System Modeling

CSC224

Week 4

Learning Objectives:

- Understand the purpose and importance of system modeling in software engineering.
- Learn different types of system models and their applications.
- Develop basic proficiency in using Unified Modeling Language (UML) for system modeling.

1. What is System Modeling?

- **Definition:**
  System modeling is the process of creating abstract representations of a system to understand, analyze, and communicate its structure and behavior.
- **Purpose of System Modeling:**
- **Visualization:** Helps in visualizing the structure and behavior of the system.
- **Communication:** Provides a common language for stakeholders and developers.
- **Analysis:** Enables evaluation of system design for feasibility, correctness, and efficiency.
- **Documentation:** Acts as a reference for implementation and maintenance.

2. Types of System Models

- **2. Types of System Models**
- System models can be categorized based on their focus. Sommerville identifies the following major types:
- **A. Context Models**
- Represent the boundary of the system and its interactions with external entities (e.g., users, other systems).
- Useful for identifying external actors and inputs/outputs.
- **Example:** A context diagram showing how a library management system interacts with users, book suppliers, and a payment system.
-

2. Types of System Models

- **B. Interaction Models**
- Represent interactions between the system components or between the system and its environment.
- **Types:**
  - **Use Case Diagrams:** Show the high-level functionality of the system and its interaction with actors.
  - **Sequence Diagrams:** Represent the sequence of interactions between objects over time.

2. Types of System Models

- **C. Structural Models**
- Represent the static structure of the system, including its components and their relationships.
- **Examples:**
  - Class diagrams that show classes, attributes, methods, and relationships (e.g., inheritance, association).
- **D. Behavioral Models**
- Represent the dynamic behavior of the system, showing how it responds to external and internal events.
- **Examples:**
  - State diagrams showing states and transitions.
  - Activity diagrams representing workflows.

3. Unified Modeling Language (UML)

- **What is UML?**
- A standardized modeling language widely used in software engineering.
- Provides a set of diagrammatic tools to represent various aspects of a system.
- **UML Diagrams Commonly Used in System Modeling:**
- **Use Case Diagrams:** Capture system functionality from the user's perspective.
- **Class Diagrams:** Represent the static structure of the system.
- **Sequence Diagrams:** Depict the order of interactions over time.
- **State Diagrams:** Model the lifecycle of a system or object.
- **Activity Diagrams:** Show workflows or processes.

4. Key UML Diagrams Explained

- **A. Use Case Diagrams**
- **Purpose:** Illustrate what the system does and who interacts with it.
- **Components:**
  - **Actors:** Entities interacting with the system (users or other systems).
  - **Use Cases:** Functions or services provided by the system.
  - **Relationships:** Connections between actors and use cases.
- **Example:** In a banking system, actors could include customers and tellers, while use cases might include “Withdraw Cash” or “Check Balance.”
-

B. Class Diagrams

- **Purpose:** Represent the structure of the system, focusing on objects and their relationships.
- **Components:**
  - **Classes:** Represent entities in the system.
  - **Attributes:** Properties of the class.
  - **Methods:** Functions associated with the class.
  - **Relationships:** Associations, inheritance, or dependencies between classes.
- **Example:** For a library management system:
- **Class:** Book
  - **Attributes:** Title, Author, ISBN.
- **Methods:** Borrow(), Return().

C. Sequence Diagrams

- **Purpose:** Show how objects interact in a particular sequence.
- **Components:**
  - **Lifelines:** Represent participating objects or actors.
  - **Messages:** Indicate communication between objects.
  - **Activations:** Show periods when an object is active.
- **Example:** For an online shopping system:
- The sequence of placing an order:
  - User selects items.
  - System calculates the total cost.
  - Payment gateway processes the payment.
  - System confirms the order.

D. State Diagrams

- **Purpose:** Model the states of an object and transitions between these states due to events.
- **Components:**
  - **States:** Represent conditions of an object.
  - **Transitions:** Show movement between states triggered by events.
- **Example:** For an elevator control system:
- States: "Idle," "Moving Up," "Moving Down," "Door Open."
- Transitions: Door opens when it reaches a floor; moves up or down when a button is pressed.

E. Activity Diagrams

- **Purpose:** Represent workflows or processes.
- **Components:**
  - **Actions:** Individual steps in a process.
  - **Decision Nodes:** Points where the workflow splits based on conditions.
  - **Swimlanes:** Divide responsibilities among actors or components.
- **Example:** For an e-commerce checkout process:
- User logs in.
- Adds items to the cart.
- Proceeds to payment.
- Confirms order.

5. Benefits of System Modeling

- **Improves Understanding:** Clarifies system functionality and behavior.
- **Enhances Communication:** Provides visual tools for developers and stakeholders.
- **Early Error Detection:** Identifies design flaws before implementation.
- **Supports Maintenance:** Serves as documentation for future updates or debugging.

6. Challenges in System Modeling

- **Over-Modeling:** Creating overly detailed diagrams that become difficult to manage.
- **Tool Dependency:** Using complex tools without understanding the underlying concepts.
- **Stakeholder Engagement:** Ensuring all stakeholders understand and agree on the models.

7. Practical Example: A Library Management System

- **Use Case Diagram:**
- **Actors:** Librarian, Member.
- **Use Cases:** Issue Book, Return Book, Search Catalog.
- **Class Diagram:**
- **Classes:**
  - **Book:** Attributes: Title, ISBN. Methods: Borrow(), Return().
  - **Member:** Attributes: Name, ID. Methods: BorrowBook(), ReturnBook().
- **Sequence Diagram:**
- Member searches for a book → System checks availability → Member borrows the book → System updates inventory.

8. Key Takeaways

- System modeling is essential for understanding and designing complex systems.
- UML provides standardized diagrams for representing structure and behavior.
- Choosing the right model depends on the problem domain and stakeholder needs.

Discussion Questions

- What are the benefits of using UML in system modeling?
- Compare and contrast class diagrams and state diagrams in terms of their purpose.
- Why is it important to involve stakeholders when creating system models?

Practical Activity

- **Objective:** Practice creating UML diagrams.
- **Task:**
  - Choose a system (e.g., an ATM, library system or Hospital Management System).
  - Develop a use case diagram for its core functionalities.
  - Create a sequence diagram for one critical process (e.g., withdrawing cash or placing an order).

Reference

- Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.