Implementation of Combined Dynamic Exam Seating Plan

Ma Kyi Thu Zar[1], Daw Yin Thida San[2]

***Abstract*—**This paper described the implementation of combined dynamic exam seating plan in technological universities. The number of students is significantly increasing year by year in any universities. The exam seating plan is one of the most important needs for every staff in universities .Due to the problems in exam seating plan, the performance of a combined dynamic exam seating plan is hugely effected to staffs. In this paper ,the combined dynamic exam seating plan is performed by rule-based system. A rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is expressed to match different majors and different years, different majors and same years, and same major and different years. The computerized Information System is then used that eliminates the difficulities and time consuming tasks in manual system. This Information System also provides effectiveness in management and decision support levels for implementing combined dynamic exam seating plan. The system is implemented by using C# programming language and Microsoft Access database software.

Keywords—Information System, Database, Rule-based System, Exam Seating

I. INTRODUCTION

NOWADAYS, there are so many universities and colleges in Myanmar. Technological universities become the vital area to improve the development of nation. Most people interested in this university to study as professional in their lives. Today, there are many students in this technological universities. As the students increased, there are many difficulties and problems to plan the combined dynamic exam seating plan. Every staff in exam seat plan solve these problems with carefully and difficulty.

In this paper, it is easy to solve for staff that face in combined dynamic exam seating plan. An exam seating plan is a complex function of the staff and records all students in all majors. The staff must know the number of students, the number of different majors, the number of classes and the number of desks. The exam seating plan is variant in time, therefore adaptive templates or methods are necessary.

This paper focuses on implementing of combined dynamic exam seating plan by using rule-based identification. A mathematical procedure is a popular method to realize the computerized exam seat planning system. Then this system outlines the requirements of seating plan and indicates how rule-based technology can be applied and implemented to accomplish those requirements.

II. APPLICATION AREA

Normally, the system is designed on Personal Computer based computerized Information System. By using personal computer in computerized Information System, the overall cost of application system can be reduced. In this system, there are six steps to arrange exam seating plan in dynamically.

Step 1: Set exam time
Step 2: Set total students per each room.
Step 3: Select the major and academic year to match.
Step 4:While no end of students from one of selected majors.
Do match and realize seating order with rule-based system.
If end of students from selected one major, then goto Step 5.
Step 5: Select different major or same major of different year randomize, then goto Step 4 to continue the system.
Step 6: If all students are finished then stop the system and output result with spread sheets for each room.

III. STUDIES ON MATHEMATICAL THEORY

In this section, Mathematical Theory related to the computerized exam seating plan will be discussed. The general concepts of probability and combination are also discussed in this section.

In probability theory, it is a way of assigning every event a value between zero and one, with the requirement that the event made up of all possible results (in this example, the event {1, 2, 3, 4, 5, 6}) is assigned a value of one.

1.Kyi Thu Zar (ME IT-7), Technological University (Monywa), (kyithuzar09@gmail.com).

2.Yin Thida San(Assistant Lecturer), Department of Information Technology, Technological University(Monywa).

The probability of an event A is written as P(A), P (A) or Pr (A). This mathematical definition of probability can extend to infinite sample spaces, and even uncountable sample spaces, using the concept of a measure.

The opposite or complement of an event A is the event [not A] (that is, the event of A not occurring); its probability is given by P (not A) = 1 - P (A). As an example, the chance of not rolling a six on a six-sided die is 1 – (chance of rolling a six) $= 1 - \frac{1}{6} = \frac{5}{6}$.

If both events A and B occur on a single performance of an experiment, this is called the intersection or joint probability of A and B, denoted as $P(A \cap B)$.

In the discussion, a combination containing k objects is a subset containing k objects. Develop a formula for computing the number of combinations of n objects taken k at a time without actually listing the combinations or subsets. The number of combinations of n objects taken k at a time is denoted $_nC_k$..

