International Journal of Information Technology and Knowledge Management
January-June 2011, Volume 4, No. 1, pp. 243-252

MACROSCOPIC VIEW ON THE STRUCTURE OF THE E-MANAGEMENT CONTROL AND EVALUATION SYSTEM FOR THE UNIVERSITY OF TECHNOLOGY, JAMAICA

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

This paper describes a macroscopic structure of the e-Management Control and Evaluation System (e-MCES) for the University of Technology, Jamaica (UTech). In our University we consider this web application as a base for a full management system for educational institution that includes strategic, academic, and financial planning and management components. Actually, the system we have been developing is becoming a fully Integrated Management System (IMS) finally. The IMS will allow the institution to respond promptly to real-world challenges and opportunities that might affect its short- and long-term strategy. This system provides an accurate diagnosis of the educational reality and an objective assessment of the impact of intervention on policies and society. The information produced by current version of the System (e-MCES) can then be used as a social control tool to press those responsible for managing the educational system to produce necessary improvements. In this technical oriented paper we would like to share our vision on the structure of similar Systems, describe current and prospective solutions. We think that our ideas and proposed solutions will help to our colleagues from other educational institution to build similar systems.
The e-MCES system has been developed fully and is being implemented in some divisions of the University in staggered manner to test and validate the system developed. The result will be reported in near future.
Keywords: Performance Based Management, Strategic/Operational Planning Management, Module/Instructor Evaluation System, Productivity/Finance Planning, e-Management Control & Evaluation Systems, V-index, Role-Based Security, .NET Technology

1. Introduction

The current economic situation in the World, global financial crisis, dictates a new breed of technology solutions to ensure that the educational institutions can quickly adapt to a new market demands. Market conditions are changing faster and more dramatically than ever before. Any educational institution that fails to adapt quickly and efficiently to market changes can miss important opportunities or even risk their very survival.

Developing a strategy in response to a fast changing market is difficult. The University of Technology, Jamaica has taken a bold new approach to strategic and financing planning, utilizing the Balanced Scorecard as a strategic management tool [1], the best practice of using Performance Based Management and Evaluation solutions for Academic and non-Academic staff of the University. The University's approach [2] of developing its e-Management Control and Evaluation System internally, as an Internet Application, satisfies its requirements for flexibility, ease of use, and responsiveness to internal and external forces. Generally, this system is able to:

- provide appropriate information on each staff member's performance;
- align objectives and resources across the University vertically and horizontally;
- allow staff members the opportunity to identify their contributions to the achievement of the University's objectives;
- allow for consultation and agreement between the staff member and supervisor regarding the procedures to be used;
- facilitate timely appraisals and analyses;
- provide ongoing constructive feedback to the staff member;
- provide information to affect decisions for confirmation or tenure;
- equip the management of the University to identify and reward good performance;
- inform decisions regarding granting of incremental salary increases;
- assist in recommendations for promotion;
- provide information for sanctions to be taken where necessary;

[1]Academic Affairs Division, University of Technology, Jamaica
[2]School of Computing & Information Technology, University of Technology, Jamaica
Email: [1]akulkarni@utech.edu.jm, [2]vpougatchev@utech.edu.jm

- give staff members the opportunity to comment on the process and on the output; and make an appeal where necessary;
- contribute to professional development planning;
- provide the University with feedback on institutional deficiencies in areas such as supervision, evaluation, professional support or performance improvement.

From our point of view the system, which satisfied functionalities, mentioned above, should consist of the following components:

1. Management & Control Solution:
   a. Online Strategic, Operational Planning Management and Control System.
   b. Online Performance Based Management System.
2. Students' Services Solution: Online Module/ Instructor Evaluation System.
3. Financial Solutions: Online productivity and finance planning.

The general structure of the e-MCES is presented on the Figure 1.

