



GOVERNMENT OF ANDHRA PRADESH COMMISSIONERATE OF COLLEGIATE EDUCATION



REQUIREMENT MODELING APPROACHES

Software Engineering
Computer Science

Smt. G.Sumalatha MTEch,(PhD)

Govt. Degree College , Salur
Email. Id : sumalathagopathoti@gmail.com

Learn more at <http://ccelms.ap.gov.in>

Objectives

- Describe requirement modelling approaches.
- Define various elements of requirement model.
- Describe scenario-based models.
- Describe class-based models.

Requirement Modeling Approaches

- Requirements modelling can be done with two approaches
 - Structured analysis
 - Object-oriented analysis.
- Structured analysis considers data and the processes that transform the data.
 - Data objects are represented with their attributes and relationships.
 - Processes transform data as it flows through the system.
- object-oriented analysis focuses on the definition of classes and how they collaborate with one another to satisfy customer requirements.

Elements of Requirement Model

- Each element of the requirements model represents the problem from a different point of view.
- Analysis modelling focus on each of these modelling elements.

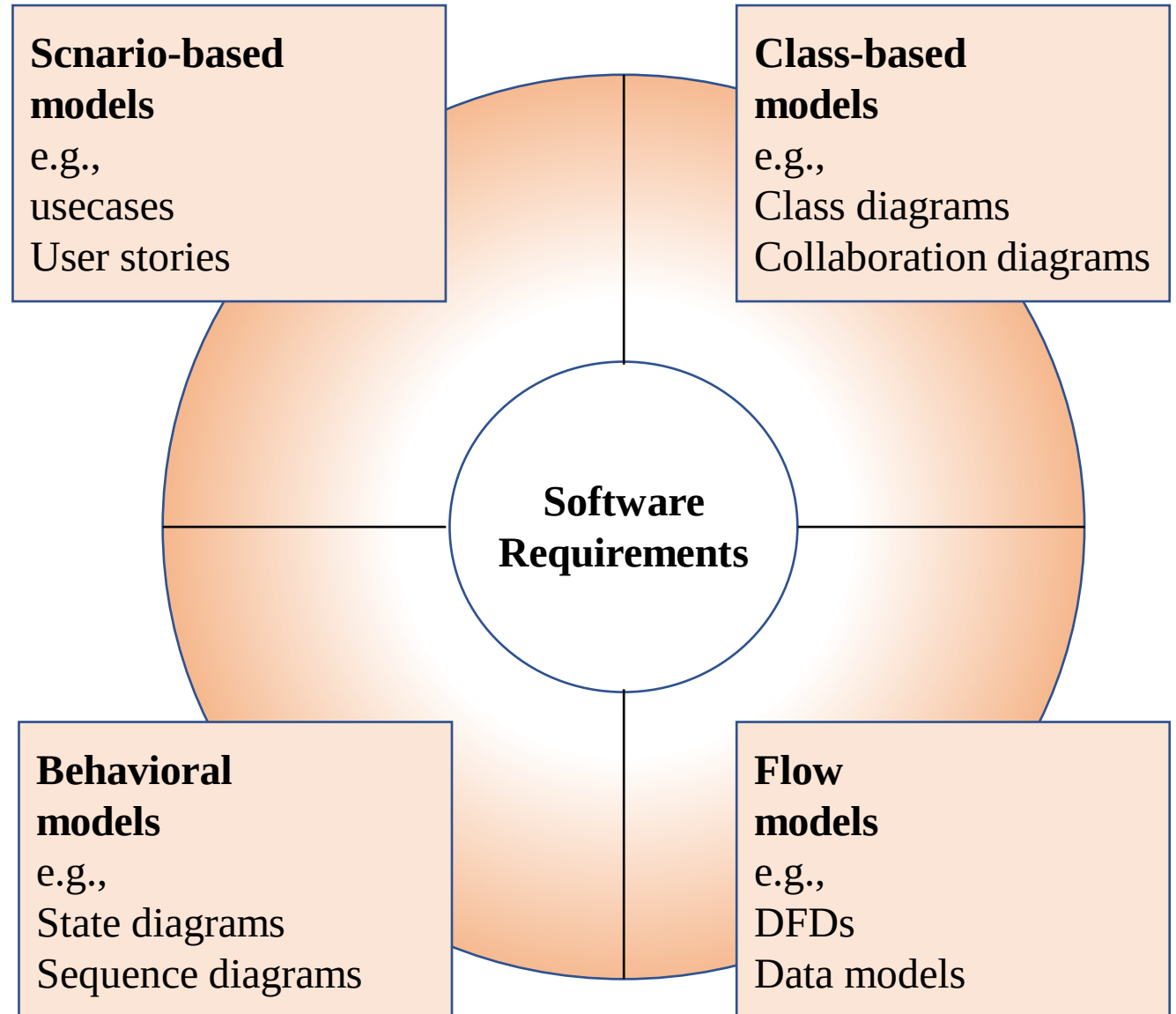


Figure: Elements of requirement model

Scenario-based Modelling

- Requirements modelling with UML (Unified Modelling Language) begins with the creation of scenarios in the form of use cases, activity diagrams.
- Preliminary usecase diagram
 - Creating preliminary usecases.
 - Refining preliminary usecases.
 - Writing formal usecases

Creating Preliminary Usecases

- Usecases are developed based on functions and activities performed by user.
- The ATM system, Customer actor performs the following functions.
 - Login
 - Withdraw money.
 - Deposit money.
 - Balance enquiry.
 - Transfer money.
 - Transaction record
