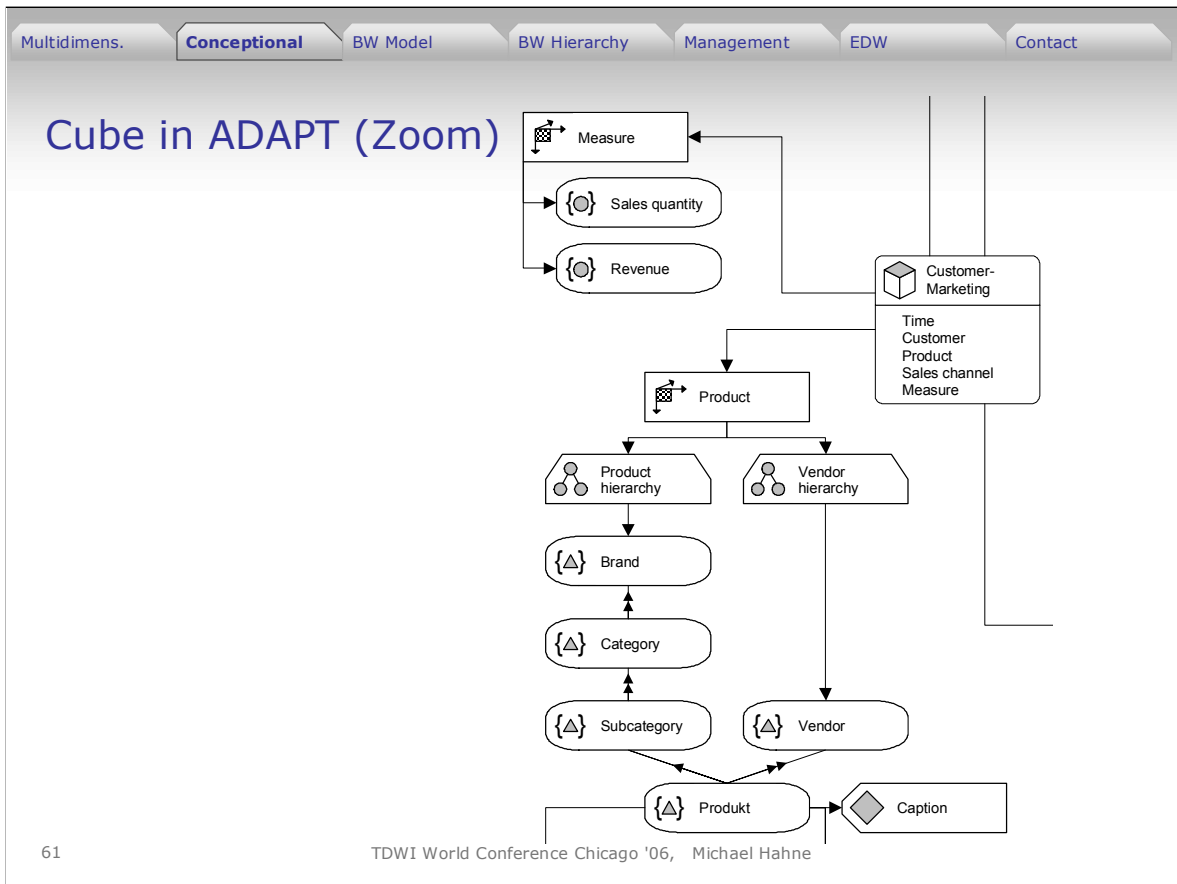
TDWI World Conference Chicago '06, Michael Hahne

Dimensions definitively cast the structure of the multidimensional data as a cube.

The Cube object is available for the representation of a cube in ADAPT. The caption and the connected dimensions are registered manually. Linking of the dimensions involved is represented by a connecting arrow object. In general, modeling with the Cube object and the attached dimension objects is sufficient to reduce the complexity of the representation. However, a complete representation is more meaningful in the case of smaller models, since all model components will be visible at once. The example shows the modeling of the customer marketing Cube, in which the dimensions are represented directly.

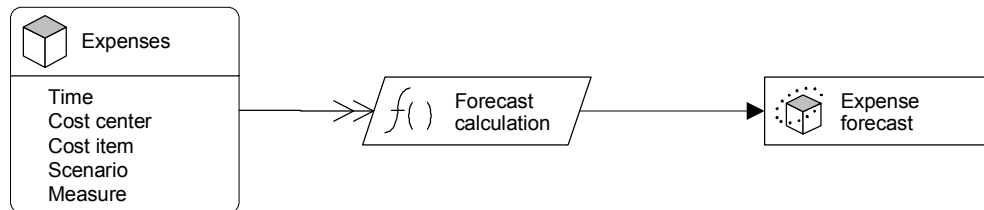
Besides the standard selections of years, quarters and months, analysis with respect to temporal aspects in the example offers additional information about fiscal years, since these differ from calendar years. Representation in ADAPT is done with two separate hierarchies in the time dimension. Special days, for example working days and the last days of a month, are as important as vacation days for analysis and are modelled as dimension scopes in the example. Holidays are modeled as attributes of the calendar date. Seasonal aspects are also to be considered, and can be modeled well using dimension scopes.

In evaluations, special attention is paid to the information accumulated about customers about direct customer business via catalog orders or online activities. Those information is grouped regionally, as illustrated in the example. Further, grouping can be performed according to family status, sex and age. In addition to the customer number, detailed information such as last name, first name, regional information, etc. also must be made available for each customer. Demographic attributes are defined as dimension scopes, while name and address information is modeled as attributes.



The dimensions definitively cast the structure of multidimensional data as a cube.

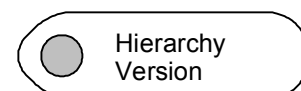
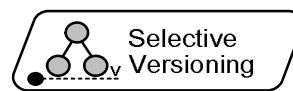
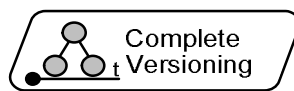
## Derived view of a cube (model)



## Extension coming up with T-ADAPT

• In addition to the objects in ADAPT there are the following objects in T-ADAPT (*M. Hahne*):

- Complete versioning
  - Special case of transactional versioning
- Selective Versioning
- New objects:



It should be possible to illustrate the report requirements specified earlier through modeling. Therefore, it must be possible to express the distinction of different versions of dimension structures. Four alternatives to the arrangement are available for implementation. The adjustment of the historical volume of data is omitted because all the requirements for this cannot be covered.

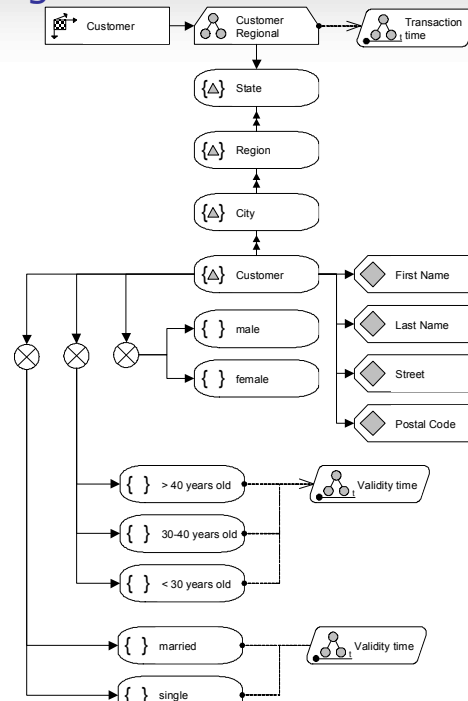
The remaining possibilities can be classified according to whether a hierarchy version is added at certain points in time, or all changes are continuously traced. These two options are described by selective timestamping and complete timestamping respectively.

Parallel hierarchies can be used directly for setting up individual versions using dimension structures in ADAPT. However, the modeling of the entire hierarchical structure using levels, elements and attributes will be performed several times over. Only the text addition with comments for complete timestamping is available in ADAPT.

The representation of object types for illustrating temporal properties is developed in this section in order to illustrate the temporal aspects of structural changes in dimensions adequately in ADAPT. The extension of ADAPT using these objects is called T-ADAPT. The newly available objects are shown here.

Time dependence can be modeled individually for each hierarchy using these objects. The conventional representation neglects any structural changes and sets a fixed hierarchical structure. Versioning can either be of a selective or of a continuous kind. A mixture within a hierarchy is impossible. Each structural change is considered when complete versioning is performed.

## Complete versioning on the basis of valid time and transaction time



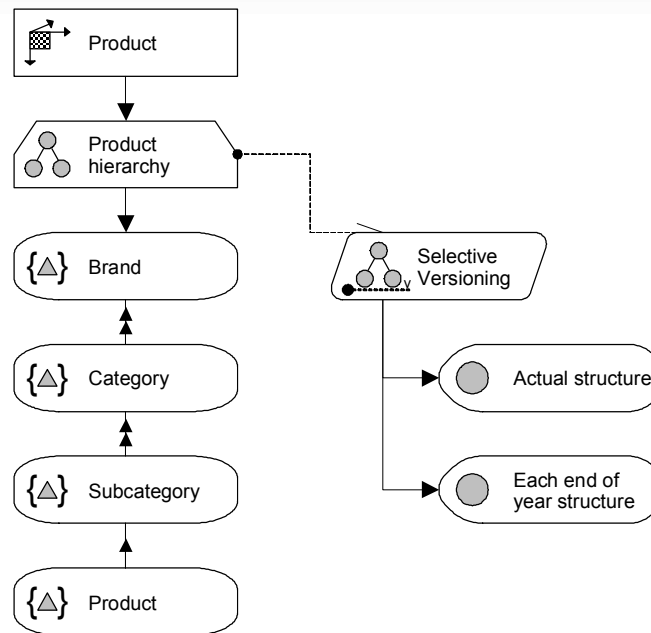
64

TDWI World Conference Chicago '06, Michael Hahne

This slide shows complete versioning for the two cases of dependence on the validity time and transaction time for the customer dimension. For the regional customer hierarchy, versioning based on transaction time is shown. The temporal characteristic is attached to the hierarchy object and is marked with the name of the central factor of influence. Apart from validity time as the determining factor for the generation of a new structural version, it is also possible to use the structure of reference dates as the basis for the hierarchy validity in each case. This view of continuous versioning is based on transaction time.

ADAPT makes it possible to group dimension elements, which can vary from time to time, to make them available by dimension scopes, separately from hierarchies. Demographic attributes of a customer are a typical example of this. Its form of timestamping is likewise specified. The dimension scopes of the demographic attributes, apart from sex, are timestamped in the product dimension according to validity time. An abbreviated way of writing is possible for timestamping of dimension scopes in demographic customer characteristics, for example. The objects in the example representing married and single People are bound to a timestamped object, as is evident from the illustration.

## Separate versions



65

TDWI World Conference Chicago '06, Michael Hahne

In the case of selective timestamping, the dotted line in the symbol indicates that only individual versions are considered. The various versions of the hierarchy are represented as independent objects attached to the temporal characteristic object. In the example, this is shown for the product dimension. Selective timestamping in the example is based on individual snapshots, because certain year-end structures are fixed in their own versions in addition to the valid up-to-date structure. The product dimension in the example is a suitable candidate for a selective timestamping in T-ADAPT on the basis of individual snapshots, because apart from the up-to-date valid product hierarchy, year-end versions are also taken into account.

It is possible to illustrate all forms of timestamping of hierarchies using the represented object types of modeling in T-ADAPT. Often, additional timestamping of attributes is required. In these cases it has to be decided whether it is better to model these attributes as their own dimensions in the modeling phase.

## Agenda: Conceptual Design

- Methods for conceptual design
- Conceptual modeling with ADAPT
- **Tool support by Microsoft Visio™**
- Tool support by SemTalk™

## Agenda: Conceptual Design

- Methods for conceptual design
- Conceptual modeling with ADAPT
- Tool support by Microsoft Visio™
- Tool support by SemTalk™

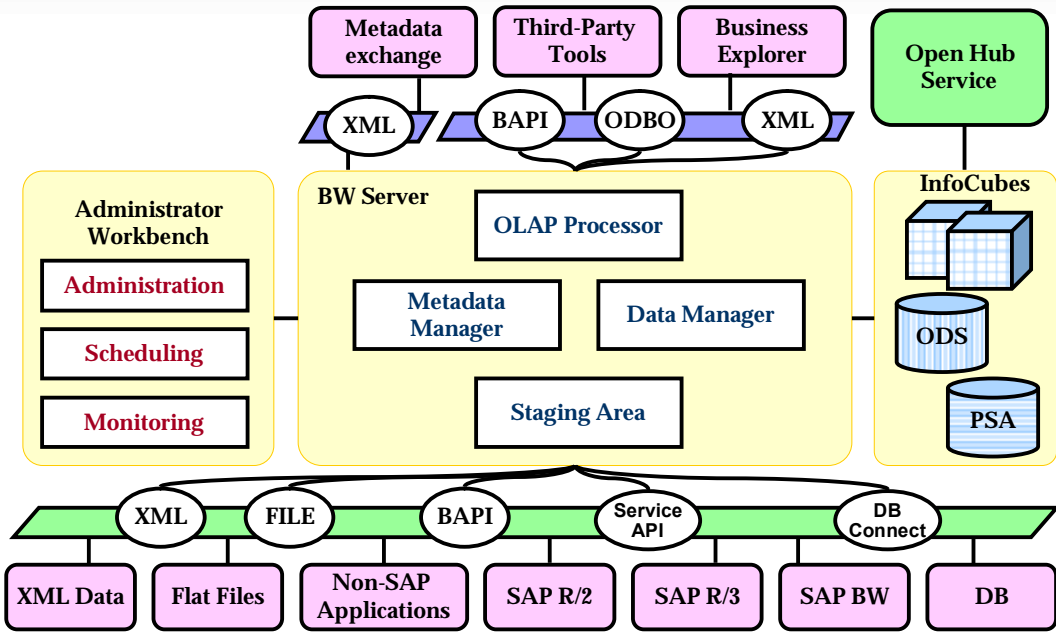
## Agenda: Data Model of SAP BW

- Architecture of the SAP Business Information Warehouse
- Extended Star Schema of the SAP AG

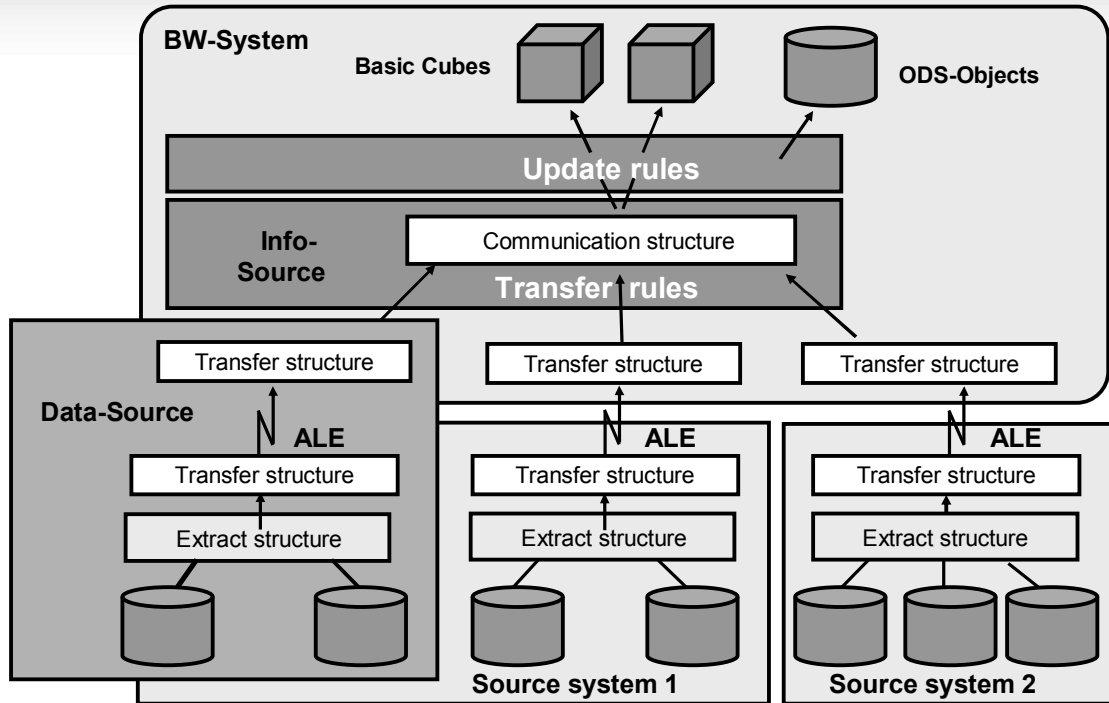
## Agenda: Data Model of SAP BW

- Architecture of the SAP Business Information Warehouse
- Extended Star Schema of the SAP AG

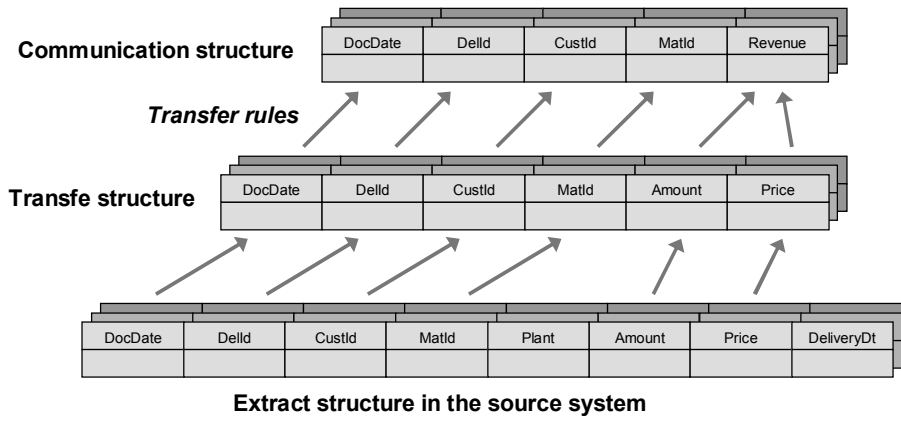
## Architecture of SAP® Business Information Warehouse



