Data Warehouse Systems: Design and Implementation

Alejandro VAISMAN
Department of Information Engineering
Instituto Tecnológico de Buenos Aires
avaisman@itba.edu.ar

Esteban ZIMÁNYI
Department of Computer & Decision Engineering (CoDe)
Université Libre de Bruxelles
ezimanyi@ulb.ac.be

Chapter 10: A Method for Data Warehouse Design

Outline

- Approaches to Data Warehouse Design
- General Overview of the Method
- Requirements Specification
- Conceptual Design
- Logical Design
- Physical Design
- Characterization of the Various Approaches

Approaches to Data Warehouse Design

- Few publications devoted to development of data warehouses
- Most written by practitioners, based on their experience in building data warehouses
- The scientific community proposed a variety of approaches too complex to be used in real-world
- Lack of a methodological framework
- Two major methods for the design of data warehouse and data marts:
  - **Top-down design**: User requirements merged **before** the design process begins, and **one schema** for the whole DW is built, from which separate data marts are produced
  - **Bottom-up design**: A separate schema built for each data mart, considering the requirements of users of the specific business area or process
    - These schemas are merged in a global schema for the entire data warehouse
- Choice depends on many factors, e.g.:
  - Professional skills of the development team
  - Size of the data warehouse
  - Users' motivation
  - Financial support
- Even when choosing bottom-up approach, data mart design requires a global DW framework
- Lack of this global framework can make integration difficult and costly in the long term

Approaches to Data Warehouse Design

- **Analysis-driven approach**: Requires identification of key users to give input about organization goals
  - Users play a fundamental role during requirements analysis
  - Users from different levels of the organization must be selected
  - Several techniques used, e.g., interviews or facilitated sessions
  - Specification obtained will include the requirements of users at all organizational levels
- **Source-driven approach**: DW schema obtained by analyzing the underlying source systems
  - Some techniques require conceptual (e.g., E/R model) or relational representations of the operational source schema
  - Less participation of users
- **Analysis/source-driven approach**: Combination of the analysis- and source-driven approaches

These approaches, originally proposed for the requirements specification phase, are adapted to the other DW design phases

Chapter 10: A Method for Data Warehouse Design

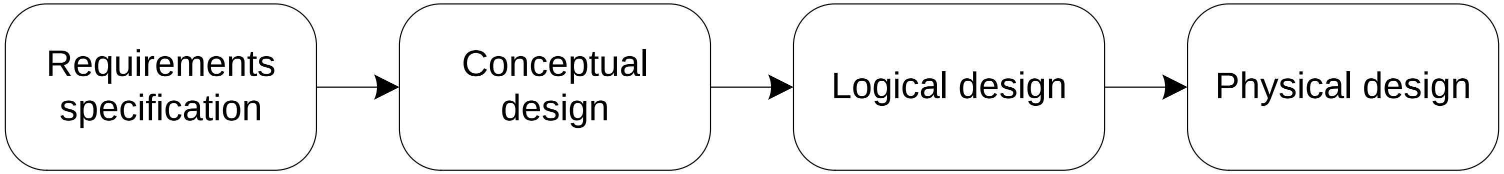
Outline

- Approaches to Data Warehouse Design
- **General Overview of the Method**
- Requirements Specification
- Conceptual Design
- Logical Design
- Physical Design
- Characterization of the Various Approaches

General Overview of the Method

- General method for DW design encompasses various approaches from research and practitioners
- Based on the assumption that DW design should follow the traditional database design phases:
  - Requirements specification
  - Conceptual design
  - Logical design
  - Physical design
- Significant differences between the design phases for databases and DW
- Phases depicted consecutively, although multiple interactions between them
- The phases may be applied to define global DW schema or individual data mart schema
- All phases include specification of business and technical metadata is in continuous development

Phases in data warehouse design



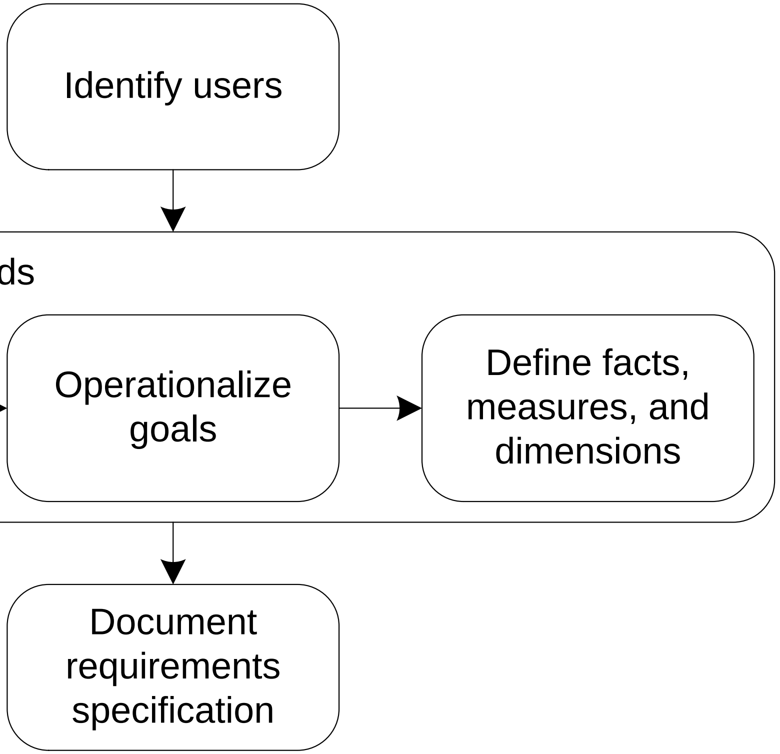
Outline

- Approaches to Data Warehouse Design
- General Overview of the Method
- ➥ **Requirements Specification**
- Conceptual Design
- Logical Design
- Physical Design
- Characterization of the Various Approaches

Analysis-Driven Requirements Specification

- Driving force for developing the conceptual schema: Analysis of user needs
- Requirements express the organizational goals and needs to support the decision-making process

Steps for requirements specification in the analysis-driven approach


Identify users
Determine analysis needs
Define, refine, and prioritize goals
Operationalize goals
Define facts, measures, and dimensions
Document requirements specification

