ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

A Novel Object Oriented Perspective Design for Automated BookBank Management System

P.Lavanya[1], R.Meena[2], R.Vijayalakshmi[3], Prof.M.Sowmiya[4] and Prof.S.Balamurugan[5]

Department of IT, Kalaignar Karunanidhi Institute of Technology, Coimbatore, TamilNadu, India[1,2,3,4,5]

ABSTRACT: The paper proposes a set of diagrams based on object orientation principle which describes the functionalities from various perspectives. The diagrams depicts the functional, behavioral and structural aspects of automated passport system. The proposed modeling methodology is based on the principle of object orientation, which allows describing both software and functionalities explicitly. Furthermore, it is illustrated how the well-known object-oriented specification language unified modeling language can be adopted, to provided an adequate formalization of its semantics, to describe structural and behavioral aspects of smart phone database management system related to both logical and physical parts. It is needed to implement the software on the basis of Object Oriented Model developed. Flaw in modeling process can substantially contribute to the development cost and time. The operational efficiency may be affected as well. Here special attention is paid in planning phase and the same is extended to the implementation phase of the paper.

KEYWORDS: Functional Model, Behavioral Model, Object, Modeling, Operational Efficiency.

I. INTRODUCTION

Modeling is a process used to define and analyze data requirements needed to support the business processes within the scope of corresponding information systems in organizations. Therefore, the process of data modeling involves professional data modelers working closely with business stakeholders, as well as potential users of the information system. There are three different types of data models produced while progressing from requirements to the actual database to be used for the information system. The data requirements are initially recorded as a conceptual data model which is essentially a set of technology independent specifications about the data and is used to discuss initial requirements with the business stakeholders. The conceptual model is then translated into a logical data model, which documents structures of the data that can be implemented in databases. Implementation of one conceptual data model may require multiple logical data models. The last step in data modeling is transforming the logical data model to a physical data model that organizes the data into tables, and accounts for access, performance and storage details. Data modeling defines not just data elements, but their structures and relationships between them. Data modeling techniques and methodologies are used to model data in a standard, consistent, predictable manner in order to manage it as a resource. The use of data modeling standards is strongly recommended for all projects requiring a standard means of defining and analyzing data within an organization, e.g., using data modeling:

- to manage data as a resource;
- for the integration of information systems;
- for designing databases/data warehouses (aka data repositories)

Data modeling may be performed during various types of projects and in multiple phases of projects. Data models are progressive; there is no such thing as the final data model for a business or application. Instead a data model should be considered a living document that will change in response to a changing business. The data models should ideally be stored in a repository so that they can be retrieved, expanded, and edited over time. Whitten (2004) determined two types of data modeling:

- Strategic data modeling: This is part of the creation of an information systems strategy, which defines an overall vision and architecture for information systems is defined. Information engineering is a methodology that embraces this approach.
- Data modeling during systems analysis: In systems analysis logical data models are created as part of the development of new databases.

Data modeling is also used as a technique for detailing business requirements for specific databases. It is sometimes called database modeling because a data model is eventually implemented in a database.

UML is an industry standard modeling notation ,which provides foundational benefits:

- We have the potential of handing your diagram to someone who already knows how to interpret the notation without being told.
- We (and our audience) can find books, training, articles, web sites, and other educational and support resources for UML.
- There has been widespread tool adoption of UML, from drawing tool templates to repository based modeling tools to integrated development environments.
- We and many of your stakeholders can increase personal market value by mastering the notation we can hire people who know how to read and write UML diagrams without requiring training.

UML is a very structured notation. The specific structure brings benefits:

- The precise structure of the diagrams assists with consistency, completeness, and scope. Recasting your architectural vision into a predefined structure forces you to ask the right questions and flesh out the appropriate details.
- The fact that it is a structured notation makes it possible for tool vendors to provide functionality to transform a diagram from one view to another or even to code.

Static analysis aims at recovering the structure of a software system, while dynamic analysis focuses on its run time behavior. We propose a technique for combining the analysis of static and dynamic architectural information to support the task of architecture reconstruction. The approach emphasizes the correct choice of architecturally significant concepts forthe reconstruction process and relies on abstraction techniques for their manipulation. The technique allowsthe software architect to create a set of architectural views valuable for the architecture description of the system. To support our technique, we outline an environment that relies on hierarchical typed directed graphs to show the system's structure and message sequence charts for its behavior. The main features of the environment are: visualization of static and dynamic views, synchronization of abstractions performed on the views, scripting support and management of the use cases.

