

CSCI 253
Object Oriented Design:
UML
George Blankenship

UML George Blankenship 1

Outline

- Background
- UML overview
- Structural diagrams
- Behavioral diagrams

UML George Blankenship 2

Rationale

- Visual aids are basic tool for construction
 - Define the attributes of the construction process
 - Define the attributes of the constructed building
- Architectural drawing
 - Site elevation describes the relationship between a building an its environment
 - Floor plan describes the relationship between the structure elements
- Architectural model
 - Visual depiction of the intended building

UML George Blankenship 3

Background

- Clear description of computational concept
 - Diagram on paper napkin
 - Flow chart
- Pictorial depiction of concept
 - Computational relationships
 - Computational flow
 - Code components
- Unified Modeling Language

UML George Blankenship 4

Historical

- Flow diagrams
 - Graph of a process
 - Processing description nodes
 - Processing decision nodes
- State machine diagrams
 - Formalized processing description based upon I/O (messages)
- File I/O diagrams
 - Define file contents (records)
 - Define file relationships (data movement)
 - Define program references

UML George Blankenship 5

Unified Modeling Language

- Common method to model relationships, behaviors, and high-level organization
 - Easy to learn
 - Efficient to write
- Models used to describe wide spread of concepts
 - Business process
 - Requirements
 - Software
- First version in 1994 (Booch, Rumbaugh, and Jacobson)
- Standardized by Object Management Group (UML 2.0)

UML George Blankenship 6

UML Diagrams

- Structural diagrams
 - Physical organization
 - Relationship between elements
- Behavioral diagrams
 - Internal operation
 - Requirements
 - Element operation
 - Internal state changes

UML George Blankenship 7

Structural Diagrams

- Class diagram
 - Static depiction of a class
 - Attributes and operations
- Object diagram
 - Static depiction of the objects of a system
- Component diagram
 - Static depiction of a system
 - Aggregation of classes into system components

UML George Blankenship 8

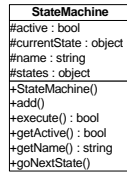
Behavioral Diagrams

- Activity and state machine diagrams
 - Processing flow
- Communication and sequence diagrams
 - Message flow
 - Message processing
- Use case diagrams
 - Functional requirements

UML George Blankenship 9

Class Diagrams

- Rectangular box with three compartments
 - Name (top compartment)
 - Attributes (middle compartment)
 - Operations (bottom compartment)



UML

George Blankenship

10

Name

- (Convention)
- Starts with capital letter
- Bold font
- Abstract class indicated by italic font
 - Operation signature
 - Cannot be instantiated (no code)

UML

George Blankenship

11

Attributes

- <visibility> / name : <type> <multiplicity> = <default>
- Visibility – public, private, protected, or package
- “/” – attribute is derived from other attributes
- Name – attribute name (starts with lower case)
- Type – primitive type or class
- Multiplicity – number of instances
- Default – value if not otherwise specified

UML

George Blankenship

12

Operations

- <visibility> name (<parameters>) : <return>
- Visibility – public, private, protected, or package
- Name – operation name (starts with lower case)
- Parameter – (similar to attribute structure)
- Return type – primitive type or class

UML George Blankenship 13

Object Diagrams

- Instantiation of class
- Name – <object name> : <class name>
(object name is normally lower case)

UML George Blankenship 14

Relationships

- Dependency (dashed line with arrow next to dependent)
 - A class uses a dependent class to perform an action
- Association (solid line)
 - A class has an associated class to perform an intrinsic operation (created by owner with separate life)
 - Arrows are used to indicate ability to navigate from one class to another class, an 'x' is used to indicate the inability to navigate
- Aggregation (solid line with diamond next to owner)
 - A class owns an associated class if the associated class has some attributes of the class (created by owner with separate life that depends upon owner)
- Composition (solid line with solid diamond next to owner)
 - A class is part of another class (attribute of owner)
- Generalization (solid line with closed arrow pointing to general class)
 - A specific class is also a more general class (inheritance)

UML George Blankenship 15

Activity Diagrams

- Focus on the execution and flow of the behavior of a system
- System behavior defined by a set of system activities
- An activity is diagrammed as a sequence of actions
 - Action is a describable computation
 - Activity diagram depicts the processing flowing from action to action

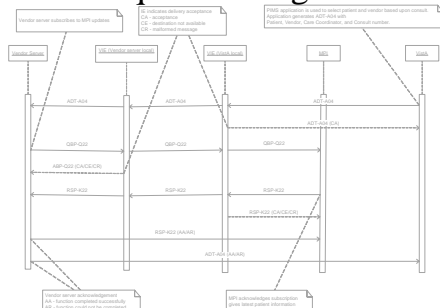
UML George Blankenship 16

Interaction Diagrams

- Captures communication between objects or system components
- Focus is on the message flow and not the processing
- Vertical lines depict individual components
- Horizontal lines depict messages sent from one component to a second component
- Vertical squares depict lifetime of a message in a component

UML George Blankenship 17

Sequence Diagram



UML George Blankenship 18

Use Case Diagrams

- Capture functionality and requirements
 - Use case – element of functionality
 - Actor – invokes the functionality
 - Subject – implements the use case (functionality)
- Focus is on functionality requirement, not processing

UML George Blankenship 19

Use Case

```
graph LR; Clinician((Clinician)) -- "VISA" --- PatientSignUp((Patient sign up)); VendorServer1[Vendor server] -- "Vendor server" --- PatientSignUp; VendorServer2[Vendor server] -- "MPI" --- MPIRegistration((MPI Registration));
```

UML George Blankenship 20