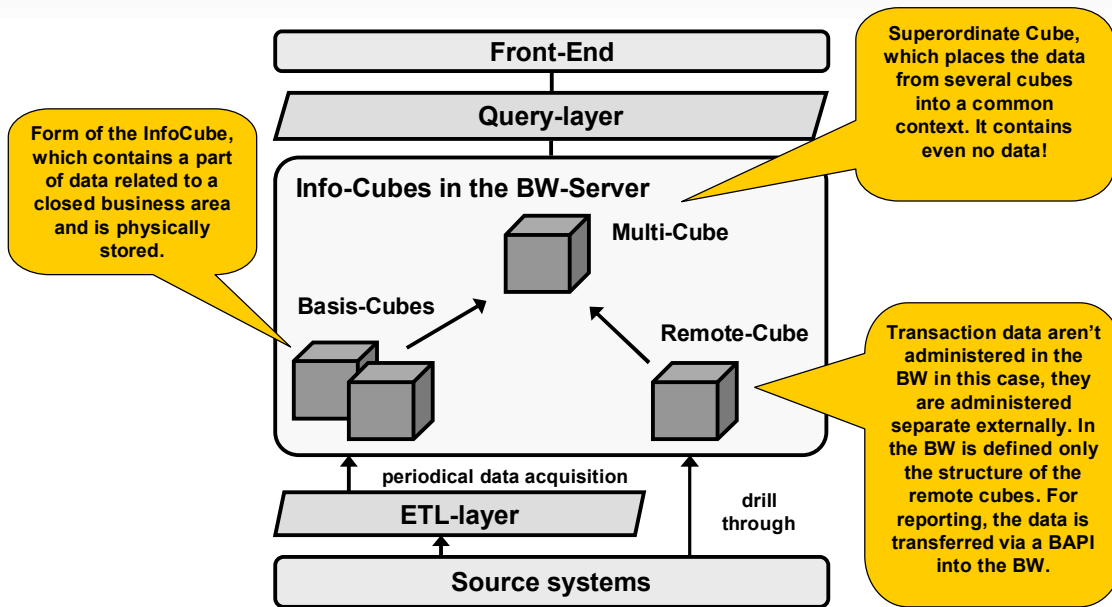
# Transformation steps



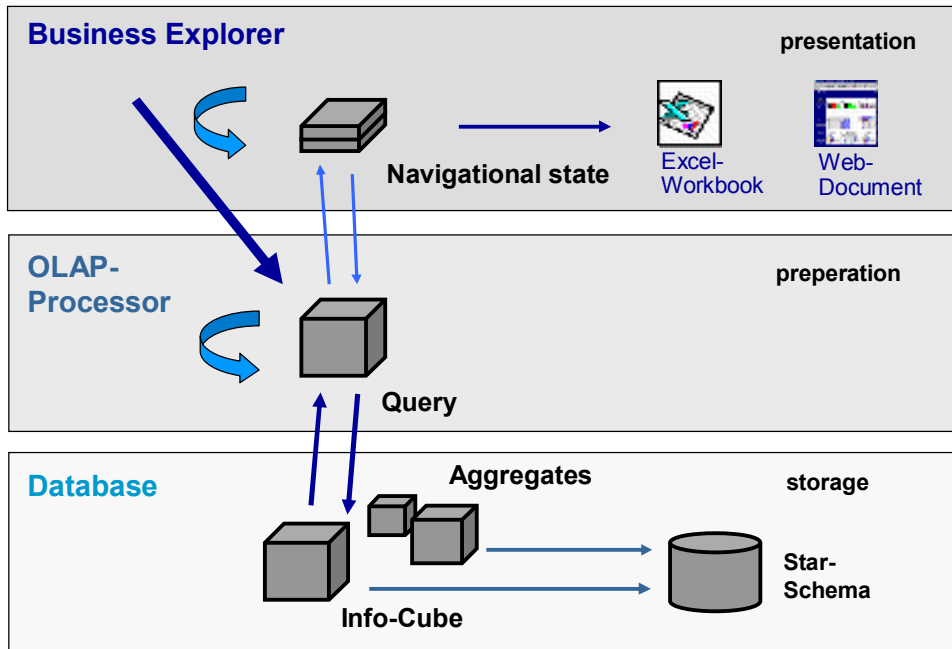
## Transformation within transfer rules



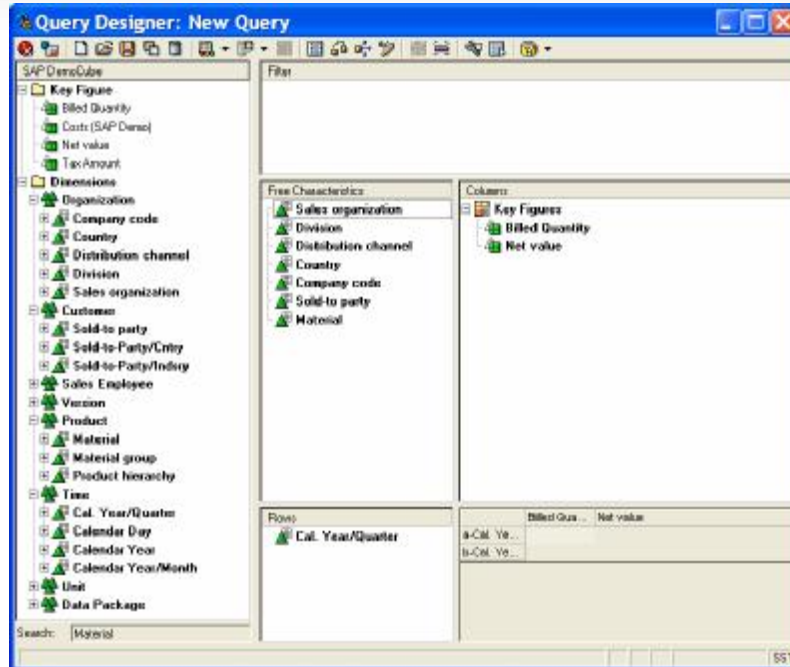
## Types of Info-Cubes



## Steps of query execution



## Query definition in QueryDesigner



75

TDWI World Conference Chicago '06, Michael Hahne

## Agenda: Data Model of SAP BW

- Architecture of the SAP Business Information Warehouse
- Extended Star Schema of the SAP AG

## Data targets in SAP® BW

Data targets are objects in which transactional data is stored for the purpose of reporting and analysis. The most important data targets are:

InfoCubes

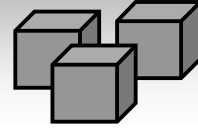


ODS Objects



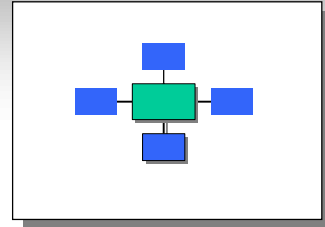
Additionally, there are further data targets in BW, which enable (for example) direct master data reporting (Info Sets, Info Objects)

## Definition of InfoCube



- The InfoCube is **central data storage** serving as the basis for reports and analyses in SAP® BW. It contains a restricted data volume, such as might relate for example to a specific well-defined business area or business unit.
- InfoCubes contain two types of data: **measures** and **characteristics**.
- The term "InfoCube" designates a table structure in which some relational tables are linked according to the so-called Star Schema (multidimensional data storage)
- Star Schema: Dimension tables are grouped in a star formation around a central fact table

## Star Schema



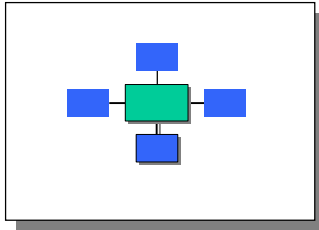
- The Star Schema is the most typical way of representing multidimensional data structures in relational data bases.
- In the Star Schema, facts are stored in a separate **fact table**, while characteristics are grouped in **Dimension tables**. The dimension tables are joined to the fact table via foreign key and primary key relationships (DIM ID).
- In this way, all data records from the fact table are marked **uniquely** by a combination of these foreign key values from the dimension tables.

Several approaches have been developed for storing multidimensional data structures in relational databases. These are usually known as Star Schemata. Facts are stored in a central fact table; this is at the center of the star schema and is surrounded by dimension tables. The components of the primary key in the fact table reference the dimension tables. The dimension tables define analytical criteria and consolidation paths. Generally, these dimension tables are denormalized.

Requirements for consistent response times lead to variants focusing especially on this aspect. A typical approach is to precalculate aggregates and store them in aggregate tables. SAP BW's data model is essentially based upon the Star Schema.

## Pros and cons of the general Star Schema

- J Good performance for data analysis
- J Very flexible when adding characteristics and measures



- K Problems come along with:
  - many-many relationships and
  - unbalanced hierarchiesbecause of the uniqueness of the primary keys in the dimension tables

Therefore, SAP AG decided to extend the Star Schema.

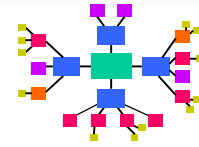
Master data is stored separately and independently from InfoCubes, in the so-called **Extended Star Schema**.

Data Storage in BW is based on a relational database technology. Multidimensional data is also stored in a slightly modified variant of the Star Schema, with some improvements on points that are weakly supported in the classical approach: primarily the treatment of m:n-relationships in dimensions, support for unbalanced structures and aspects of structural changes in dimensions. Furthermore, in SAP BW it is possible to save data at different levels in cubes, because in general only input on the lowest level is recognized. This is the case for actual data on a daily level and budget data on a monthly level. for example.

## Extended Star Schema

The Extended Star Schema gives access to:

- Master data tables and their corresponding attributes
- Text tables with extensive multilingual captions
- External hierarchy tables for structured data access

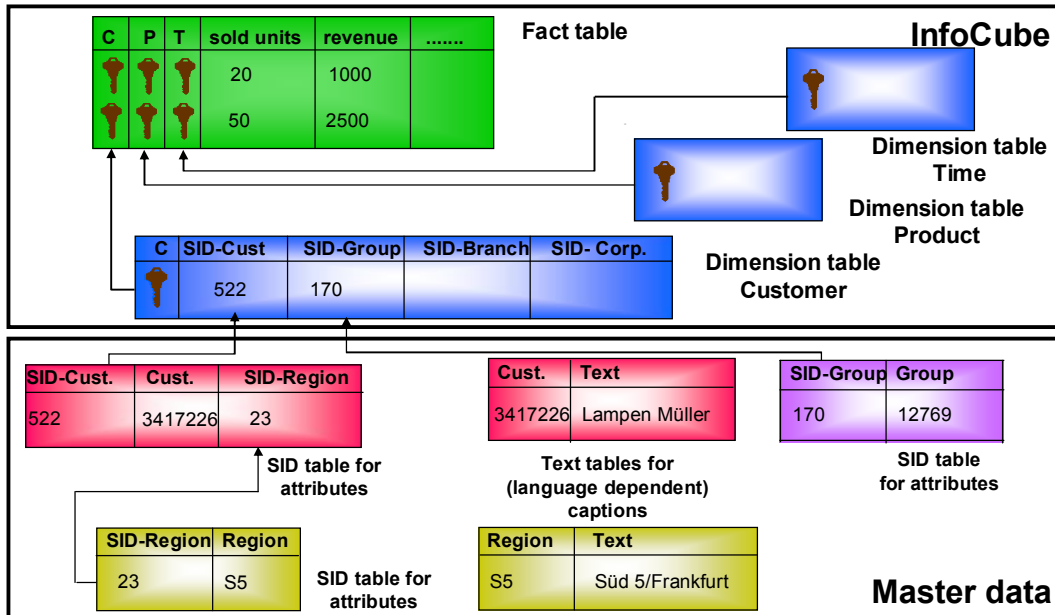


Master data and hierarchy tables are joined to the fact table via **SID Tables** (pointer tables) and dimension tables



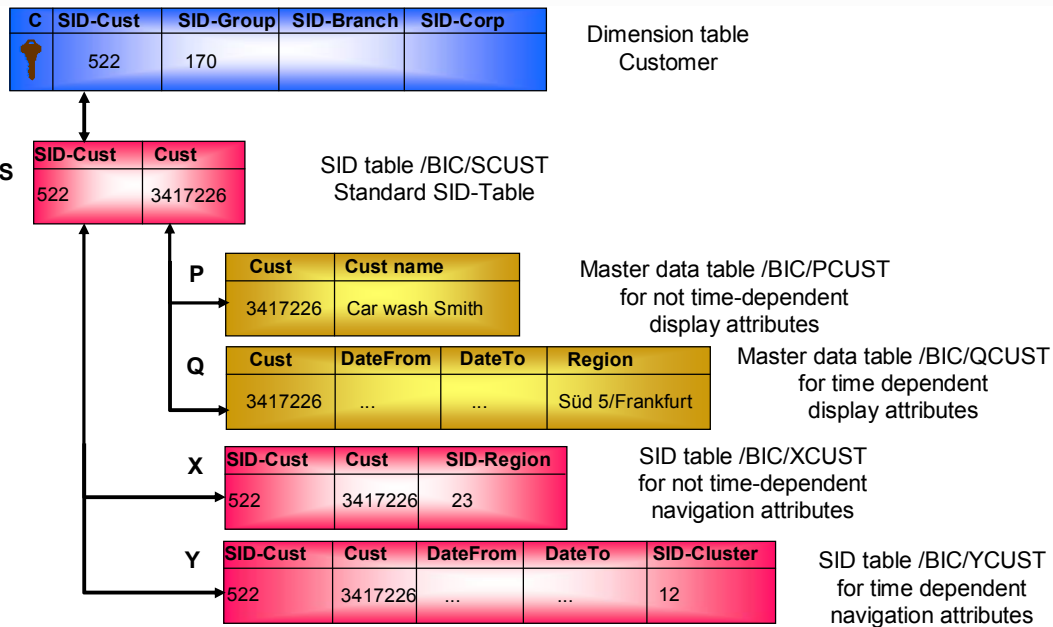
The basic object for storing multidimensional data in the Business Information Warehouse is the InfoCube. One fundamental principle in BW is the sharing of structural informations in dimensions that are spreaded to all InfoCubes. Master data is therefore valid for all multidimensional structures in a system. The fact tables in the enhanced Star Schema such as BW uses are also in the center of the model. Facts are called key figures. The fact table is surrounded by the dimension tables with appending master data tables.

## Connecting master data to cubes via SID Tables



The usage of the terms dimension and dimension table in SAP BW (and the terminology of multidimensional modeling in general) can be slightly confusing. In BW's Enhanced Star Schema, the master data is joined to the facts with dimension tables. Only the reference is stored in these tables, while the real dimension member values are stored in separate master data tables. The basic object used to build multidimensional models in BW is the InfoObject. It is identified uniquely by its technical name, and is comparable with a field in a table defined by name and data type. InfoObjects are defined globally in the BW system and represent the objects to be analyzed, i.e. the key figures and also the criteria for analytics given as characteristics. All data objects in BW have InfoObjects as their basic components.

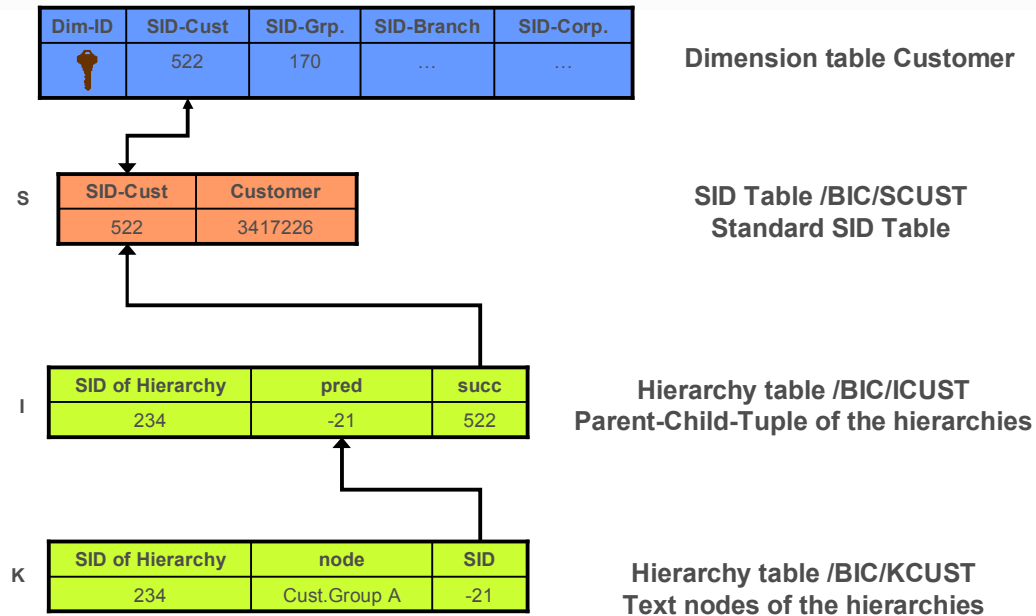
## Different master data tables



Very often, the objects used to build dimension members are given as numerical keys (for example, a material number). An explanation is provided by a textual description of the elements, for example the material corresponding to the material number. In Business Information Warehouse, an essential feature is support for textual descriptions in different languages. These are stored in separate tables, with the texts kept together with the language code. For example, ISO country codes match to different country names in different languages. The Enhanced Star Schema considers attributes of dimension elements as master data attributes of characteristics. They are stored in separate tables. For example, a customer object has attributes such as the address and contact person. For the material, the standard price might be an attribute like the material group, for example. In BW, attributes of characteristics are divided into those that can be queried without the base characteristic and those with only a descriptive function. The former type of attributes are called navigational attributes, and the latter are called display attributes.

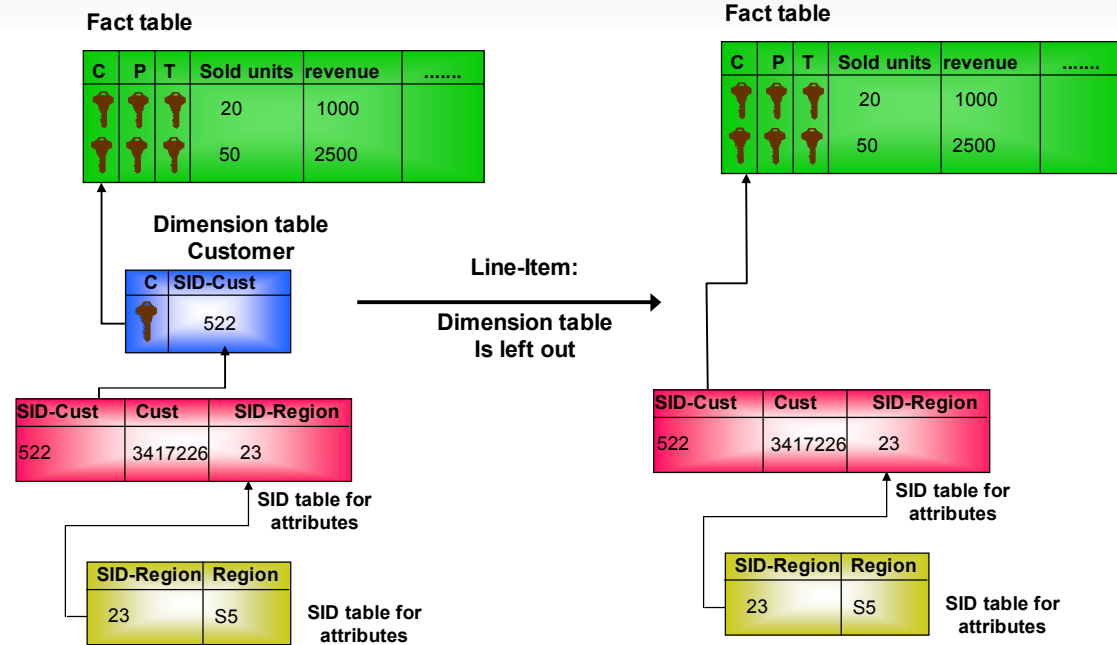
It is possible to define any master data in the Business Information Warehouse with time stamps, and therefore attributes. Texts and hierarchies can be given with validity information. These time dependent attributes are joined to InfoCubes by different SID tables.