Analysis-Driven Requirements Specification

Phase 1: Identify Users

- Considers users at various hierarchical levels in the organization when analyzing requirements
  - **Executive users**: Require global, summarized information; help in understanding high-level objectives and goals, and the overall business vision
  - **Management users**: Require more detailed information about a specific area of the organization; provide insight into the business processes
  - **Professional users**: Responsible for a specific section or set of services and may demand specific information related to their area of interest
- Identification of users must consider different entities in an horizontal division of the organization

Analysis-Driven Requirements Specification

Phase 2: Determine Analysis Needs

- **Define, refine, and prioritize goals**
  - Goals of the company: the same for everyone, the entire company pursues the same direction
  - Clear specification of goals: Essential to guide user needs and convert them into data elements
  - Analysis needs should be expressed considering both **general** and **specific** goals
  - Specific and general goals must be aligned, to ensure a common direction of the overall process
  - Goal-gathering process: Interviews and brainstorming sessions
  - The list of goals should be analyzed to detect redundancies and dependencies for example:
    - Combine, discard, define as subgoals, etc.
  - This may require additional interaction with users

Analysis-Driven Requirements Specification

Phase 2: Determine Analysis Needs

- **Operationalize goals**
  - For each identified goal, define a collection of representative queries through interviews with the users to capture **functional requirements**
  - Each user is requested to provide a list of queries in natural language
  - The analyst identifies and disambiguates them (e.g., what does "the best customer" mean?)
  - Query analysis and integration: Users review and consolidate queries
  - A prioritization process is finally carried out
    - A possible priority hierarchy: **areas** → **users** → **queries of the same user**
  - **Nonfunctional requirements**, e.g., data quality, also specified and associated to each query
- **Define facts, measures, and dimensions**
  - Analysts identify the underlying facts and dimensions from the queries defined in the previous phase, a manual process
    - E.g.: If in the documentation we have: "Name of top five customers with monthly average sales higher than $1,500", we can guess data elements: customer name, month, and sales
    - Also include which data elements will be aggregated and the functions that must be used
    - Specify the granularities required for the measures, and information about additivity

Analysis-Driven Requirements Specification

Phase 3: Document Requirements Specification

- The information obtained in the previous phases should be documented
- The documentation delivered is the starting point for the technical metadata
- Documentation should include all elements required by the designers and also a dictionary with:
  - Terminology
  - Organizational structure
  - Policies
  - Constraints of the business
  - Other information that may be needed
- For example, the document could express in business terms:
  - What the candidate measures or dimensions actually represent
  - Who has access to them
  - What operations can be done
- This document will not be final, additional interactions could be necessary during conceptual design

Analysis-Driven Requirements Specification for the Northwind Case Study

Identify Users

- Three groups of users identified:
  - **Executive**: The members of the board of directors of the Northwind company, who define the ultimate company goals.
  - **Management**: Managers at departmental levels, for example, marketing, regional sales, and human resources.
  - **Professional**: Professional personnel who implement the indications of the management. Examples are marketing executive officers.

Analysis-Driven Requirements Specification for the Northwind Case Study

Determine Analysis Needs: Goals

- Start with the specification of the goals
- We just address a **general goal**: Increase the overall company sales by ten percent yearly
- This goal can be decomposed into **subgoals**:
  (1) Increase sales in underperforming regions
  (2) For customers buying below their potential, increase their orders (in number of orders and individual order amount)
  (3) Increase sales of products selling below the company expectations
  (4) Take action on employees performing below their expected quota
- Further sessions with the users are carried out to understand their demands in more detail, and operationalize the goals and subgoals (see next slide)
- We assume a common vocabulary has been previously defined in a data dictionary

Analysis-Driven Requirements Specification for the Northwind Case Study

Determine Analysis Needs: Operationalize Goals

(1) Increase sales in underperforming regions:
- (a) Five best selling (measured as total **sales** amount) pairs of customer - supplier countries
- (b) Countries, states, and cities whose customers have the highest total **sales** amount
- (c) Five best selling (measured as total **sales** amount) products by customer country, state, and city

(2) For customers buying below their potential, increase their orders (in number and order amount):
- (a) Monthly **sales** by customer compared to the **sales** for the same customer, in the previous year
- (b) Total number of orders by customer, time period (for example, year), and product
- (c) Average **unit price** per customer

(3) Increase sales of products selling below the company expectations:
- (a) Monthly **sales** for each product category for the current year
- (b) Average **discount** percentage per product and month
- (c) Average **quantity** ordered per product

(4) Take action on employees performing below their expected quota:
- (a) Best selling employee per product per year with respect to **sales** amount
- (b) Average monthly **sales** by employee and year
- (c) Total **sales** by an employee and his/her subordinates during a certain time period

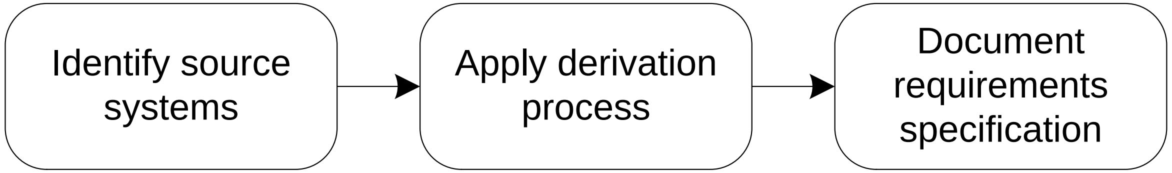
Dimensions and Measures for the Analysis Scenarios of the Northwind Case Study

Dimensions /measures	Hierarchies and levels	Analysis scenarios											
		1a	1b	1c	2a	2b	2c	3a	3b	3c	4a	4b	4c
Employee	**Supervision** Subordinate → Supervisor **Territories** Employee ⇆ City → State → Country → Continent	–	–	–	–	–	–	–	–	–	✓	✓	✓
Time	**Calendar** Day → Month → Quarter → Semester → Year	–	–	–	✓	✓	✓	✓	✓	–	✓	✓	✓
Product	**Categories** Product → Category	–	–	✓	–	✓	–	✓	✓	✓	✓	–	–
Customer	**Geography** Customer → City → State → Country → Continent	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–
Supplier	**Geography** Supplier → City → State → Country → Continent	✓	–	–	–	–	–	–	–	–	–	–	–
Quantity	–	–	–	–	–	–	–	–	–	✓	–	–	–
Discount	–	–	–	–	–	–	–	–	✓	–	–	–	–
SalesAmount	–	✓	✓	✓	✓	–	–	✓	–	–	✓	✓	✓
UnitPrice	–	–	–	–	–	–	✓	–	–	–	–	–	–

