Journal of Computations & Modelling, vol.6, no.1, 2016, 151-168
ISSN: 1792-7625 (print), 1792-8850 (online)
Scienpress Ltd, 2016

Object Orientation Validation of ATM Using FSM

Ram Pratap[1], Taskeen Zaidi[2] and Nityanand[3]

Abstract

ATM is a computerized machine that provides bank customers to gain access to their accounts using magnetic encoded plastic card and code number. ATM is a real-time system that is very complex in design and application. The ATM provides 24*7 hours service, customer can withdraw cash up to certain limit with in a day any time. ATM is very beneficial for travellers, as they need not carry large amount of cash with them. The customer also withdraws brand new currency notes from ATM card. ATM also provides privacy to customer transactions. The Unified Modeling Language that is an object-oriented approach is used for modeling purposes know days by researchers and scientists. UML represents various static and dynamic views of system. The present paper represents class, sequence, activity and state model of ATM system. Valid and invalid Test cases are also designed for validation of models.

Keywords: Software engineering; UML; FSM; ATM

[1] SRMU. E-mail: rampratapmca11@gmail.com
[2] SRMU. E-mail: taskeenzaidi867@gmail.com
[3] SRMU. E-mail: nityananddwivedi29@gmail.com

Article Info: *Received* : December 22, 2015. *Revised* : February 18, 2016.
Published online : March 1, 2016.

1 Introduction

Little work has been done to employ UML for modeling of parallel and distributed applications. The Pllana and Fahringer[13,14] has develop various building blocks to control and enhance flow of data. Mapping of applications on process topologies is also illustrated. Brief introduction on UML versions, models, concepts, views and environment as well as role of UML in software development on various platforms are well described in [2]. UML was implemented by OMG [10,11,12] in 1996 which aims to maintain the specifications by publishing documents and tools related to UML.Well known object oriented language UML was used[21]to construct model for execution of task in distributed manner, for validation of UML model state diagram is converted to Finite state machine and test cases are generated for proposed model. An ATM network for distributed database system was designed by hossain [8]. In 1967,Barclays introduced first ATM machine in Hendson branch, London and ATM emerges as smaller, faster and easier medium [3]. A trend in transaction infrastructure and two-factor authentication with respect to ATM was evaluated in [19] and also performance of ATM machine was evaluated. Fingerprint authentication in ATM machine was proposed by [8] for providing security to the ATMs. It is composed of three-tier structure and it emphasized on reducing ATM fraud. Biometric verification at ATM user interface overview is explained by[7].PII (personal identification image) method for ATM security is proposed in [17].Wang etal.[20] has proposed a fingerprint orientation model on 2D Fourier expansion and also its applications to singular point detection fingerprint indexing. Automated fingerprint recognition system and its key challenges including research scope is well illustrated in [1]. Object oriented modeling using UML is well described in [9]. Alternative PIN entry method referred as cognitive trapdoor games which make harder for criminal to obtain PINs even he get entire input and output of a PIN entry procedure was introduced by [16]. Concept of graphical password scheme was introduced by [22]. Shoulder surfing resistance with eye-gaze entry in recalling graphical password was explained by [6].

Shaukat [18] described a state based approach for integration testing in UML models. Debashish [4] has used UML for constructing, visualizing and modeling. A spacial model of ATM networks for transaction fees between banking and non banking services was introduced by [5].

2 Background

2.1 ATM

An electronic device that enables the customer to perform online transactions without involving cashier, clerk and bank teller. The customer make cash withdrawal, check account balances, transfer money as well as purchase prepaid mobile phone credit by using ATM card. ATM is one of the best ways for official exchange rates for foreign travellers. Typical PIN based ATM has following processes:

- Insert ATM card to establish interface.
- Enter your PIN and press the execution key for the system to match the PIN, if PIN doesn't matches, then user access is denied to the next stage and he or she is requested to repeat the operation with the correct PIN for a fixed retries.
- If the PIN matches then the transaction interface is displayed and user has to select the account type (savings, current or credit).
- The withdrawal amount is then selected from the display options. The system then checks if the available balance is sufficient for servicing the request. The machine will dispense the money in case of sufficient balance while insufficient balance causes the machine to decline the transaction with the display of an explanatory message such as "insufficient fund".
- The machine prompts user if a new transaction is to be performed. If the response

is ‘Yes’, the transaction interface is again displayed and the first five operations (a-e) are repeated. If ‘No’ the transaction is terminated and card ejected.

