

## DEVELOPMENT OF CMS AS A RICH INTERNET APPLICATION

**Zoran T Lovreković**

Higher Education Technical School of Professional Studies, Skolska 1, Novi Sad, Serbia,  
lovrekovic@vtsns.edu.rs

**Original Scientific Paper**  
**doi:10.5937/jouproman6-18295**

**Abstract** - The paper discusses the development of web CMS as a rich Internet application. In the case of the development of a real CMS, technologies for making such an application have been analyzed, as well as the methods of making a web service for communicating the client to the server side. Ways to handle the error are suggested. Alternatives to the way of querying the database were investigated. The problems were identified in the development of this RIA, and the proposed solutions were discussed. The characteristic looks of the screens of this application are shown which show a qualitative difference between the classic web application and the RIA. Appropriate recommendations that can serve as a guide to the development of such applications are given.

**Keywords** - CMS, RIA development Apache Flex, AMFPHP

### 1. Introduction

In order for the Higher Education Technical School of Professional Studies in Novi Sad to have a better and stronger connection with high school students as a their potential students, the School Council decided that it is necessary to develop a web application that will enable all interested high school students to monitor relevant news competitions, trainings, and courses organized by the School specifically for this Social Group, as well as to enable them to exchange experiences and ideas and help each other through appropriate forums and discussion groups. Web applications must be developed as CMS so that all participants can easily upload and download relevant content and participate in communication and exchange of attitudes, ideas and knowledge. In addition, a Rich User Interface is needed, in order to further develop the sense of ease and benefits of using the application.

The classic web applications have to create a user interface of just a few html forms elements (text boxes, text area, check box, radio button, select menu and buttons). By using CSS and Javascript, it is possible to slightly enhance this interface, and even provide asynchronous responses to individual display components, but with a lot of effort, and few positive effects compared to the capabilities of the classic desktop applications [1].

### 2. Selection of technologies for RIA developing

There are three dominant platforms in the market for the development of web applications with a rich interface: Apache (former Adobe) Flex, Microsoft Silverlight and Java FX, although Microsoft is suspending Silverlight support, and Android has announced that it will suspend Flash Player support that is necessary for execution of Flex-based applications. By comparing the capabilities of these three technologies, it is easy to see that Apache Flex offers far more possible component of the user interface, and that it is open to connect to server-based technologies of a different type, unlike Silverlight that relies on Microsoft's platform, or Java FX Oriented exclusively to Java.

Also, for Flex, there are a number of development environments, some of which are commercial and some free, Adobe Flash Builder IDE has been selected as the development environment based on the Eclipse IDE for Java programming language.

Flash Builder has all the features (Figure 1) that should provide one such environment: standard features: syntax highlighting, automatic code completion, closing brackets, auto-slashing blocks of commands, code blending, class-type search, dynamic syntax checking, navigation to declarations and debugger (a debug version of Flash Player must be installed), and advanced features: finding a reference to a declaration (in a file, project or workspace), code refactoring (secure renaming of declarations and file transfer), code versioning (CVS), profiler (a set of functionality that allows code optimization), network monitor (overview of communication between server and client ), "Wizards" for connecting visual components in both directions with server technologies, design mode (visual functionality of user interface development) [2].

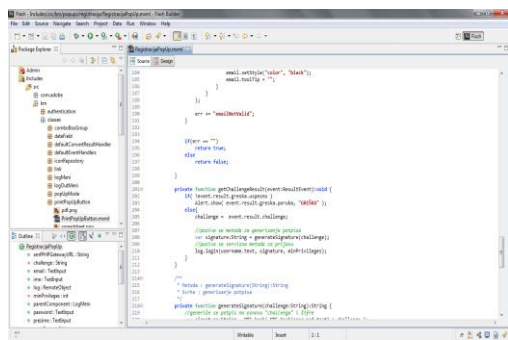


Figure 1. Example of the Flash Builder IDE screen

The Flex development package offers three components for communicating with the server part of the application[3]:

- RemoteObject (remoting via AMF protocol)
- WebService (SOAP Web Services)
- HTTPService (HTTP POST and GET requests).

According to Adobe documentation, the first option is better than the other, because it allows communication with less use of user memory, smaller size messages, and faster-processed calls. Performance tests confirm that AMF remoting has a significant advantage. In addition to good performance, a significant factor in choosing a way to communicate is good support for AMF server technologies. Taking all these into account AMF remoting has been selected as the way to communicate when creating an application[4].