## Hierarchy tables



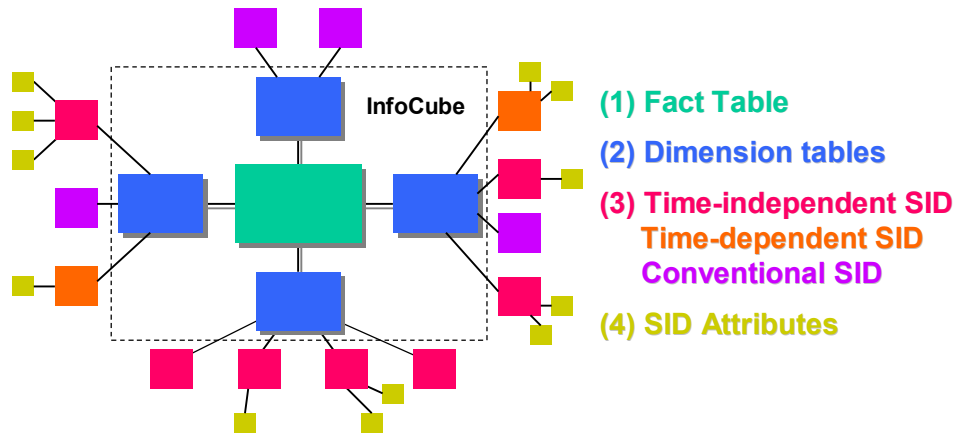
An example of a hierarchy of a characteristic is the grouping of countries into regions and continents (this will be used later on). These explicitly defined hierarchies are stored in the Business Information Warehouse, also in the separate master data tables. One has to be careful with BW's usage of the term "hierarchy". The standard meaning, such as applies to like the structure of a dimension, corresponds to 1:n relationships between dimension members. Hence there are three different forms of hierarchical structures in BW, defined by external hierarchies, characteristics in the cube and navigational attributes of characteristics. The master data is not directly joined with the fact table of the InfoCube, but linked by further SID tables that translate artificial dimension keys (surrogate id's) into keys for characteristics. This enables m:n relationships between characteristics in a dimension. This concept is shown here.

## Line-Item dimensions



If a dimension consists only of one characteristic, it is possible to simplify the physical model by omitting the dimension table, since this only represents a one-one mapping from the dimension identifier to the characteristic sid.

## Complexity of the extended Star Schema, at a glance



Querying of the model always starts with the dimensions. Depending on the actual query, different SID tables will be used to restrict the qualifying tuples of the dimension tables that are used to join the fact table. The core metadata information of an InfoObject is its data type. The classical numeric data types are supplemented by types of quantities and amounts that are given with a unit of measure. As an example, for amounts of money, the corresponding currency is the unit. Units are stored in BW automatically and conversions between different units (currency conversion, for example) are possible. Conversion rules are not static; different types can be used such as ordinary currency conversion with average exchange rates, fixed courses and so on.

## Agenda: Modeling hierarchies in SAP BW

- Variants for modeling hierarchical dimension structures
- Temporal aspects and time stamping
- Modeling guidelines

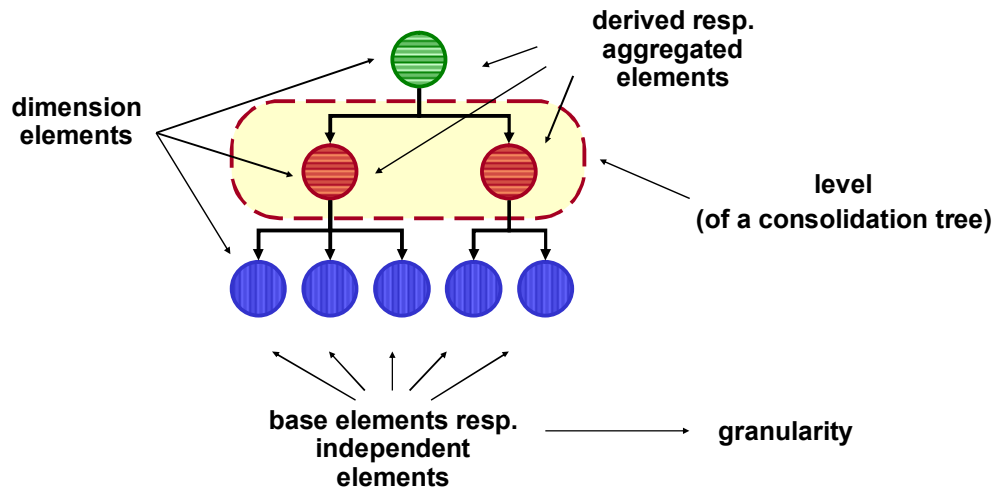
## Agenda: Modeling hierarchies in SAP BW

- Variants for modeling hierarchical dimension structures
- Temporal aspects and time stamping
- Modeling guidelines

On-Line Analytical Processing (OLAP) as a fundamental principle for building analytical systems to support users in their business administration needs that is based upon the multidimensional conceptual model with cubes and dimensions as main components enabling multiple variants of navigation and projections to different consolidation levels.

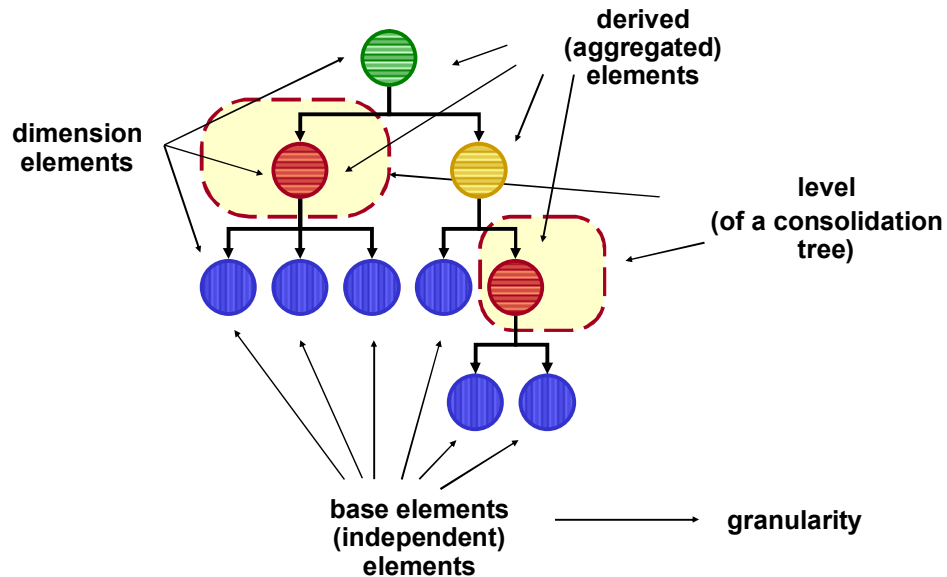
The aspect of modeling hierarchical dimension structures is an essential point in designing analytical information systems and hence in BW many different hierarchical structures are supported. Hierarchies can be implemented in three different ways in the Business Information Warehouse. Besides the possibility of hierarchies based upon characteristics in dimensions one can implement hierarchical structures in the master data either with navigational attributes or with external hierarchies. The decision which choice fits best has to be done in the phase of modeling and the different possibilities can be used simultaneously to recognize various reporting demands. There are many aspects that have to be considered in this decision process. One very important aspect is the performance. Generally speaking hierarchies in fact are faster to query than those in master data. Poorest performance is connected with time-dependant external hierarchies. Hierarchies based upon characteristics are the only way to model structures valid only for one cube because master data is valid for all cubes in a system. A further criterion is the flexibility in reorganising dimension structures which only with master data can easily be done. The flexibility of various dimension types that are possible with external hierarchies are often an argument for this way of modeling hierarchical structures. The treatment of all reporting demands is possible with the technique of time-dependant structures in BW but one has to take care of the aspect that for querying such structures one has to define a key query date that sets the point in time to be recognized and in each query only one key date is valid for all time-dependant objects of the query. The more flexibility of modeling time-dependant structures is connected with more complexity and poorer performance.

## Dimensions and balanced hierarchies



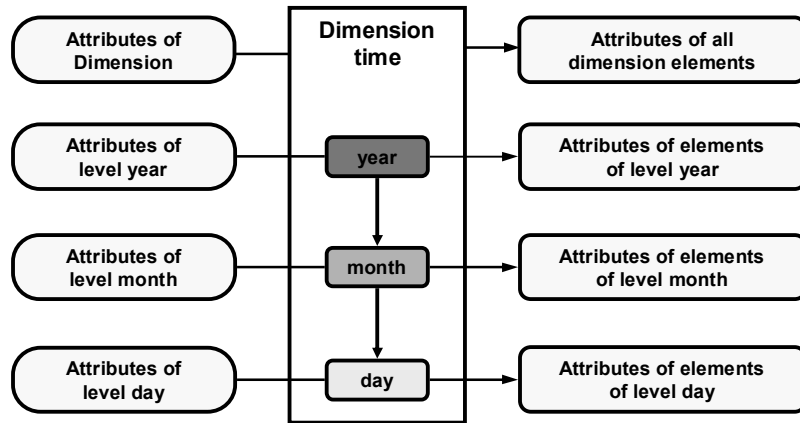
The fundamental concept of a data warehouse is multidimensionality: data storage is based on multidimensional data structures which presents data in an n-dimensional cube where cells represent the values. The edges of the cube are called dimensions. These qualify the values and are the basis for deep analysis for business management purposes. The elements in a dimension can be grouped, for example, for a product dimension a grouping of elements to form product groups is possible. These groupings are built up from 1:n-relationships. A set of such relationships form a navigational or consolidation path in a dimension, and the base elements on the lowest level form the level of deepest granularity, therefore defining the most detailed level of analysis. The other elements are called derived or consolidated elements. The different parts of a dimension are summarized on this slide.

## Unbalanced hierarchies

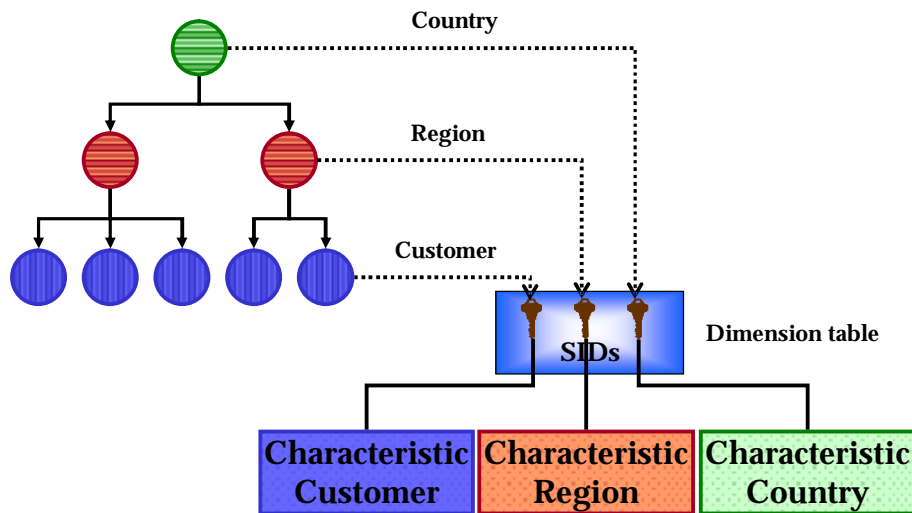


Often, there are parallel hierarchies in which the consolidation paths split into different forks which may join again at a higher level. If the condition of 1:n-relationships is dropped, the resulting structures are based upon m:n-relationships. The assignment of calendar weeks to months is an example of this type of dimension structure. Furthermore, the condition of balanced structures can be dropped, resulting in unbalanced structures. In the case of unbalanced dimensions, the grouping of elements into classes is a special case because the groups do not correspond to the levels that are defined by the length of a minimal path from the node to a root element. These groups have to correspond more to the analytical needs than to the counting of vertices.

## Attributes



## Hierarchies within a dimension via characteristics

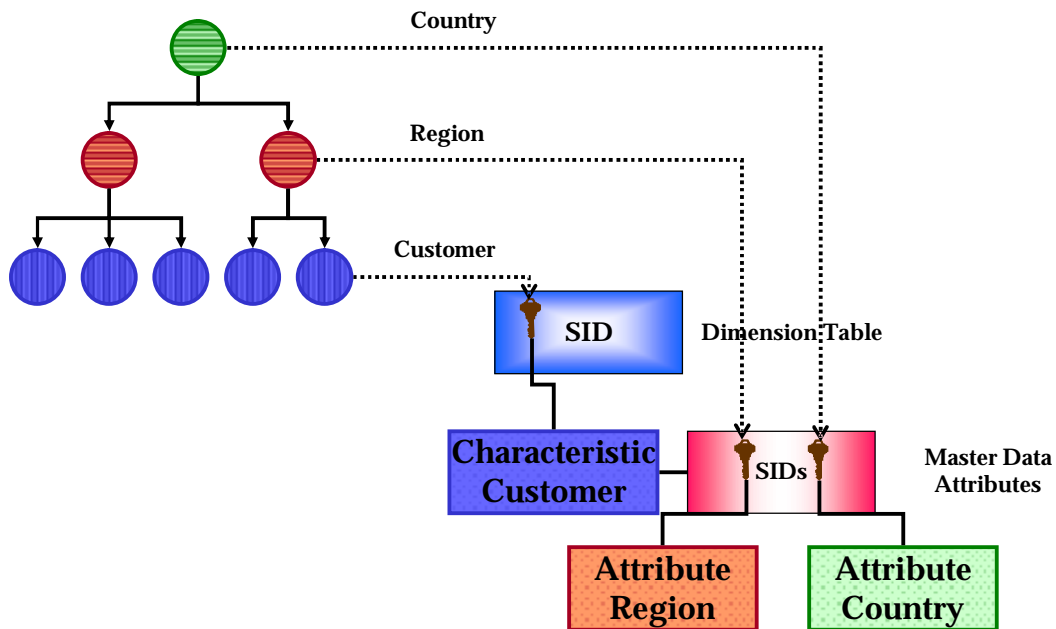


The first possibility for modeling dimensional structures in BW is given by mapping each named level of consolidation of the dimension, as defined in the conceptual model, to a characteristic in a dimension in BW (remember the different meaning of the term dimension). Therefore the assigning of data to consolidation elements is given by the fact data and fixed at load time. This relationship between the different levels is not stored in master data valid for all cubes, but in the InfoCube. This principle is shown here. Unbalanced structures are impossible with this kind of dimension structure modeling because each level maps to a characteristic. The Business Information Warehouse does not know about the hierarchical structure of characteristics in a dimension, such as customer, region and country, for example. Therefore there is no predefined path of navigation. Hierarchical relationships between characteristics cannot be restructured without general reorganization of data, and so reporting is based on the historical truth.

## Hierarchies within a dimension via characteristics

- Each level is represented by an InfoObject → number of levels should be fixed
- Generally faster than attributes and external hierarchies
- Include the higher hierarchical levels to aggregates
- No predefined drill-down paths

## Navigational attributes as basis of hierarchical structures



95

TDWI World Conference Chicago '06, Michael Hahne

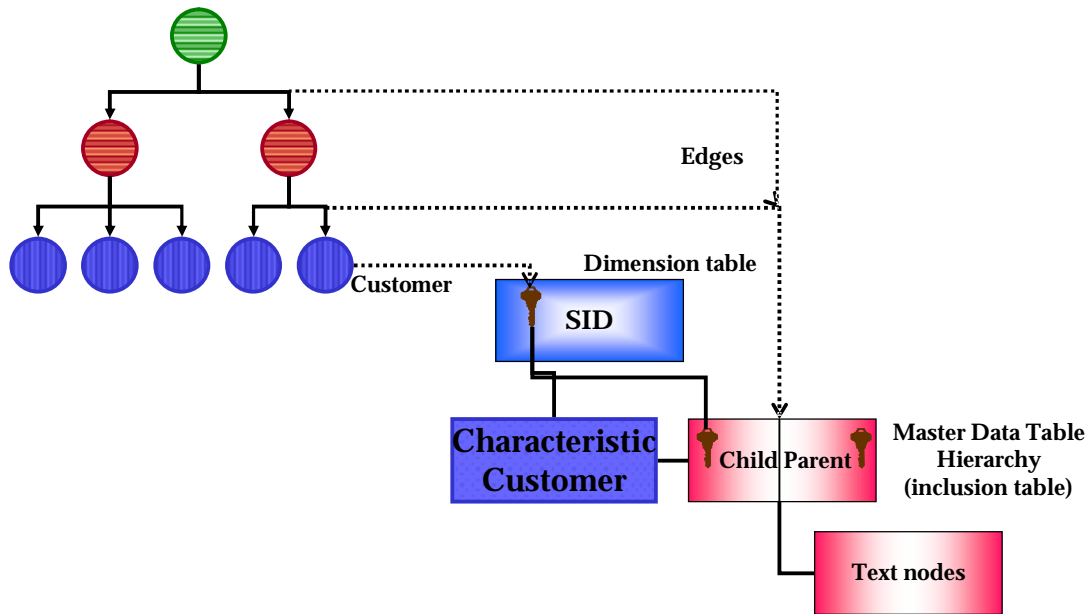
When characteristics are the basis of a hierarchy, reorganization is needed to recognize different structures in one dimension. This is a reason for modeling dimension structures in the master data. Then every level of consolidation corresponds with an attribute of the base characteristic. This is shown here. In the most extreme case, where the dimension consists only of one characteristic, all hierarchical information is stored in the master data. In cases with only one characteristic, the dimension can be defined as a line item dimension in which the dimension table is omitted and the fact table is directly joined to the master data table.

The storage of dimension structures as attributes in master data implies that they are valid for all InfoCubes and that changes are very simple to implement, since only master data table updates are needed. In this type of modeled hierarchy in BW there are also no predefined paths of consolidation, and so levels can be left out in drilling down or rolling up. In general, this possibility of modeling enables reporting only with the current hierarchical structure. If different structures need to be recognized because of reporting demands, it is possible in BW to implement the attributes as time-dependant. This means all attribute values are stored together with the validity information in two additional fields, "valid from" and "valid to". For time-dependant attributes, one cannot define aggregates to realize performance improvements.

## Navigational attributes as the basis of hierarchical structures

- Each level is represented by an InfoObject → number of levels should be fixed
- Include the higher hierarchical levels to aggregates
- No predefined drill-down paths
- Poor performance without aggregates
- Increased flexibility for reorganization

## External hierarchies in BW



97

TDWI World Conference Chicago '06, Michael Hahne

The most common way to model dimension structures in BW is through definition of external hierarchies that are based on a parent-child-representation and hence enable unbalanced structures. They are also stored in master data and therefore are valid for each InfoCube. For every InfoObject, many different external hierarchies can be defined in different versions. External hierarchies are based on nodes, with predecessors and successors if required. Elements without successors are called "leaf elements", while those without a predecessor are "root elements". Nodes in an external hierarchy can be textual nodes or nodes based upon characteristics.

External hierarchies are stored in master data spread over different tables. The parent-child tuples are stored in the inclusion table. Nodes for characteristics are stored with the SID, and textual nodes get an artificial identifier of negative numbers. The different tables and their connections are summarized here.

One can assign multiple parent elements to a node, dropping the restriction to 1:n-relationships between consolidation levels. Consistency of calculation is ensured automatically by BW.

Using different versions and multiple hierarchies modeled as external hierarchies, different reporting demands can be implemented. To match the validity with a special time, these can be defined as time-dependant, recognizing either the whole hierarchy or each parent-child tuple individually. This enables reporting according to all possible dimension structures given in the complete time period.

