UNIT III: Software Design

CHAPTER 12 DESIGN CONCEPTS

CHAPTER 13 ARCHITECTURAL DESIGN

CHAPTER 14 COMPONENT-LEVEL DESIGN

CHAPTER 12 DESIGN CONCEPTS

- 12.1 Design within the Context of Software Engineering
- 12.2 The Design Process
- 12.3 Design Concepts
- 12.4 The Design Model

CHAPTER 13 ARCHITECTURAL DESIGN

- 13.1 Software Architecture
- 13.3 Architectural Styles

CHAPTER 14 COMPONENT-LEVEL DESIGN

- 14.1 What Is a Component? (Component Views)
- 14.2 Designing Class-Based Components
- 14.3 Conducting Component-Level Design

UNIT III: Software Design

CHAPTER 12 DESIGN CONCEPTS

CHAPTER 13 ARCHITECTURAL DESIGN

CHAPTER 14 COMPONENT-LEVEL DESIGN

CHAPTER 13 ARCHITECTURAL DESIGN

13.1 Software Architecture

13.3 Architectural Styles

13.1 Software Architecture: Definition

- Software architecture of a program or computing system is the structure or structures of the system which comprise
  - Software components
  - Externally visible properties of those components
  - Relationships among the components
- Focus is on the software component
  - A program module
  - An object-oriented class
  - A database
  - Middleware

13.1 Software Architecture: Enables Software Engineer

SA is a representation that enables a software engineer to:

(1) analyze the effectiveness of the design in meeting its stated requirements,

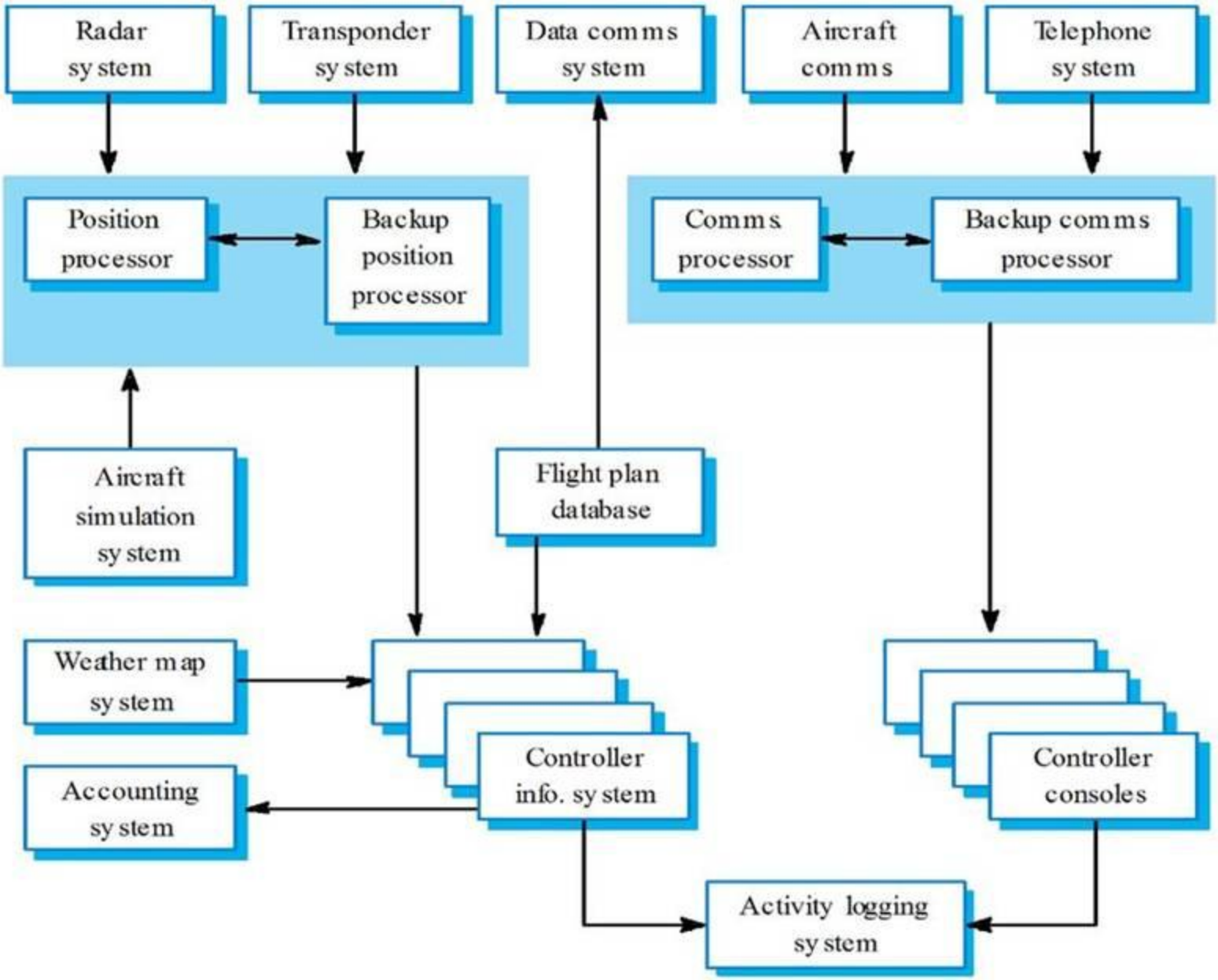
(2) consider architectural alternatives at a stage when making design changes is still relatively easy, and

