Object Oriented UML Modeling for ATM Systems

Rajni Pamnani, Pramila Chawan, Satish Salunkhe
Department of computer technology,
VJTI University, Mumbai

Abstract

The Object-Oriented Modeling assists the programmer to address the complexity of a problem domain by considering the problem not as a set of functions that can be performed but primarily as a set of related, interacting Objects. This article is based on the approach of Object Oriented modeling through Unified Modeling Language (UML) for an ATM system in Banking sector. The main aim of this article is to provide a flexible and faithful environment for a customer, who wants to do online banking transactions. After successful authentication, the customer can withdraw the desire amount (within the prescribed limit) from the ATM machine or can transfer amount to other account. User can also change the pin code. The ATM has solved many problems and now customer can have money transaction 24/7. In this article UML Class, Interaction diagram, Activity diagram & Use Case diagram are also designed for the ATM System.

Key Words: UML dynamic modeling, UML static modeling, Use case modelling

1. Introduction

The Unified Modeling Language (UML) is a very dominant modeling graphical language for specifying, constructing and documenting the artifacts of software system. UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development UML is a collection of best engineering practices that have successful in the modeling for a design

of a huge and complex systems. Modeling is very important for readability and reuse of the systems. UML offers a set of notations and rules for using the same. The main task of the UML is to create a simple, well documented and easy to understand software model for the people. The UML modeling consists of nine diagrams to model a software system & these diagrams are Use case Diagram, Class Diagram, Object Diagram, State Diagram, Activity Diagram, Sequence Diagram, Collaboration Diagram, Component Diagram & Deployment diagram.

Nowadays knowledge based system is most popular and needed in every sector like medical sector, banking sector, engineering sector and traveling sector etc. In the Banking sector, knowledge means simply having the knowledge from expert. Knowledge modeling is the major activity used to understand the problem and sending the solution rapidly.

2. Features of ATM Systems

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

The ATM must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rupees 100. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
3. A customer must be able to make a transfer of money between any two accounts linked to the card.
4. A customer must be able to make a balance inquiry of any account linked to the card.

3. Object-Oriented Analysis

Object-oriented analysis looks at the problem domain, with the aim of producing a conceptual model of the information that exists in the area being analyzed. Analysis models do not consider any implementation constraints or how the system is to be built. The identified objects reflect entities and operations that are associated with the problem to be solved.

3.1.UML static modeling for ATM systems

This part describes the way that system should look. It analyses the structure and substructure of the modeled system based on objects, attributes, operations and relationships.

3.1.1. Use case modeling for ATM systems

In an automated teller machine shown in Figure 1, the Bank Customer can withdraw cash from an account, transfer funds between accounts, or deposit funds to an account. These correspond to specific goals that the actor has in using system.

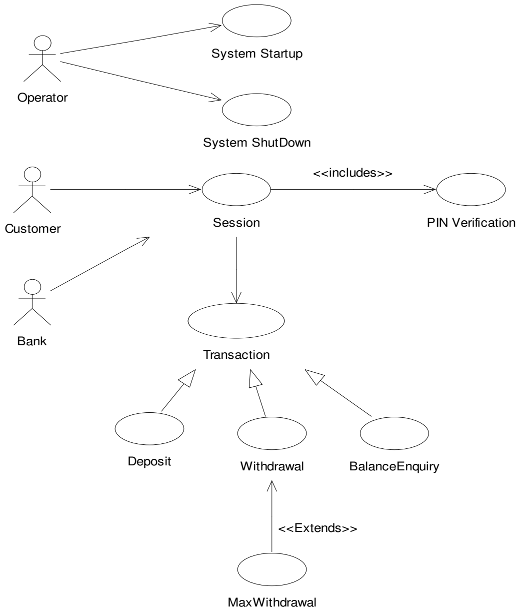

Figure 1 : Use Case Diagram for ATM System

Withdrawal Transaction Use Case

A withdrawal transaction asks the customer to choose a type of account to withdraw from. The system verifies that it has sufficient money on hand to satisfy the request before sending the transaction to the bank.

