Biomedical Research 2017; Special Issue: *ISSN 0970-938X*
www.biomedres.info

A software design technique for developing medical expert systems through use case analysis.

Lalitha R[1], Latha B[2*], Sumathi G[3]

[1]Rajalakshmi Institute of Technology, Research Scholar, Anna University, Chennai, India

[2]Sri Sai Ram Engineering College, Chennai, India

[3]Sri Venkateswara College of Engineering, Chennai, India

Abstract

Medical expert systems are useful since they assist the doctors in making decisions on treatment plan. Such a system can be developed by applying effective design techniques in order to improve the performance. Since medical expert systems are domain dependent, requirements analysis must be carried out before software design. In the object oriented design, use case diagram play a major role. The analysis of use case diagrams can be carried out by analysing the class hierarchy more effectively. Use case analysis with slicing technique will identify the details of one level of a class hierarchy. Therefore, a new use case slicing based approach is used to design software for developing a medical expert system. This work focuses on providing assistance to asthma and breathing related diseases. For this purpose, a knowledge based system is designed and implemented in this work which uses rules for making effective decisions. This knowledge based system is assisted with a feature selection module and a classification module. From the experiments conducted in this research work, it had been observed that the proposed expert system provides suitable decisions and recommendations on asthma treatment.

Keywords: Analysis, Slicing technique, Software design, Use case diagrams, Medical expert systems, Asthma.

Accepted on November 15, 2016

Introduction

Asthma is a serious disease since the patient is suffering from breathing problems. In such a scenario, the medical history of a patient can be analysed well by an automated intelligent system. Medical expert systems are domain dependent and have two main components namely knowledge base and inference engine. Medical applications can be developed by using the decisions made by expert systems. In order to develop such applications software design and implementation are important tasks after requirement analysis. Therefore, software design is an important activity in the software development life cycle. In the past, many design metrics have been proposed including high cohesion and low coupling. However, automated design from use case diagrams provided fast design methods and can help to structure the software design diagrams such as class diagrams, activity diagrams, sequence diagrams and deployment diagrams. Among them, class diagrams can be drawn by knowing the inheritance hierarchy which can be obtained from requirement specification.

On the other hand, sequence diagrams help to perform job sequencing that will help the software project managers to optimize the time and resources. Therefore, it is necessary to have an automatic software based method which can provide design diagrams and design documents with required quality. Recently, data flow diagrams are replaced by Unified Modeling Language (UML) techniques to perform object oriented analysis and design. Such a design is constrained by many object oriented metrics including abstraction, modularity, depth of inheritance, number of children per class, weighted methods for class, level of cohesion and coupling. Moreover, the pictorial representation of all symbols, diagrams with text in UML enables a better understanding of the activities, process involved in the project under development [1].

The UML diagrams are used to represent the analysis, design and implementation phases in software engineering [2]. Among the various diagrams in UML, the use case diagrams play a prime role in software development because all the other diagrams in UML like class diagrams, activity diagrams, interaction diagrams are generated based on the use case flows in the use case diagrams [3]. In the analysis phase of a software development, they are used to analyze the requirements and to represent the relationships between different processes involved in the project under development. The number of modules to be developed, the responsibilities of the actors, and the relationship between the actors are decided from the use case diagrams. They are also used to design the database at the backend. But, in case of very large projects in real time environment, the analysis and decision making process through

use case diagrams was very difficult and also may lead to different interpretations [1].

Hence, new techniques are proposed in this work by slicing the use case diagrams and to make the design better by enhancing the cohesion among the modules and by decreasing the coupling. The slicing is performed using the Model Dependency Graph (MDG) based approach proposed by Jaiprakash et al. [4] in order to identify the major components of the use case diagrams and to make suitable design. This model is used here to identify a number of parameters called features and a medical expert system is proposed in this work which performs rule firing, rule matching and effective decision making for design. The main advantage of analyzing the use case diagram with this proposed model is that the application of rules by the inference engine which uses the feature selection and classification module to identify the most important modules, class hierarchy, calling dependencies and strength of each module. In addition, it identifies the symptoms of asthma and suggests a suitable treatment methodology. Therefore, the proposed system is capable of analysing the past and present to predict the future so that a treatment plan can be recommended.

