IBM Software Group

Essentials of Visual Modeling with UML

Module 7: Other UML Diagrams

Rational. software

Objectives

- Demonstrate how to read and interpret a:
  - Statechart diagram
  - Deployment model
  - Component diagram

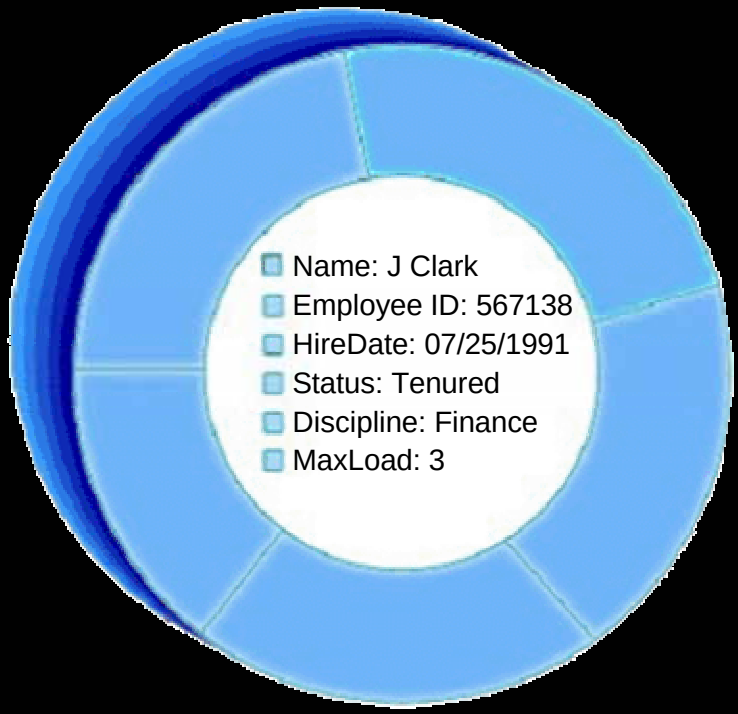
Where Are We?

- ★ Statechart diagrams
- Deployment diagrams
- Component diagrams

Review: An Object Has State

- The state of an object is one of the possible conditions in which an object may exist.
- The state of an object normally changes over time.

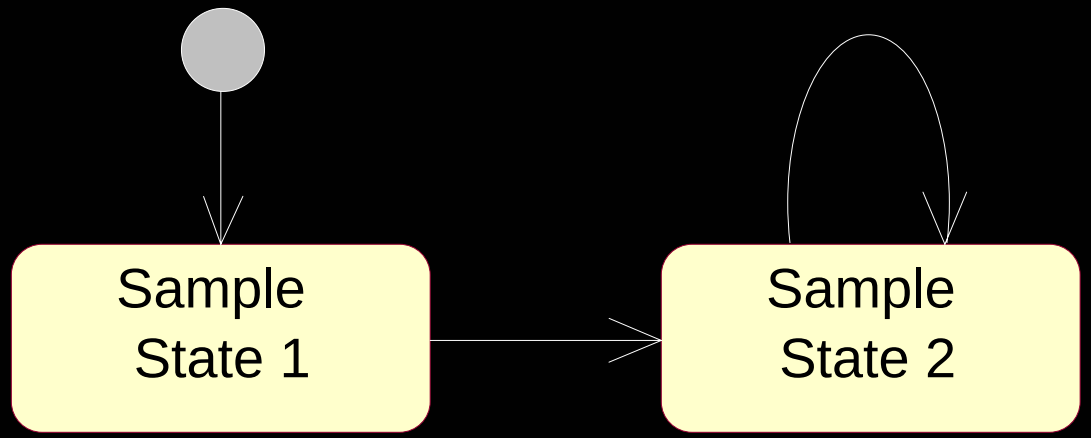
Name: J Clark
Employee ID: 567138
Date Hired: July 25, 1991
Status: Tenured
Discipline: Finance
Maximum Course Load: 3 classes



Professor Clark

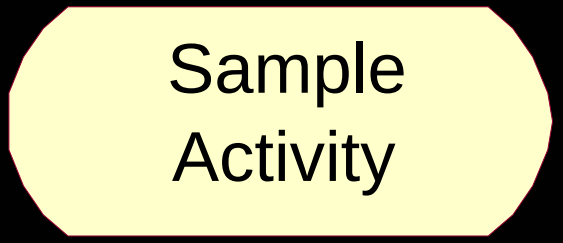
What Are Statechart Diagrams?