Pre-conditions: The customer must have a valid ATM card and PIN.
Post-conditions: The customer receives the cash amount that he wanted to withdraw, with a receipt, if indicated. The customer's account balance is updated in the system.
Specifications:
Primary Actor: Customer
Stakeholders:

- <u>Customer</u>: Wants quick, accurate withdrawal of cash
- <u>Bank</u>: Wants to give fast, accurate and reliable service to the customer
- <u>Bank that owns ATM</u>: (If not the same as the customer's bank): Wants to charge the customer the correct amount of surcharge on the withdrawal.
- <u>ATM Administrator</u>: Wants to ensure that the ATM always has sufficient cash for a predicted number of withdrawals per day.

Normal flow of events:

1. The customer inserts ATM card into the ATM machine and enters PIN.
2. The system validates the ATM card and PIN .
3. The customer selects the 'Cash Withdrawal' option from the Options Menu.
4. The system prompts the customer to enter the amount of cash that he or she wants to withdraw.
5. The customer enters a cash amount and selects the 'Submit' option on the Cash Withdrawal Screen.
6. The system validates the amount entered; checks account balance and that the machine has enough cash for the transaction, and asks the customer if he or she wants a receipt for the transaction.
7. The customer selects 'Yes' on the Receipts Screen.
8. The system ejects the ATM card, provides the cash, prints the receipt and updates the account balance of the customer in the system.

Alternate flow of events:

1. The customer has entered invalid PIN- The system prompts the customer to enter a valid PIN.
2. If ATM card is not compatible-The system rejects the ATM card and displays an error message.
3. The customer has entered an amount that exceeds the withdrawal limit.
4. The system rejects the transaction & displays an error message.

Deposit Transaction Use Case

A deposit transaction asks the customer to choose a type of account to deposit to. If the transaction is approved, the machine accepts an envelope from the customer containing cash and/or checks before it issues a receipt.

Pre-conditions: The customer must have a valid ATM card and PIN.
Post-conditions: The customer receives the receipt of cash amount/cheque that he has deposited. The customer's account balance is updated in the system.
Specifications:
Primary Actor: Customer
Stakeholders:

- <u>Customer</u>: Wants quick, accurate withdrawal of cash
- <u>Bank</u>: Wants to give fast, accurate and reliable service to the customer
- <u>ATM Administrator</u>: Wants to ensure that the ATM always has sufficient cash for a predicted number of withdrawals per day.

3.1.2. Class Diagram

Shown below is the class diagram for the ATM system. The basic structure of the class diagram arises from the responsibilities and relationships discovered when doing the CRC cards and Interaction Diagrams. (If a class uses another class as a collaborator, or sends a message to an object of that class during an Interaction, then there must either be an association linking objects of those classes, or linking the "sending" class to an object which provides access to an object of the "receiving" class.)