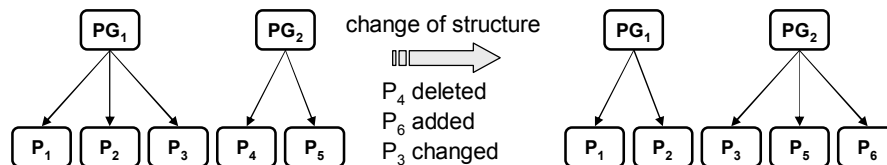
## External hierarchies in BW

- Reasonable alternative when there are frequent changes to dimension structure
- Enables unbalanced structures
- Several hierarchies per Info Object are possible
- Poor performance, similar to navigational attributes
- Problems with big hierarchies
- In the case of time dependency, only time stamping the entire structure will enable aggregates

## Agenda: Modeling hierarchies in SAP BW

- Variants for modeling hierarchical dimension structures
- **Temporal aspects and time stamping**
- Modeling guidelines

## Time dependency: changes to consolidation trees



Time aspects are very important in multidimensional models. This is indicated by the special role of the time dimension. Time aspects of facts are inherently stored, but the tracing of changes in dimensions is not supported in general. This aspect is essential for analytics. Reporting demands define what kind of support for time aspects in dimensions is needed. In practice, data often must be analyzed according to the actual dimension structure, as well as according to some kind of historical structures. Another reporting demand requires consideration of the structure for each fact tuple stored in the cube that was valid at loading time. This is termed the "historical truth". Also, one might report only those structural components that are valid for all reported periods, that is, "not compare apples and oranges".

### References:

Hahne, Michael (2003): Time aspects in SAP® Business Information Warehouse, in: Jardim-Gonçalves; Cha, J.; Steiger-Garção (Ed.): Concurrent Engineering - The Vision for the Future Generation in Research and Applications, Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications, 26-30 July 2003, Madeira, Portugal, A.A. Balkema Publishers, Lisse et al, 2003, p. 69-74.

## Example of "Slowly Changing Dimensions"

### Product dimension in 2005-04

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

### Product dimension in 2005-05

Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

### Fact table

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

## Reporting requirements - Scenarios

### Reporting scenario: current structure

Productgroup	Rev. 2005-04	Rev. 2005-05
PG X	100	100
PG Y	300	400

### Reporting scenario: old structure

Productgroup	Rev. 2005-04	Rev. 2005-05
PG X	200	200
PG Y	200	200

### Reporting scenario: "historical truth"

Productgroup	Rev. 2005-04	Rev. 2005-05
PG X	200	100
PG Y	200	400

### Reporting scenario: comparable results

Productgroup	Rev. 2005-04	Rev. 2005-05
PG X	100	100
PG Y	200	200


## Scenario I : Report with current structure

**Product dimension in 2005-05**

Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

**Fact table**

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100



Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	300	400

## Query path "current structure" with navigational attributes

S-table of Productgroup

Productgroup SID	Productgroup
4711	PG X
4712	PG Y

X-table of Product

Product SID	Product	Productgroup SID
1001	P A	4711
1002	P B	4712
1003	P C	4712
1004	P D	4712
1005	P E	4712

Dimension Table Product

Product SID	Product DIM ID
1001	25
1002	26
1003	27
1004	28
1005	29

Fact table

Product DIM ID	„Period“	Revenue
25	„2005-04“	100
26	„2005-04“	100
27	„2005-04“	100
28	„2005-04“	100
25	„2005-05“	100
26	„2005-05“	100
27	„2005-05“	100
28	„2005-05“	100
29	„2005-05“	100

Productgroup	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	300	400

## Query path "current structure" with external hierarchy

K-table of Product

SID	nodename
-2	PG X
-3	PG Y

I-table of Product

pred	succ
-2	1001
-3	1002
-3	1003
-3	1004
-3	1005

Dimension Table Product

Product SID	Product DIM ID
1001	25
1002	26
1003	27
1004	28
1005	29

Fact table

Product DIM ID	„Period“	Revenue
25	„2005-04“	100
26	„2005-04“	100
27	„2005-04“	100
28	„2005-04“	100
25	„2005-05“	100
26	„2005-05“	100
27	„2005-05“	100
28	„2005-05“	100
29	„2005-05“	100

Product group	Revenue 2005-04	Revenue 2005-05
PG X	100	100
PG Y	300	400

## Scenario II : Report with old structure

**Fact table**

**Product dimension in 2005-04**

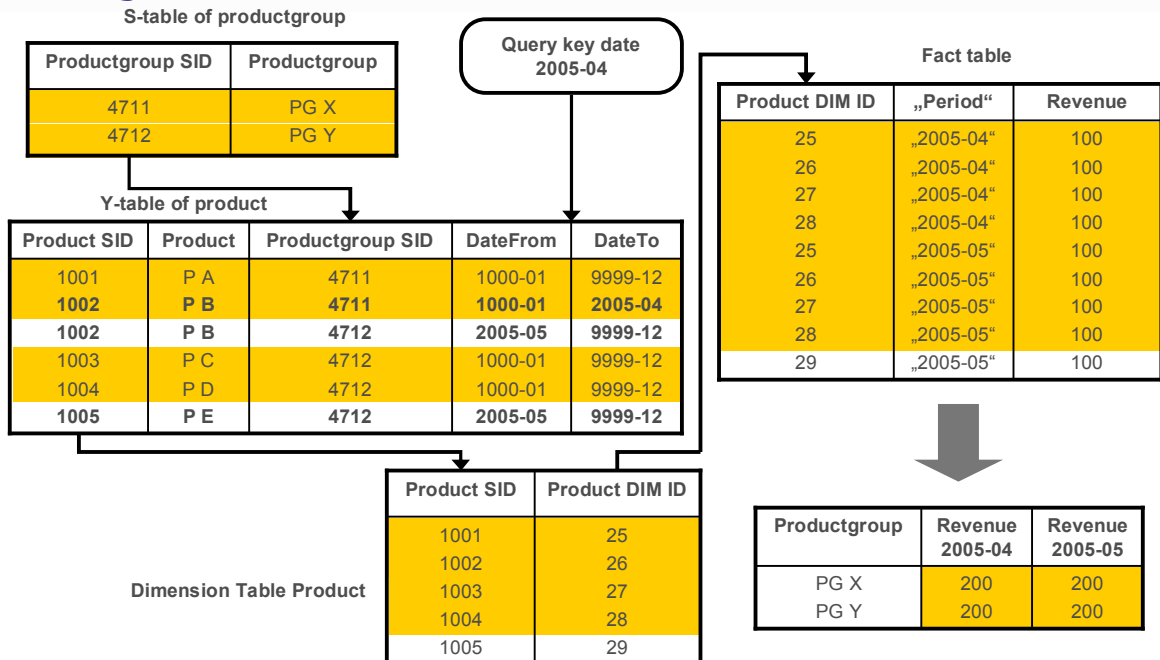
Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

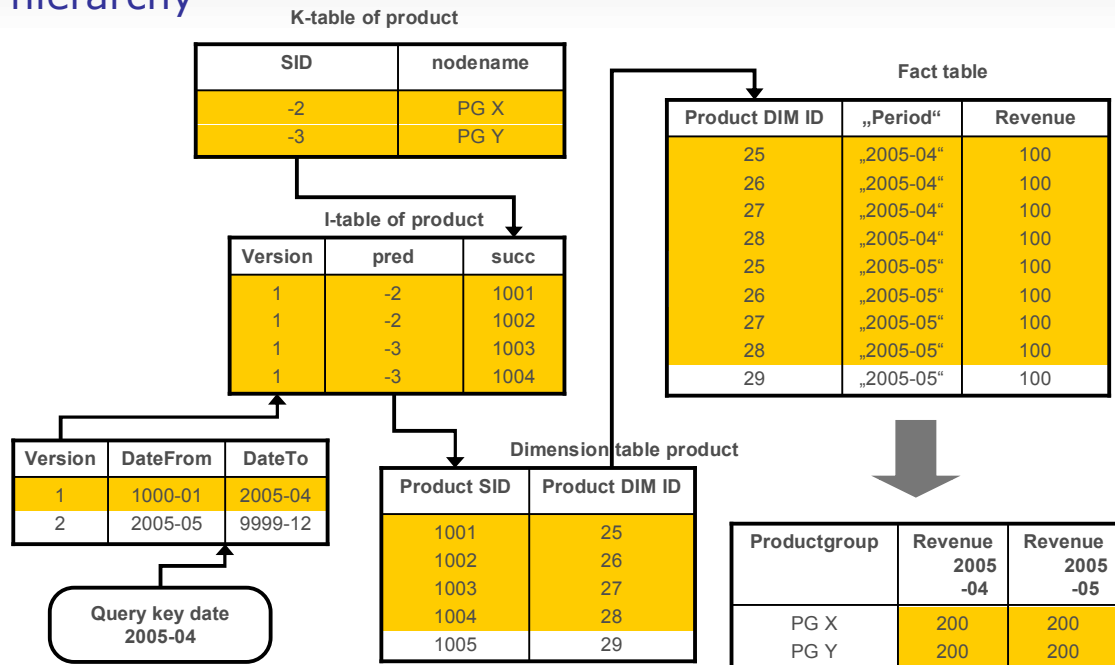
  

Productgroup	Rev. 2005-04	Rev. 2005-05
PG X	200	200
PG Y	200	200

## Query path "old structure" with time-dependent navigational attributes



## Query path "old structure" with time-dependent hierarchy



108

TDWI World Conference Chicago '06, Michael Hahne

## Scenario III : Report with "historical truth "

### Product dimension in 2005-04

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

### Product dimension in 2005-05

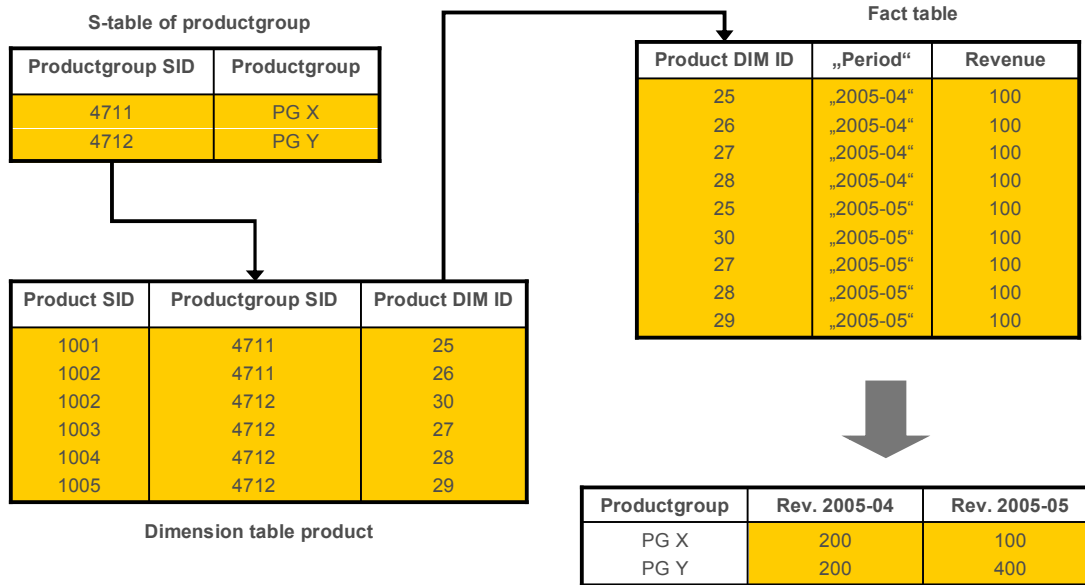
Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

### Fact table

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

Productgroup	Rev. 2005-04	Rev, 2005-05
PG X	200	100
PG Y	200	400

## Query path "historical truth" with characteristics



## Scenario IV : Report with Comparable Results

### Product dimension in 2005-04

Product	Productgroup
P A	PG X
P B	PG X
P C	PG Y
P D	PG Y

### Product dimension in 2005-05

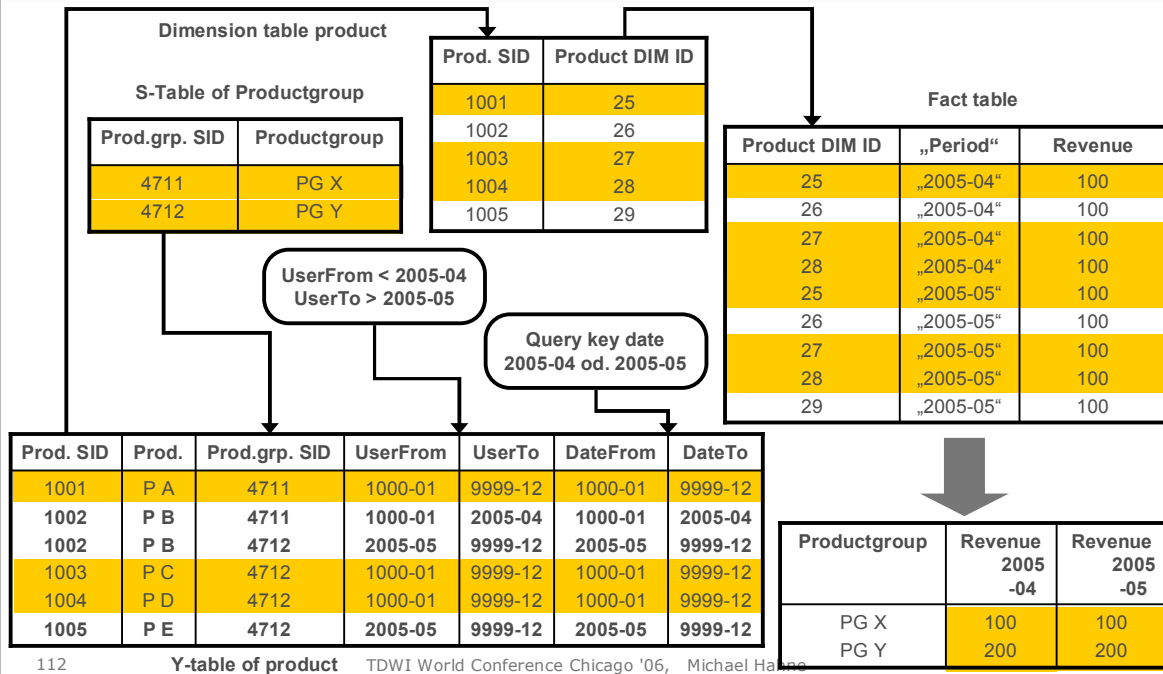
Product	Productgroup
P A	PG X
P B	PG Y (changed)
P C	PG Y
P D	PG Y
P E	PG Y (new)

### Fact table

Product	Period	Revenue
P A	2005-04	100
P B	2005-04	100
P C	2005-04	100
P D	2005-04	100
P A	2005-05	100
P B	2005-05	100
P C	2005-05	100
P D	2005-05	100
P E	2005-05	100

Productgroup	Rev. 2005-04	Rev. 2005-05
PG X	100	100
PG Y	200	200

## Query path "comparable results" with time-dependent navigational attributes



## Agenda: Modeling hierarchies in SAP BW

- Variants for modeling hierarchical dimension structures
- Temporal aspects and time stamping
- **Modeling guidelines**

## Modeling Guidelines

- Modeling of dimensions
- Design of InfoProvider

## Guidelines for conceptual modeling of dimensions

- Number of dimensions should be between four and ten (optimally between six and eight)
- Number of hierarchy levels (seven at most)
- Number of elements per consolidation element (a maximum of fifteen to twenty elements is advisable)
- Determination of dimensions:
  - 1:1 relationship unsuitable (-> attributes)
  - 1:N relationship determines dimension hierarchy
  - M:N relationship rather two different dimensions

## Guidelines for logical modeling of dimensions

- Model characteristics with high cardinality as line-item dimensions
- Attributes that change frequently should be modeled with their own dimensions (use line item where possible!)
- Group characteristics with very low cardinality (e.g. scenario) into one dimension, in order to reduce the number of dimensions and to restrict the number of dimensions to 16 at most
- Distribute characteristics of a hierarchy that have high cardinality into separate dimensions (parent characteristics in their own dimension)

## Criteria to help decision-making in logical modeling of dimension structures in BW

- Versioning
- Scope
- Performance
- Navigational paths
- Unbalanced dimension structures
- Leaves with multiple parent elements
- Structural changes and reorganization

## Hierarchy Guidelines: Versioning

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>• Transactional view isn't possible (no "as posted")</li><li>• Different types of views are possible (hierarchy versions and time-dependent hierarchies)</li></ul>	<ul style="list-style-type: none"><li>• Only the transactional view ("as posted") is possible</li></ul>	<ul style="list-style-type: none"><li>• Transactional view isn't possible (no "as posted")</li><li>• Time-dependent attributes enable different views</li></ul>

118

TDWI World Conference Chicago '06, Michael Hahne

## Hierarchy Guidelines: Scope

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>Hierarchy is part of master data and is valid for each InfoCube in the system (where the underlying InfoObject is used)</li></ul>	<ul style="list-style-type: none"><li>Only valid in the InfoCube</li></ul>	<ul style="list-style-type: none"><li>Hierarchy is part of master data and is valid for each InfoCube in the system (where the underlying InfoObject is used)</li></ul>

## Hierarchy Guidelines: Performance

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>Aggregates should be used for good query performance</li></ul>	<ul style="list-style-type: none"><li>Good performance (even without aggregates)</li></ul>	<ul style="list-style-type: none"><li>Aggregates should be used for good query performance</li></ul>

## Hierarchy Guidelines: Navigational paths

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>• Drill-down path is predefined by the structure of the consolidation tree</li></ul>	<ul style="list-style-type: none"><li>• Levels can be skipped because there isn't a predefined drill-down path (all characteristics in a dimension are equal)</li></ul>	<ul style="list-style-type: none"><li>• Levels can be skipped because there isn't a predefined drill-down path (all navigational attributes of a characteristic are equal)</li></ul>

## Hierarchy Guidelines: Unbalanced dimension structure

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>• Unbalanced hierarchies are possible</li></ul>	<ul style="list-style-type: none"><li>• Each characteristic corresponds to a certain level of the hierarchy, therefore only balanced structures are possible</li></ul>	<ul style="list-style-type: none"><li>• Each characteristic corresponds to a certain level of the hierarchy, therefore only balanced structures are possible</li></ul>

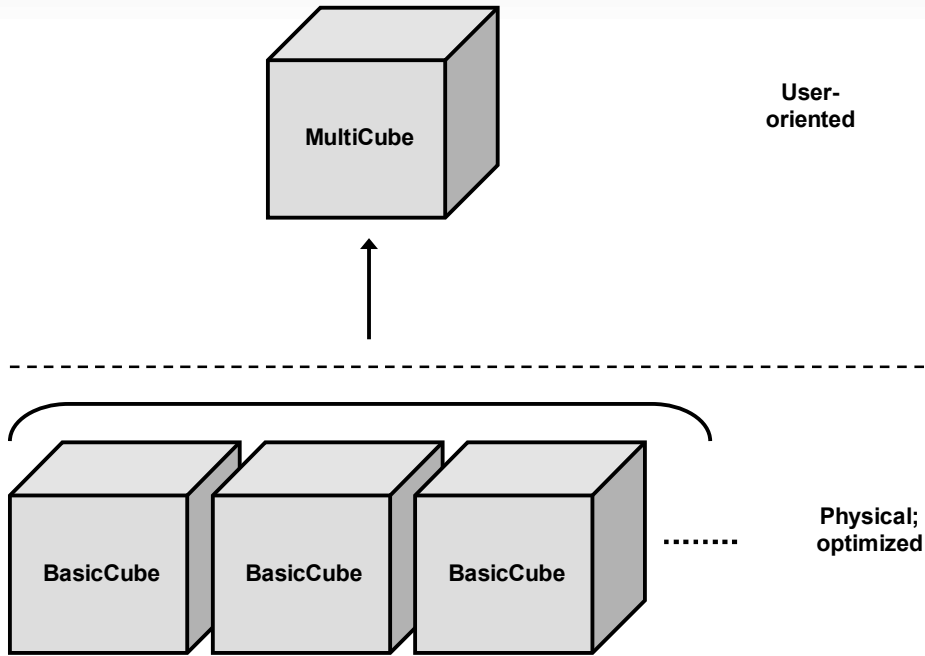
## Hierarchy Guidelines: Leaves with multiple parent elements