(3) reduce the risks associated with the construction of the software.

13.1 Software Architecture: Importance (Why)

- SA is an enabler for communication between all stakeholders.
- SA highlights early design decisions that will have a profound impact on all software engineering work that follows and, as important, on the ultimate success of the system as an operational entity.
- SA constitutes a relatively small, intellectually graspable model of how the system is structured and how its components work together.

Air traffic control system architecture



13.1 Software Architecture: Architectural Descriptions

- The IEEE Computer Society has proposed IEEE-Std-1471-2000, *Recommended Practice for Architectural Description of Software-Intensive System*, [IEE00]
  - to establish a conceptual framework and vocabulary for use during the design of software architecture,
  - to provide detailed guidelines for representing an architectural description, and
  - to encourage sound architectural design practices.

- The IEEE Standard defines an *architectural description* (AD) as a "a collection of products to document an architecture."
  - The description itself is represented using multiple views, where each *view* is "a representation of a whole system from the perspective of a related set of [stakeholder] concerns."

13.3 Architectural Styles

- **Genre** implies a specific category within the overall software domain. Within each category, you encounter a number of subcategories.
- **Style:** Variation from the original (from a genre) that carries on the interpretation or any other.

Ex: Within the genre of *buildings*, you would encounter the following general styles: houses, condos, apartment buildings, office buildings, industrial building, warehouses, and so on.

13.3 Architectural Styles

Home is the Architectural Genre

American Home is the Architectural Style

13.3 Architectural Styles

Temple is the Architectural Genre

Hindu Temple is the Architectural Style

13.3 Architectural Styles : Styles

Each style describes a system category that encompasses:

(1) **a set of components** (e.g., a database, computational modules)

(2) **a set of connectors** enable "communication, coordination and cooperation" among components,

(3) **constraints** that define how components can be integrated to form the system,

(4) **semantic models** that enable a designer to understand overall properties of a system by analyzing the known properties of its constituent parts.

