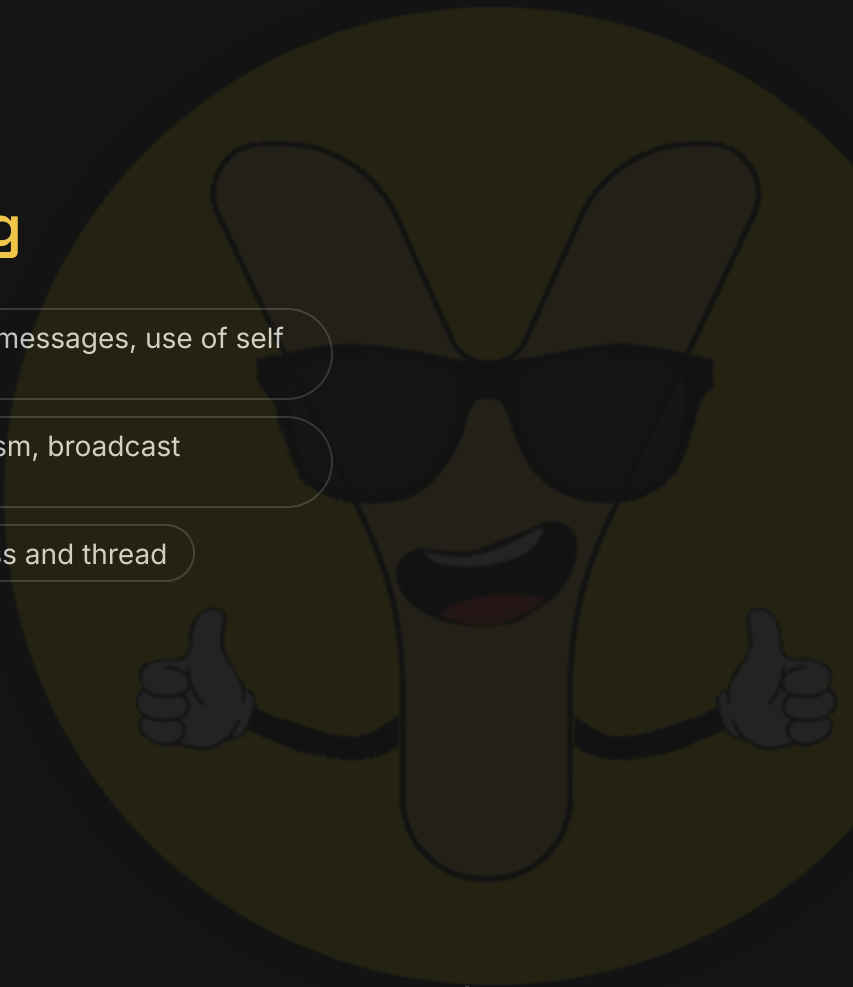
AKTU · 3RD YEAR · SEMESTER 5 · BCS054

Object Oriented System Design with C++

Unit 2 — Basic Behavioural and Architectural Modeling

- Collaboration Diagrams: Terms, Concepts, depicting a message, polymorphism in collaboration Diagrams, iterated messages, use of self in messages
- Sequence Diagrams: Terms, concepts, depicting asynchronous messages with/without priority, call-back mechanism, broadcast messages
- Basic Behavioural Modeling: Use cases, Use case Diagrams
- Activity Diagrams
- State Machine
- Process and thread



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Collaboration Diagrams: Terms and Concepts

Collaboration diagrams (Communication diagrams in UML 2.x) emphasize the structural organization of objects that send and receive messages.

- Object Role
  - Represents an interaction participant depicted as a rectangle: objectName : ClassName.
- Links
  - Solid paths connecting objects derived from class association relationships.
- Sequence Numbering
  - Messages use hierarchical numbers (1, 1.1, 2) to show sequential order and nested calls.
- Directional Arrow
  - Placed along link paths pointing in the direction of message dispatch.

Polymorphism, Iterations, and Self in Collaboration Diagrams

Collaboration diagrams model complex dynamic behaviors through explicit annotations on message links.