- Usecases are written first in an informal narrative fashion. If more formality is required, the same usecase is rewritten using a structured format.

Creating Preliminary Usecases

- Structured format results in sequence of action performed by customer

For example:

Usecase: withdraw money, sequence of actions are as follows

1. The customer inserts ATM card.
2. The customer selects the language.
3. The customer enters 4-digit ATM pin.
4. The customer selects the type of transaction.
5. The customer selects the type of account.
6. The customer enters the withdrawal amount.
7. The customer collects the cash.
8. The customer takes the printout if needed.
9. The customer goes for another transaction.

Refining Preliminary Usecases

- Each step in the primary scenario is evaluated, and look for secondary scenario.
- For example, consider steps 3 and 6 in structured format
 - 3.The customer enters 4-digit ATM pin.
 - 6.The customer enters the withdrawal amount.
- Is it possible that the actor will encounter some error condition at 3rd step? “Customer could not enter into the system”. This error condition becomes a secondary scenario.
- Is it possible that the actor will encounter some other behaviour at this point? If yes, the customer again inserts the card and then enter valid four-digit pin number.
- Each of the situations described above is characterized as a use-case exception.
- In some cases, an exception will precipitate the development of another use case

Writing Formal Usecase

- Usecase are written formally, by specifying its goal, prerequisites, trigger condition, scenarios and exceptions.
- In many cases, there is no need to create a graphical representation of a usage scenario.
- However, diagrammatic representation can facilitate understanding, particularly when the scenario is complex.

