AARHUS
UNIVERSITY
DEPARTMENT OF ENGINEERING

Architecture & Design of Embedded Real-Time Systems (TI-AREM)

Architectural Design Patterns 1.

- Subsystem and Component Architecture Patterns

(BPD. Chapter 4.1-4.7, p. 141-184)

Version: 9-2-2015

Agenda

1. SOLID – 5 OO design principles
2. Layered Pattern
3. Five-Layer Pattern
4. Microkernel Architectural Pattern
5. Channel Architecture Pattern
6. Recursive Containment Pattern
7. Hierarchical Control Pattern
8. Virtual Machine Pattern

Architecture Patterns and guidelines

- A ***pattern*** is a general solution to an often occurring design problem
- An architectural patterns describes a solution on the **highest system level**
- **SOLID – 5 OO design principles**
  - **Can be used as guidelines also on the architecture level**

1. SOLID – design principles

- **S:** Single responsibility principle (SRP)
- **O:** Open/Closed principle (OCP)
- **L:** Liskov's substitution principle (LSP)
- **I:** Interface segregation principle (ISP)
- **D:** Dependency inversion principle (DIP)

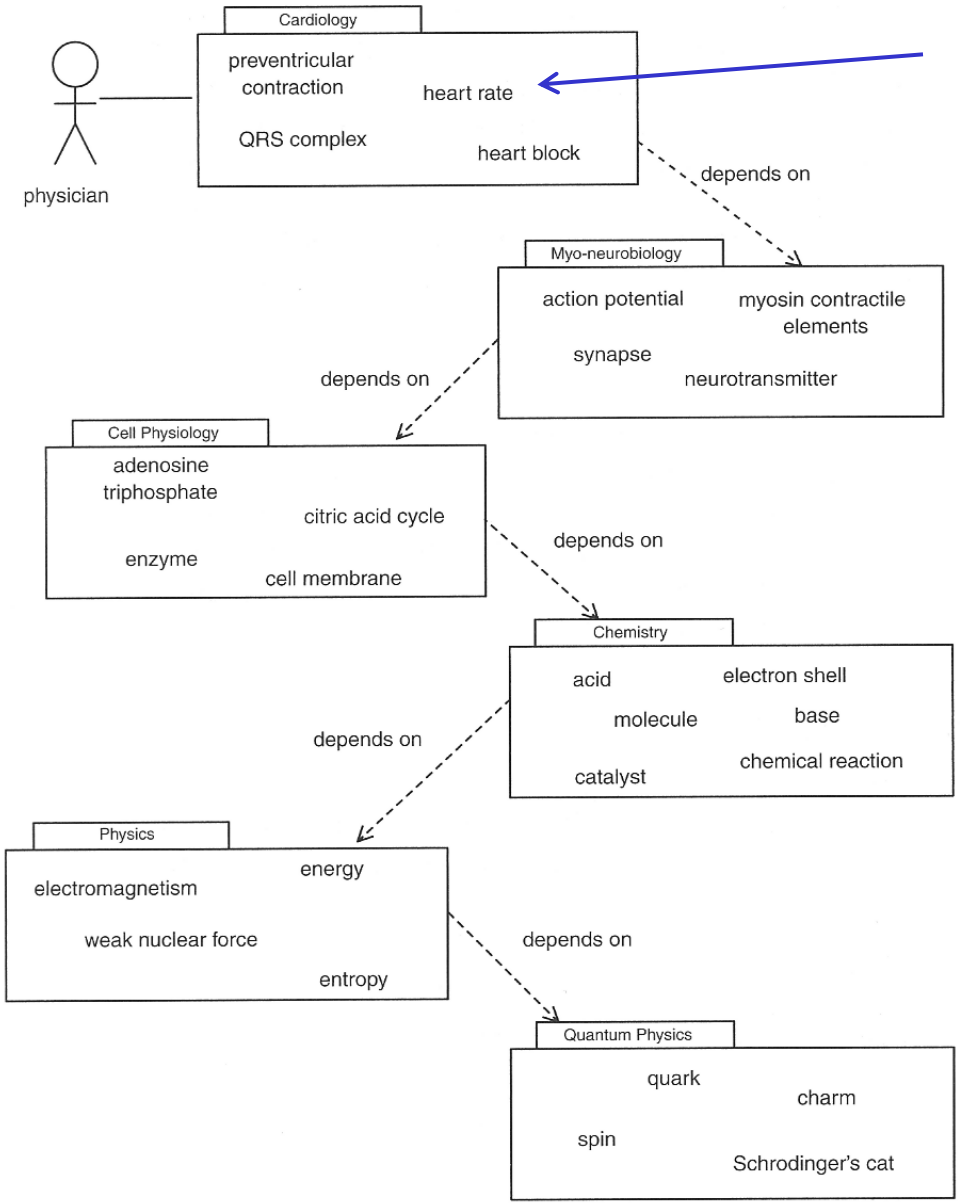
Ref. Five design principles defined by Robert C. Martin

2. Layered Pattern

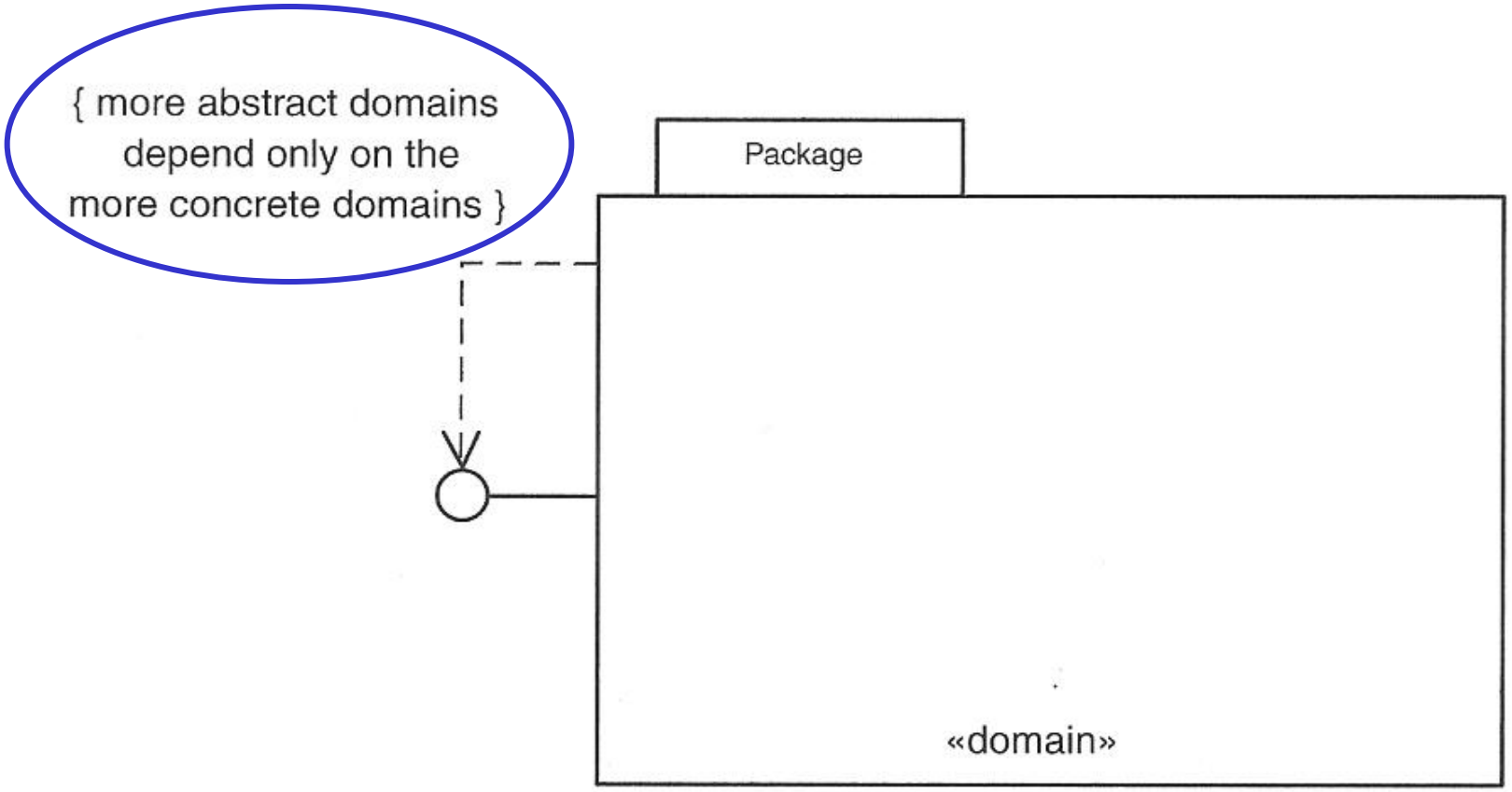
The Layered Pattern organizes domains into a hierarchical organization based on their level of abstraction

Cardiology Conceptual Hierarchy (4-1)

cardiology concepts



Layered Pattern Structure



Layered Pattern Consequences

- In a **Closed Layered** Architecture
  - classes in one layer can only invoke operations of classes in the same layer or **in the next layer down**
  - may result in loss of performance
  - have better encapsulation
- In an **Open Layered** Architecture
  - classes in one layer may invoke operations of classes in the same layer or **any layer below it**
  - have better performance

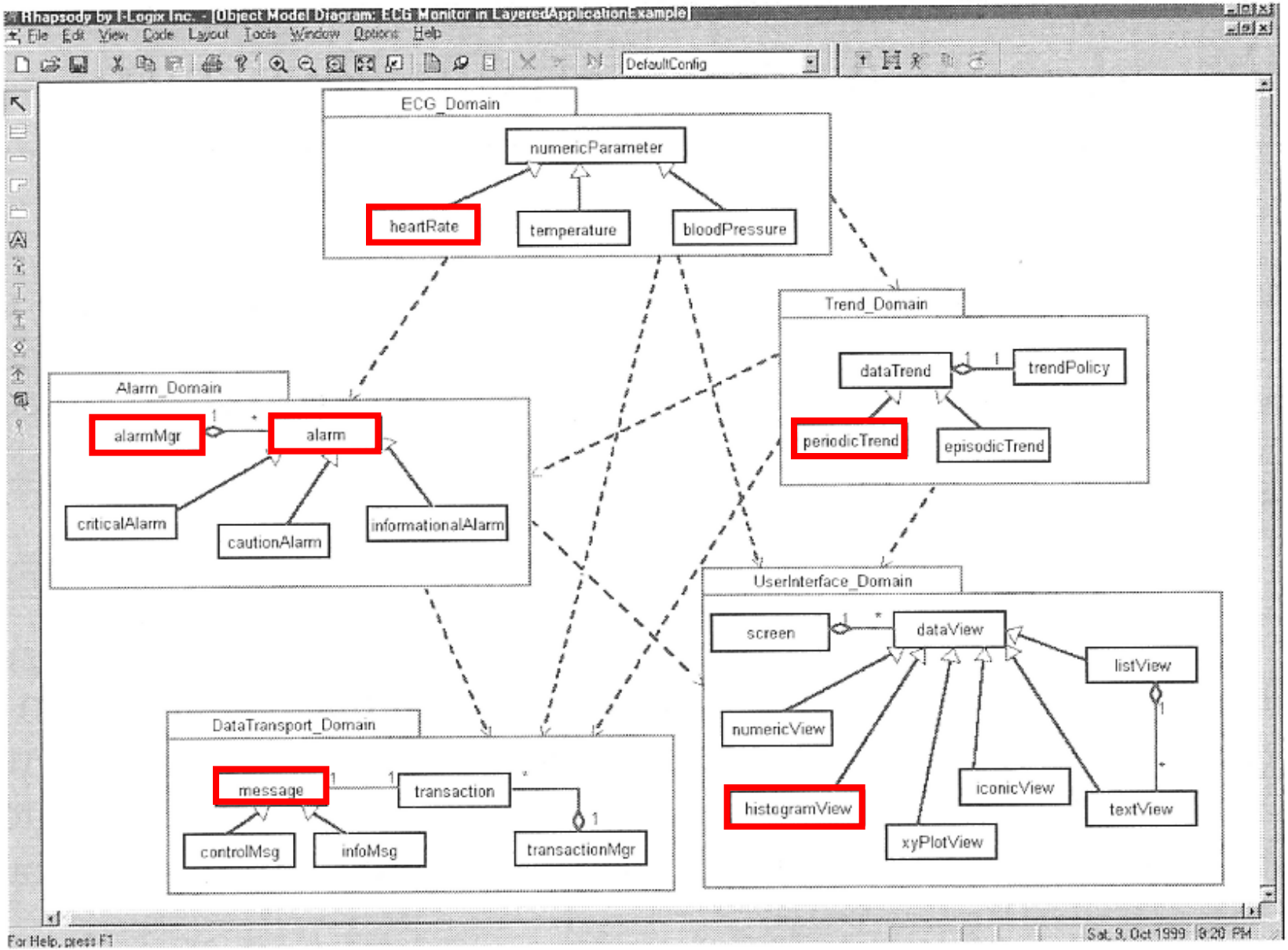
Implementation Strategies

- Important relationship between layers:
  - one-way client-server associations among layers
  - it is **crucial** that these associations between layers are one way, allowing messages to be sent to the lower layers