- Polymorphism
  - Message dispatched to abstract target line; resolved dynamically to concrete derived implementations.
- Iterated Messages
  - Notated using an asterisk (*), e.g., 1.1* [i := 1..N]: processItem() for loops.
- Use of Self
  - Represented by a reflexive loop link connecting an object rectangle back to itself.
- Guard Conditions
  - Enclosed in square brackets, e.g., 1.2 [balance > 0]: withdraw().

Sequence Diagrams: Terms and Concepts

SEQUENCE DIAGRAM

An interaction diagram that highlights the precise temporal sequence of message exchanges along vertical lifelines.

- Lifeline: Vertical dashed line showing object existence over time.
- Activation Bar: Narrow rectangle indicating active operation execution.
- Synchronous Message: Solid arrowhead line blocking execution until a response is received.
- Return Message: Dashed arrow returning control back to caller.

Advanced Sequence Diagram Messaging Patterns

UML sequence diagrams model complex concurrent and event-driven communication styles.

- Asynchronous Messages
  - Open arrowhead line; caller continues execution without waiting for response.
- Priority Messaging
  - Stereotypes like <<priority=high>> or tagged values attached to message lines.
- Call-Back Mechanism
  - Asynchronous receiver calls back an operation on original caller upon task completion.
- Broadcast Messages
  - Single event sent simultaneously to multiple target lifelines.

Sequence Diagram vs Collaboration Diagram

Feature	Sequence Diagram	Collaboration Diagram
Primary Focus	Time ordering of messages	Structural organization of objects
Visual Layout	Vertical lifelines with time axis	Network of object links
Sequence Tracking	Implicit top-to-bottom order	Explicit sequence numbers required
Lifeline / Lifecycle	Shows explicit creation & destruction	Focuses on structural topology
Modification	Easy to insert new execution steps	Easy to adjust object connections

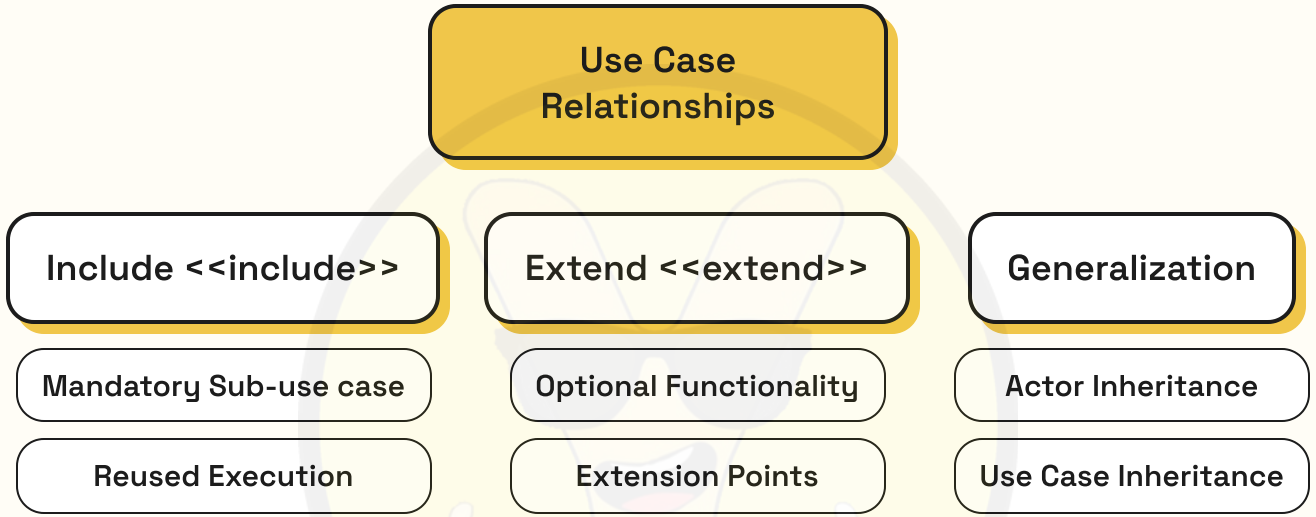
Basic Behavioural Modeling: Use Cases

USE CASE DIAGRAM

A diagram that captures high-level system functionality by illustrating interactions between external actors and internal use cases.