Source-Driven Requirements Specification

- Based on the data available at the source systems
- Aims at identifying all multidimensional schemas that can be implemented starting from the available operational databases
- Operational databases analyzed exhaustively to discover the elements that can represent facts with associated dimensions, hierarchies, and measures

Steps for requirements specification in the source-driven approach



Source-Driven Requirements Specification

Phase 1: Identify Source Systems

- **Aim**: To determine the existing operational systems that can be data providers for the warehouse
- External sources are not considered
  - They can be included when the need for additional information has been identified
- Relies on system documentation, preferably represented using the E/R model or relational tables
- In many situations this documentation may be difficult to obtain, e.g., if:
  - Data sources include implicit structures not declared through the DDL
  - Redundant and not normalized structures had been added to improve query response time
  - Database not well designed, or databases reside on legacy systems whose inspection is difficult
- In these situations, reverse engineering can be applied to rebuild the logical and conceptual schemas
- Data sources must be analyzed to assess their suitability to satisfy nonfunctional requirements
- The same data may be available from more than one source, but reliability, availability, and update frequency of these sources may differ from each other

Source-Driven Requirements Specification

Phase 2: Apply Derivation Process

- Many techniques to derive multidimensional elements from operational databases
- All require operational databases represented using either the E/R or relational model
- Facts and measures determined analyzing the existing documentation
- Facts and measures associated to elements frequently updated
  - If the operational databases are relational, they may correspond to tables and attributes
  - If the operational databases are represented using the entity-relationship model, facts could be entity or relationship types, while measures may be attributes of them
- Alternative: Involve users who understand the operational systems and can help to determine what data can be considered measures
- Identifying facts and measures is the most important aspect of this approach
- Procedures to derive dimensions and hierarchies may be automatic, semiautomatic, or manual
  - **Automatic** and **semiautomatic procedures** require knowledge about the conceptual models used for the initial schema and its subsequent transformations
  - **Manual procedures** allow designers to find hierarchies embedded within the same entity or table

Source-Driven Requirements Specification

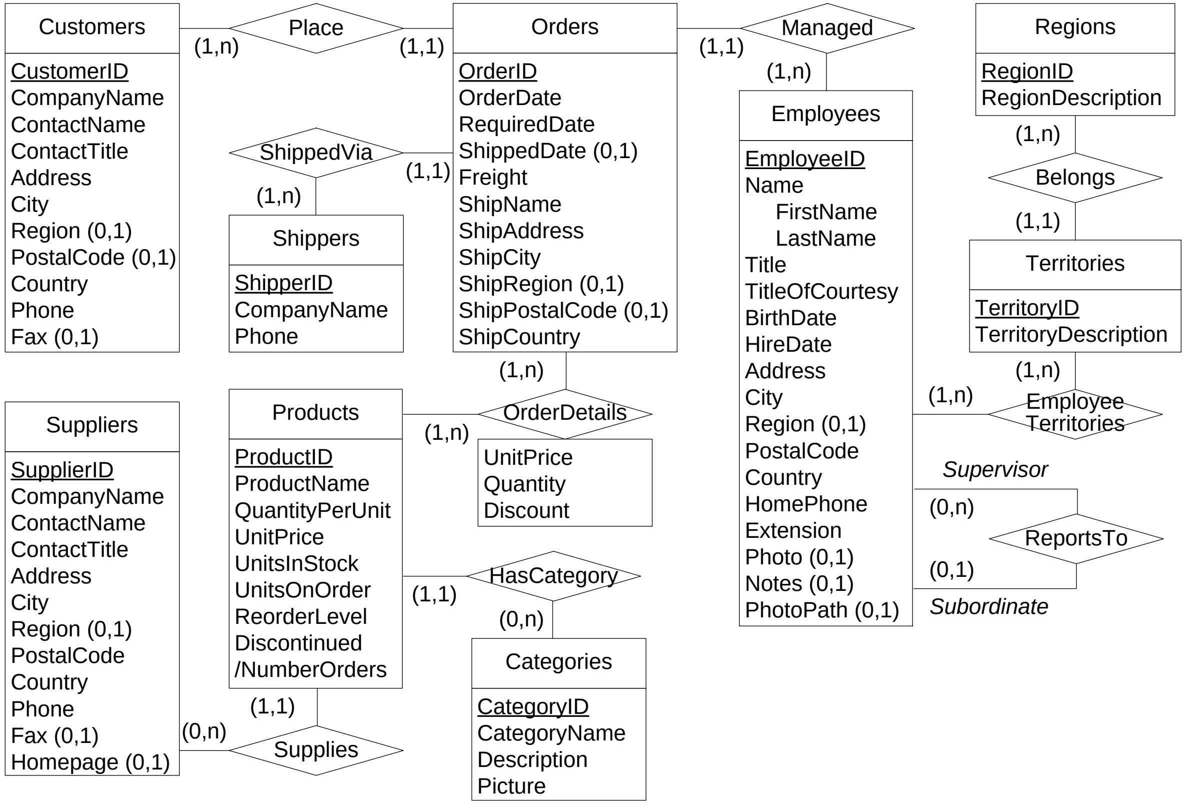
Phase 3: Document Requirements Specification

- Like in the analysis-driven approach, the requirements specification phase should be documented
- The documentation should describe those elements of the source systems that can be considered as facts, measures, dimensions, and hierarchies
- This will be contained in the technical metadata
- It is desirable to involve a domain expert to define business terminology and tell, for example, whether measures are additive, semiadditive, or nonadditive

Source-Driven Requirements for the Northwind Case Study

- We assume E/R schema of the operational database is available, and quality data can be obtained
- We skip the step of identifying the source systems, except for the geographic data

The ER schema for the Northwind database



Source-Driven Requirements for the Northwind Case Study

Apply Derivation Process

- We chose a manual derivation process to provide a more general solution
- We start by identifying candidate facts
- OrderDetails, with attributes that represent numeric data: candidate to be a fact
  - Candidate measures for this fact are attributes UnitPrice, Quantity, and Discount
  - A fact should be associated to an order line → products in OrderDetails are subsumed in the Orders table
  - Each record now becomes a fact, called Sales
- A sales fact is associated with a unique employee (in entity type Employees), shipper (in entity type Shippers), and customer (in entity type Customers)
- Also associated with three dates: order date, required date, and shipped date (potential dimensions)
- The other many-to-many relationship type is EmployeeTerritories, without associated attributes
  - Initially we can consider it a candidate to be a nonstrict hierarchy rather than a fact

