Design of an Ameliorated Methodology for the Abstraction of Usable Components of Object Oriented Paradigm from the Software Requirement Specification (SRS)

First A. Dr. Shivanand M Handigund¹, Second B. Dr. R J D'Souza², Third C, Syed Naimatullah Hussain³
¹BIT/CSE, Bangalore, India
²NITK/MACS, Surathkal, India
³NITK/MACS, Surathkal, India
Email: smhandigund@gmail.com
Email: rjd@nitk.ac.in
Email: snuh@rediffmail.com

Abstract – The object oriented design requires that the view element required for the design is to be abstracted from the SRS. So it is required to transform the requirements into object oriented paradigm and then proceed for the development. We are intending in our ensued project, to develop a sequence of methods in the form of methodology, those take the requirements and then transform it into object-oriented paradigm. We are intending to develop an automated (with least human intervention) sequence of methodology that takes requirements specification as input and abstracts required elements for the object oriented system. This is a semiautomatic methodology. In few steps of our methodology whenever the human intervention is required the detailed guidelines for that of the process is framed to facilitate the human worker to take unique unambiguous decision.

I. INTRODUCTION

The software development project normally starts with customers’ requirements. The customers are in general, strategic management people of the organization who are the user’s of the ensuing tool. So the requirements of the ensuing system reflect their processing mindset. This will not serve the evolving process of organization. Presently, this will not serve the development process effectively. Now a day, people feel that the naturalness virtue of object-oriented paradigm made it more reliable durable and stable. The object oriented design requires that the view element required for the design is to be abstracted from the SRS. So it is required to transform the requirements into object-oriented paradigm and then proceed for the development. We are intending in our ensued project, to develop a sequence of methods in the form of methodology, those take the requirements and then transform it into object-oriented paradigm. We are intending to develop an automated (with least human intervention) sequence of methodology that takes requirements specification as input and abstracts required elements for the object oriented system. This is a semiautomatic methodology. In few steps of our methodology whenever the human intervention is required the detailed guidelines for that of the process is framed to facilitate the human worker to take unique unambiguous decision.

Few researchers [1, 6] have suggested some techniques for certain stages of the design of object classes. Although, these guidelines may facilitate to certain extent for the abstraction of object class name but since these methodologies are based on conjectures. There is not authentication of correctness and completeness of the end product of the ensued abstractions. We have made an attempt develop a methodology that identifies the object-oriented specifications in the form of object structures, object methods and the interrelationships, from the requirements of an information system. This semi automatic methodology comprises of a sequence steps like feasibility analysis, for object structure identification, resolution of synonyms & homonyms issues, regrouping of attributes of entities & functionalities through the design of data flow diagrams and elimination of imbalance between data & procedure selection along with authentication of correctness & completeness of the abstractions at each stage. In the proposed methodology we have surmounted this lacuna and we have avoided conjecturing.

II. OBJECTIVE OF THE STUDY

To develop a single semi automated methodology for the abstraction of different view elements from SRS, those are required for object oriented system as existing methodology is manual, without the authentication of completeness and correctness. Following are the objectives of our automated methodology.

- View elements: class name, attributes domain pair, method name, interrelationship among
different object classes, functional dependencies between attributes of same class
- The authentication of completeness and correctness is achieved through one-one nouns and noun phrases & on-to correspondence the group comprising of object class names, Actors and Attributes.
- The abstracted elements are refined using good database design principles and good software engineering principles
- Widens the scope of existing methodology
- Eliminate conjectures & oxymoron
- Replace the arbitrary clumsy

II. LITERATURE SURVEY

A. Work Already Carried Out By Few Researcher’s

Noun Phrase Approach

Short falls in Noun Phrase Approach
- Rebecca Wirf’s et al have proposed an iterative method, which manually selects attribute name from the noun phrases and eliminates some of them based on some reasoning. They have not specified when the iteration needs to be stopped, nor specified how the completeness and the correctness is achieved.
- They have identified object attributes from the noun phrases and object methods from verbs. This is questionable issue as there is vast flexibility in the English language for using verbs.

