Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

ISSN NO. 2320-5407

Article DOI: 10.21474/IJAR01/3789
DOI URL: http://dx.doi.org/10.21474/IJAR01/3789

RESEARCH ARTICLE

ABSTRACTION OF OBJECT ORIENTED CLASS FROM SOFTWARE REQUIREMENTS.

Syed Naimatullah Hussain and Dr. Syed Zakir Ali.

1. JJTU, Rajasthan, India.
2. SECAB India.

……………………………………………………………………………………………………....

Manuscript Info

……………………….

Manuscript History

Received: 01 February 2017
Final Accepted: 06 March 2017
Published: April 2017

Key words:-
Programme, Object, Class, Sofware Requirements

Abstract

………………………………………………………………….

This paper abstracts the object-oriented class in the form object structures, object methods and their inter-relationships. This is achieved through the bridging of two different paradigms of procedure-oriented and of data-oriented methods, and then blending this bridged abstraction to the object-oriented paradigm. Here, the research has abstracted all the good features of the three paradigms with application of good database design principles.

……………………………………………………………………………………………………....

Introduction:-

Understanding the current state of requirements engineering practice in software industry is an important step for researchers and practitioners in order to improve software quality process. In spite of attempts to practice requirements engineering in small and medium-sized software enterprises in order to improve software quality, the significant issues relating to this area such as inconsistent and incomplete requirements, and inadequate requirements management tools still remain. Numerous requirement techniques have been proposed in the last decade in order to reduce requirement engineering challenges. However, Large number of requirement engineering techniques causes confusion while technique selection process. Thus, there is a need to investigate requirement techniques.

The proposed methodology overcomes all the short falls like synonymies and homonyms issue, absence of correctness and completeness authentication and limitation in the number of view elements abstracted through existing methodologies. The proposed method identifies the functional dependencies from the data flow diagram.

Requirements gathering:-

Requirement engineering field has been intensively studied in last decade. Even though requirement still one of the most critical processes in software development. Recent studies show that 56% of system defects are coming from requirement. Additionally, by one estimate requirement errors cost 10 times more than coding errors. Thus, there is a need to identify those critical requirement engineering challenges

The requirement of the information system contains the business rule of the information system along with the branches and various applications. For example, the requirements of the college information system may contain some of the business rules as follows.

Student admitted to the college based on Entrance Exam ranking and he will chose the branch, programme and the college.

Corresponding Author:- Syed naimatullah Hussain.
Address:- JJTU, Rajasthan, India.

Faculty teaches in the college assign course through the departments.
Each faculty teaches one or more subjects in one or more of the department.
Each faculty needs to stay in the quarters provided.
A student admitted to a programme, needs to study and appear in five or more courses relevant to a programme
Each student needs to stay in hostel provided
Each student appears for a internal assessment(IA) test, in each subject and gets marks
Each faculty evaluates student in his/her courses taught
Each faculty teaches their subjects in a classroom
A faculty may teach the allocated subjects in the same/different classroom on each day

Based on the functionalities involved in the behavior of the application the entities,attributes, and interrelationships are identified and this forms the data dictionary, which contains entity and attributes and their interrelationships.

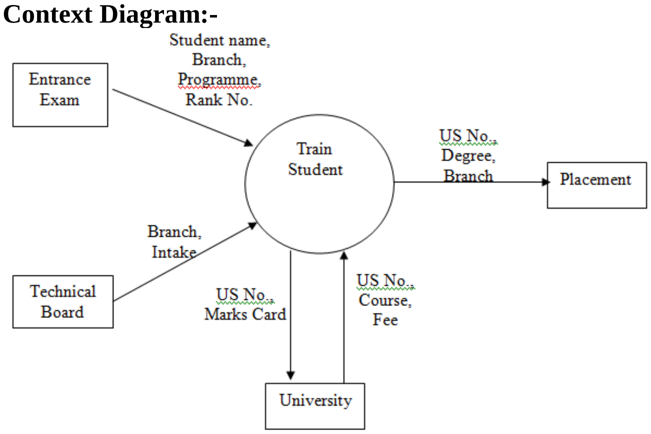
In the college information system example undertaken, the abstracted entities are:
Entrance Exam {Std-no., rank, branch, programme}.
COLLEGE{colg-cd, colg-nm, principal, location}
DEPARTMENT {dept-nm, hod, contact}
PROGRAMME{pgrm-nm, pgrm-coord, no.of std, min-qual,requrd}
FACULTY{techr-name, designation, deprt,special, course_code,}
COURSE {crse_cd, crse-title, max-marks, min-marks}
MARKS {us-no. crse-cd, marks-obtain, grade_obtained, prereq_crse-code}