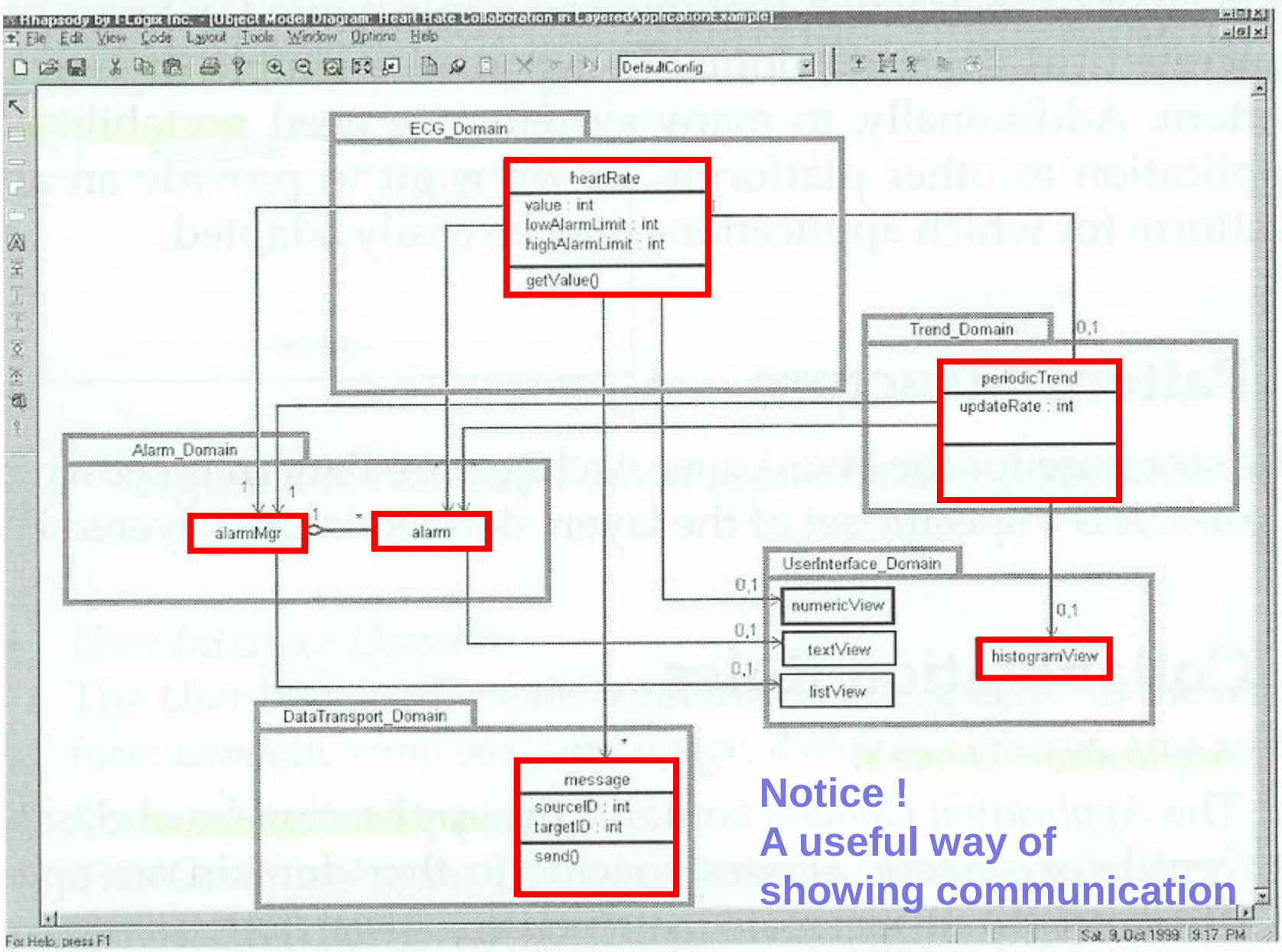
Related Patterns

- The Five-Layered Architecture Pattern
  - is a particular adaptation of this pattern common to real-time and embedded systems
- The Recursive Containment Pattern
  - is to physical architecture what the layered pattern is to logical architecture

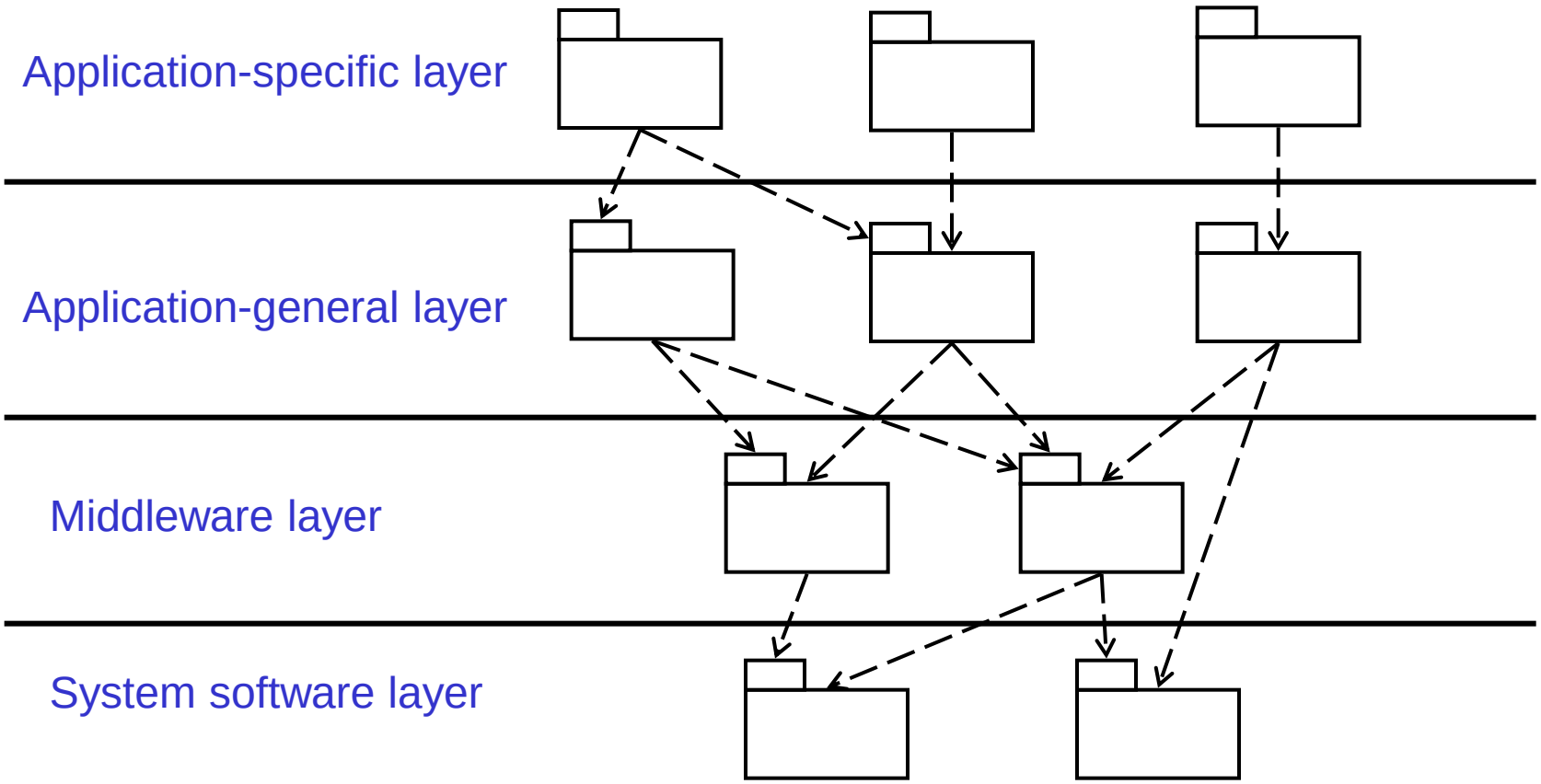
ECG Domain Model (4-3)



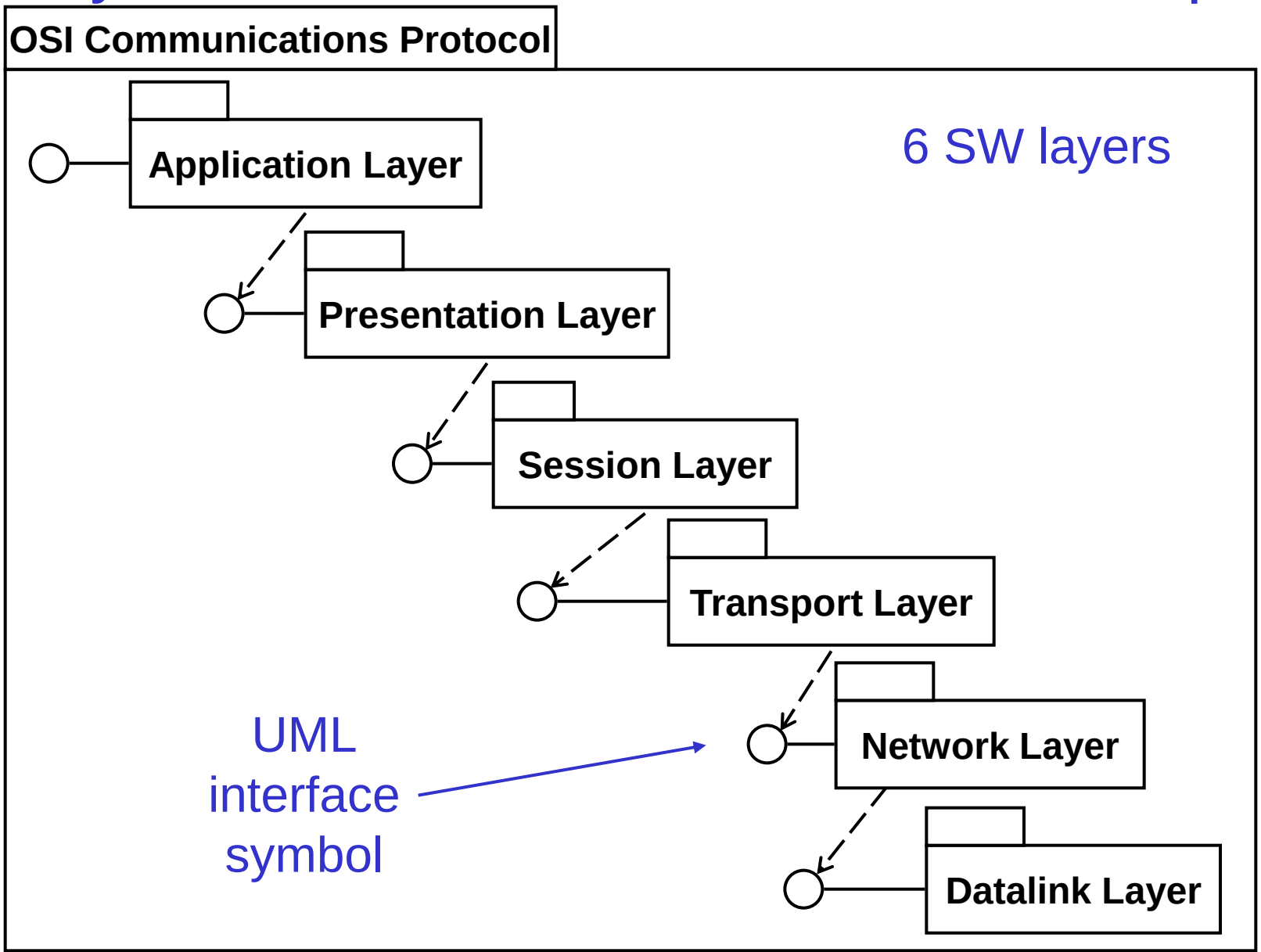
ECG Collaboration / Communication (4-4)