- A statechart diagram shows a state machine.
- It specifies the sequence of states that an object can be in:
  - The events and conditions that cause the object to reach those states
  - The actions that take place when those states are reached



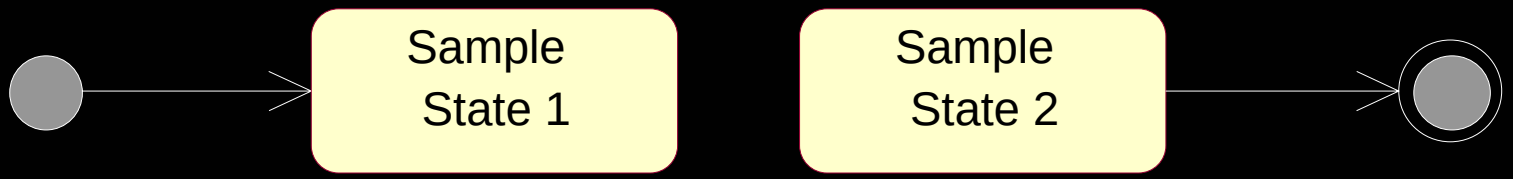
Drawing States

- A state is represented as a rounded rectangle on a statechart diagram.
- As a comparison, note the subtle difference between a state and an activity.



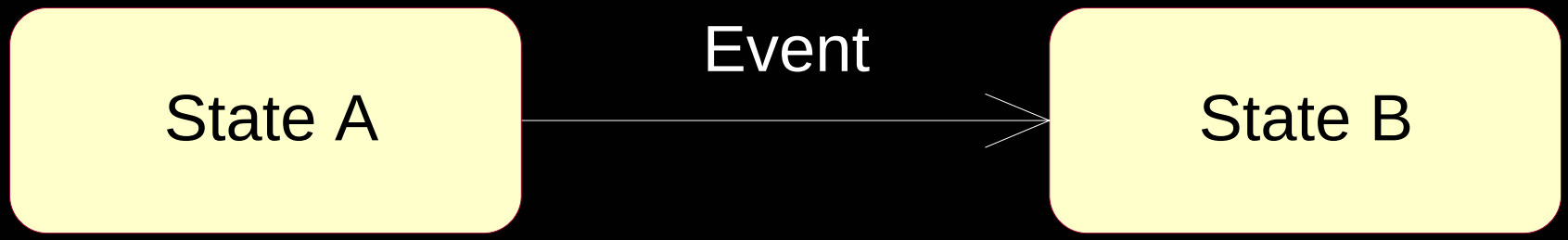
Special States

- The initial state is the state entered when an object is created.
  - An initial state is mandatory.
  - Only one initial state is permitted.
  - The initial state is represented as a solid circle.
- A final state indicates the end of life for an object.
  - A final state is optional.
  - A final state is indicated by a bull’s eye.
  - More than one final state may exist.



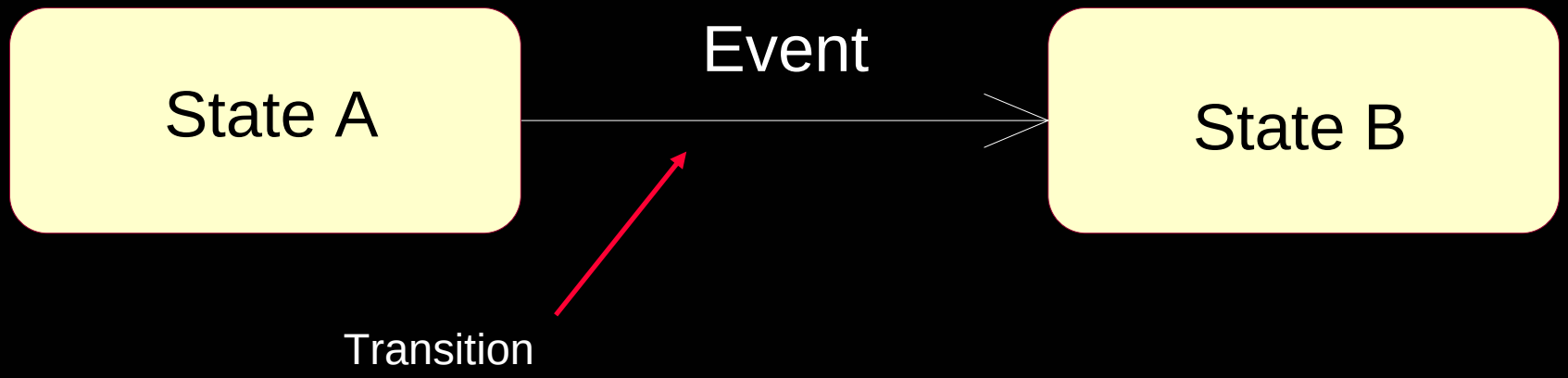
What Are Events?