3 UML Static and Dynamic Diagrams

3.1 Use Case diagram

The use case diagram represents the flow of functionalities between user of the system and system itself. The ATM system composed of various conditions for cash withdraw.

Preconditions:

The customer should posses ATM card.

Network connection of bank system working properly and must be active.

Some cash must be in ATM systems to dispense.

Withdraw service must be working properly.

Events:

Following events occurs during cash withdraw.

ATM card insert

Checked for authenticate user using pins

Select withdrawal by account type

Select account type

Select amount of cash to be withdrawn

Confirm withdrawn

Remove card from ATM system

Dispense cash

Use case ended

Scenarios:

(i) Success scenario:

Cash withdraws

Cash withdraws with receipt printed

(ii) Failure scenario:

Unauthenticated user login

Unauthorized /invalid card

Insufficient cash in ATM system

ATM Card jam

Cash dispensing, failure

Non-responding of bank system

Post conditions:

ATM system dispensed cash and returned card to customer.

The ATM system returned the card to the customer, but not supplied the amount of cash registered withdrawn from customer account.

The ATM system returned the card to the customer and no withdrawal registered to customer's account.

Use case diagram(see Figure 1) represents the interaction between the user and system and system itself. Mentioned above user enters card if card is valid machine prompt to enter pin if pin is valid then user select service and account. User makes transaction if the limit withdrawal limit reached, insufficient balance in the account or no money in ATM machine then transaction failed. But if transaction is successful then user removed the card and print receipt is generated and user account is updated.

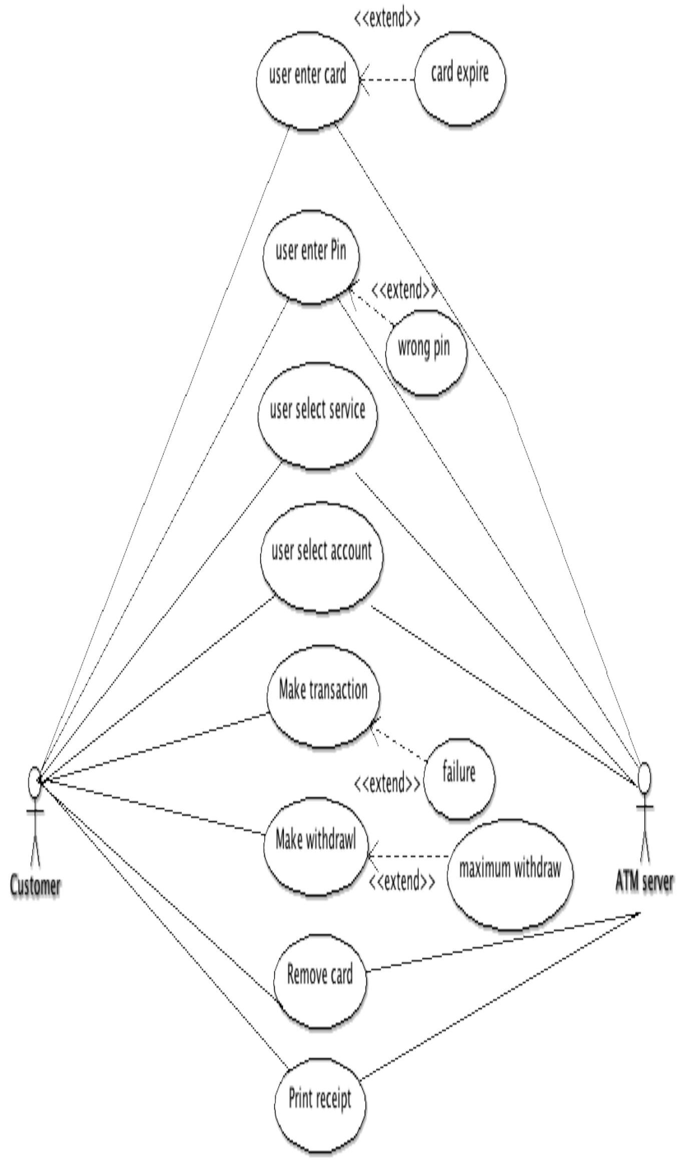

Figure 1: Use Case diagram for ATM.