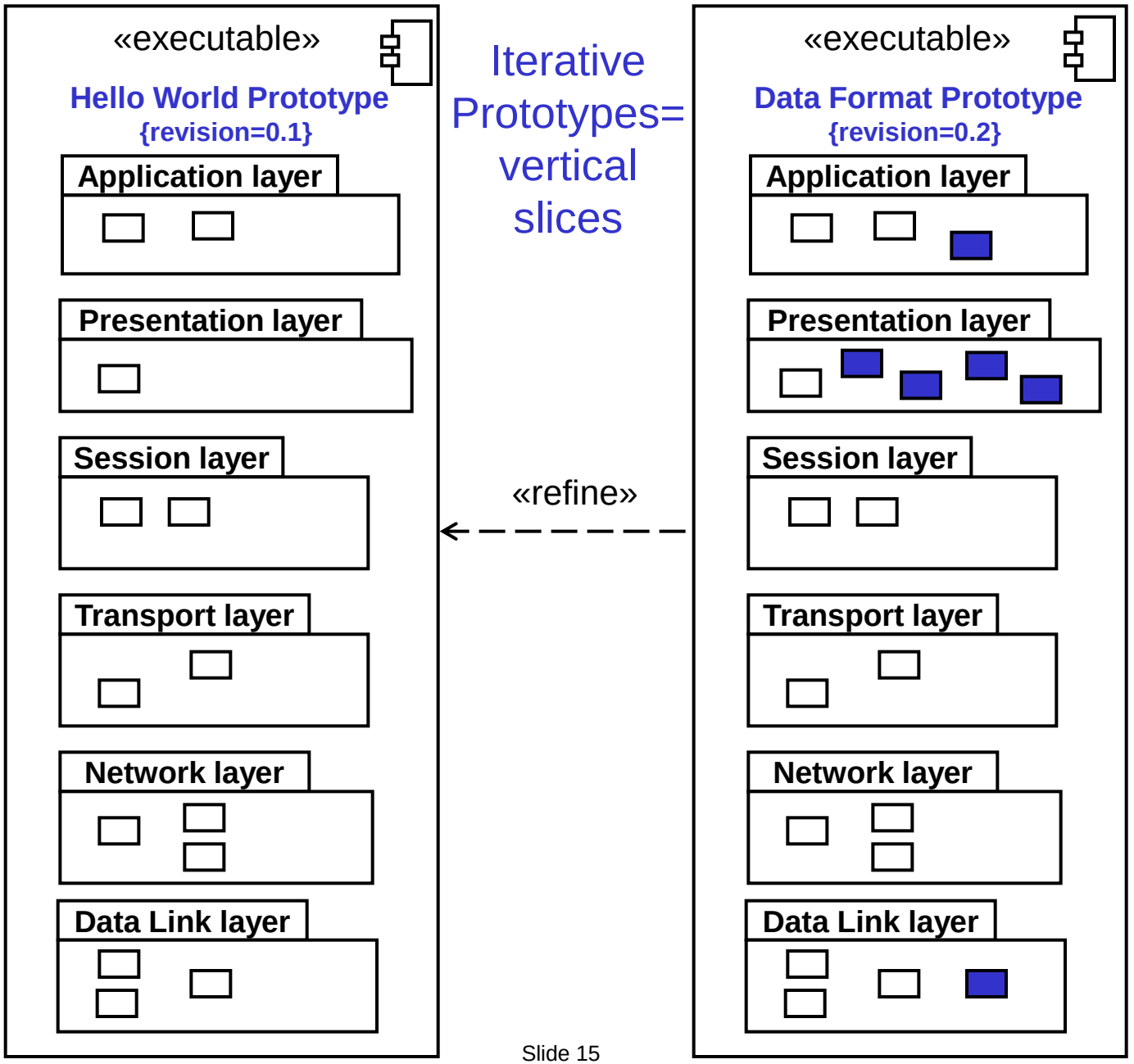


Typical Layered Subsystems



Layered architecture - OSI Model Example

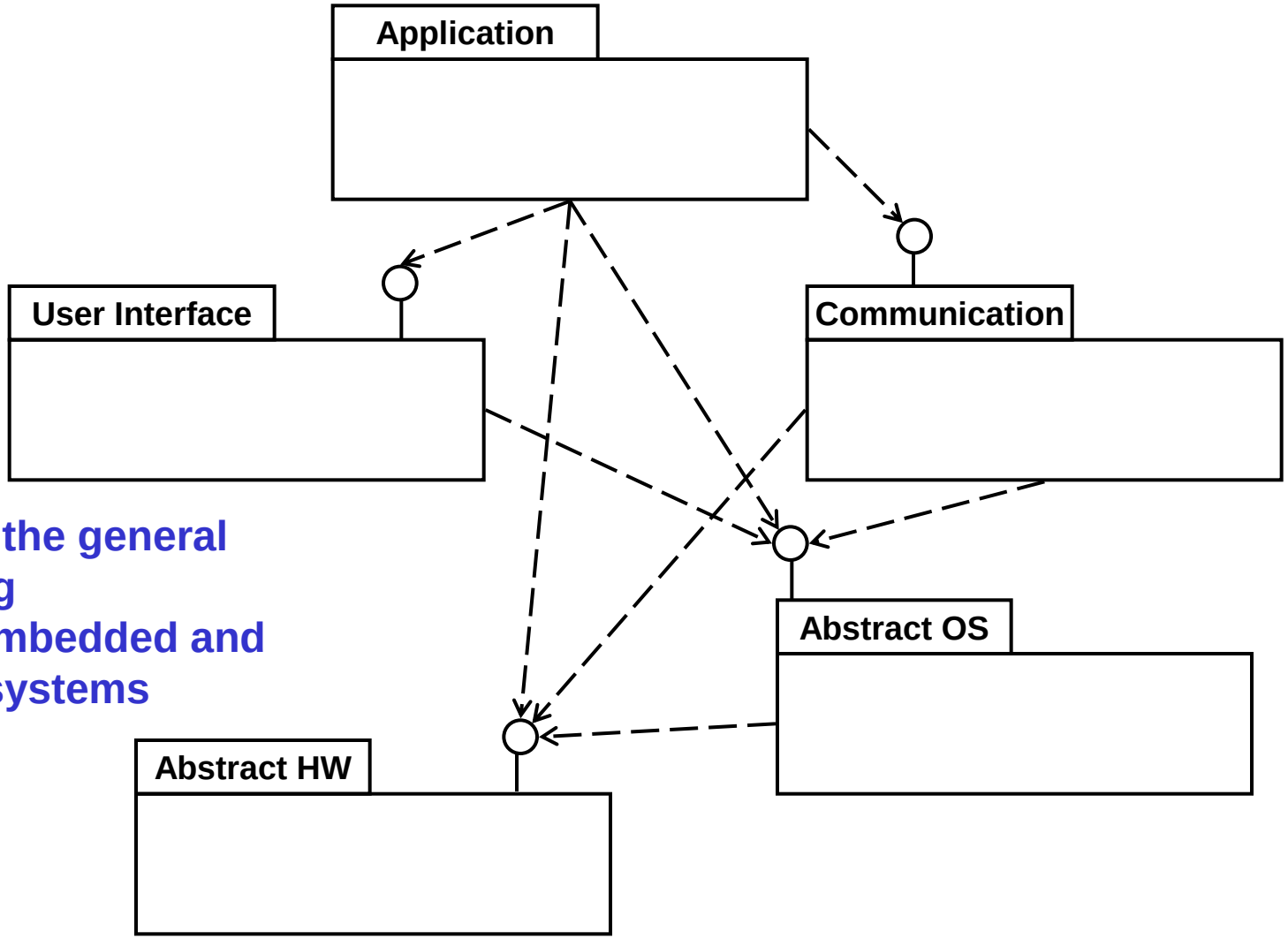



«executable»
Hello World Prototype
{revision=0.1}
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data Link layer
Iterative Prototypes= vertical slices
«refine»
«executable»
Data Format Prototype
{revision=0.2}
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data Link layer

3. Five-Layer Architecture Pattern

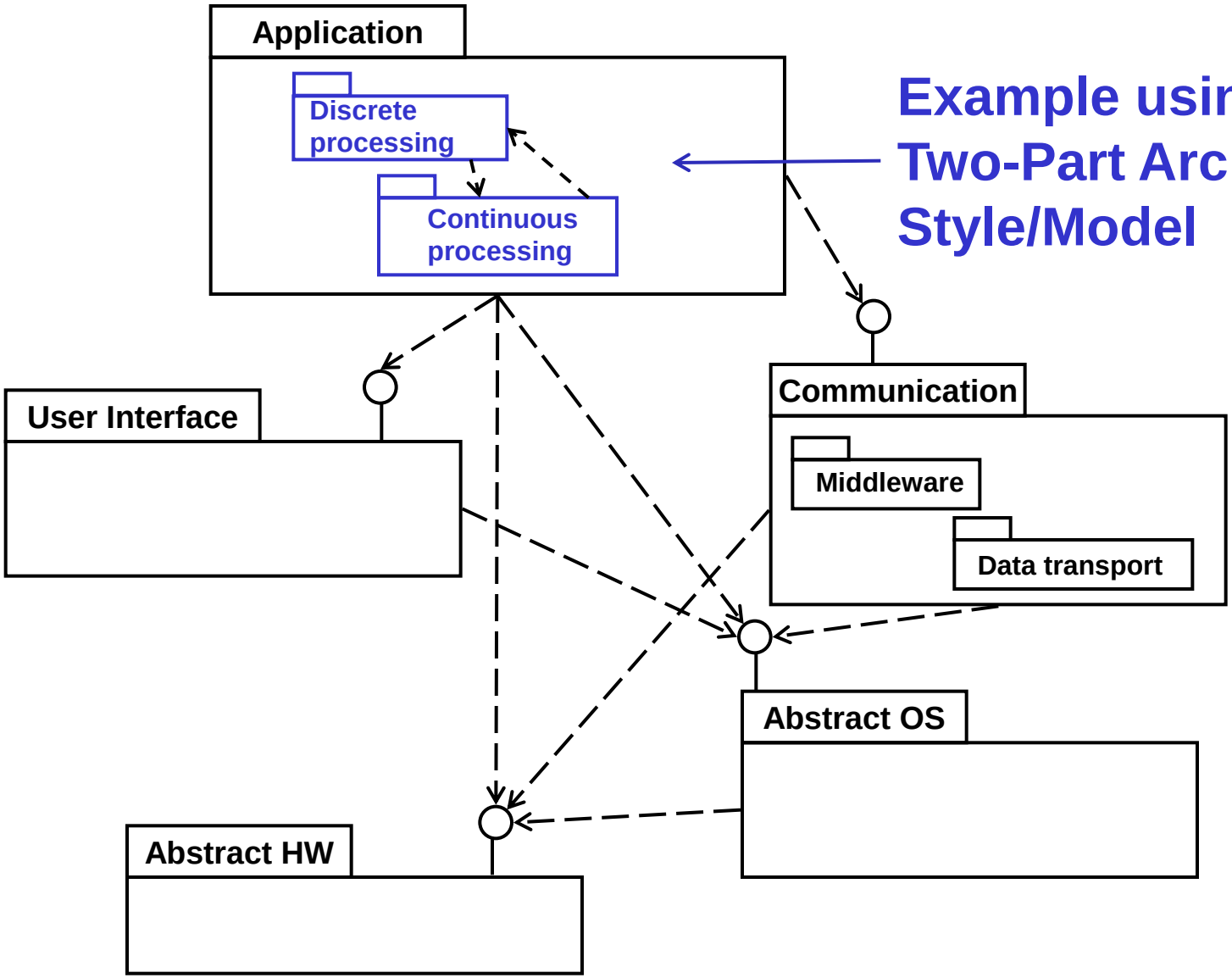
The Five-Layer Architecture Pattern is a specific architecture useful for the general structuring of many embedded and real-time systems