- An event is the specification of a significant occurrence that has a location in time and space.
  - An event is an occurrence of a stimulus that can trigger a state transition.
  - Example:
    - Adding a student to a course
    - Creating a new course

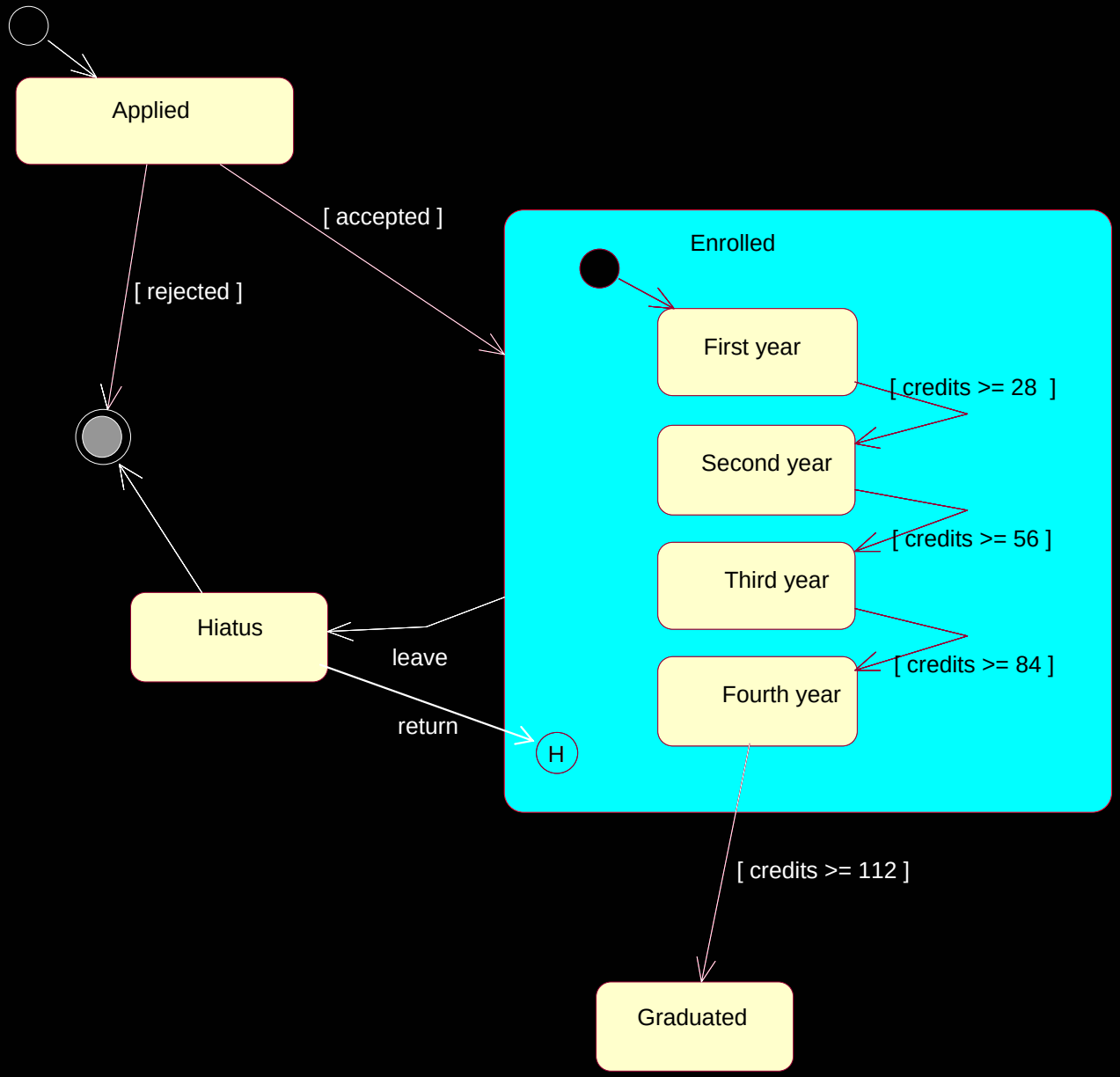


What Are Transitions?