3.2 Class Diagram

Discarding the unnecessary and incorrect classes according to following rules identifies right classes:

a. **Redundant classes**

If two classes are expressing the same information we have to select a more descriptive name. For example, customer and user are redundant in ATM systems so the customer is retained.

b. **Irrelevant class**

If a class has little or nothing to do with the problem, then it should be eliminated; in ATM cost is outside the scope of ATM transaction software.

c. **Vague classes**

A class should be specific. Some tentative classes may have ill-defined boundaries are to be broad in scope so we have to eliminate these. In ATM system, security provision, record keeping provision and banking network are vague classes.

d. **Attributes**

Names that primarily describe the individual objects should be restated as attributes as name, age, weight and address are usually used as attributes.

e. **Operations**

If a name describes an operation that applied to objects and not manipulated in its own right, then it is not a class.

(f) **Roles**

The name of a class must reflect its intrinsic nature and not a role that it plays in association.

(g) **Implementation constructs**

The classes having constructs extraneous to the real world should be eliminated from the analysis model.

Class diagram (see Figure 2) represents the structure in general or static view of UML model as defined in Object Management Group (OMG). Shows customer has an account with a bank and bank issued an ATM card to the user. Customer can have saving or current account with the bank. ATM card can be used for online transaction as well as for withdrawal of money from ATM machine.

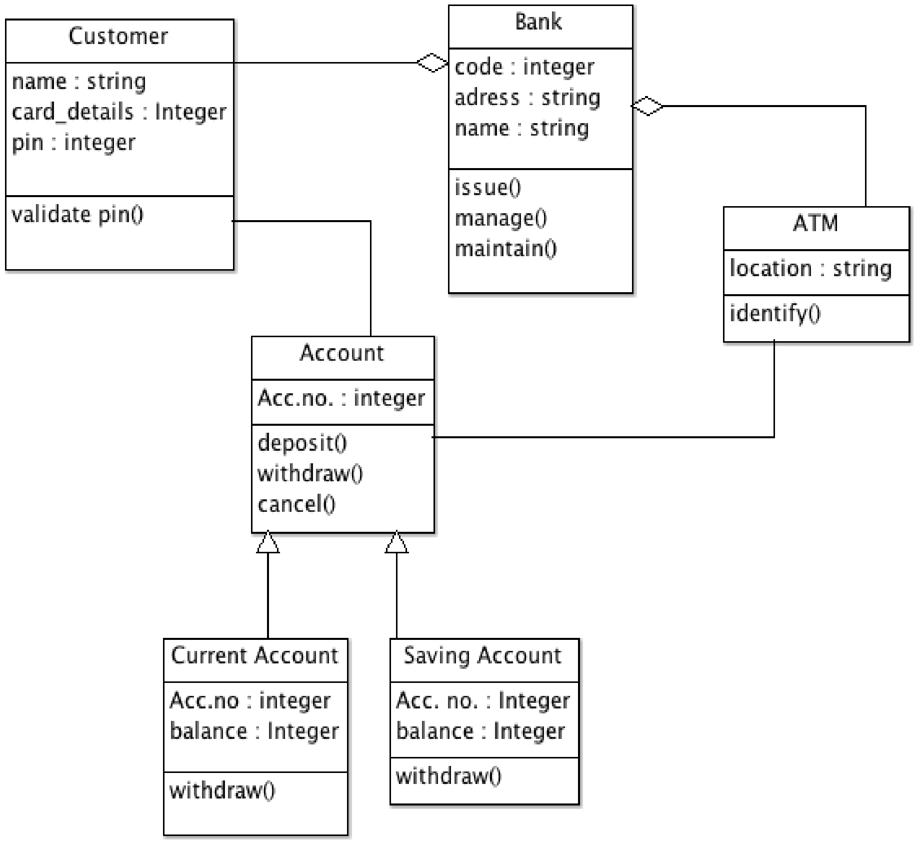

Figure 2: UML Class Diagram for ATM system.