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>• Many-many relationships between the levels of the hierarchy are possible and consolidated correctly</li></ul>	<ul style="list-style-type: none"><li>• Many-many relationships between hierarchy levels are only possible as they are defined by the transactions</li><li>• ("as posted" view)</li></ul>	<ul style="list-style-type: none"><li>• Many-many relationships between levels are impossible</li></ul>

## Hierarchy Guidelines: Structural changes and reorganization

External hierarchy	Hierarchy within a Dimension (characteristics)	Hierarchy defined by navigational attributes
<ul style="list-style-type: none"><li>• Quick changes and reorganization possible</li></ul>	<ul style="list-style-type: none"><li>• Reloading cube(s) is required for reorganization</li></ul>	<ul style="list-style-type: none"><li>• Reorganization is possible (additional attributes and/or changes of master data)</li></ul>

## Two-layer concept of Cube Modeling



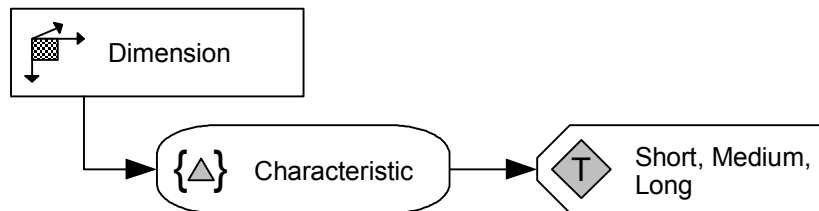
## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- Common Warehouse Metamodel (CWM)
- Implementation of the CWM in BW

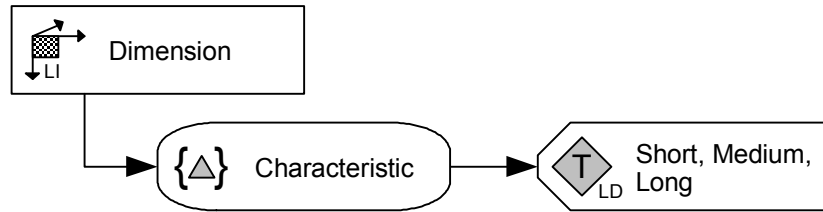
## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- Common Warehouse Metamodel (CWM)
- Implementation of the CWM in BW

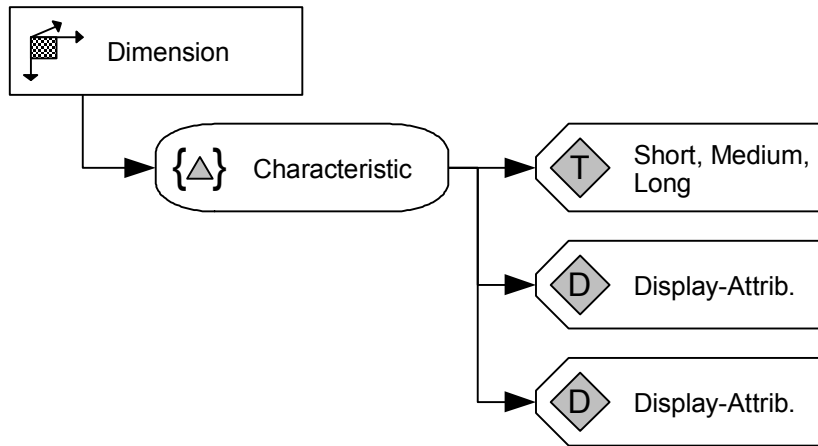
## Dimension with one characteristic



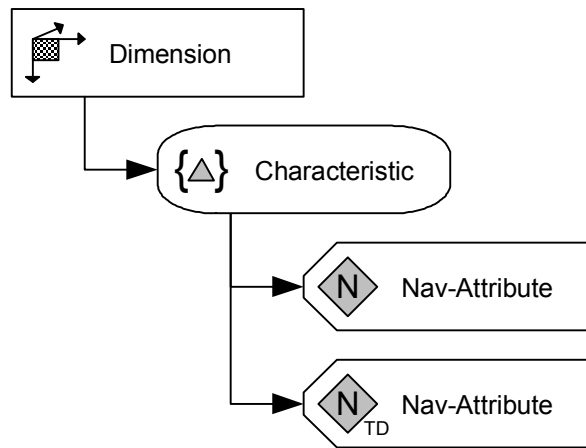
# Line-Item-Dimension



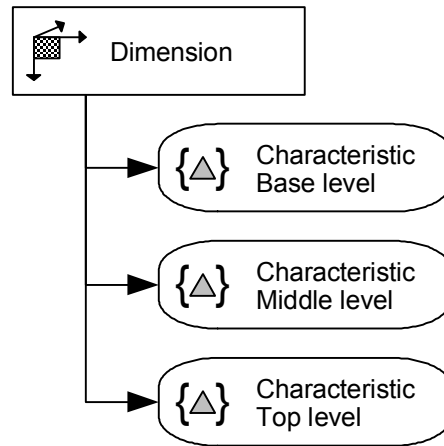
## Display attributes of a characteristic



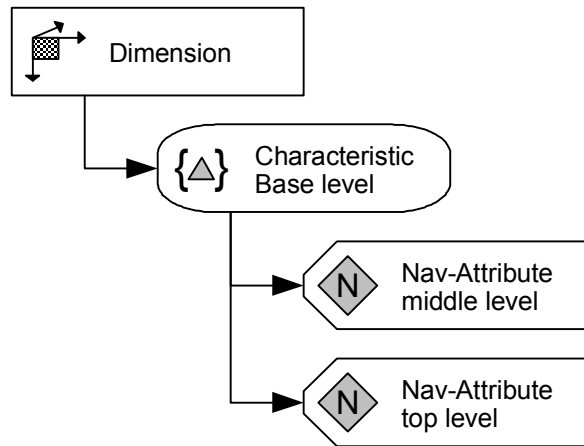
## Navigational attributes of a characteristic



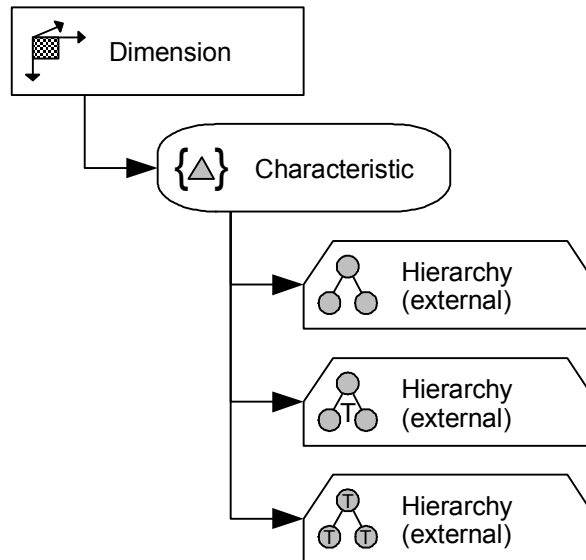
## Hierarchy within a dimension with characteristics



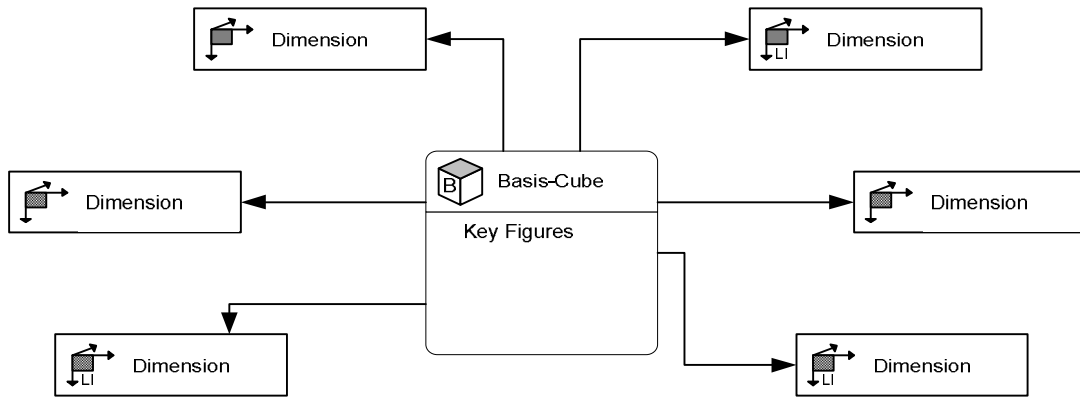
## Hierarchical structure with navigational attributes



## External hierarchy



# Modeling of BasicCubes



## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- Common Warehouse Metamodel (CWM)
- Implementation of the CWM in BW

The process of building and structuring information systems gives rise to a large amount of information that describes these systems, for example, extensive data models with data objects and relations. The storage and administration of metadata is a central component of operational as well as analysis-oriented information systems based on the Data Warehouse concept and OLAP technology.

Metadata has a far-reaching significance in many respects, and is particularly important for the efficient running of analysis-oriented information systems. It is of fundamental importance for the entire ETL process (extraction, transformation, load), as well as for successful use of the system.

## Definition of metadata

- Concise:  
**"Data about data"**  
not sufficiently precise
- More precise:  
**"Data that describes the meaning and structure of business data, as well as how it is created, accessed and used"**  
[Devlin, 1997]
- Commonly used:  
**"Each kind of information that is required for planning, building and running information systems"**  
[Bauer/Günzel, 2001]

A common concise description of metadata says that it is "data about data". This catchy definition is, however, not sufficiently exact. The term metadata is more precisely defined by Devlin as "Data that describes the meaning and structure of business data, as well as how it is created, accessed and used". According to another common definition, "metadata" means each type of information required for the planning, the building and the use of an information system.

## Models

- Constructivist approach to modeling:

**Models are the result of a process of modeling that generates a "mini-world" by abstracting from the real world**

- Classification:

**Models for description, explanation and decision making**

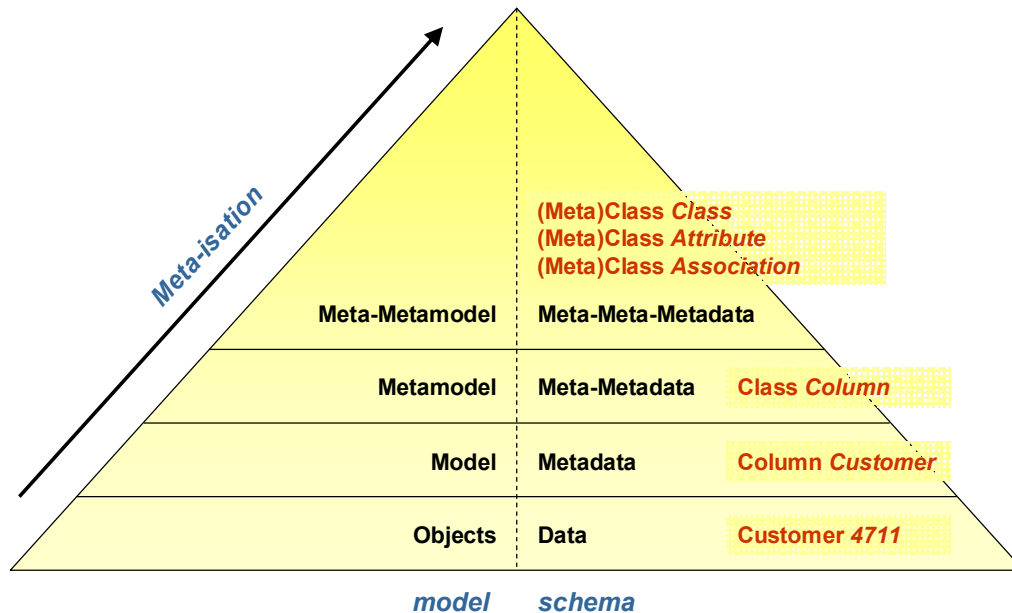
- Sometimes confused usage of the word "model":

sometimes: language of modeling (**model**)

other times: concrete instance (**schema**)

In accordance with the construction-oriented modeling approach, models are the result of a modeling procedure in which the real world of discourse – the so called "mini world" – is illustrated in an abstract way. In the literature, many modeling-related terms and classification possibilities are differentiated. A possible distinction of models is based on their function, according to which the description, explanation and decision models have to be differentiated. Here we are looking at descriptive models primarily.

## Abstraction layers of modeling



139

TDWI World Conference Chicago '06, Michael Hahne

To enable classification of the concept of metadata, the representation of different abstraction levels of modeling is illustrated in this figure. A complementary differentiation is made between the models as modeling language and their concrete instances. The representation is based upon the English terminology, as a consequence of which the modeling language is characterized by the word "model" while the models created are called "schemes".

At the lowest level of abstraction, the data and objects are to be found as an image of the "mini world". As a result of a further abstraction, models arise that describe the structure of the objects and/or data. Development of these models results in metadata, which is therefore on a higher abstraction level. If models and the creation of models become the subject itself of the modeling, we speak of "metamodels". A further stage of abstraction results in the transition to models, focused on the description of Metamodels. This process – designated as metaization – can in principle be taken further, however, the four-level metamodel architecture as described is sufficient for the purposes of this article. One of the concept's goals in this context is to illustrate an object range in a descriptive model, for the creation of which a language is used. Accordingly, a language is also needed for the formulation of a metamodel, which, in this sense, is a model of a model itself. In this context it is a question of finding, in the case of the superordinate model, a linguistic description tool, which describes the language in which the subordinate model is formulated.

## Meta-isation

- Metamodels:  
When models and modeling itself become the subject of modeling
- Further abstraction:  
Proceed to models that focus on the description of metamodels
- This process is called meta-isation and can be continued indefinitely
- Objective of creating a model:  
Representing an area of interest in a descriptive model, by means of language.

## Relevance of metadata in Data Warehousing

- For end users: permits use of information in the Data Warehouse in a relatively autonomous way
  - Definition of used business terms (glossary)
  - Mapping of business terms to data objects in the Data Warehouse
  - Description of reports, contact persons, access rights
- For development and administration: technically oriented metadata
  - Data sources, rules for improving data quality
  - Transformation rules, steps of data consolidation
  - Mapping of data sources to data warehouse models

A metadata database system is an essential component in the Data Warehouse reference architecture, where information about the system components is stored and administered. Apart from pure administrative purposes, this serves the users in particular as a fast and safe way of finding of necessary information in the Data Warehouse. The user is to be put into a position in which he can use, to a large extent independently, the information stored in the analysis-oriented information system.

The three most essential fields of application for metadata in the Data Warehouse are the administration of the system, the description of the meaning of the stored content and the navigation component.

Co-workers concerned with the development and administration of Data Warehouses need technically-oriented metadata analogous to the metadata used in operational systems. This includes data concerning the data sources as well as rules for the improvement of data quality. This also includes rules for transformation and consolidation steps as well as mapping information between the data sources and the Data Warehouse models, not excluding the metadata of the data models of the Data Warehouse.

Users of a Data Warehouse need a broad spectrum of metadata to be able to understand and evaluate the data it contains, which enables an interpretation of the Data Warehouse contents. In particular, the definition of the business terms used, as well as the linking of specialized vocabulary with data objects of the Data Warehouse, must be mentioned in this context. Other important metadata is the description of possible reports, responsible partners and conditions for access authorizations to certain data areas.

An end user-suitable navigation component which is based on metadata is necessary to enable an extensive autonomous access to the Data Warehouse. This component enables, for example, free formulation of queries as well as functionality for navigation, electronic report distribution and access to data in operational systems.

## Relevance of metadata in Data Warehousing

- „best-of-breed“ vs. „end-to-end“
  - Heterogenous complex system landscape → uniform Data Warehouse solution
  - Improved Interoperability by means of meta data exchange
- Meta data management is crucial to improving efficiency and effectivity of a data warehouse solution

Two basic, counter rotating principles play a crucial role when constructing an enterprise-wide system landscape for information supply for analytic processes. Beside the "best of breed" approach, where separate particularly suitable software is used for each field, "end to end" solutions try to cover all tasks on basis of only one BI Suite. The use of dedicated tools often results in a heterogeneous complex system landscape, which has to be integrated to a uniform Data Warehouse solution. There are often found different tools for Data Warehouse storage, the implementation of ETL processes as well as OLAP -, Data Mining and reporting tools. The exchange of meta data between the individual tools is a much promising approach especially for the improvement of interoperability in heterogeneous landscapes.

The meta data management is a central function for the increase of effectiveness and efficiency of a Data Warehouse solution. The necessary coordination processes require an organizational framework for meta data management besides tools for meta data administration in order to guarantee consistency, actuality and quality of meta data.

References:

Hufford, D.: Metadata Repositories: The Key to Unlocking Information in Data Ware-houses. In (Barquin, R.; Edelstein, H. Hrsg.): Planning and Designing the Data Ware-house, Upper Saddle River, 1997.

Gardner, S. R.: Data Warehouses and Metadata: The Importance of Metadata Manage-ment. In (Siu, B. et al. Hrsg.): Data Mining, Data Warehousing & Client/Server Databases, Proceedings of the 8th International Database Workshop, Industrial Volume, Singapore, 1997.

Marco, D.: Building and Managing the Metadata Repository. A Full Lifecycle Guide, New York et al., 2000.

## Classification of meta data

- DSS vs. operativ:
  - operational meta data:
    - Source systems
    - data structures
    - field definitions
    - source data structures
    - transformation process
    - target system
  - DSS meta data:
    - Data storage
    - Data structures in the data warehouse
    - Objects of the Business Model
    - Navigation possibilities

The classification of meta data in a Data Warehouse can take place according to different criteria. A very rough differentiation distinguishes between operational and dss meta data. Operational meta data contains information about source systems, sources of original data, data structures, field names as well as information about the transformation process and the addressed target data sources. The dss meta data connects the physical Data Warehouse data with the models of the business processes and end user tools. It also covers data concerning the data storage and structures in the Data Warehouse, the objects of the business model as well as the navigation possibilities.

## Classification of meta data

- Main functions of a Data Warehouse:
  - Meta-Data: Meta data management, basis of the other functions
  - Source: Identification of data sources
  - Load: Data transfer and transformation
  - Storage: Storage type, data distribution, archiving
  - Query: Reporting and analysis, OLAP, simulation and Data Mining
- Threedimensional classification:
  - Processes: Design, Populate, Administer, Analyze
  - Users: Technical User oder Business User
  - Data: Operational, Data Warehouse, Data Mart

The five main functions of a Data Warehouse lead to another approach, where the meta data administration (Meta-Data) forms the foundation for the other four main functions. They contain the identification of data sources (Source), the data load and transformation process (Load), memory function (Storage) including aspects of storage type and data distribution and archiving. Also contained is the querying and evaluation function (Query), which is mainly focused on OLAP tools, simulation methods and Data Mining techniques.

This classification is similar to the use in the Data Warehouse process with the phases Design, Populate, Administer and Analyze. A three-dimensional classification results concerning users, processes and data by the differentiation of Technical user and Business user on the one hand as well as the distinction in the data view (operational, Data Warehouse and Data Mart) on the other hand.