Five-Layer Architecture Pattern Structure

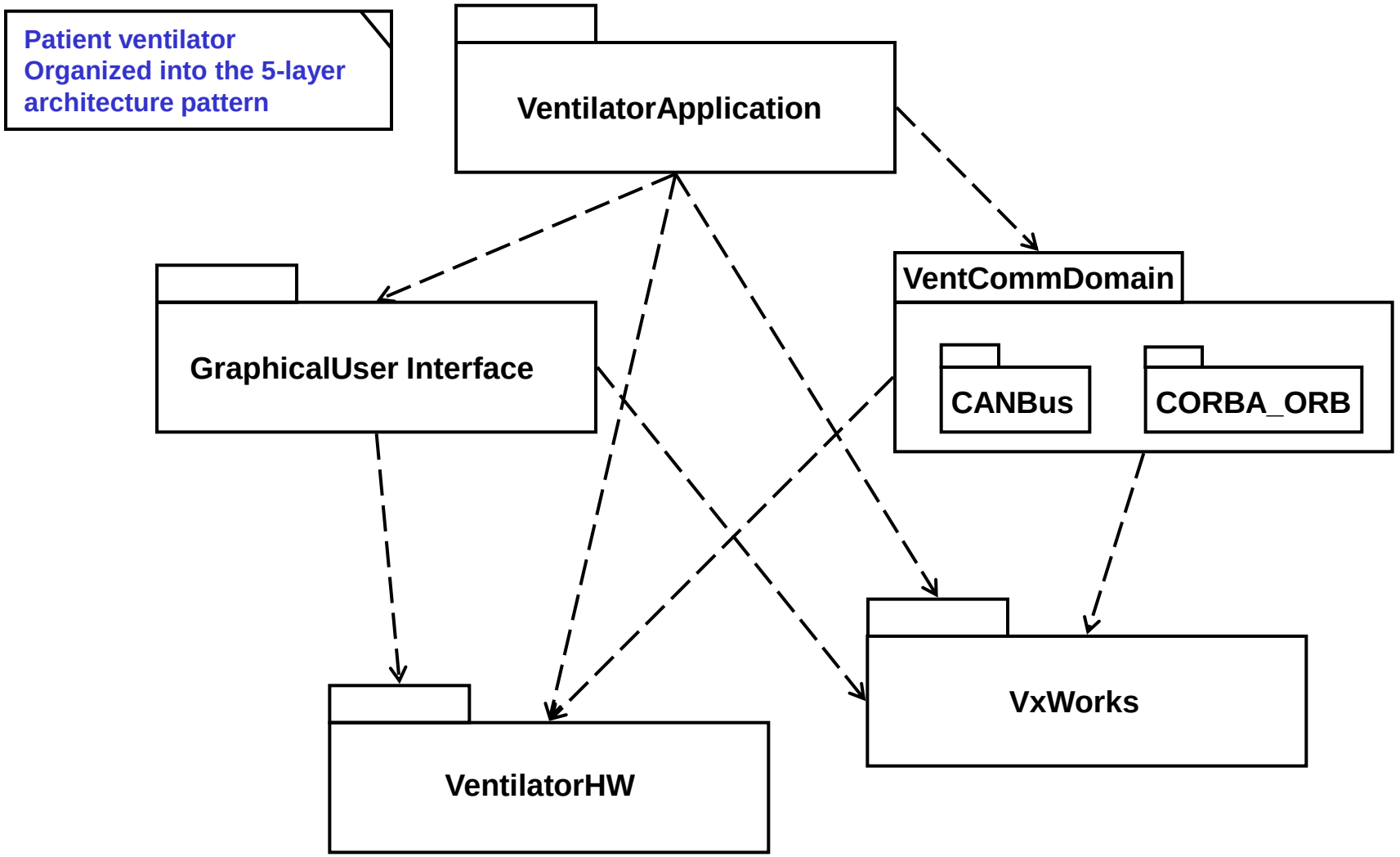


Useful for the general structuring of many embedded and real-time systems

Five-Layer Architecture Pattern Example



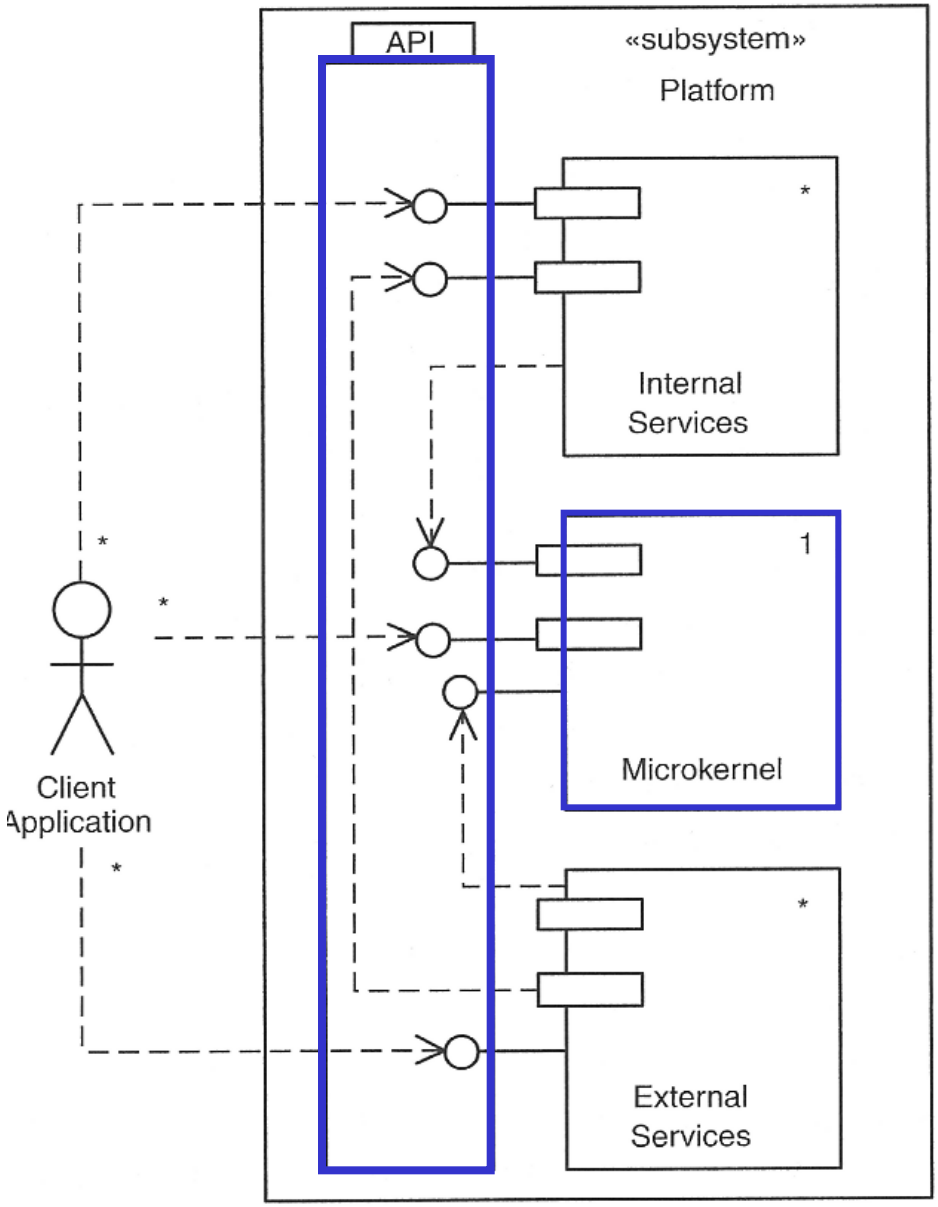
Ventilator Example Domains (4-6)



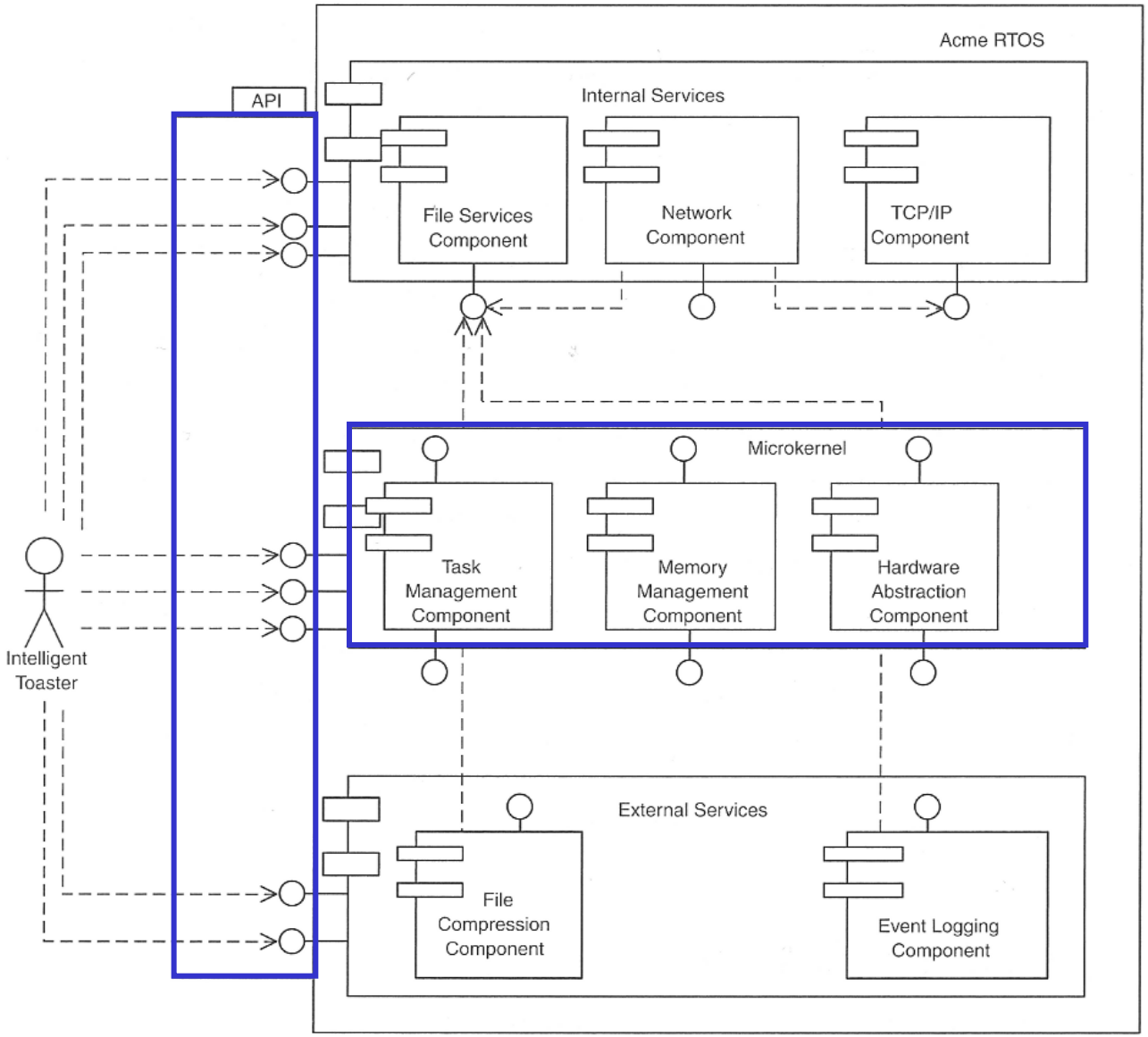
4. Microkernel Architectural Pattern

The Microkernel Architecture Pattern is a useful pattern when a system consist of a core set of services that may be augmented at build-time with a variety of additional services