MySQL and PHP are used on the server side. The decision to apply PHP and MySQL was very simple. Both open source technologies, very flexible, work under various types of web servers, and a large number of very serious web applications of successful world companies are references to these technologies.

For PHP, at the moment, dozens of IDE can be used. It was decided to use Aptana Studio (version 3.4.2). This development environment is like Adobe Flash Builder built on the Eclipse platform, and with it shares the standard functionality (Figure 2).

For the AMF protocol in the programming language PHP currently exists two extensions: -php-amf3 (<http://php-amf3.sourceforge.net/>) which is free and a Baguette AMF (<http://www.baguetteamf.com/>) which is commercial and four free libraries: AmfPHP (<http://www.silexlabs.org/amfphp/>), SabreAMF (<https://github.com/evert/SabreAMF>), WebORB for PHP (<http://www.themidnightcoders.com/products/weborb-for-php>) and Zend\_Amf (<http://framework.zend.com/manual/1.12/en/Zend.Amf.html>).

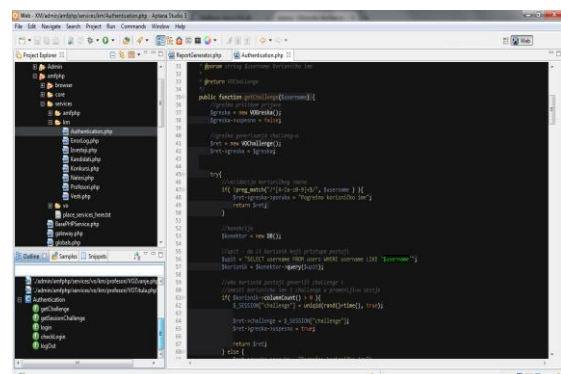


Figure 2. Example of the Aptana Studio IDE screen

Php-amf3 has limited capabilities and provides encoding / decoding of data for the AMF protocol. Baguette AMF can be used as an additive for AmfPHP and do it up to 20 times faster[5]. Unfortunately, this acceleration currently costs \$ 1200.

AmfPHP is the easiest to use from available libraries. This library allows you to build a remote service using the AMF protocol, as well as mapping user-defined classes. Also, the AmfPHP library also comes with a web application for testing the service. For the

purposes of writing ValueObject classes, class diagrams and drawing tools for UML diagrams can be used, which have the function of generating code. At the moment, there is only Software Ideas Modeler as a free tool on the market that allows generating both ActionScript and PHP code. There is also Enterprise Architect, but this is commercial software.

### 3. Web service development

AMF remote services were developed using the AmfPHP library. Based on the class diagram we get the ValueObject (VO) class. These classes have only attributes. Services are classes that have only the service methods that are called from the client side. Service methods can be roughly divided into two groups:

- those that return data on the error in execution
- those that return the list of objects and error data.

The list must be implemented as a string, since this type of data has both PHP and ActionScript. This fact and the nature of the service methods leads to two possible VO designs whose classes will have the feedback values (responses) of service methods. One solution is that one VO class is projected for the response method (Figure 3). The disadvantage of this approach is that some service methods do not have to belong to the two previously observed groups. Also, any objects can be found in the array of objects.

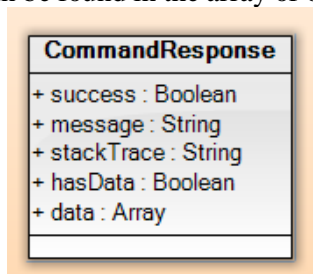


Figure 3. Inappropriate response of the service method

A better design would be to extract the error data into a special class VOGreska (Figure 4) whose instances will be the answers of the first-type service methods. For each method of the second type is written a class that contains a set of objects of the corresponding VO class (VOTitula) and instance of the class

VOGreska. This approach is applied in the development of the services for this web application.

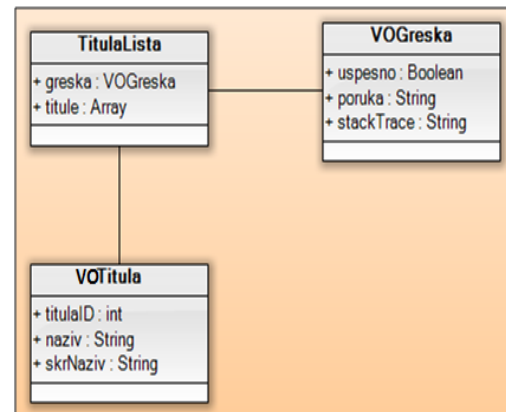


Figure 4. Principle of designing a VO class

The next step in creating a service is the choice of how to interact between the service and the MySQL database management system. Options are:

- query execution
- Execution of prepared statements
- Calling stored procedures.