References:

Poe, V.: Building a Data Warehouse for Decision Support, Upper Saddle River, 1996.

Do, H. H., Rahm, E.: On Metadata Interoperability in Data Warehouses, Technical Report Nr. 01(2000), Institut für Informatik, Universität Leipzig, Leipzig, 2000.

## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- **Common Warehouse Metamodel (CWM)**
- Implementation of the CWM in BW

## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- **Common Warehouse Metamodel (CWM)**
- Implementation of the CWM in BW

## Common Warehouse Metamodel (CWM)

- Model for Data Warehouse metadata
- Developed since 1998: active participation of IBM, NCR, Hyperion Solutions and Oracle among others
- Certified by the Object Management Group (OMG) as a Data Warehouse standard in 2000 for the first time
- Embedded in OMG's Model Driven Architecture (MDA)
- Based upon more than 200 Classes und 150 Associations (UML class diagram)
- Metadata exchange via XMI (XML Metadata Interchange) and via CORBA interfaces
- Basic principles: completeness, comprehensibility, independence, compatibility, interoperability, extensibility, availability

The Common Warehouse Metamodel (CWM) adopted by the Object management Group was established as the industry-wide standard for the modeling of metadata in Data Warehouse systems, as well as for their exchange. It offers a solid foundation for the modeling of extensive metadata in the context of Data Warehousing, based upon known standards and detached from proprietary metadata structures. Interpreting the metadata model as an ontology provides support for the distributed modeling of metadata necessary in real world environments, and enables a better presentation of data warehouse contents to the end user.

The Common Warehouse Metamodel (CWM) defines a model for metadata in Data Warehousing, which was developed through the active participation of prominent software producers such as IBM, NCR, Hyperion Solutions and Oracle, amongst others, since 1998. It was first passed as a standard by the Object Management Group (OMG) in the year 2000. It is an increasingly widespread reference model for Data Warehouse metadata.



## CWM metamodel layering

<b>Management</b>	<b>Warehouse Process</b>			<b>Warehouse Operation</b>		
<b>Analysis</b>	<b>Transformation</b>		<b>OLAP</b>	<b>Data Mining</b>	<b>Information Visualization</b>	<b>Business Nomenclature</b>
<b>Resource</b>	<b>Object Model</b>	<b>Relational</b>	<b>Record</b>	<b>Multidimensional</b>		<b>XML</b>
<b>Foundation</b>	<b>Business Information</b>	<b>Data Types</b>	<b>Expression</b>	<b>Keys and Indexes</b>	<b>Type Mapping</b>	<b>Software Deployment</b>
<b>Object Model</b>	<b>Core</b>		<b>Behavioral</b>	<b>Relationships</b>		<b>Instance</b>

The clarity of the many classes within the CWM was increased by introducing thematically focused packages. These packages are arranged in five layers, each of which cover the model components of one phase in data warehousing. The definition of the fundamental constructs takes place in the lowest layer Object Model, and all further layers are based on this. The metamodels for specification of data types, expressions and architecture models are summarized in the second layer (Foundation). The third layer, Resource, contains objects for definition of source and target structures of the Data Warehouse. The packages for transformation and (multidimensional) evaluation processes are in the Analysis layer. The Management layer defines metamodels for controlling and monitoring. The packages of each of the individual layers are summarized in this figure.

The Object Model layer is structured into the Core, Behavioral, Relationships and Instance packages. The Core meta model is independent of other packages and covers the fundamental metamodel classes and associations. The referred aspects of model elements such as operations, events and interfaces are summarized in the Behavioral metamodel. The various types of relation are defined in the Relationships package, and the Instance package forms the basis for the production of data instances.

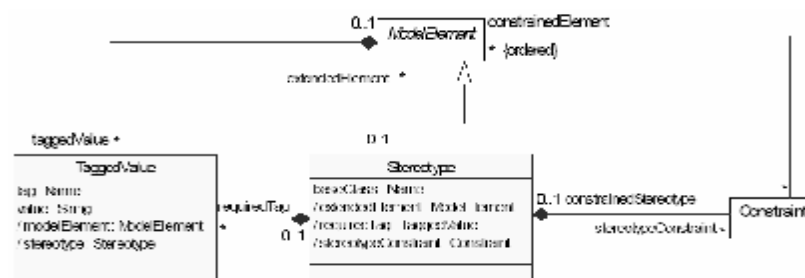
The Foundation layer comprises the following supplementary metamodels: Business Information describing information and partners, Data Types describing elementary and complex data types, Type Mapping describing the mapping of simple data types to each other, Keys and Indexes representing indices and keys as well as expressions, and Software Deployment. The Resource layer comprises metamodels for the description of data structures, for instance relational databases in the Relational package and multi-dimensional data structures in the Multidimensional package, that are independent of specific systems and hence specify general elements such as dimensions and dimension objects. Structured data types are taken into consideration in the Record package, and XML files are supported by the XML package.

(to be continued on next page)



## Extending the CWM

- Tagged Values based upon the package *Core*
- Stereotypes also based upon the package *Core*
- Most flexible: Extension classes



151

TDWI World Conference Chicago '06, Michael Hahne

The Common Warehouse Metamodel enables an architecture- and platform-independent framework for Data Warehouse metadata. Almost all generally relevant aspects are covered in the various packages, which serve as a common basis for general metamodels of various Data

Warehouse systems. The CWM also requires enabling a framework for metamodels, which considers the characteristics of individual software tools and offers mechanisms for flexible extension of the metamodels.

Two fundamental extension techniques are provided for extending the Common Warehouse Metamodels. On one hand, the UML object model provides options for expansion over Tagged Values and Stereotypes; on the other, non-normative extension possibilities are available

with the definition of additional metamodel packages.

Further attribute values, of the form attribute name (tag) = attribute value (value), can be added as supplements to model elements through the Tagged Values extension. On one hand, this extension is independent of version changes in the CWM specification, but on the other hand is limited to string data types. Therefore values cannot be retrieved directly by standard interfaces. Several Tagged Values can be grouped in terms of content by stereotypes, which also offer no possibility to define additional associations.

The possibility of defining additional metamodel elements such as classes, associations and packages offers very flexible ways for extending the Common Warehouse Model. This method should be closely modeled on the concepts of the Meta Object Facility, and falling back on

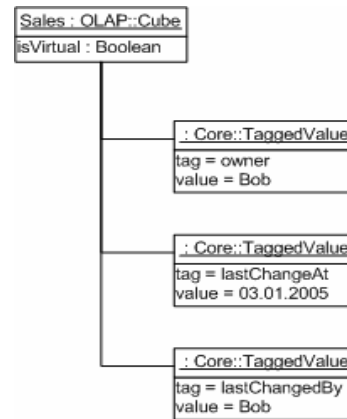
inheritance possibilities, so that the extension does not depart too much from the CWM standard. Some examples of non-normative extensions of the CWM are offered by the OMG Extension Packages, which for example suggest metamodels for specific system products such as Hyperion Essbase and Oracle Express.

References:

Object Management Group (OMG): Common Warehouse Metamodel (CWM) Specification. Volume 2 – Extensions, Version 1.0, OMG, o.O., 2001.

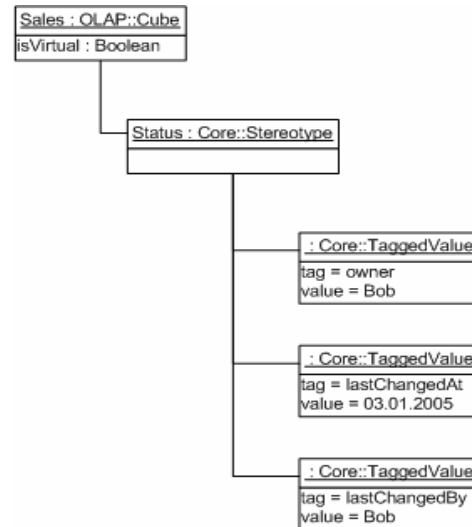
## Extending the CWM: Tagged Values

- Tagged Values: Taken from the UML specification (Package *Core*)
- Extension by pairs: Attribute name (tag) = Attribute value (value)
- Independent of release changes in the CWM specification
- Restricted to string values
- No direct access (The user must know about their existence)
- No additional associations in the metamodel



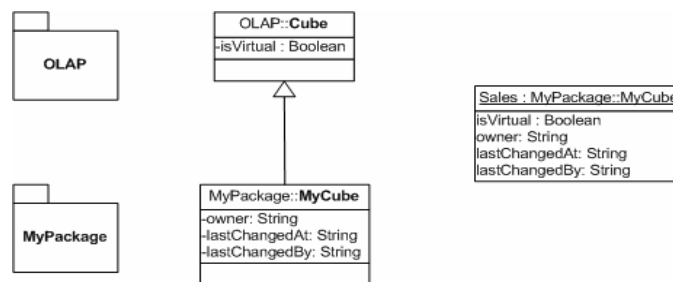
## Extending the CWM: Stereotypes

- Stereotype: Taken from the UML specification (Package *Core*)
- Independent of release changes in the CWM specification
- No additional associations in the metamodel
- Stereotypes are used to group multiple Tagged Values



## Extending the CWM: Extension classes

- Most flexible way to extend the CWM: User-defined Classes, Associations and Packages
- Degree of freedom for design very high
- Risk of departing from standard
- Inheritance mechanisms should be used
- Dependent from release changes of the CWM
- OMG's Extension Packages are based upon this concept (Essbase, Express)



## Need to extend the CWM

- Highly focused on technical metadata
- Missing concepts for describing user rights
  - Heterogenous DW landscapes are frequently characterized by various concepts for user administration and the coordination of user rights and roles
  - Standardization of metadata regarding user rights and authorities enables:
    - Adjustment of user rights
    - Increase transparency and thereby avoiding security leaks

A detailed examination of the Common Warehouse Metamodel shows that a very strong emphasis on technical metadata. Some suggestions for meaningful extensions have been specified in literature.

Data Warehouse systems offer very flexible data access possibilities, and so require a special consideration of aspects relating to data security and authorization administration. Generally, each component of a heterogeneous Data Warehouse system landscape implements its own concepts for the administration of users, roles and authorizations. A standardization of authorization-related metadata would make the alignment of authorizations possible, enabling increased transparency in order to avoid security holes. A unified log-in functionality could be developed on this basis. So far, however, the CWM does not offer any possibilities for metadata representation of authorizations and users.

## Need to extend the CWM

- Missing concepts for quality management
  - Conceptual, functional, technical
  - Acceptable intervals and alerting mechanisms
- Insufficient consideration of multiple mechanisms of report grouping and distribution via different channels
- The Organizational Data Warehouse process can be incorporated through extension of the CWM

Quality management, which enables BI applications to ensure conceptual, functional and technical quality during the phases of planning and operation, is necessary to maintain acceptance of a Data Warehouse solution. Performance indicators and acceptance ranges for data quality and appropriate alerting mechanisms also relate to this. However, there are currently no model components which are able to represent this type of metadata in the Common Warehouse Metamodel.

Along with Knowledge Workers, who perform very diverse analyses with flexible tools, there are also pure report receivers and users of standard report solutions who must be acknowledged as users of Data Warehouse systems. Various mechanisms for grouping reports and distributing reports over various channels, as well as elements of the report definitions, are insufficiently covered in the Common Warehouse Metamodel. Therefore they still offer potential for the improved integration of different analysis tools by appropriate model components of an extension of the CWM.

The model components implemented in the CWM are mainly concerned with the static aspects of a Data Warehouse. Adequate metamodels are available only for the representation of transformation processes. Adequate model components for the organizational component of a Data Warehouse are notably missing from the CWM. Extension of the CWM is a good approach for considering organizational Data Warehouse processes.

## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- Common Warehouse Metamodel (CWM)
- **Implementation of the CWM in BW**

## BW CWM package: *types*

<b>Management</b>	<b>whm</b>		
<b>Analysis</b>	<b>transformation</b>	<b>bex</b>	<b>authorization</b>
<b>Resource</b>	<b>core</b>	<b>relational</b>	<b>olap</b>
<b>Foundation</b>	<b>types</b>		<b>foundation</b>

- BW specific data types:
  - Boolean
  - Integer
  - Name
  - String
  - Time

## BW CWM package : *foundation*

<b>Management</b>	<b>whm</b>		
<b>Analysis</b>	<b>transformation</b>	<b>bex</b>	<b>authorization</b>
<b>Resource</b>	<b>core</b>	<b>relational</b>	<b>olap</b>
<b>Foundation</b>	<b>types</b>		<b>foundation</b>

- Associations, e.g. for the connection of objects along the data flow in BW
- Attributes for object properties such as owner, status and caption, for example
- Business Content information

## BW CWM package : *core*

<b>Management</b>	<b>whm</b>		
<b>Analysis</b>	<b>transformation</b>	<b>bex</b>	<b>authorization</b>
<b>Resource</b>	<b>core</b>	<b>relational</b>	<b>olap</b>
<b>Foundation</b>	<b>types</b>		<b>foundation</b>

- Class definitions for
  - InfoObjects
  - Currency conversion
  - Persons
  - ABAP code
  - Various source systems

## BW CWM package : *relational*

<b>Management</b>	whm		
<b>Analysis</b>	transformation	bex	authorization
<b>Resource</b>	core	relational	olap
<b>Foundation</b>	types		foundation

- ODSObjects
- InfoSets
- Indices

## BW CWM package : *olap*

Management	whm		
Analysis	transformation	bex	authorization
Resource	core	relational	olap
Foundation	types		foundation

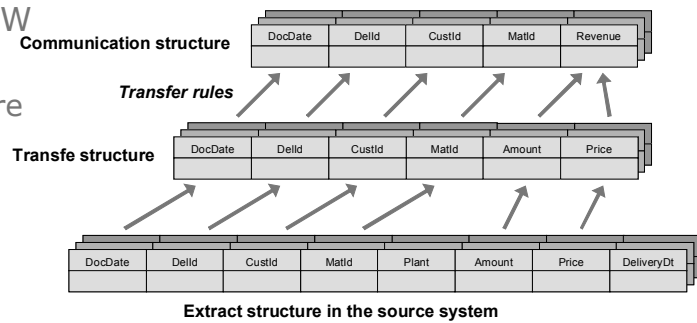
- Classes for InfoProvider and Aggregates
  - Classes *InfoCube*, *InfoProvider*, *MultiProvider*, *VirtualCube*, *Aggregate*
- Reporting Agent
  - Classes *ReportAgentPackage*, *ReportAgentSetting*

## BW CWM package : *transformation*

<b>Management</b>	<b>whm</b>		
<b>Analysis</b>	<b>transformation</b>	<b>bex</b>	<b>authorization</b>
<b>Resource</b>	<b>core</b>	<b>relational</b>	<b>olap</b>
<b>Foundation</b>	<b>types</b>		<b>foundation</b>

- Transformations within BW

- Transfer structure
- Communication structure
- Info-Source
- Update-Rule
- etc.



## BW CWM package : *bex*

<b>Management</b>	<b>whm</b>		
<b>Analysis</b>	<b>transformation</b>	<b>bex</b>	<b>authorization</b>
<b>Resource</b>	<b>core</b>	<b>relational</b>	<b>olap</b>
<b>Foundation</b>	<b>types</b>		<b>foundation</b>

- Classes for data analysis with BEx Analyzer
  - Classes *ExcelWorkbook*, *InfoSetQuery*, *Query*, *QueryCalculatedKeyFigure*, *QueryElement*, *QueryFormula*, *QuerySelection*, *QueryStructure* and *QueryVariable*

## BW CWM package : *authorization*

<b>Management</b>	whm		
<b>Analysis</b>	transformation	bex	authorization
<b>Resource</b>	core	relational	olap
<b>Foundation</b>	types		foundation

- Information about user rights according to the SAP role concept  
– Classes *Authorization*, *Role* and *UserGroup*

## BW CWM package : *whm*

Management	whm		
Analysis	transformation	bex	authorization
Resource	core	relational	olap
Foundation	types		foundation

- Objects needed for operating the Business Information Warehouse

–Classes *EventChain*, *InfoPackage* and *InfoPackageGroup*

- Classes used for organizing and structuring of objects by means of catalogs and areas

–Classes *InfoObjectCatalog*, *InfoArea* and *Application*

## Agenda: Model Management

- Graphical model representation with Visio™
- Metadata in Data Warehousing
- Common Warehouse Metamodel (CWM)
- Implementation of the CWM in BW

## Summary: CWM and BW model management

- The Common Warehouse Metamodel (CWM): industry-wide standard for modelling and exchange of meta data
- The CWM offers a promising approach of integration of components in a heterogeneous data warehouse landscape
- Independent of architecture and platform, no specific tools considered, can be easily and flexibly expanded
- Major weak points of the CWM: insufficient support of
  - User rights
  - Meaning of Data Warehouse content,
  - Quality management
  - Organizational Data Warehouse Process
- SAP BW: Implementation of a CWM compliant extension:
  - Reasonable extension that can be useful as basis for further standardization especially in the area of user rights

The Common Warehouse Metamodel (CWM) passed by the Object management Group was established as industry-wide standard for the modelling of meta data in Data Warehouse systems as well as for their exchange in the meantime. It offers a solid foundation for the modelling of extensive meta data in the context of Data Warehousing which is based upon known standards and is detached from proprietary meta data structures. Flexible possibilities are available for extension, so that also system-specific requirements can be considered.

The missing and/or insufficient meta models within the areas of authorizations, semantics of Data Warehouse contents, quality management, report system and support of organizational processes are substantial weak points of the Common Warehouse Metamodels.

