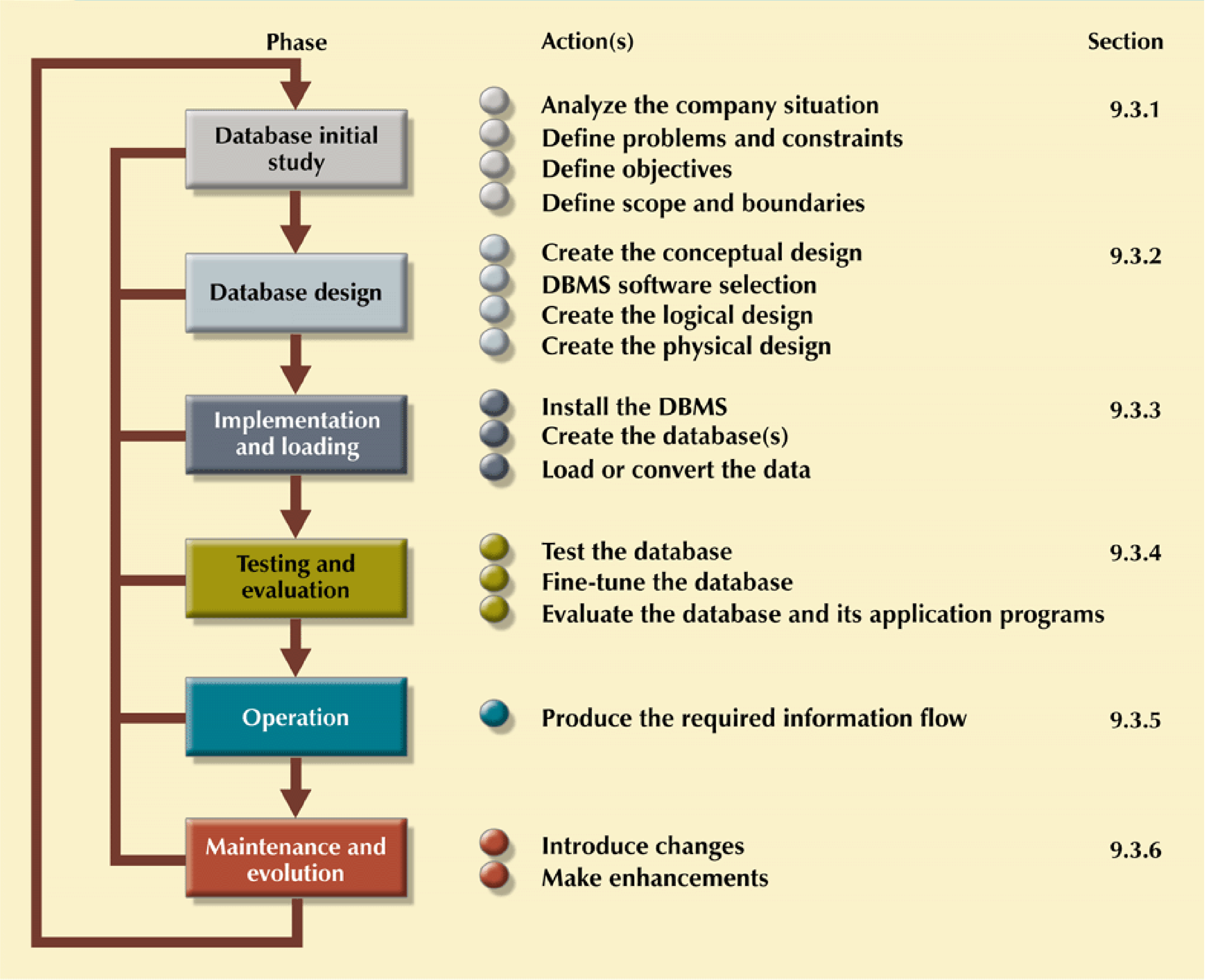
Database Systems:
Design, Implementation, and Management

Lesson 9

The Database Life Cycle (DBLC)

- Six phases:
  - Database initial study
  - Database design
  - Implementation and loading
  - Testing and evaluation
  - Operation
  - Maintenance and evolution

FIGURE 9.3 The Database Life Cycle (DBLC)