Source-Driven Requirements for the Northwind Case Study

Apply Derivation Process

- We now analyze potential dimensions and hierarchies
- We start with the temporal dimension
  - Users have indicated a granularity at the level of day, and that analysis by month, quarter, semester, and year are needed
  - This defines a Time dimension, and the hierarchy Date → Month → Quarter → Semester → Year
  - We call this hierarchy Calendar
  - Three roles for the Time dimension: OrderDate, ShippedDate, and DueDate
- A sales fact is associated to three other potential dimensions: Employee, Customer, and Supplier
- A careful inspection of these geographic data showed that the data sources were incomplete
  - External data sources need to be checked
  - Example: Wikipedia and GeoNames
- We need several kinds of hierarchies to account for all possible political organization of the countries

Source-Driven Requirements for the Northwind Case Study

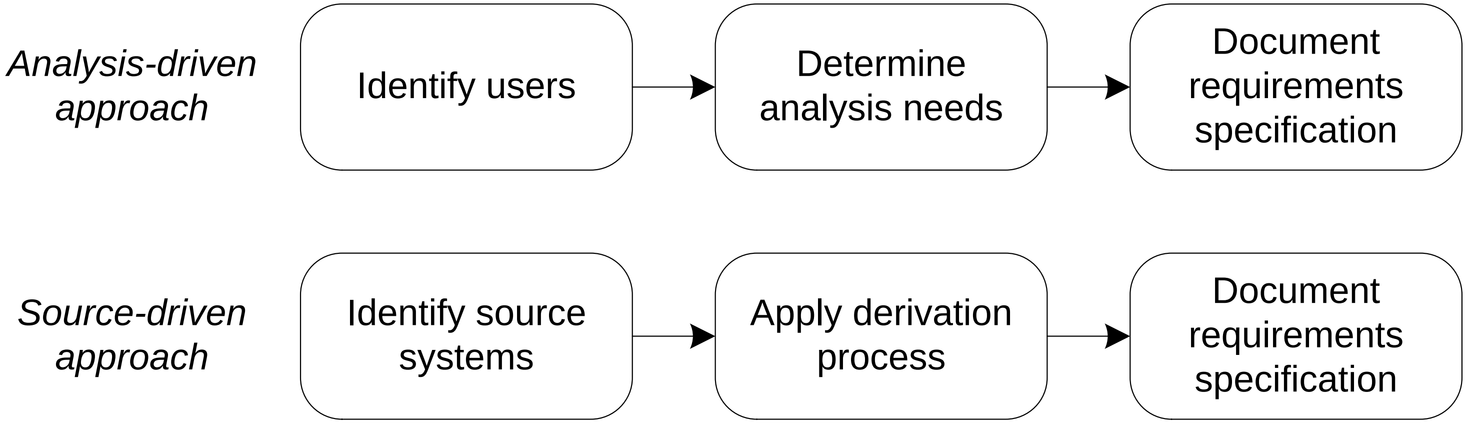
Table below summarizes the result of applying the derivation process. Employee⇆ City indicates a many-to-many relationship between levels Employee and City, all other hierarchies are many-to-one

Facts	Measures	Dimensions and cardinalities		Hierarchies and levels
Sales	UnitPrice Quantity Discount	Product	1:n	**Categories** Product → Category
		Supplier	1:n	**Geography** Supplier → City → State → Region → Country
		Customer	1:n	**Geography** Supplier → City → State → Region → Country
		Employee	1:n	**Supervision** Subordinate → Supervisor **Territories** Employee ⇆ City → State → Region → Country
		OrderDate	1:n	**Calendar** Date → Month → Quarter → Semester → Year
		DueDate	1:n	**Calendar** (as above)
		ShippedDate	1:n	**Calendar** (as above)
		Order	1:1	

Analysis/Source-Driven Requirements Specification

- Combines both previous approaches
- Can be used in parallel to achieve an optimal design
- Two types of activities:
  - One that corresponds to analysis needs
  - The other one represents the steps involved in creating a multidimensional schema from operational databases
- Each type of activity results in the identification of elements for the initial multidimensional schema

Requirements specification in analysis/source-driven approach



Chapter 10: A Method for Data Warehouse Design

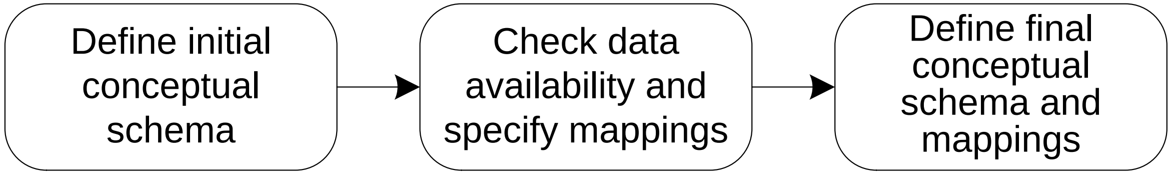
Outline

- Approaches to Data Warehouse Design
- General Overview of the Method
- Requirements Specification
- ➥ **Conceptual Design**
- Logical Design
- Physical Design
- Characterization of the Various Approaches

Analysis-Driven Conceptual Design

- Requirements specification provides the elements for building the **initial conceptual DW schema**
- This schema represents a set of data requirements in a clear and concise manner that can be understood by the users
- Design of a conceptual schema: Iterative process composed of three steps:
  - Development of the initial schema
  - Verifify that the data in this schema are available in the source systems
  - Mapping between the data in the schema and the data in the sources

Steps for conceptual design in the analysis-driven approach


Define initial conceptual schema
Check data availability and specify mappings
Define final conceptual schema and mappings

Analysis-Driven Conceptual Design

Phase 1: Develop Initial Conceptual Schema

- Well-specified analysis requirements lead to clearly distinguishable multidimensional elements: facts, measures, dimensions, and hierarchies
- A first approximation to the conceptual schema can be developed
- Should be validated against its potential usage for analytical processing
- Can be done by first revising the list of queries and analytical scenarios and by consulting the users
- Designers should know the features of the multidimensional model in use and pose more detailed questions (if necessary) to clarify any unclear aspect
  - E.g., a schema may contain different kinds of hierarchies, dimensions can play different roles, derived attributes and measures could be needed
- The refinement of the conceptual schema may require several iterations with the users