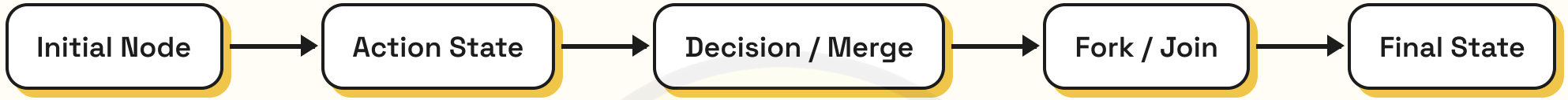
- Actor: External user, trigger, or system interacting with the system.
- Use Case: Ellipse representing a discrete unit of system functionality.
- System Boundary: Rectangle framing internal use cases apart from actors.
- Scope: Captures functional requirements without exposing structural design.

Relationships in Use Case Diagrams



Hierarchical classification of structural relationships connecting actors and use cases in UML.

Activity Diagrams: Operational Workflow



Flow of execution and control passing through activities, conditions, and parallel threads.

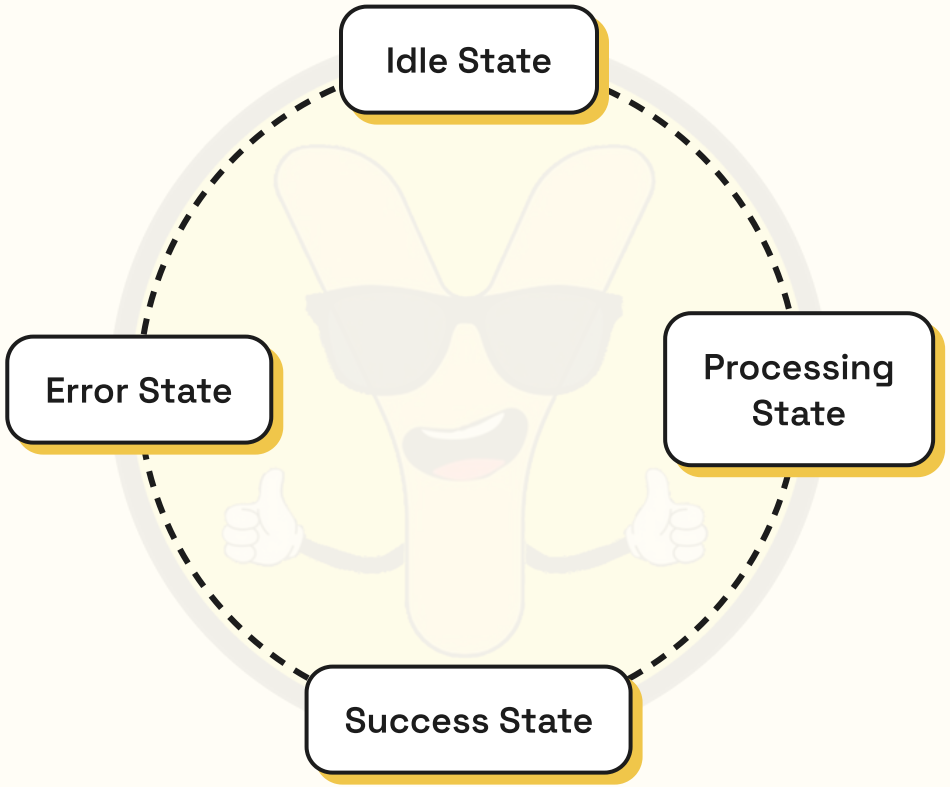
State Machine Diagrams

STATE MACHINE DIAGRAM

A diagram depicting the lifecycle of an object as it transitions through various states in response to event triggers.

- State: Abstraction of an object's condition or activity during its lifetime.
- Transition: Movement from source state to target state triggered by an event.
- Trigger Event: Significant occurrence that can force state transitions.
- Guard Condition: Boolean expression that must be true for transition to fire.

State Machine Lifecycle Example



Cyclic state transitions of an event-driven system component during execution.