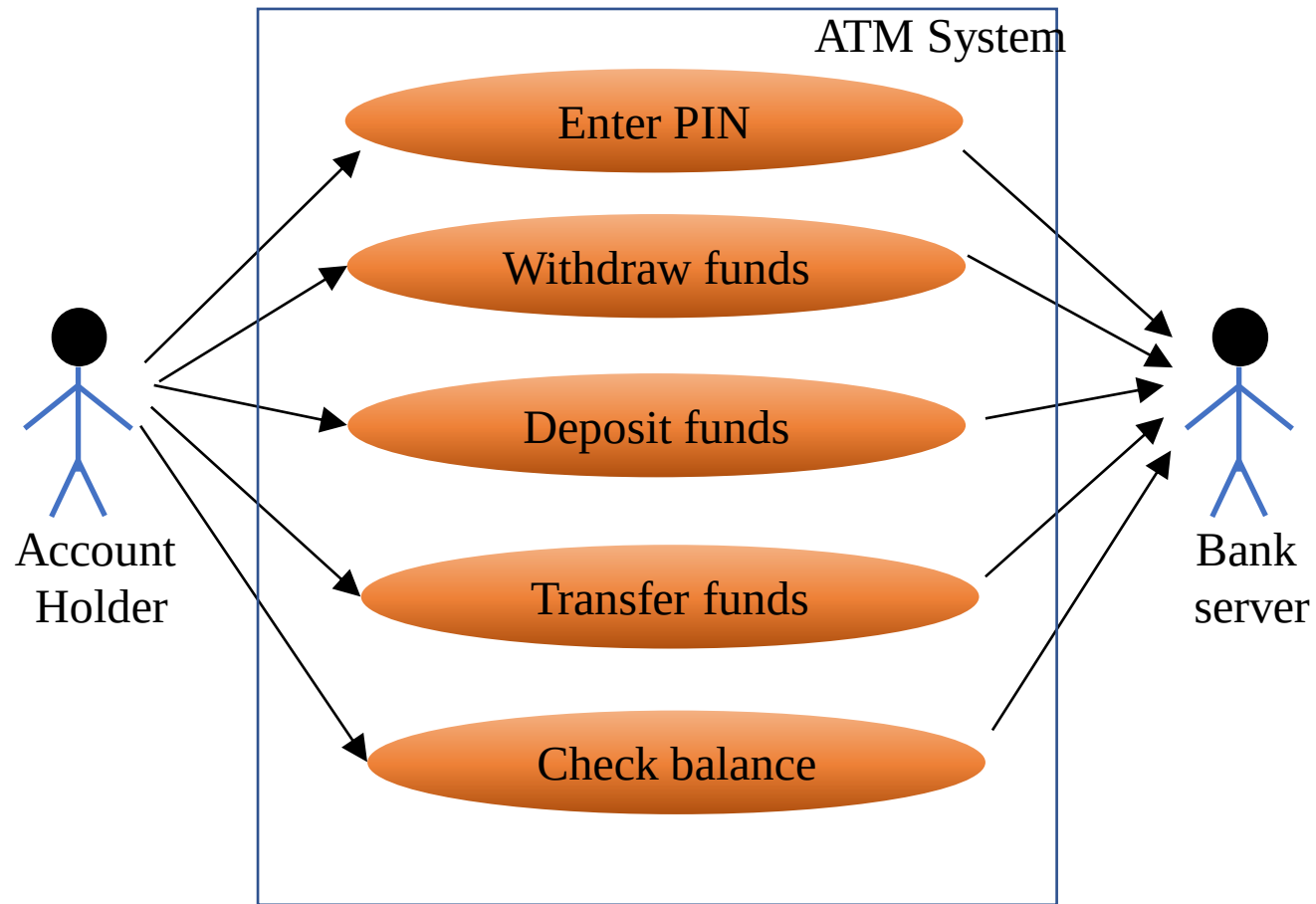


Figure: Preliminary usecase diagram for ATM system

Class-based Modelling

- The elements of a class-based model include
 - Classes and objects
 - Attributes
 - Operations
 - Class responsibility-collaborator (CRC) models
 - Collaboration diagrams

Class Diagram

- Steps to follow while building class diagrams
 - Identify the classes by examining usecases.
 - Classes are identified by looking at the nouns such as external entities, things, occurrence of events, roles, organizational units, places, and structures.
 - Develop meaningful set of attributes of the identified classes.
 - Define operations that
 - Manipulate data.
 - Enquiry about state of the object.
 - Perform computations.
 - Monitor an object.