The remainder of this paper had been organized as follows: Section 2 provides a survey of related works and compares them with the proposed work. Section 3 depicts the architecture of the proposed system. Section 4 details the algorithms proposed in this work to provide medical advice and decision making. Section 5 provides results and discussions. Section 6 gives conclusion on this work and suggests some future works.

Materials and Methods

Related works

There are many works which are available in the literature which discuss about requirement analysis, software design, use case slicing and medical diagnosis [2,4-7]. Among them, Jaiprakash et al. [4] proposed a dynamic Slicing Technique for slicing the UML Architectural Models. They used a Model dependency Graph based approach to perform effective and dynamic slicing of diagrams used in object oriented analysis and design. Their work provides an important contribution for improving the analysis and design of object oriented systems based on use cases. Sethukkarasi et al. [6] proposed a fuzzy classifier based on cognitive maps for medical decision making. Ganapathy et al. [8] developed a medical expert system that uses fuzzy temporal rules for effective decision making on medical dataset. Rodrigo et al. [9] introduced a prediction model for prediction of workloads using real traces of requests to web servers from clients. Lallchandani et al. [5] proposed a technique for dynamic model slicing using state information. In their model, they constructed precise slices of UML architectural models which were difficult in nature since their model information was distributed across several diagrams with implicit dependencies among them.

Kusumoto et al. [10] proposed a new method called Use Case Point method to estimate the software development effort in the earlier phases of software development life cycle. The authors developed an automatic use case measurement tool which is used to classify the complexity of actors and use cases in the use case model. Swain et al. [11] proposed a method to generate Use Case Dependency Graphs from use case diagrams and Concurrent Control Graphs from sequence diagrams. Based on these, test cases were generated by them which are suitable for detecting synchronization and they were used for system testing. Berkovich et al. [12] developed a requirements data model for product service systems which is able to describe the various types of requirements and the relationship between them. Carnevali et al. [13] explained the design of a framework for testing of concurrent timed systems. In their model, the authors constructed a partially stochastic time Petri net to model the non-deterministic and non-controllable values and parameters. Kishore and Anjan [14] developed a maturity model in which they considered the key attributes of product engineering. It had four dimensions and the maturity across all the four dimensions namely process, architecture, infrastructure and people were considered. Alrajeh and Kramer [15] developed a process integration model for effective checking using inductive learning. The inductive learning process was used by the authors and to compute the operational requirements. Rosa and Lopes [16] proposed a framework for system modeling which provides suitable solutions based on behaviors and rules.

Gloria [17] proposed a work flow model to describe the dynamism of work flow. It was an intuitive approach and uses the logical termination concept to validate the work flow conditions. Oh et al. [18] proposed an independent software framework for search based software testing which is a common software platform and provides to reduce the time and effort. Apart from suggesting the design patterns, the proposed model also reduced the development time and effort by using common functions in the given framework. Prackweiser et al. [19] developed a business process model for understanding and analyzing the use case diagrams to improve the design. Mirna et al. [20] proposed a methodology that enables the organizations to implement a multi model software development environment based on business goals to enhance communication and information sharing. Marcos et al. [21] developed a new framework that helps the software developer to understand about the data organization in metric access method. Ionita [22] developed a model to study the use of UML for business software development.

The challenges in Requirements Engineering were analyzed through on-line survey by Bano et al. [23]. Pasupathy and Bhavani [24] investigated the object oriented metrics proposed by various researchers and the authors also proposed a methodology to determine the program efficiency and quality. Claudia et al. [25] developed a new framework to reverse engineer the Model Driven Architecture models from object oriented code. In their model, the abstraction levels are linked to models, Meta models and formal specifications. Moreover,

the use case diagrams were reverse engineered to Java code and analyzed.

