International Journal of Information Technology and Knowledge Management
July-December 2011, Volume 4, No. 2, pp. 329-337

ONLINE OPERATIONALISATION OF PROCESSES IN E-MANAGEMENT SYSTEM FOR A UNIVERSITY

Ashok B. Kulkarni[1] & Valeri Pougatchev[2]

This paper describes session processes occur in the Management and Control Block (MCB) of the e-Management Control and Evaluation System (e-MCES) we have developed for the University of Technology, Jamaica (UTech). We present and discuss technical solutions, we have found and implemented within the e-MCES. Based on the MCB component of the system we discuss some essential concepts of building such systems. We are sure that ideas and approaches we have proposed and implemented, may be useful for managers, researchers, developers of educational institutions and practitioners, who are involved in processes of building an effective Educational Management Information Systems (EMIS).

Keywords: Performance Based Management, Just-in-time Compilation, e-Management Control & Evaluation Systems, Role-Based Security, .NET Technology.

1. Introduction

One of the keys to successful management of any organization, including Educational Institution is the ability to understand and apply modern management principles and techniques effectively. As high-performance organizations, Educational Institutions may be interested in having an effective, low-costly, efficient and robust multiuser (Internet or Intranet) system that is aimed at creating an environment which allows for the development of the full potential of its human resources in order to achieve its goals and objectives. That system on the one hand, provides an accurate diagnosis of the educational reality and an objective assessment of the impact of intervention policies to society. On the other hand, the information produced by e-MCES may then be used as social control tools to press those responsible for managing the educational system to produce necessary improvements. For development of educational institution's strategy and policy, timely and precise analysis of current business activity on all levels of the organization, including financial effectiveness and customer satisfaction are not possible without wide implementation computerized information systems, modern methods of communications and latest achievements in the Theory of Management. For example, under old management solutions envisaged only 5% of employees understand corporate business strategies and what is expected of them in meeting goals [1, 2]. Traditional paper-based system cannot serve educational institutions any more [3]. A new EMIS must be: (1) An Integrated System; (2) effectively able utilise the main information resources of the institution; (3) based on contemporary quantitative management approach, such as Balanced Scorecard (BSC) strategy management approach [1] and 360-degree evaluation feedback [4]. It must be sensitive to the best practices, new realities in dynamically growing educational market, be customisable, not expensive and cultivate a new management environment in the educational institutions [5, 6]. This research and development of the e-MCES has been done accordingly a Strategic Plan of the University of Technology, Jamaica. Design of the system is based on the Microsoft .NET 2.0 platform using Active Server Pages .NET (ASP.NET) and C#2005 programming language are main development tools [7, 8].

[1]Academic Affairs Division, University of Technology, Jamaica, 237 Old Hope Road, Kingston 6
[2]School of Computing & Information Technology University of Technology, Jamaica, 237 Old Hope Road, Kingston 6
E-mails: [1]akulkarni@utech.edu.jm, [2]vpougatchev@utech.edu.jm

2. General Structure of the e-MCES

One of the key ideas in our research and development of new systems for the university is that we do not reject and, if possible, we do not rebuild systems (currently in use by the University), which are serving well and don't need a general modification. Maintaining access to the information resources of these systems, and to the experience and skills of the personnel of the University who use them, is an integral part of the process of the computerization of university life [9]. Figures 1, 2 present a general structure of the e-MCES we have built in the University of Technology, Jamaica. The bottom of the Figure 1 shows, for instance, that Human Resources System (HRS) and Integrated Students Administrating System (ISAS) are the foundation and main information suppliers of the DB_1 – the data storage for the MCS block. DB_2 – is data storage for Module (or Subject)/ Instructor Evaluation System (MIES). The procedures of

interchange this information have already been described and presented in [10-12] and it is beyond the scope of this paper.