3.3 Sequence Diagram

A series of diagrams represents the dynamic behavior of an object-oriented system. It is also used to model the way a use case is realized through a sequence of messages between objects. Sequence diagram represents how objects interact with each other. Emphasizes time ordering of messages. Modeled simple sequential flow, branching, iteration, recursion and concurrency.

Sequence diagram (see Figure 3) that shows dynamic behavior for interaction between customer and ATM server. Customer insert card and enter pin, pin is verified by ATM machine then customer selects operation to perform, customer enters amount to withdraw, if cash dispense then print receipt is generated.

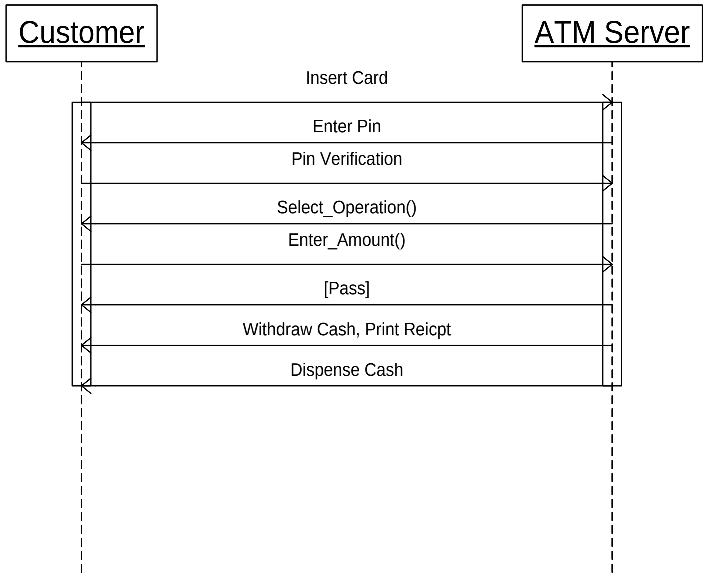

Figure 3: UML Sequence Diagram for ATM system

3.4 Activity Diagram

Activity diagram(see Figure 4) are same as flow charts and data-flow diagrams from structured development. Activity diagrams represent the dynamic behavior of a system. Process flows in the ATM system are captured in the
Activity diagram. Activity diagram shows the workflow of a system by modeling the flow of control from activity to activity.

Activity diagram which shows user insert ATM card, if the card is correct then machine prompt to enter pin other wise termination occur.

If pin is correct then user selects account type and if pin is correct then termination occur after 3-5 retries. After entering pin user enters amount to be withdraw. Again if user has sufficient fund in account then transaction will be successful and machine dispensed cash but if there is insufficient fund then transaction failed and process terminated.