II. CLASS DIAGRAM

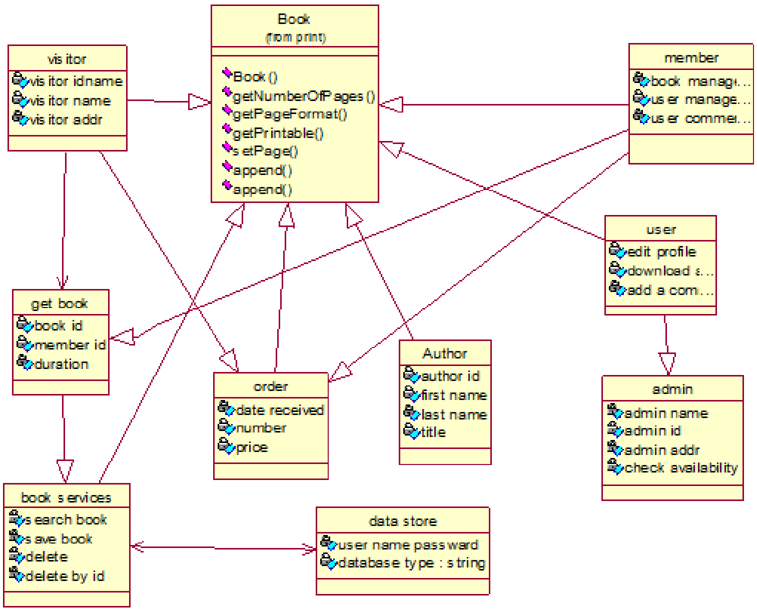

Fig 1 Class Diagram for Book Bank

III. USE CASE DIAGRAM

- **Use cases**. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- **Actors**. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- **Associations**. Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.
- **System boundary boxes (optional)**. You can draw a rectangle around the use cases, called the system boundary box, to indicates the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not. System boundary boxes are rarely used, although on occasion I have used them to identify which use cases will be delivered in each major release of a system.
- **Packages (optional)**. Packages are UML constructs that enable you to organize model elements (such as use cases) into groups. Packages are depicted as file folders and can be used on any of the UML diagrams, including both use case diagrams and class diagrams. I use packages only when my diagrams become unwieldy, which generally implies they cannot be printed on a single page, to organize a large diagram into smaller ones
- **Include**: **Include** is like a reference to next part, the use case is not completed without it. This part should be referenced from more places otherwise its use has no sense. An including use case calls or invokes the included one. Inclusion is used to show how a use case breaks into smaller steps. The included use case is at the arrowhead end.

- **Extend**: An extending use case adds goals and steps to the extended use case. The extensions operate only under certain conditions. The extended use case is at the arrowhead end.

Importance of Use cases:

- Use cases are important because they are in a tracking format. Hence they make it easy to comprehend about the functional requirements in the system and also make it easy to identify the various interactions between the users and the systems within an environment. They are descriptive and hence clearly represent the value of an interaction between actors and the system. They clarify system requirements very categorically and systemically making it easier to understand the system and its interactions with the users. During the analysis phase of the project's System Development Life Cycle, use cases help to understand the system's functionality.