- STUDENT {us-no., std-nm, prgm,crse, min-qual}
- CLASS ROOM{room-no, teacher-name, sub,hr, pgm}
- HOSTEL {host-nm, warden, tot.no.of-rooms}
- QUARTERS{qrtrs- no., tchr-name, d-occ,d-rtd, d-lvng}

Identifications of Synonyms and Homonyms:-
In an organization, there are several users. Each of them has his/her own perspective in coining the entities and attributes. Because of the flexibility in the English language, many meaningful names may be coined for the same attribute/entity. The meaningful names of the same entity form synonymous group (or synonymy) [2]. There are many methodologies available in the schema integration literature [4, 5, 6, 7, 11]. Each synonymy is given with a generic name. Similarly, the contest specific use of attribute or entity names may result in, the use of same attribute/entity with different meaning in different contest. This situation leads to the existence of homonyms. This situation is to be resolved by assigning different names to each of such homonym word.

Elimination of redundant Attributes:-
Each attribute of each entity is studied in isolation with other attributes of other entities, for their absolute necessity in characterizing the entity. Unnecessary attributes are eliminated. If an attribute/group of attributes is common for two or more entities, this common attributes group is separated to form a link entity [8]. This process implicitly identifies the interrelationships, which has been discussed in one of the following paragraphs.

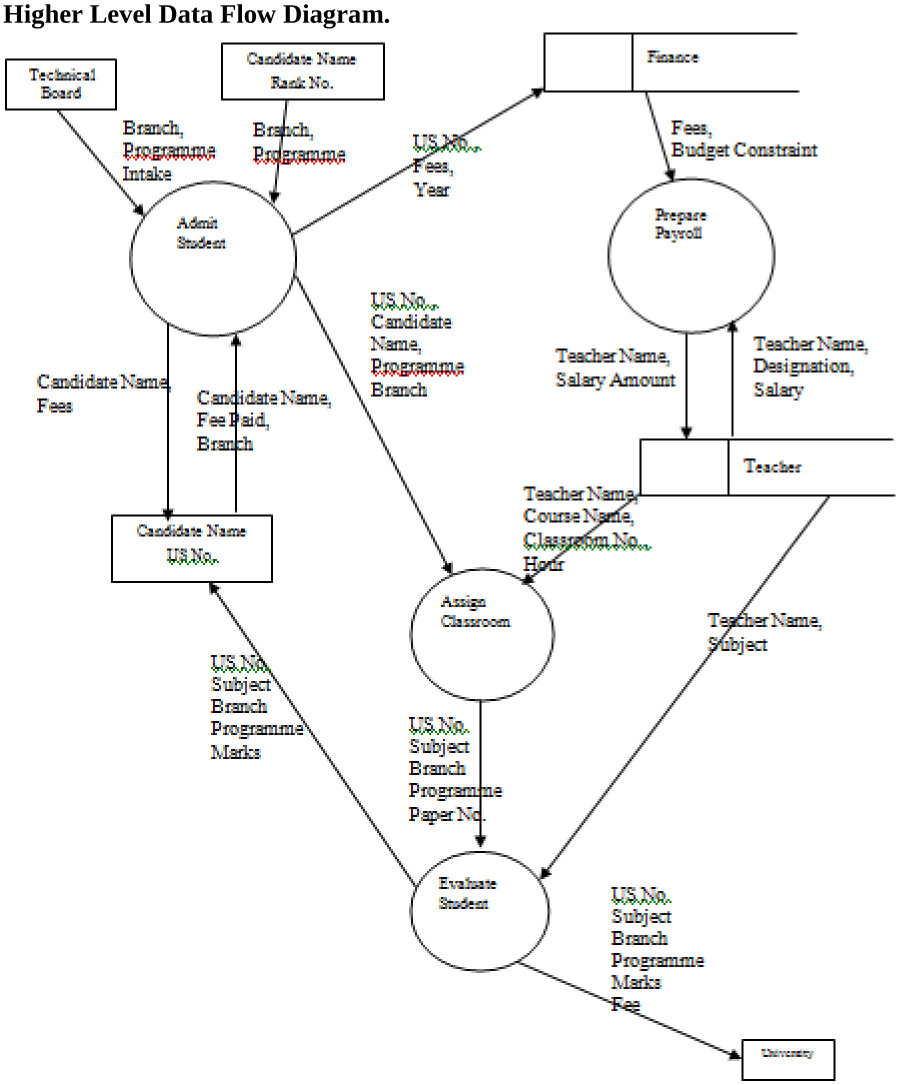
Context Diagram:-



In the above context diagram, the attributes Entrance Exam, Technical Board, University, PLACEMENT are depicted as the actors and TRAIN STUDENT 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 Technical Board. 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.