Colombo automatic composition algorithm was used by Elgammal and El-Sharkawi [26] to model services and service composition. In their work, the transformation rules to specify the services were proposed and Extensible Mark Up Language was used to model the documents. Kumar and Babu [27] presented the various limitations that exist in the software development process. The authors proposed a new quality improvement model called Kano Lean Six Sigma Model to identify the exact requirements and to categorize them based on the nature of defect. This helps to implement the main functionality and to meet the expectations of the customer. Venkatesh and John [28] proposed a framework for analyzing and slicing concurrent Java programs. In their model, the Graphical User Interface was based on Kaveri and Eclipse. Booch et al. [1] explained about the design patterns in software development process. In their work, the artifacts in UML design and their features were explained with case studies. Chen and Wang [29] proposed a class relationship flow model to provide analysis for inheritance, association and aggregation flow in relevance to the relationships. This work represents a work flow model for developing object oriented systems. Kim et al. [30] proposed a dynamic slicing approach for analyzing the software architecture effectively.

From all these related works, it can be observed that many research works are being carried out in the object oriented software design. Most of the works discussed about the use of UML diagrams for effective design of object oriented systems. Moreover, few works considered the challenges in the transformation of analysis modeling to design. Moreover, all phases of the software development life cycle and testing were considered by many authors. The use of rules and use case slicing were also discussed in the literature. However, there is no single work which considers an analysis on requirement documents using natural language processing techniques for effective semantic analysis to improve the design. Moreover, the role of business intelligence in software systems is not considered in many works. However, the consideration of analysis documents, natural language processing and metrics will help to improve the design. Therefore, both system requirements analysis documents, rules and software architecture slicing must be consider while transforming the analysis model to design model. Therefore in this paper, both rules and intelligent agents are used for effective analysis and design of object oriented software for effective medical diagnosis.

System architecture

The overall architecture of the proposed work is shown in Figure 1. It consisted of seven major components namely System Requirement Specification (SRS) document, Use case repository, Graph generation and slicing module, feature selection module, classification module, decision manager, natural language analyzer and rule base.

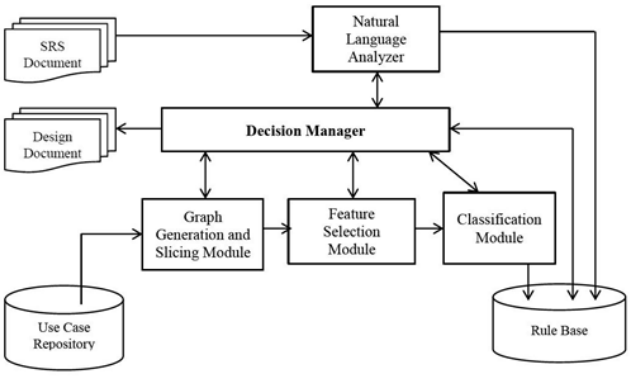

Figure 1: *System architecture.*

The system obtains the input from the use case repository which consists of use case diagrams for each application. The use case diagrams are used by the graph generation module which forms a model and then generates graphs from the use cases. The SRS documents which consist of functional and non-functional requirements of the system are used by the natural language analyzer to find the use cases and their importance. This helps to identify the important parameters and to give weights to them. The feature selection module uses the information from the graph generation module and the parameters provided by the natural language analyzer to perform feature ranking and feature selection. The classification module classifies the various modules into hierarchies in order to perform effective design. The classified results are stored in the form of rules in the rule base and they are sent to the decision manager for making design decisions. The overall control of the system is with the decision manager. All activities of the system are monitored and coordinated by the decision manager. The decision manager applies the rules to make effective decisions and is responsible for converting the use case diagrams into design diagrams and design documents.

Proposed work

In this paper, an automated use case slicing model for object oriented design is proposed which is useful for designing medical expert systems. This proposed model uses graph generation and slicing techniques proposed by Jaiprakash et al. [4] for effective slicing of use case diagrams. Moreover, this model helps the medical doctors for providing a treatment plan to the asthma patients. Thus it enhances the use case diagrams with artificial intelligence by applying rules for effective decision making.

