American Journal of Systems and Software, 2014, Vol. 2, No. 2, 50-55
Available online at http://pubs.sciepub.com/ajss/2/2/3

DOI:10.12691/ajss-2-2-3

Improving Automatic Exams Using Generic UML Model for Better Analysis and Performance Evaluation

Muhammad Rashid Naeem*, Weihua Zhu, Adeel Akbar Memon, Muhammad Tahir

School of Software Engineering, Chongqing University, Chongqing, P.R. China
*Corresponding author: rashidnaeem717@yahoo.com

Received May 03, 2014; Revised May 13, 2014; Accepted May 14, 2014

Abstract In this paper we are proposing a generic visual model for automatic exam system. We use UML as modeling language to model this system. This model is totally generic and extendable according to the requirements of any type of institution which conducts automatic exams. Furthermore, this model also helps in improving automatic exam system by showing performance analysis of students and evaluation using WEKA data mining methods.

Keywords: *automatic exams, UML, performance analysis, data mining, visual modeling*

Cite This Article: Muhammad Rashid Naeem, Weihua Zhu, Adeel Akbar Memon, and Muhammad Tahir, "Improving Automatic Exams Using Generic UML Model for Better Analysis and Performance Evaluation." *American Journal of Systems and Software*, vol. 2, no. 2 (2014): 50-55. doi: 10.12691/ajss-2-2-3.

1. Introduction

Exams aim to be the primary level of student performance assessment. It helps to know effectiveness of the teaching methods. Exams show level of validity and reliability. A valid and reasonable exam consists of multiple queries in form of difficultly and easiness. The positive and negative queries submissions are used to distinguish between the worst and best students. As we know, the manual system of exams is very costly and time consuming and with the increase in number of student's appearances in universities and colleges exams, it is very difficult to manage and work with the paper system which results in difficulty of correctly verify student performance evaluation with maximum accuracy. On other hand, computer brings revolution in the field of study and nowadays computer became an essential part of study. This leads to the need of automatic exam to evaluate student's performance. In next section we briefly discuss related work to the student examination and evaluation system. In 3rd and 4th section we discuss system structure of our model and use UML to model exam system in generic way. In 5th section we analyze and evaluate student performance data using data mining classification and prediction methods etc.

2. Related Work

In [1] N. Omara et al describes the Blooms Taxonomy cognitive domain to verify student cognitive level during examination. Bloom taxonomy consists of 6 levels such as knowledge, comprehension, application, analysis, synthesis and evaluation. This can be a good way to reduce difficulty in performance analysis of students but still there is need of automatic examination systems for students' evaluation due to their increasing appearances in exams.

Recently M. Yagci and M. Unal in [2] carried out a design and application of an adaptive exam system. They suggest an Enhanced web-based adaptive exam system which consists of three types of users is Administrator, Lecturer, Student, where students' data evaluation was carried out through statistical methods and assessments.

G. Frosini et al in [3] describe a tool to build software systems to replace role of examiner during typical exam to reduce effort of anxiety and use computerized adaptive testing to increase assessment efficiency. According to their model Database consists of queries submitted by author, Design module handles insertion of the queries, analysis module analyze results gathered during the exams and calculates particular indexes to determine easiness and difficulty of submitted quires.

E. Desouza, M. Fleming in [4] discusses the effectiveness of online exam comparing to paper exam. They also describes that students gain better results in online exams.

The online system has many benefits but it also have some disadvantages. M. A Sarrayrih, M. Ilyas discusses in [5], the Challenges of Online Exam and problems such as personal identity, unauthorized network interference.

From [1,2,3,4,5] we conclude that automatic exams is need of current education systems and for student evaluation it should be build on generic basis as well as it requires to handle security issue and need better performance evaluation algorithms.

3. Automatic. Exam Model & System Structure

For this model, we use UML to design our software system. UML is the graphical notation language to give visual model of software solution. UML provides an understandable way to programmer to create software solution. [6,7] describes the benefits and use of UML at distributed environments and their consistency. Here, following UML diagrams are used to model generic exam system. We use Use-case Diagram, Activity Diagram, Sequence Diagram and Class Diagram to model our solution.