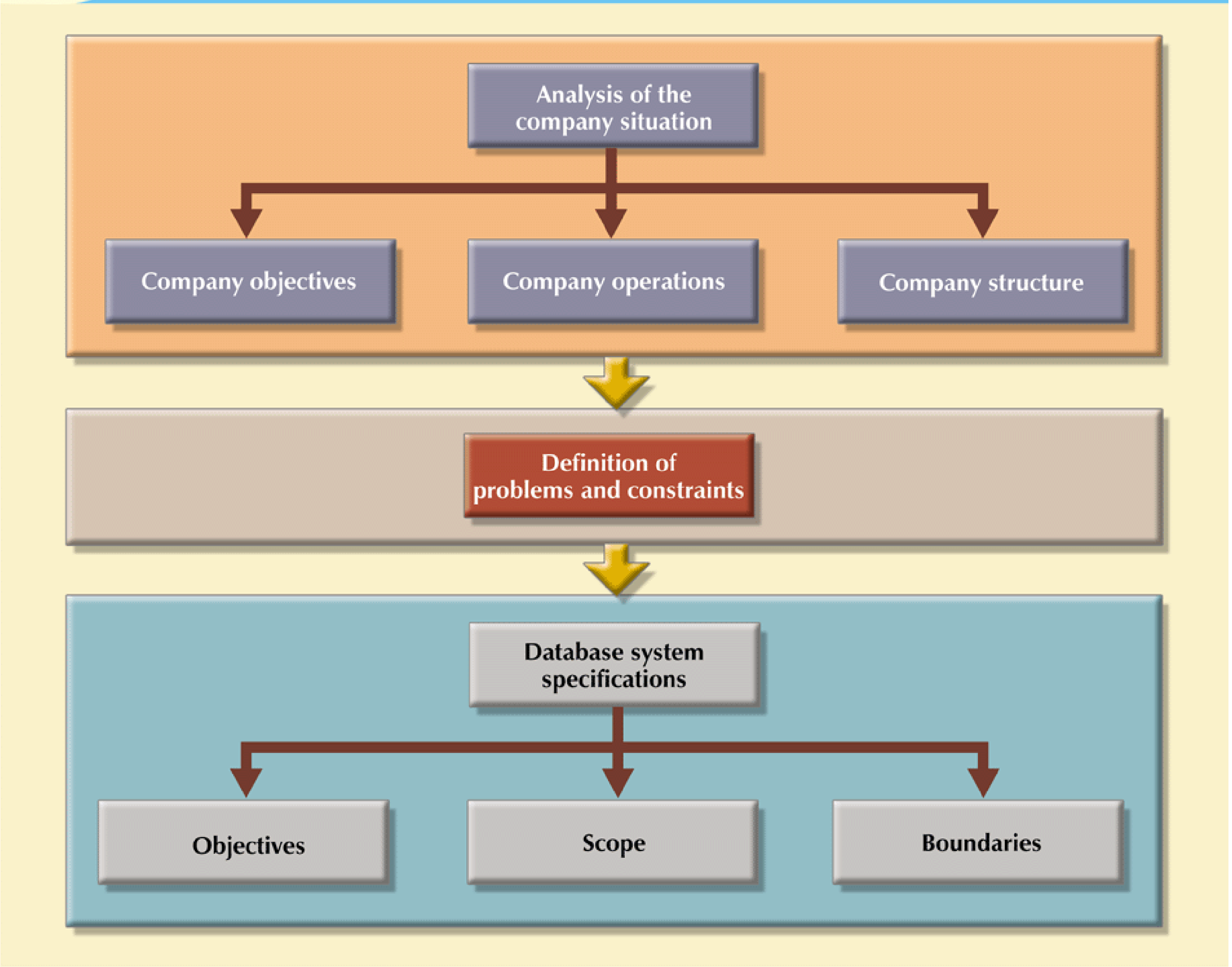


Phase	Action(s)	Section
Database initial study	Analyze the company situation Define problems and constraints Define objectives Define scope and boundaries	9.3.1
Database design	Create the conceptual design DBMS software selection Create the logical design Create the physical design	9.3.2
Implementation and loading	Install the DBMS Create the database(s) Load or convert the data	9.3.3
Testing and evaluation	Test the database Fine-tune the database Evaluate the database and its application programs	9.3.4
Operation	Produce the required information flow	9.3.5
Maintenance and evolution	Introduce changes Make enhancements	9.3.6

The Database Initial Study

- Overall purpose:
  - Analyze company situation
  - Define problems and constraints
  - Define objectives
  - Define scope and boundaries
- Interactive and iterative processes required to complete first phase of DBLC successfully

FIGURE 9.4 A summary of activities in the database initial study


Analysis of the company situation
Company objectives
Company operations
Company structure
Definition of problems and constraints
Database system specifications
Objectives
Scope
Boundaries

The Database Initial Study (cont’d.)