13.3 Architectural Styles : Styles

A. Data-centered architectures

B. Data flow architectures

C. Call and return architectures

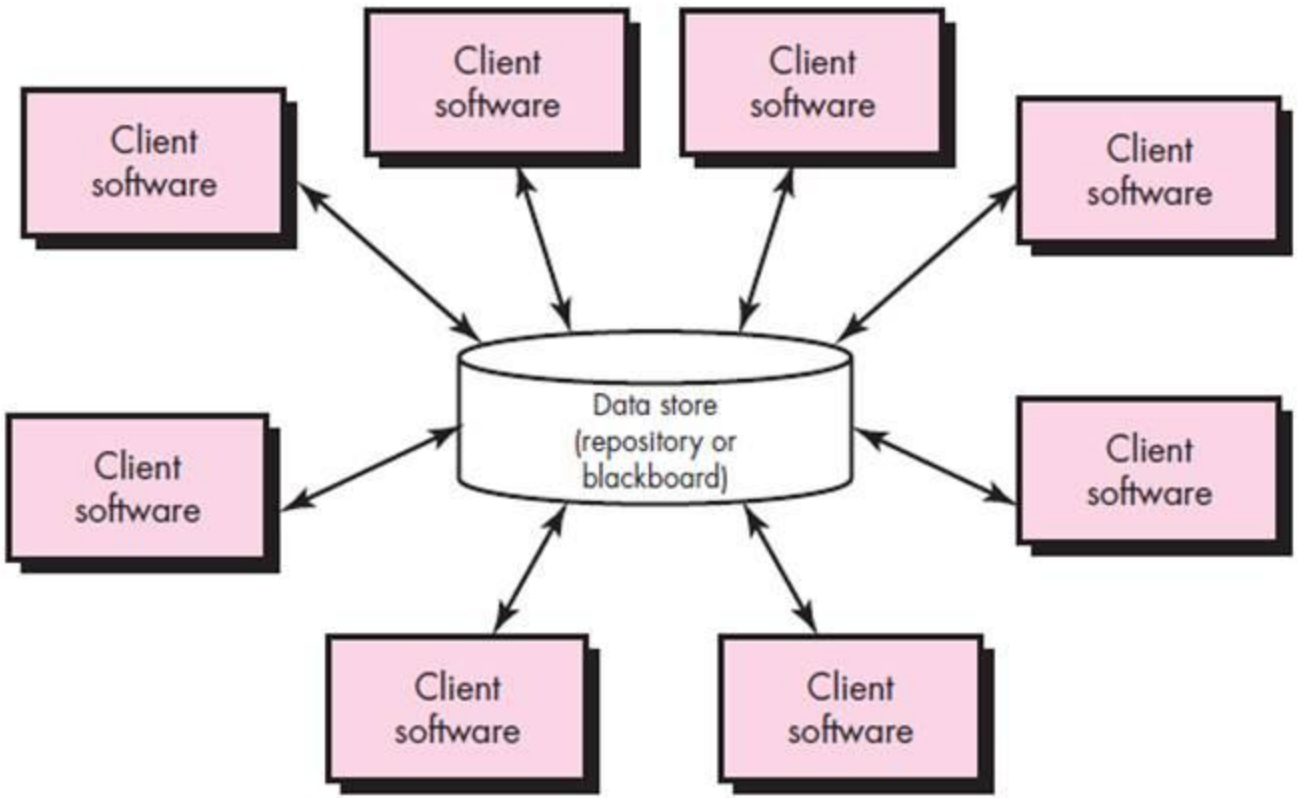
D. Object-oriented architectures

E. Layered architectures

13.3 Architectural Styles : Styles

A. Data-centered architectures

FIGURE 9.1
Data-centered architecture

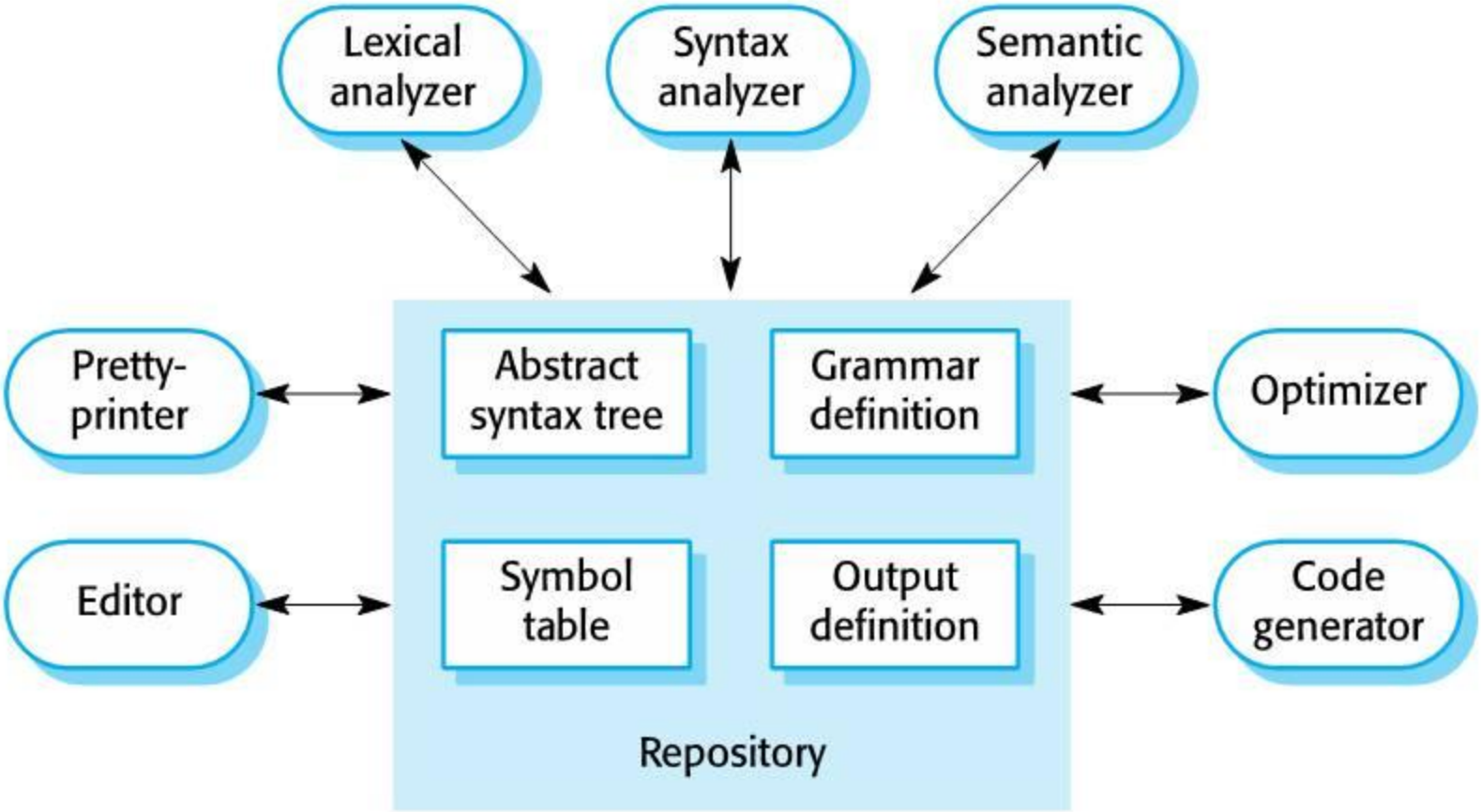


13.3 Architectural Styles : Styles

A. Data-centered architectures