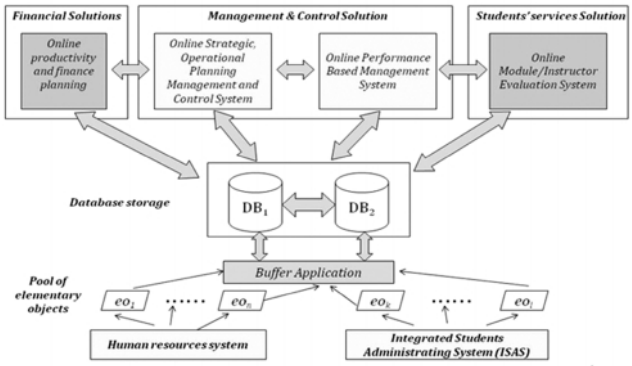


Fig. 1: General Structure of the e-Management Control & Evaluation System [13]

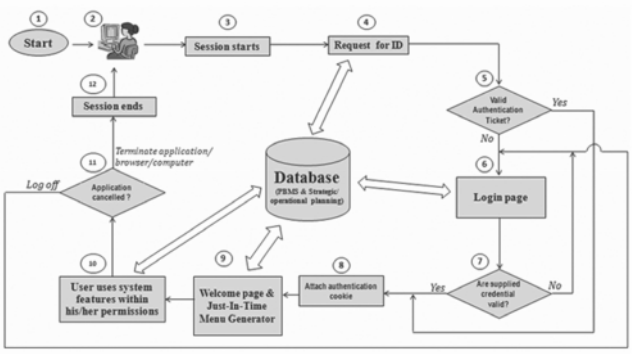


Fig. 2: Data Flow Diagram of the Management & Control Solution block

Here we discuss in more details a Management & Control Solution block, which includes two online systems: (1) Strategic, Operational Planning Management and Control; and (2) Performance Based Management. For simplicity sake we use a Data Flow Diagram (DFD), presented by Figure 2, as a guide on this path.

Actually, each user of the system has to follow following main processes: (1) - Opening a session (steps are marked by circle with numbers inside - 1, 2, 3), (2) – Log-on procedure or User Identification (steps 5, 6, 7, 8); (3) – Granting permissions and generating Just-In-Time (JIT) menu (step 9); (4) – Access to granted information resources for the user (step 10); (5) – Log-off procedure (steps 11, 12). We will examine each of these steps in details.

3. Opening a Session (Steps 1, 2, 3)

Unlike HTTP protocol, ASP.NET uses session state to keep track of each user session, which solves the problem of state management for ASP.NET application. The process begins when a user at a web browser requests a start web page of the e-MCES, typing a URL (Uniform Resource Locator) into the browser's address box. At this time ASP.NET creates an individual for this particular user a session state object (SSO) that is kept on the server whenever a user starts a new session. The SSO contains a unique session ID (SID), and this SID is sent back to the browser's cookie (by default) and forth between the server and the browser each time the user requests a page. When browser sends another request to the server, it automatically includes the cookie that contains the SID with request. Due limitation of space of the paper and because we are not writing a technical documentation, we omit the situation when browser's cookie is off; ASP.NET has other alternative techniques to bypass it.

4. Log-on Procedure (Steps 5, 6, 7, 8)

The start ASP page requests a University ID from the user. It matches it with existing IDs in the database and rejects the user if there is no ID, presented by him/her with compliment message. This is a first and primitive barrier on the way of identification of the user. Indeed, it is not great secret to know ID of somebody, but it rejects a number of persons who do not belong to the University community.

Step 5 identifies a process of Role-Based Security (RBS), which effectively supports a highest level of the user's authentication. We have selected this approach and implemented it in the e-MCES. There are three existing types of authentication in ASP.NET application: - Windows-based, Forms-based, and Passport authentication. As a matter of fact, Windows-based and Passport-based authentications are typically not valid solutions for large-scale public websites. For applications of this type, the most common authentication option is a Form-based, which we have selected in the e-MCES. Process of the user's authentication involves a several application's, network's and server's procedures. It works with a cookie on the client machine, which carries out the authentication ticket - specific and hidden information about current session. The authentication ticket is a unique ID, which is associated with current client machine and server. The Figure 3 shows

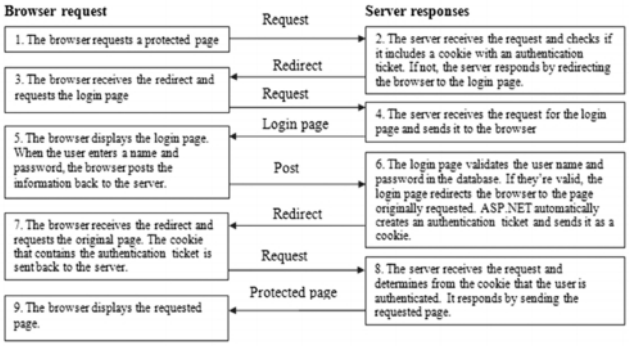


Fig. 3: HTTP Requests and Responses with Forms-based Autxhentication, According to Steps 5, 6, 7, 8 [14]

a series of exchanges that occur between a web browser and a server when user attempts to access a page that's protected by forms-based authentication.