- Analyze the company situation
  - General conditions in which company operates, its organizational structure, and its mission
  - Discover what company’s operational components are, how they function, and how they interact

The Database Initial Study (cont’d.)

- Define problems and constraints
  - Formal and informal information sources
  - Finding precise answers is important
  - Accurate problem definition does not always yield a solution

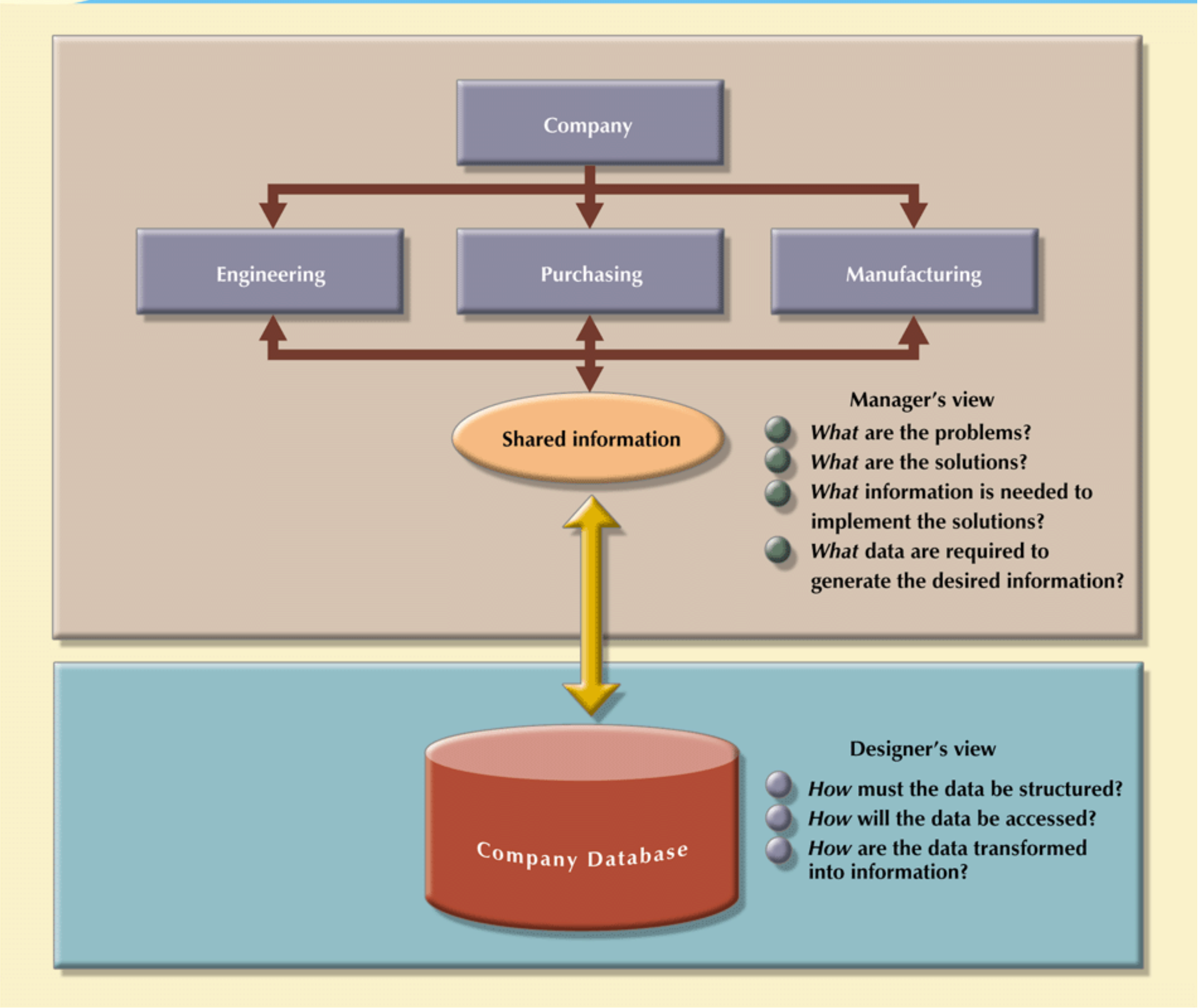
The Database Initial Study (cont'd.)