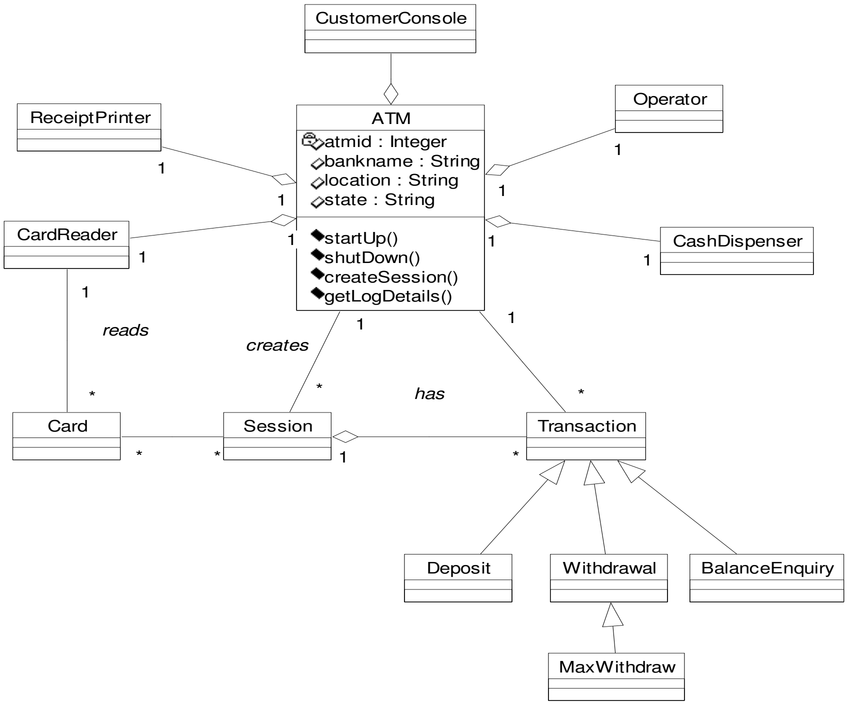

Figure 2 : Class Diagram for ATM System

Attributes and Methods used in each class are described below:

Class Name	Attributes	Methods
ATM	atmid:integer, bankname:string, state:string,,location:string	performStartup(),performShutdown() createSession(),getLogDetails()
CustomerConsole	atm:ATM	displayMenu(), displayMessage() readPIN()
CardReader	atm:ATM	readCard(), ejectCard()
CashDispenser	Initialcash:integer, totalcash:integer	setInitialCash(), checkMaxCash() dispenseCash()
Operator	atm:ATM	switchOn(), switchOff(), checkATMStatus()
Session	atm:ATM,pin:integer,state:string	createSession(),verifyPIN()
Transaction	atm:ATM, session: Session, pin:integer, balance:integer	createTransaction()
Deposit	amount:integer, bankname:string, pin:integer	getDetails(),performDeposit()
Withdrawal	amount:integer, bankname:string, pin:integer, balance:integer	getDetails(),,performWithdrawal()
BalanceEnquiry	pin:integer, bankname:string	getDetails(),performEnquiry()

3.2. Dynamic Modeling for ATM System

This section describes the way that the system should work. It analyses the system behavior, including sequence and collaboration diagrams, activity diagram, and state diagram.

3.2.1 State Diagrams

The transition between difference states is represented as an arrow between states, and a condition of that transition occurring may be added between square braced. This condition is called a guard.

