澳 門 理 工 學 院
Instituto Politécnico de Macau
Macao Polytechnic Institute

Online Learning Materials

COMP223: Software Engineering

Architecture Design

Dr. Kim, Song-Kyoo (Amang)
Associate Professor,

Computer Science Program
MACAO POLYTECHNIC INSTITUTE
Macau, SAR

Session (Chapter 6) Objectives

- Architectural design decisions
- Architectural views
- Architectural patterns
- Application architectures

Online Learning Materials

澳門理工學院
Instituto Politécnico de Macau
Macao Polytechnic Institute

Architecture Design

Architectural design (1/8)

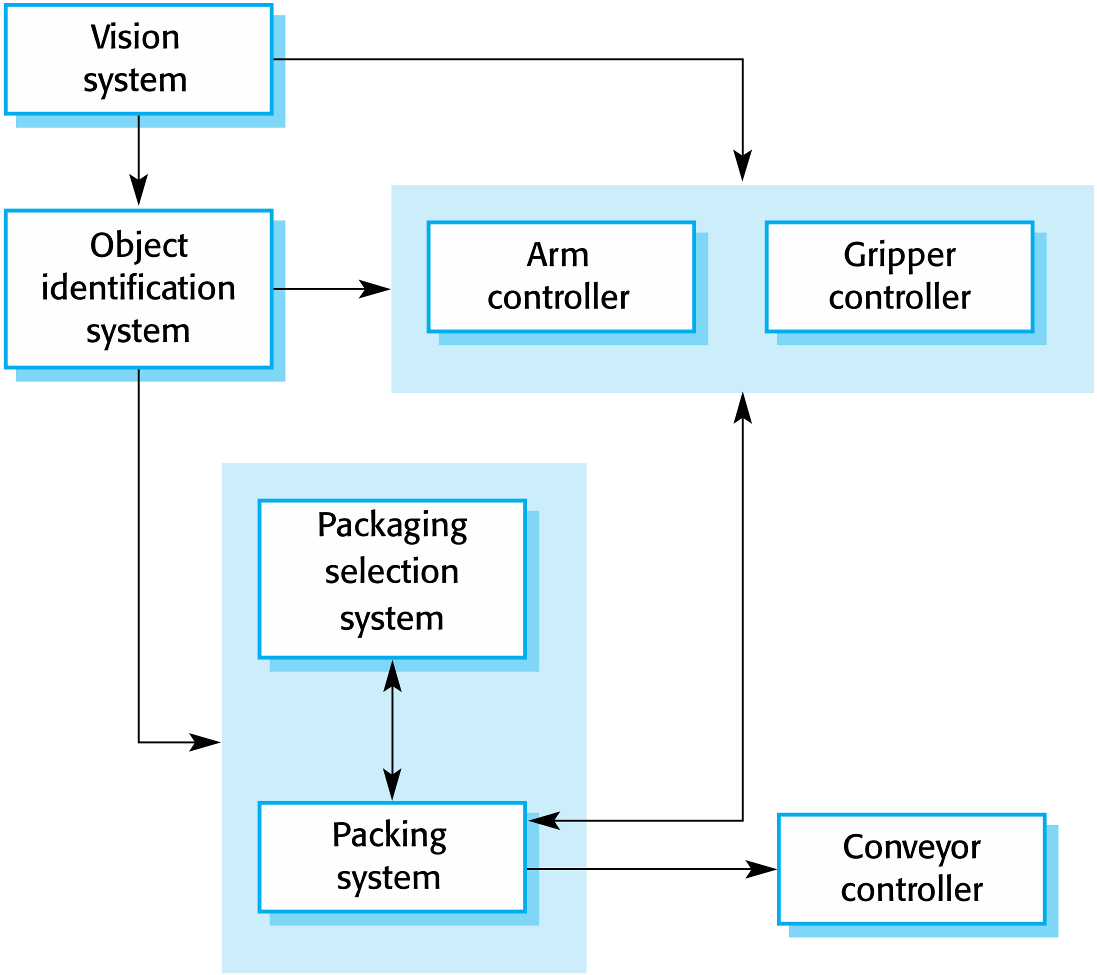
- Architectural design is concerned with understanding how a software system should be organized and designing the overall structure of that system.
- Architectural design is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them.
- The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

Architectural design (2/8)

- Agility and architecture
  - It is generally accepted that an early stage of agile processes is to design an overall systems architecture.
  - Refactoring the system architecture is usually expensive because it affects so many components in the system

Architectural design (3/8)

- The architecture of a packing robot control system



Architectural design (4/8)

- Architectural abstraction
  - Architecture in the small is concerned with the architecture of individual programs.
  - At this level, we are concerned with the way that an individual program is decomposed into components.
  - Architecture in the large is concerned with the architecture of complex enterprise systems that include other systems, programs, and program components.
  - These enterprise systems are distributed over different computers, which may be owned and managed by different companies.