Analysis-Driven Conceptual Design

Phase 2: Check Data Availability and Specify Mappings

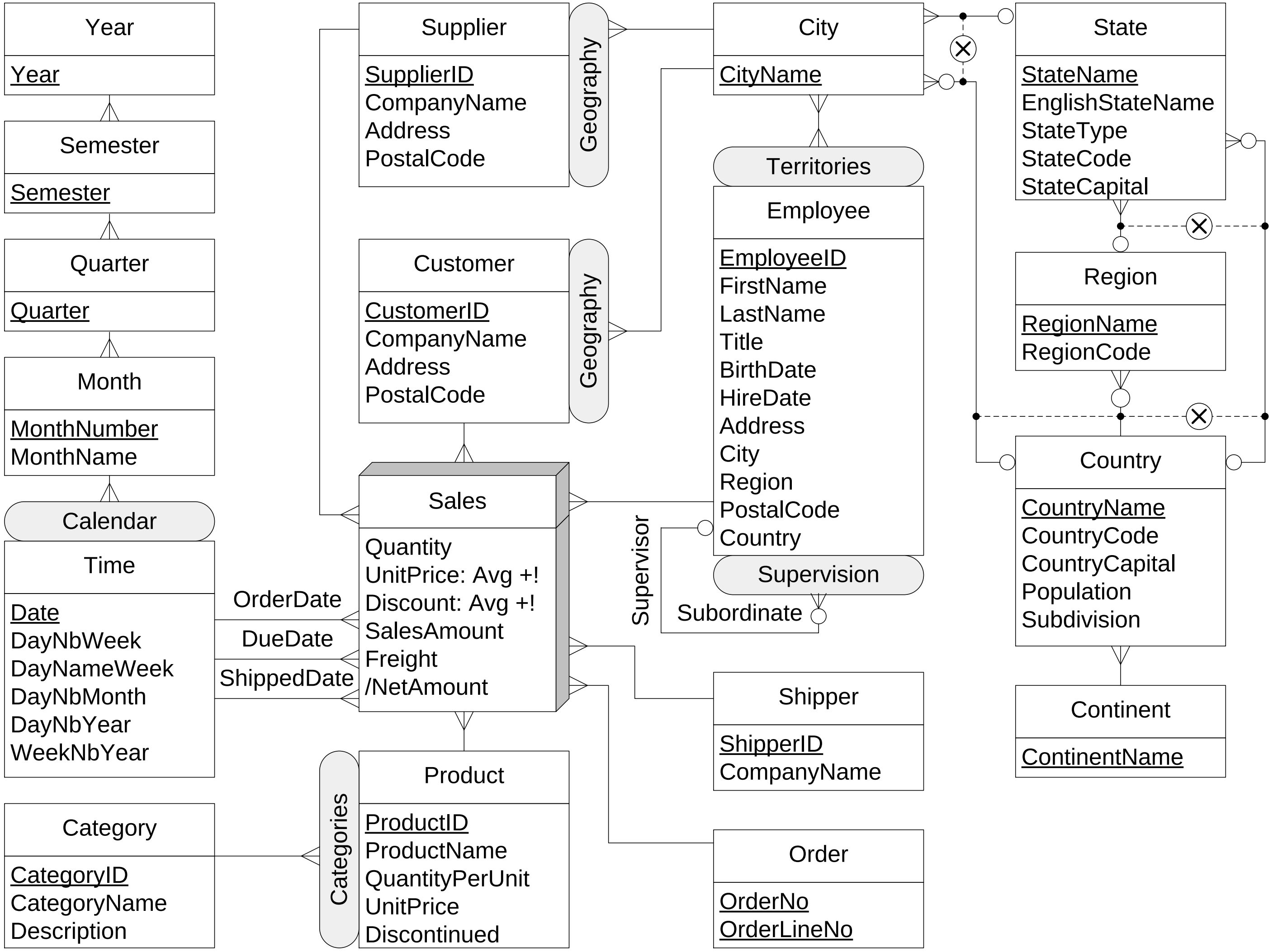
- Data contained in the source systems determine whether the proposed conceptual schema can be transformed into logical and physical schemas
- All elements in the conceptual schema checked against the data items in the sources
- Result of this step: a specification of the mappings for all elements of the multidimensional schema that match data in source systems
- This mapping can be represented either descriptively or using model-driven engineering techniques
- Specification includes also a description of the required transformations, if necessary
- Important: Determine data availability early to avoid developing logical and physical schemas for which the required data may not be available

Analysis-Driven Conceptual Design

Phase 3: Develop Final Conceptual Schema and Mappings

- Data available at the sources for all elements of the conceptual schema → initial schema = final schema
- If not all multidimensional elements can be fed with data from the source systems, a new iteration with the users required
- This is to modify user requirements according to the availability of data
- Result: A new schema should be developed and presented to the users for acceptance.
- Changes to the schema may require modification of existing mappings

Conceptual Schema of the Northwind Data Warehouse



Analysis-Driven Conceptual Design for the Northwind Case Study

Develop Initial Schema

- Conceptual schema is based on the queries and on the table summarizing requirements
- Source data are organized into orders → must transform orders data into sales facts during ETL
- Measures Quantity, UnitPrice, Discount, SalesAmount obtained directly from the sources; Freight is produced in the ETL; NetAmount is derived from the data cube
- Aggregate functions also specified, following the requirements
- Orders are associated with different time instants → Time dimension participates in the Sales fact with roles OrderDate, DueDate, and ShippedDate (not indicated in the requirements table)
- Most scenarios include aggregation over time, to the levels indicated in the queries, then, Time dimension contains four aggregation levels
- Dimension Product and parent level Category follows, e.g., from query "Monthly **sales** for each product category for the current year" (query 3(a))
- Many-to-many relationship between Employee and City defines a nonstrict hierarchy, discovered analyzing the content of the source database in the requirements phase
- For HR analysis (queries 4(a) to 4(c)), we need to analyze sales by employee supervisors, a recursive hierarchy Supervision in dimension Employee

Analysis-Driven Conceptual Design for the Northwind Case Study

Check Data Availability and Specify Mappings

Source table	Source attribute	DW level	DW attribute	Transformation
Products	ProductName	Product	ProductName	—
Products	QuantityPerUnit	Product	QuantityPerUnit	—
Products	UnitPrice	Product	UnitPrice	—
...	...	...	...	...
Customers	CustomerID	Customer	CustomerID	✓
Customers	CompanyName	Customer	CompanyName	—
...	...	...	...	...
Orders	OrderID	Order	OrderNo	✓
Orders		Order	OrderLineNo	✓
Orders	OrderDate	Time	—	✓
...	...	...	...	...