Events and Signals in Behavioural Modeling

Events and signals drive dynamic behavior and control flow across object-oriented system models.

- Event
  - Specification of a noteworthy occurrence localized in time and space.
- Signal
  - An explicit, named asynchronous specification sent between objects.
- Call Event
  - Operation invocation on an object, either synchronous or asynchronous.
- Time Event
  - Triggered by elapsed duration, e.g., after(10 milliseconds).
- Change Event
  - Fires when a condition evaluates true, e.g., when(temp > 100C).

Processes vs Threads in Behavioural Modeling

Attribute	Process	Thread
Memory Context	Has isolated address space	Shares address space with parent process
Creation Cost	Heavy overhead for creation	Lightweight creation overhead
Communication	Requires Inter-Process Communication (IPC)	Communicates via shared memory
Failure Scope	Crash does not corrupt other processes	Crash can bring down entire process
UML Representation	Active object with isolated execution boundary	Sub-thread executing inside process context

Timing Diagrams

TIMING DIAGRAM

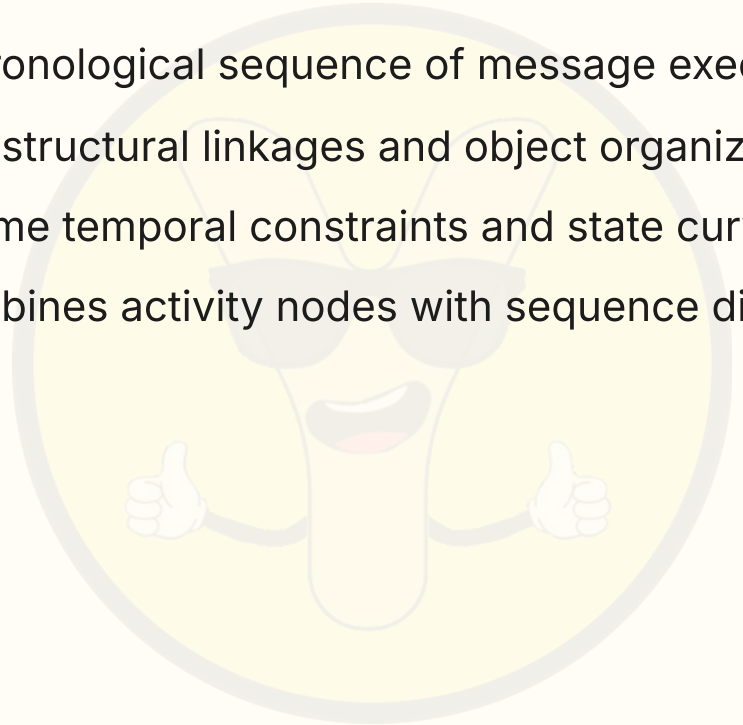
An interaction diagram detailing explicit time duration constraints and continuous state changes of objects on a temporal axis.

- State Lifeline: Continuous line moving between discrete state levels.
- Time Axis: Horizontal axis marked in exact time units (ms, seconds).
- Purpose: Critical for hardware drivers, embedded controllers, and real-time systems.
- Focus: Identifies time delays, setup times, and timing violation conditions.