COMBINATION	PERMUTATIONS					
{A,B,C}→	ABC	BCA	CAB	CBA	BAC	ACB
{A,B,D}→	ABD	BDA	DAB	DBA	BAD	ADB
{A,B,E}→	ABE	BEA	EAB	EBA	BAE	AEB
{A,C,D}→	ACD	CDA	DAC	DCA	CAD	ADC
{A,C,E}→	ACE	CEA	EAC	ECA	CAE	AEC
{A.D.E}→	ADE	DEA	EAD	EDA	DAE	AED
{B.C.D}→	BCD	CDB	DBC	DCB	CBD	BDC
{B,C,E}→	BCE	CEB	EBC	ECB	CBE	BEC
{B.D.E}→	BDE	DEB	EBD	EDB	DBE	BED
{C,D,E}→	CDE	DEC	ECD	EDC	DCE	CED

Table. 1 Combination table

IV. THE PRINCIPLE OF THE RULE-BASED SYSTEM

This involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied. A general algorithm for this might be:

while values for attributes remain to be input
  read value and assign to attribute
  evaluate conditions
  fire rules whose conditions are satisfied

An inference engine for a rule-based system whose basic components are:

Attributes: X1, X2 , ... , Xn1
Conditions: C1, C2, ... , Cn2
Rules: R1, R2, ... , Rn3
Actions: A1, A2, ... , An4

For rules such as:
R1: if X1 > 100 then A1;
R2: if X1*0.6 < X2 + 50 and X2 < 43 then A2, A3;

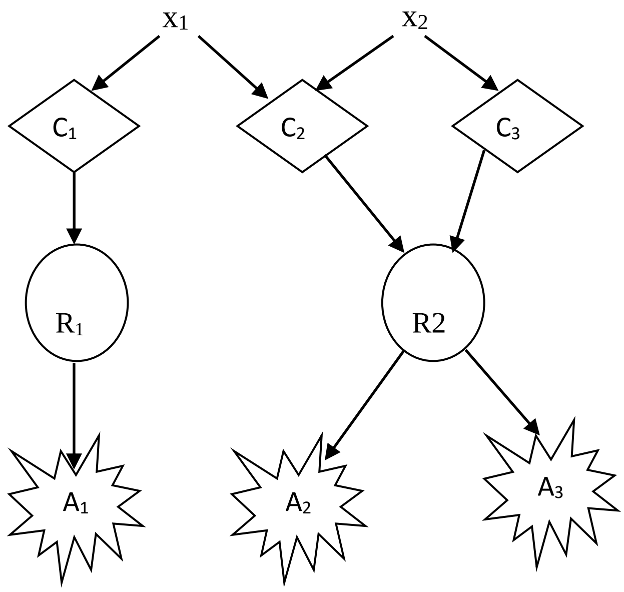


Figure. 1 Relationship Graph

Form the conditions:
C1: X1 > 100
C2: X1*0.6 < X2 + 50
C3: X2 < 43

In using IF-THEN rules for seating plan, a rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form IF condition THEN conclusion. An example is rule R1,

R1: IF major=different AND year=different THEN match-seat=yes.

The "IF"-part (or left-hand side) of a rule is known as the rule antecedent or precondition. The "THEN"-part (or right-hand side) is the rule consequent .

In the rule antecedent, the condition consists of one or more attribute tests (such as major=different, and year=different) that are logically ANDed. The rule's consequent contains a class. R1 can also be written as

R1: (major=different) ^ (year=different) → (match-seat=yes)

Moreover, another IF-THEN rules may express of the form IF condition THEN conclusion.

Expressions of rule R2 and rule R3 are:

R2: IF major=different AND year=same
THEN match-seat=yes.
R3: IF major=same AND year=different
THEN match-seat=yes.

Rule-based systems, as defined above, are adaptable to a variety of problems. In a forward changing system, the initial facts are processed first, and keep using the rules to draw new conclusions given those facts.