Data transformation between sources and the data warehouse

- Rightmost column indicates whether a transformation is required
- For example, ProductName, QuantityPerUnit, and UnitPrice in the operational database can be used without any transformation in the DW as attributes in attributes of level Product
- Table is a simplification of the information that should be collected, additional detailed documentation should be included for mappings and transformations

Analysis-Driven Conceptual Design for the Northwind Database

Develop Final Conceptual Schema and Mappings

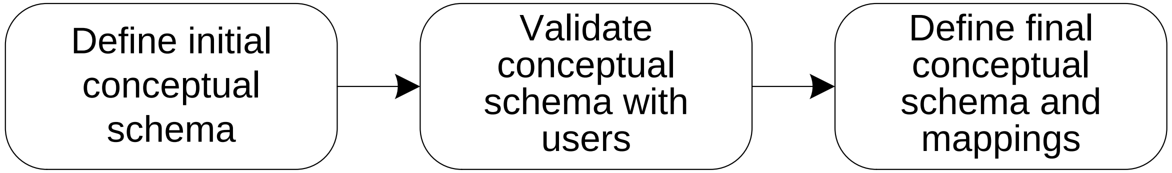
- Revision and additional consultation with users are required to adapt the multidimensional schema to the content of the data sources
- Some of the issues found during the revision process were:
  - We need to create and populate the dimension Time
    - The time interval of this dimension must cover the dates contained in the table Orders of the Northwind operational database
  - The dimensions Customer and Suppliers share the geographic hierarchy starting with City
  - This information is incomplete in the operational database → data for State, Country, and Area must be obtained from an external source
- Metadata for the source systems, DW, and the ETL processes are also developed in this step

Conceptual Design

Source-Driven Conceptual Design

- Once the operational schemas have been analyzed, the initial data warehouse schema is developed
- Not all facts will be of interest for decision support → input from users is required to identify which facts are important
- Users can also refine the existing hierarchies, since some of these are sometimes “hidden” in an entity type or a table
- The initial data warehouse schema is modified until it becomes the final version accepted by the users

Steps for conceptual design in the source-driven approach


Define initial conceptual schema
Validate conceptual schema with users
Define final conceptual schema and mappings

Source-Driven Conceptual Design

Phase 1: Develop Initial Schema

- Multidimensional elements have been identified in the requirements specification phase → development of an initial data warehouse conceptual schema is straightforward
- The usual practice is to use names for the various schema elements that facilitate user understanding
- However, users are familiar with the technical names used in the source systems
    - In this case, a dictionary of names can facilitate communication with the users

Source-Driven Conceptual Design

Phase 2: Validate Conceptual Schema with Users

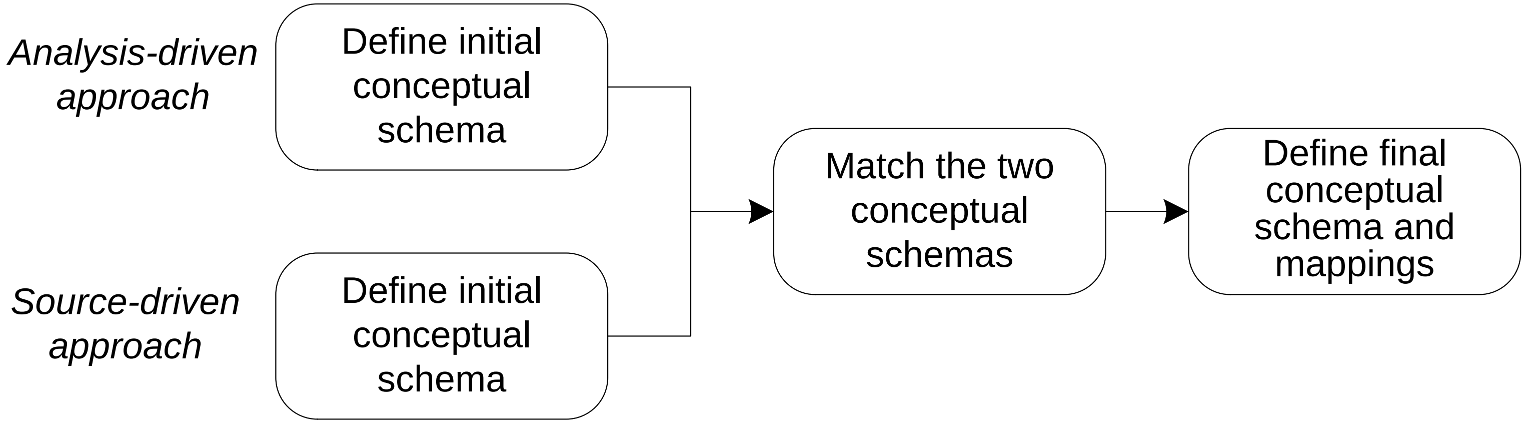
- Start from schema obtained starting from the data sources
- So far participation of the users has been minimal; here, users incorporated in a more active role
- Users are at professional or administrative level, because of their knowledge of the underlying systems
- Initial schema examined in detail, may require modifications for several reasons
  - It may contain more elements than those required for the analysis purposes
  - Some elements may require transformation (e.g., attributes into hierarchies)
  - Some elements could be missing although they exist in the sources (e.g.,due to confusing names)

Phase 3: Develop Final Conceptual Schema and Mappings

- Users' recommendations incorporated into the initial schema, leading to a final conceptual schema that should be approved
- An abstract specification of mappings and transformations (if required) between the data in the source systems and the data in the data warehouse is defined

Analysis/Source-Driven Conceptual Design

Steps for conceptual design in the analysis/source-driven approach


Analysis-driven approach
Define initial conceptual schema
Source-driven approach
Define initial conceptual schema
Match the two conceptual schemas
Define final conceptual schema and mappings

- Two activities: analysis requirements and exploration of the source systems
- Leads to two DW schemas:
  - The schema obtained from the analysis-driven approach
  - The data warehouse schema that can be extracted from the existing operational databases following the source-driven approach
- Both initial schemas must be matched

