UML MODELING FOR ONLINE BANKING SYSTEM USING OBJECT ORIENTED DATABASES

Dr. Harsh Dev
Professor
PSIT, Kanpur, India
doctorharshdev@rediffmail.com

Suman Kumar Mishra
Research Scholar
Bhagwant University,
Ajmer, Rajasthan, India
sumansunil_532@yahoo.com

Ajay Pratap
Research Scholar
B.B.A. University,
Lucknow, UP, India
pratap Aj@yahoo.co.in

ABSTRACT

In the present Scenario, the competitive environment in banking sector, paradigm is changing rapidly. Various new things can be seen in banking sector, because it is adopting new concepts and technology to improve their business & to win the satisfaction of their customers. The present paper discusses the role of Unified Modeling Language (UML) in the Object Oriented Database Management System (OODBMS) in banking sector. The Object Oriented Database technology is considered as the fifth generation database technology. OODBMS can be considered as a combination of two technologies, Database Management System and Object Oriented System. A case study of Online Banking System has been taken to explain how various accounts and transactions can be classified. Object oriented Data Modeling technique has been used to store the data and it is represented with the help of UML Use-Case diagram, Class Diagram, Sequence diagram and State diagram.

Keywords: UML, OODBMS, Use-Case diagram, Class Diagram, Sequence diagram and State diagram.

1. THE NEED OF OODBMS

The advancement in technology, especially internet, intranet and information technology has led to new techniques for doing business transactions in banking sector. Earlier Relational Database Management Systems were backbone to deal with central server and business transactions of Centralized banking system (CBS). They were also supporting Disaster Recovery system (DRS) who were using parallel queries and recovery of data. Today’s object–oriented database management system (OODBMS) support complex queries and handle the large amount of the data. In given table below comparison has been given between Relational Data Model and Object Oriented Data Model (ODM) [10]:

630
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Difficulties</th>
<th>Relational Data Model</th>
<th>OODM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Non-homogenous data</td>
<td>Homogenous tuples in a relation</td>
<td>Group similar objects as a class.</td>
</tr>
<tr>
<td>2.</td>
<td>Variable – length and long – strings</td>
<td>Numbers, short – string and fixed – length records.</td>
<td>Support New data types (Class)</td>
</tr>
<tr>
<td>3.</td>
<td>Complex Objects</td>
<td>Hierarchical data structure</td>
<td>Support Retrieval as a single unit</td>
</tr>
<tr>
<td>4.</td>
<td>Version control</td>
<td>Do not support old version</td>
<td>Support previous and next version attributes</td>
</tr>
<tr>
<td>5.</td>
<td>Schema evaluation</td>
<td>Do not support efficient mechanism</td>
<td>Changes to the definition inside a class and changes to the structure of the class lattice.</td>
</tr>
<tr>
<td>6.</td>
<td>Equivalent objects</td>
<td>Do not provide mechanisms to Model semantics of equivalent representations.</td>
<td>Different representations of the same object</td>
</tr>
<tr>
<td>7.</td>
<td>Long transactions</td>
<td>Returns to the most recent state possible - not just to the state at the last committed transaction.</td>
<td>Support for versions can be integrated with transaction management to simplify concurrent access and recovery</td>
</tr>
</tbody>
</table>

An object-oriented database is a persistent and sharable repository and management of an object-oriented database is a collection of objects defined by an object-oriented data model, that is, objects that capture the semantics of objects supported in object-oriented programming [11]. Object-Oriented Database System = Object-Oriented Concepts + Database Features Supported in Conventional Database Systems + Memory-Resident Object Management + Features Specific to CAx Applications + Object-Oriented Database Programming Languages.

2. INTRODUCTION TO UML

UML design tools and notations are used to graphically depict object-oriented analysis and design models. UML is a language for specifying, visualizing and constructing the artifacts of software systems and business modeling. In recent years, the Unified Modeling Language (UML) has emerged as the defector standard for the representation of software engineering diagrams (Rumbaugh et al.1999) [2]. The UML class diagram contains classes, interfaces, collaborations, and dependencies, associations and interface relationships. We are creating a unified modeling language (UML) structure by specifying the use case, classes, and activities in the application.
In UML modeling technique, we have three related but different viewpoints: Class Model, State Model and Interaction Model. The class model represents the static structural “data” aspects of a system. The state model represents the temporal, behavioral, “control” aspect of system and the interaction model represents the collaboration of individual objects, interaction aspect of the system. Class diagram, State diagram, use-case diagram, sequence and activity diagrams are used in these three modeling techniques.

3. CASE STUDY: ONLINE BANKING SYSTEM

In the present Scenario, the competitive environment in banking sector, paradigm is changing rapidly. Various new things can be seen in banking sector, because it is adopting new concepts and technology to improve their business & to win the satisfaction of their customers. The customers and Employees are the driving forces who are responsible for the survival of banking industries in the competitive environment and it largely depends on providing quality services and delivery of quality service in banking sector is very important and profitable. Object oriented database and UML concepts plays a pivotal role in improving the quality of services in the banking sector. The various services in banking sector are ECS (Electronic Clearing Services), RTGS (Real time gross settlement), EFT (Electronic Fund Transfer), ATM (Automated Teller Machine), Debit & Credit cards and many more value added services. These services use object oriented database concepts to return more effective and efficient output [14]. The analysis of the banking system operations with relational backend has been done for last two-years. This analysis was conducted using banking manuals and then the Branch computerization system. We will implement online banking system with the object oriented databases modeled with the use of UML. Information on entities and their attributes and relationships are fed into the object oriented database management system (OODBMS).

