Technical Aspects of the Integrated Management Information System for Educational Institutions

Valeri Pougatchev[+], Ashok B. Kulkarni[++]

***Abstract* —** This paper briefly describes several technical aspects and general structure of the Integrated Management Information System (IMIS) for Educational Institutions, proposed by authors.

Authors consider the IMIS as a full management system for educational institutions that includes strategic, academic, and financial planning and management components. This approach will allow the institution to respond promptly to real-world challenges and opportunities that might affect its short- and long-term strategies.

Some of authors' original technical and security solutions of the IMIS have been practically implemented in the e-Management Control and Evaluation System (e-MCES) in the University of Technology, Jamaica (UTech).

Solutions, which have been implemented in the e-MCES do not depend on size of the institution and on diversity of Academic and non-Academic roles and can be implemented to any educational institution with different structure and educational policy or to some part of it without recompiling it or with little adjustments. This allows step-by-step deployment of the system without affecting currently working mechanism of management.

***Keywords* —** Integrated Management Information System, e-MCES, .NET technology, Multitier Architecture, Object-Oriented Approach, Role-based security, Navigations, Connection Pooling, Just-in-Time Menu Generator.

I. 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-cost, efficient and robust multiuser (Internet/Intranet) Integrated Management Information 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. This 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 IMIS 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 the latest achievements in the Theory of Management. Traditional paper-based system cannot serve educational institutions any more. A new IMIS must be:

1. An Integrated System;
2. Effectively able to utilise the main information resources of the institution;
3. Based on contemporary quantitative management approach, such as Balanced Scorecard (BSC) strategy management approach and 360-degree evaluation feedback.

This research and development of the e-MCES has been done according to the Strategic Plan of the University of Technology, Jamaica. Playing a role for the top management system in the educational institution, e-MCES does not discard existing software used extensively by institution for a long time in their computerized systems. It establishes new relations between data and information resources of the institution, working at higher level of communication.

The structure of e-MCES database is adaptable to new criteria of educational institutions performance evaluation, because it has already accumulated almost all necessary information about business activity of the institution.

II. Structure of the e-Management Control and Evaluation System

From our point of view, the e-Management Control and Evaluation System (e-MCES) internally should consist of the following components [1, 2]:

- Management & Control Solution
  a. Strategic, Operational Planning Management and Control System
  b. Performance Based Management System
- Students' services Solution - Online Module/Instructor Evaluation System

Manuscript received April 13, 2011. This work has been done accordingly a Strategic Plan of the University of Technology, Jamaica.

[+]Mr. Valeri Pougatchev is a Senior Lecturer of the School of Computing and Information Technology of the Faculty of Engineering and Computing at the University of Technology, Jamaica. He is a PhD candidate in Computer Science area. He can be reached by: e-mail – *vpougatchev@utech.edu.jm*, or phone – (876) 423-0315.

[++]Prof. Ashok Kulkarni is a Deputy President of the University of Technology, Jamaica and Head of Academic Division. He can be reached by: e-mail – *akulkarni@utech.edu.jm*, or phone – (876) 927-1611

- Financial Solutions - Online productivity and finance planning

The general structure of the e-MCES is presented in Fig. 1.

Main components of the e-MCES are as follows:

- *Information resources*. Information resources of the current version of the system are based on *Human Resources* database (*HRDB*) and University portal – *Integrated Students Administrating System* (*ISAS).*
- *Online Module/Instructor Evaluation System (OLMIES).* This system provides evaluations of modules delivered by departments/schools and evaluation of teachers by students.
- *Online Strategic, Operational Planning Management and Control.* There are Performance of Academic and non-Academic staff appraisal system and the Strategic, Operational planning system.
- *Online productivity and finance planning.* The educational institution productivity measurement system which provides aggregate objective metrics on productivity of faculty, departments and other institution's units vis-a-vis financial inputs and budget.

III. Development Platform

The e-MCES is a Web-based ASP.NET application. At the time of development, three main technologies and platforms were available to develop Web applications: Active Server Pages (ASP), Java Server Pages (JSP), and the open source Web platform commonly referred to as LAMP (Linux plus Apache plus MySQL plus either Perl, Python, or PHP as the programming language). Although each has language-specific and architecture-specific features, all these Web development platforms are designed to create interactive pages as part of a Web-based application [2]. To some extent, all enable the developers to separate programming logic from the page layout through the use of components that the page itself is responsible to call and render. Besides this common ultimate goal, significant differences exist among those platforms, most of which relate to the programming model and languages they promote and support. For example, JSP exploits the Java framework of classes and, with JavaBeans, provides an effective extensibility model for reusing components [3]. In addition, JSP supports tag customization and lets developers associate code with a custom tag definition. Finally, because it's a key element of the Java 2 Enterprise Edition (J2EE) platform, JSP relies on the Java language, as opposed to the scripting languages used by both ASP and LAMP platforms [4]. What really differentiates ASP.NET from the plethora of other Web development technologies is, the abstract programming model it propounds the Web Forms model. This development platform is designed to create interactive pages as part of a Web-based application [2].

ASP.NET like other Web development environments works on top of the HTTP protocol and takes advantages of HTTP commands and policies to set up two-way, browser-to-server communication and cooperation.

In addition, the whole ASP.NET platform comes as a native part of the Microsoft .NET Framework. ASP.NET applications compile pieces of code, are made of reusable and extensible components, can be authored with first-class language C#, which we have extensively used in our process of development, and can access the entire hierarchy of classes in the .NET Framework.

There are seven important features about ASP.NET [5]:

- ASP.NET is a part of the .NET Framework;
- ASP.NET is compiled, not interpreted;
- ASP.NET is a multilanguage;
- ASP.NET is hosted by the Common Language Runtime;
- ASP.NET is object-oriented;
- ASP.NET is multiservice and multibrowser;
- ASP.NET is easy to deploy and configure.