From the paper exam point of view, we assume that the following structure consist necessary requirements for an automatic exam system to work. The Figure 1 demonstrates the environmental structure of system and how software interacts with different components which are part of a larger system.

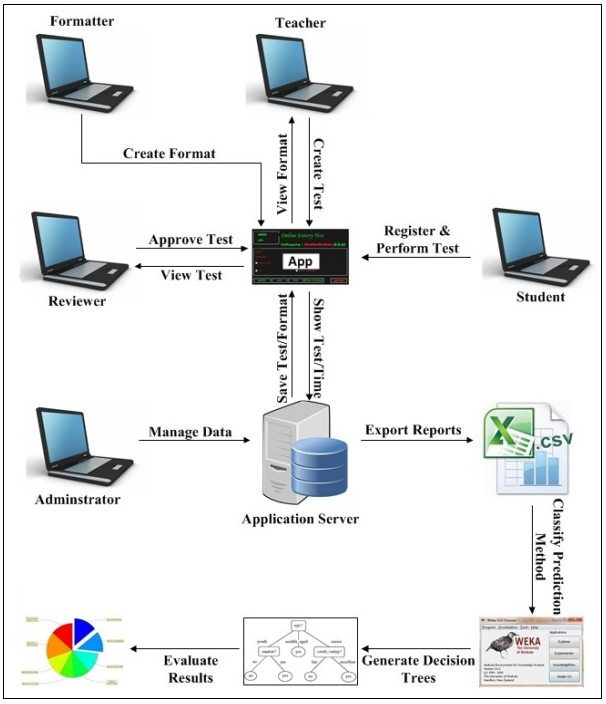

Figure 1. System Structure of Automatic Exam Model

In Figure 1, software is connected to the central server. Server is responsible for handling data and input/output and saving queries by students' into CSV file, where administrator is responsible for managing data traffic in server. As a generic exam system we have 4 people who are directly interacting with the app. Formatter will be responsible to set exam format according to institution exam pattern. Teacher is responsible for creating questions & answers according to format. Reviewer will give review for test then test will available for student. The queries submitted by students will be sent to WEKA Tool (data mining software) for performance evaluation. M. Hall in [8] discusses the success of WEKA in data mining field as well as customized GUI provided in WEKA 3.4 and above with extensibility, to integrated WEKA with programming languages.

4. UML Model

4.1. Use-case Diagram of Automatic Exam System

Use-case diagram shows simple interaction of user with the system and what kind of interaction they have with the system. Use-case diagram is also quoted as "Blue Prints of the System". It presents the simplest graphical representation of what system must do. It consists of basic requirements of software system. In [9] W. Shen, S. Liu discusses the formalization and usage of Use-case Diagram. They also discuss how use-case diagram can be used to verify potential errors in requirement model and reduce time, labor and expenditures in software development. Figure 2 demonstrates a generic interaction of users with the system.