Analysis/Source-Driven Conceptual Design

- Several aspects should be considered in this matching process
  - Terminology
  - Similarity between dimensions, levels, attributes, or hierarchies
- Solutions proposed in academic literature: Highly technical, complex to implement
- Ideally, user needs covered by the data in the operational systems, no other data are needed
  - Schema is accepted, mappings between elements in sources and the data warehouse are specified
- Additionally, documentation is developed, with warehouse and source systems metadata, etc.
- In real-world this does not occur. Usually, two situations:
  - Users require **less** information than what the operational databases can provide
    - Another iteration of the analysis- and source-driven approaches is required
  - Users require **more** information than what the operational databases can provide; Users may:
    - Reconsider their needs and limit them to those proposed by the analysis-driven solution
    - Require the inclusion of external sources or legacy systems not considered previously
- Each type of activity results in the identification of elements for the initial multidimensional schema

Chapter 10: A Method for Data Warehouse Design

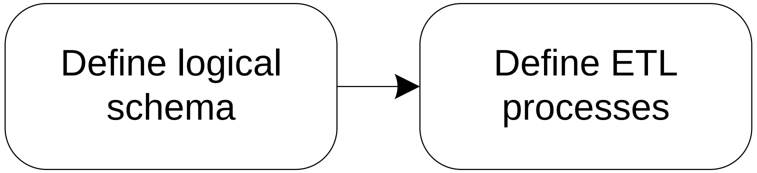
Outline

- ◆ Approaches to Data Warehouse Design
- ◆ General Overview of the Method
- ◆ Requirements Specification
- ◆ Conceptual Design
- ➥ **Logical Design**
- ◆ Physical Design
- ◆ Characterization of the Various Approaches

Logical Design

- Two steps:
  - Transformation of the conceptual multidimensional schema into a logical schema
  - Specification of the ETL processes, considering transformations indicated in the previous phase

Steps for logical design



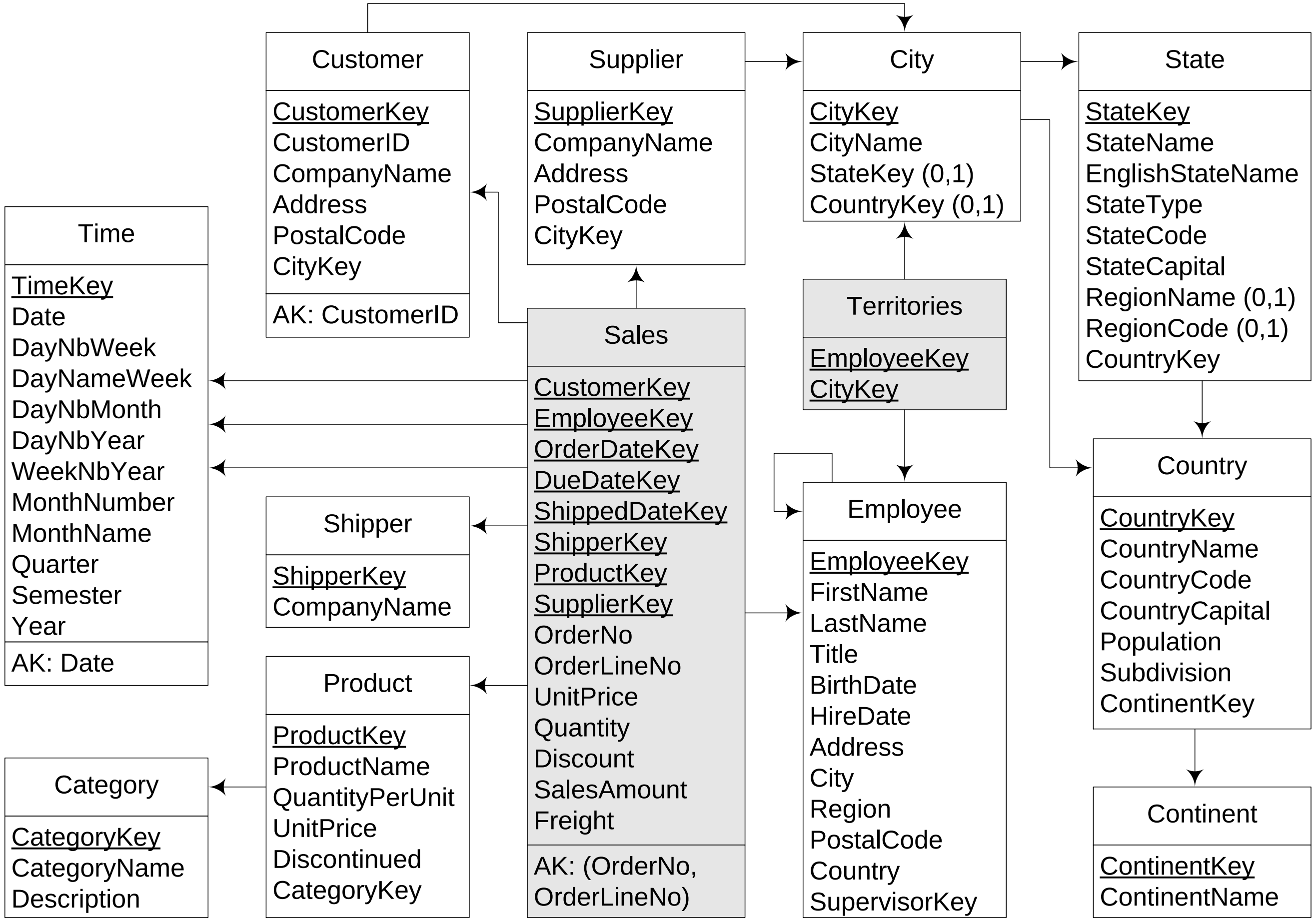
Step 1. Define Logical Schema

- After conceptual design has been completed, map the resulting conceptual schema to a logical schema
- Mapping rules depend on the conceptual model used

Logical Design for the Northwind Case Study

Step 1. Define Logical Schema

Relational schema of the Northwind data warehouse



Customer
CustomerKey
CustomerID
CompanyName
Address
PostalCode
CityKey
AK: CustomerID

Supplier
SupplierKey
CompanyName
Address
PostalCode
CityKey

City
CityKey
CityName
StateKey (0,1)
CountryKey (0,1)

State
StateKey
StateName
EnglishStateName
StateType
StateCode
StateCapital
RegionName (0,1)
RegionCode (0,1)
CountryKey