B. Common Class Pattern Approach

Short falls in Common Class Pattern Approach
- Common class pattern approach considers the entity as the object class structure. We know that the object class, the structure, the entity, the states and methods are encapsulated together,, but in the entity concept functionality may utilize part of an entity attributes or it may contain number of attributes across different entities. Thus it fails to distinguish between entity and the object class structure.

C. Use-Case Driven Approach

Short falls in Use-Case Driven Approach
- This identifies the information system as scenarios, the union of all scenarios becomes the behavior of the system. This is a questionable concept as people are finding some activities of the system are beyond the scopes of use-case scenarios.

IV. PROPOSED SOLUTION METHOD/ALGORITHM

A. The Methodology

We have proposed a methodology, which will help in developing an automated methodology for the abstraction of object structure from the SRS. The input to the system is software requirements specification (SRS). The output of the methodology is the object structure functional dependencies, and also attribute domain pair. This methodology comprises a sequence of semiautomatic methods. Each stage of each method is proposed with details of either procedure, if it is automated, or guidelines, if it is a manual process.

B. System Requirement Specification (SRS)

Automation in Technical Institute / College

Functional Requirements:
Admit Student:
Student will get entrance exam card based on his/her performance, preference of college and availability of seats. The student will submit entrance exam card, original documents referenced in the card along with appropriate fees to the college and gets acknowledgement.

Register For Course:
The academic section verifies the documents referenced in the card, if satisfied assign roll number to each student branch wise, year wise and after admission process is completed, sends the list to the concerned department.

Classroom Allocation:
Each batch of each semester is assigned with a classroom thus a student is attached to a classroom in which the teaching learning process takes place as per the schedule

Faculty Work Load:
A faculty member teaches the student the allocated subject in the allocated classroom at the allocated time day and hour.

C. Identify Different Nouns and Noun Phrases &

Abstract Referenced and Defined Nouns from SRS (IV.B)

Identify noun/noun-phrases(N), Adjectives(Aj), Transitive Verbs(Vt), intransitve verb(Vi), Intransitive verb(Vi), impersonal verb(Vimp), Auxiliary Verb(Vaux), Adverb and Adverbial Phrases(Av), discard (Vaux), convert passive voice to active voice.

<table>
<thead>
<tr>
<th>Noun</th>
<th>Adverb</th>
<th>Adjective</th>
<th>Vt</th>
<th>Vi</th>
<th>Vaux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Formally</td>
<td>Admit</td>
<td>Provide</td>
<td>Is</td>
<td>Can</td>
</tr>
<tr>
<td>Purpose</td>
<td>suitably</td>
<td>Allocated</td>
<td>Intended</td>
<td>Receive</td>
<td></td>
</tr>
<tr>
<td>Document</td>
<td>only</td>
<td>Eligible</td>
<td>Admitted</td>
<td>Fix</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 1

To identify the functional dependencies among different attributes of object classes we are abstracting the referenced and defined nouns from the SRS(IV.B).

REFERENCES

{Entrance exam card, performance, seat, college} → {Student}
{Entrance exam, fees, student} → {Acknowledgement, College}
D. Resolve Synonym, Homonym Issues From DOD (Data Oriented Data) & POF (Procedure Oriented Functionalities)

Step-1 Collect all attributes & entities their functional dependencies, interrelationship and procedure from customer team of user’s CTU’s.
Step-2 Design entities attributes along with primary keys and foreign keys. Develop ER model (DOD) collected from functionality (POF).
Step-3 Resolve synonym & homonym from both from DOD & POF.
Step-4 visualize all the functionalities that can be abstracted from the entity and attributes of DOD Similarly identify the attributes and entities required for the functionality POF

Establish On-to Correspondence Between DOD & POD (procedure oriented data)

Match the attributes of DOD with attributes of POD & store them separately in DD (matched data) from DOD & POD (Procedure Oriented Data), for unmatched attributes see whether an attribute of one set has a matching part in the attribute of other set, decompose the other set of attributes and add matching attributes to corresponding pool.

Now we may have subsets containing unmatching attributes. Now consider in each set whether an attributes is synonymous with any attributes of other attributes, if so add to the pool of respective attribute, discard unmatched attributes of DOD and decompose the unmatched attributes of POD & add in both.