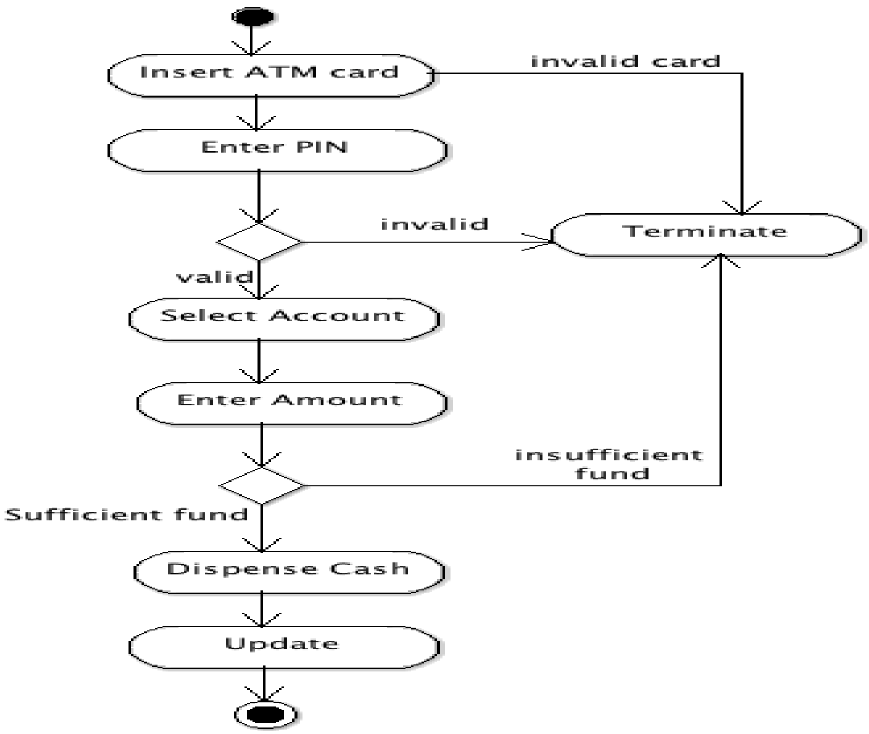

Figure 4: UML Activity Diagram for ATM system

3.5 State Diagram

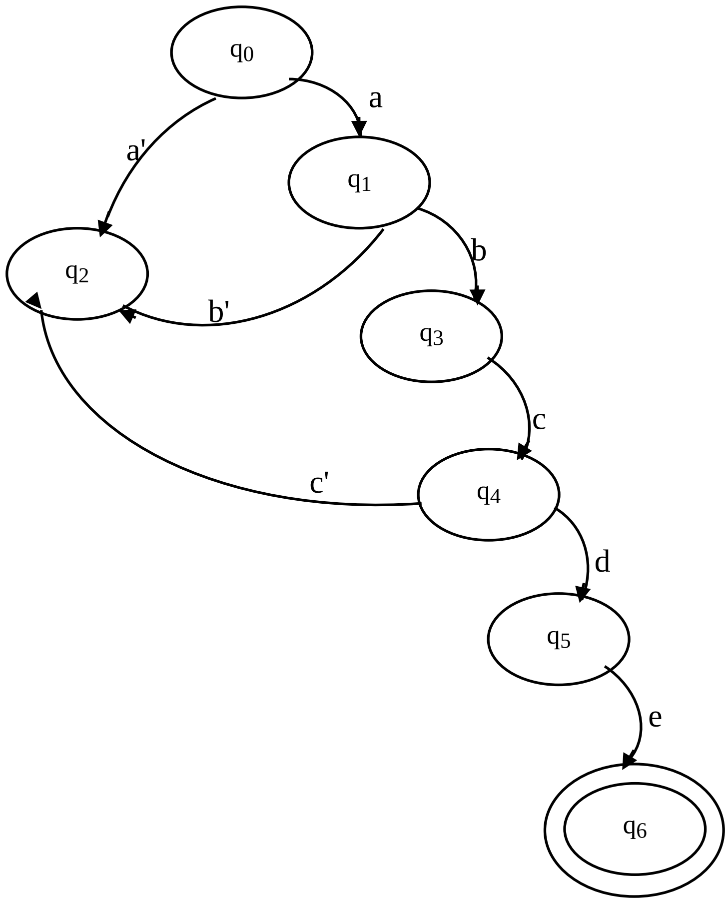

Figure 5: UML state Diagram for ATM system

Table 1: Events for State transition.

a	Enter card
a'	Invalid card
b	Enter pin

b'	Wrong pin entered
c	Select account
d	Sufficient funds
c'	Insufficient funds
e	Remove card

Table 2: Representation of Activity for state transition

q_0	User enter card
q_1	User enter pin
q_2	terminate
q_3	User select account
q_4	Make transaction
q_5	Dispense cash
q_6	update