The graph generation and slicing module consists of two phases namely identification phase and slicing phase. The identification phase identifies the use case diagrams pertaining to an application and helps to bring them from the use case repository. The slicing phase uses the existing graph generation approach proposed by Jaiprakash et al. [4] to slice the use case diagram into actors and use cases and the use cases are used for

graph generation. For demonstrating the proposed work, a medical registration application has been developed and implemented using UML modeling and also with the proposed model. In the medical registration module, the patient details, doctor details and the medicine and treatment details are stored. A knowledge base consisting of general rules for problem solving and domain rules for finding and treating asthma are stored. This knowledge base can be updated by adding more rules provided by the classifier through training. In addition, it provides a facility for carrying out the course registration for medical students. In this work, the use case diagram present in the use case repository is considered for analysis. To facilitate the analysis process, the use case diagram is sliced based on the technique proposed by Jaiprakash et al. [4]. The steps of the slicing algorithm are as follows:

Algorithm for use case slicing

Input: Use case diagrams

Output: Actors and segments of use cases

Step 1: Read SRS document and perform tokenization of sentences to identify the words from requirements

Step 2: Apply grammar and perform syntax analysis by constructing parse trees

Step 3: Use semantic markers and dictionary to perform semantic analysis

Step 4: Prepare a set of nouns to identify the actors and use cases from the use case diagrams Repeat until there are no more use case diagrams in the repository

Begin

Step 5: Call picture-reader ( ) to read one use case diagram from the use case repository Repeat until there are no more actors in the current use case diagram

Begin

Step 6: Identify and consolidate the number of outgoing arrows from the current actor

Step 7: Slice the use cases as Outgoing Slices (OS) that are related to step 6

Step 8: Identify and consolidate the number of incoming arrows from the current actor

Step 9: Slice the use cases as Incoming Slices (IS) that are related to step 8 End

Step 10: Name the slices as [actor-OS/IS n] where actor denotes the actor name, OS denotes the outgoing slice, IS denotes the Incoming slice and n denotes the slice number

Step 11: For each actor and use case perform a match with natural language analysis report

Step 12: If actors match with natural language analyzer set actors=true

Step 13: If use cases match with natural language analyzer set use cases=true

Step 14: If actors=true and use cases=true then Return actors and use cases

else

Return “Fail”

End

For the use case diagram in Figure 2, the slicing process is explained using Figures 3-5 in which the slices formed are as follows:

- Figure 4: Students are based on outgoing arrows.
- Figure 5: College are based on Outgoing arrows.
- Figure 6: College and student are based on incoming arrows.

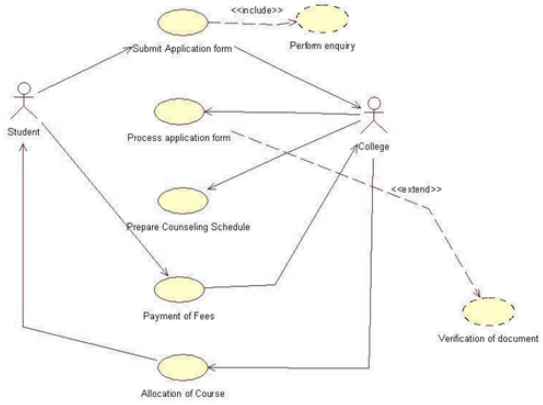

Figure 2. *Use case diagram for course registration system.*

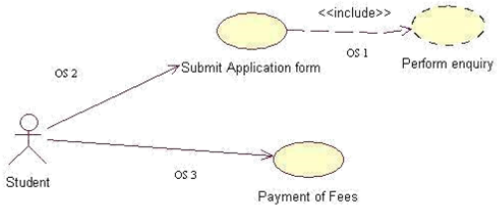

Figure 3. *Slice: Student–OS 1.*

These slices are now considered for further analysis by the decision manager. In this work, priorities are assigned based on rules which are formed by the classification algorithm. Moreover, initial priorities are assigned based on the information gain values obtained for each use case. This information gain values are computed by the feature selection module which ranks the features based on priorities and information gain values. The algorithm for Assigning priorities is as follows:

Feature selection algorithm

Input: Actors and use cases with natural language frequency and contribution feedback

Output: Use cases arranged in decreasing importance

Step 1: Read actors and use case slices

Step 2: For each actor and use case slices for this actor do Begin

Step 3: Identify the presence of use cases with <<include>> / <<extend>>relationship. If present, assign the top priority as 10 for the use caseand priority 5 for the use case which includes/extends the relation.