In view of the above, for writing code-behind modules, classes, and business objects in *e-MCES,* we have selected the C# programming language [3, 5, 6].

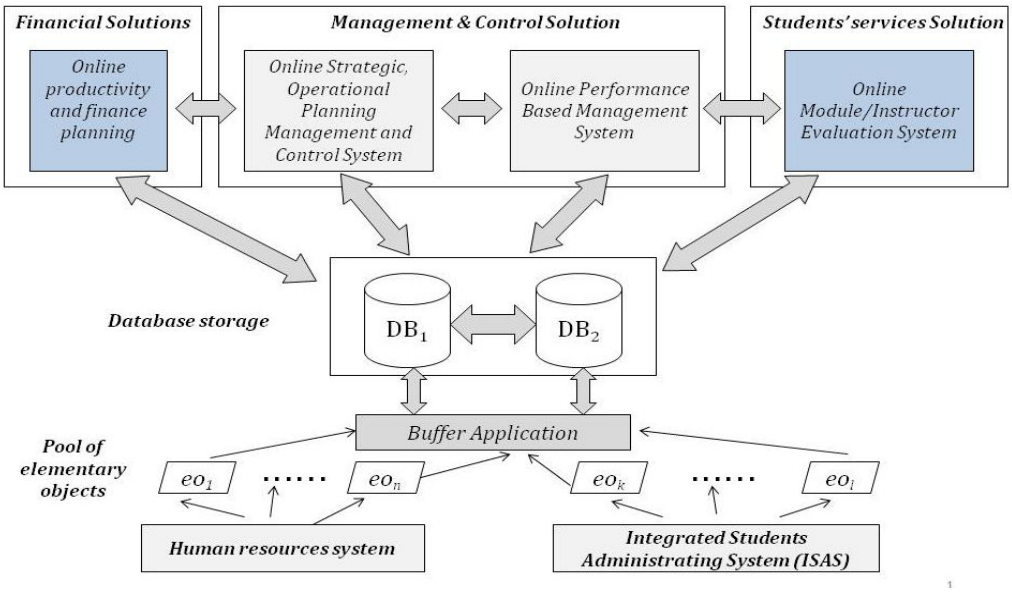


Fig. 1 General structure of the e-Management Control & Evaluation System [1, 2]

IV. Multitier Architecture

Web based e-MCES is built, using an up-to-date *"Multi-tier Architecture"* (sometimes referred to as *"n-tier architecture*", where *n > 2*, or a *"multilayer architecture",* or a *"distributed application"*), where processing is distributed between the client and the server, and business logic is captured in the middle tier [6, 7]. The n-tier architecture isolates each major piece of functionality, so that the presentation (user interface) is independent of the processing business rules and business logic, which in turn are separate from the data. This model requires much more analysis and design up front, but greatly reduces maintenance costs and increases functional flexibility in the long run. A distributed application is only one approach that spreads its execution and serves hundreds or thousands of clients simultaneously, running in different environments, operating systems, platforms and providing the highest level of performance, scalability and security [6].

Unlike client/server design, a multitier application has middle tier – *Business* (or *Logic*) layer. Client/server architecture has several disadvantages in comparison with Multitier Architecture:

1. It cannot accommodate easy client interaction because each client runs a separate instance of the application.
2. The business rules (internal and confidential data handling procedures of the organization) of the University must be presented on the client side instances of applications. It will not protect the system from the hacker's attacks threats.
3. Maintenance of all instances of the application on thousands of client machines will be major problem for system administration.
4. The distributed application needs to serve not dozens but hundreds and thousands of simultaneous users. In this case, server-side database quickly becomes a bottleneck, which has no easy solution.

The tiers in the Multitier Architecture model are shown in Fig. 2. Here:

- The *presentation tier* consists of the ASP.NET pages that manage the appearance of the application. This layer can include bound controls and objects that bind the data controls to the data.
- The *business tier* contains the data access classes that manage the data access for the application. This layer can also contain business objects that represent business rules as well as the calculations.
- The *data tier* consists of the database that contains the data for the application. It includes SQL statements that do the database access and are saved in stored procedures within the database.

For internet based application client side *("Application front end")* can have any web-browser: Internet Explorer, Mozilla Firefox, Opera, Netscape Navigator, Safari, etc., and can work on different platforms. Microsoft's .NET Framework uses an *Extensible Markup Language* (XML) – a platform independent language, which gives ASP.NET applications a rich set of features for using and manipulating XML data.

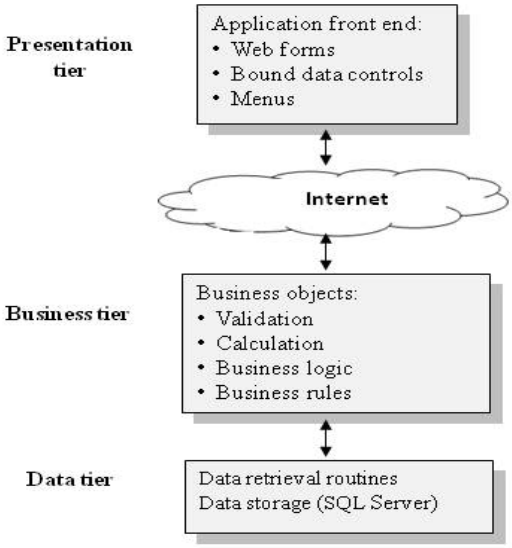


Fig. 2 Multi-tier architecture of web application (number of tiers is greater than 2)

ASP.NET uses a multilayer architecture that revolves around a few key concepts of Access Data Object (ADO.NET), such as *Connection, Command, DataReader, DataAdapter* and *DataSet* objects [6]. ADO.NET is designed to combine the best features of its predecessors while adding features most frequently requested by developers – greater XML support, easy disconnected data access, more control over updates, and greater update flexibility. ADO.NET is designed to help developers build efficient multi-tier database applications across intranets and the Internet.