4 Test Cases for ATM System Using FSM

The activities are in (see figure 5) represented as q_0, q_1, q_2, q_3, q_4, q_5, and q_6 with number of events as represented in above see table and 2; total events are a, a’, b, b’, c, d and e and f. On a’ represents complement of an event and similar definition is also considered for other events. From the above figure, the relationships between activity and event in the form of transition functions are described below along with equivalent grammar:

$\delta(q_0,a) = q_1 \Rightarrow q_0 \rightarrow aq_1$
$\delta(q_0,a') = q_2 \Rightarrow q_0 \rightarrow a'q_2$
$\delta(q_1,b) = q_3 \Rightarrow q_1 \rightarrow bq_3$
$\delta(q_1,b') = q_2 \Rightarrow q_1 \rightarrow b'q_2$
$\delta(q_3,c) = q_4 \Rightarrow q_3 \rightarrow cq_4$
$\delta(q_3,c') = q_2 \Rightarrow q_3 \rightarrow c'q_2$
$\delta(q_4,d) = q_5 \quad q_4 \rightarrow dq_5$

Test case 1:Invalid card

$q_0 \rightarrow aq_1$

$q_0 \rightarrow a'q_2$

By replacing terminals on RHS of production rules,we can get,

$q_0 \rightarrow aa'q_2$

Test case 2:Invalid password

$q_0 \rightarrow aq_1$

$q_1 \rightarrow bq_3$

$q_3 \rightarrow b'q_2$

By replacing terminals on RHS of production rules, we can get,

$q_0 \rightarrow abb'q_2$

Test Case 3:Insufficient funds:

$q_0 \rightarrow aq_1$

$q_1 \rightarrow bq_3$

$q_3 \rightarrow cq_4$

$q_3 \rightarrow c'q_2$

By replacing terminals on RHS of production rules,we can get

$q_0 \rightarrow abcc'q_2$

Test case 4:Transaction successful and card removed

$$q_0 \rightarrow aq_1$$

$$q_1 \rightarrow bq_3$$

$$q_3 \rightarrow cq_4$$

$$q_4 \rightarrow dq_5$$

- $q_5 \rightarrow eq_6$

-

By replacing terminals on RHS of production rules, we can get

$$q_0 \rightarrow abcdefq_6$$

5 Testing using Use Case Scenario for ATM System

Scenario is the real life situation in which the user interacts with system and failures occur.

We developed a matrix see Table 3, start by identifying data elements required to execute use cases. For each scenario we identified test case that causes some error condition and then valid (V) for successful or invalid (I) for alternate flow are used to represent conditions and "IM"(impractical) indicates that this condition is not applicable to the test case.

Table 3: Testing using Use case scenarios.

TC ID#	**Scenario / Condition**	**valid card**	**PIN**	**Account #**	**Amount Entered (or chosen)**	**Amount in Account**	**Amount in ATM**	**Expected Result**
TC 01	Scenario 1 transaction successful	V	V	V	V	V	V	Successful cash withdrawal.
TC 02	Scenario 2 - ATM unable to dispense money	V	V	V	V	V	I	ATM not working due network problem, use case ended.
TC 03	Scenario 3 - Insufficient funds in ATM	V	V	V	V	V	I	Warning message,Enter Amount
TC 04	Scenario 4 - Incorrect PIN (> 1 left)	V	I	V	IM	V	V	Warning message, return to, Enter PIN
TC 05	Scenario 4 - Incorrect PIN (= 1 try left)	V	I	V	IM	V	V	Warning message, return to, Enter PIN

TC 06	Scenario 4 - Incorrect PIN (= 0 tries left)	V	I	V	IM	V	V	Warning message, card blocked, end of use case
TC 07	Scenario 5 Maximum withdrawal limit reach	V	V	V	V	V	V	Message maximum withdrawal for today reach end of use case.
TC 08	Scenario 6 Transaction Unsuccessful	v	v	v	v	I	V	Message retries entering with less money.

6 Conclusion

The ultimate objective deals with modelling of ATM system functioning, which is used to derive various valid and invalid test cases for ATM system working. Use cases are also discussed that shows the valid, invalid and impractical condition for ATM system functioning. The expected results generated by use case scenario testing shows various warning messages if the test case is invalid.