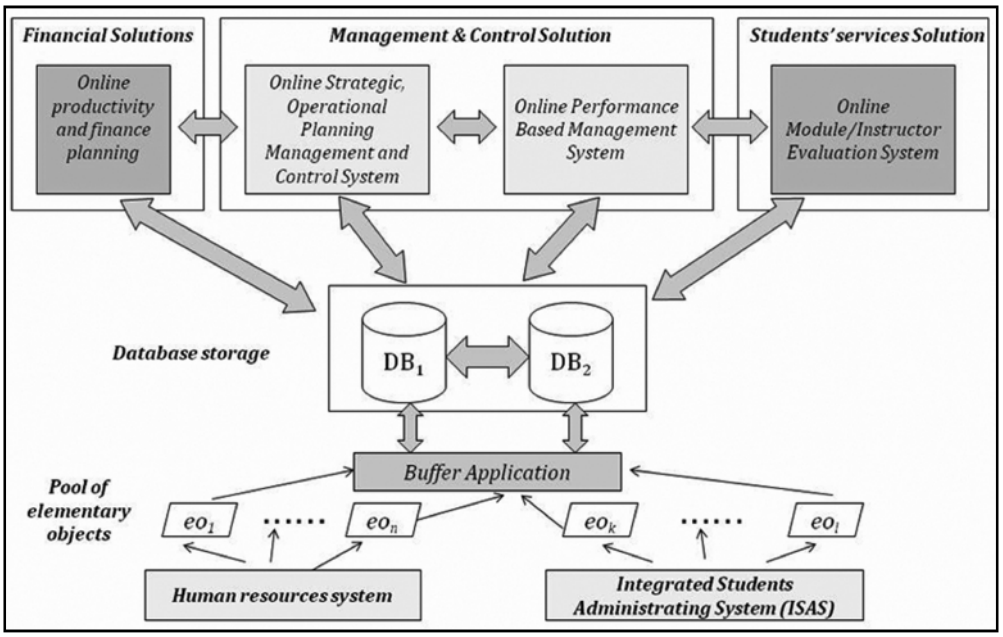

Fig. 1: General Structure of the e-Management Control & Evaluation System

One of the key ideas [3] in our research and development of the e-MCES for the university is: 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.

2. MAIN COMPONENTS OF THE e-MCES

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). ISAS provides information to students about their assignments and exam results on-line. HRDB is built using 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. e-MCES is based on other more contemporary platforms. An integration of these different information systems is a problem for developers. Our solution is based on the idea of creating a "Pool of elementary objects", generally shown on the Figure 1 and described in more details in [3]. In this approach information is collected from HR and ISAS automatically by software ("Buffer Application" in the Figure 1) specially designed for that purpose. This software creates a set of mutually independent Elementary Information Objects. They can be considered a Buffer between real information resources and e-MCES database-destination. In fact the process of getting information from HR and ISAS is a "one-way street", which means that it inherits valuable data from the "parent" system and populates the e-MCES, without

impacting on the data processes of the parent. That buffer is platform independent and allows the integration of different Information Systems of the University which, although developed under different operating systems, still successfully operate as a unified information system of the university. Some aspects of that solution are described in [4].

Online Module/Instructor Evaluation System (OLMIES): This is one of significant components of the e-MCES. According the UTech's terminology Module is a Subject, like for instance - "Introduction to Programming" subject. This system provides evaluations of modules delivered by departments/schools and of lecturers by students. We appreciate the importance of that feedback from the students and this is a valuable component of overall evaluation for academic staff. The OLMIES is flexible, anonymous, secure, efficient, and customizable. It is a database-driven system with strong data mining and reporting capabilities that aid effective decision-making, described in [5]. In 2004-2005 Academic Year, the first version of that System was piloting by our University for a number of students' groups online. Based on experience of this testing, currently we are in process of upgrading this system. In Figure 1, the components depicted on shaded background indicate fact that they are under construction or just only in our plans. In this paper, we describe some unique and very useful solutions we have discovered and implemented in this system.