V. Performance, Scalability and Connection Pooling

The e-MCES is a distributed application. One common mistake of software developers is to look at distributed components as full partners in object-oriented design. Unfortunately, distributed programming, such as an ASP.NET programming, is a compromise between networking technology and object-oriented practice [8]. For example, when we are speaking about *Performance* and *Scalability* of the web-application, we have to realize, that these issues are not the same and must be investigated at the system design level. The *Performance* is a measure of the application's speed. The *Scalability* indicates how this speed varies as the client load increases. We can see the difference between these issues for different web solutions, which use a session state objects. In some web applications,

using state objects usually performs faster for small number of clients. However, as the number of simultaneous clients increases, they reach a bottleneck and perform sluggishly. Fig. 3 depicts this relationship.

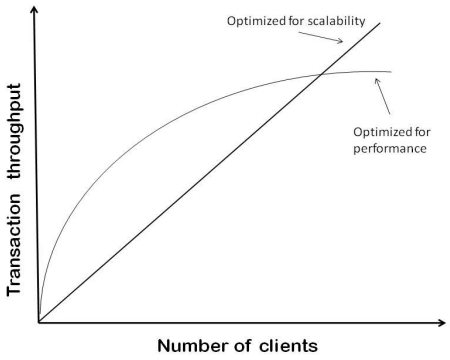


Fig. 3 Performance vs. scalability [4]

One of best example of the distinction between performance for small number of users and scalability to large numbers of clients occurs with database connection pooling. Let us consider two configurations.

First one – *"Configuration A"* (Fig. 4) for a small size of educational institution's staff. Another one – *"Configuration B"* (Fig. 5) for a large institution (UTech with number of staff greater than 1,500 employees is an example of large organization).

In Configuration A, all clients have local copies of the database connection service class (object). Therefore, they can use it locally (and speedily) in process.

Connection cannot be pooled between clients, because the connection object is always instantiated in the process of the client. A fragment of ASP.NET page code, which provides this instantiation, is as follows:

```
<asp: SqlDataSource ID="EmployeesList" Runat-"server"
    SelectCommand="SELECT FirstName, LastName, Title
    FROM Employees" ConnectionString="Data
    Source=127.0.0.1; Integrated Security=SSPI; Initial
    Catalog=EMCES2010">
</asp: SqlDataSource>
```

Here we can see that *SqlDataSource* object creates an instance of connection object, which is associated with application that is called by user and serves it individually.

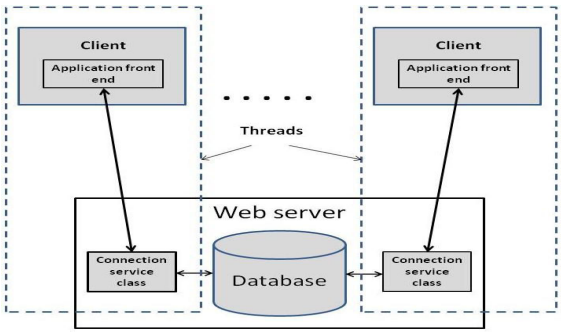


Fig. 4 Configuration A: best performance for small educational institution

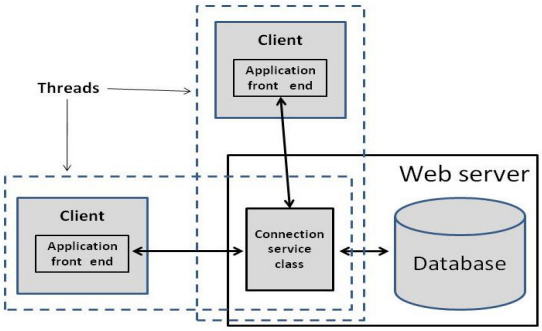


Fig. 5 Configuration B: best scalability for big educational institution

In Configuration B (Fig. 5), the connection service class lives on the database server and communicates with out-of-process clients through ADO.NET access classes. This intrinsically adds overhead because all calls must travel over process boundaries and the network. It allows connections to be pooled among all clients. In e-MCES, we use this technique. A fragment of code from some code-behind class of the e-MCES is given below. In this *C#* example, three new *SqlConnection* objects to e-MCES2010 (2010 year) and to e-MCES2011 (2011 year) databases are created, but only two connection pools are required to manage them. Note that the first and second connection strings differ by the value assigned for *Initial Catalog:*

```
using (SqlConnection connection = new
        SqlConnection("Integrated Security=SSPI;
        Initial Catalog= eMCES2010 ")) {
    connection.Open();      // Pool A is created.
  }
using (SqlConnection connection = new
          SqlConnection("Integrated Security=SSPI;
          Initial Catalog= eMCES2011")) {
    // Pool B is created because the connection
    // strings differ.
    connection.Open();
  }
using (SqlConnection connection = new
        SqlConnection("Integrated Security=SSPI;
        Initial Catalog= eMCES2010 ")) {
     // The connection string matches pool A.
     connection.Open();

}
```

Naturally the multithreading operations add overheads and reduce speed for large institutions.

VI. Security Solutions

The middle (*Business*) tier in multi-tier architecture is frequently the heart of a distributed application. All the information moves through this layer: data on its way from the database to the user interface and user information posted from a form goes back to the database. This is where

the *brains* and *rules* of the application reside as well as the heart of the application security policy.

The e-MCES has been developed for all members of staff of the university, including academic/non-academic, administrative, technical and ancillary staff with different areas of responsibilities. Some of them are managers and supervisors while others are employees – and all of them are members of the one university's community. Because the system serves all of us in the institution, with a single Internet interface and common data storage, it must have restricted access to the resources. Our solution is based on the concept of the *role* of the currently logged-on user [1].