Microkernel Architectural Pattern Structure



Example: nanoOS Model (4-8)

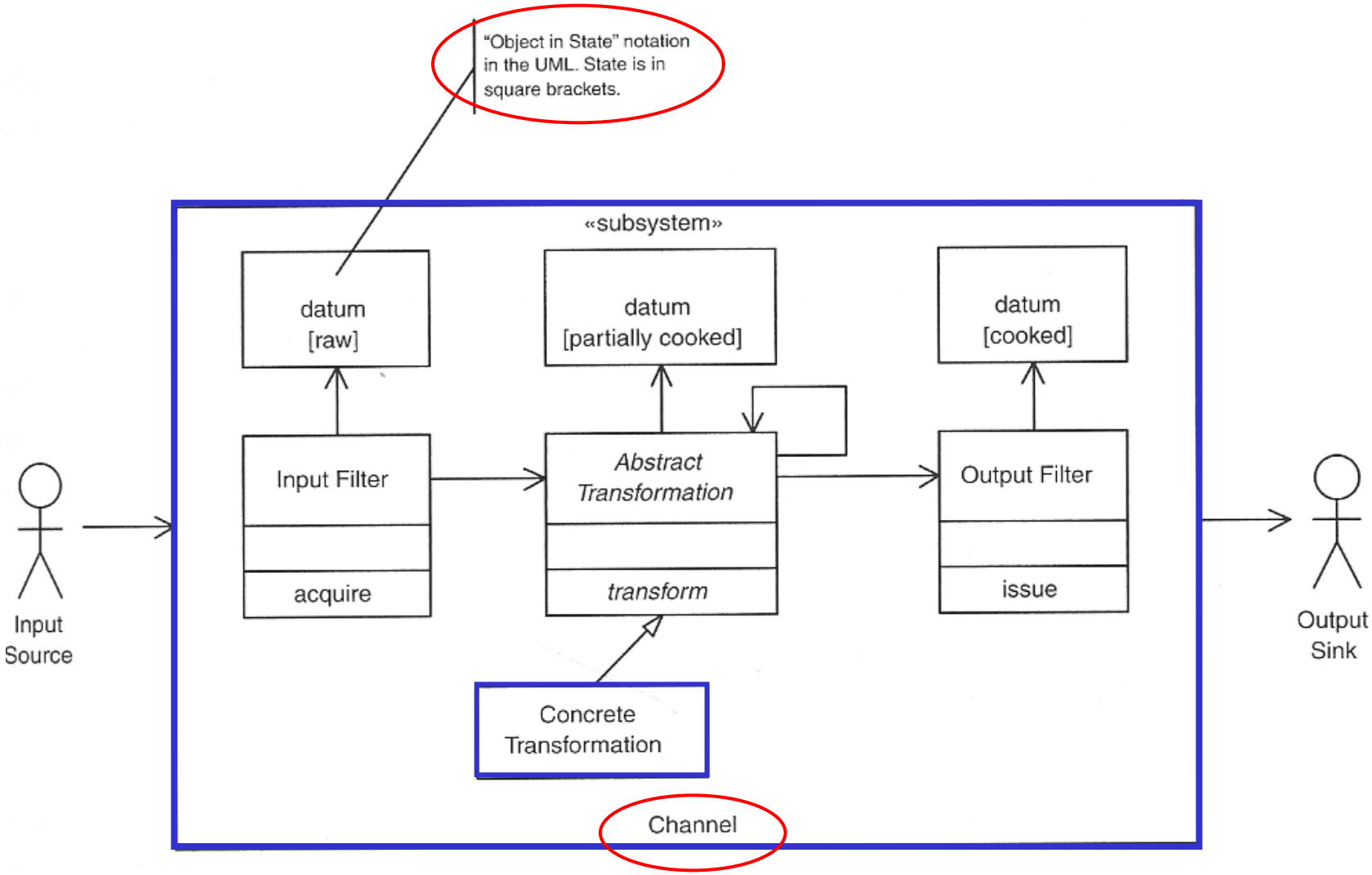


5. Channel Architecture Pattern

The Channel Architecture Pattern is useful in two different situations:

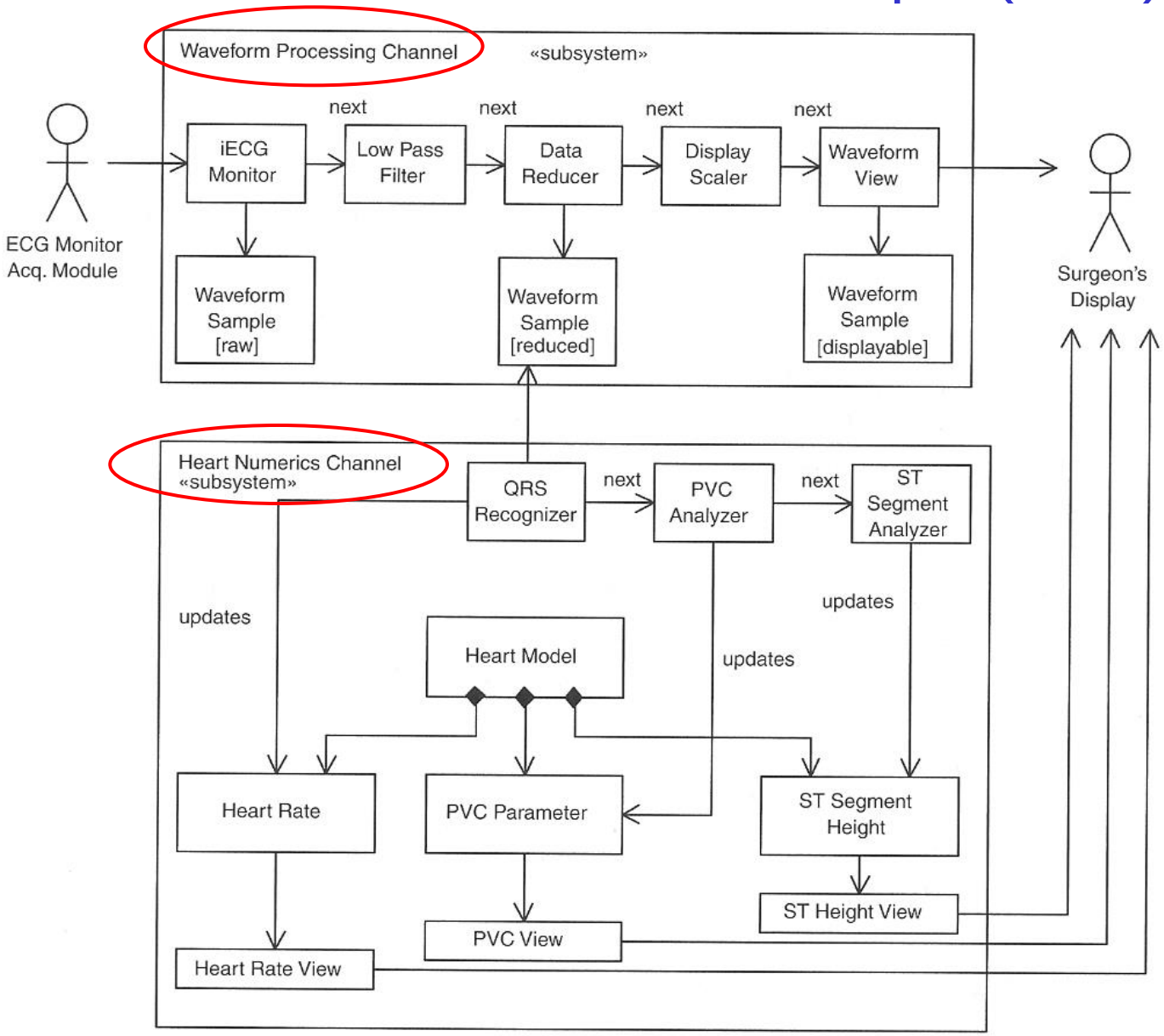
1. **when data within a stream is sequentially transformed in a series of steps**
2. **at large scale the pattern offers architectural redundancy for high-reliability and safety-critical applications**

Channel Architecture Pattern Structure



a kind of "Pipe and Filter Pattern"

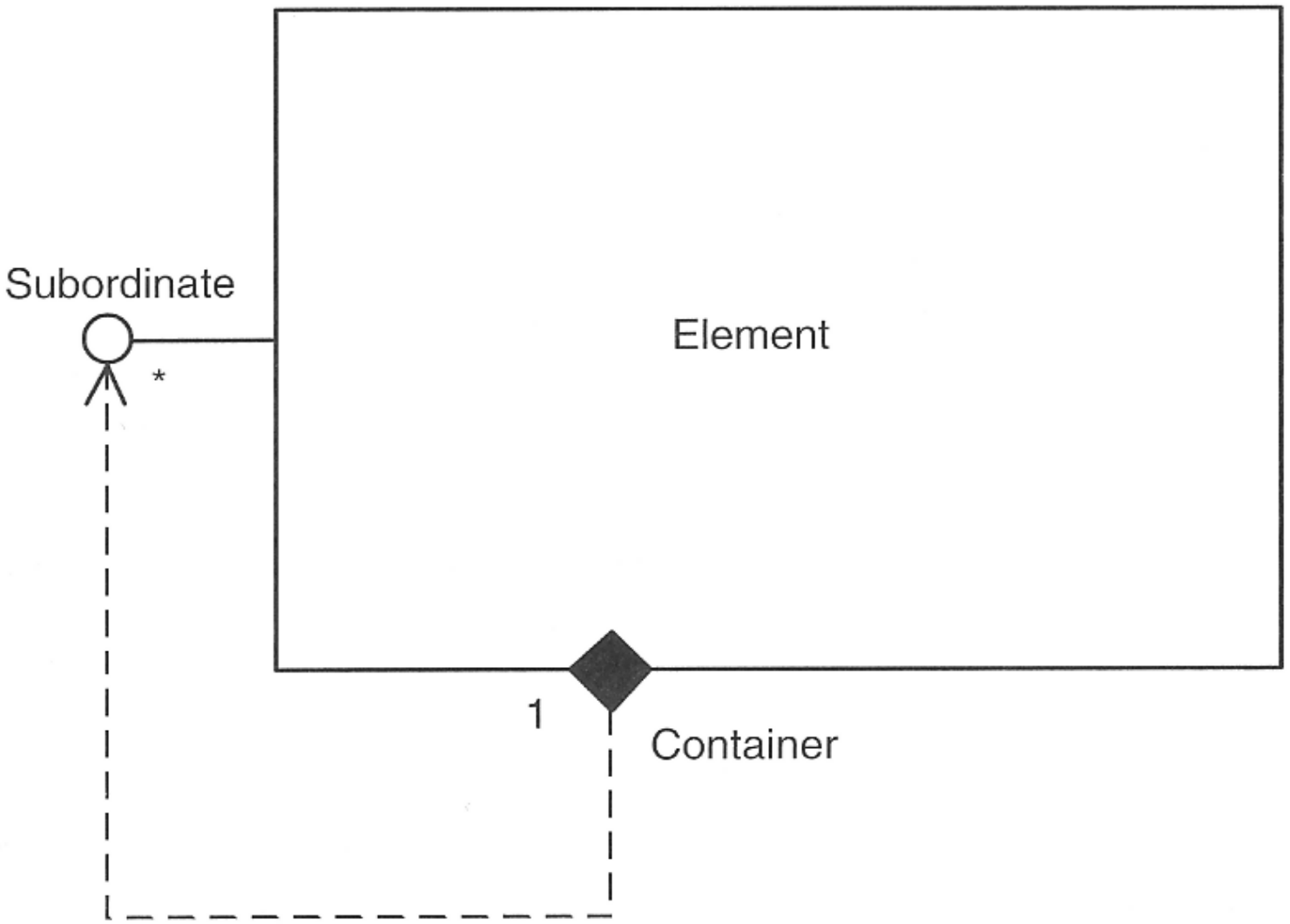
ECG Monitor Channel Pattern Example (4-10)