Step 4: If there are no slices with stereotype, then assign the next priority (priority-1) number in sequential order after following step 3.

Step 5: Compute information gain value using priority of actor and its use cases End for.

Step 6: Sort the use case slices in the decreasing order of information gain values.

Step 7: If the difference between the information gain values of any two use cases is less than the one third value of highest use case value eliminate such use cases.

Step 8: Return all the selected use cases as features for the classifier.

Based on this algorithm, the priorities are assigned for the use case diagram in Figure 2. The use case diagram with priorities assigned is given in Figure 6.

Now, the analysis is done for Figure 6. For every actor, the priorities are summed up based on outgoing arrows by adding up include/extend stereotype if any for outgoing arrows. Hence, sum of priorities is:

- Student: 6 (1+2+3)
- College: 10 (2+3+4+1)

Similarly, for every actor, the priorities are summed up based on incoming arrows. Hence, sum of priorities based on incoming arrows by adding up include/extend stereotype is

- Student: 1
- College: 4 (1+2+1)

By finding the sum of priorities, the actor with maximum responsibility can be easily found. Also, it helps to identify the coupling nature of actors. As these actors are created as database, the strength of relationship between the databases can be known well. By looking at the number of use cases assigned to actor, the cohesiveness of the actor can also be determined in this layer. In this work, classification is performed using Multiclass Support Vector Machines (MSVM) algorithm [7] in the following way.

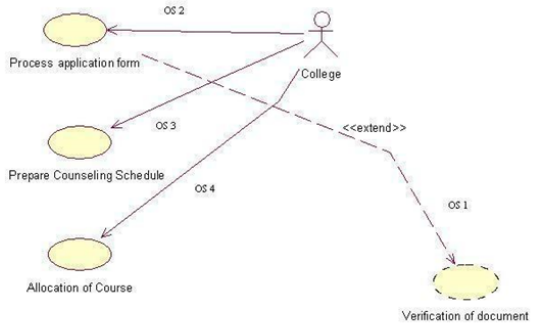

Figure 4. *Slice: College-OS 1.*

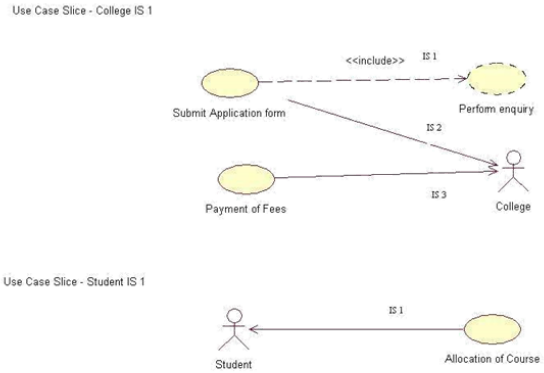

Figure 5. *College-IS 1 and Student-IS 1.*

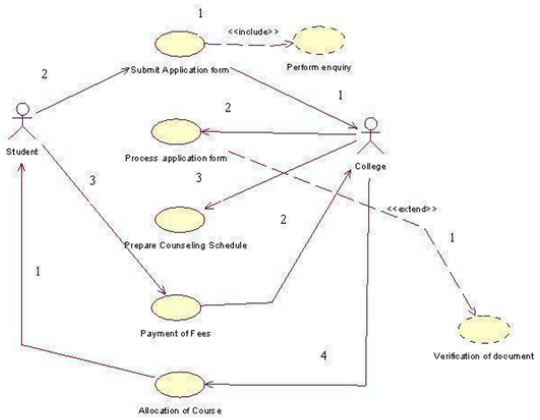

Figure 6: *Use Case Diagram with Priorities*

Classification process using MSVM

Input: Dataset and features

Output: Classified results and rules

Step 1: Read the dataset

Step 2: Read the features which are selected based on ranking

Step 3: Read the number of classes ‘n’

Step 4: Call MSVM to classify the dataset using this n classes and features [7]

Step 5: Send the classified results to the decision manager for making design decision and report generation