Using ***role-based security*** (or simply *RBS*), it is possible to programmatically determine the role/roles of the current user interacting with a given type or type member.

.NET provides four possible security policies:

1. *Forms:* A RBS implementation for ASP.NET;
2. *Generic:* Enables us to define our own custom RBS system;
3. *Passport:* A RBS implementation for MS .NET Passport;
4. *Windows:* A RBS implementation for Win32 user account system (32-bit implementation).

Because the Form-based principal policy is used extensively when securing ASP.NET in web applications, we have decided to use it in our solution. The .NET security model enables us to restrict access to type allocation and type member invocation using *Imperative RBS*, which types directly into the code, making run-time demands and decisions when needed. With this approach, we gain the capability to monitor access violation gracefully in the code via try/catch constructs and/or simply deny a given course of action.

Using *role-based security* (or simply *RBS*), it is possible to programmatically determine the role/roles of the current user interacting with a given type or type member.

Note *The process of creating users and assigning them to their respective roles, which is not covered here, is the responsibility of the System Administrator and is described in technical documentation of the system. We have created a comprehensive and easy on-line mechanism for maintenance the system.*

To programmatically obtain the identity of the current user via the *RBS* model, we must obtain a *principal object* from the current thread of exception via *Tread.CurrentPrincipal* object. Technically speaking, a principal object is some type implementing the *System.Principal.IPrincipal* interface:

```
public interface IPrincipal {
    IIdentity Identity { get; }
    bool IsInRole(string role);
}
```

Evidentially, the read-only *IPrincipal.Identity* property returns an object implementing *System.Security.Principal.IIdentity*, which is defined as:

```
public interface IIdentity {
    string AuthenticationType { get; }
    bool IsAuthenticated { get; }
    string Name { get; }
}
```

Before obtaining a principal object via *Thread.CurrentPrincipal*, the calling assembly needs to inform the Common Language Runtime (*CLR*) of the principal policy.

The programming procedures mentioned above allow the developer to build a flexible and secure mechanism for the user's authentication.

Actually, each user of the system has to follow the following main processes, presented in Fig. 6 [9].

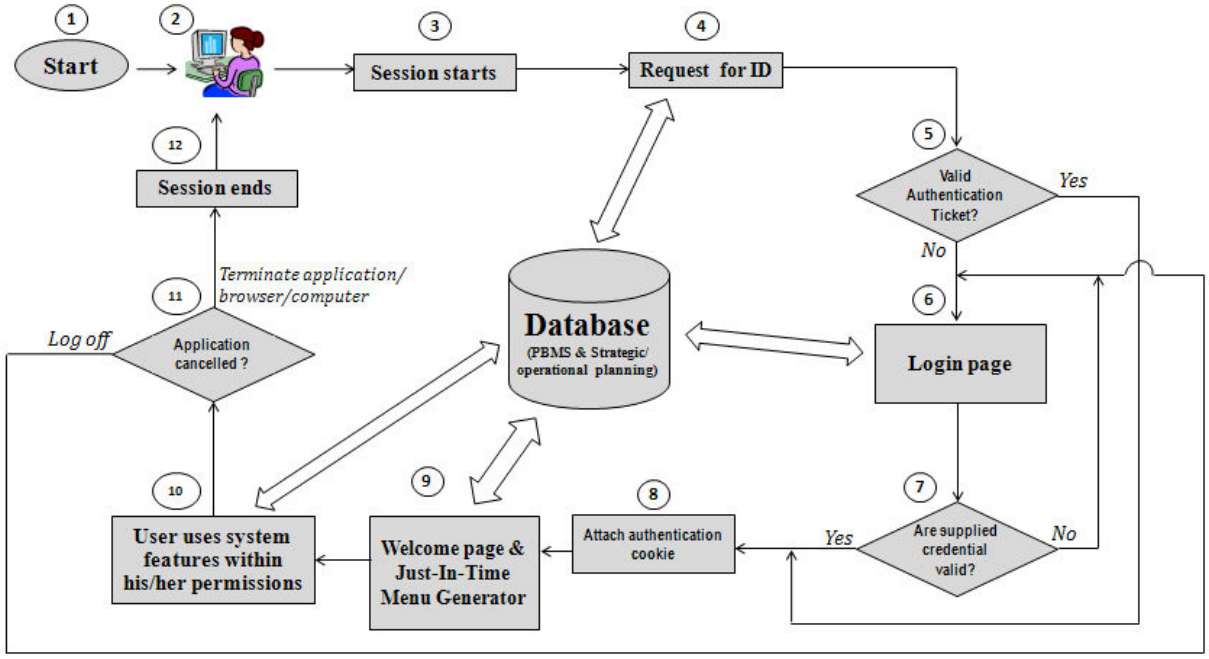


Fig. 6 Data flow diagram of the Management & Control Solution block

- (1) - Opening a session (Fig. 6, steps are marked by circle with numbers inside - 1, 2, 3),
- (2) – *Log-on* procedure or User Identification (Fig. 6, steps 5, 6, 7, 8);
- (3) – Granting permissions and generating Just-In-Time *(JIT)* menu (Fig. 6, step 9);
- (4) – Access to granted information resources for the user (Fig. 6, step 10);
- (5) – *Log-off* procedure (Fig. 6, steps 11, 12).

We will examine each of these steps in detail.

Opening a session (Fig. 6, 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 individually for this particular user a *session state object (SSO)*, which 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 the request. Due to limitation of space and because we are not writing the technical documentation, we omit the situation when browser's cookie is off; ASP.NET has other alternative techniques to bypass it.

Log-on procedure (*Fig. 6*, 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 the highest level of the *user's authentication*. The process of the user's authentication involves 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 Fig. 7 shows 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 the successful login.