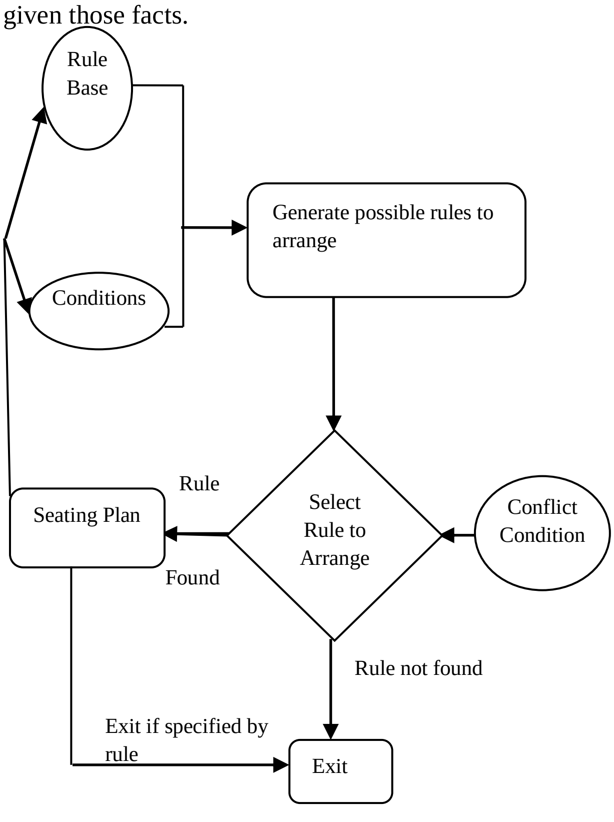


Figure. 2 Forward Changing

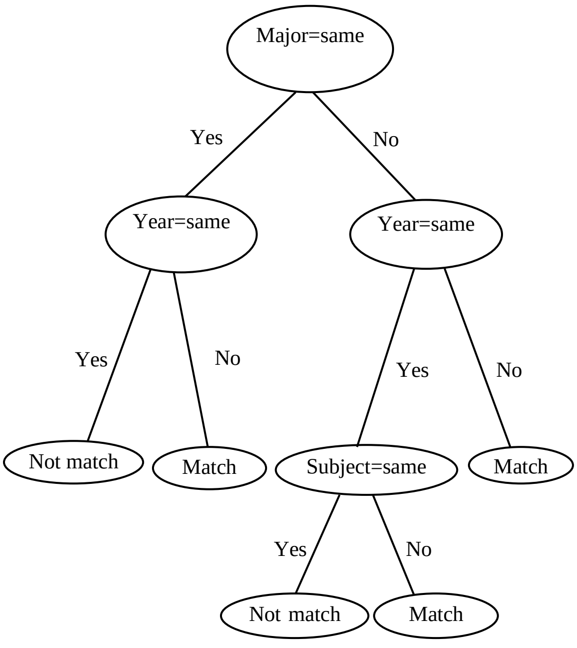


Figure 3.Decision Tree

In figure 3 semantic trees come a decision tree. Each node of a decision tree is linked to a set of possible solutions. Each parent node, that is each node that is not a leaf (and thus has children) is associated with a test, which splits the set of possible answers into subsets representing every possibility of the test's outcomes.

V. PROPOSED SYSTEM ARCHITECTURE

The system composes of three parts: firstly, input data is obtained from the user and stored to student database. After input process, user needs to define desire seating plan to rule-generator. Rule-generator will be stored as defined rule to rule database. Appropriate seating plan is arranged by using student database and rule database. The proposed system architecture is shown in Figure. 4.