Online Strategic, Operational Planning Management and Control Component: The success of each Educational Institution generally is based on many components – there are Performance of Academic and non-Academic staff, quality of educational programmes and their relevance to the real World demands, ability of the Institution to achieve their targets, which are based on Strategic, Operational planning and Strategy (University targets), etc. The last three components are critical from our point of view with ability of the Institution to achieve its targets, following its Mission statement. We suppose that the Ability of achieving its targets can be considered as a one of the main indicators of its success. In our research, we accept definitions given by U.S. Department of Energy [6]:

- Strategic Planning is a process of developing a mission and long-range objectives and determining in advance how they will be accomplished.
- Operational Planning is process of setting short-range objectives and determining in advance how they will be accomplished.
- Strategy (University Targets) is a plan for pursuing the mission and achieving the objectives.
- Mission statement (at the organization) tends to stand for long periods of time. But they should be examined and debated periodically both by those to whom the organization "reports" and by those accountable for carrying them out. A whole hierarchy of missions exists in a large organization, and each level derives its mission from the mission of the parent.

The last definition reflects the main ideas of organization's mission statement – to be dynamic, well structured on all levels of University's hierarchy, and inheritable.

We share the point of view of [7], that Strategic Planning, Operational Planning of each Academic and non-Academic Units of the University do not stand in isolation, all they are integral parts of the University Strategic Plan, further articulating the strategic objectives for some particular area cascading down to each Division, Faculty, School/Department, Unit, etc. The start and end point for the planning round process is, of course, the University Plan. The strategic goals and objectives are the ends to which we will strive (or what we hope) to accomplish in the long run (before the end of the planning period). They should be in line with the University's strategic plan and each objective should be clearly linked to delivery of the vision. At the same time, it is helpful to make distinctions between different kinds of the objectives:

- Ideal (never reached);
- Objectives (reached in the planning period);
- Goals (might not be reached in the planning period).

One of the main goals of our system is an ability to provide a Transparency of viewing of the University's and staff's achievements on all levels. Everybody must to feel that he/she is a member of one crew and be able to see his/her contribution in common success. It is easier for employees to understand what and why some goals need to be accomplished. We absolutely agree that if employees are part of the process, they will accept it. If they know there is no employee participation – (it doesn't matter how good the plan), it will not work. It will be a chaos in the University.

In [8], we have introduced a quantifiable, mathematical algorithm, based on ideas of Linear Algebra of calculating a Total Accomplishment of the University strategic goals. Actually at the final stage it generates a number, which we consider as an indicator of the University's success on the way of achieving its Goals. The main point here is that this algorithm is universal and can be implemented for getting the same indicator for any Unit of the Institution (Faculty, School, Department, Programme, etc.). We call this indicator as V-index. The set of University's V-indices (year by year) allows to Senior Management of the Institution to get a prospective view of its way to achieve its Strategic Goals.

3. MULTITIER ARCHITECTURE IN e-MCES

The e-MCES is built based on the one of the most popular approaches in web development is a "multitier" application, where number of tiers are greater than two. In e-MCES we have implemented a "three-tier" approach. The multitier Internet architecture provides scalability, robust, multitasking, highest level of security and performance, even with high volumes of data [9]. The tiers in this model are (see Figure 2):

- 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 saved in stored procedures within the database.

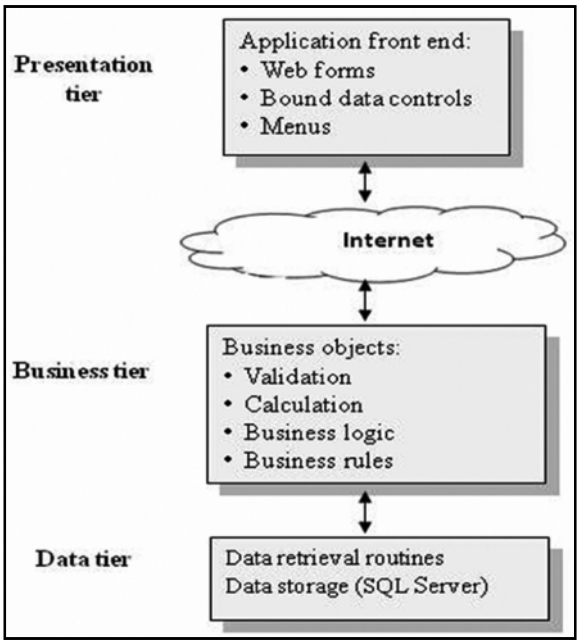

Fig. 2: Three Tier Architecture of Web Application

4. DEVELOPMENT PLATFORM DECISIONS IN e-MCES

The "e-Management Control and Evaluation System" is a Web 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. To some extent, all enabled 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 [10], JSP exploits the Java framework of classes and, with JavaBeans, that provides an effective extensibility model for reusing components. 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, (a first-class, compiled language) as opposed to the scripting languages used by both ASP and LAMP platforms.