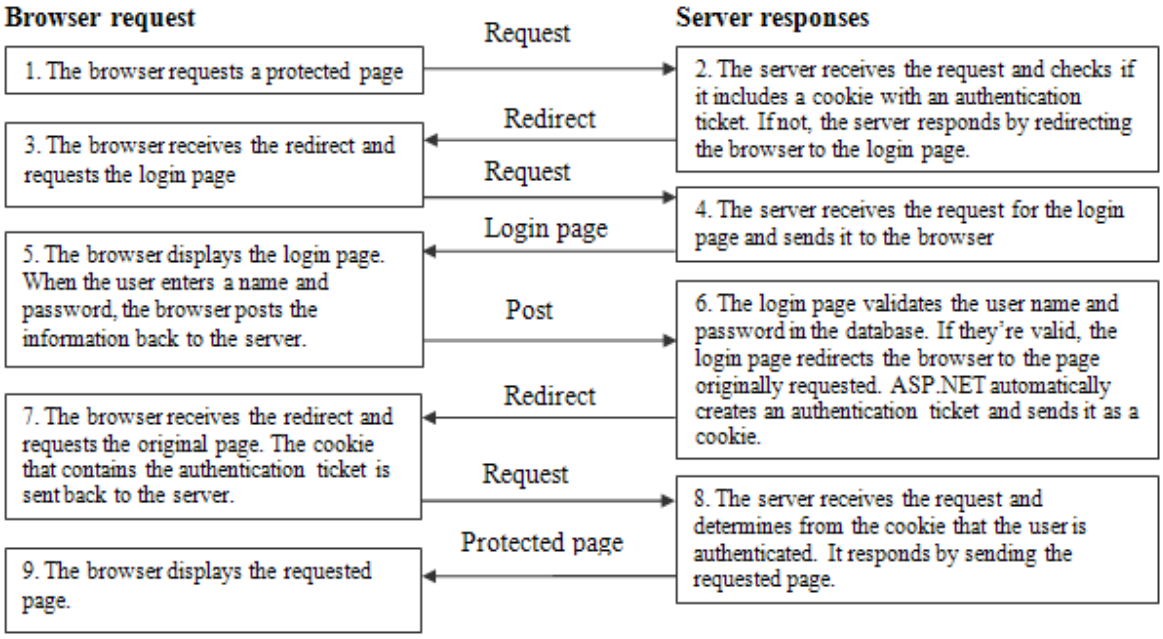


Fig. 7 HTTP requests and responses with forms-based authentication, according to steps 5, 6, 7, 8 [10]

The membership API provides a set of classes with wide range of methods to let the developer manage different users with differing 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's 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 is 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 expressed as follows:

```
<membership>
    <providers>
        <add name="AspNetSqlMembershipProvider"
             type="System.Web.Security.SqlMembershipProvider,
             System.Web, Version=2.0.0.0, Culture=neutral,
             PublicKeyToken=b03f5f7f11d50a3a"
             connectionStringName="LocalSqlServer"
             enablePasswordRetrieval="false"
             enablePasswordReset="true"
             requiresQuestionAndAnswer="true"                    (1)
             applicationName="/" requiresUniqueEmail="false"
             passwordFormat="Hashed"
             maxInvalidPasswordAttempts="5"
             minRequiredPasswordLength="7"
             minRequiredNonalphanumericCharacters="1"
             passwordAttemptWindow="10"
             passwordStrengthRegularExpression=""/>
    </providers>
</membership>
```

The content of the configuration file (1) describes a security policy of the application by values of its attributes. Some of them are:

- *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 [11, 12]. The e-MCES can modify security algorithm using *HashAlgorithm* 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 = HashAlgorithm.Create("MD5");
```

```
        byte[] fileHashValue = alg.ComputeHash(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 nonalphabetivcal 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 this is to guarantee a high level of security in the 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. Fig. 8 shows a portion of hierarchy for the Human Resources Department *(HRD)*. The number of *web.config* files on different levels identifies permissions to get access to some specific unit. For example,

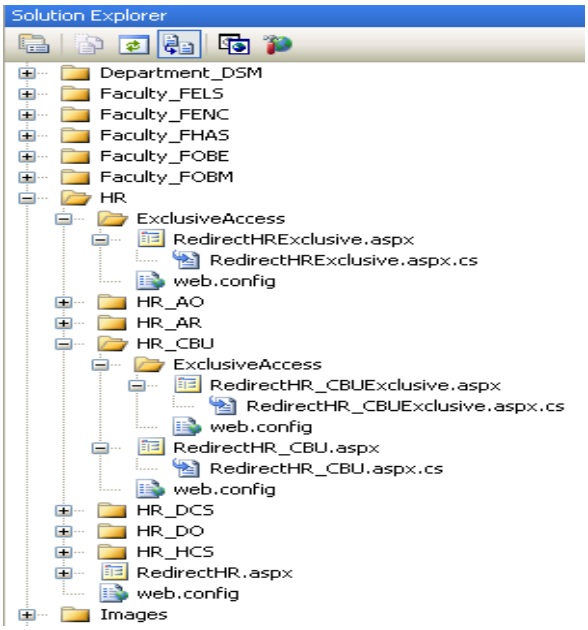


Fig. 8 A fragment the HRD hierarchy in e-MCES project and content of *web.config* files