- Database system objectives must correspond to those envisioned by end users
  - What is proposed system's initial objective?
  - Will system interface with other systems in the company?
  - Will system share data with other systems or users?
- **Scope**: extent of design according to operational requirements
- **Boundaries**: limits external to system

Database Design

- Necessary to concentrate on data characteristics required to build database model
- Two views of data within system:
  - – Business view
    - Data as information source
  - – Designer’s view
    - Data structure, access, and activities required to transform data into information

FIGURE 9.5 Two views of data: business manager and database designer



Company

Engineering

Purchasing

Manufacturing

Shared information

Manager's view

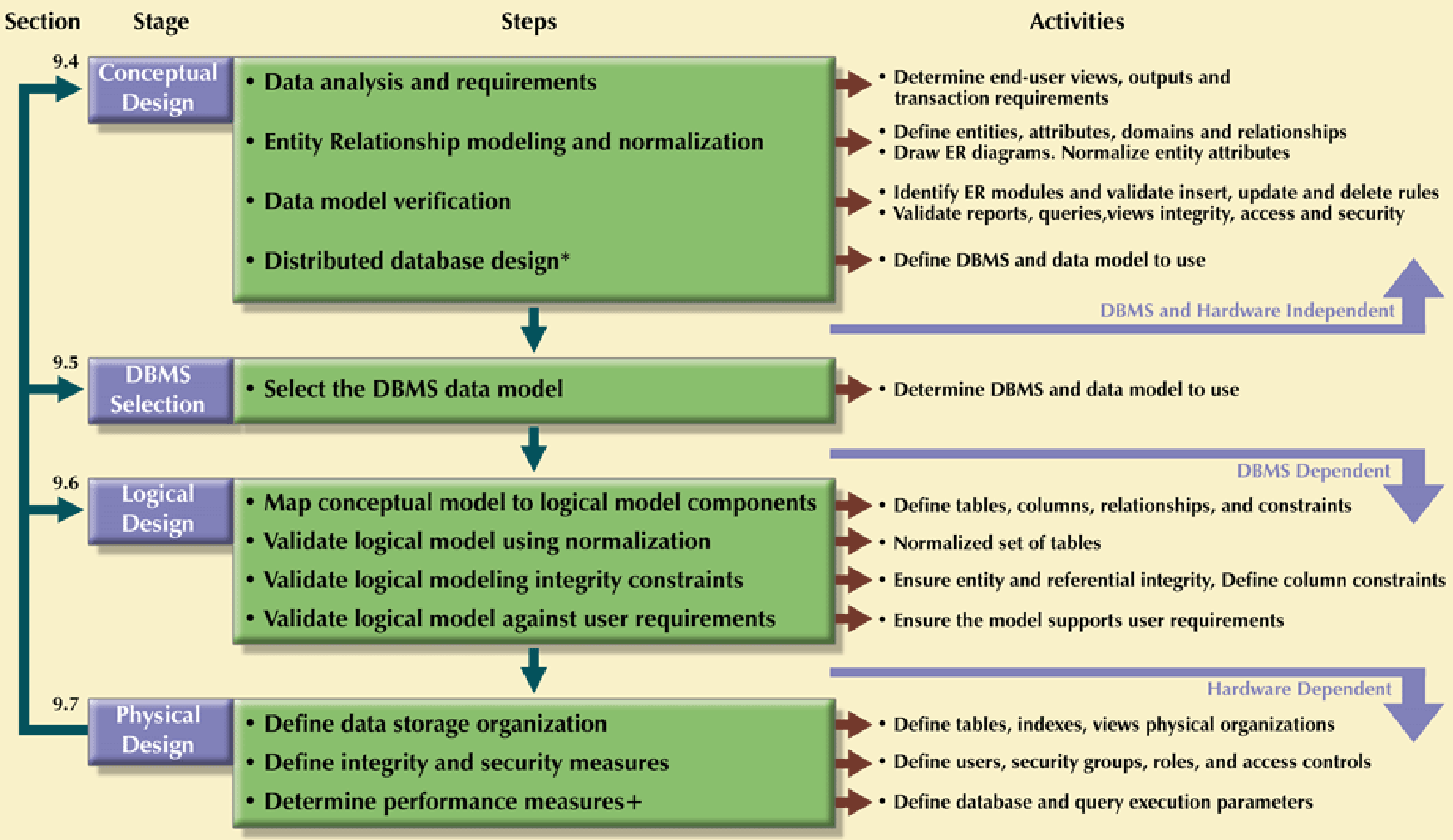
- *What* are the problems?
- *What* are the solutions?
- *What* information is needed to implement the solutions?
- *What* data are required to generate the desired information?