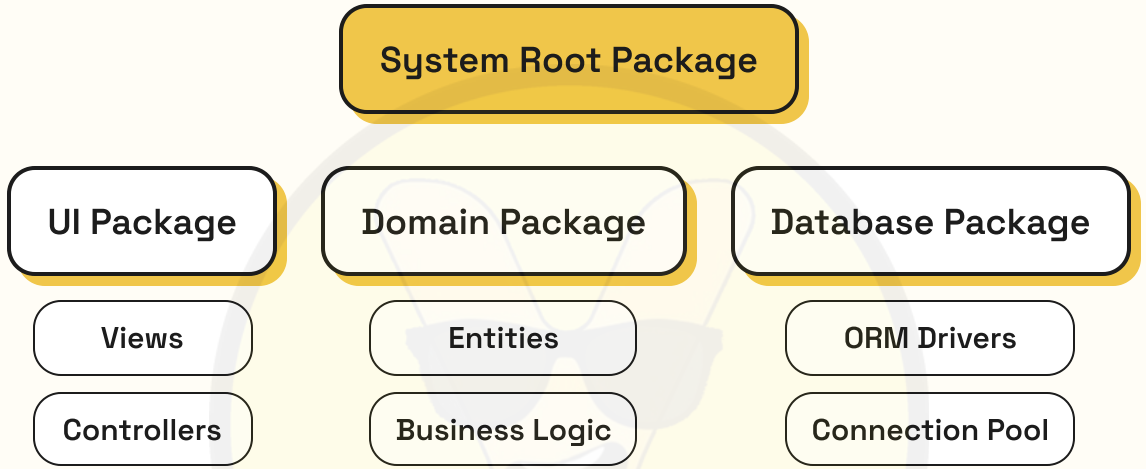
Overview of Interaction Diagrams

UML defines four interaction diagrams to capture different dimensions of system behavior.

- Sequence Diagram: Focuses on chronological sequence of message execution.
- Collaboration Diagram: Focuses on structural linkages and object organization.
- Timing Diagram: Focuses on real-time temporal constraints and state curves.
- Interaction Overview Diagram: Combines activity nodes with sequence diagram fragments.



Package Diagrams: Structural Organization



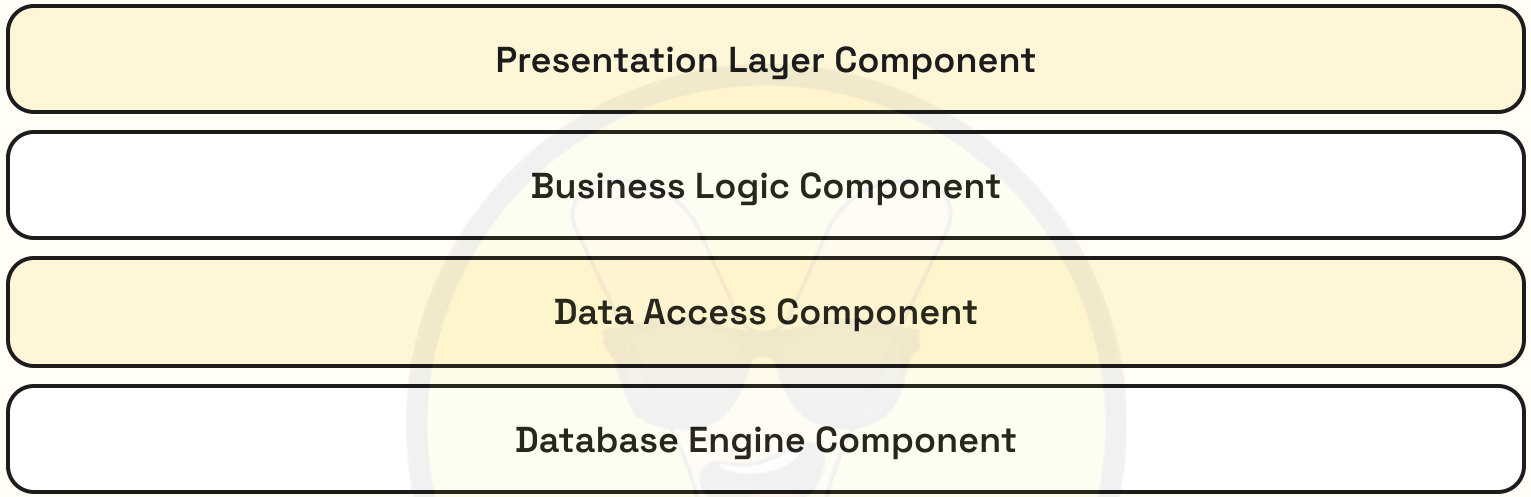
Grouping UML model elements into packages to establish modular system architecture and dependencies.

Architectural Modeling: Components and Deployment

Architectural modeling bridges logical design to physical software components and hardware execution environments.