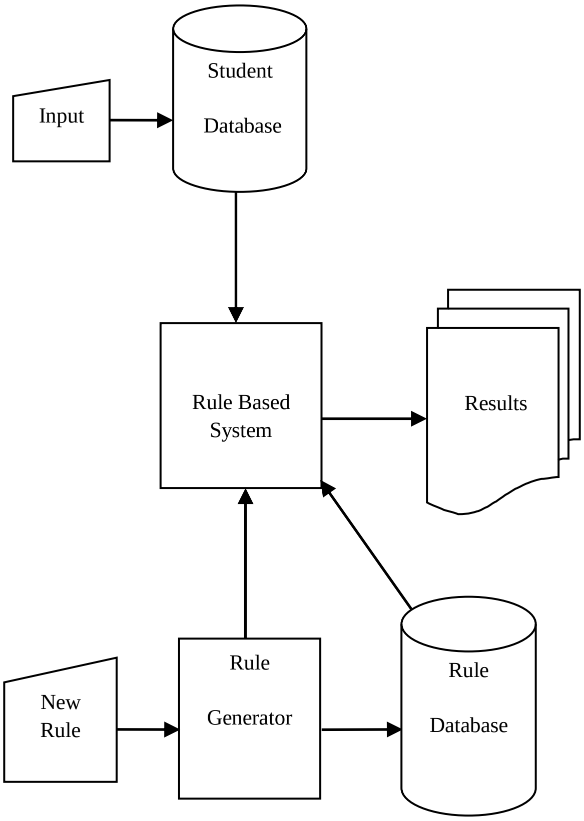


Figure. 4 Overall System Design

The exam seating information system is created as the integrated system, aiming to computerize all seating plan processes. All data are stored in individual database. Exam officer can be manipulating desire database separately. Processes use general university registers, general rule base classifications.

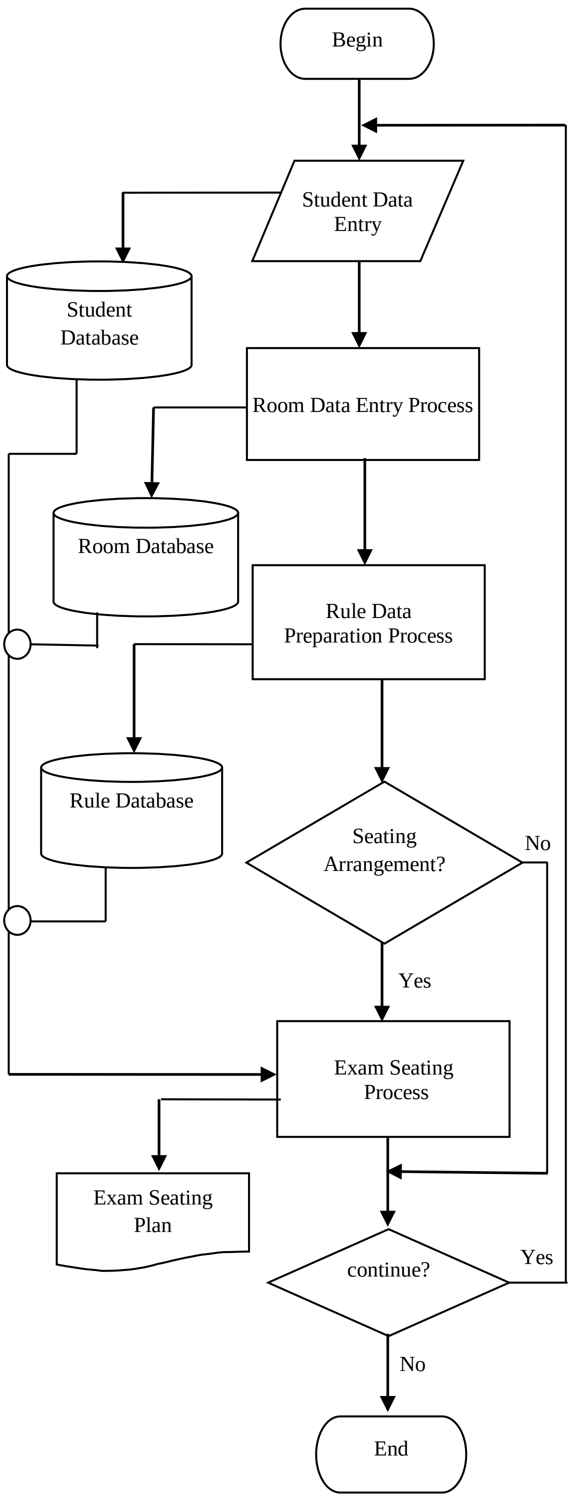


Figure. 5 Overall Program Flow of Exam Seating Plan

VI. TEST AND RESULT INTERFACES

The exam administrator not only enters student record but also updates the student's information (as shown in Figure. 6). In this section, the administrator needs to apply students' name and their roll number only. If the administrator wants to delete the existing student record, can remove in student table.