Company Database

Designer's view

- *How* must the data be structured?
- *How* will the data be accessed?
- *How* are the data transformed into information?

FIGURE 9.6 Database design process



Section	Stage	Steps	Activities
9.4	Conceptual Design	• Data analysis and requirements	• Determine end-user views, outputs and transaction requirements
		• Entity Relationship modeling and normalization	• Define entities, attributes, domains and relationships • Draw ER diagrams. Normalize entity attributes
		• Data model verification	• Identify ER modules and validate insert, update and delete rules • Validate reports, queries,views integrity, access and security
		• Distributed database design*	• Define DBMS and data model to use
9.5	DBMS Selection	• Select the DBMS data model	• Determine DBMS and data model to use
9.6	Logical Design	• Map conceptual model to logical model components	• Define tables, columns, relationships, and constraints
		• Validate logical model using normalization	• Normalized set of tables
		• Validate logical modeling integrity constraints	• Ensure entity and referential integrity, Define column constraints
		• Validate logical model against user requirements	• Ensure the model supports user requirements
9.7	Physical Design	• Define data storage organization	• Define tables, indexes, views physical organizations
		• Define integrity and security measures	• Define users, security groups, roles, and access controls
		• Determine performance measures+	• Define database and query execution parameters

* See Chapter 12, Distributed Database Management Systems
+ See Chapter 11, Database Performance Tuning and Query Optimization

Implementation and Loading

- Actually implement all design specifications from previous phase:
  - Install the DBMS
    - Virtualization: creates logical representations of computing resources independent of physical resources
  - Create the Database
  - Load or Convert the Data

Testing and Evaluation

- Occurs in parallel with applications programming
- Database tools used to prototype applications
- If implementation fails to meet some of system’s evaluation criteria:
  - Fine-tune specific system and DBMS configuration parameters
  - Modify physical or logical design
  - Upgrade software and/or hardware platform

Testing and Evaluation (cont'd.)

- Integrity
  - Enforced via proper use of primary, foreign key rules
- Backup and Recovery
  - **Full backup**
  - **Differential backup**
  - **Transaction log backup**