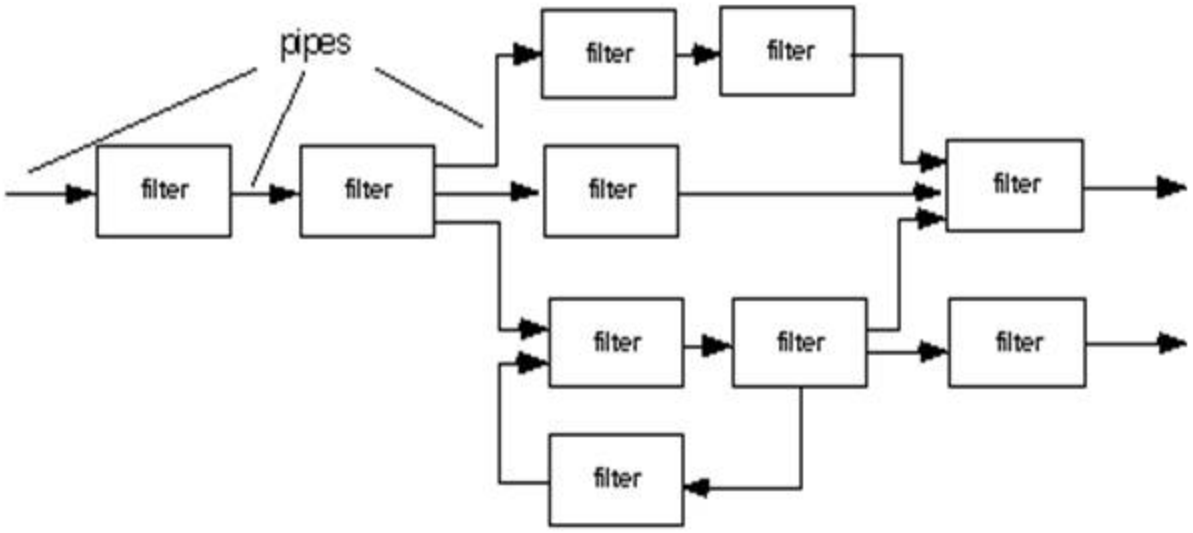
- Data store resides at the center of this architecture
- Accessed frequently by other components that update, add, delete, or otherwise modify data
- Client software accesses the data independent of any changes to the data or the actions of other client software.
- Promote *integrability, that is, existing* components can be changed and new client components added to the architecture without concern about other clients.