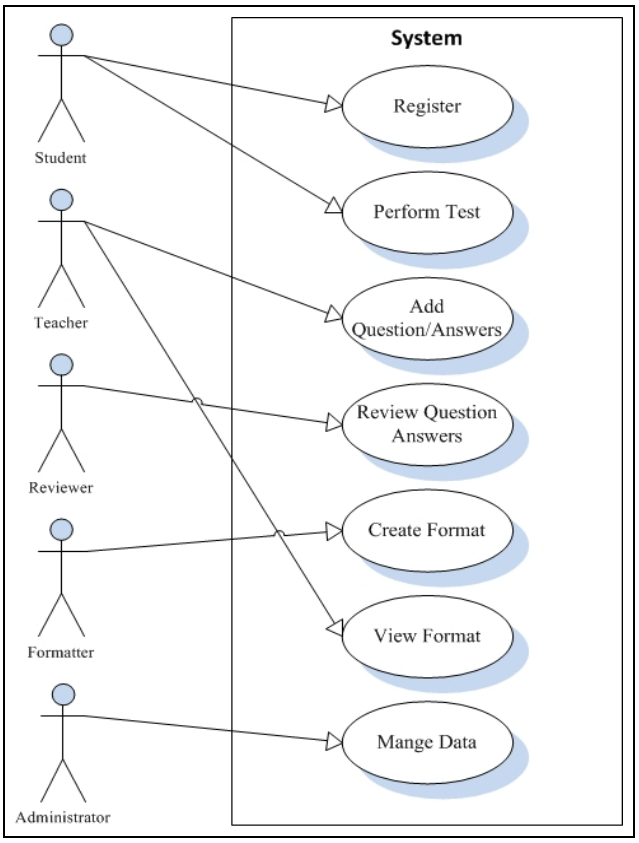

Figure 2. Use-case Diagram of Automatic Exam Model

4.2. Activity Diagram of Automatic Exam System

Activity Diagram is basically flow chat. It shows the graphic presentation of stepwise set of activities performed by actors with support of choice, iteration etc. They can be regarded as a form of flow charts but the basic difference between activity and flow chart diagram is flow chart lacks in expressing concurrency. Besides this activity diagram shows dynamic aspect of the system. Figure 3 Demonstrate the activity diagram of automatic exam system. In this figure we can see the basic flow of automatic exam system between different actors i.e. Student, Teacher, Reviewer, Formatter.

As we can see in the following activity diagram in Figure 3, there are four main actors in the system where Student do test registration, perform test and view report activities. In [10] R. M. Bastos el al describes the benefits and workflow modeling of activity diagrams in production systems. In most of automatic exam systems, there is always an initial result therefore it will be generated on the

spot when student finishes exam. After that collection of result is used for evaluation and selecting good student.

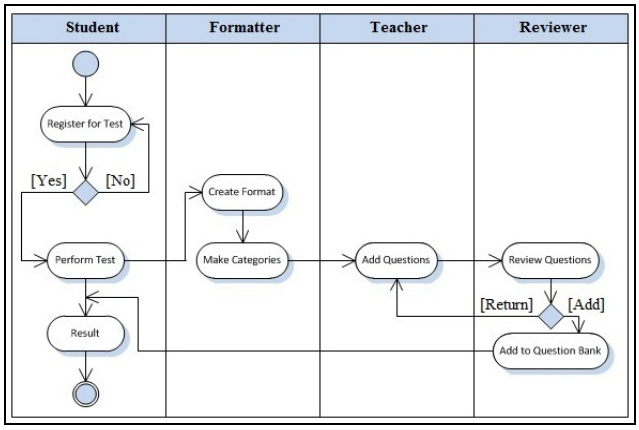

Figure 3. Activity Diagram of Automatic Exam Model

4.3. Sequence Diagram of Automatic Exam System

Sequence Diagram is used to graphically represent the software processes in detail. It shows how processes operate with each other and how they make interactions dynamically. Basically the sequence diagrams show the logical view of system under development. They are also known as event diagrams and event scenarios. They represents event scenario in details and objects lifeline in graphical order. In our design model we use sequence diagram to show detail processes of our system. We also use MVC (Model-View-Controller) pattern to implement our system.

MVC is the most popular model for software development in software industry. As it separates model and view in separate packages or logically and use controller to make connection between both. It makes code more flexible reusable and easier for understanding.

In [11] A. Holzinger, K. H. Strugg describes the benefits of using MVC as software design paradigm in terms of software confidence and user satisfaction.

Figure 4 is demonstrating the sequence of processes and interaction between processes in detail. The controller class is interacting with the model classes to achieve the functionality of MVC pattern. Format, Category, Question, Review, Register, Test are model classes in exam system. In Figure 4, there is a class WEKA which is actually an external utility class contains methods from WEKA Tool (Data mining Software). As we know, WEKA is best known open source tool for data analysis especially when data is so big, as discussed by M. Hall in [8]. We need to implement data mining because there could be thousands of students who will take exam and it's very difficult to choose best of them and categorized and visualize their performances. Therefore, we use WEKA Tool. In our model, Test class passes collection of datasets into WEKA utility Class as "*CVSExport()*" method for further analysis.