E. Group the Functional Dependencies for Identification of the Object Structure Attributes of Object Class from SRS(B)

Eg:

Functional Dependencies

\{Entrance Exam card, Performance, seats, college\} \rightarrow \{Student\}
\{Entrance exam, fees, student\} \rightarrow \{Acknowledgement, College\}

Grouping the Attributes on RHS from FD’S

Student

Student, acknowledgement, college, semester, Performance, borrowed books, shortage, Lab exam, admission, staff

Performance

Performance, student

College

College, acknowledgement, order, engg degree

Staff

Staff, student

Lab Exam

Lab exam, student

Classroom

Classroom, batch, semester, student

Object Oriented

Fig 3 illustrates a perfect balance between the data and procedure oriented approach, here we identify the attributes using object oriented paradigms.
F. Separate Out Actor from Functional Dependencies

Attributes which are only referenced or defined are Actor (interface).
Eg: University, parents, candidate, management, entrance exam cell, supplier, KCSR
Attributes which are both referenced & defined are the attributes of object class.
Eg: Student, college, semester, subject, classroom, books, test

G. Context Level Diagram

In the above context diagram, the attributes Entrance Exam, AICTE, University, Degree, management are depicted as the actors and manage teaching learning process is depicted as the lone process. The data stores, data flows and the sub processes are within this process. Here, a student is admitted to college when he/she qualifies for the entrance exam. To get admission to a college for a requisite branch of requisite programme, he/she has to produce his/her name, rank no, branch, programme allocated, to the college. The college management ensures that the admission of the candidate does not overflow the total intake allocated by AICTE. The university examination activity starts with the candidates’ sending of their details like US No., Course Nos., branch, programme & Fees payment. University will conduct examinations and send marks details to the respective US Nos. To seek placement activity, a student has to produce proof of his/her US No., Degree, and Branch and marks card.

H. Logical DFD:

I. Authentication of Correctness and Completeness from SRS

Identify object class for the correctness and completeness.
Correctness means whether all the abstraction are correct or whatever, I have abstracted is correct
Eg. Student, Faculty, and corridor here corridor is deleted because it is not an object class.
Completeness means whether all the elements are abstracted or whether all the attributes are covered.
Eg. Student Name whether student name is answer to any of the attributes.

J. Normalization

- College { college-name, branch, Programme, Intake, Compliance-report, Subject }
- Branch { Hod, branch-name, No.-of-faculty }
- Programme { Prgm-name, Prgm-co-ordinator, no. of student, qualification }
- Faculty { Fname, Fid, designation, brch-name, specialization }
- Student { USno, sname, prgm,branch, qual }
- Salary { Fid, Fname, Fbasic, date-of-next-incrmnt. date-of-present-scale }
- Class Room { room-no, Fname, sub, fhour, pgm, brnch, sem availty day }
- Marks { USNo, sub-code, marks-obtn, grade-obtn, min-mrks, max-mrks }
K. Two Levels of Data Integration.

V. CONCLUSION

This paper discusses the framework of our proposed research. The framework is developed on the study of different methodologies that exist for the abstraction of object classes from the software requirements specification. The available methodologies are designed with broken sequence of methods. Moreover piece-wise methods are manual methods without facilitating the opportunity for authenticating the correctness and completeness.

These methods can only be used to abstract object class names, currently no methods are available to abstract object attributes, methods moreover in the proposed methodology, we have used good database design principles to strengthen the high cohesion and low coupling properties, moreover we are planning to develop methodology for the abstraction of object method inculcating good software engineering design principles, thus the methodology transfers the business information into domain elements. Since we propose to use good data base design principles and good software engineering principles. The abstracted object classes will be free from the anomalies and blend of balanced approach. We are intending to refine the abstracted object classes to be more natural than random object classes.

In the proposed paper we have identified the functional dependencies (FDs) and minimization of FDs, and Two levels of data integration for refinement of object method. Still there is need to use good software engineering principles to identify object attributes involved in the object method.

Our methodology is semi automatic it has advantages over existing methodologies in the sense that other methodologies have not given the clear cut methodology, sound reasoning for the steps. We have attempted to address these challenges in our methodology.

As the SRS is prepared by number of users in organization, due to flexibility in English language, different people use different words for the same meaning, automation is very difficult but however we propose an automated method in near future.

VI. REFERENCES