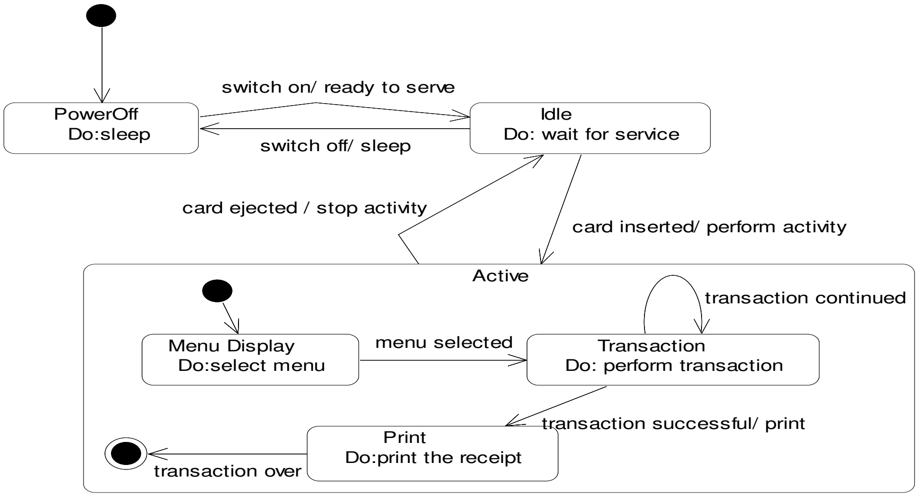

Figure 3 : State Diagram for ATM System

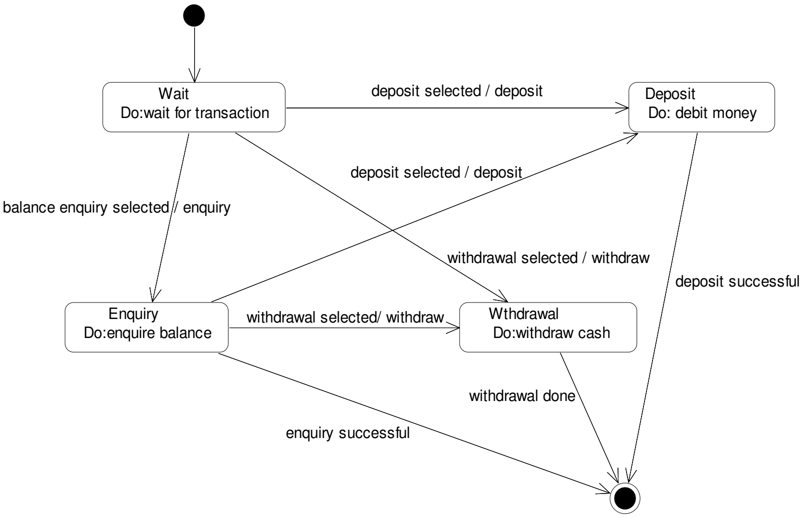

Figure 4 : State Diagram for Transaction

3.2.2. Interaction Diagrams

UML defines two types of Interaction Diagram: the Sequence Diagram and the Collaboration Diagram. Interactions between **objects** are represented by interaction diagrams – both **sequence** and **collaboration** diagrams. An example of a collaboration diagram is shown below. Objects are drawn as rectangles and the lines between them indicate links – a link is an instance of an association. The **order of the messages** along the links between the objects is indicated by the number at the head of the message.

Sequence diagram shows the relationship between classes arranged in a time sequence. Within a sequence diagram an object is shown in a box at the top. The sequence diagram of the above UML class model is designed and used for online money transactions. The communication between two objects represented by an arrow & the message on that arrow, the vertical lines show the life of the objects.