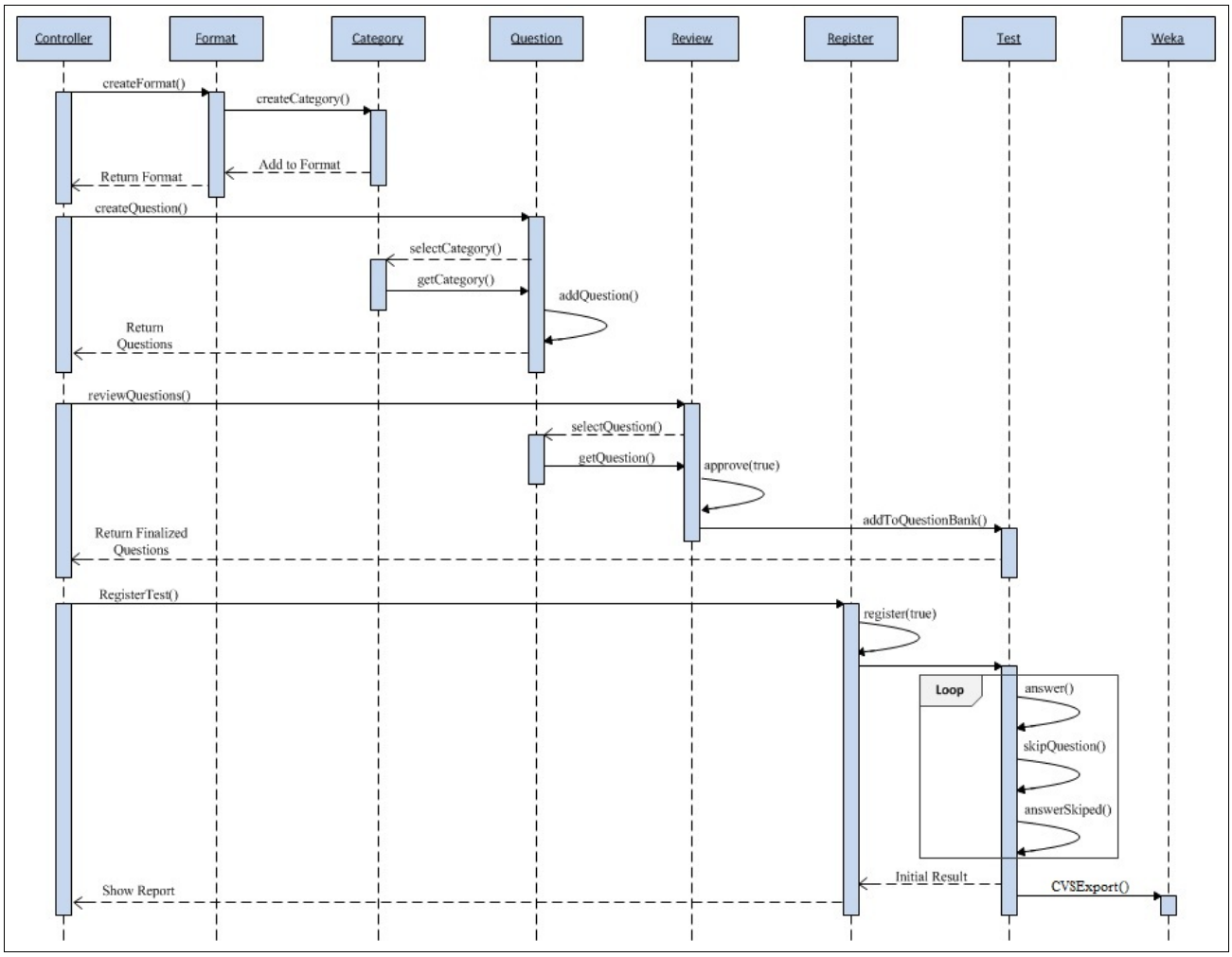

Figure 4. Sequence Diagram of Automatic Exam Model

4.4. Class Diagram of Automatic Exam System

Class diagram describes the static structure of the system. It consists of classes, attributes and operations. It also describes the relation among the classes. The class diagram gives a structure of software that how different classes interact with each other. It also provides blue prints of the system structure to the maintenance team before they start inspecting of actual code.