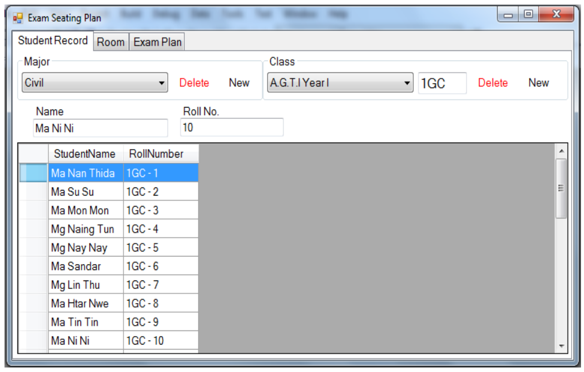


Figure. 6 Student Data Entry for A.G.T.I Year I of Civil.

In room preparation section, exam administrator can arrange room information such as room number, the number of students per each room. Single desk system is used in this system. Administrator can type desire room number and how to place desks in row and column lines in this room. Sample room arrangement result is shown in Figure. 7.

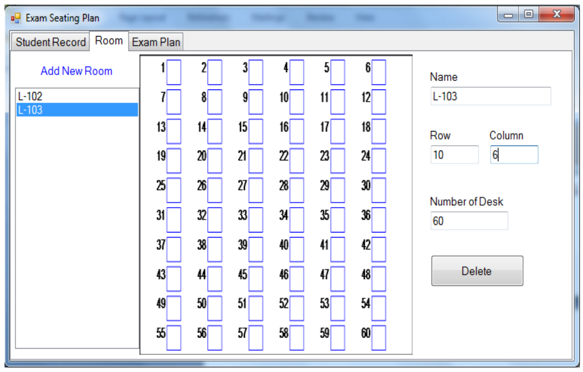


Figure.7 Room Data Entry

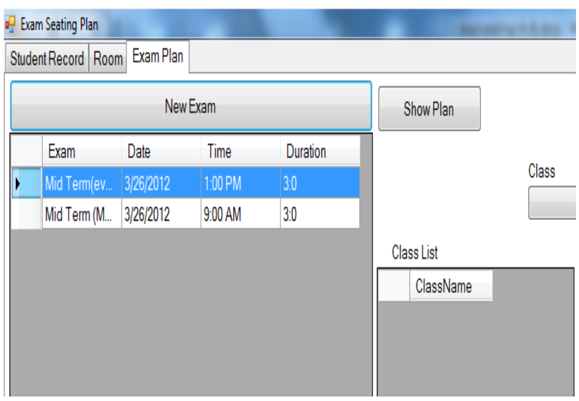


Figure. 8 After Finished New Exam Time

After set the exam data, administrator can view exam information in new exam table as shown in Figure 8.

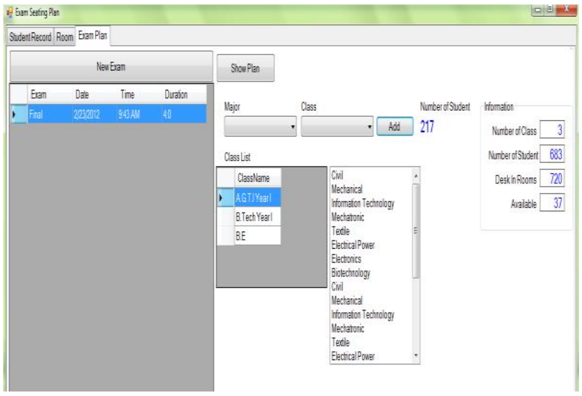
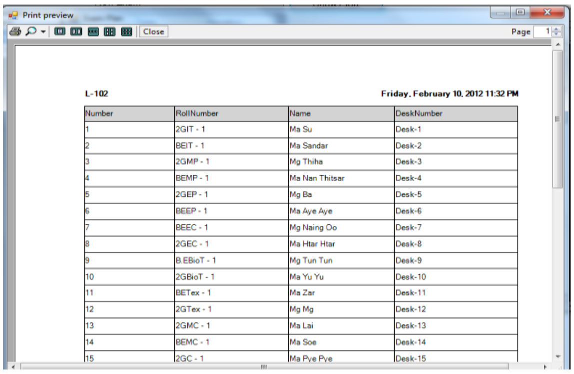


Figure. 9 Three Classes Pair of All Majors

This example implementation describes selecting three academic years of all majors for combine exam is shown in Figure. 9.