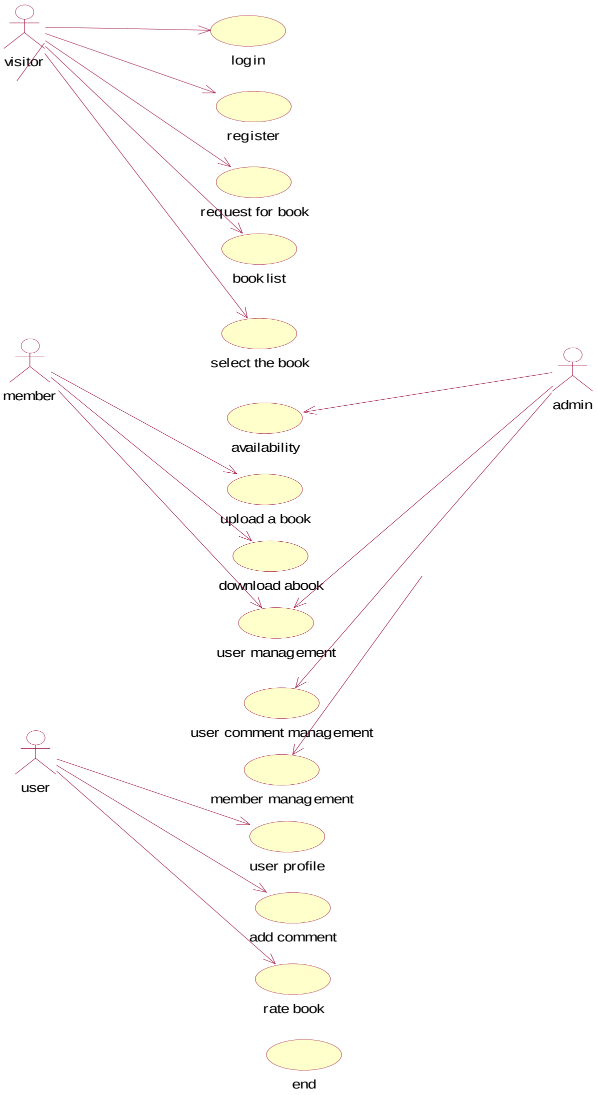

Fig2. Use Case Diagram for Automated Bookbank System

IV. INTERACTION DIAGRAM

Once the use cases are specified, and some of the core objects in the system are prototyped on class diagrams, we can start designing the dynamic behavior of the system.

Diagram Elements:

- Object. Each of the objects that participate in the processing represented in the sequence diagram is drawn across the top. Note that objects are used in this diagram while classes are used in use cases, class diagrams, and state-transition diagrams.
- Lifeline. A dotted line is dropped from each object in the sequence diagram. Arrows terminating on the lifeline indicate messages (commands) sent to the object. Arrows originating on the lifeline indicate messages sent from this object to another object. Time flows from top to bottom on a sequence diagram.
- Active. To indicate that an object is executing, i.e., it has control of the CPU, the lifeline is drawn as a thin rectangle.
- Message. A horizontal arrow represents a message (command) sent from one object to another. Note that parameters can be passed as part of the message and can (optionally) be noted on the diagram.
- Return. When one object commands another, a value is often returned. This may be a value computed by the object as a result of the command or a return code indicating whether the object completed processing the command successfully. These returned values are generally not indicated on a sequence diagram; they are simply assumed. In some instances the object may not be able to return this information immediately. In this case, the return of this information is noted on the diagram later using a dotted arrow. This indicates the flow of information was based on a previous request.
- Conditional. Square brackets are used to indicate a conditional, i.e., a Boolean expression that evaluates to TRUE or FALSE. The message is sent only if the expression is TRUE.
- Iteration. Square brackets preceded by an asterisk (*) indicate iteration. The message is sent multiple times. The expression within the brackets describes the iteration rule.
- Deletion . An X is used to indicate the termination (deletion) of an object.

V. CONCLUSION AND FUTURE WORK

The paper proposes a set of diagrams based on object orientation principle which describes the functionalities from various perspectives. The diagrams depicts the functional, behavioral and structural aspects of automated passport system. The proposed modeling methodology is based on the principle of object orientation, which allows describing both software and functionalities explicitly. Furthermore, it is illustrated how the well-known object-oriented specification language unified modeling language can be adopted, to provided an adequate formalization of its semantics, to describe structural and behavioral aspects of smart phone database management system related to both logical and physical parts. It is needed to implement the software on the basis of Object Oriented Model developed. Flaw in modeling process can substantially contribute to the development cost and time. The operational efficiency may be affected as well. Here special attention is paid in planning phase and the same is extended to the implementation phase of the paper.

REFERENCES

1. B.Powmya,NikitaMaryAblett,V.MohanaPriya,S.Balamurugan,"An Object Oriented Approach to model the secure Healthcare Database system", In the proceedings of international conference on compter, communication and signal processing(IC3SP),in Association with IETE Students Forum and The Society of Digital Information and Wireless communication SDIWC,2011,pp.2-3.
2. Conrad Bock, U.S. National Institute of Standards and Technology, "UML 2 Composition Model", Journal of Object Technology, Vol.3, No.10, 2007
3. Thomas M.Atwood, "The Case for Object Oriented Databases" IEEE Spectrum, 0018-9235/91/ 0002-004, pp 44-48, 2007
4. James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen "Object Oriented Modeling and Design", Prentice-Hall of India Private Limited.
5. Grady Booch, "Object-Oriented Analysis & Design with Applications", Pearson Education
6. Rogert.S.Pressman, "Software Engineering:A Practioner's Approach" Fifth Edition, McGraw-Hill Higher Education.
7. Armour, Frank, and Granville Miller. *Advanced Use Case Modeling: Software Systems*. Addison-Wesley, 2000.
8. http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/state.htm