In e-MCES we use several standard server controls, which are offered by ASP.NET and make programming security-related aspects of web applications easier than ever before: Login, PasswordRecovery, ChangePassword, etc. These controls rely entirely on the membership API (Application Programming Interface) and selected provider to execute standard operations such as validating credentials, displaying error messages, and redirecting to the originally requested page (in e-MCES it is a Welcome.aspx page) in case of successful login. The membership API provides a set of classes with wide range of methods to let developer manage users and roles:-adding a new user and editing any associated user information such as – e-mail and password, creating and managing association between users and roles. The Membership class defaults to a provider that stores user confidential access information to a SQL Express database in predefined format, through its property Provider. This property returns a reference to the membership provider currently in use and selected from the configuration files web.config (is located in root directory of the application) and machine.config (located in C:\WINDOWS \Microsoft. NET\ Framework\vvv\CONFIG on the server, where vvv is a version of ASP.NET). An example of a child <provider> element under which provider configures is as follows:

```
<membership>
    <providers>
        <add name="AspNetSqlMembershipProvider"
          type="System.Web.Security.Sql
          MembershipProvider, System.Web,
          Version=2.0.0.0,
          Culture=neutral,PublicKeyToken= b03f
          5f7f11d50a3a"
          connectionStringName="LocalSqlServer"
          enablePasswordRetrieval="false"
          enablePasswordReset="true" requires
          Question AndAnswer="true"                (4.1)
          applicationName="/" requires Unique
          Emai                    ="false"
          passwordFormat="Hashed"
          maxInvalidPasswordAttempts="5"
          minRequiredPasswordLength="7"
          minRequiredNonalphanumericCharacters="1"
          passwordAttemptWindow="10"
          passwordStrengthRegularExpression=""/
          >
    </providers>
</membership>
```

The content of the configuration file (4.1) describes a security policy of the application by values of its attributes. Some of them:

- connectionStringName="LocalSqlServer" - describes that application uses a standard .mdf security database, predifined by ASP.NET, which originally is located in App_Data folder. We have improved this solution, removed this database to the SQL Server from this application directory. It gave us a number of benefits, including its better maintenance.
- passwordFormat="Hashed"– declares that confidential information about users access (user name, password, etc.) will be encrypted. Usually ASP.NET provides several hashing algorithms with 128-bits strength, for instance –Massage Digest method (MD5) and Secure Hash Algorithm (SHA) with different modifications [15, 16]. The e-MCES can modify security algorithm using Hash Algorithm object. A portion of C# code is as follows:

```
static void Main(string[] args)
{
    // Open a local configuration file on the
    C drive
    FileStream fs = new FileStream (@"C:\
    eMCES_Config_File.txt", FileMode.Open);
    // now generate a hash code for this file
    using MD5 hashing algorithm
    HashAlgorithm alg=Hash Algorithm.
    Create("MD5");
    byte[] fileHashValue = alg. Compute
    Hash (fs);
    ..................................
}
```

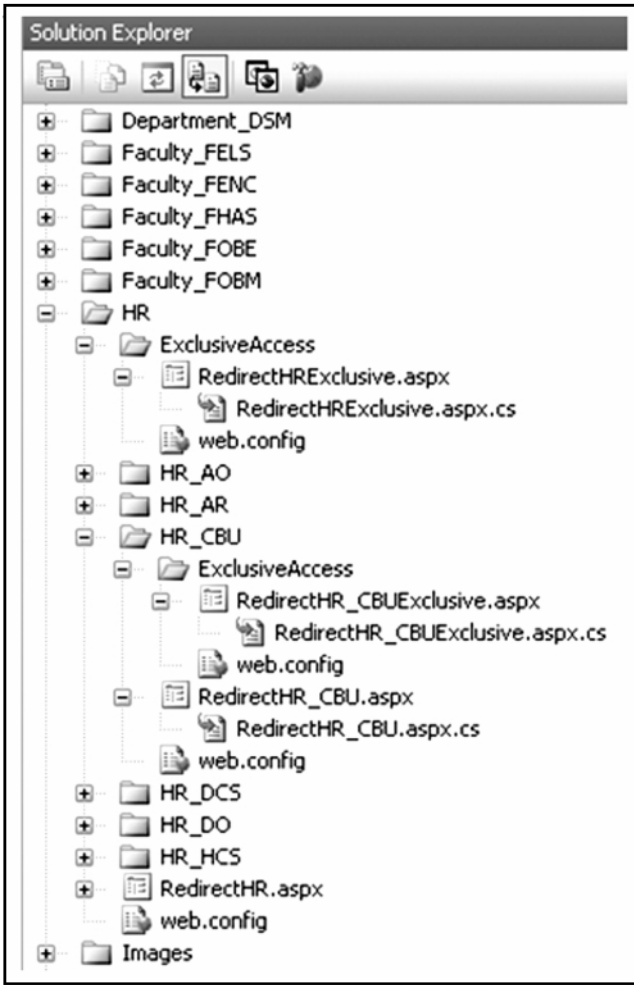
- maxInvalidPasswordAttempts="5" – declares a number of unseccessful attempts are allowed
- minRequiredPasswordLength="7" – declares a minimum length of the password
- minRequiredNonalphanumericCharacters= "1"–describes that at least one nonalphabe-tivcal charcted it must be presented in password