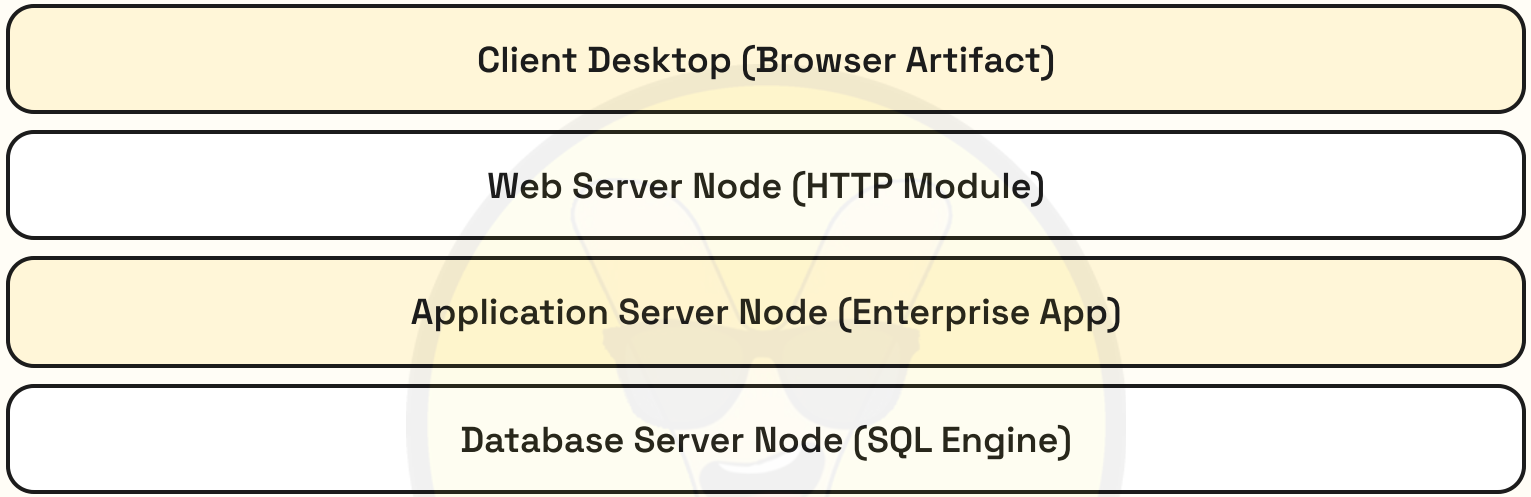
- Software Component
  - Modular, replaceable piece of system code encapsulating internal logic.
- Physical Node
  - Hardware platform or execution environment hosting deployed software.
- Provided Interface
  - Services implemented and offered by a component to other entities.
- Required Interface
  - Services needed by a component to fulfill its responsibilities.
- Artifact
  - Physical execution file like DLL, executable, or database schema.

Component Diagram Layered Architecture



Layered software composition showcasing component dependencies and interfaces.

Deployment Diagram Topology



Physical deployment mapping of software artifacts onto hardware computing nodes.

Component Diagram vs Deployment Diagram

Dimension	Component Diagram	Deployment Diagram
Primary Target	Logical software components	Physical hardware execution nodes
Key Elements	Components, Interfaces, Dependencies	Nodes, Artifacts, Communication Paths
Abstraction Level	Implementation-level software design	Physical deployment architecture
Focus	Packaging and interface contracts	Network layout and runtime allocation
Project Stage	Design and coding phases	System deployment and ops phases

Exam Corner: Likely AKTU Questions

- **Q1** Differentiate between Sequence Diagrams and Collaboration Diagrams with suitable examples.
- **Q2** Explain Use Case Diagrams along with include and extend relationships using an online shopping system example.
- **Q3** Define Activity Diagrams. Explain decision nodes, fork, join, and swimlanes.
- **Q4** Describe State Machine Diagrams, defining states, events, guards, and action execution steps.
- **Q5** Explain Component and Deployment diagrams. How do provided and required interfaces differ?
- **Q6** Compare processes and threads in the context of behavioral and architectural modeling.
- **Q7** Write short notes on Timing Diagrams, Package Diagrams, and Signal/Event handling in UML.

Unit 2 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 3 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