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. What really differentiates ASP.NET from the plethora of other Web development technologies is the abstract programming model it propounds the Web Forms model. In addition, the whole ASP.NET platform comes as a native part of the Microsoft .NET Framework. ASP.NET applications are compiled pieces of code, are made of reusable and extensible components, can be authored with first-class language C#, which we have used in our process of development, and can access the entire hierarchy of classes in the .NET Framework.

MacDonald and Szpuszta in [11], enumerate seven important features about ASP.NET:

1. ASP.NET is integrated with the .NET Framework. The massive collection of functionality that .NET Framework is grouped into a logical hierarchical container is called a namespace. Different namespaces with different features, taken together, offer functionality for nearly every aspect of distributed development.
2. ASP.NET is compiled, not interpreted. ASP.NET applications are always compiled. ASP.NET applications actually go through two stages of compilation. In the first stage, the C# code we write is compiled into an intermediate language called Microsoft Intermediate Language (MSIL) code, or just IL. The second level of compilation happens just before the page is actually executed. At this point, the IL code is compiled into low-level native machine code. It is known as just-in-time (JIT) compilation.

3. ASP.NET is Multilanguage: It is up to developer to opt one programming language or another, when he/she develops an application. It is no matter what language the developer uses, because the code will be compiled into IL. In a sense, IL is the language of .NET, and it is only language that the Common Language Runtime (CLR) – compiler recognizes.
4. ASP.NET is hosted by the Common Language Runtime: The most important aspect of ASP.NET is that it runs inside the runtime engine of the CLR. The whole of the .NET Framework – that is, all namespaces, applications, and classes – are referred to as managed code. Some of the benefits are as follows:
   - Automatic memory management and garbage collection for inaccessible objects.
   - Type safety, which a compiled code assemblies to be completely self-sufficient.
   - Extensible metadata, which describes a code of application and allows for the developer to provide additional information to the runtime or other services.
   - Structured error handling, which allows developer organize error-handling code logically and concisely.
   - Multithreading, which allows calling methods, to read files, or communicate with web services asynchronously, without needing to explicitly create, new threads.
5. ASP.NET is object-oriented.
6. ASP.NET is Multidevice and Multibrowser: One of the greatest challenges web developers face is a wide variety of browsers they need to support. ASP.NET solves this problem in a remarkably intelligent way by using a rich suit of web server controls.
7. ASP.NET is Easy to Deploy and Configure: ASP.NET simplifies this process considerably. Because .NET Framework provides the same core classes, deploying an ASP.NET application is relatively simple. In most cases, developers simply need to copy all the files to a virtual directory on a production server.

   In view of the above, for writing code-behind modules, classes, and business objects in e-MCES, we extensively use C# 2005 programming language, as described in detail in [12], [13], and [14].

5. THE e-MCES PROJECT STRUCTURE

The entire e-MCES Project composed of .aspx, .ascx files, code-behind modules, classes, config files, etc., presents an Engine of the System (EOS). All these files should be located (as a one whole unit of software) in the Windows file system of the Application Server into a System of Folders (SOF). After detailed research, we accepted the idea that the organizational structure of the files should almost reflects a real hierarchical structure of the University. For future discussion we need to introduce two figures – Figure 3 and its counterpart - Figure 4 (see below).

In Figure 3, a part above the dashed-line belongs to Senior Management of the University. For simplicity sake, we consider a vertical hierarchical line, belongs to the Deputy President, who is a Head of the Academic Division in the UTech. All Academic units (Faculties, Colleges, Schools, and Departments, etc.) report to this person. Figure 4 downwards (the Figure 3) expanding to the Faculty of Computing & Engineering then to the School of Computing & Information Technology, etc.