## Agenda: Enterprise Data Warehousing with BW

- Aspects of physical data model
- Staging architectures in BW
- Multi-Layer EDW architecture
- Information Lifecycle Management

## Agenda: Enterprise Data Warehousing with BW

- Aspects of physical data model
- Staging architectures in BW
- Multi-Layer EDW architecture
- Information Lifecycle Management

## Compression of Info-Cubes

**F fact table  
without compression**

Request	Produkt	Periode	Umsatz
1	P1	2003-04	40
1	P2	2003-04	20
1	P3	2003-04	50
1	P4	2003-04	40
1	P5	2003-04	30
2	P1	2003-04	30
2	P2	2003-04	40
2	P3	2003-04	60
2	P4	2003-04	40
2	P5	2003-04	50
3	P1	2003-04	40
3	P2	2003-04	60
3	P3	2003-04	20
3	P4	2003-04	60
3	P5	2003-04	80



compression of all  
Requests up to Request 2

**E fact table**

Produkt	Periode	Umsatz
P1	2003-04	70
P2	2003-04	60
P3	2003-04	110
P4	2003-04	80
P5	2003-04	80

**F fact table**

Request	Produkt	Periode	Umsatz
3	P1	2003-04	40
3	P2	2003-04	60
3	P3	2003-04	20
3	P4	2003-04	60
3	P5	2003-04	80

## Partitioning on database level

**fact table**

Product	Period	Rev.
P1	2003-04	110
P2	2003-04	120
P3	2003-04	130
P4	2003-04	140
P5	2003-04	150
P1	2003-05	110
P2	2003-05	120
P3	2003-05	130
P5	2003-05	150
P6	2003-05	160



**partitioning on  
database level  
by time characteristic**

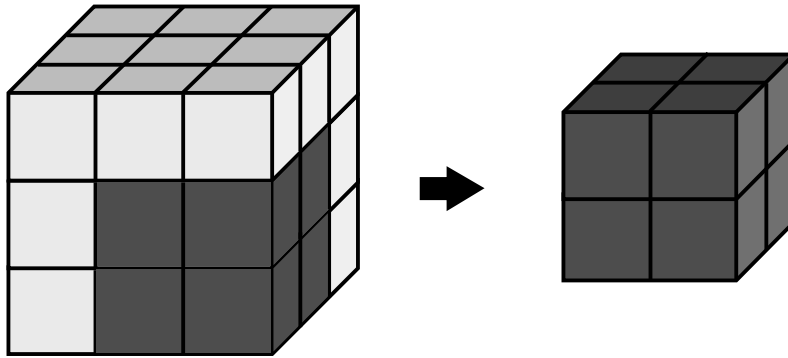
**partitioned fact table for  
2003-04**

Product	Period	Rev.
P1	2003-04	110
P2	2003-04	120
P3	2003-04	130
P4	2003-04	140
P5	2003-04	150

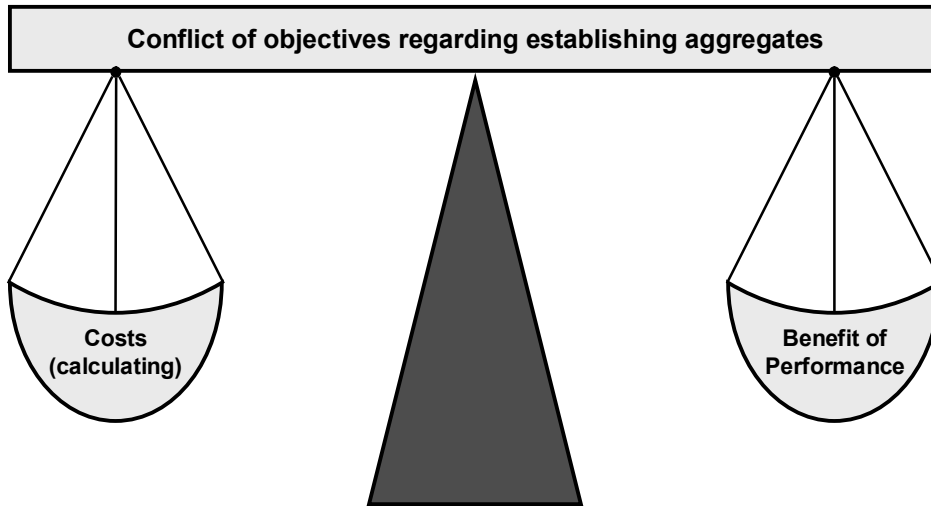
**partitioned fact table for  
2003-05**

Product	Period	Rev.
P1	2003-05	110
P2	2003-05	120
P3	2003-05	130
P5	2003-05	150
P6	2003-05	160

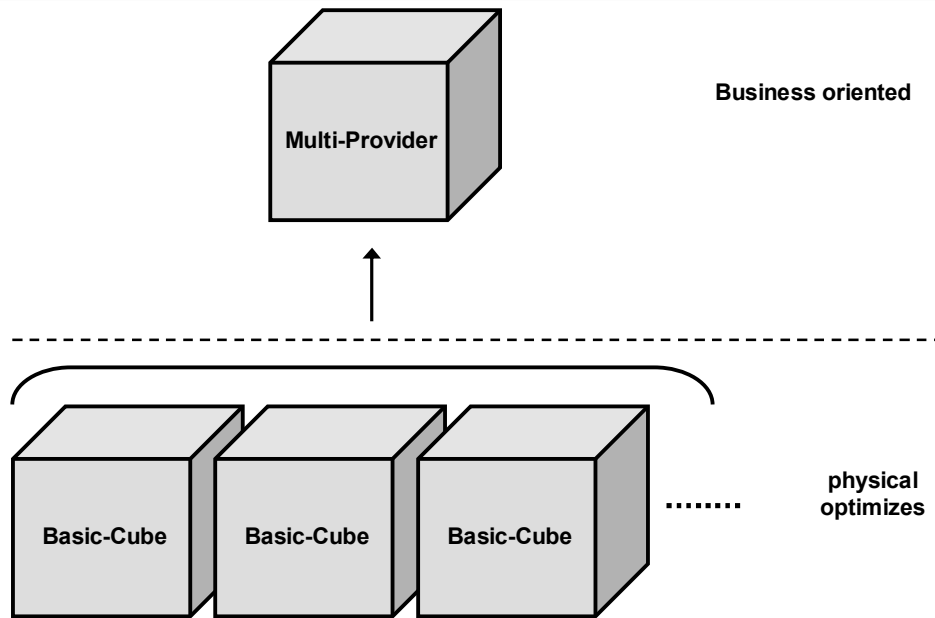
# Aggregates



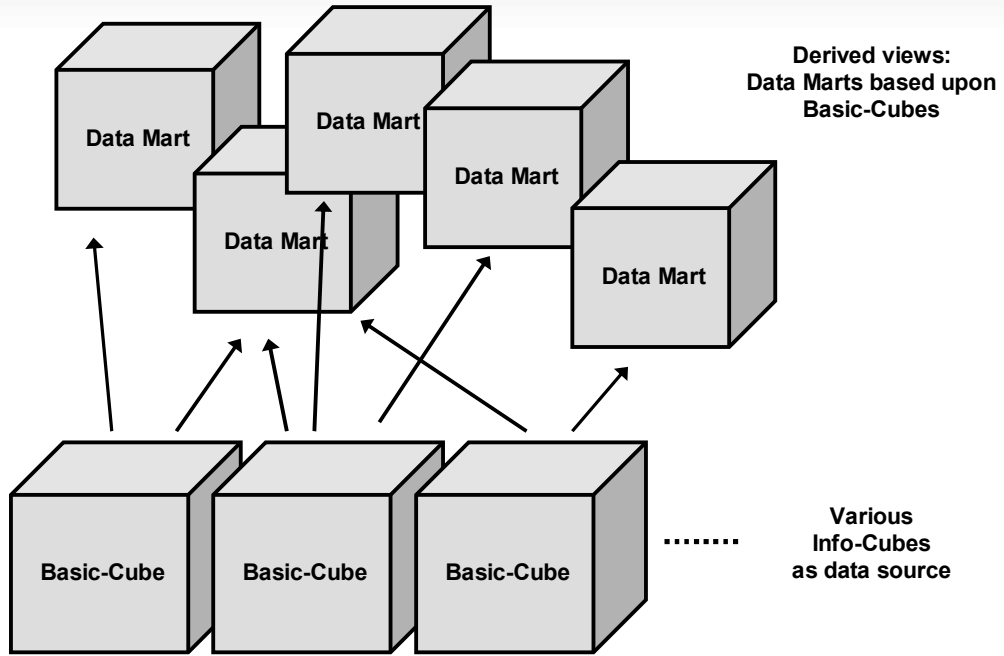
## Aggregates: trade-off



## 2-Layer concept of cube modelling



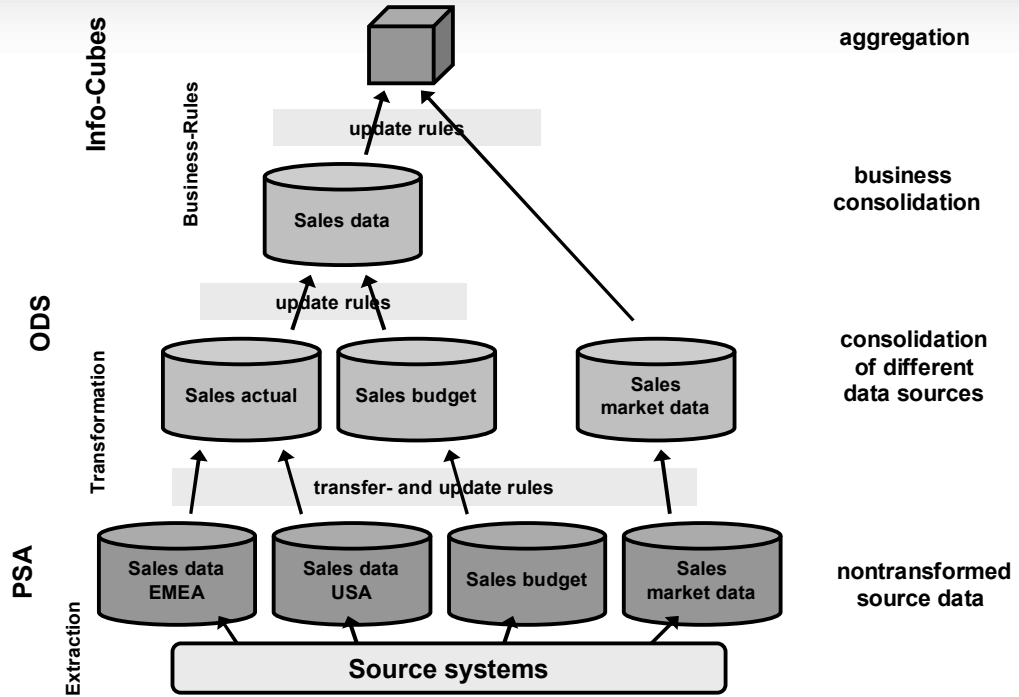
# Data Mart concept



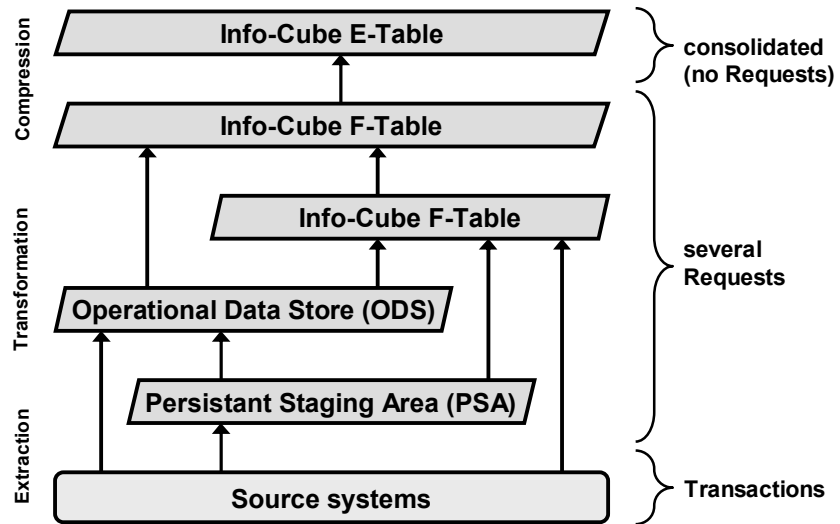
## Agenda: Enterprise Data Warehousing with BW

- Aspects of physical data model
- Staging architectures in BW
- Multi-Layer EDW architecture
- Information Lifecycle Management

# Data flow and integration architecture



## Data flow when using compression of Basic-Cubes

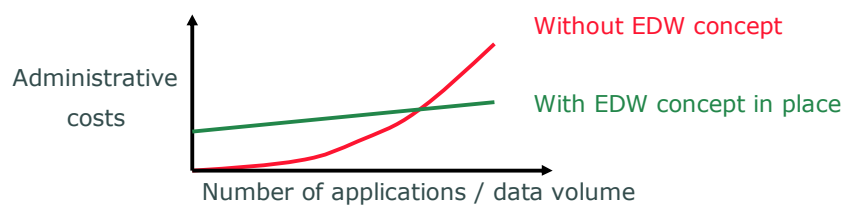


## Agenda: Enterprise Data Warehousing with BW

- Aspects of physical data model
- Staging architectures in BW
- **Multi-Layer EDW architecture**
- Information Lifecycle Management

## Motivation of EDW concept

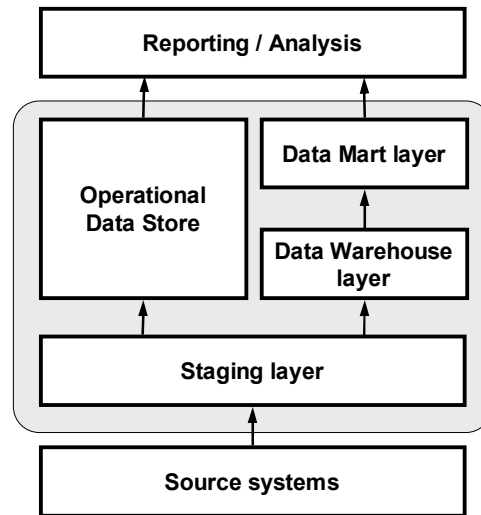
- Data growth
- Increasing number of applications
- Resulting in
  - Increasing administrative costs
  - Higher risk of breakdown of applications
  - Risk of total breakdown



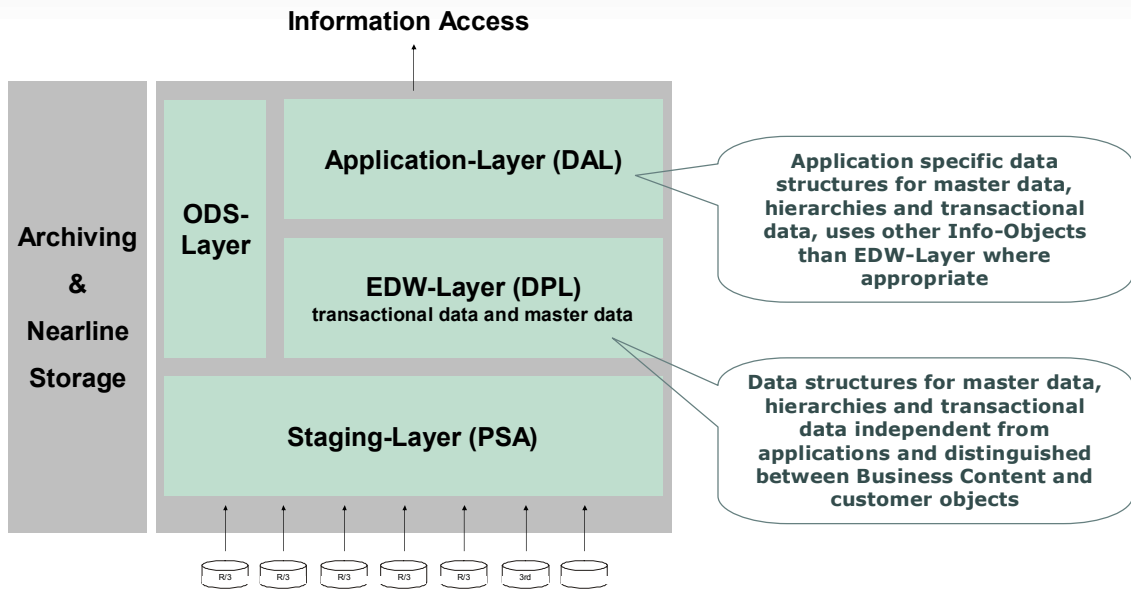
### Enterprise Data Warehousing

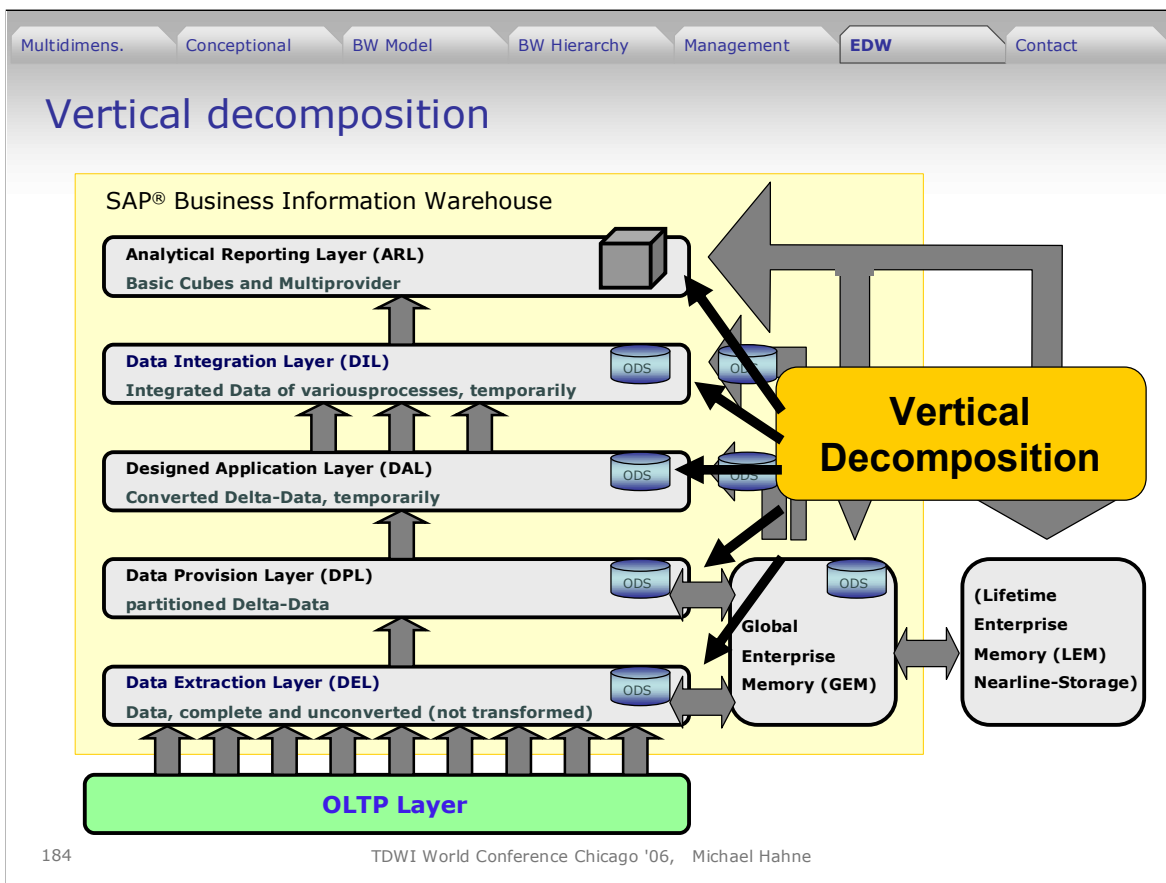
Enterprise data warehousing (EDW) allows customers to create and operate a data warehouse in an enterprise-wide environment. This encompasses the integration of heterogeneous systems, supports various system topologies and their development, and describes a layering methodology for highly flexible access to information.

## Conceptional Multi-Layer-Architecture



# Layer Architecture





The vertical decomposition ensures that reloads, that might be necessary sometimes, can be handled completely in BW itself without the need of data extraction from source systems again.

The Data Extraction Layer as well as the Data Integration Layer are optional and can be left out. But due to data growth it's good advice to implement them in the first place.