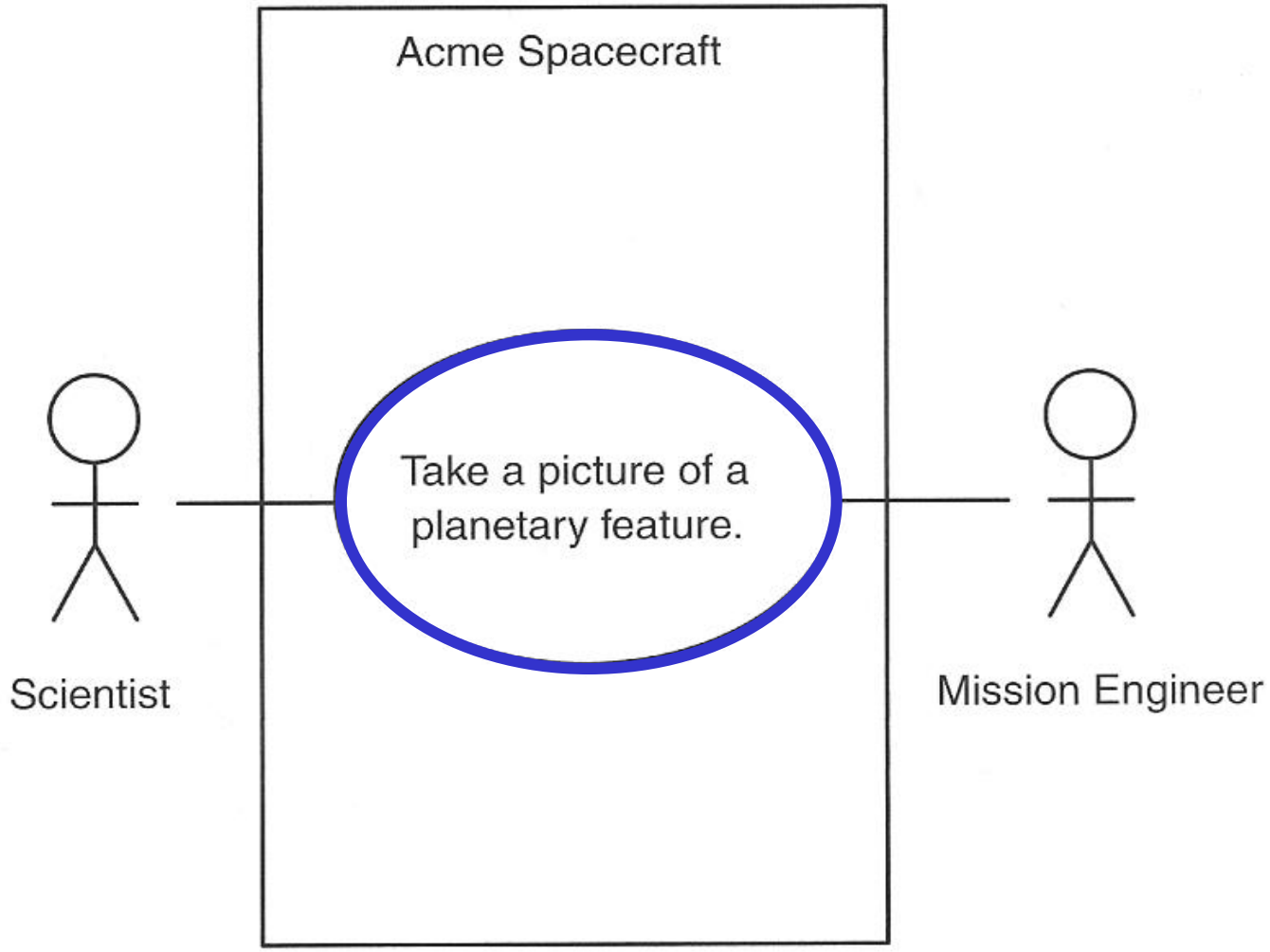
6. Recursive Containment Pattern

The Recursive Containment Patterns is a valuable pattern for very complex systems that realize thousands of requirements

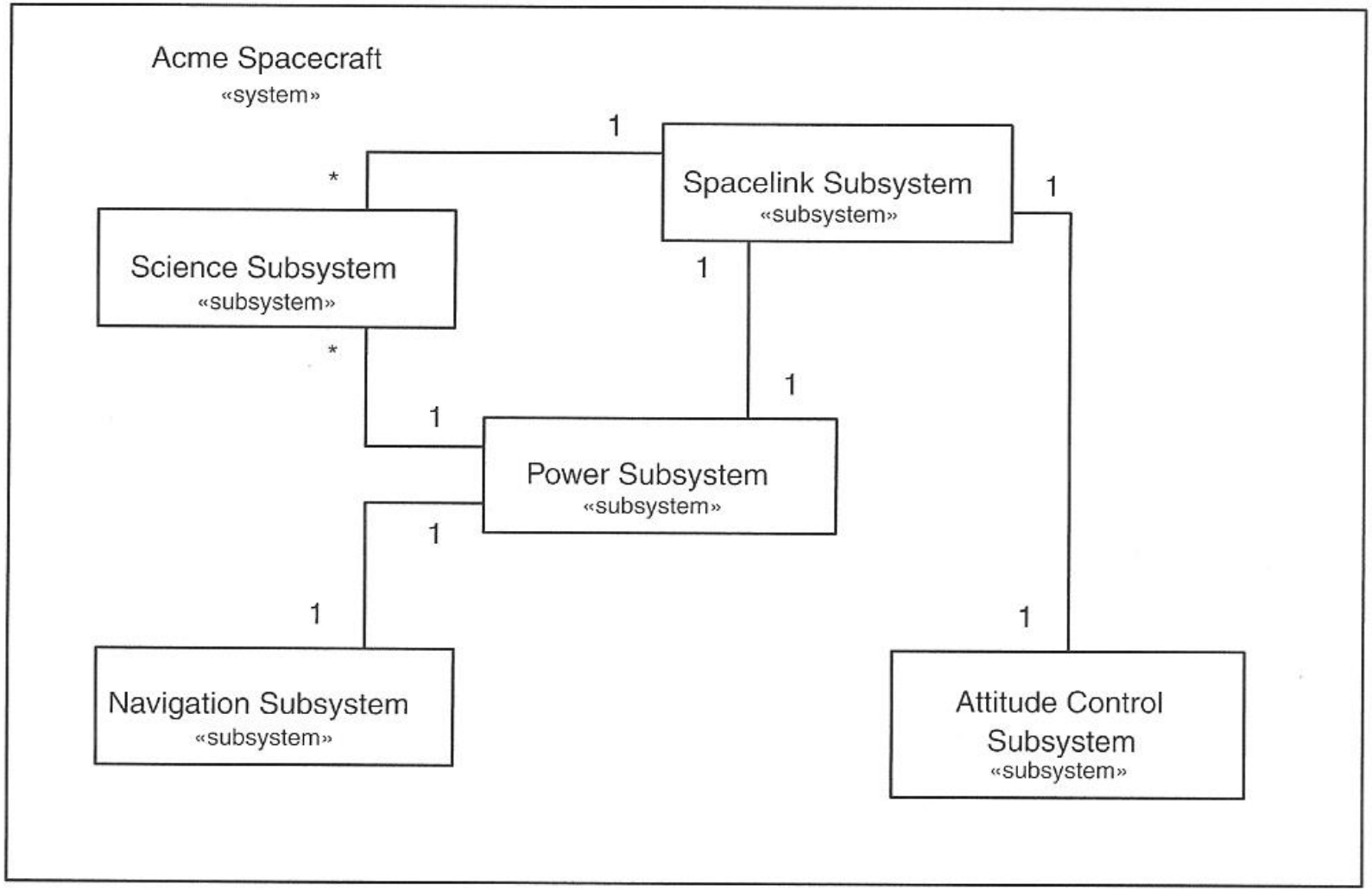
Recursive Containment Pattern Structure



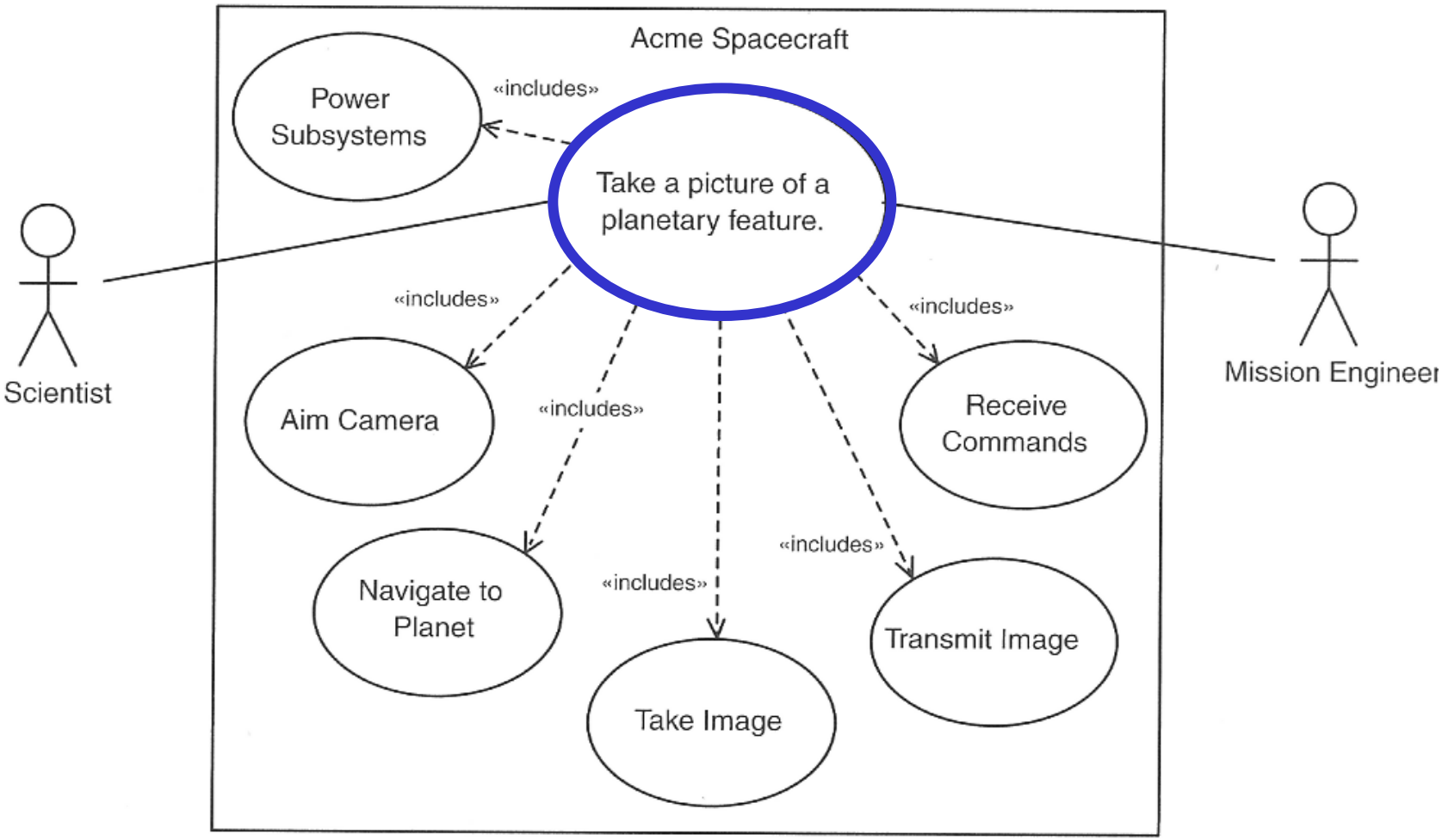
High Level Use Case (4-12)