Time
TimeKey
Date
DayNbWeek
DayNameWeek
DayNbMonth
DayNbYear
WeekNbYear
MonthNumber
MonthName
Quarter
Semester
Year
AK: Date

Sales
CustomerKey
EmployeeKey
OrderDateKey
DueDateKey
ShippedDateKey
ShipperKey
ProductKey
SupplierKey
OrderNo
OrderLineNo
UnitPrice
Quantity
Discount
SalesAmount
Freight
AK: (OrderNo, OrderLineNo)

Territories
EmployeeKey
CityKey

Shipper
ShipperKey
CompanyName

Employee
EmployeeKey
FirstName
LastName
Title
BirthDate
HireDate
Address
City
Region
PostalCode
Country
SupervisorKey

Country
CountryKey
CountryName
CountryCode
CountryCapital
Population
Subdivision
ContinentKey

Product
ProductKey
ProductName
QuantityPerUnit
UnitPrice
Discontinued
CategoryKey

Category
CategoryKey
CategoryName
Description

Continent
ContinentKey
ContinentName

Logical Design for the Northwind Case Study

Step 1. Define Logical Schema

- Based on users' needs, query performance, and data reuse, decide between a star or snowflake schema
- Calendar hierarchy is only used in the Time dimension, for performance reasons we include these hierarchies in a single table → a star representation for the Time dimension
- The hierarchy City, State, Region, Country, and Area is shared by Territories, Geography (for customers), and Geography (for suppliers)
  - To favor reuse, we chose the snowflake representation for this hierarchy
  - Exception: Region, embedded in the table State
  - The hierarchy City → State → Region → Country → Area is ragged
  - Attributes RegionName and RegionCode embedded in the State table (star representation)
  - For other attributes: snowflake solution
  - Example: The City table has embedded the StateKey and CountryKey as optional foreign keys
  - This way, if a city directly belongs to a country, we can reference the country directly
- Territories is a nonstrict hierarchy
  - For mapping, create bridge table Territories referencing both the Employee and the City tables

Logical Design

Step 2. Define ETL Processes

- Before implementing the ETL processes, several additional tasks must be specified in more detail
- All transformations of the source data should be considered
  - Some are straightforward, e.g., the separation of addresses into their components (for example, street, city, postal code)
  - Other required decisions: Whether to recalculate measure values to express them in euros or dollars, or use the original currency and include the exchange rate
  - A preliminary sequence of execution for the ETL processes should also be determined

Chapter 10: A Method for Data Warehouse Design

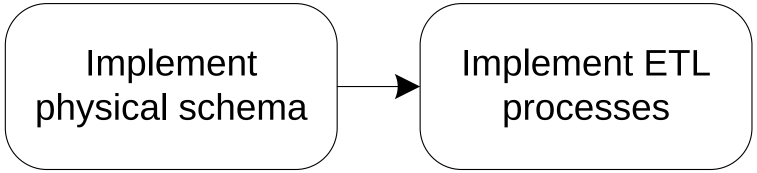
Outline

- Approaches to Data Warehouse Design
- General Overview of the Method
- Requirements Specification
- Conceptual Design
- Logical Design
- ➥ **Physical Design**
- Characterization of the Various Approaches

Physical Design

- Two aspects:
  - One related to the implementation of the data warehouse
  - Another one that considers the ETL processes
- Logical schema is converted into a tool-dependent physical database structure
- Physical design decisions should consider both the proposed logical schema and the analytical queries specified during requirements

Steps for physical design



Physical Design

- Should enable to manage large amounts of data, refresh the DW, perform complex operations, etc.
- Depend on the facilities provided by the DBMS (storage methods, indexes, partitioning, parallel query execution, aggregation functions, and view materialization, etc), e.g.:
  - If a query often requests employee names, dimension Employee can be fragmented vertically: attributes FirstName, LastName, and City in one partition, the other ones in another partition
- The Sales fact table could be partitioned horizontally according to time, if queries frequently require the most recent data
- During physical design we must define indexing scheme
- The designer should be aware of the possibilities of the DBMS that she will use
- SQL Server does not support bitmap indexes, while Oracle does
- SQL Server comes equipped with the option to define column-store indexes
- We must also define which will be the most common queries and the materialized views that we need
- Again, SQL Server does not support materialized views directly, but through indexed views

Chapter 10: A Method for Data Warehouse Design

Outline

- Approaches to Data Warehouse Design
- General Overview of the Method
- Requirements Specification
- Conceptual Design
- Logical Design
- Physical Design
- **Characterization of the Various Approaches**

Analysis-Driven Approach: Summary

Advantages

- Provides a comprehensive specification of the needs of stakeholders from a business viewpoint
- Facilitates a better understanding of the facts, dimensions, and the relationships between them
- Promotes acceptance of the system through continuous interaction with potential users
- Enables the specification of long-term strategic goals

Disadvantages

- The specification of business goals can be difficult, and its result depends on the techniques applied and the skills of the developer team
- Requirements specification not aligned with business goals may produce a complex schema that does not support the decision processes at all organizational levels
- The duration of the project tends to be longer than the duration of the source-driven approach
- The users' requirements might not be satisfied by the information existing in the source systems

Source-Driven Approach: Summary

Advantages

- Ensures that the DW reflects the underlying relationships in the data
- Guarantees that the DW contains all necessary data from the beginning
- Reduces the user involvement required to start the project
- Facilitates development process if well-structured and normalized operational systems exist
- (Semi-) automatic techniques can be applied if E/R or relational schemas exist for operational DBs

Disadvantages

- Only business needs reflected in the underlying source data models can be captured
- System may not meet users' expectations: company's goals and users' requirements not considered
- Difficult to apply when logical schemas of operational systems are hard to understand
- Based on existing data → cannot be used to address long-term strategic goals
- Hierarchies may be hidden in various structures, for example in generalization relationships
- It is difficult to motivate end users to work with large schemas developed for and by specialists
- The derivation process can be difficult without knowledge of the users' needs, since for instance, the same data can be considered as a measure or as a dimension attribute

Analysis/Source-Driven Approach: Summary

Advantages

- Generates a feasible solution, supported by the existing data sources, which better reflects users' goals
- Alerts about missing data (required to support decision-making) in the operational databases
- If the source systems offer more information than what the business users initially demand, the analysis can be expanded to include new aspects not yet considered

Disadvantages

- The development process is complicated, since two schemas are required (one obtained from the requirements, and another derived from the source systems)
- The integration process to determine whether the data sources cover the users' requirements may need complex techniques