9. http://visualcase.com/tutorials/interactions-diagram.htm
10. http://developer.com/design/article.php/2206791
11. Balamurugan Shanmugam, Visalakshi Palaniswami, "Modified Partitioning Algorithm for Privacy Preservation in Microdata Publishing with Full Functional Dependencies", Australian Journal of Basic and Applied Sciences, 7(8): pp.316-323, July 2013
12. Balamurugan Shanmugam, Visalakshi Palaniswami, R.Santhya, R.S.Venkatesh "Strategies for Privacy Preserving Publishing of Functionally Dependent Sensitive Data: A State-of-the-Art-Survey", Australian Journal of Basic and Applied Sciences, 8(15) September 2014.
13. S.Balamurugan, P.Visalakshi, V.M.Prabhakaran, S.Chranyaa, S.Sankaranarayanan, "Strategies for Solving the NP-Hard Workflow Scheduling Problems in Cloud Computing Environments", Australian Journal of Basic and Applied Sciences, 8(15) October 2014.
14. Charanyaa, S., et. al., , A Survey on Attack Prevention and Handling Strategies in Graph Based Data Anonymization. International Journal of Advanced Research in Computer and Communication Engineering, 2(10): 5722-5728, 2013.
15. Charanyaa, S., et. al., Certain Investigations on Approaches forProtecting Graph Privacy in Data Anonymization. International Journal of Advanced Research in Computer and Communication Engineering, 1(8): 5722-5728, 2013.
16. Charanyaa, S., et. al., Proposing a Novel Synergized K-Degree L-Diversity T-Closeness Model for Graph Based Data Anonymization. International Journal of Innovative Research in Computer and Communication Engineering, 2(3): 3554-3561, 2014.
17. Charanyaa, S., et. al., , Strategies for Knowledge Based Attack Detection in Graphical Data Anonymization. International Journal of Advanced Research in Computer and Communication Engineering, 3(2): 5722-5728, 2014.
18. Charanyaa, S., et. al., Term Frequency Based Sequence Generation Algorithm for Graph Based Data Anonymization International Journal of Innovative Research in Computer and Communication Engineering, 2(2): 3033-3040, 2014.
19. V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa," Certain Investigations on Strategies for Protecting Medical Data in Cloud", International Journal of Innovative Research in Computer and Communication Engineering Vol 2, Issue 10, October 2014
20. V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa," Investigations on Remote Virtual Machine to Secure Lifetime PHR in Cloud ", International Journal of Innovative Research in Computer and Communication Engineering Vol 2, Issue 10, October 2014
21. V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa," Privacy Preserving Personal Health Care Data in Cloud" , International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014
22. P.Andrew, J.Anish Kumar, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, "Investigations on Evolution of Strategies to Preserve Privacy of Moving Data Objects" International Journal of Innovative Research in Computer and Communication Engineering, 2(2): 3033-3040, 2014.
23. P.Andrew, J.Anish Kumar, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, " Certain Investigations on Securing Moving Data Objects" International Journal of Innovative Research in Computer and Communication Engineering, 2(2): 3033-3040, 2014.
24. P.Andrew, J.Anish Kumar, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, " Survey on Approaches Developed for Preserving Privacy of Data Objects" International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014
25. S.Jeevitha, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, " Privacy Preserving Personal Health Care Data in Cloud" International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014.
26. K.Deepika, P.Andrew, R.Santhya, S.Balamurugan, S.Charanyaa, "Investigations on Methods Evolved for Protecting Sensitive Data", International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 4, Decermber 2014.
27. K.Deepika, P.Andrew, R.Santhya, S.Balamurugan, S.Charanyaa, "A Survey on Approaches Developed for Data Anonymization", International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 4, Decermber 2014.
28. S.Balamurugan, S.Charanyaa, "Principles of Social Network Data Security" LAP Verlag, Germany, ISBN: 978-3-659-61207-7, 2014
29. S.Balamurugan, ,M.Sowmiya and S.Charanyaa, "Principles of Scheduling in Cloud Computing" Scholars' Press, Germany,, ISBN: 978-3-639-66950-3, 2014
30. S.Balamurugan, S.Charanyaa, "Principles of Database Security" Scholars' Press, Germany, ISBN: 978-3-639-76030-9, 2014