Architectural design (5/8)

- Advantages of explicit architecture
  - **Stakeholder communication:** Architecture may be used as a focus of discussion by system stakeholders.
  - **System analysis:** Means that analysis of whether the system can meet its non-functional requirements is possible.
  - **Large-scale reuse:** The architecture may be reusable across a range of systems
  - Product-line architectures may be developed.

Architectural design (6/8)

- Architectural representations
  - Simple, informal block diagrams showing entities and relationships are the most frequently used method for documenting software architectures.
  - But these have been criticized because they lack semantics, do not show the types of relationships between entities nor the visible properties of entities in the architecture.
  - Depends on the use of architectural models.
  - The requirements for model semantics depends on how the models are used.

Architectural design (7/8)

- Box and line diagrams
  - Very abstract - they do not show the nature of component relationships nor the externally visible properties of the sub-systems.
  - However, useful for communication with stakeholders and for project planning.

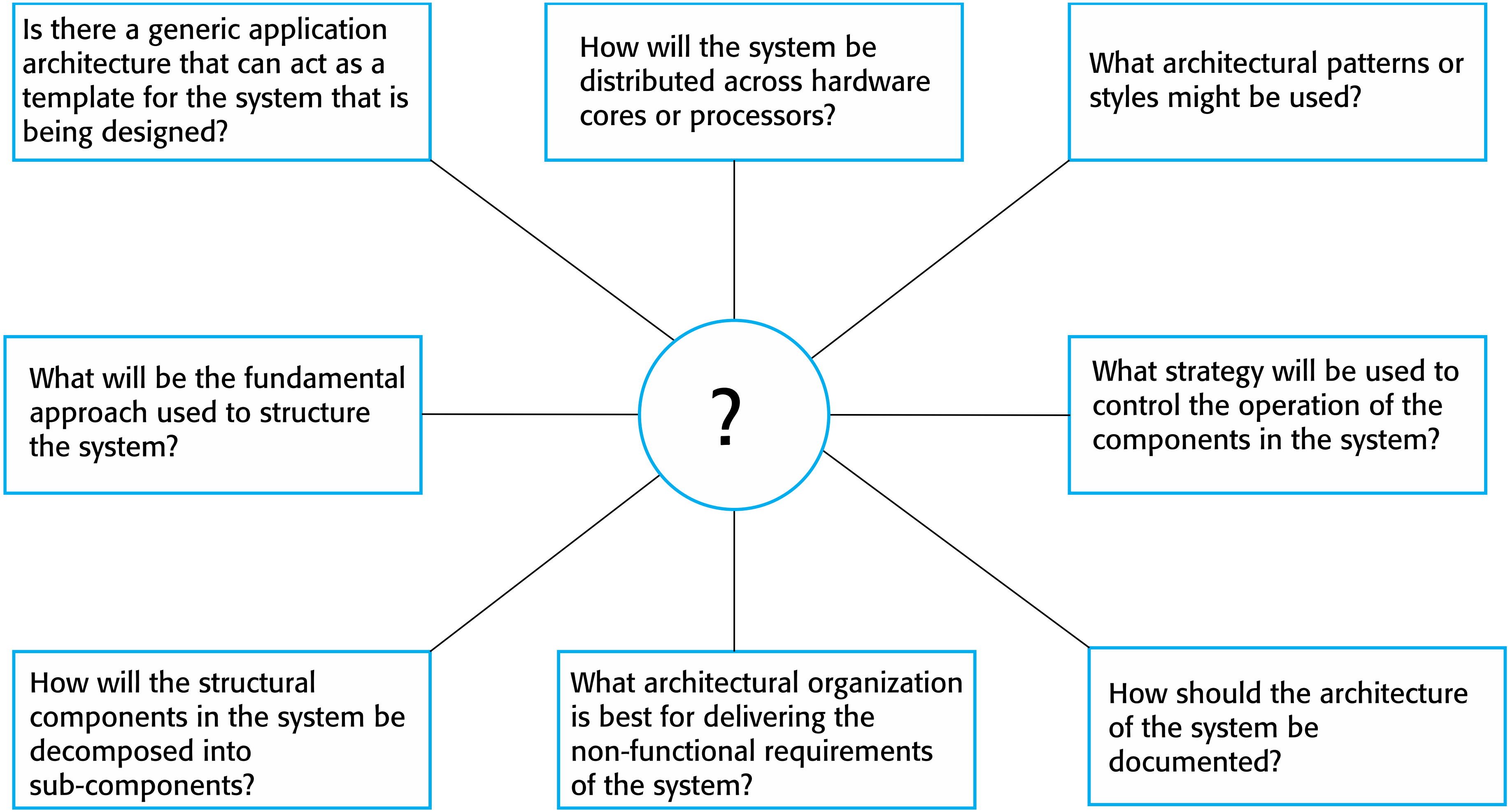
Architectural design (8/8)