Class Diagram

- Consider the customer class defined for ATM system.
- Class is modelled with a rectangle having three parts.
- First part represents the name of the class.
- Second part represents the attributes of customer class.
- Third part represents the operations performed by customer class.

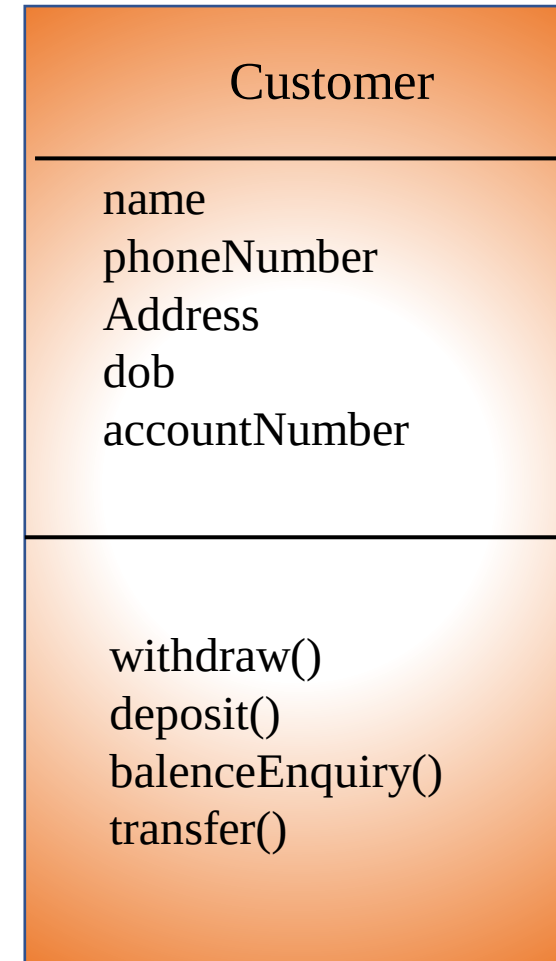


Figure: Class diagram for customer class in ATM System

Class Responsibility Collaborator Model

- CRC models helps us to identify and organize classes of that are relevant to product requirements.
- This model makes use of actual or virtual index cards, that are used to develop organised representation of classes.
- Index cards consist of responsibilities and collaborators.
- Responsibilities are any thing that class knows or does something.
- Collaborators are the those classes that provide information to fulfil a responsibility.

Class: Withdrawal Transaction	
Description	
Responsibilities	Collaborators
Get specifics from customer	ATM, Session
Send to bank	BANK
Dispense cash, issue receipt, notify bank	ATM

Figure: CRC model index card

Collaboration Diagram

- It is also called as communication diagram.
- Illustrate relationships and interactions among various classes.
- These are used to model dynamic behaviour of usecases.