The configuration information from web.config file overrides the same information in machine.config file.

The structure of the Application project maps a general structure of the educational institution. The reason for that is to guarantee a high level of security in process of granting

access to different units of the institution. The structure of the application includes different types of programming modules. There are – executive classes, which provide some functions requested by user, redirected classes, which redirect executive procedure to the executive classes, etc.

Figure 4 shows a portion of Human Resources Department (HRD) hierarchy. The numbers of web.config files on different levels identify permissions to get access to some specific unit. For example:



- web.config file in EsxclusiveAccess node of the HR (Head root for HRD):

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <system.web>
        <authorization>
            <allow roles="HR_Senior_Director" />
            <deny users="*" />
        </authorization>
    </system.web>
</configuration>
```

- web.config file in EsxclusiveAccess node of the Compensation & Benefits Unit (HR_CBU) of HR:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <system.web>
        <authorization>
            <allow
            roles="HR_CBU_Compensation_
                Benefits_Mgr" />
            <deny users="*" />
            </authorization>
    </system.web>
</configuration>
```

Fig. 4: A Fragment the HRD Hierarchy in e-MCES Project and Content of Web. Config Files

Here the "HR_Senior_Director" and the"HR_ CBU_ Compensation_Benefits_Mgr" are roles associated with users Senior Director of Human Resources Department and Manager of the Compensation & Benefits Unit respectivelly. An <autherization> element grants an access to the node (which includes a .aspx classes Redirect HRExclusive. aspx or RedirectHR _ CBUExclusive. aspx - see Figure 4) of the structure, who is in role which is described by this element and denies access for others. Class with prefix Redirect in the name of the class declares, that this class works with menu for the current user and actually plays a distributor role for the executive procedure. In another words, it redirects the execution procedure of the application to the classes, which are associated with menu items. We use the term execution procedure in a broad sense – we call a program representation executable if it can be executed directly as in machine code, or inderectly using an interpreter. One of the main benefits of the ASP.NET in comparison with other web technology, like PHP, ASP, Perl, etc., is that it finally produces a compiled (not interpreted) code for execution. Here is a C# fragment of the code behind of the RedirectHRExclusive.aspx class:

```
redirect = Session["PageProcessor"].ToString();
d = Page.Request.QueryString["D"].ToString();
E = Page.Request.QueryString["E"].ToString();
. . . . . .
Session["AppraiserRole"] = HR_Senior_Director;
Session["CanUpdate?"] = "Yes";
. . . . . .
Session["PeerID"] = PublicClass. Peer. Supervisor.
GetHashCode();
```

```
Session["UnitID"] = HR;  Session["UnitName"] = HR_Title;
switch (d)
{
case "eval":
   Session["MainCategory"]= PublicClass. Main Categories. Administrative.GetHashCode();
   Session["IsSupervisor"]="Yes"; Session ["IsHead"] = "Yes";
   Session["enableToSeeSupervisorComments"] = "No";
    if (E == "Mng")
   {
   Session["SupUnit"] = ""; Session ["Staff Category "] = PublicClass. Categories. Unit-Managers.GetHashCode();
   }
   else
   {
   Session["SupUnit"]= HR_AO;Session["Staff Category"] = "90";
   }
   redirect + = "CompetencyAdminStaff.aspx";
   Page. Response.Redirect(redirect); break;
   ..........................................
```

In this example a variable redirect holds a name "CompetencyAdminStaff.aspx" of the executive ASP page with requested functions. The security warranty is that the system has no another way to get any executive ASP page but only through this procedure.