- A transition is a change from an originating state to a successor state as a result of some stimulus.
  - The successor state could possibly be the originating state.
- A transition may take place in response to an event.
- Transitions can be labeled with events.



Example: Statechart



Where Are We?

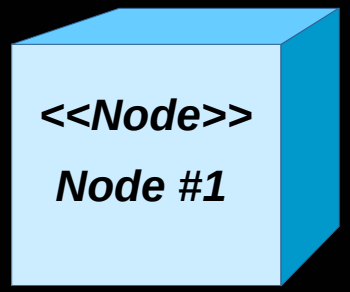
- Statechart diagrams
- ★ Deployment diagrams
- Component diagrams

What Is a Deployment Diagram?

- The deployment diagram shows:
  - Configuration of processing nodes at run-time
  - Communication links between these nodes
  - Component instances and objects that reside on them

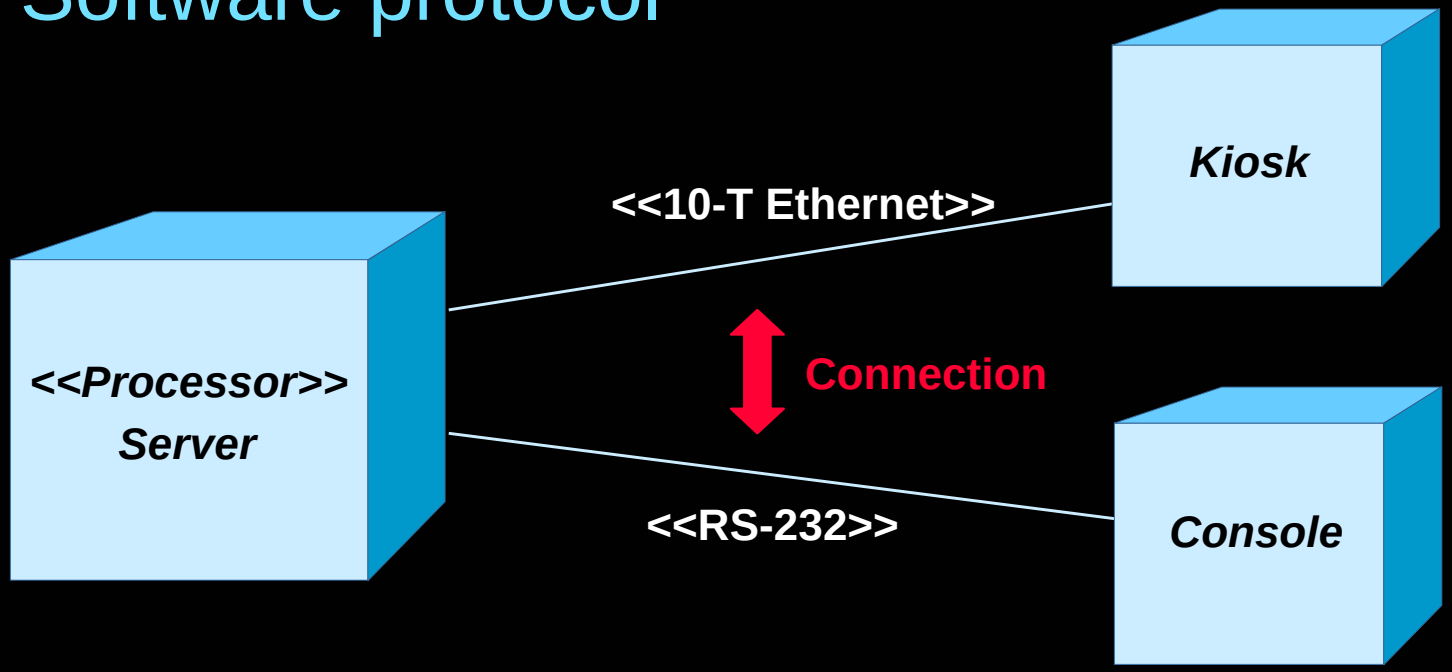
What Is a Node?