A Repository Architecture for a Language Processing System (Data-centered architectures)

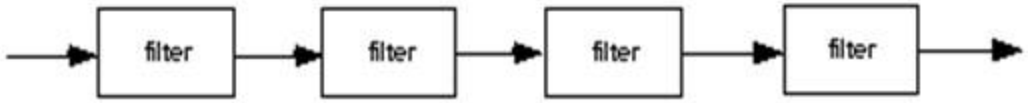


13.3 Architectural Styles : Styles

B. Data flow architectures



(a) pipes and filters



(b) batch sequential

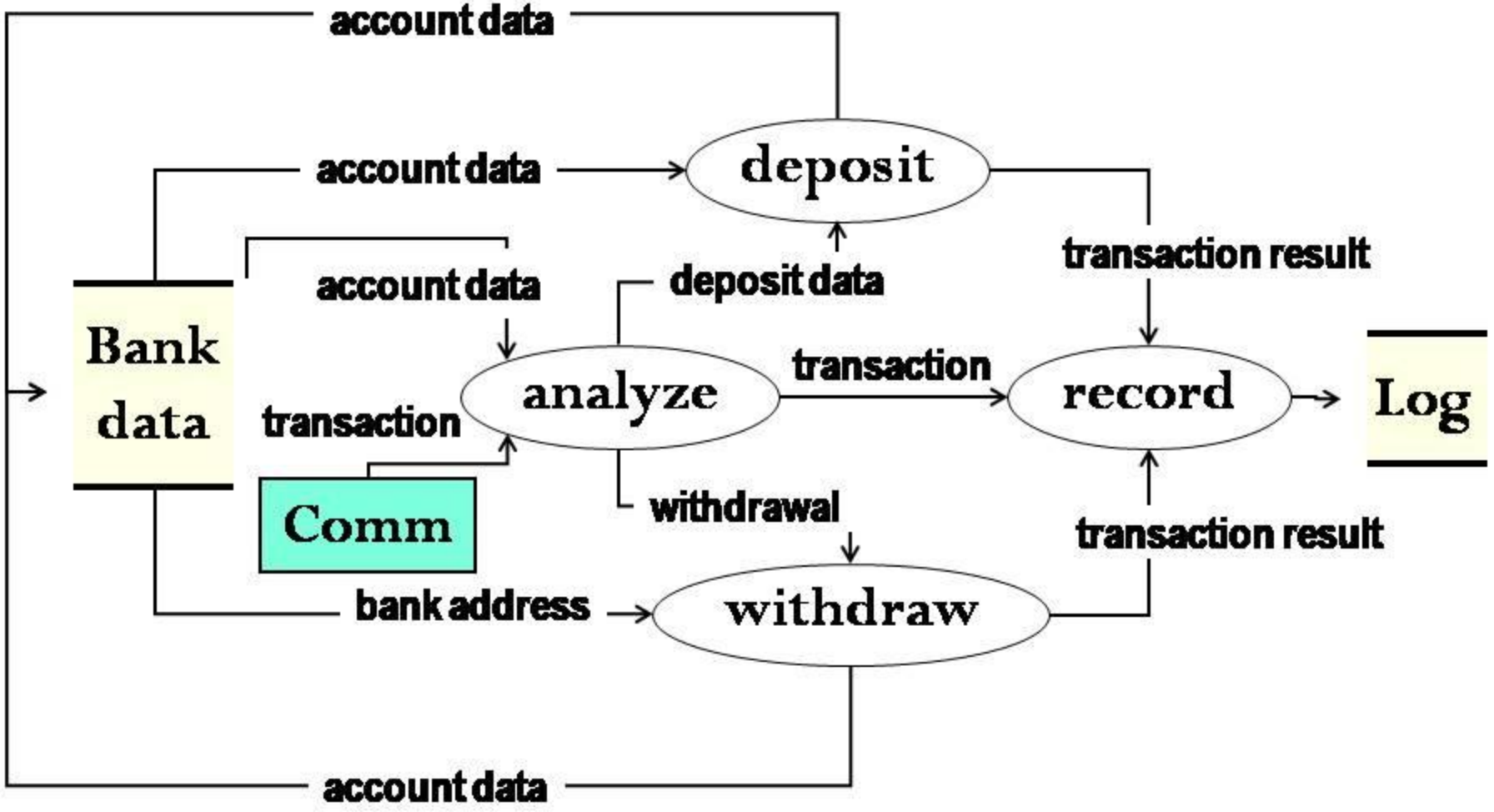
13.3 Architectural Styles : Styles

B. Data flow architectures

1. Applied when input data are to be transformed through a series of computational or manipulative components into output data.