In [12], K. Robles illustrate how to facilitate retrieval process in software development using class diagram and techniques to reuse them at design stage rather than in coding.

Figure 5 demonstrates the Class Diagram of automatic exam system. In this diagram, Controller class, as we mentioned in section 4.3 that we use MVC design approach, therefore it is responsible as a medium between model classes and interface. There is also a utility class WEKA containing data analysis methods for example Classifications, Associations, and Decision Trees etc. Actually these methods are defined by WEKA Tool and they could be different and many more. This feature to use a component into class diagram is defined in UML 2.1 and above as stereotypes. In [13] IBM Rational Software Modeler Version 7.0.5, under UML Model Element stereotype section, describe that stereotype *<<utility>>* can be used as model element *class* that doesn't have *instances*, whose *attributes* and *operations* have class scope. Using this way, we add data mining operations of WEKA Tool in our Model and ensures usability of data mining. In next section we will describe how to benefit from WEKA Data mining techniques in our model.

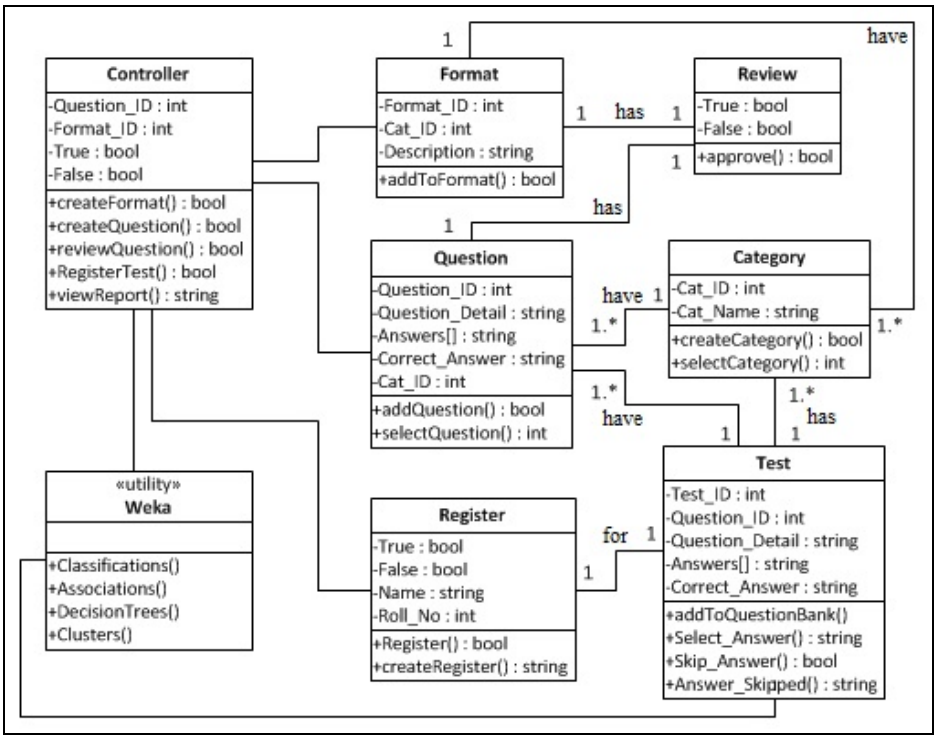

Figure 5. Class Diagram of Automatic Exam System

5. Results Analysis & Performance Evaluation

As we mention in section 3 that after getting results from students' exam, system will export datasets into CSV file and send it to WEKA Tool by utility class *WEKA* to apply data mining operations on that data as described in Figure 4 and Figure 5 sequence and class diagrams respectively. Table 1 demonstrates the sample student results data generated by system after finishing exam. In coming subsections we are going to use sample student data generated by automatic exam system to analyze using WEKA Data mining operations.