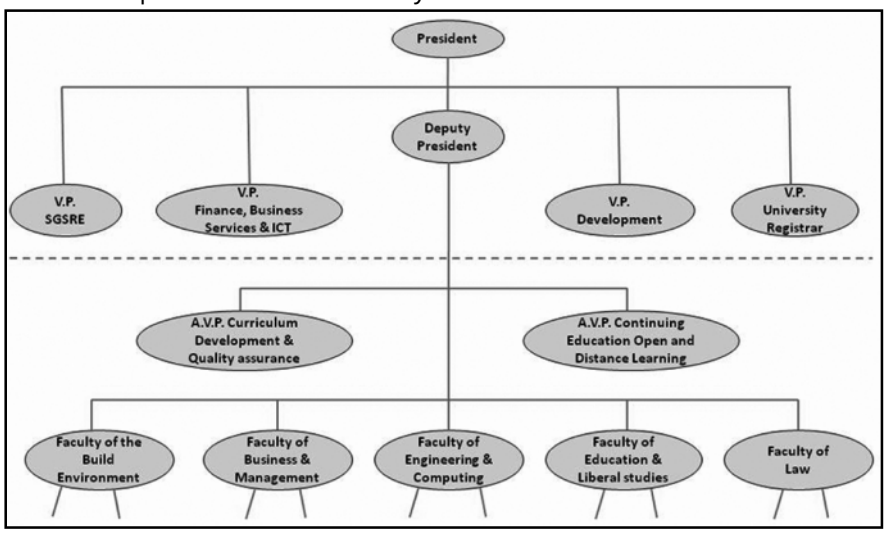

Fig. 3: Structure of the Senior Management up to Faculty Levels of the University of Technology, Jamaica

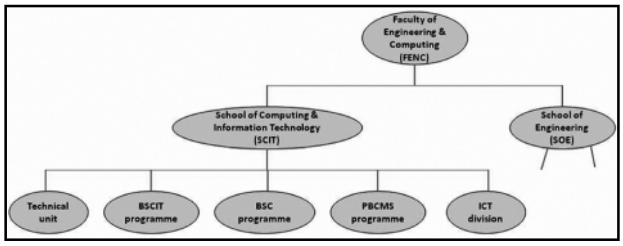

Fig. 4: Structure of the Faculty of Engineering & Computing Structure with Substructure of School of Computing & Information Technology

A portion of this structure, presented by Figures 3 and 4, is presented in Project Solution Explorer (Visual Studio 2005) by Figure 5:

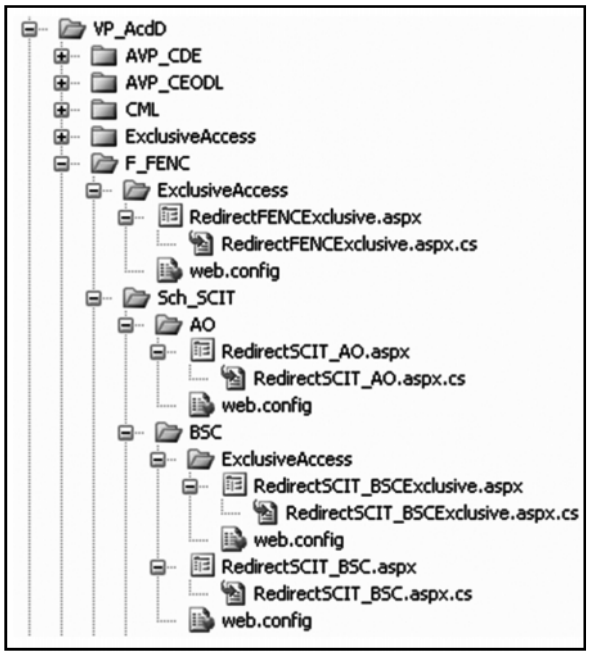

Fig. 5: Linkage of the Deputy President, Faculty of Engineering & Computing and other Unit's Structures as Presented in Visual Studio 2005

In Figure 5:

- Folder VP_Acd is a head folder of the Deputy President's Office (Vice-President, Academic Affairs Division).
- Folder F_FENC is a head folder of the Faculty of Engineering & Computing Dean's office.
- Folder Sch_SCIT is a head folder of the School of Computing & Information Technology (SCIT).
- Folder BSC is head folder of the Bachelor of Science in Computing Programme (BSC).