2. Set of components: *filters, connected by pipes that transmit data from one component to the next.*
3. *Each* filter works independently of those components upstream and downstream
4. Designed to expect data input of a certain form, and produces data output of a specified form.
5. Filter does not require knowledge of workings of its neighboring filters.

Example of *Pipe & Filter Data Flow* Architecture

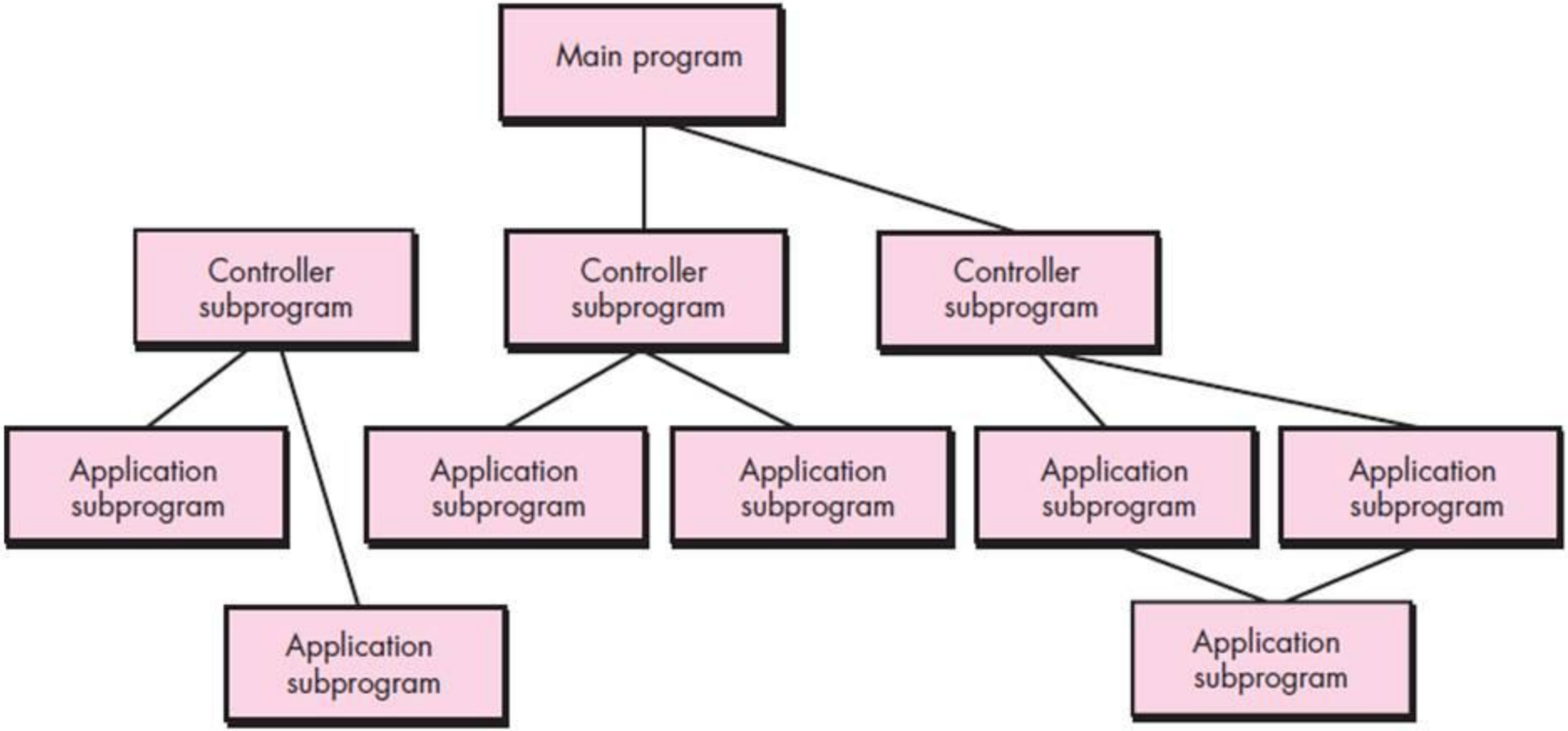
Requirement: Maintain financial transactions