Table 1. Sample student dataset generated by automatic exam system

Roll #	Student Results					
	Student	Department	General Sc.	Math's	Computer	Remarks
1	Agarwal V	Computer Sc	78	80	66	Pass
2	Agnew P	Computer Sc	45	78	66	Pass
3	Ahmad H	Statistics	78	30	50	Fail
4	Alambritis G	Physics	65	68	70	Pass
5	Allen J	Economics	87	50	60	Pass
6	Anthony A	Math's	54	30	30	Fail
7	Antoniou V	Computer Sc	76	70	70	?
8	Archy J	Economics	76	40	40	Fail
9	Armstrong B	Math's	86	30	40	?
10	Arvanitakis I	Physics	68	70	80	Pass

5.1. Classification Rule

In classification, data is mapped into predefined groups and classes from unorganized data. Classification helps to predict missing values in dataset. Table 1 illustrate sample dataset of student exam. In this table as we see at Roll No 8 we have missing Remarks value. Using classification we can predict the missing value using methods provided by WEKA Tool. First it takes *training set* from Table 1 and predicts missing values. For predicting values it uses *training set* to create prediction model and then use *test set* to determine the accuracy of prediction. *J48* classifier of WEKA is considered to be best method for classifying unorganized data and especially for making decision trees. After performing classification on our student result data we get following summary as shown in Table 2.

Table 2. Accuracy of predicted results for student data

Summary of Classification		
Correctly Classified Instances	9	90.1034 %
Incorrectly Classified Instances	1	6.8966 %
Kappa statistic	0.8619	
Mean absolute error	0.069	
Root mean squared error	0.2626	
Relative absolute error	13.7931 %	
Root relative squared error	52.4816 %	

On other hand, decision trees provides a clear indication that which fields are most important and shows a prefect view of data flow in form of trees. Figure 6 demonstrates the decision tree generated from Student Data.

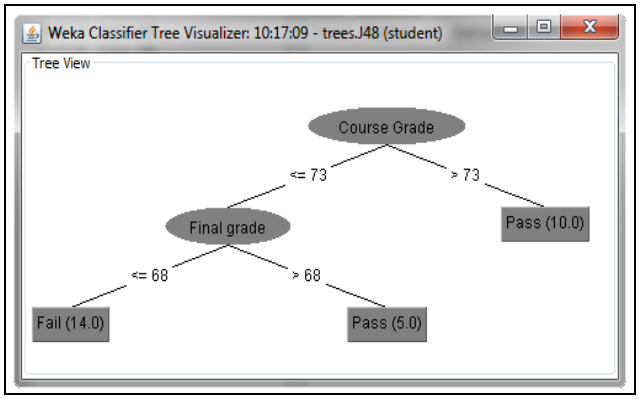

Figure 6. Tree Structure generated by WEKA Tool

5.2. Association Rule

In terms of data mining, association rules are used to find best relationship among data items. Apriori association algorithm in WEKA is considered to be the good for finding associations between datasets elements. Apriori compute all possible rules and finds the best with minimum support and minimum confidence between rules. In automatic exam system, using association rules, we can find best combinations of students on the basis of department and student Remarks. From Table 1 we get the following associations rules:

Minimum support: 0.25 (2 instances)
Minimum metric <confidence>: 0.9
Best rules found:
1. Department=Computer Sc 3 ==> Remarks=Pass 3
2. Math's=30 3 ==> Remarks=Fail
3. Computer=66 2 ==> Department=Computer Sc 2
4. Department=Physics 2 ==> Remarks=Pass 2
5. Department=Math's 2 ==> Math's=30 2
6. Department=Math's 2 ==> Remarks=Fail 2
7. Math's=70 2 ==> Remarks=Pass 2
8. Computer=40 2 ==> Remarks=Fail 2
9. Computer=66 2 ==> Remarks=Pass
10. Computer=70 2 ==> Remarks=Pass 2

5.3. Clustering

Clustering plays an important role in field of data mining. In clustering we find group of objects that are similar to one another but they are different from objects in other groups. Clustering helps in automatic exam system to distinguish between good and bad performance in exam and characterized them in groups. We use WEKA KMeans algorithm to create clusters of student sample data in Table 1. Table 3 describes the clusters generated by KMeans algorithm.