- A physical element that exists at run-time and represents a computational resource.
- Types:
  - Processor Node
    - Executes system software
  - Device Node
    - Support device
    - Typically controlled by a processor

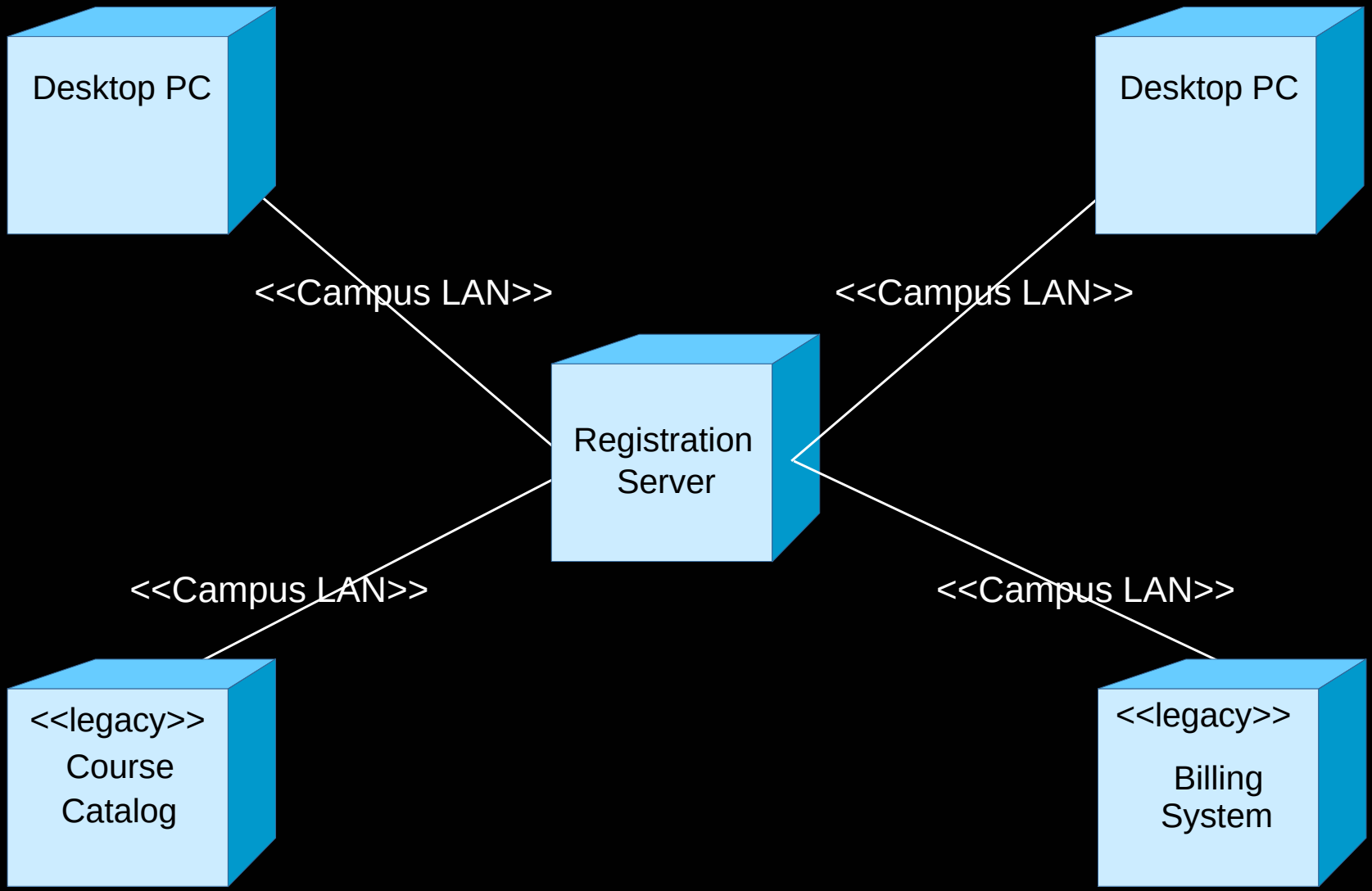


What Is a Connection?

- A connection represents a:
  - Communication mechanism
    - Physical medium
    - Software protocol