Higher Level Data Flow Diagram.



In the higher-level data flow diagram [9] above, the process is decomposed into manageable sub processes along with data flows of the data files used within the system.

Identification of functional and multivalued dependencies:-

These entities are now refined with elimination of redundant attributes and entities. These can serve as first cut object structures. Now the functional dependencies and the multi-valued dependencies that may exist amongst the attributes of each entity are to be identified. The undesirable functional dependencies are to be eliminated using normalization process in sequence from the first normal form to the Boyce-Codd normal form (BCNF). The undesirable multi-valued dependencies are identified through the one-to-many relationships between different attributes of each entity. These are eliminated by decomposing each such entity using fourth normal form and project join normal form (PJNF).

Entrance Exam:-

College Name	Rank	Branch	Programme

Since, the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key therefore, there is no multi value dependency and relation is already in the project join normal form (PJNF).

College:-

College Code	College Name	Principal	Location

Here, the college code and college name are the two candidate keys. We consider college code as primary key. The college name determines the principal. Here a non-key attributes another non-key, and therefore the relation is not in third normal form. Therefore, the relation can be decomposed into two relations as follows:

- COLLEG (College code, College name)
- COLLEGE-LOC (College name, Principal location)

These relations are already in PJNF as they comprise single attribute primary keys.

Department:-

Dept. Name	HOD	Building No.	Telephone

Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key and therefore, there is no multi value dependency and relation is already in the project join normal form (PJNF).

Programme:-

Prog. Name	Prog Co-coordinator	No. Of Students	Max. Qual. reqd

Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key and therefore, there is no multi value dependency and relation is already in the project join normal form (PJNF).

Faculty:-

Teacher Name	Designation	Specialization	Course Code

Since a teacher can teach more than one subject, the primary key is composite key comprising teacher name, course code. The relation is already in first normal form. This is not a second normal form as teacher name determines. Therefore, the relation is decomposed into two relations as follows:

a. TEACHER-SPEC (Teacher name, Designation, Specialization)
b. TEACHER_COURSE (Teacher Name, Course Code)

They are already in BCNF and PJNF.

Course:-

Course Code	Course Title	Programme	Max. Marks	Min. Marks

Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key therefore there is no multi value dependency and relation is already in the project join normal form (PJNF).

Marks:-

US No.	Course Code	Programme	Marks Obtained	Grade	Pre Requisite Course Code

Here, US No. and Course code together form the primary key but US No. determines Course-code, therefore it is not in the second normal form. Therefore the relation is decomposed into:

c. MARKS-US NO (US No., Programme). It is already in PJNF
d. MARKS-COURSE (US No., Course-code, Marks, Grade, Pre-requisite)

The second relation is not in third normal form and therefore this is decomposed into:

- MARKS_NO (US No., Course No., Marks),
- MARKS_GRADE (Marks, Grade, Course No., Pre-requisite)

These are in PJNF

Student:-

US No.	Student Name	Programme	Course	Min Qualification

Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key and therefore, there is no multi value dependency and relation is already in the project join normal form (PJNF).

Class Room:-

Room No.	Teacher Name	Subject	Hour	Programme

Room No., Teacher Name, Subject together form primary key. Here, Hour determines the Subject. Therefore, the relation is not in BCNF. Therefore relation is decomposed into:

- TEACHER_ROOM (Room No, Teacher Name , Subject, Programme) and
- STUDENT_HOUR (Hour, Programme)

These decomposed relations are in PJNF.

Hostel:-

Hostel Name	Warden	Total No. Of Rooms

Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key and therefore, there is no multi value dependency and relation is already in the project join normal form (PJNF).

Hostel Room:-

B.

Hostel Name	Room No.	US No. Of Student	Date of Occupancy	Date after which to quit

Here the primary key is US No. Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key therefore there is no multi value dependency and relation is already in the project join normal form (PJNF).

Quarters.