- Use of architectural models
  - As a way of facilitating discussion about the system design
    - A high-level architectural view of a system is useful for communication with system stakeholders and project planning because it is not cluttered with detail.
    - Stakeholders can relate to it and understand an abstract view of the system. They can then discuss the system as a whole without being confused by detail.
  - As a way of documenting an architecture that has been designed
    - The aim here is to produce a complete system model that shows the different components in a system, their interfaces and their connections.

Architectural design decisions (1/4)

- Architectural design is a creative process so the process differs depending on the type of system being developed.
- However, a number of common decisions span all design processes and these decisions affect the non-functional characteristics of the system.

Architectural design decisions (2/4)



?

Is there a generic application architecture that can act as a template for the system that is being designed?

How will the system be distributed across hardware cores or processors?

What architectural patterns or styles might be used?

What will be the fundamental approach used to structure the system?

What strategy will be used to control the operation of the components in the system?

How will the structural components in the system be decomposed into sub-components?

What architectural organization is best for delivering the non-functional requirements of the system?

How should the architecture of the system be documented?

Architectural design decisions (3/4)

- Architecture reuse
  - Systems in the same domain often have similar architectures that reflect domain concepts.
  - Application product lines are built around a core architecture with variants that satisfy particular customer requirements.
  - The architecture of a system may be designed around one of more architectural patterns or ‘styles’.
  - These capture the essence of an architecture and can be instantiated in different ways.

