Decision Support System on Adaptive Examination System

Rukli
Muhammadiyah University of Makassar
Makassar, Indonesia
rukli@unismuh.ac.id

Abstract. This study aims to create an adaptive testing system based on user preferences where user references can be adapted to the test system. An adaptive examination system is added to the decision support system feature where a minimum domain and quota are received from the highest score order of the test results. Research uses data from the domains of mathematics, Indonesian, and English. Response data processing using the Bilot_MG program the choice of a single parameter logistics model. Making applications using java programs. The results of the study produced the UniversiCAT application where the results of the CAT output test were processed into information for the user on the decision support system feature. Dynamic weighting of subject domains to produce a temporary ranking between all test examinees. The determination of graduation is determined by the rules of acceptance, namely the quota limit and the minimum score set by the school.

Keywords: Computerized Adaptive Testing, Decision Support System.

INTRODUCTION

Now, the millennium era engulfs the world of digitalisation and automation. Computer and network technology is experiencing a significant development leap including testing systems. Computer Based Testing (CBT) is increasingly admired while Paper and Pencil Test (PPT) is increasingly abandoned by users. PPT has limitations in terms of leakage level of exam questions, test examinees can work together during exams, printing costs are quite high, the level of readability of the printing results is sometimes blurred or damaged, late printing and problems arriving at the test location, not a few test administrators and administration, opportunities engineering test scores, and delays in exam results to the user.

Almost all limitations of PPT can be reduced or eliminated by CBT. For example, test questions do not need to be printed so that the leakage of exam questions and the cost of printing and securing questions can be reduced. The question of being kept server so that the sending of questions manually by sea, air and land is not necessary. Delays in sending data on exam questions due to reduced weather and natural disasters. The test results can immediately be printed and known so that there is no value engineering. However, in terms of measurement and assessment of changes in PPT to CBT, it is less functional because CBT compared to PPT is only limited to changes in media, namely from paper to computer. This has not paid attention to the core problem, namely the relationship between the ability of the test examinees and the characteristics of the test questions. The CBT version in Indonesia uses the term Computer Assisted Test then called the Computer-Based National Examination (CBNE) both using full online and semi-online networks for use in schools today.

Rightly, the design of giving questions to the test examinees experienced a shift according to the principles of measurement and assessment. that is, from giving questions to test examinees statically to the CBNE to giving questions to the test examinees dynamically. The dynamics in terms of the characteristics of the problem adapt to the ability of the current test examinees. The mechanism is planted in a computer in the form of an adaptive algorithm so as to produce a system known as Computerized Adaptive Testing (CAT). Several CAT applications have been applied, for example J-CAT (Imai, 2008), OAVTS (Wu, 2004), SIETTE (Guzman, 2005), and CAAS (Hoe, et al., 2005). However, the model is only limited to adaptive test algorithms and is internal in the process of finding adaptive questions. CAT has not paid attention to the output of the test examinees' response, ie the score procedure becomes information to the user.

Now, CAT has undergone engineering towards a multi-faceted model in giving questions to test examinees (Triantafillou, et al., 2006; Rukli, 2010; Rukli and Hartati, 2011). However, the sophistication of the test system model if it does not pay attention to the needs of the user, the model is ineffective in the technology industry revolution century 4.0. The revolution is a century of excellent
service based on user needs without insulation. One model in the information system related to the user reference is the Decision Support System (DSS).

According to Scoot-Morton (Turban, et al., 2001), DSS is a computer-based interactive system to help decision makers utilize data and models to solve unstructured problems. Turban, et al. (2001) provide limitations on DSS from basic to the most ideal, namely interactive, flexible and adaptive computer-based information systems to support unstructured management problem solving, improve the quality of decision making, use data, and allow decision makers, and use their insights own. So, DSS is a vehicle in accommodating user preferences according to the environmental objectives attached to it.

Furthermore, the media does not automate or replace users but provides space for users to improve their work efficiency. Thus, DSS is a computer-based information system that provides interactive information support for managers and business practitioners during the decision making process. The process uses analytical models, special databases, assessments and decision makers’ views, and interactive computer-based modeling processes to support semi-structured business decision making. Based on the description, it can be concluded that the DSS is a specific information system to assist managers, for example in the decision making selection of test examinees who will enter the school related to semi-structured problems with the help of computers.

The decision making process at the school level is the result of a process of communication and participation of the entire system. The whole can be an organizational system both internally and externally in the process of receiving new students. One of the main tasks of school level management is carried out by the principal and his staff is to maintain the existence and improve school performance in order to remain operating in an uncertain environment (globalization). To do this, principals are often faced with various alternative actions. Various alternatives are often counterproductive and an alternative action is needed to work together. Therefore, the existing steps must be arranged so that the frame model processes decision making both at the strategy, tactics and operational levels can be implemented effectively.