13.3 Architectural Styles : Styles

C. Call and return architectures

FIGURE 9.3 Main program/subprogram architecture



13.3 Architectural Styles : Styles

C. Call and return architectures

Main program/subprogram architectures. This classic program structure:

decomposes function into a control hierarchy where a "main" program invokes a number of program components that in turn may invoke still other Components.

Remote procedure call architectures. The components of a main program/subprogram architecture are distributed across multiple computers on a network.

13.3 Architectural Styles : Styles

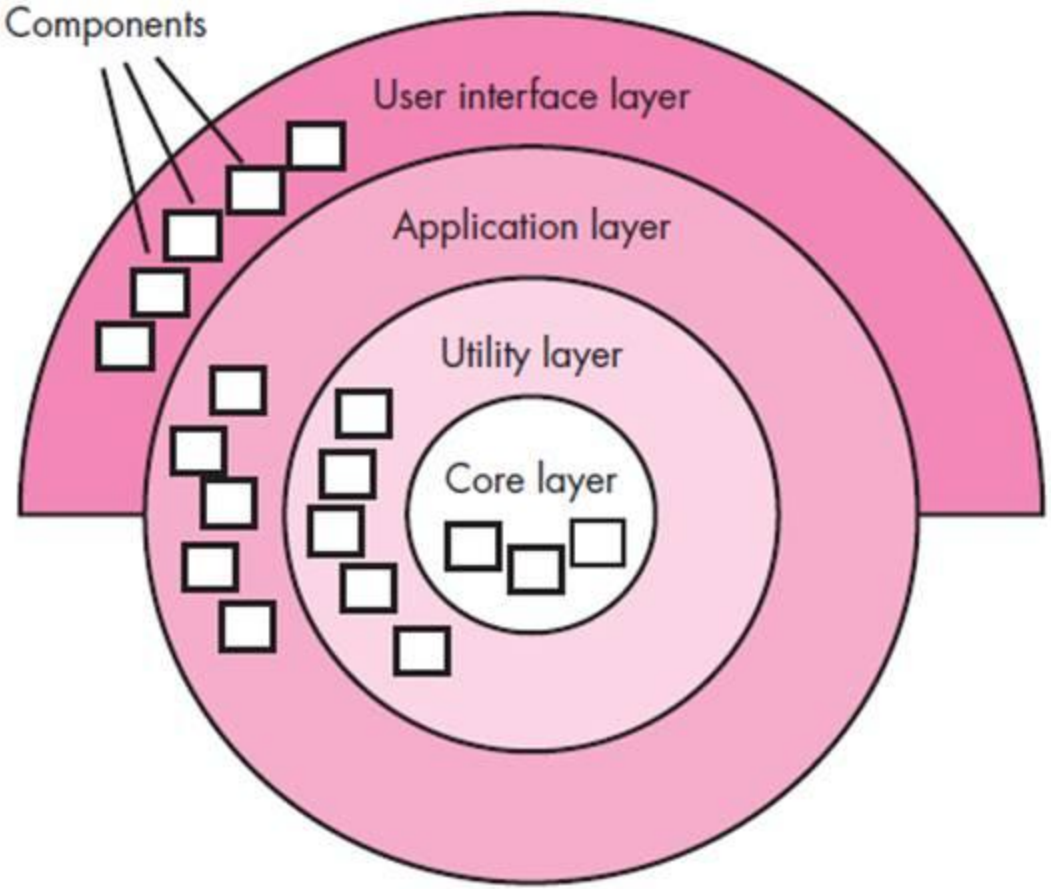
D. Object-oriented architectures

- Components of a system encapsulate data and the operations that must be applied to manipulate the data.
- Communication and coordination between components are accomplished via message passing.
- Keeps the internal data representation hidden and allows access to the object only through provided operations.
- Permits inheritance and polymorphism.