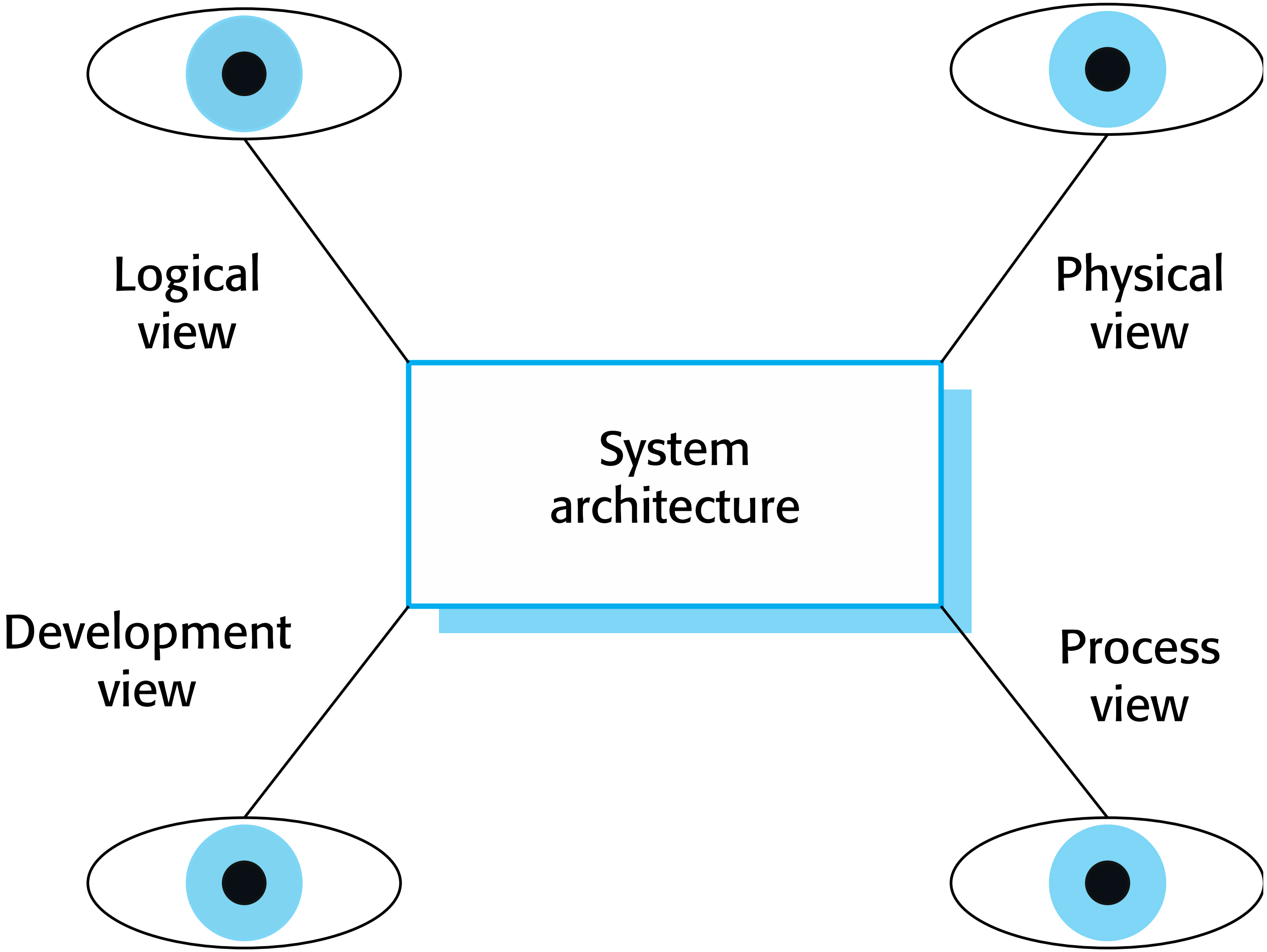
Architectural design decisions (4/4)

- Architecture and system characteristics
  - **Performance:** Localize critical operations and minimize communications.
  - Use large rather than fine-grain components.
  - **Security:** Use a layered architecture with critical assets in the inner layers.
  - **Safety:** Localized safety-critical features in a small number of sub-systems.
  - **Availability:** Include redundant components and mechanisms for fault tolerance.
  - **Maintainability**
  - Use fine-grain, replaceable components.

Architectural views (1/4)

- What views or perspectives are useful when designing and documenting a system’s architecture?
- What notations should be used for describing architectural models?
- Each architectural model only shows one view or perspective of the system.
  - It might show how a system is decomposed into modules, how the run-time processes interact or the different ways in which system components are distributed across a network.
  - For both design and documentation, you usually need to present multiple views of the software architecture.

Architectural views (2/4)



Architectural views (3/4)

- 4 + 1 view model of software architecture
  - **A logical view,** which shows the key abstractions in the system as objects or object classes.
  - **A process view,** which shows how, at run-time, the system is composed of interacting processes.
  - **A development view,** which shows how the software is decomposed for development.
  - **A physical view,** which shows the system hardware and how software components are distributed across the processors in the system.
  - Related using use cases or scenarios (+1)

Architectural views (4/4)

- Representing architectural views
  - Some people argue that the Unified Modeling Language (UML) is an appropriate notation for describing and documenting system architectures.
  - I disagree with this as I do not think that the UML includes abstractions appropriate for high-level system description.
  - Architectural description languages (ADLs) have been developed but are not widely used.

Session Summary (1/3)

- A software architecture is a description of how a software system is organized.
- Architectural design decisions include decisions on the type of application, the distribution of the system, the architectural styles to be used.
- Architectures may be documented from several different perspectives or views such as a conceptual view, a logical view, a process view, and a development view.
- Architectural patterns are a means of reusing knowledge about generic system architectures.

澳門理工學院
INSTITUTO POLITÉCNICO DE MACAU
MACAO POLYTECHNIC INSTITUTE