This is one direction with the life pattern of the technology industry process 4.0 which emphasizes efficiency but still leads to a more optimal goal. The same thing in the implementation of testing or adaptive testing where the user needs complete, fast, and accurate information and is relevant about the suitability of the test results report. The suitability can be translated as determining criteria and weighting according to user preferences.

Therefore, this study aims to describe the implementation of user references using a combination of CAT-DSS on the school entrance examination system.

METHOD

A. DSS Model

The characteristics of the test questions are based on a single parameter logistics model and initialization is set to 0 for all test examinees. Test examinees take the exam and get a score. Schools weigh subjects according to preference. Furthermore, processing the results using the DSS feature to produce students who deserve to be accepted by the school.

B. Diagram Use Case

The use case diagram describes the system functionality or requirements that must be fulfilled by the application. Use case diagrams state the relationship between systems and actors in the form of use case diagrams. (Rosenberg, D., et al. 2007) and (Rumbaugh, J., et al. 1991).

Diagram use case involves three actors. First, the administrator is the highest level that has full access to applications that interact with the system. His behavior is to add questions, make problems, edit questions, remove problems. Second, test examinees have limited access by requesting test questions, and looking for test scores. Third, schools look for test examinees who have high abilities according to acceptance preferences, propose test questions, and weight the ability of test examinees on each desired domain.

C. Analysis of Question Items

The Bilog-MG program is an analysis program for test questions with response patterns of 0 and 1 (Hambleton, et al. 1991; Hambleton, 1989, and Liang, et al. 2009), for example multiple choice forms. The number of test examinees' responses taken was 500 in each domain (subject) and by using a single parameter logistic model which would produce the characteristics of the problem, namely the level of difficulty of the problem. The step of the problem analysis uses the Bilog-MG program as follows. 1) Test
examinee response is copied from the question database storage file. 2) The results are pasted in notepad and then saved, for example with the name jp2mat. 3) r03mat1 answer key is made a separate file for example kp2mat name. 4) Both files are stored in the same folder, for example D:/kp2mat.txt for keys and D:/jp2mat.txt for answers. 5) Run the Bilog-MG program, select the data and examine the data, select the data file and then navigate to the jp2mat94.txt folder and specify the caselid column and response string in the column that is in notepad, then set the format to ok. Pay attention to the key column and the answer must be the same. 6) Select the data then the key item navigate to the folder in question and then type kp2mat.txt then enter, on the possible key code write 01 then select all existing keys according to the number of questions in mat1, then ok. 7) Choose the setup there are three choices, work according to the previous command, especially the general section select the PL 1 model, it means one parameter logistic model, as well as other choices done according to the previous stages. 8) Then specify the output, ie the results of the analysis that will be displayed. 9) Select RUN.

D. Procedure for Collecting Test Items on ServerCAT

The application server activity process receives requests from users to display test questions. Starting from the user in this case the examinee opens the CAT server address, the browser sends a request to the server and the server sends a response in the form of a login page to authenticate the user. If the user has the right to use the application, the server will provide a Main Page which will be displayed in the browser, the Main Page contains a list of menus.

When the user selects the Problem Bank Menu, the browser sends the request and is captured by the XMLHttpRequest object to send the request asynchronously to the server, on the server side the request will be read at once back to authenticate the user. If the process is valid, the server will read the data on the server data, the data will be converted into HTML tags and will be sent back to the client in the form of a response. The response will be captured by the XMLHttpRequest object, if it is valid then the object will update the DOM (Document Object Model) by allowing the response data from the server.

RESULT AND DISCUSSION

A. UniversiCAT application

This application requires Windows, XAMPP for MySQL web server and database server, Apache Tomcat web server for JSP / Servlet container, and Dojo as Ajax engine. Client web applications can use any operating system, provided that they are able to run web browsers such as Mozilla FireFox, Internet Explorer, Safari and others.

B. Main Page

The UniversiCAT application is a web-based application. Applications can be deployed to any server, but in this case the application path is shown in the localhost address (the application is run on the local computer).

C. Registration Process

To start using the application, a test examinee can register himself through the sign up button found on the main page. This process will save user data in the user table. To display the sign up form, is done by the code in figure 2.

```html
<button id="toolSignup" dojoType="dijit.form.Button" iconClass="mailIconOk">
  Sign Up
</button>

<script type="dojo/method" event="onClick">
  dijit.byId('dlgSignup').attr('href','modul/signup/signup.jsp');
  dijit.byId('dlgSignup').show();
</script>

Gambar 2. Menampilkan Form Sign Up
```

D. Domain Management and Test Items

The question is the basic material for testing students' abilities, where the component of the level of difficulty of the problem, is the determining factor of whether a problem is chosen as the next question, based on the students' current abilities. Figure 8 is a list of questions that can only be opened by application users with Administrator access rights.