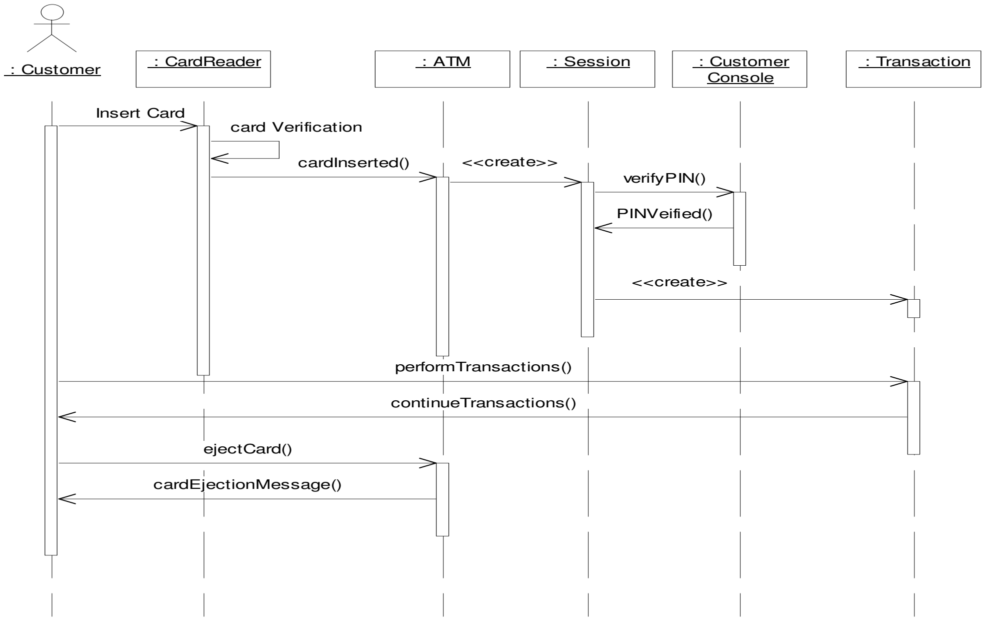

Figure 5 : Sequence Diagram for ATM Session

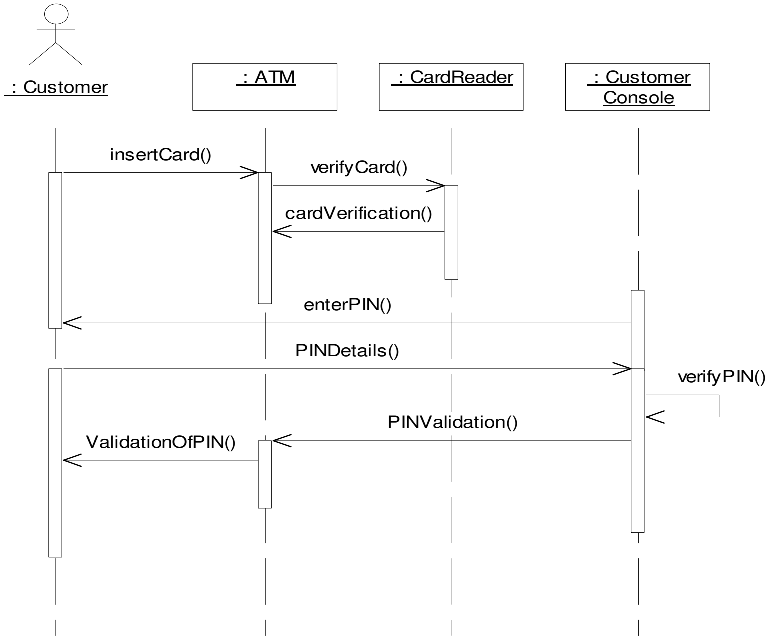

Figure 6 : Sequence Diagram for PIN verification

4. Object Oriented design

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design.

4.1. Component Diagram

A component diagram in the Unified Modeling Language, depicts how components are wired together to form larger components and or software systems.

Components diagrams can be used to illustrate the structure of arbitrarily complex systems.

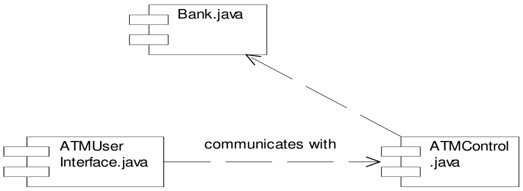

Figure 7 : Component Diagram for ATM System

4.2. Deployment Diagram

A deployment diagram in the Unified Modeling Language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show the allocation of Artifacts to Nodes according to the Deployments defined between them. Deployment of an artifact to a node is indicated by placing the artifact inside the node.

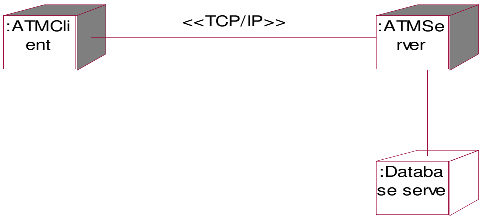

Figure 8 : Deployment Diagram for ATM

5. Conclusions

This article has investigated system requirements, use case diagrams and the UML model for the ATM System. Such diagrams were designed to support the system designers and developers, to help customers in observing a software system from various perspectives, and to improve their understanding of cohesion and abstraction in designing a system or software. The requirements produced are sufficiently detailed to form the basis for the development of an information system using both the practical analysis of use cases and the conceptual model for UML. The analysis of business and user needs, system requirements, and use cases are essential steps for building a system based on Unified Modeling Language.

From the above work it is concluded that the UML modeling is a powerful language used to design for the software research problems. In this article complete modeling is done for ATM system which is efficient & useful for the software developer to convert the above model through Object Oriented language. UML could be adopted for knowledge modeling as well. While UML in its current state has its limitations, it is an extensible language and thus can be used to support the knowledge modeling activity through the profiles mechanism.

6. References

[1] G., Booch, J.Rumbaugh, and I Jacobson,.,"The Unified Modelling Language User Guide", Addison Wesley, Reading, MA,(1999).

[2] Booch, G., "Object-Oriented Analysis and Design with Applications", second Edition. Addison-Wesley (1994).

[3] M.Blaha, and, J., Rumbaugh "Object-Oriented Modeling and Design with UML", Second Edition, Upper Saddle River, New Jersey, Prentice Hall,(2005).

[4] http://jdjua.com/uml.htm

[5] Lan Anh Tran , "A UML Model of the Client Tracking System", 18th Australasian Conference on Information systems

[6]"ATM System", Wikipedia, www.wikipedia.org/wiki/Automated_teller_machine