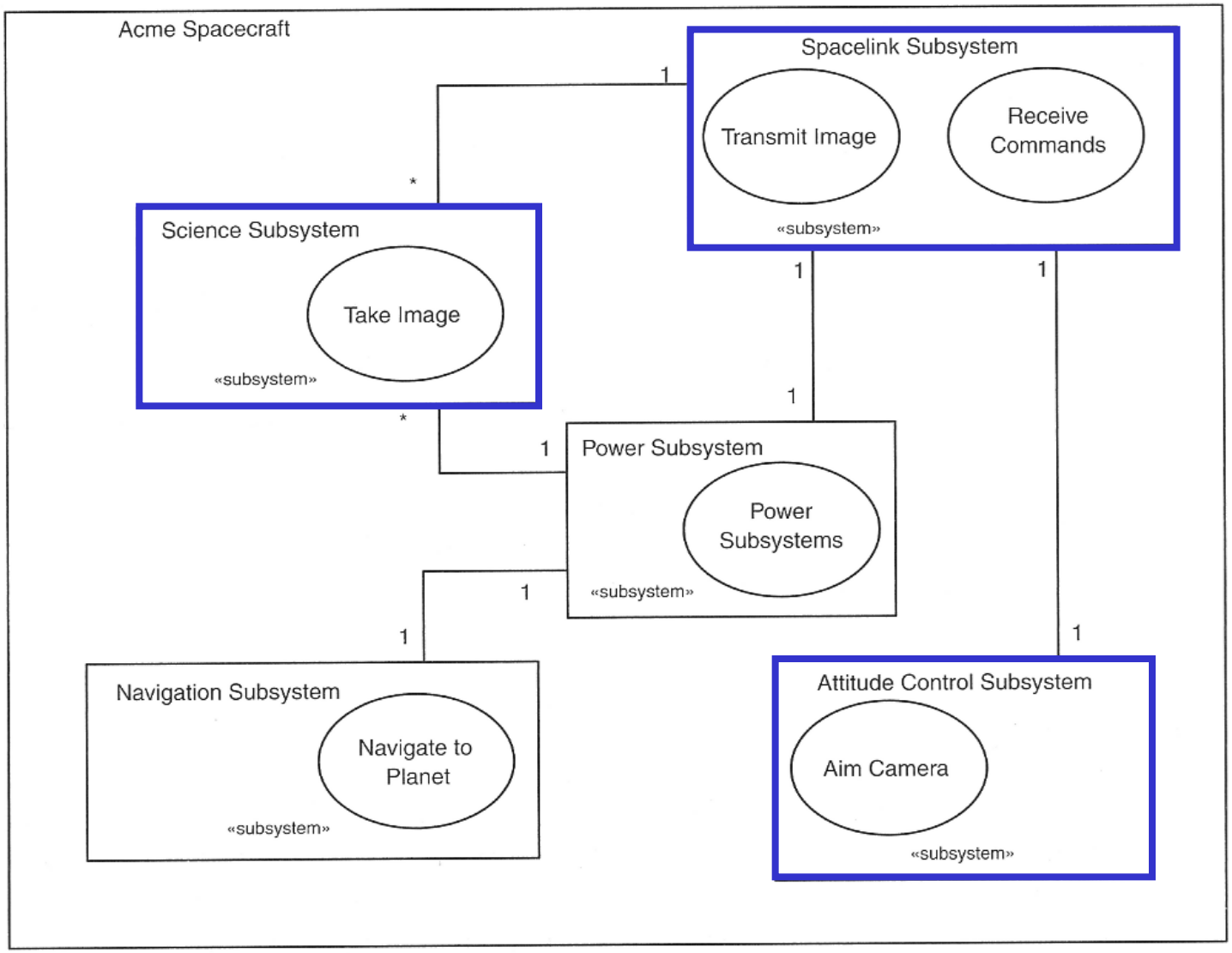
Spacecraft Subsystem Model (4-13)



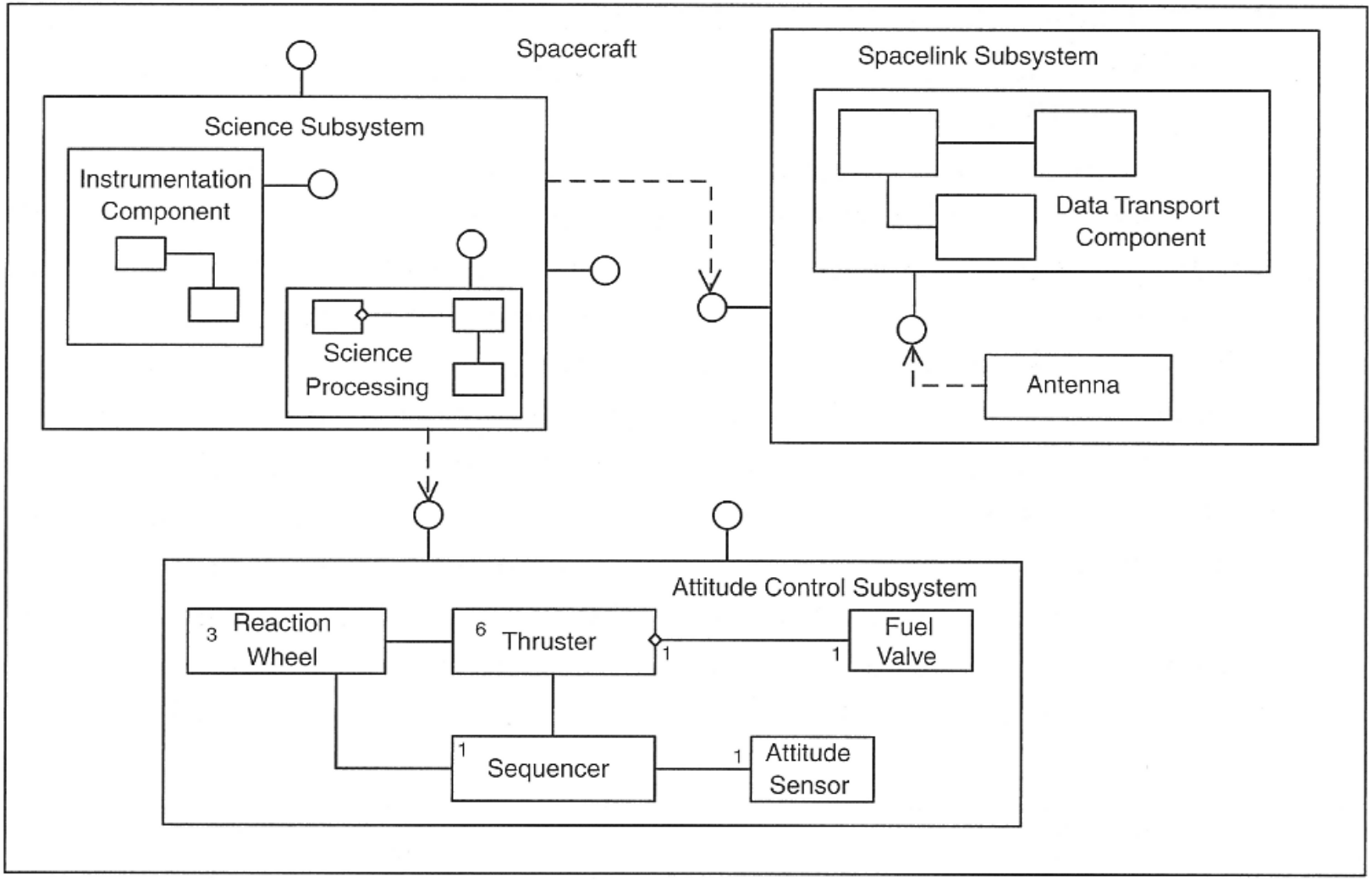
Decomposed Use Case (4-14)



Mapping Decomposed Use Cases to Subsystems (4-15)



Spacecraft Subsystems Details (4-16)



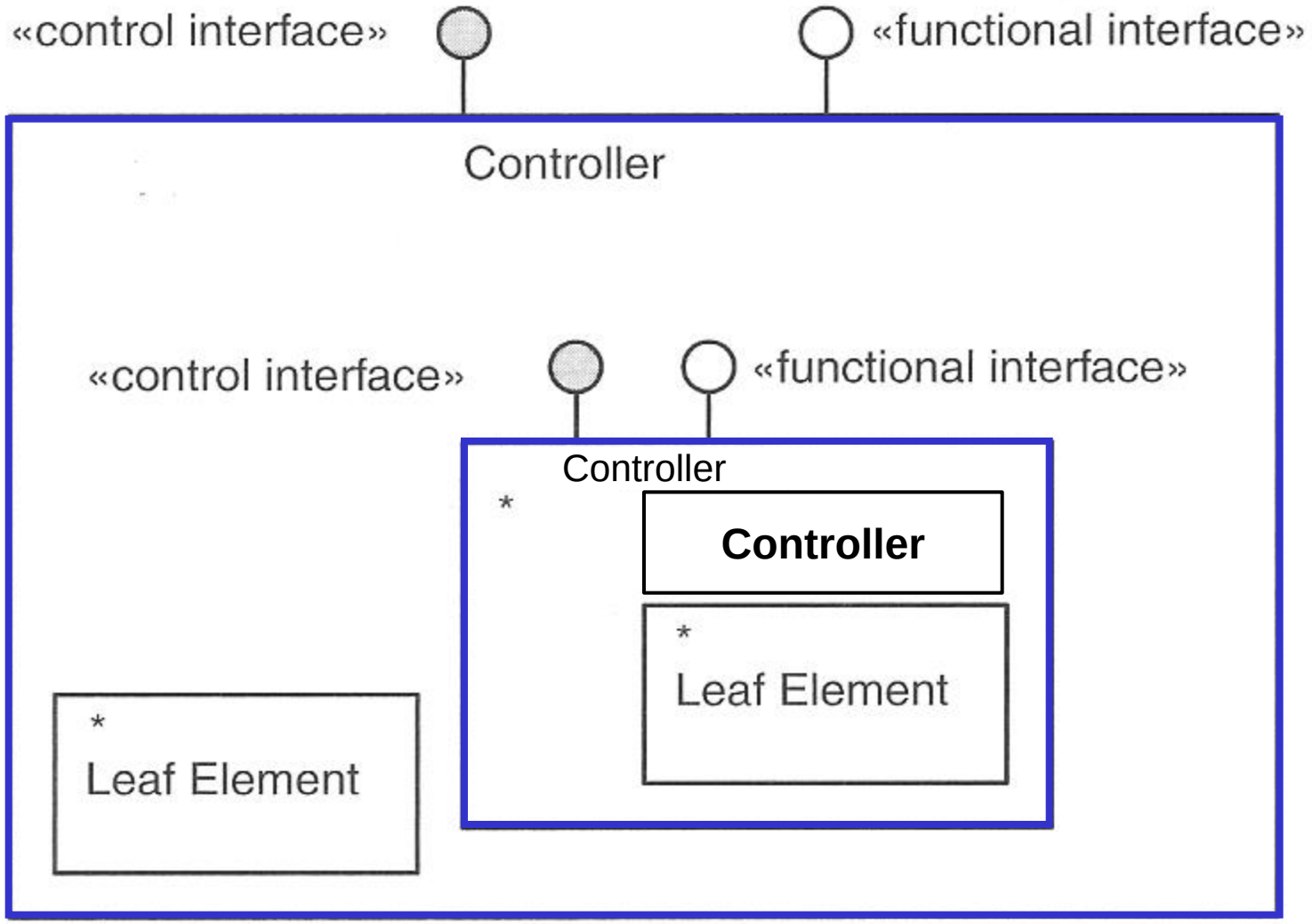
7. Hierarchical Control Pattern

A specialized form of the Recursive Containment Pattern that distributes complex control algorithms among its various pieces