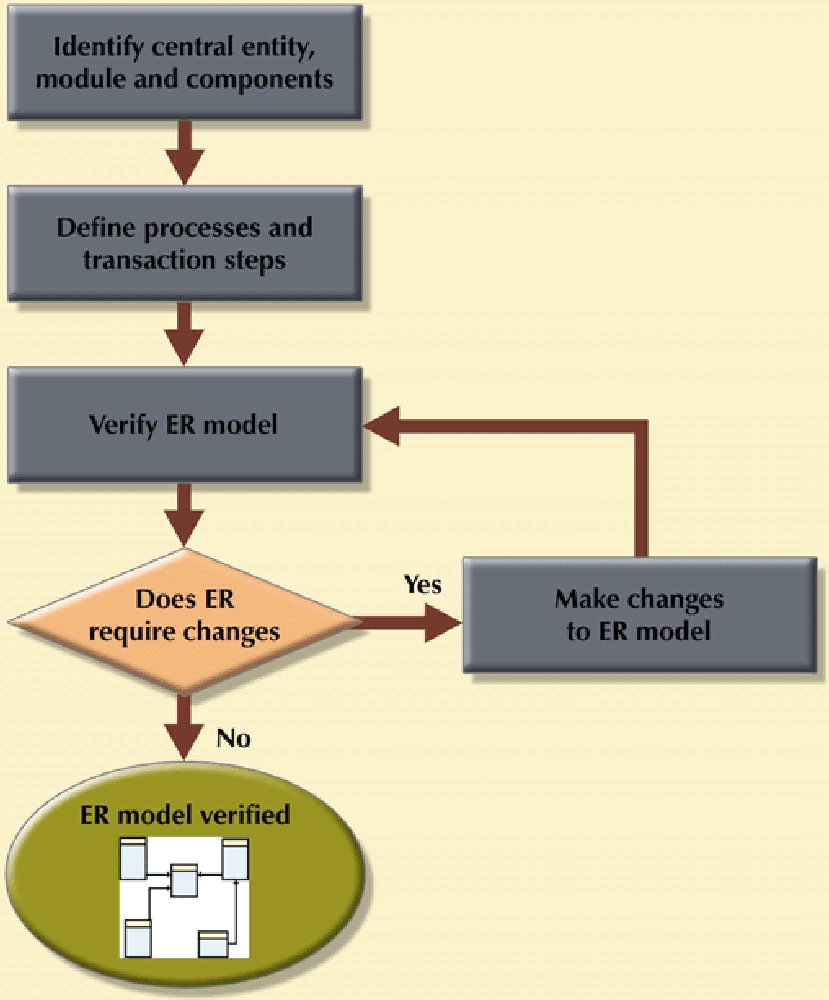
Operation

- Once database has passed evaluation stage, it is considered operational
- Beginning of operational phase starts process of system evolution
- Problems not foreseen during testing surface
- Solutions may include:
  - Load-balancing software to distribute transactions among multiple computers
  - Increasing available cache

Maintenance and Evolution

- Required periodic maintenance:
  - Preventive maintenance (backup)
  - Corrective maintenance (recovery)
  - Adaptive maintenance
  - Assignment of access permissions and their maintenance for new and old users
  - Generation of database access statistics
  - Periodic security audits
  - Periodic system-usage summaries

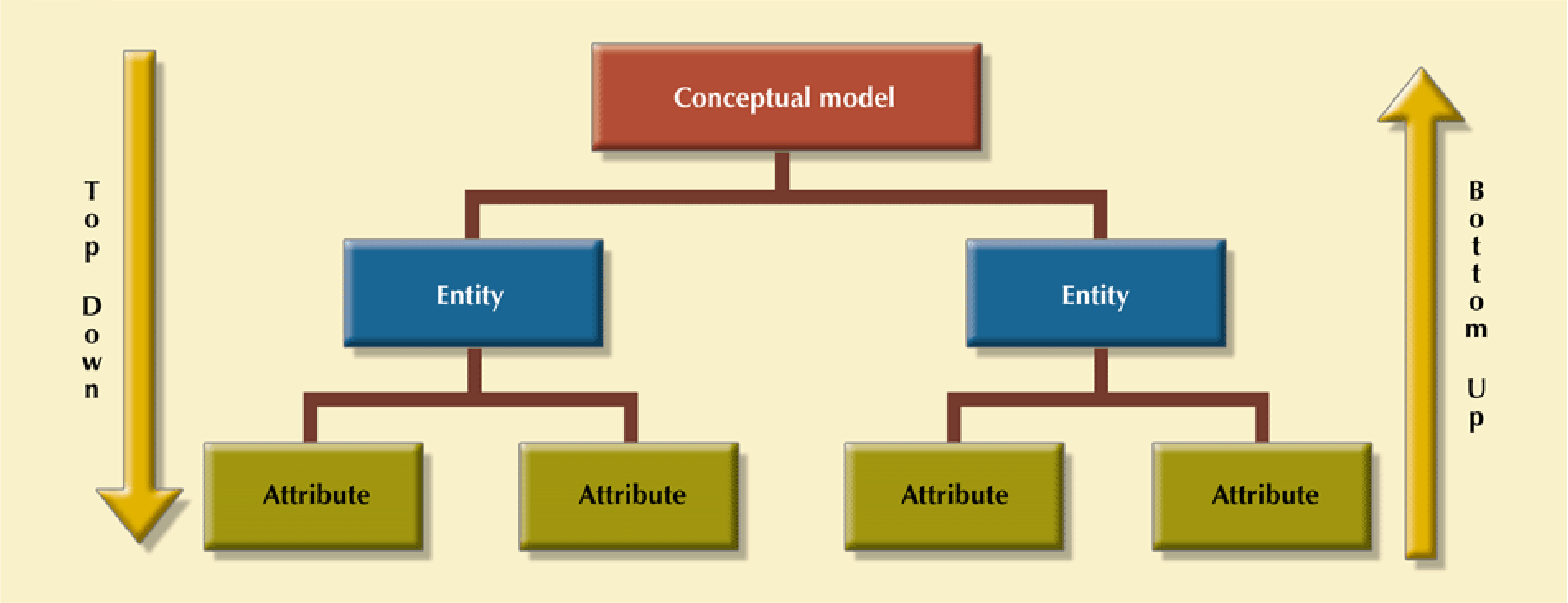
FIGURE 9.12 Iterative ER model verification process