5. WELCOME PAGE AND JUST-IN-TIME MENU GENERATOR (STEP 9)

A menu grants to a user his/her permissions. Here we have encountered a problem, that for large organization with many different positions of staff it is very difficult to maintain the permission policy for each member of staff. Moreover, organizations are very dynamic – new positions are created, others are closing, and staff migration between these positions is usual process at high frequency. Traditional maintenance of these processes updating an application (redesign and recompiling) is waste of time and forces. To create one very big and universal menu predicting new positions is not possible easily. We have found an original solution for solving this problem offering a Just-In-Time Compilation (JITC) approach for process of the building menu [17]. Accordingly [18] Just-In-Time compilation process includes any translation performed dynamically, after a program has started execution and is used to gain the benefits of static compilation and interpretation. ASP.NET is compiled, not interpreted web technology [19]. It generates HTML document, to be interpreted by the browser of client machine. These two factors stimulated our research to create JITC.

In e-MCES, we consider menu of two levels – parent and its subordinate - child. We have created a JIT Menu Generator (JITMG) – a C# procedure, which uses tokens of menu as input information. Under tokens we consider all necessary elements of future menu – texts of items, which will be seen by user and touchable (in terms of ability to be hyperlinks) by user, destination addresses (URLs) associated with these items, and link to the user's role. Figure 5 shows a fragment of database, which holds tokens for JITMG. Table Roles receives a text value with User's role. Then two SQL query retrieves set of tokens associated with this particular role from five linked tables:

- SELECT L1.[ID], L1.[Text] AS Text1 FROM mnuLevel1 AS L1 WHERE RoleName = '" + uRole + "'"
- SELECT L2.[Location], L2.mnuLevel1ID, L2.[Text] AS Text2 FROM mnuLevel2 AS L2 INNER JOIN mnuLevel1 AS L1 ON L2.mnuLevel1ID = L1.[ID] "WHERE L1.RoleName = '" + uRole + "'"

Here, uRole is a text variable, which holds a name of user's role.

After that, a C# programming procedure of the Welcome.aspx page code-behind class uses the set of tokens, which were retrieved by SQL queries mentioned above, Navigation object from .NET Framework and generates a JavaScript program, which resides into a HTML response as its integral part. We consider this solution including C# and ASP.NET source code is an intellectual property of the University of Technology, Jamaica hence it is not presented in more details. Appendix A shows a snapshot of the browser's screen with menu, generated for user Mrs. Jennifer Williams, who is in role of a Senior Director of Human Resources Department.

A7ppendix B shows fragments of JavaScript menu, generated automatically by JITMG. For the first session of the user System Administrator provides a temporary password, which is combination of two strings of user's University's ID, separated by dot symbol. This temporary password must be updated by the user as soon as possible due the security reason. The web.config file of the application (is located in root directory of the application) holds a number of these attempts and can be modified by the System Administrator without recompiling the application. Default value of this number is 5. As mentioned above, the confidential information (User ID, password, etc.) kept in the database in encrypted value and cannot be restored even by System Administrator. In this case, the user must be registered again with the system. This approach ensures a high level of security.