Domain page and question bank use tree view and listview. Each node in the tree shows the domain name in the database. If the node is selected, the listview will show a list of questions on the domain. To load the question data in accordance with the domain dynamically through the following technical procedures. First, display the domain_main.jsp page that is displayed in the main section, using the JavaScript code in Figure 3.
domain : function(){
    document.getElementById('active_divUtama').value='modul/domain/domain_main.jsp';
    dojo.xhrPost(
        url: 'actions/dosavedivutama.jsp',
        form: 'mainActiveDivUtama'
    );
    dijit.byId("divUtama").attr('href','modul/domain/domain_main.jsp');
},

Figure 3. Domain_Main.jsp page

In the code, the current active module is stored, namely domain_main.jsp, by using the DoSaveDivUtama.jsp call. Once stored, then the attr() function is used, which changes the attribute of an object belonging to the dojo, in the divUtama component to load the domain_main.jsp page, which is the main page of the appearance of the domain tree and the list of data questions. Secondly, the domain data and questions displayed are derived from the ItemFileWritesore component, which is defined as in Figure 4.

Figure 4. Displaying Domain Data and Test Items

E. The Process of Working on Test Exams

The test process is carried out on the file starting with DomainDomain.jsp, whose form is shown in Figure 5. Each exam form will only be displayed within a 3 minute time limit. If the test examinee has not answered until the deadline, the system will give the problem more easily on the database based on the calculation of current student abilities. Calculation of students' abilities using the Next button, which will run the JavaScript code in Figure 6.

Figure 5. Handling Answer Questions

In the code, the dialog box will first be displayed to confirm the answer to the question in Figure 6.

Figure 6. Question Answer Confirmation

The process of calculating the ability of test examinees in certain iterations, using the DoJawabSoal servlet that implements algorithms to calculate students' abilities as in Figure 7.
protected void processRequest(HttpServletRequest request, HttpServletResponse response) 
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    int u = 0;
    double P = 0; // e^(theta - b)/1+e^(theta-b)
    double Q = 0;
    double PQ = 0;
    double SE = 0;
    double sumPQ = 0;
    double sumUminusP = 0;
    double selisihSE = 0;
    int idsoalSelanjutnya;

    // u : cek nilai jawaban soal yg benar
    String jawabanBenar = (DB.getInstance().findValue("soal", "jawaban_benar", "idsoal=" + idsoal)).toUpperCase();
    if (jawaban != null) {
        if (modul.equals("ontime")) {
            if (jawabanBenar.equals(jawaban.toUpperCase())) {
                u = 1;
            } else {
                u = 0;
            }
        } else {
            u = 0;
        }
    } else {
        u = 0;
    }

    Figure 7. CAT algorithm - Variable definition

    Calculation of the value of $u$, is done in the code in Figure 9. Previously the correct answer value was taken from the database. However, this value is only calculated by the value entered by the test examinee from the form if the examinee answers the question before the deadline expires. If true, then the value of $u$ is 1, while if it's wrong, then the value of $u$ is 0.

    Q = 1 - P;
    uMinusP = u - P;
    PQ = P * Q;
    // nilai yg di-sum : u-P dan PQ
    if (request.getSession().getAttribute("sumPQ") == null) {
        sumPQ = PQ;
    } else {
        sumPQ = PQ + Double.parseDouble(request.getSession().getAttribute("sumPQ") + "");
    }

    if (request.getSession().getAttribute("sumUminusP") == null) {
        sumUminusP = uMinusP;
    } else {
        sumUminusP = uMinusP + Double.parseDouble(request.getSession().getAttribute("sumUminusP") + "");
    }
    request.getSession().setAttribute("sumPQ", sumPQ);
    request.getSession().setAttribute("sumUminusP", sumUminusP);
    SE = 1 / (D * Math.sqrt(sumPQ));

    Figure 9. Calculation of P, Q and PQ values
Furthermore, the P value, namely the probability of students answering correctly and Q is the value of 1 - P is calculated in the code in Figure 16. The accumulation of PQ and PQ values need to be stored, so that the value is stored in session variables as shown in Figure 10. Standard Error, SE, calculated based on the accumulation of PQ values, shown in the last line of code in Figure 10.

```java
if (u == 0) {
    sidSoal = DB.getInstance().findValue("soal", "idsoal", " b <= " + (b - 0.2) + idSoalSebelumnya + " and iddomain=" + request.getSession().getAttribute("iddomain") + " order by b desc limit 1");
} else {
    sidSoal = DB.getInstance().findValue("soal", "idsoal", " b >= " + (b + 0.1) + idSoalSebelumnya + " and iddomain=" + request.getSession().getAttribute("iddomain") + " order by b asc limit 1");
}
if (!sidSoal.equals("") { 
    idsoalSelanjutnya = Integer.parseInt(sidSoal);
} else { 
    soalHabis = true;
}
System.out.println("idsoalSelanjutnya" + idsoalSelanjutnya);
if (idsoalSelanjutnya == 0) { 
    soalHabis = true;
}
if (iterasi_ke == 1 && !soalHabis) { 
    out.print("soal berikutnya;" + idsoalSelanjutnya);
} else { 
    if (!soalHabis) { 
        double SEsebelumnya = Double.parseDouble(request.getSession().getAttribute("SEsebelumnya") + ");
        selisihSE = Math.abs(SE - SEsebelumnya);
        System.out.println("selisih SE "+ selisihSE);
        if (selisihSE < 0.01) { 
            out.print("selesai ujian;");
        } else { 
            out.print("soal berikutnya;" + idsoalSelanjutnya);
        }
    }
}
```