Step 6: Perform rule generation

Step 7: Send rules to knowledge base for storage

Step 8: Use rules available in the knowledge base to check cohesion and coupling

Step 9: Provide design recommendations

Step 10: Display the reports

The features selected by the feature selection module are used by the classification module with the monitoring of the decision manager. Moreover, the decision manager gets the feedback from the natural language analyzer that reads the SRS documents and identifies the importance of each use case by performing morphological analysis, syntax analysis, semantic analysis and pragmatic analysis. The decision manager is responsible for making the coordination between the feature selection and the classification modules. Finally, the design document is prepared and a report is generated on the design. Report generation is taken care of by the decision manager. It uses the information provided by other modules to generate different types of reports based on various parameters taken for analysis. The decision manager provides the following reports to the project manager on demand:

- A schedule showing the sequence of slices to be developed which is similar to a Gantt Chart
- A consolidated report showing number of use cases assigned to an actor
- A consolidate report showing number of slices, number of actors, total number of use cases.

These reports are used for identifying the complexity of the software, preparing a rough estimate on software completion schedule, preparing an estimate on cost involved in software production, to identify the risks involved in integration etc.

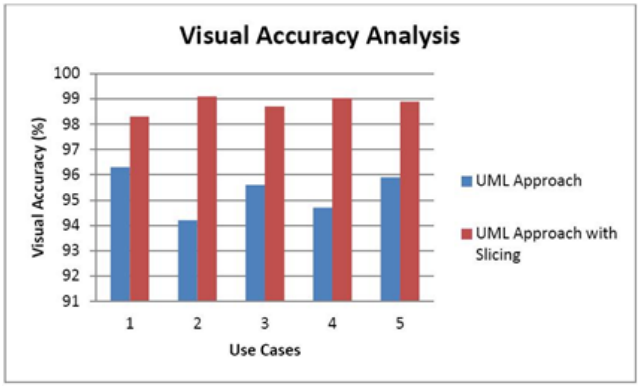

Figure 7: *Visual accuracy analysis.*

Result and Discussion

This proposed model has been implemented and has been tested by creating a course registration application for medical students. For this purpose, students from four different specializations namely ENT, Heart, Lung and Asthma and Eye specialization were considered amounting to a total of 360 students from each year and hence 1440 students from all the four years of engineering branches. A total of 120 faculty members from these four departments and Science and Humanities departments were considered for carrying out the experiments who were offering 330 theory courses and 70 laboratory courses. The results obtained from this work are as follows. Figure 7 shows the accuracy of feature selection using UML diagrams without slicing and withslicing. From Figure 7, it is observed that the UML approach with slicing provides more visual accuracy in comparison with the normal UML approach. The main extension provided in the proposed model is the use of priorities and importance of each use case in the proposed model. In the normal UML diagrams, all use cases are shown with equal priority. On the other hand, the proposed model distinguishes between modules with high priorities and modules with lower priorities.