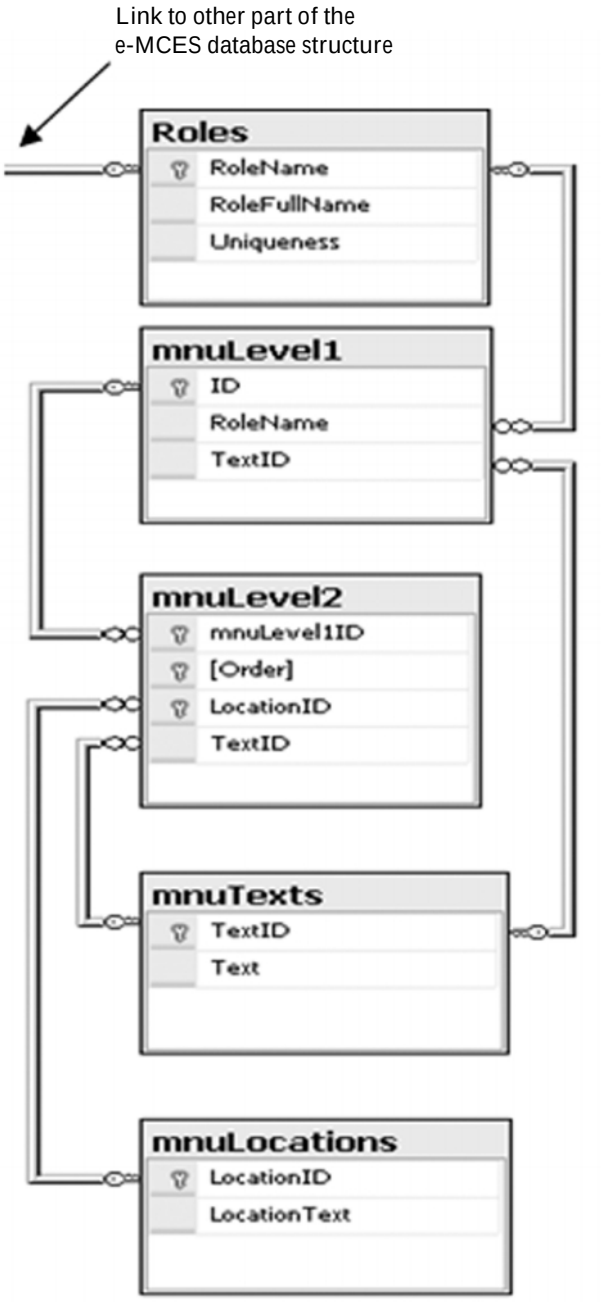


Fig. 5: Fragment of database, which holds tokens for JIT

The JIT Menu Generator solution has many benefits. Two of them are:

- Using this technology, it does not matter how big organisation is and how many different positions are there,
- To add a new position, new user, assign a new user to some position, or change position for some particular user is the responsibility of the person from HRD, who is assigned for these functions – he/she can do it through e-MCES interface which provides these functions, without recompiling an application – it is a usual routine for administrator of the system.

6. MENU FOR SENIOR DIRECTOR OF HUMAN RESOURCES DEPARTMENT (CASE STUDY)

Table 1 shows a full content of menu, generated by JITMG for Senior Director of HRD (Appendix A shows that some choice has been made by the user):

Table 1
A Content of Full Menu for Senior Director of HRD

##	Content of menu Level 1 ("parent")	Content of menu Level 2 ("child")
1	University's	the University corporate score
2	HR evaluation	total score
3	Detail view of	the Human Resources score evaluation
4	Self	the Human Resources Senior Director's competency
5		evaluation
6	Evaluation of	managers of the Human Resources department
7		individual manager's achievements
8		admin. staff of the Human resources Department (Senior Director's office)
9		individual staff achievements (admin.)
10	Planning	maintenance of HR operational plan (last year)
11		maintenance of HR operational plan (next year)
12		maintenance of Admin. office operational plan (last year)
13		maintenance of Admin. office operational plan (next year)
14	Targets	assigning objectives/targets to the Units (last year)
15		assigning objectives/targets to the Units (next year)
16		selection of the Admin office objectives/targets (last year)
17		selection of the Admin office objectives/targets (next year)
18	Reports	view of Scores list of Managers and Supervisors
19		view of Scores list of HR office Administrator staff (Senior Director's office)
20		view of Scores list of the Development unit's staff
21		view of Scores list of the Compensation & Benefits unit's staff
22		view of Scores list of the Health Care Center staff
23		view of Scores list of the health Care Center Medical Doctors staff
24		view of Scores list of the Day Care Center staff

According Table 1 we can see that e-MCES offers 24 functions for Senior Director of the HRD, including planning options for HRD's units and their managers/ supervisors (not for employees of these units, because it is responsibility of their managers/supervisors), viewing a scores of evaluation of all HRD staff, providing a vertical alignment for the items of plans are located on different levels (details of this process is are not given in this paper but can be seen in [20]), viewing a total score of HR Units, HRD itself, and total corporate score of the University. Here we can see, that automatically built menu reflects a hierarchical structure of the University with respective functions and responsibilities. e-MCES generates a programming classes (ASP.NET and other supported modules), which are handling processes advertized in menu. Only authorized person can obtain functions provided by automatic menu only for the session time. After terminating session it is discarded. It exists just in time of user's session.

User can terminate session by Log-out process, when he/she is pressing a Logout button, or closing a browser, or terminating a computer (steps 11, 12).