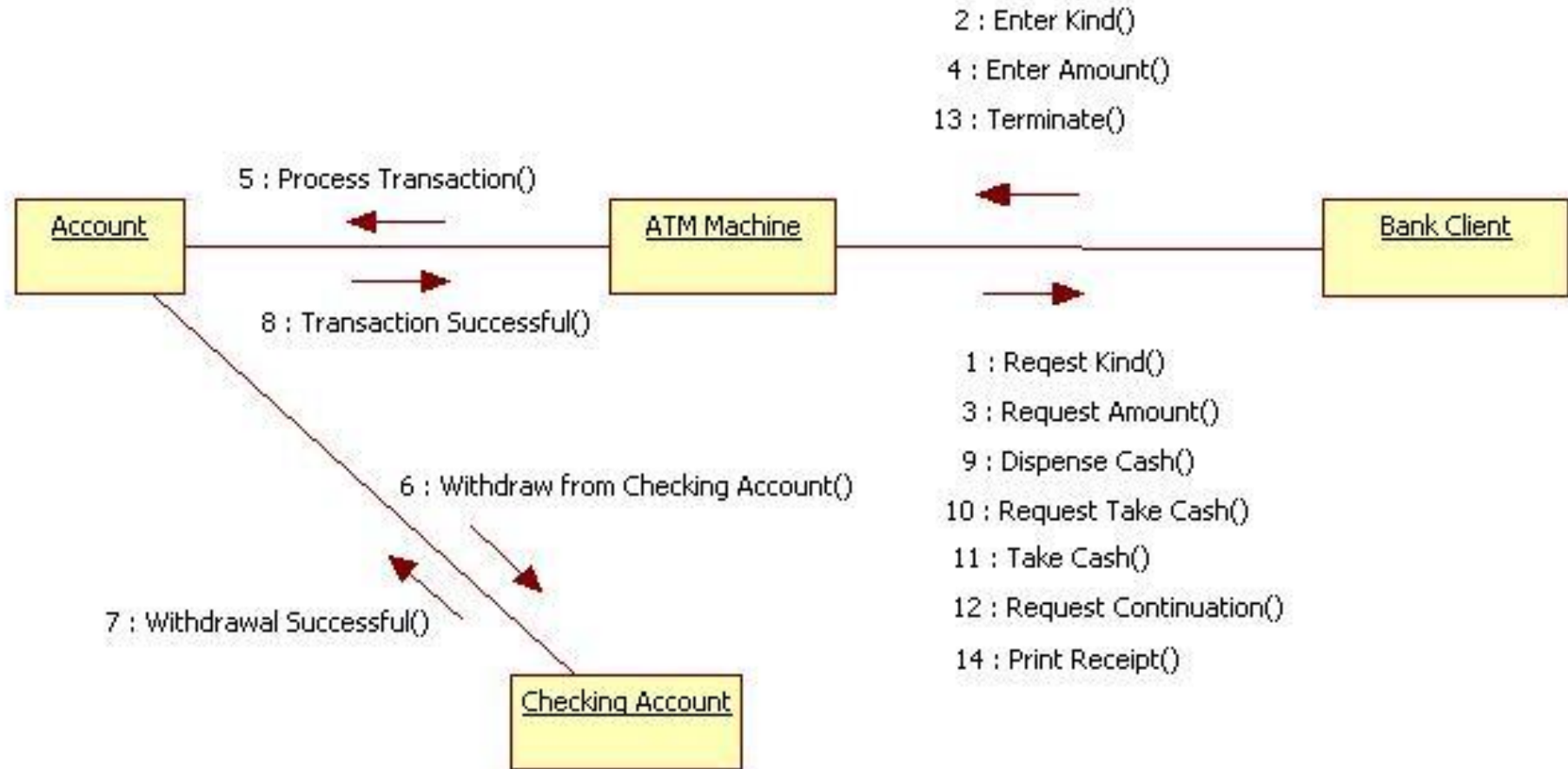


Figure: Collaboration Diagram for ATM System

<https://images.app.goo.gl/nYZa8qwgeocXtLnA7>

Summary

- Two approaches of requirement model.
- Scenario-based models
 - Creating preliminary usecases.
 - Refining preliminary usecases.
 - Writing formal use case.
- Class based models
 - Classes
 - Attributes
 - Relationships
 - CRC models
 - Collaboration diagrams

References

Text Books:

1. Roger Pressman S., “Software Engineering: A Practitioner's Approach”, 7th Edition, McGraw Hill, 2010.
2. Sommerville, “Software Engineering”, Eighth Edition, Pearson Education, 2007

Web Links:

1. <https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-355j-software-engineering-concepts-fall-2005/lecture-notes/cnotes2.pdf>
2. <https://drive.google.com/file/d/1K327JTpX4P1DPjlNurjKznzZKoBk2Qot/view>
3. https://cdn.shopify.com/s/files/1/0457/4009/7694/files/software_engineering_pdf_pressman_7th_edition.pdf

Thank You



Smt. G. Sumalatha MTech, (PhD),
email ID:sumalathagopathoti@gmail.com