Example: Deployment Diagram

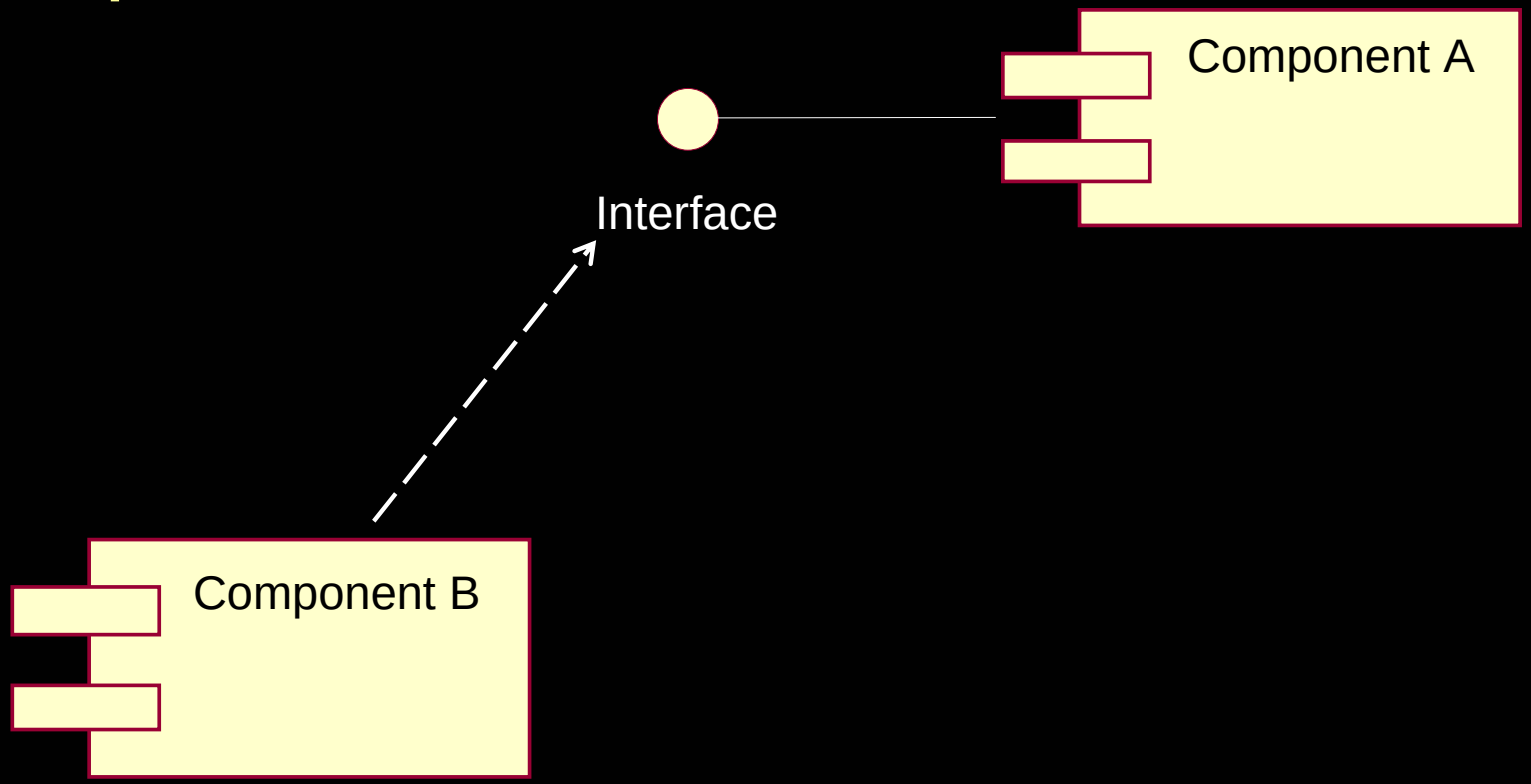


Where Are We?

- Statechart diagrams
- Deployment diagrams
- ★ Component diagrams

What Is a Component Diagram?

- A diagram that shows the organization of and the dependencies among a set of components.



What Is a Component?

- A modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.
- It conforms to and provides the physical realization of a set of interfaces.

Components UML 1.4 and Beyond

- UML 1.4 introduces concept of Artifacts:
  - An Artifact represents a physical piece of information that is used or produced by a software development process. Examples of Artifacts include models, source files, scripts, and binary executable files.
- To distinguish between artifacts in general, and the artifacts that make up the implementation, we introduce a new term:
  - Implementation Element - the physical parts (UML artifacts) that make up an implementation, including software code files (source, binary or executable), and data files.

Components UML 1.4 and Beyond (cont.)

- Component becomes similar to subsystem:
  - can group classes to define a larger granularity units of a system
  - can separate the visible interfaces from internal implementation
  - can have instances that execute at run-time
- The distinction between "component" and "artifact" is new in UML 1.4.
  - Many tools, profiles, and examples continue to use "component" to represent implementation elements

Review

- Define state. How do you determine the classes with significant state?
- What is a statechart diagram? Describe the different parts of the diagram.
- How do statechart diagrams map to the rest of the world?
- What is the purpose of a deployment diagram?
- What is a component diagram?