- *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>
```

Here the *"HR_Senior_Director"* and the *"HR_CBU_Compensation_Benefits_Mgr"* are roles associated with the 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 *RedirectHRExclusive.aspx* or *RedirectHR_CBUExclusive.aspx* - see Fig. 8) of the structure, who is in the 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 other 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 indirectly using an interpreter. One of the main benefits of the ASP.NET in comparison with other web technologies, 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 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.MainCategories.Administrative.GetHashCode();
    Session["IsSupervisor"] = "Yes"; Session["IsHead"] = "Yes";
    Session["enableToSeeSupervisorComments"] = "No";
```

```
if (E == "Mng")
{
  Session["SupUnit"] = "";  Session["StaffCategory"] =
     PublicClass.Categories.UnitManagers.GetHashCode();
}
else
{
  Session["SupUnit"] = HR_AO; Session["StaffCategory"] = "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 other way to get any executive ASP page but only through this procedure.

Welcome page and Just-in-Time Menu Generator (Fig. 6, step 9)

A *menu* grants to a user his/her permissions. 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 some the staff migration between these positions is a usual process at high frequency. Traditional maintenance of these processes updating an application (redesign and recompiling), is waste of time and energy. In the next section we describe our original solution for this problem.

VII. Navigation and Just-in-Time Menu Generator

Navigation is a fundamental part of the e-MCES. In conjunction with security policy, it allows for the system to provide set of various *Menus* for different categories of users and their roles. The *menu* grants to a user his/her permission. In the University, like UTech, with number of employees more that 1,500, this is a big problem. We need to keep and maintain the *Menus* for all categories of staff and this is an enormous job. We have invented an novel solution for it in our system [13]. Actually, we do not keep a set of *Menus* in the system or in some supported configuration files at all. Instead, we keep them in the database as the fragments of any menu called as *tokens*. The system generates a virtual menu for each user *Just in Time Compilation (JITC)* of his request and keeps it during the current session of the user [13]:

- collecting essential tokens from the database, using our specific algorithm,
- at run-time generating a JavaScript menu program
- renders *HTML* code to the final *DHTML* code,
- sends it to the client browser.

The *Just-In-Time Compilation* process generates an HTML document, to be interpreted by the browser of the client's machine.

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 the necessary elements of the future menu – texts of items, which will be seen by user and are accessible (in terms of ability to be hyperlinks) to the user, destination addresses (URLs) associated with these items, and link to the user's role. Fig. 9 shows a fragment of database, which holds tokens for JITMG:

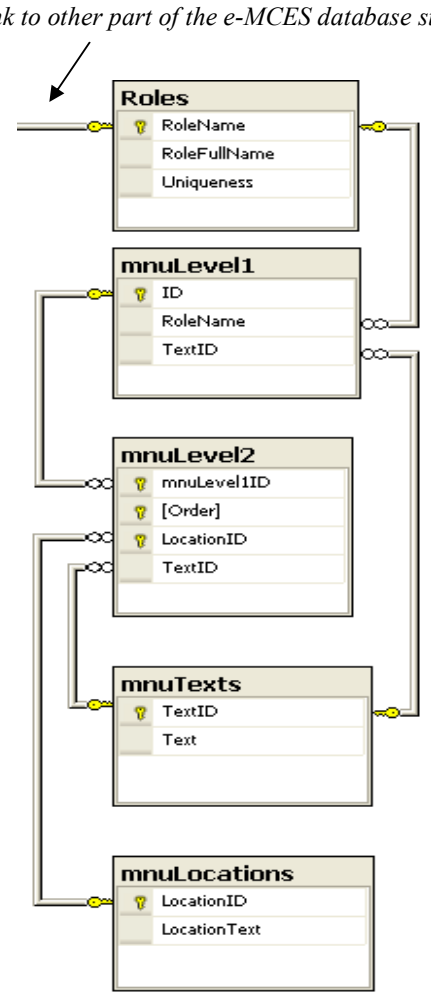


Fig. 9 Fragment of database, which holds tokens for *JIT Menu Generator*

Table *Roles* receives a text value with *User's role*. Then two SQL queries retrieve 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, utilizes a *Navigation* object from .NET Framework and generates a JavaScript program, which resides into a HTML response as its integral part.

Appendix A shows a snapshot of the browser's screen with menu, generated for user *Mrs. Jennifer Williams*, who is in the role of a *Senior Director of Human Resources Department*.

Appendix 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 to security reasons. 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.) are 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.

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 or to 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 the application – it is a usual routine for administrator of the system.

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):

##	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
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
13		maintenance of Admin. office operational plan (next
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
17		selection of the Admin office objectives/targets (next
18	Reports	view of Scores list of Managers and Supervisors
19		view of Scores list of HR office Administrator staff
20		view of Scores list of the Development unit's staff
21		view of Scores list of the Compensation & Benefits
22		view of Scores list of the Health Care Center staff
23		view of Scores list of the health Care Center Medical
24		view of Scores list of the Day Care Center staff

Table 1 The content of full menu for Senior Director of HRD