Qrtr. No.	Teacher Name	Date of Occupancy	Date of Retirement	Date of Leaving

Here Quarter No is a primary key and the date of leaving depends on the date of retirement. Therefore the relation is not in the third normal form and is decomposed into two relations as follows:

- QUARTER-TEACHER (Quarter No., Teacher Name, date of occupancy, date of retirement)
- QUARTER_LEAVING (Date of retirement Date of Leaving)

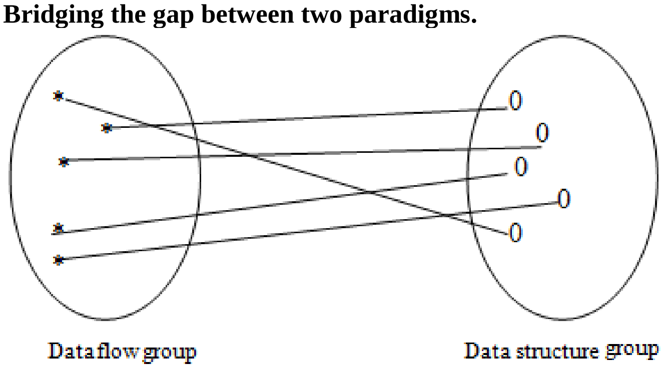
Quarter No. and Date of retirement are in PJNF.

Trust:-

Member Name	Designation	Elected/Selected/ ex-Official	Contact No.

Since the primary key of this relation contains single attribute and no-key determine another non-key. The relation is in BCNF. Moreover primary is not a composite key and therefore, there is no multi value dependency and relation is already in the project join normal form (PJNF).

Bridging the gap between two paradigms.



Now, we have identified first cut object structures using good database design principles on one hand. On the other hand, we have identified attributes for each dataflow through the design of higher-level data flow diagram. The object-oriented paradigm is the perfect balance [3] of these two paradigms. Thus, the design of object-oriented specifications need to blend the data oriented (object structures) paradigm with procedure oriented (Attributes group each representing a dataflow) paradigm. There needs to be a one-one and onto correspondences [1, 10] between the two sets of structures identified. This also implicitly verifies and validates the selection of object structures.

Now, we study the mapping between two groups, one group comprising attributes groups of data flows, each group representing a dataflow and on the other side, the refined object structures. We identify one-one onto correspondences between these two sets of elements. If an object structure contains one or more dataflow groups then, the corresponding functionalities are assigned to the contained object structure as objects methods. This process continues for all the matching object structures. Now, we take the set union of unmatched object structures and study the possible consideration of one-to-one mapping with left out dataflow groups. Each such matching data flow group forms an object structure with its destined process as object method. The left out data flow groups are manually studied for possible participation. Similarly, the left out object structures are studied for possible formation of abstract classes.

Conclusion:-

The Authors have succeeded in making the process semiautomatic, with least human intervention. The intervention is necessitated at critical points where intelligence is necessary.

Acknowledgement:-

Firstly, I would like to express my sincere gratitude to my advisor Prof. Dr S.M.Handigund for the continuous support related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time...

References:-

1. K.P Jayant, Garg,Kumar, Rana. "An approach of Software Design Testing Based on UML diagram. International Journal of Advanced Research in Computer Science and Software Engineering. Volume-4, Issue-2, February 2014
2. .S. Anandha Mala and Dr.G.V.Uma "Object Oriented Visualization of Natural Language Requirement Specification and NFR Preference Elicitation. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.8, August 2006.
3. Lilac A. E. Al-Safadi "Natural Language Processing for Conceptual Modeling" A semi-automatic approach for designing databases. International Journal of Digital Content Technology and its Applications Volume 3, Number 3, September 2009
4. Sunguk Lee "UML for Database System and computer application. Interantional Journal of Database System and Computer Application. Volume 5, number 3, 2012
5. Tiwari,Alpika,Dubey "Merging of Dataflow Diagrams with Unified Modeling Language" Internation Journal of Scientific and Research Publication Volume-2,issue-8, August-2012
6. Rosziati,Yen "A Formaol Model for Data Flow Diagrams Rules" APRN Jounal of System and Software, volume-1 No.2, May-2011
7. Shivanand M. Handigund, "Reverse Engineering of Legacy COBOL Systems", Ph. .D. Thesis, 2001 I. I. T. Bambay, Mumbai.8)AliBahrami "Object Oriented System Developments" McGraw-Hill International Editions 1999.
8. Sinan Si Albir. "UML in nutshell" , Shroff publishers and distributors private Ltd. 1998
9. S.B. Navathe C. Batini, M. Lenzerini. A comparative analysis of methodologies for database schema integration. ACM Computing Surveys, 18(4):323-364,1986.
10. Jaco de Bakker and Erik de Vink. Control Flow Semantics. The MIT Press, Cambridge, Masschusetts, 1996.
11. Joseph Fong et. al. Methodology of Scheme integration for new database applications: A practitioner's approach. Journal of Database Management, 10(1):3-18,1999.