a. Use Case Diagram:
A use case diagram consists of actors and the use cases. The actors are the direct external user of the system. In other words we can say that it is an object or the set of objects that communicates directly with the system but that is not the part of system. In below figure 1, Use Case Diagram of Online Banking System, two actors employees and customers. The use case is a piece of functionality that the system can provide by interacting with the actors. The diagram involves a sequence of messages between the use cases and the actors.
b. Class Diagram:
In class diagram, we handle the things that are used in the system. The classes can be related to each other in number of ways, like they can be associated, dependent, specialized or packaged. A system can have a number of class diagrams because not all classes participate in a single class diagram. In figure: 2, class diagram involves the customer, employees, transaction, account and loan and one subclass diagram branch.
Figure 2: Class Diagram for Online Banking System

The customer opens an account in the bank and the concerning employee who handles all the activities related to account checking like eligibility of customer for account and if the Customer
is not eligible then inform to the Customer for not fulfilling the requirement of the account, only eligible Customer can get the application form for opening the account. After getting the application form from the Customer, the application is checked by the Employee for completeness, verification of signature, and also the employee confirms the latest address which was filled by the Customer and the employee creates a separate file for each account having different unique account number. After this the application is approved by the senior employee. The entry of the applications should be done by using the interface software provided at the branch. The Customer’s data should be uploaded to the online through the same software and stored in the Online banking system database. The Customer’s data should be uploaded to the online through the same software and stored in the Online Banking System database. The details of the account can be viewed via online by entering the account number.

c. Sequence Diagram:
Sequence diagram tells how objects interact with each other i.e. how messages are being send and received between objects. This diagram has two axes: The vertical axis shows time and the horizontal axis shows the objects. Figure: 3 presents the sequence diagram for the employee operations of Banking System.

![Sequence Diagram for Online Banking System](image)

**Figure 3: Sequence Diagram for Online Banking System**

As shown above this diagram shows that how an Employee handle the transactions in the Online Banking system. The sequence diagram shows the complete issuing process of operations. The five main objects are represented at the top of diagram. The communications between two objects are shown by an arrow along with communication message. The vertical line shows the
life line of the object. The employee will be provided the account opening form and other relevant material for opening the account. Employee checks the form to verify the eligibility criteria for opening the account. If the form is complete in all aspects then employee sends the form to the Branch. The Branch issues the account number and sends the information to the customer. The main purpose of this diagram is to represent the execution of the account opening system and to check whether it is working properly or not.

d. State Diagram:
State diagram is used to describe the behavior of a system. State diagrams describe all of the possible states of an object as events occur. Each diagram usually represents objects of a single class and tracks the different states of its objects through the system [13]. In below figure 4 state diagrams show the track of banking operations.

As shown above, state diagram shows the possible states of customer opening account process to perform the transactions in the online banking system. The customer firstly find the location of bank branch and fill the application form of opening an account, and then employee checks all the eligibility criteria of opening an account, if the customer is eligible, then account is created. Now the customer can perform any operation by mentioning the type of the client id and password. Client id and password verify with help of online banking database and start the operation.

4. EXPERIMENTAL RESULTS
We have implemented the online banking system using object oriented concepts and UML. The system uses an Intel Core i3 – 330 M Processor with 3 GB DDR3 RAM. Windows 2003 or higher version of operating system can be used and SQL server 2000 or higher version can be used as a DBMS. Rational Rose Software is used for the designing of UML diagrams. The proposed system is implemented and the experimental results are shown below.
Figure 5: Screenshot of Result

Sample Query 1:

Select account_name, receipt_no from tbfd where account_no = "000600450022588";

The result of the above sample query is shown in the figure as account_no ‘000600450022588’.

Sample Query 2:

Select account_name, interest_rate from tbfd where deposit_date = "09/01/2012" Having account_no = "000600450031068";

The result of the above sample query is shown in the figure as account_no ‘000600450031068’.

5. CONCLUSION AND FUTURE WORK

From the above analysis and discussion, we observed that Object-Oriented Database Management Systems (OODBMS) were limited in the past because of a lack of standardization. But today we have standard OODBMS such as Objectivity and db4o through which it is easy to deal with the complex and huge amount of banking data. In this paper we have discussed how the traditional banking system is converted into innovative banking system. The present work can be easily extended by designing of the data cubes for the many major fields shown in the sample database and one can easily extract the information in more efficient manner about the
customers and their account numbers along with the date of deposit and date of renewal. The same work can also be extended for the Mobile banking, Retail Banking and large multimedia databases.

ACKNOWLEDGEMENTS
Authors are grateful to Dr. D. A. Jain, Research Director, Bhagwant University, Ajmer, Rajasthan for providing the excellent facility in the computing lab of Bhagwant University, Ajmer, Rajasthan, India. Thanks are also due to University Grant Commission, India for financial support.

REFERENCE:
3. A Case Study of an Object Database Implementation, Billy Gibson, APM. 1608.00.02 Draft 10th October 1995.
6. Diego Calvanese, Maurizio Lenzerini. Making Object-Oriented Schemas More Expressive. in Proc. of PODS’94,