Table 3. Sample student dataset generated by automatic exam system

Attribute	Cluster Cancroids by KMeans		
	Data (10)	Cluster # 1 (6)	Cluster # 2 (4)
Student	Agarwal V	Ahamad H	Agarval V
Department	Computer Sc	Physics	Computer Sc
General Sc.	76	54	76
Math's	30	30	40
Computer	40	30	66
Remarks	Pass	Pass	Pass

Besides data mining techniques, we can also analyze student result data using graphical presentations as we discussed in section III in our system structure.

6. Conclusion

In this paper, we propose a generic model of automatic examination system using UML Diagrams. We hope this model not only helps institutions to implement automatic exam system but also facilitate its customization and extendibility according to the institutional environment.

Furthermore, having integration of this system with WEKA tool, it can be more suitable and efficient system for predicting students' performances and success. Currently WEKA is working as a component of Automatic Exam System. In future, we plan to make this system more efficient and merge WEKA functionality within our software.

References

[1] N. Omara, S. S. Harisa, R. Hassana, H. Arshada, M. Rahmata, N. F. A. Zainala, R. Zulkiflib, Automated analysis of exam questions according to bloom's taxonomy, *Procedia - Social and Behavioral Sciences*, 59 (2012) 297-303.

[2] M. Yagci and M. Unal, Designing and implementing an adaptive online examination system, *Procedia - Social and Behavioral Sciences*, 116 (2014) 3079-3083.

[3] G. Frosini, B. Lazzerini, F. Marcelloni, *Performing automatic exams, Computers & Education*, 1 June 1998.

[4] E. Desouza, M. Fleming, A Comparison of In-Class and Online Quizzes on Student Exam Performance, *Journal of Computing in Higher Education*, Vol. 14(2), pp. 121-134.

[5] M. A Sarrayrih, M. Ilyas, Challenges of Online Exam, Performances and problems for Online University Exam, *IJCSI International Journal of Computer Science Issues*, Vol. 10, Issue 1, No 1.

[6] H. Shen, Maintaining constraints of UML models in distributed collaborative environments, *Journal of Systems Architecture*, Volume 55, Issues 7–9, July–September 2009, Pages 396-408.

[7] F. J. Lucas, F. Molina,A. Toval, A systematic review of UML model consistency management, *Information and Software Technology*, Volume 51, Issue 12, December 2009, Pages 1631-1645.

[8] M. Hall, The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, Volume 11, Issue 1, Page 10-18.

[9] W. Shen, S. Liu, Formalization, Testing and Execution of a Use Case Diagram, *Formal Methods and Software Engineering*, Volume 2885, 2003, pp 68-85.

[10] Bastos, R.M, Ruiz, D.D.A, Extending UML activity diagram for workflow modeling in production systems, *System Sciences*, 7-10 Jan. 2002, 3786-3795.

[11] A. Holzinger, K. H. Struggl, Appling Model-View-Controller (MVC) in design and development of information systems, *e-Business (ICE-B), Proceedings of the 2010 International Conference*, 26-28 July 2010, Pages 1-6.

[12] K. Robles, A. Fraga, J. Morato, J. Llorens, Towards an ontology-based retrieval of UML Class Diagrams, *Information and Software Technology*, Volume 54, Issue 1, January 2012, Pages 72-86.

[13] IBM, *IBM Rational Software Modeler Version 7.0.5 Information Center*, IBM Corporation, 2005-2007, [E-Book] Available: IBM Support Portal.