7. CONCLUSION

In this paper we have described the processes occurring during user's session in e-MCES. We have selected here role-based security approach in e-MCES in conjunction with Form-based authentication procedures, which are based on highest level of encryption technique (hashing procedures used a MD5 & SHA mathematical hashing algorithms). This is currently up-to-date technology. We consider that the security and navigations processes are different sides of the same "coin" – Granting Permissions for the User. Solutions we have found during our research and development can be implemented in any Information Management Systems in general and in Educational institutions in particular.

APPENDIX A

Welcome page with automatically generated menu

APPENDIX B

Fragments of JavaScript menu, generated automatically by JITMG

```
// java script program handling
<script type="text/javascript">
//<![CDATA[
var theForm = document.forms['fPBMS'];
if (!theForm)
  {
theForm = document.fPBMS;
}
function__doPostBack(eventTarget, eventArgument)
  {
   if (!theForm.onsubmit || (theForm.onsubmit() != false))
  {
    theForm.__EVENTTARGET.value = eventTarget;
    theForm.__EVENTARGUMENT.value= event Argum-
    ent;
    theForm.submit();
  }
}
  //]]>
  /script>
    ......
  <scriptsrc="/EMCES2010 Web Resource.axd?
    d=GJ56476UwNAULC srfu P4Q2 &amp; t=63 4 2 0
    3587922187500" type="text/javascript"></script>
  <scriptsrc="/EMCES2010 WebResource. axd?d = 5v
    uYZmLBPR4bkOEVL2FIA2 &amp;t= 63420358
    7922187500" type="text/javascript"></script>
      ......
  //Items of the parent menu
      <td style="white-space:nowrap;width:100%;"><a
      class="NavigationMenu_1NavigationMenu_3
      NavigationMenu_8"href="javascript:__doPostBack
      ('Navigation Menu','~*|*Choice Error.aspx? P=HR1
      ')">University's</a></td>
      ......
      <tdstyle="white-space:nowrap;width:100%;"><a
      class="NavigationMenu_1 NavigationMenu_3
      NavigationMenu_8" href= "javascript :__
      doPostBack('NavigationMenu',' ~*|*ChoiceError.
      aspx? P=HR2')"> HR evaluation</a></td>
      ......
//Items of child menus
<td style="white-space:nowrap;width:100%;"><a
   class="NavigationMenu_1 NavigationMenu_3
   NavigationMenu_8" href=" javascript:__doPostBack
   ('NavigationMenu','~*|* ChoiceError.aspx? P=HR4
   _31')"> Reports:</a></td>
    ........
<td style="white-space:nowrap;width:100%;"><a
   class="NavigationMenu_1 NavigationMenu_5"
   href="javascript:__doPostBack('NavigationMenu','
   ~*|*ChoiceError.aspx?P=HR2
   President*| *Redirect President. aspx? D=DeptEval
   &amp; F=HR&amp;G=T')"> total score</a></td>
  .........
<td style="white-space:nowrap;width:100%;"><a
   class="NavigationMenu_1 NavigationMenu_5"
   href =" javascript:__ doPostBack (' NavigationMenu
   ','~*|* ChoiceError . aspx ? P = HR3 \\~*|*President
   *|*RedirectPresident.aspx? D=DeptEval &amp;
   F=HR&amp;G=D')">the Human Resources score
   evaluation</a></td>
    ........
<td style="white-space:nowrap;width:100%;"><a
   class="NavigationMenu_1 NavigationMenu_5"
   href =" javascript:__doPostBack (' NavigationMenu
   ','~*|* ChoiceError. aspx? P=HR4 _31 \\~*|* HR *|*
   ExclusiveAccess*|*RedirectHRExclusive.aspx ?
   D=reports&amp;F=totalScoreAdmStaffDCS
   &amp;E=0')">View a Scores list of the Day Care
   Centre staff</a></td>
    ........
// java script program handling
<script type="text/javascript">
//<![CDATA[
   var NavigationMenu_Data = new Object();
   NavigationMenu_Data.disappearAfter = 500;
   NavigationMenu_Data.horizontalOffset = 2;
   NavigationMenu_Data.verticalOffset = 0;
   NavigationMenu_Data.hoverClass = 'Navigation
   Menu_17';
   NavigationMenu_Data.hoverHyperLinkClass=
   'NavigationMenu_16';
   NavigationMenu_Data.staticHoverClass=
   'NavigationMenu_15';
   NavigationMenu_Data.staticHoverHyperLinkClass
   ='NavigationMenu_14';
   //]]>
</script>
```

REFERENCES

[1] Kaplan R., Norton D., (1996), "The Balanced Scorecard: Translating Strategy into Action, A Landmark Achievements", Harvard Business School Press, Boston, MA, U.S.

