

# Software Design and Architecture

Software Management Structure and Software Elements

# Software Management Structure

- ▶ A large software project is normally designed and implemented by several project teams, each having its well-defined responsibilities at specific SDLC process stages
- ▶ In this structure, each element consists of specific manipulation (design, implementation, debugging, etc.) of specific code units assigned to the same project team, and the connectors are derived from runtime dependency among the code units and software process dependencies
- ▶ Software management structures are also often used for project resource allocation

# Software Elements

- ▶ At runtime each software element has its well-defined functions and these elements are connected into a dependency graph through connectors
- ▶ The elements of a software architecture are usually refined through multiple transformation steps based on the element attributes and the project requirement specification
- ▶ Based on the function assignment each software element may have different synchronization and performance constraints
  - ▶ reentrant , multiplicity, ...

# Guidelines for Mapping Runtime Elements

- ▶ If an element is reentrant, it can be implemented by a thread or a process
- ▶ If an element is not reentrant and multiple threads or processes may need to communicate with it, it must be run on separate threads or processes so they can be thread-safe
- ▶ If an element has high multiplicity and its performance is important to the global system performance, an application server should be used for its implementation

## Guidelines for Mapping Runtime Elements ...

- ▶ If there are heavy computations in the elements for deployment at a particular location, a cluster of processors should be considered for added CPU data processing power
- ▶ If an element is assigned complex but well-defined functions similar to those of some commercial off-the-shelf software components and the performance of this element is not critical, then it is more cost-effective to use an existing software component to implement the element's functions

# Guidelines for Mapping Runtime Elements ...

- ▶ A complex element can be expanded into a sub-system with its own elements and connectors
- ▶ A complex element can be transformed into a sequence of vertical layered elements if each layer provides a virtual machine or interface to its immediate upper layer element and each layered element hides away some low-level system details from the upper layers
- ▶ A complex element can be transformed into a sequence of horizontally tiered elements if the business logic can be achieved by processing data with a sequence of discrete processing stages and these processing stages can be implemented by tiered elements with well-defined interfaces and balanced workloads

# Summary

- Introduce software management structure
- Introduce software elements
- Guidelines to map software elements