Figure.10 describes print preview of student list.



Number	RollNumber	Name	DeskNumber
1	2GIT - 1	Ma Su	Desk-1
2	BEIT - 1	Ma Sandar	Desk-2
3	2GMP - 1	Mg Thiha	Desk-3
4	BEMP - 1	Ma Nan Thitsar	Desk-4
5	2GEP - 1	Mg Ba	Desk-5
6	BEEP - 1	Ma Aye Aye	Desk-6
7	BEEC - 1	Mg Naing Oo	Desk-7
8	2GEC - 1	Ma Htar Htar	Desk-8
9	B.EBioT - 1	Mg Tun Tun	Desk-9
10	2GBioT - 1	Ma Yu Yu	Desk-10
11	BETex - 1	Ma Zar	Desk-11
12	2GTex - 1	Mg Mg	Desk-12
13	2GMC - 1	Ma Lai	Desk-13
14	BEMC - 1	Ma Soe	Desk-14
15	2GC - 1	Ma Pye Pye	Desk-15

Figure. 10 Print Preview of Student List

Figure.11 describes print preview of seating plan.

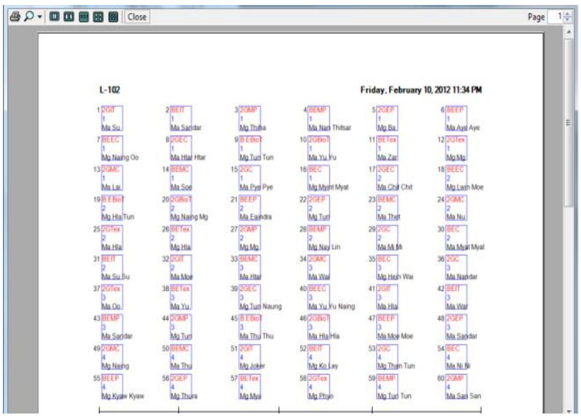


Figure. 11 Print Preview of Seating Arrangement

VII.CONCLUSION

Rule-based classification systems have been applied in a vast number of application areas. An important advantage of this exam seating arrangement is that the implementation is expressed as easy-to-understand. This paper has described a new method for developing more easily maintainable rule-based expert systems, which is based on dividing the rules into groups and concentrating on those facts that carry information between rules in different groups. This method was supported by a study of the connectivity of rules and constraint facts.

ACKNOWLEDGEMENT

The author is deeply grateful to his Excellency Minister U Aye Myint, Ministry of Science and Technology for opening Master of Engineering Course at Technological University (Monywa). The author is highly grateful to Dr. Wunna Swe, the Rector of the Technological University (Monywa) for his permission and suggestions for completion of this paper. The author would like to thank to Daw Yin Thida San, Assistant Lecturer, Department of Information Technology, Technological University (Monywa) for her kindness, graceful attitude and permission to do this paper.

REFERENCES

[1] Boyer, R. S. and Moore, J. S, A Fast String Searching Algorithm, Communications of the ACM, 20 (1977).

[2] Burgi, Mark, Interpretations of Negative Probabilities, 2009, p.1.

[3] Franklin, J., The Science of Conjecture: Evidence and Probability Before Pascal, Johns Hopkins University Press. (2001). pp. 22, 113, 127

[4] Gorman, Michael. "Management Insights". Management Science, 2011.

[5] Ivancevic, Vladimir; Tijana Ivancevic. "Quantum Leap". 2008. p 16

[6] Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques (Second Edition)

[7] http://en.wikipedia.org/wiki/Probability.htm

[8] http://www.slideshare.net/Covering (Rules-based)Algorithm.htm