[2] "The Secret to Effective, No-Hassle Performance Reviews", HR Insiders Series, For Small & Midsized Businesses, (2007), SuccessFactors, San Mateo, ttp://www.successfactors.com

[3] Brown, J.S., Duguid, P.D. (2000), "Social Life of Information", Harvard Business School Press, London.

[4] Zenger H. John, Folkman Joseph, (2002), "The Extraordinary Leader, Turning Good Managers into Great Leaders", McGraw-Hill, ISBN 0-07-138747.

[5] Hua H., Herstein, J., (March 2003), "Education Management Information System (EMIS): Integrated Data and Information Systems and Their Implications in Educational Management", Annual Conference of Comparative and International Education Society New Orleans, LA, USA

[6] Ellison, R. (2004), "A Practical Guide to Working with Education Management Information Systems". London: DfID. Retrieved from http://www.paris21.org/documents/2402.pdf

[7] Esposito Dino (2006), "Programming Microsoft ASP.NET 2.0. Core Reference", Microsoft, Microsoft Press, Redmond, Washington, USA

[8] Nagel Christian, Evjen Bill, Glynn Jay, Skinner Morgan, Watson Karli, Jones Allen, (2006), "Professional C# 2005. Programmer to Programmer", WROX, Wiley Publishing, Inc. Indianapolis, IN, USA

[9] Pougatchev V. (2008), "Online Performance Based Management and Evaluation System at the University of Technology, Jamaica: Information Resources and Security Solutions", The 11th IASTED International Conference on Computers and Advanced Technology for Education (CATE-2008), Crete, Greece, (September 29 – October 1, 2008)

[10] Pougatchev V., (October, 2006), "An Object-Oriented Approach in Development of On-Line Evaluation Systems for Educational Institution", The 9th IASTED International conference on Computers and Advanced Technology for Education CATE-2006, Peru, Lima, October 4-6, 2006, http://www.actapress.com/Abstract.aspx?paperId=28579

[11] Pougatchev V., (October, 2007), "Development Evaluation Systems for Educational Institutions using Reusable Information Objects", The 10th IASTED International conference on Computers and Advanced Technology for Education (CATE-2007), Beijing, China, (October 8-10, 2007), http://portal.acm.org/citation.cfm?id=1650171

[12] Pougatchev V., Ellis J., Glasgow S., Johnson N., "Online Performance Based Management and Evaluation System as an Instrument to Manage the Quality of Institutional Performance at the University of Technology, Jamaica", Journal of Research in Innovative Teaching, National University, San Diego, California, USA, Vol-2, Issue 1, March 2009, http://www.nu.edu/assets/resources/pageResources/7638_JournalofResearch09.pdf

[13] Kulkarni A., Pougatchev V., "Macroscopic View on the Structure of the e-Management Control and Evaluation System for the University of Technology, Jamaica", International Journal of Information Technology & Knowledge Management, ISSN 0973-4414, Vol-IV, Issue-I, December, 2010

[14] Murach Joel, Boehm Anne, (2006), "Murach's ASP.NET 2.0 web programming with C# 2005", Mike Murach & Associates, Inc. ISBN: 1-890774-31-6

[15] National Institute of Standards and Technology. FIPS 180: Secure Hash Standard, May 1993., Available from: http://csrc.nist.gov.

[16] FIPS 180-1. Secure hash standard, NIST, US Department of Commerce, Washington D.C., Springer-Verlag, 1996.

[17] Pougatchev V., (2008), "Online Performance Based Management and Evaluation System at the University of Technology, Jamaica: Information Resources and Security Solutions", The 11th IASTED International conference on Computers and Advanced Technology for Education (CATE-2008), Crete, Greece, (September 29 – October 1, 2008)

[18] Aycock John, (2003), "A Brief History of Just-In-Time", ACM Computing Surveys, Vol. 35, No. 2, June 2003, pp. 97–113.

[19] MacDonald M., Szpuszta M, (2006), "Pro ASP.NET 2.0 in C#2005", Special Edition, APRESS®, ISBN: 1-59059-768-0

[20] Kulkarni A., Pougatchev V., "V-index as a Measure of the Targets/Objectives Accomplishment of the Strategic and Operation Plans for an Educational Institution Unit (Case Study)", Journal of Research in Innovative Teaching, National University, San Diego, California, USA Volume 4, Issue 1, March 2011 (sent to publisher)