Identify central entity, module and components
Define processes and transaction steps
Verify ER model
Does ER require changes
Yes
Make changes to ER model
No
ER model verified

Database Design Strategies

- **Top-down design**
  - – Identifies data sets
  - – Defines data elements for each of those sets
    - Definition of different entity types
    - Definition of each entity’s attributes
- **Bottom-up design**
  - – Identifies data elements (items)
  - – Groups them together in data sets

FIGURE 9.14 Top-down vs. bottom-up design sequencing



Centralized vs. Decentralized Design

- **Centralized design**
  - When data component is composed of small number of objects and procedures
  - Typical of small systems
- **Decentralized design**
  - Data component has large number of entities
  - Complex relations on which complex operations are performed
  - Problem is spread across several operational sites

FIGURE 9.15

Centralized design

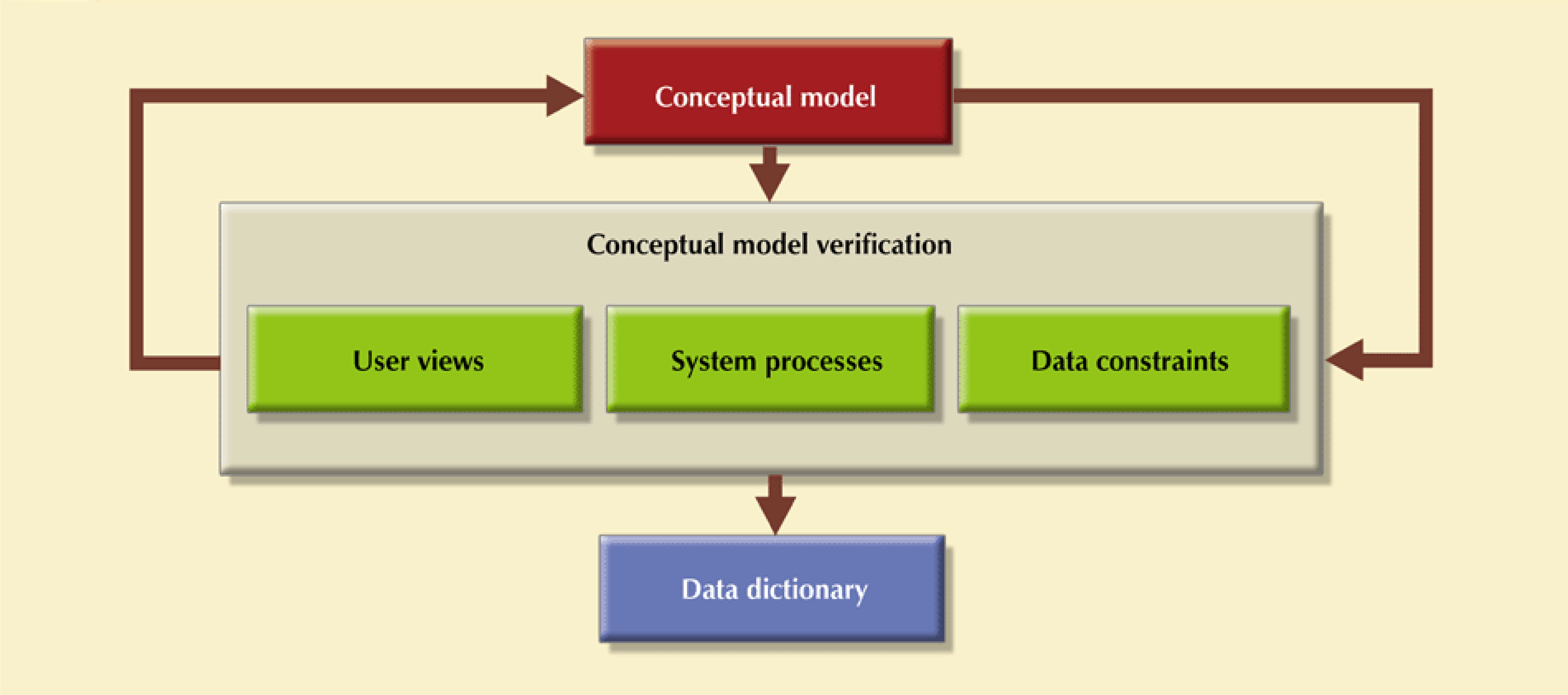
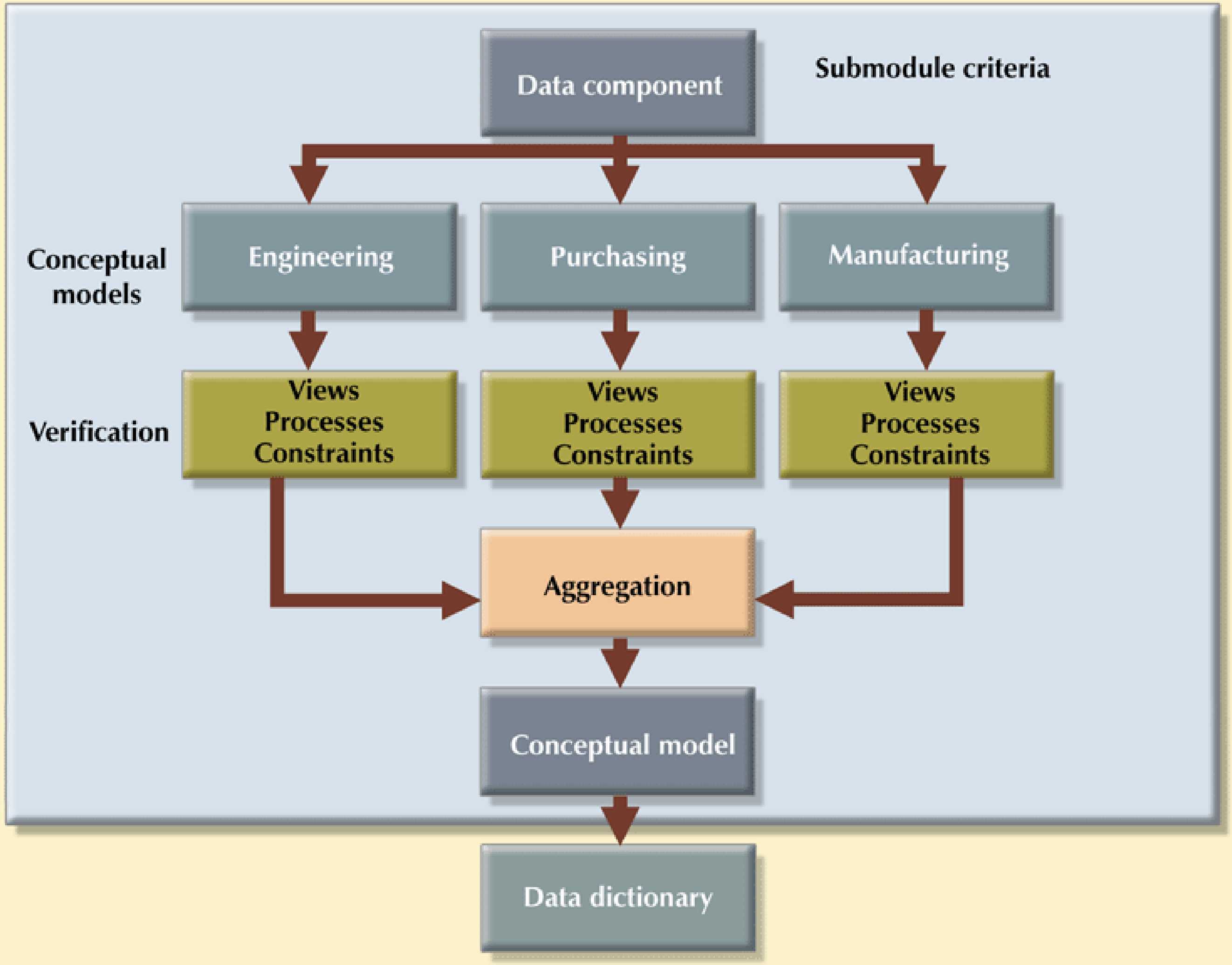


FIGURE 9.16

Decentralized design


Data component
Submodule criteria
Conceptual models
Engineering
Purchasing
Manufacturing
Verification
Views
Processes
Constraints
Views
Processes
Constraints
Views
Processes
Constraints
Aggregation
Conceptual model
Data dictionary