13.3 Architectural Styles : Styles

E. Layered architectures

FIGURE 9.4
Layered architecture

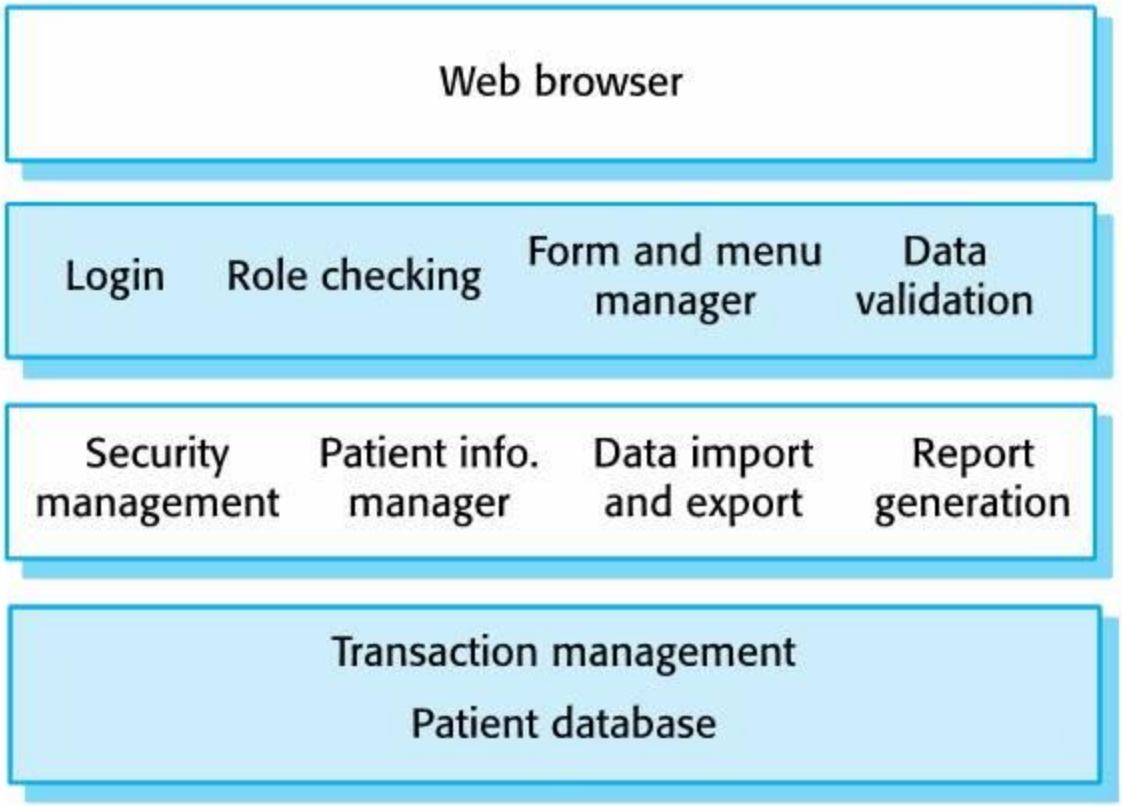


13.3 Architectural Styles : Styles

E. Layered architectures

- Number of different layers are defined, each accomplishing operations that progressively become closer to the machine instruction set.
- Outer layer, components service user interface operations.
- Inner layer, components perform operating system interfacing.
- Intermediate layers provide utility services and application software functions.

The Architecture of the Mentcare System



- UI has been implemented using a web browser.
- 2nd layer: provides user interface functionality. It includes components to allow users to log in to the system and checking components.
- 3rd layer implements the functionality of the system and provides components that implements system security, patent information creation and update,……
- Lowest layer, which is built using the commercial database management system, provides transaction management and persistent data storage.

13.3 Architectural Styles : Patterns

13.3 Architectural Styles : Organization and Refinement