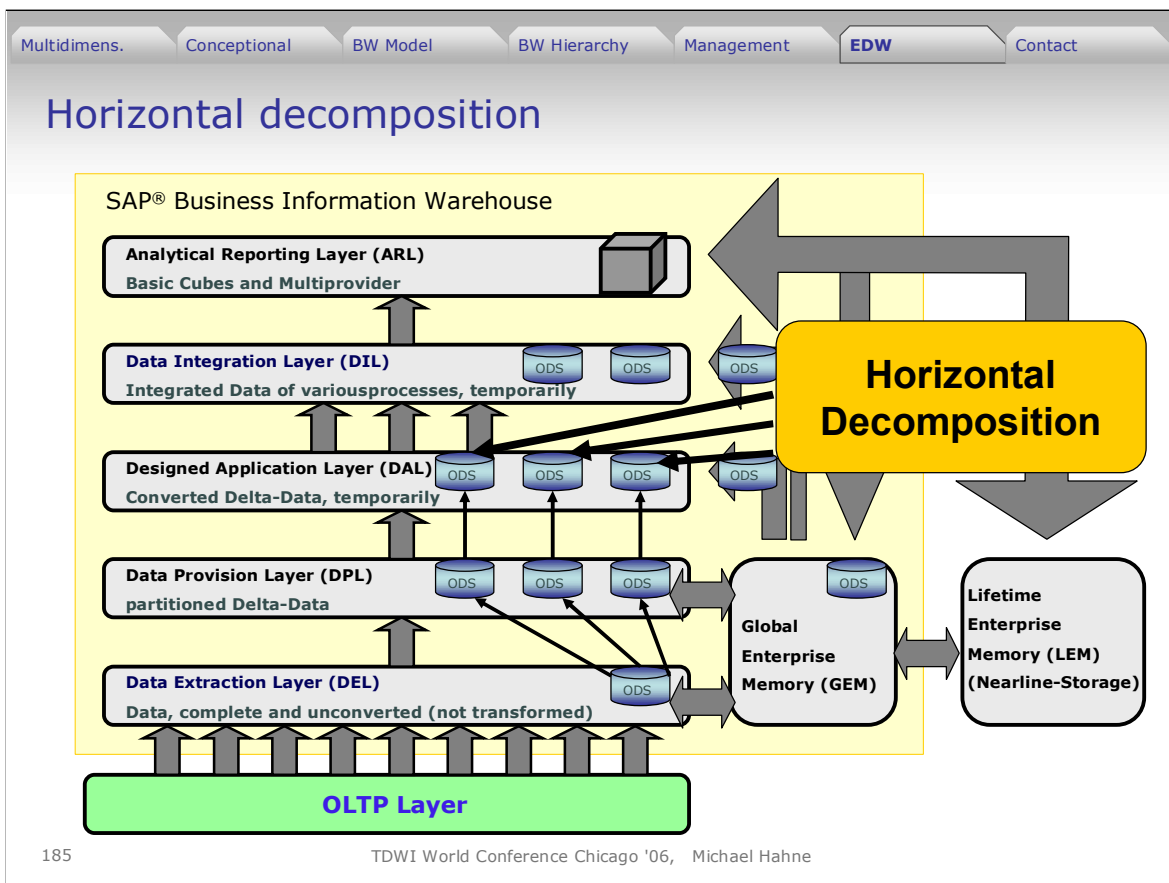
The basic principle of this multi layer approach is best described by the increasing specialization regarding the analytical application.

In the Data Extraction Layer data is stored unconverted and without applying any transformations.

The first partitioning is set up in the Data Provisioning layer. This enables faster delta load mechanisms.

The Data Application Layer focuses on the reporting requirements and stores data converted and enriched. In addition the Data Integration Layer enables the integration of data from different business processes.

The first layer that uses InfoCubes is the Analytical Reporting Layer. It consists of highly partitioned InfoCubes that are combined in MultiProviders. This principle strongly relates to the two layer concept of cube modelling mentioned earlier. This enables data models for reporting that are tightly connected to business requirements and in addition the physical data storage is optimized with regard to query performance.



By means of horizontal decomposition of Data Provision Layer as well as Designed Application Layer, reloads and rebuilds of data targets affect only the relevant line in the ETL process. In addition, the enrichment of data can be implemented in different ways for different values for a partitioning characteristic. For example, the enrichment of data often differs with regard to characteristics like sales organization, plant or business unit. Implemented in this way, the enrichment is designed very lean and more comprehensible.

Hence, the so-called single point of truth is implemented in the Designed Application Layer and Data Provision Layer.

DPL and DAL are the only layers where harmonizing and enrichment of data is implemented.

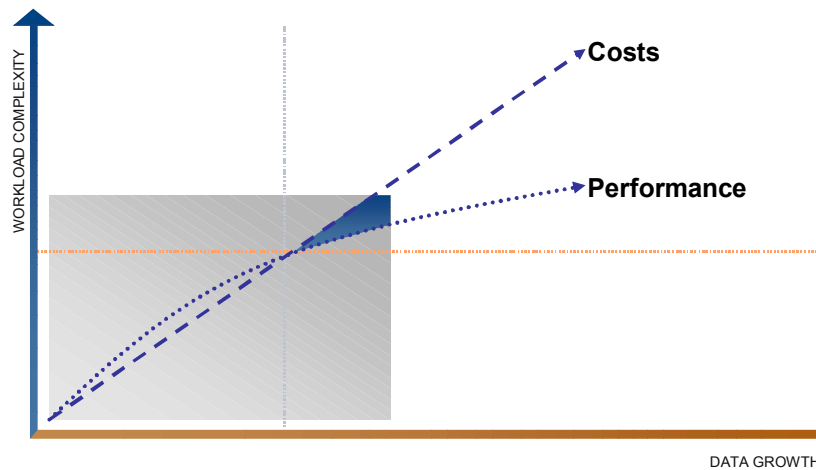
The Global Enterprise Memory enables fast load processes.

## Agenda: Enterprise Data Warehousing with BW

- Aspects of physical data model
- Staging architectures in BW
- Multi-Layer EDW architecture
- Information Lifecycle Management

## Existing Warehouse Under Stress

- The Warehouse can cope only to a point
  - Incremental growth already planned
  - More tuning and performance measures taken



187

TDWI World Conference Chicago '06, Michael Hahne

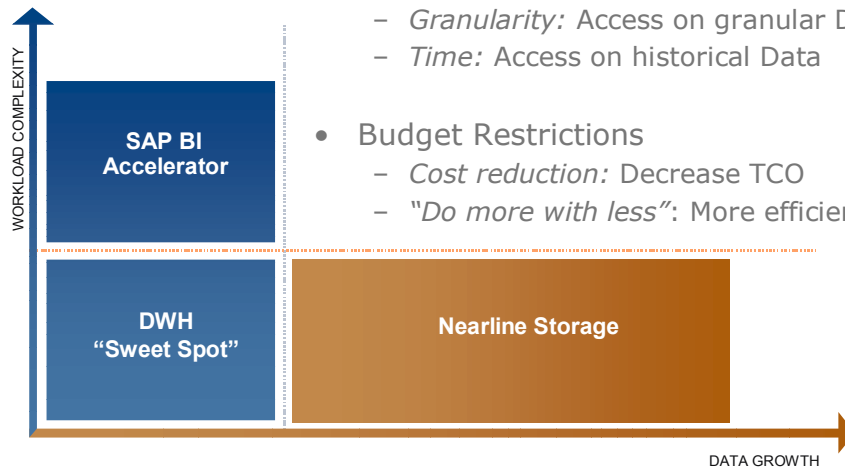
The data warehouse is an essential tool for any organization looking to create a single consistent view of their business to help guide critical decisions. Success in building such a view depends on the ability to access data from as many relevant sources as possible. However, the world of analytic data warehousing is currently facing some daunting new challenges that threaten to hinder access to the full range of corporate data.

The recent explosive growth in the volumes of data generated by organizations in the course of normal business operations has been widely noted. Analysts have estimated that data volumes are growing in excess of 100 percent of one year's data annually, even before considering new legal requirements for data retention. Beyond the simple growth in generated operational data, however, increasingly complex analytical demands are also contributing to the expansion in the amounts of data that must be handled by the data warehouse, and are becoming difficult to satisfy using a traditional approach to data management. Business users are recognizing that to gain a better competitive edge, their data warehouses need to become more powerful in a variety of ways:

- They must offer a more comprehensive view of the business and the competitive environment, incorporating information from across the enterprise and beyond
- The information they contain must be more up-to-date and delivered more quickly to users
- Detailed data must be readily available "just-in-case", for example to satisfy one-off analytical requests requiring the construction of new SAP® Business Information Warehouse (SAP BW) InfoCubes/Data Marts
- They must be more accessible to a greater number of users around the world, and around the clock.

## SAP BI Accelerator and SAP BI Nearline Concepts

- Ability to compete in the Business
  - *Availability*: worldwide, 24x7, ad hoc
  - *Actuality*: Information JIT
  - *Performance*: fast Access
  - *Application Area*: enterprise-wide
  - *Granularity*: Access on granular Data
  - *Time*: Access on historical Data
- Budget Restrictions
  - *Cost reduction*: Decrease TCO
  - *"Do more with less"*: More efficiency

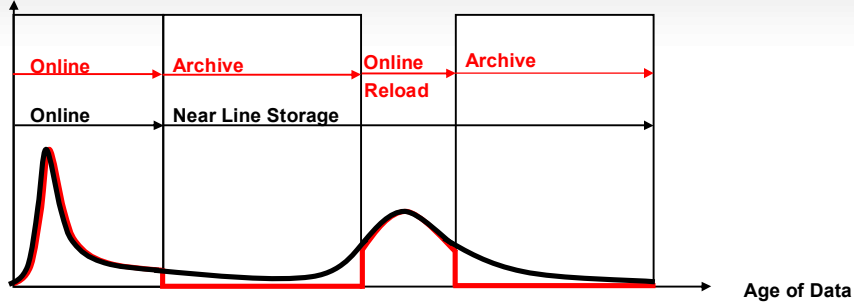


## Data Aging: Categorizing Information According to Importance

	Online Database	Near-Line Storage	Classic Archive
Frequently read / changed data	ü		
Rarely read data	ü	ü	
Very rarely read data	ü	ü	ü

## Classic Archiv vs. Near-Line

Access Frequency/Possibility



### Archiving (SAP BW 3.X)

- n ADK-based (Archive Development Kit) archiving solution for InfoCubes and ODS objects
- n Cost-reduction due to storing data on alternative storage media
- n Archived data must be reloaded into the SAP NetWeaver BI database for analysis purposes

### NLS (SAP NW 2004s BI)

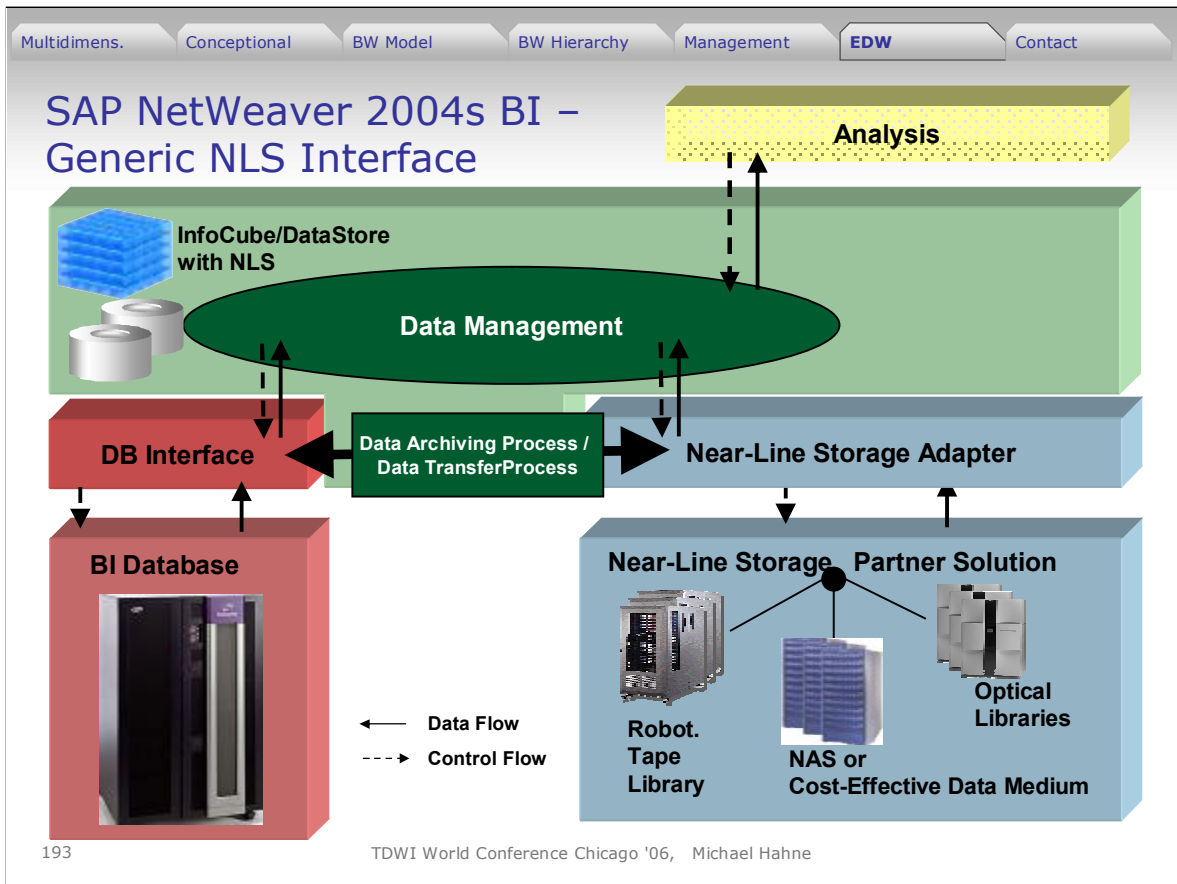
- n SAP NetWeaver BI analyses have direct access to NLS data
- n Availability of historic data while reducing costs
- n Reloading of data into the InfoCube or DataStore Object only necessary in exceptional cases

## Benefits of a Fundamental ILM Strategy for BI

- Increase Volume
  - Manage and use even larger amounts of information more effectively
  - Information available for any time frame for ad-hoc analyses and rebuilds
- Reduce Resource Consumption
  - Reduction of hardware costs for hard drive hardware on the BW side
  - Main memory and CPU as well as costs for system administration
- Increase Availability
  - Quicker, simpler software- and release management in BW
  - Reduced backup- and recovery times
  - Intelligent data access
- Optimize Performance
  - Speed up loading processes in SAP NetWeaver BI
  - SAP NetWeaver BI query response times in the dialog

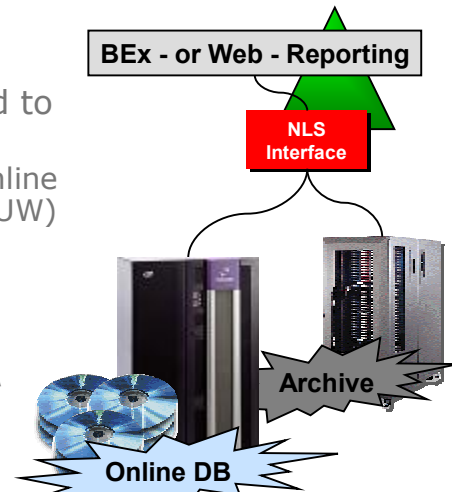
## The Near-Line Storage Solution for SAP NetWeaver BI

- Transparent access to „non-archived“ and „archived“ data for queries
- NLS support for InfoCubes and DataStore objects
- Open interface for certified partners,  
Present development partners:
  - PBS Software – CBW®
  - FileTek – StorHouse®
  - OuterBay - LiveArchive®
  - SAND Technology - Searchable Archive®

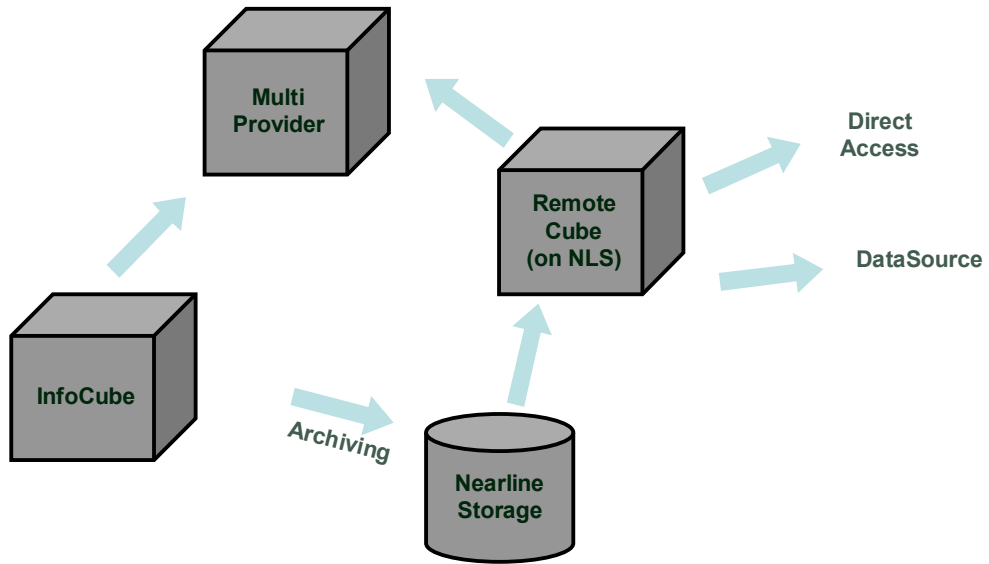


## Consistency between nearline and online

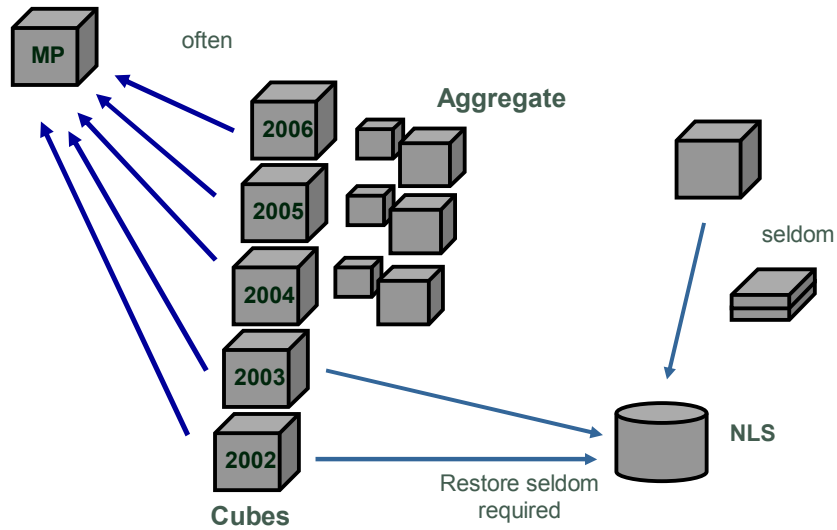
- Analysis and Reporting operate on a combination of online- and near-line datasets. The consistency of the data is an absolute prerequisite.
- Archiving processes into different near-line storage levels have to fulfill transactional requirements with regard to maintaining consistency
  - Archiving and deletion of data in the online database form a logical unit of work (LUW)
  - Rollback mechanisms available for individual archiving steps.
  - The „archive“ gets the character of a database.
  - The archive data are usually ‚read only‘



# NLS with BW 3



# Logical Partitioning of InfoCubes à Near-Line-Concept



## Contact

Dr. Michael Hahne  
Freiherr-vom-Stein-Str. 13a  
55559 Bretzenheim  
Germany

Email:  
michael@hahneonline.de

URLs:  
www.hahneonline.com  
www.hahneonline.de  
www.t-adapt.com

fon 0049 (671) 920 36 62  
fax 0049 (671) 920 36 61

Dr. Michael Hahne  
Productmanager Applications  
SAND Technology

Email:  
michael.hahne@sand.com

URL:  
www.sand.com