Chapter 2, Modeling with UML: UML 2 Highlights

Outline for this class

✓ Overview of important changes in UML 2
  ➢ Deployment diagrams
  • Sequence diagrams
UML 2 Deployment Diagrams

Two node types:

- **Device**
  - a physical computational resource with processing capability upon which artifacts may be deployed for execution.

- **Execution environment**
  - a node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts.

---

Deployment Diagram Changes II

- **Artifacts** can now manifest any packageable element, not just components
  - An artifact is the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system.

- **Manifestation** (the concrete physical rendering of one or more model elements by an artifact) is shown by a dependency with keyword <<manifest>>
Deployment Diagram Changes III

- A deployment diagram can have a deployment specification

Interaction Diagrams
Interaction Diagrams

• New concept of interaction fragments

• Before we go into detail with interaction fragments, let’s cover the concept of an interaction.

Interaction Diagrams

• Four types of interaction diagrams:
  • Sequence diagrams
  • We will not study the following (by now at least):
    • Communication diagrams
    • Interaction overview diagrams
    • Timing diagrams
  • The basic building block of an interaction diagram is the interaction
    • An interaction is a unit of behavior that focuses on the observable exchange of information between connectable elements
Example of an Interaction: Sequence Diagram

Interaction Fragment

- Interaction Fragment
  - Is a piece of an interaction
  - Acts like an interaction itself
- Combined Fragment
  - Is a subtype of interaction fragment
  - Defines an expression of interaction fragments
- An expression of interaction fragments is defined by
  - An interaction operator and interaction operands.
Interaction Operators

• The following operators are allowed in the combination of interaction fragments:
  • alt
  • opt
  • par
  • loop
  • critical
  • neg
  • assert
  • strict
  • seq
  • Ignore
  • consider

Alt Operator

• The interaction operator **alt** indicates a choice of behavior between interaction fragments
  • At most one interaction fragment (that is, an InteractionOperand) is chosen
  • The chosen interaction fragment must have an explicit or implicit guard expression that evaluates to true at this point in the interaction
    • A guard can be
      • a boolean expression (called InteractionConstraint)
      • else (a reserved word)
    • If the fragment has no guard expression, true is implied.
Example of a Combined Fragment using the alt operator

- The interaction operator **alt** indicates a choice of behavior between interaction fragments

Opt and Break Operators

**option:**
- The interaction operator **opt** designates a choice of behavior where either the (sole) operand happens or nothing happens.

**break:**
- The interaction operator **break** represents a breaking scenario: The operand is a scenario that is performed instead of the remainder of the enclosing interaction fragment.
Parallel and Critical Operator

par
The interaction operator $\text{par}$ designates a parallel merge between the behaviors of the operands of a combined fragment.

critical
The interaction operator $\text{critical}$ designates that the combined fragment represents a critical region.

Example of a Critical Region

Problem statement: The telephone Operator must make sure to forward a 911-call from a Caller to the Emergency system before doing anything else. Normal calls can be freely interleaved.