According to 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 [14]). The Senior Director of HRD can view 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 corresponding functions and responsibilities. e-MCES generates programming classes (ASP.NET and other supported modules), which handle the processes advertized in menu. Only authorized person can obtain functions provided by automatic menu only during the session time. After terminating session, it is discarded. It exists just in time for the user's session.

The user can terminate session by Log-out process, when he/she is pressing a *Logout* button, or by closing a browser, or by terminating a computer (Fig. 6, steps 11, 12).

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 detail.

VIII. Information Resources

Processes of adapting different Information Systems by academic and non-academic units of educational institution during different periods of time are characterized by different technologies and platforms used. Computer technology has been changing very fast and information systems created in earlier time, sometimes are not compatible with systems developed later, but anyway, they can be considered as an information resources [15].

Information resources of the e-MCES are based on the Human Resources database (HRDB) and University portal — Student Integrated Administration System (ISAS). ISAS

provides information to students about their assignments and exam results on-line. HRDB is built using the FoxPro database and has successfully served the staff of the HR department and management of the University for many years. ISAS is built using MS SQL-Server 2000 database and developed using VB6/ASP programming languages/environment. The e-MCES is based on other more contemporary platforms. Integration of these three different systems is a problem for developers. Our solution is based on the idea of creating a *"Pool of elementary objects*", which is populated by data derived from different currently using systems [15, 16].

This pool of objects is platform independent and allows the integration of different Information Systems of the University. In essence, these objects are independent of each other and store necessary data for populating database resources of information systems, we are going or planning to implement.

These technologies are well described in [15-17].

IX. Conclusion

In this paper, we have described the processes occurring during the user's session in e-MCES. We have selected here a role-based security approach in e-MCES in conjunction with *Form-based* authentication procedure, which are based on highest level of encryption technique (hashing procedures used a *MD5 & SHA* mathematical hashing algorithms). This is currently the most 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.

As a relatively young university, the University of Technology, Jamaica has made great strides improving a performance culture in the institution. In this regard, the leadership of the University played a critical role in designing and deploying the system to ensure its effectiveness.

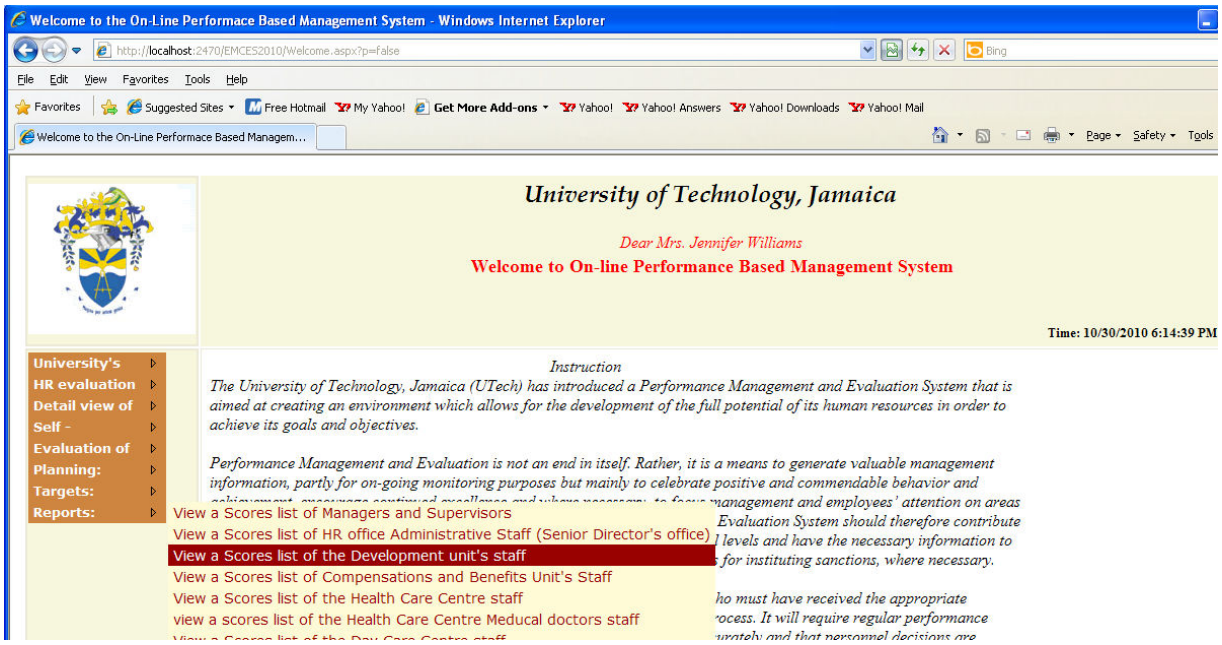
An important objective in developing the e-MCES was to 'connect' more closely each staff member to the planning and evaluation processes. Employee involvement is one of the best ways to create a positive culture that thrives on performance measurement. Through the Project Steering Committee, staff at all levels had an opportunity to provide input into all phases of creating the integrated management system and by so doing to create they are bought-in.

The intention of the e-MCES is to provide a robust yet flexible online tool for the University to measure how well it is able to achieve its strategic targets, while providing a mechanism for communicating with each staff member their individual contribution to the success of the institution.

A practical experience of using previous online evaluation systems in the University of Technology, Jamaica proves that we have developed the most effective way for improving the evaluation process of the University staff.

Appendix

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 = eventArgument;
      theForm.submit();
   }
}
//]]>
</script>
    . . . . . .
<script src="/EMCES2010/WebResource.axd?d=GJ56476UwNAULC-srfuP4Q2&amp;t=634203587922187500" type="text/javascript"></script>
<script src="/EMCES2010/WebResource.axd?d=5vuY-ZmLBPR4bkOEVL2FlA2&amp;t=634203587922187500" type="text/javascript"></script>
    . . . . . .

// Items of the parent menu
   <td style="white-space:nowrap;width:100%;"><a class="NavigationMenu_1 NavigationMenu_3 NavigationMenu_8"
   href="javascript:__doPostBack('NavigationMenu','~*|*ChoiceError.aspx?P=HR1')"> University's </a></td>
   . . . . . .
   <td style="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*|*RedirectPresident.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&am
   p;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 = 'NavigationMenu_17';
    NavigationMenu_Data.hoverHyperLinkClass = 'NavigationMenu_16';
    NavigationMenu_Data.staticHoverClass = 'NavigationMenu_15';
    NavigationMenu_Data.staticHoverHyperLinkClass = 'NavigationMenu_14';
//]]>
</script>
```