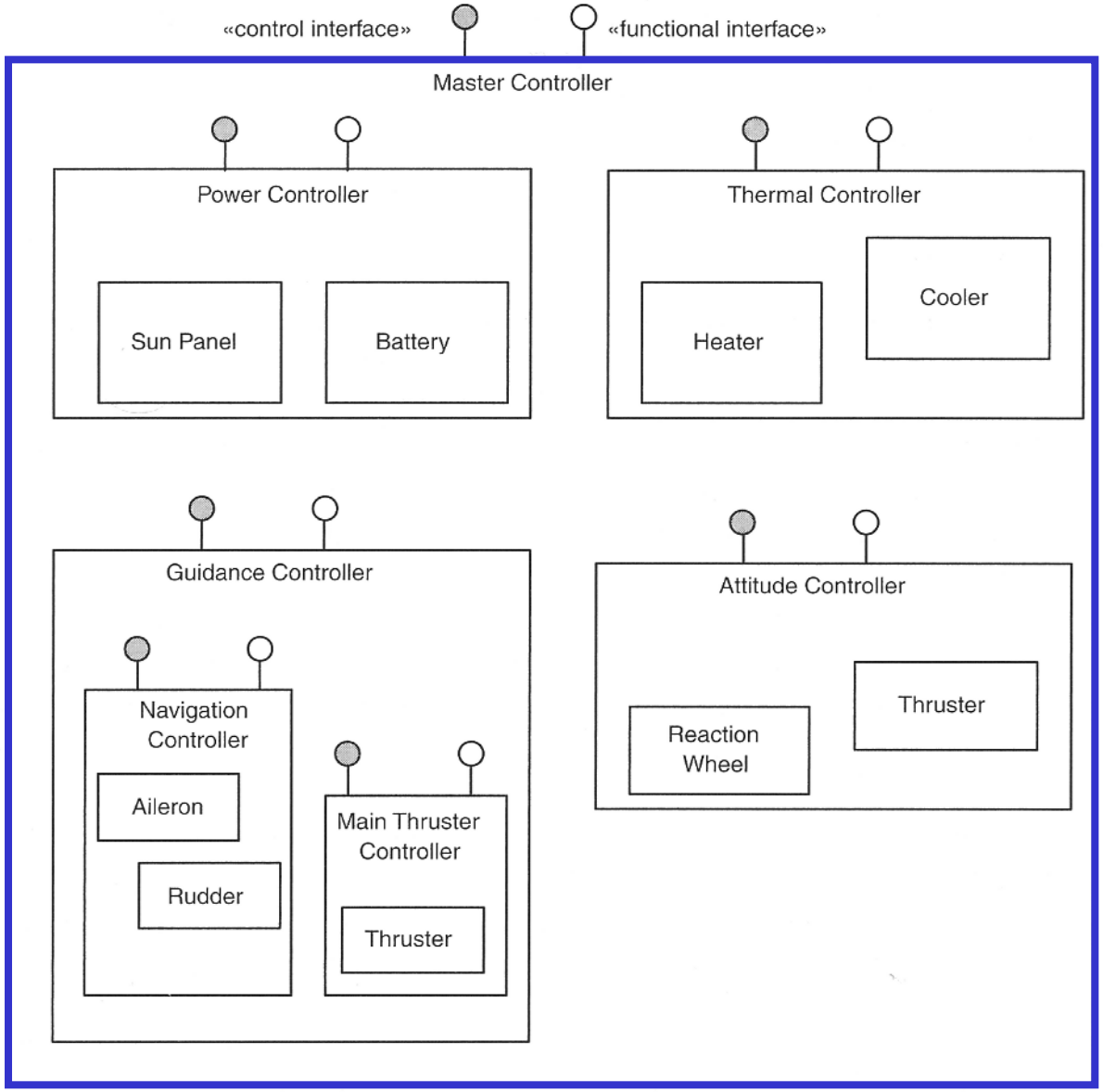
Abstract

- Two types of interfaces:
  - **Control interface**
    - set the quality of service (QOS) and selects policies
  - **Functional interface**
    - execute the desired behavior using the **QoS** and **policies** set via the control interface
- It is common to use a State Diagram to realize the control interface

Hierarchical Control Pattern Structure



Hierarchical Control Pattern Example



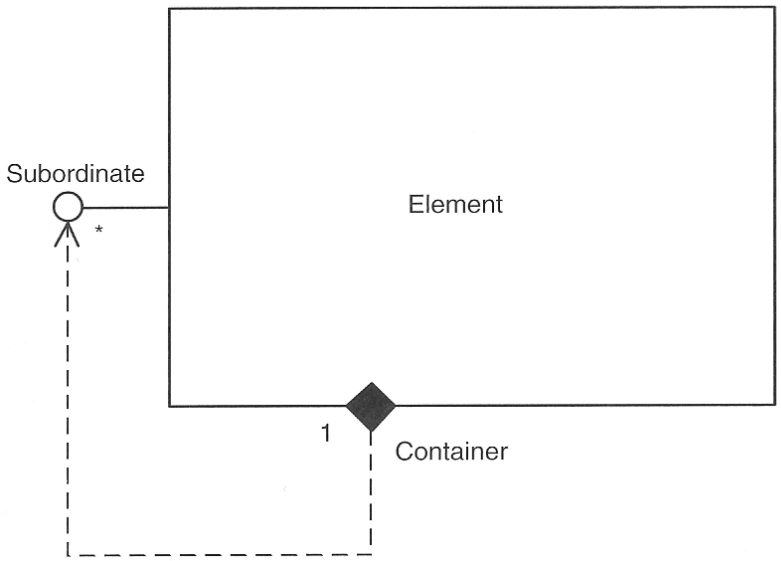
Implementation Strategies

- This pattern is often implemented bottom-up
  - a set of leaf elements in a tightly coupled collaboration are found to be jointly configurable
  - is seems useful to add a superordinate controller and define a state diagram for this

Related Patterns

- A specialized form of the Recursive Containment Pattern
  - based on the **GoF Composite Pattern** and Whole-Part Pattern

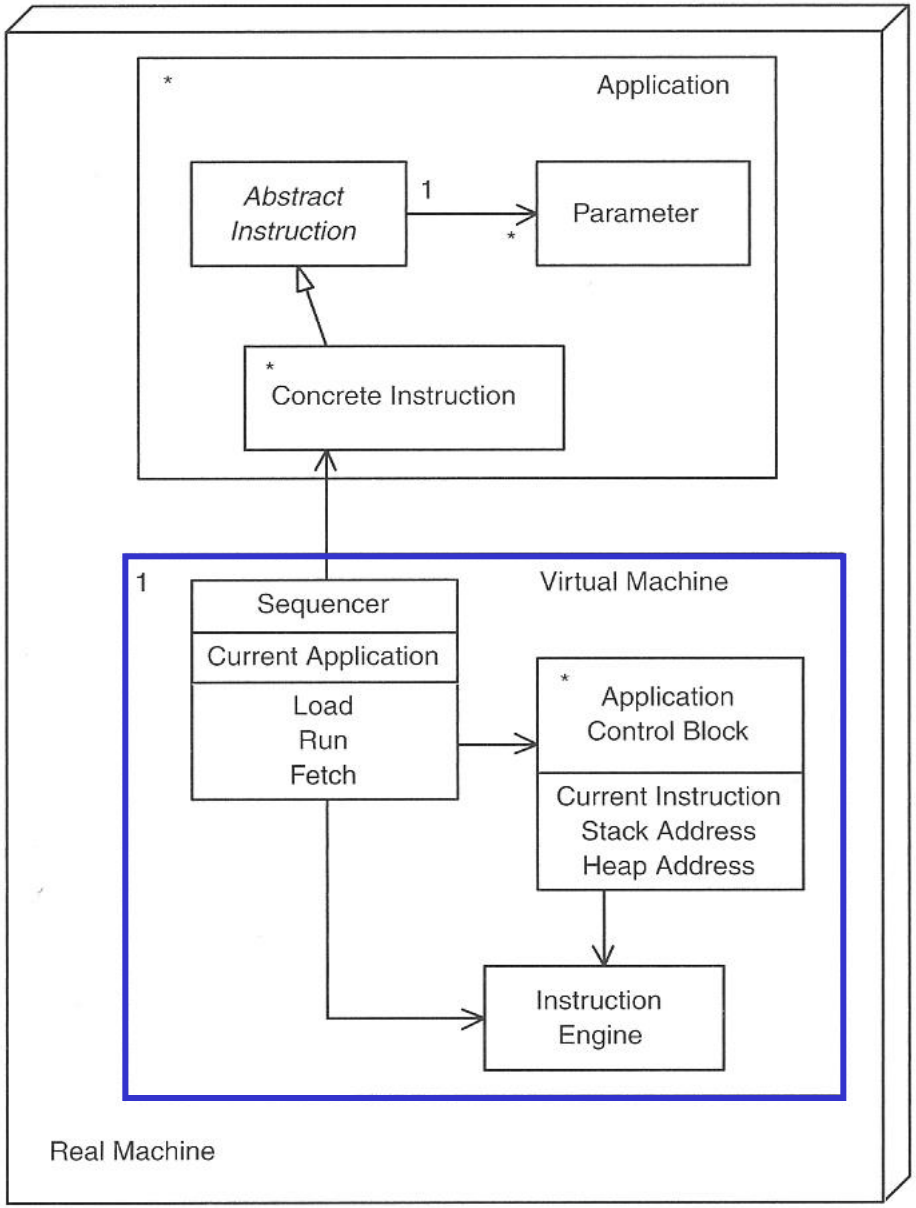
Recursive Containment Pattern



8. Virtual Machine Pattern

The Virtual Machine Pattern optimizes application portability at the expense of run-time efficiency

Virtual Machine Pattern Structure



Summary

- **SOLID – 5 OO design principles**
- **Seven architectural patterns:**
  - Layered Pattern
  - Five-Layer Pattern (Embedded Systems)
  - Microkernel Architectural Pattern
  - Channel Architecture Pattern
  - Recursive Containment Pattern
  - Hierarchical Control Pattern
  - Virtual Machine Pattern