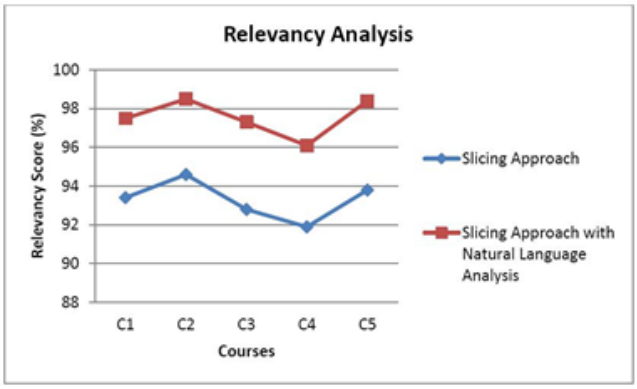

Figure 8: *Relevancy score analysis.*

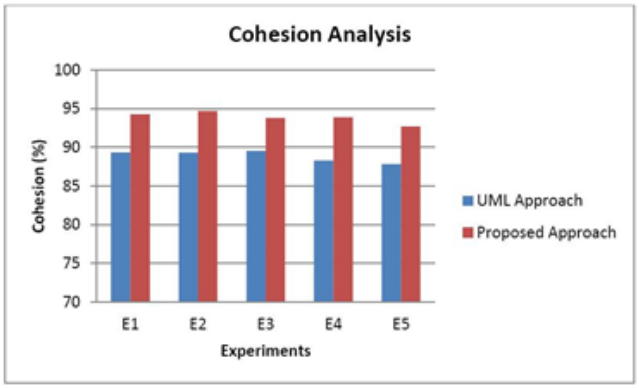

Figure 9: *Cohesion analysis.*

Figure 8 shows the relevancy analysis obtained after classifying the courses based on branch of study, level of students and specialization of faculty for student registration in the elective courses. The classification was performed using Multiclass Support Vector Machines. Here, two approaches were used for feature selection namely slicing approach and slicing approach with natural language analyzer. From the experiments conducted in this work, the results are obtained and are summarized in Figures 9 and 10.

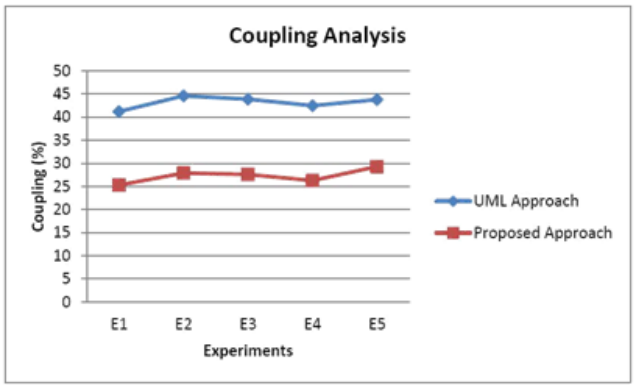

Figure 10: *Coupling analysis.*

From these figures, it is observed that both UML approach and the proposed approach provide good design with high cohesion and low coupling. However, the percentage of cohesion is more in the proposed approach than the UML approach due to the use of UML slicing and natural language analysis. Similarly, coupling percentage is also decreased in the proposed approach when it is compared with the UML approach. The natural language analyser analyzed the classes and identified the inheritance hierarchy more effectively by identifying the super classes and the corresponding sub classes. Therefore, the design document provides more accurate class diagrams and sequence diagrams leading to improvement in design documents. Moreover, the activity diagrams drawn using the proposed model are more accurate when it is checked with the course registration application. All these improvements are due to the presence of UML slicing and natural language analysis.

Conclusion and Future Work

In this paper, we proposed a medical expert system along with a facility for medical college registration which is useful for automatic design of object oriented applications by using SRS documents and use case diagrams. It also provides a facility for analysing asthma disease and recommends a suitable treatment plan. For this purpose, an existing slicing technique is used which performs effective slicing of use case diagrams. Moreover, a new feature selection algorithm is proposed in this work which uses slicing and information gain values with rules to assign priorities in order to select the most relevant features. The use cases are classified using Multiclass Support Vector Machines which is used to form rules to make the design principles. In addition, a natural language analyzer feature is provided in this work to perform effective analysis of SRS documents. Finally, design documents and necessary reports are generated and the system has been tested using a course registration application for medical students. From the experiments conducted in this work, it is observed that the slicing and natural language analysis methods used in this work help to develop a system that provides automatic design documents with high cohesion and low coupling. Future works in this direction can be the use of fuzzy logic to make more effective decisions.