REFERENCES

[1] V. Pougatchev, S. Glasgow, J. Ellis, N. Johnson, (March 2009). 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, CA, USA, 2(1), pp. 53-78

[2] V. Pougatchev, A. Kulkarni, (2011, January-June). 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,* IV(I), ISSN: 0973-4414, pp. 243-252

[3] D. Esposito. (2006). *Programming Microsoft ASP.NET 2.0. Core Reference*, Microsoft, Microsoft Press, Redmond, Washington, USA

[4] A. Taylor, (2002). *JSP and Java TM: the complete guide to Web site development*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, ISBN 0-13-091813-X

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

[6] M. MacDonald. (2003). *Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting.* Microsoft Press, Redmond, Washington, USA

[7] T. Connolly, C. Begg, R. Holoweczak (2008). *Business Database Systems*, Addison-Wesley, NY, USA, ISBN 978-1-4058-7437-3

[8] V. Pougatchev, A. Kulkarni, (2011), (Invited paper), *Technical Aspects of the Online e-Management Control and Evaluation System for Universities*, Proceedings of the 10th WSEAS International Conference on APPLIED COMPUTER and APPLIED COMPUTATIONAL SCIENCE (ACACOS '11), Recent Researchers in Applied Computer and Applied Computational Science, Venice, Italy, March 8-10, 2011, ISBN: 978-960-474-291-3, ISSN: 1792-8559, pp. 22-25, *http://www.wseas.us/conferences/2011/venice/acacos/*

[9] V. Pougatchev, A. Kulkarni, (2010, June). Online Operationalization of Processes in e-Management Control and Evaluation System for a University. *International Journal of Computer Science and Communication,* ISSN 0973-4414, 4(2)

[10] J. Murach, A. Boehm, (2006), *"Murach's ASP.NET 2.0 web programming with C# 2005"*, Mike Murach & Associates, Inc., Fresno, CA, USA, ISBN: 1-890774-31-6

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

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

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

[14] V. Pougatchev, A. Kulkarni, (March 2011), 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, 4(1), pp. 14-31

[15] V. Pougatchev. (2008, September 29 – October 1). *Online Performance Based Management and Evaluation System at the University of Technology, Jamaica: Information Resources and Security Solutions*. Proceedings of the 11th IASTED International conference on Computers and Advanced Technology for Education (CATE-2008) Crete, Greece, *http://www.actapress.com/Abstract.aspx?paperId=34196*

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

[17] V. Pougatchev. (2006, October 4-6), *An Object-Oriented Approach in Development of On-Line Evaluation Systems for Educational Institutions.* Proceedings of the 9th international conference on Computers and Advanced Technology (CATE-2006), Peru, Lima

AUTHORS

Mr. Valeri Pougatchev is a Senior Lecturer in the School of Computing and Information Technology, University of Technology, Jamaica. He has a Master degree in Mathematics from the Voronezh State University, USSR, and is currently a PhD candidate at the University of Technology, Jamaica.

He has been working in the area of Information and Communication Technology (ICT) for approximately 30 years in Russia, Jamaica, and the United States. He is the author of two books, published in the Russian Federation, and more than 30 publications in reviewed International journals and conference proceedings. During the period 2004-2005 he served as a consultant of the UNESCO on applications of ICT in Tertiary Education and Vocational Training for the Caribbean Region and a principal investigator of the UNESCO Project on Standards for ICT-Based Education in the Caribbean Region.

In January, 2008 Mr. Pougatchev was nominated for the National Medal for Science and Technology. Jamaica. In October 2006 he received an Invited Speaker award on the 9th IASTED International CATE-2006 conference, Lima, Peru. In May 2007 he was a recipient of the University of Technology President's Research Initiative Award.

He is a member of the International Program Committees of fourteen International Computer Science conferences, Editorial Board of the International "Journal of Research in Innovative Teaching", National University, USA, 2008 and has reviewed more than 100 research papers since 2004

His research areas include but are not limited to information science, software engineering of enterprise-wide systems, research, design and development of an Integrated Management Systems for educational institutions.

Mr. Valeri Pougatchev is a member of ACM association, Jamaica Computer Society, Jamaican Historical Society, Jamaican-German Society, and a contributing member of the Cousteau Society, a nonprofit Environmental International organization dedicated to marine conservation, research and exploration, USA, France for more than 11 years.

His private interests are hiking, reading, classical music and SCUBA diving. He is a member of the International Professional Association of Dive Instructors (PADI), USA since 2007 with SCUBA diver qualification: PADI Dive Master

Professor Ashok B. Kulkarni joined the University of Technology, Jamaica in Kingston on September 15, 2008 as Deputy President (Academic Affairs and Quality Assurance).

Prior to that, he was:

- Dean, Faculty of Science and Technology, Gulbarga University (2005-2007), Professor & Chairman
- Dean of Applied Electronics, Gulbarga University (1987-2008)
- Associate Professor of Physics, Gulbarga University (1983-1987)
- Lecturer in Physics, Meerut University, India (1972-1983)
- Research Fellow, Department of Atomic Energy, Govt. of India (1970-1972)
- Research Fellow, Council of Scientific & Industrial Research (1969-1970)

He is a member of several scientific and professional organizations:

- Institute of Electrical and Electronics Engineers (IEEE)
- Fellow of Institution of Electronics & Telecommunication Engineers
- Committee for Nuclear Reactors, Ministry of Energy, Government of Jamaica since 2009
- International Centre for Environmental and Nuclear Science (ICENS), University of the West Indies, Mona Campus, Jamaica
- Panel of Experts for European Union Funding Agency for FP-7 schemes since 2009
- Editorial Board of the International Journal "Advances in Fuzzy Systems"

He has successfully supervised fourteen (14) PhDs, 30 MPhils, and more than 600 M.Sc. students, published approximately 90 research papers in Journals and 130 in Conferences.

He has also worked at the University of Technology, Lae, Papua New Guinea during 2008 for seven (7) months.

His major research interests included but are not limited to Nuclear Electronics, Reactor Engineering, PC-based Instrumentation, Modern Control Engineering, Fuzzy Logic Controller, Microwave Antennas and propagation.

He has reviewed several research papers for reputed Internal Journals since 2008.

His private interests are singing, listening to music, reading, collecting of coins, cricket and table tennis.