Here we must to note, that the start Folder for each hierarchy is a folder associated with Senior Management level of the UTech. Each oval object in Figures 3 and 4 presents a "node", which plays a significant role in entire strategic policy of the system. The general structure of each node is presented in Figure 6.

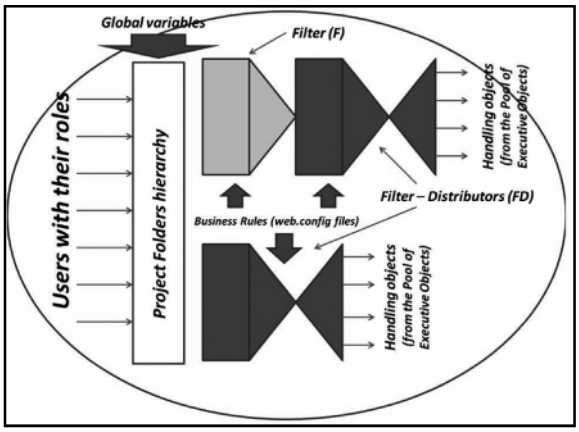

Fig. 6: Structure of Node, Associated with a Folder of the Project

The e-MCES is based on Role-Based Security (RBS), which we will describe in more detail in section 6 of the paper. Here we just make a glance on the File Folder structure of the Project. The Project has two main storages – one for modules (classes), which serve some specific functionalities or groups of functionalities, and another one is a security mechanism, which provides an access to these modules. Sometimes we call these modules as a Handling Objects (HO). All attempts to get access to the HO must go through the Nodes. The system does not allow getting access to HO bypassing a Node. If somebody tries to do it without valid authorization and authentication, he will be redirected to the Login page and this attempt will be registered in the Log journal (to register a fact of the illegal access attempt without a valid identification of the User for future investigation). In Figure 6, we can see two types of objects – Filter (F) and Filter-Distributer (FD). The security procedure on that stage as follows:

- F and FD accept the request from the User to get some functionality (to get access to handling objects);
- Based on Role, associated with User and data from web.config file (example of that file is in Figure 7), which serves current node, it will grant/not grant an access to the functionalities he is requesting.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
    <system.web>
        <authorization>
            <allow roles="Chancellor" />
            <allow roles="President" />
            <allow roles="Pro-Chancellor" />
            <deny users="*" />
        </authorization>
    </system.web>
</configuration>
```

Fig. 7: A web.config File of the Node of ProChancellor Folder

In the Figure 7, we can see that web.config file is a XML file. It authorizes access (filters users) to the content of ExclusiveAccess folder, which has a Distributer module that redirects User to the Handling Object he requests. In our example, this Filter allows to get access to HO only for Users, who play a role of Chancellor, ProChancellor or President. It denies access for anybody else. Important observation is – when the Project of the system will be compiled, .NET technology does not compile web.config file, and this allows the System Administrator to update this file using a simple text editor, like Notepad, without of recompiling the entire project/structure.

The next storage of the Project is a Pool of Handling Objects. It contains a number of modules, classes, which provide some functionality, offered by the system. This is a Kernel of the System. In Section 4 we mentioned, that the one of the great benefit of .NET technology is that each ASP.NET page is compiled, not interpreted. This factor makes the application very fast for executing, in comparison to other technology. The compiled enterprise Project can be located on the Web Server in compiled version, without source codes of the programs and modules. In fact this is very important advantage, due two reasons:

1. It will protect the original version of the system against unauthorized attempts to update something in the system, even by System Administrator.
2. It will keep confidentiality and intellectual rights on the version of the system currently being used.

6. Role-Based Security and More About Our Security Solutions

Security is an essential part of the e-MCES. This system has been developed for all members of staff of the University, including Academic/Non-academic, Administrative, Technical, Ancillary staff, etc. with different responsibilities and areas of responsibility. Some of them are managers and supervisors while others are employees – and all of them are members of the one university community. We have to evaluate a contribution to the success of the University by everybody. Because the system serves of all of us in the university, with a single Internet interface and common data storage, it must be possible to restrict access to the resources. Our solution is based on the concept of the "role" of the currently logged-on user, which is described in [2].

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 will not be covered here, is the job of the System Administrator and will be described in future publications.

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. It's interested in leveraging. .NET provides four possible principal policies:

- Forms: A RBS implementation for ASP.NET;
- Generic: Enables us to define our own custom RBS system;
- Passport: A RBS implementation for MS .NET Passport;
- Windows: A RBS implementation for Win32 user account system.

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 in code, making run-time demands and decisions where needed. With this approach we gain the capability to monitor access violation gracefully in code via try/catch constructs and/or simply deny a given course of action.

The process of getting access to the system by user is comprised of the following two steps:

1. The StartPage.aspx asks user to enter a University ID. At that step the system checks whether the user is a member of the university community using the

main data storage of the system – MS SQL Server. If the data entered by the user does not match the list of UTech IDs, it informs the user of this and does not allow progress into the system. This step is known as the Authentication of the user.

2. The second step of validation of the user is a process of Authorization. At that level we use a standard ASP.NET Login Control, which provides an out-of-the-box Web User Interface (UI) for the purpose of credential validation. Beyond offering the traditional UI, the Login control makes use of the specified membership provider to perform validation. Given all of this intrinsic functionality, we built a Login.aspx file with no code whatsoever, providing and nesting subelements into <asp:Login> tag, that map to the contain control. An ASP.NET uses a Membership class, which can be configured to perform user validation with various membership providers. The providers that ship with the Framework provide a "canned" implementation of data storage for user information. In this case we can use our own data store of user information, in parallel with an .mdf database under App_Data folder in the System project, which is created by Framework and can be seen in the Solution Explorer. To have an App_Data folder within a Project is a standard solution offered by .NET Framework. We improved this approach by moving the specific and secure database into SQL-Server Management Studio. This solution has become very useful for maintenance of the data and also for security reasons. Only Database Administrator has access to that DB now. The default membership provider is the local instance of Microsoft SQL Server. This out-of-the-box behavior is catalogued within the <membership> element in the machine.config file, which is located in the C:\WINDOWS\ Microsoft.NET\ Framework\ v2.0.50727\CONFIG:

```
<connectionStrings>
 <add name="LocalSqlServer" connectionString="data
source=.\SQLEXPRESS;Integrated
Security=SSPI;AttachDBFilename=|DataDirectory|aspnetdb.mdf;User Instance=true"
providerName="System.Data.SqlClient" />
</connectionStrings>
. . . . . .
<membership>
 <providers>
<add name="AspNetSqlMembershipProvider"
type="System.Web.Security.SqlMembershipProvider, System.Web, Version = 2.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
connectionStringName="LocalSqlServer" enablePassword Retrieval="false"
enablePasswordReset="true" requiresQuestionAndAnswer= "true"
applicationName="/" requiresUniqueEmail="false" passwordFormat="Hashed"
maxInvalidPasswordAttempts="5" minRequiredPassword Length="7"
minRequiredNonalphanumericCharacters="1" password AttemptWindow="10"
passwordStrengthRegularExpression="" />
 </providers>
</membership>
```

Fig.8: Settings of the System in machine.config/web.config Files

Most of these attributes are thankfully self-describing. Because of the page limit, we are not going to discuss these attributes in detail, although that would be worthwhile. We just need to emphasize that these settings can be overridden by settings of the web.config file in the application. Using .mdf database in our application allows us to provide a high level of security for sensitive information like Username, Password, Secret question/answer, etc., converting and storing them in the DB into hash encrypted version, using settings (Figure 8), with an actual value (password, for example) in the database like this:

"UF4s4VbBvLooupH4Bvv2dIIZZWM="

This means nobody (including Database Administrator) can use this information to gain illegal access to the specific functionalities of any user of the system.

7. Navigation and Just in Time (JIT) Menu Building Process

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. In the University, like UTech, with number of employees more that 1,300, this is a big problem. We need to keep and maintain a Menus for all categories of staff and this is enormous job. We have found a very original solution in our system. Actually we do not keep set of Menus at all. Instead of it, we keep in the database the fragments of any menu as a tokens. The system generates each virtual menu for each user Just in Time (JIT) of his request and keeps it during the session of the user:

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

When User logs out (or not connected to the system for long time) the virtual menu is destroyed and deleted from the memory. We consider that solution as a University Intellectual property and cannot deliver in detail in this paper.

Basically the Menu solution we have discovered and implemened in the system, does not depend of the size of the Institution and number of roles of their employees. It is easy to maintain, to create a new account for the new User with new role, without rebuilding and recompiling of the entre system.

8. Conclusion

As a relatively young university, the University of Technology, Jamaica has made great strides improving the robustness of its strategic planning process and aligning it with a measurement and evaluation system in order to create a performance culture in the University. In this regard, the leadership of the University plays a critical role in designing and deploying the System to ensure its effectiveness. It is acknowledged that even clearly stated and well-defined goals and expectations would be of little value unless the University, its Divisions, Departments, Units and individual members were apprised of performance against goals in a timely and useful fashion. Effective feedback, therefore, is a crucial ingredient in the System and this is facilitated by way of periodic reporting of progress and results against identified objectives and expectations.

An important objective in developing the e-MCES was to "connect" each staff member more closely to the strategic planning and evaluation process. Employee involvement is one of the best ways to create a positive culture that thrives on performance measurement.

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

References

[1] Kaplan, R., and Norton, D. (1996), "Translating Strategy into Actions. The Balanced Scorecard", Harvard Business School Press, Boston, Massachusetts, USA.

[2] Pougatchev V., Glasgow S., Ellis J., 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, 2, Issue 1, 2009.

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

[4] Pougatchev V., "Development Evaluation Systems for Educational Institutions using Reusable Information Objects", The 10th International Conference on Computers and Advanced Technology for Education CATE-2007, Beijing, China, October 8-10, 2007.

[5] Pougatchev V., George N., Lue G., Williams R., "Developing an On-line Course/Instructor Evaluation System", Advanced Technology for Learning, 3, No. 3, November 2006, ACTA Press, ISSN: 1710-2251.

[6] U.S. Department of Energy, Guidelines for Strategic Planning (DOE/PO-0041), January 1996, http://orau.gov/pbm/links/sp-giud.pdf.

[7] Hamill Jo., "Knowledge Management Operational Plan and Milestones", August 2006 KMC: 21-08-06, Knowledge Strategy Committee, the University of Edinburgh.

[8] Pougatchev V., "A Quantifiable Approach in Strategic and Operational Planning in Educational Institutions", The 12th IASTED International conference on Computing and Advanced Technology for Education CATE-2009, 22-24 November 2009, U.S. Virgin Islands.

[9] Hayen Roger L., (2007), "SAP R/3 Enterprise Software: an Introduction", McGraw-Hill, Irwin, ISBN-13: 978-0-07-299067-6.

[10] Esposito Dino, (2006), "Programming Microsoft ASP.NET 2.0", 2005 Edition, Microsoft Press, Redmond, Washington, USA.

[11] MacDonald Matthew, (2003), "Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting", Microsoft Press.

[12] Nagel C., Evjen B., Glynn J., Watson K., Skinner M., Jones A., (2006), "Professional C# 2005", Programmer to Programmer. Wrox. Wiley Publishing, Inc., ISBN -13:978-0-7645-7534-1.

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

[14] Ozu Nikola, Watt Andrew, Marcus Daniel, Williams Kevin, Birbeck Mark, (2001), "Professional XML, 2nd", (2001), Wrox Press Ltd., Birmingham, UK.