Architectural Design

Lecture # 25

Lecture by Engr. Sidra

Objectives

- The objective of this chapter is to introduce the concepts of software architecture and architectural design. When you have read the chapter, you will:
  - Understand why the architectural design of software is important;
  - Understand the decisions that have to be made about the system architecture during the architectural design process;
  - Have been introduced to the idea of architectural patterns, well-tried ways of organizing system architectures, which can be reused in system designs;

Software Architecture

- The software architecture of a program or computing system is the structure or structures of the system, which comprise the software components, the externally visible properties of those components, and the relationships among them.
- The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design.
- The output of this design process is a description of the software architecture.

Architectural design

- An early stage of the system design process.
- Represents the link between specification and design processes.
- Often carried out in parallel with some specification activities.
- It involves identifying major system components and their communications.
- Not incremental, even in Agile, cost of change is high.

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 as distribute 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.

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.

Why Architecture?

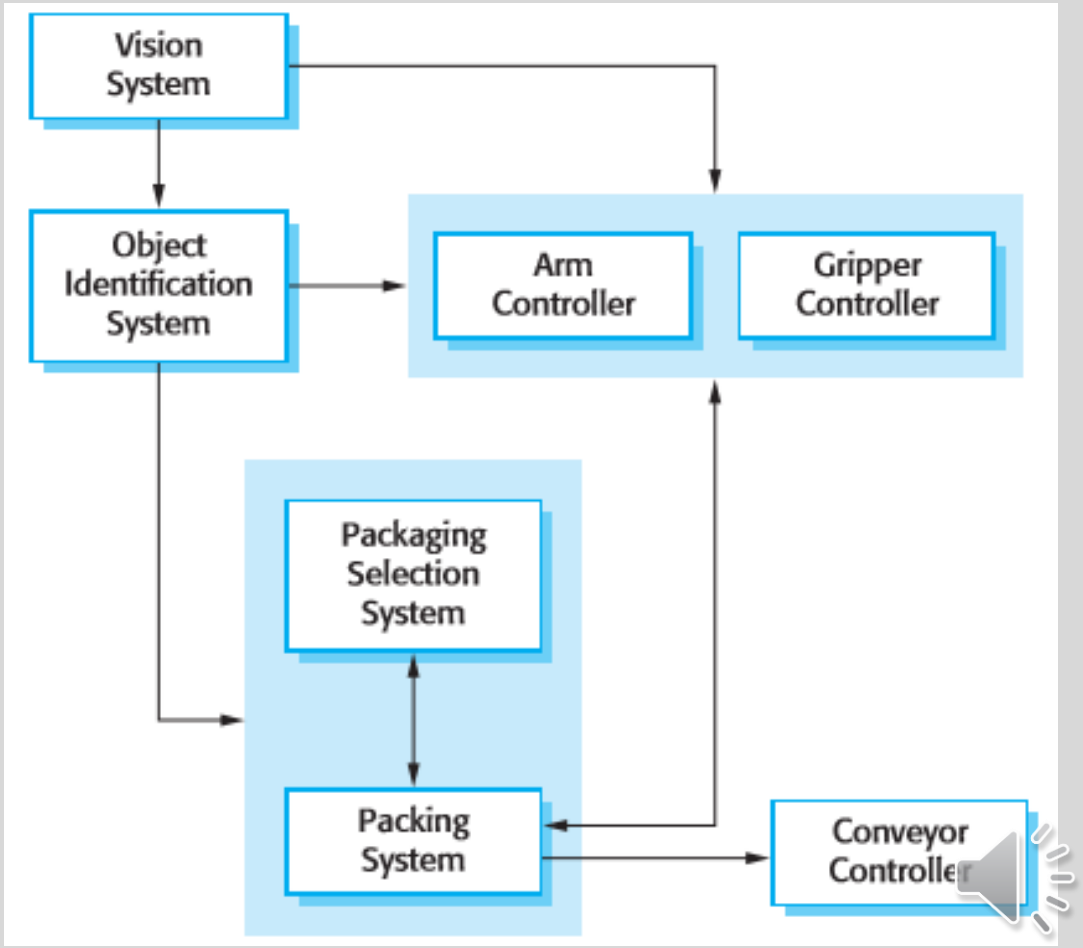
- Architecture is a representation of a system that enables the software engineer to :
  - analyze the effectiveness of the design in meeting its stated requirements,
  - consider architectural alternatives at a stage when making design changes is still relatively easy, and
  - reduce the risks associated with the construction of the software.

Why is Architecture Important?

- Representations of software architecture are an enabler for communication between all parties (stakeholders) interested in the development of a computer-based system.
- The architecture 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.
- Architecture "constitutes a relatively small, intellectually graspable model of how the system is structured and how its components work together" [BAS03]

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.



Architectural Design Decisions

- 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.
  - Is there a generic application architecture that can be used?
  - How will the system be distributed?
  - What architectural styles are appropriate?
  - What approach will be used to structure the system?
  - How will the system be decomposed into modules?
  - What control strategy should be used?
  - How will the architectural design be evaluated?
  - How should the architecture be documented?

Architectural views

- 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.

4 + 1 view model of SW arch. (Krutchen)

- A logical view, which shows the key abstractions in the system as objects or object classes. (Requirements)
- 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.
- All the views above related through a user view (+1)