Figure 10. Further Problem Determination

Then based on u value, that is true or not the answer, then the next question is determined, that is if the value of u = 0, or the value is wrong, then the question will be taken with the difficulty level or b, smaller than the current difficulty level of 0.2. As for the value of u = 1, or the correct value, then the problem will be taken with a level of difficulty or b greater with the current level of difficulty of 0.1. Then the test termination process is taken into account, namely by checking the current SE difference, with the previous SE. If the value of the SE value is <= 0.01, then the process is stopped by sending a message "finished the test;".

F. Domain Weighting

Domain weighting is needed for the needs of each school in determining the final score obtained by each test examinee, based on the value of each domain. This function can only be accessed by users with school access rights. The total weight of each domain is exactly equal to 100, so the final score will be determined based on the value of the examinees in each domain. The domain weighting value is stored in the School_Domain table. The following is the domain weighting code that is implemented in the Servlet DoSaveDomain Weighting in Figure 11.
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        DB.DB="rukli";
        DB.PASSWORD="adminadmin";
        String domain[][]=DB.getInstance().getDataSet("select iddomain from domain", false);
        System.out.println("domain =" + domain.length);
        String sql;
        for(int i = 0 ; i < domain.length;i++){
            int iddomain = Integer.parseInt(domain[i][0]);
            System.out.println("domain =" + iddomain);
            if(request.getParameter("txtBobot_"+iddomain)!=null){
                double bobot = Double.parseDouble(request.getParameter("txtBobot_"+iddomain)+".0");
                sql = "update Sekolah_domain set bobot=" + bobot + " where idsekolah=" + iddomain;
                System.out.println(sql);
                System.out.println("domain =" + iddomain);
                DB.getInstance().executeQuery(sql);
            }
        }
    } finally {
        out.close();
    }
}

Figure 11. Domain Weighting Storage Code

The code will save to the Sekolah_Domain table based on each iddomain that is stored in the input of the previous domain weight.

G. Test Examinee's Weighting Ability

Weighting the ability of test examinees is needed for the school in determining the priority of new student admissions. The data for each domain has been prepared. The weight value is to calculate the final score of each test examinee, so that the list of test examinees can be sorted. In this way, the school will get a complete report on the priority of acceptance of test examinees. Figure 12 shows the test examinee's report that can only be accessed by schools.

The test examinee's report is divided into two, namely the complete report of the test examinees and the priority report of acceptance of the test examinees in the school. Test examinee acceptance priority report, resulting from a complete report of the test examinees. Thus, the final value of each examinee can be determined specifically by each school by changing the domain weighting rules. Figure 12 shows three test examinees from different schools. Ahmad got a total score of 77, Budianto got a score of 83, and Sitti Hadijah got a score of 91.7 where each of the test examinees took a passing examination at a different time.

Figure 12. Test Results

H. Acceptance Report

The priority of receiving the results of the exam through CAT is then handled by the school to carry out the acceptance priority scale, which is the highest score and the minimum test score. After obtaining the final result and saved it to the sekolah_examinee table, then the system will re-read the school_peserta table, but only show test examinees who have the minimum score to be accepted as new at the school. Whereas if the new student admission quota has been met, then the process of receiving the test examinee data is stopped. The test results of Figure 18 are further processed by setting a quota of 5 students to be accepted and the lowest score of 70. Figure 13 shows the results of the reference output of a junior high school.
CONCLUSION

Test examinees take the exam adaptively where the level of difficulty of the item is adapted to current abilities. All test examinee exam results are recorded on the examinee's take-up database. The results were carried out by weighting the subjects so as to produce a temporary ranking among all test examinees. The determination of graduation is determined by the acceptance scale, namely the two quota rules received and the minimum test score.

ACKNOWLEDGEMENT

I would like to thank all those who have helped the process of conducting this research, especially the academic community of Muhammadiyah University of Makassar who have provided internal funding.

*****

REFERENCES