In the presented model, object-oriented approach is followed and model is platform independent.

This work can be extended to design a biometric strategy based on fingerprint matching and face recognition to measure ATM security in India E-banking sector.

References

[1] Anil KJ, Jianjiang F, Karthik N., Fingerprint Matching. *IEEE Computer Society*, (2010), 36-44.

[2] Booch, G, Rumbaugh, J., and Jacobson, I., *The Unified Modeling Language User Guide*, Addison Wesley, Reading, MA, 1999.

[3] Das SS, Jhunu D., Designing a Biometric Strategy (Fingerprint) Measure for Enhancing ATM Security in Indian E-Banking System, *International Journal of Information and Communication Technology Research*, (2011), 197-203.

[4] Debasish K, and Debasis S., A novel approach to generate test cases from UML activity diagrams. *Journal of Object Technology*, **8**(3), (2009), 65–83.

[5] E.W. Croft and B.J. Spencer, Fees and surcharging in automatic teller machine networks: non-bank atm providers versus large banks, 2003.

[6] Forget, A., Chiasson, S., and Biddle R, Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. *In Proceedings of the 28th ACM International Conference on Human Factors in Computing Systems* - CHI 2010, Atlanta, Georgia, USA, (April 2010), ACM, Apr. 2010.

[7] Lynne C, Antonella DA, Johnson G., Usability and Biometric Verification at the ATM Interface, Ft. Lauderdale, Florida, USA, **5**, (2003), 153-160.

[8] Mali P, Salunke S, Mane R. and Khatavkar P., Multilevel ATM Security Based On Two Factor Biometrics, *IJERT*, **1**, (2012), 8.

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

[10] OMG, Unified Modeling Language Specification, available online via http://www.omg.org, 2001.

[11] OMG, Unified Modeling Language Specification, available online via http://www.omg.org, 2001.

[12] OMG, Unified Modeling Language (UML)–Version 1.5, OMG document formal/2003-3-01, (2003), Needham MA.

[13] Pllana, S., and Fahringer, T., *On Customizing the UML for Modeling*

Performance Oriented Applications, In UML, Model Engineering Concepts and Tools, Springer-Verlag, Dresden, Germany, 2002.

[14] Pllana, S., and Fahringer, T., *UML Based Modeling of Performance Oriented Parallel and Distributed Applications*, Winter Simulation Conference, 2002.

[15] R Hossain, et al., Designing an Atm Network Based on Distributed Database Systems, no date.

[16] Roth, K. Richter, and R. Freidinger. A pin-entry method resilient against shoulder surfing, In CCS '04: *Proceedings of the 11th ACM conference on Computer and communications security*, NY, USA, (2004), 236–245.

[17] Santhi B, Kumar RK, Novel Hybrid Technology in ATM Security Using Biometrics. *Journal of Theoretical and Applied Information Technology*, **37**, (2012), 217-223.

[18] Shaukat A, Lionel C. Briand, Muhammad J., Hajra A., Muhammad Z., Iqbal, Z., and Nadeem. A., A state-based approach to integration testing based on UML models. *Inf. Softw. Technol*., **49**(11-12), (2007), 1087-1106.

[19] Sunday AO, Evaluating the Performance of Two-Factor Authentication Solution in the Banking Sector, *International Journal of Computer Science* Issues, **9**, (2012), 457-462.

[20] Wang Y, Hu J, Phillips D., A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing, *IEEE Trans Pattern Anal Mach* Intell, **29**, (2007), 573-585.

[21] Taskeen Z., and Vipin, S., Modeling and Validation of Execution of Tasks on High Speed Network under Distributed Environment, *ACM SIGSOFT* (ISSN No.: 0163-5948), USA, **38**(3), (16 May, 2013).

[22] Wiedenbeck, S., Waters, J., Sobrado, L. and Birget. J.-C., Design and evaluation of a shoulder-surfing resistant graphical password scheme, In AVI '06, *Proceedings of the working conference on Advanced visual interfaces*, New York, NY, USA, (2006), 177–184.