References

1. Booch G, Maksimchuk RA, Engle MW, Young BJ, Conallen J. Object oriented analysis and design with applications. Pearson Edu (3rd edn.) 2010.
2. Wu CS, Khoury I. Web service composition: From UML to optimization. 2013 Fifth International Conference on Service Science and Innovation, 2013.
3. Zhu LZ, Kong FS. Automatic conversion from UML to CPN for software performance evaluation. Procedia Eng 2012; 29: 2682-2686.
4. Jaiprakash TL, Mall R. A dynamic slicing technique for UML architectural models. IEEE Transact Software Eng 2011; 37: 735- 771.
5. Lallchandani R, Mall JT. Integrated state-based dynamic slicing technique for UML models. IET Software 2010; 4: 55-78.
6. Sethukkarasi R, Ganapathy S, Yogesh P, Kannan A. An intelligent neuro fuzzy temporal knowledge representation model for mining temporal patterns. J Intell Fuzzy Syst 2014; 26: 1167-117.
7. Ganapathy S, Sethukkarasi R, Yogesh P, Vijayakumar P, Kannan A. An intelligent temporal pattern classification system using fuzzy temporal rules and particle swarm optimization. Sadhana 2014; 39: 283-302.
8. Ganapathy S, Yogesh P, Kannan A. Intelligent agent based intrusion detection system using enhanced multiclass SVM. Comput Intell Neurosci 2012; 2012: 1-10.
9. Calheiros RN, Masoumi E, Ranjan R, Buyya R. Workload prediction using ARIMA Model and its impact on cloud applications' QoS. IEEE Transact Cloud Comput 2015; 3: 449-458.
10. Kusumoto S, Mattukawa F, Inoue K. Estimating efforts by use case points: method, tool and case study. Proc 10th Int Symp Softw Metr 2004; 1-8.
11. Swain SK, Mohapatra DP, Mall R. Test case generation based on use case and sequence diagram. Int J Software Eng 2010; 3: 32-39.
12. Berkovich M, Leimeister JM, Hoffman A, Kremer H. A requirements data model for product service systems. Requirements Eng J 2014; 19: 161-186.
13. Carnevali L, Ridi L, Vicario E. A quantitative approach to input generation in real time testing of stochastic systems. IEEE Transact Software Eng 2013; 39: 292-304.
14. Kishore H, Anjan G. Software product engineering maturity model. CSI Communications 2015.
15. Alrajeh D, Kramer J. Elaborating requirements using model checking and inductive learning. IEEE Transact Software Eng 2013; 39: 361-383.
16. Rosa L, Lopes A. Self-management of adaptable component based applications. IEEE Transact Software Eng 2013; 39: 403-421.
17. Cravo G. Workflow modelling and analysis based on the construction of task models. Sci World J 2015.

18. Oh H, Baik JM, Lim SH. A model independent S/W Framework for search based software testing. Sci World J 2014; 2014: 1-11.
19. Prackweiser C, Buchmann R, Grossman W, Karagiannis D. Overcoming heterogeneity in business process modeling with rule based semantic mappings. Int J Software Eng Knowledge Eng 2014; 24: 1131-1158.
20. Munoz M, Mejia J, Hurtado GG. A methodology for establishing multi-model environments in order to improve organizational software processes. Int J Software Eng Knowledge Eng 2014; 24: 909-933.
21. Vieria MR, Chino FJT, Caetano TJR, Triana AJM. A visual framework to understand similarity queries and explore data in metric access methods. Int J Business Intell Data Mining 2015; 5: 370-397.
22. Ionita D. UML in business administration. J Knowledge Manage Econom Informa Technol 2011; 1: 1-15.
23. Bano M, Ikram N, Niazi M. Requirements engineering challenges in service oriented software engineering: an exploratory online survey. Int J Software Eng 2013; 6: 21-43.
24. Pasupathy S, Bhavani R. Analysing the efficiency of program through various load metrics. J Theor Appl Informa Technol 2014; 61: 346- 351.
25. Pereira CT, Martinez LI, Favre LM. Recovering use case diagrams from object oriented code: an MDA based approach. Int J Software Eng 2012; 5: 3-23.
26. Elgammal A, El-Sharkawi M. Using UML to model web services for automatic composition. Int J Software Eng 2010; 3: 87-113.
27. Kumar AG, Babu DR. Design and application of new quality improvement model: kano lean six sigma for software maintenance Project. Arabian J Sci Eng 2016; 41: 997-1014.
28. Ranganath VP, Hatcliff J. Slicing concurrent java programs using Indus and Kaveri. Int J Software Tools Technol Transfer 2007; 9: 489-504.
29. Chen JL, Wang FJ. Flow analysis of class relationships for object oriented programs. J Informa Sci Eng 2000; 3: 24-31.
30. Kim T, Song YT, Chung L, Huynh DT. Software architecture analysis: a dynamic slicing approach. J Comput Informa Sci 2000; 1: 91-103.

***Correspondence to:**

B Lalitha

Rajalakshmi Institute of Technology

Anna University

India