The last option is not suitable because the files from the user side are placed in the file system of the server and carry with them the interaction with resources that are not objects of the database management system. Part of the code that interacts with the file system can be implemented in a different way, but then the service uniformity is lost. The execution of the prepared statements takes precedence over the usual querying when more than one query executes with different parameters in one execution of the service method. Since the prepared statements perform the necessary actions preceding the execution before the first query, all subsequent queries are performed more quickly. Considering the nature of the application, most service methods will have one execution of a single query by calling, so the first option is most appropriate. In the event that one query is executed several times with different parameters, the prepared statements will be applied.

The next step is to select the PHP API to work with MySQL. PHP has three such extensions (libraries). MySQL extension is obsolete in many way.

The library must have an object-oriented API and must enable transactions to be successfully applied when creating this type of web application. PDO, as well as mysqli, has an object-oriented interface, enable prepared statements and stored procedures, as well as transaction support. However, the PDO provides more stable support for error handling using exceptions, so this library will be used to write the service. PDO also provides appropriate driver[6] that enable work with other database management systems with unaltered code.

A database connection and queries are done by instantiating the DbConector class and invoking its methods. Listing of DbConector class is given as follows:

```
////////////////////////////////////D
bkonektor.php////////////////////////////////////
////////////////////////////////////
<?php
require_once('SistemPromenljive.php')
;

/**
 * Connection to the database
 */
class DbKonektor extends PDO {
    /**
     * Initialization of the object
     *
     * @param void
     */
    public function __construct() {
//loading of system variables
        $systemskep =
        SistemPromenljive::getSistemProm();

        $host =
        $systemskep['dbhost'];
        $baza =
        $systemskep['dbime'];
        $korisnik =
        $systemskep['dbkorisnik'];
        $lozinka =
        $systemskep['dblozinka'];

//database connection - parrent
//constructor call
// UTF8 charset
        parent::__construct("mysql:host
=$host;dbname=$baza;charset=utf8",
        korisnik, $lozinka);

//error exception throw
```

```
        $this->setAttribute(PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION);

        //default database
        $this->query("USE $baza");
    }
}
/**
 * Documentation of service
error for service methods that have
no interaction with the database
 * @param Exception $error
 * @return void
 */
public function
reportError($error) {
    self::logError($error);
}

/**
 * Documentation of service
error in database
 * @param Exception $error
 * @return void
 */
public static function
logError($error){
    $systemskep =
    SistemPromenljive::getSistemProm();
    $host =
    $systemskep['dbhost'];
    $baza =
    $systemskep['dbime'];
    $korisnik =
    $systemskep['dbkorisnik'];
    $lozinka =
    $systemskep['dblozinka'];

    $link =
    mysqli_connect($host,
    $korisnik, $lozinka, $baza);

    //error writing in table
//'errors' of the database
    $upit = "INSERT INTO
errors(
                tip,
                poruka, linija, fajl, backTrace,
                datum, vreme)
                VALUES".

    sprintf("( '%s', '%s', '%s',
's', '%s', ",
    "SQLSTATE[".$error->getCode()."]",
    mysql_real_escape_string($error->getMessage()), $error->getLine(),
```

```
mysql_real_escape_string($error-  
>getFile()),  
mysql_real_escape_string($error-  
>getTraceAsString()));  
  
"CURDATE(),CURTIME());"  
  
    mysqli_query($link, $upit);  
  
    mysqli_close($link);  
}  
  
}  
?>
```

An example of using the DbConector class:

```
public function getTitulaList() {  
    // reading error  
    $greska = new VOGreska();  
    $greska->uspesno = true;  
  
    $ret = new VOTitulaLista();  
    $ret->greska = $greska;  
  
    try{  
        //database connection  
        $konektor = new  
DbKonektor();  
  
        $upit = "SELECT  
titulaID, naziv, skrNaziv FROM  
titula";  
        $rezultat = $konektor-  
>query($upit);  
  
        $ret->titule =  
$rezultat->fetchAll(PDO::FETCH_CLASS,  
"VOTitula");  
  
        // release of resources  
        $rezultat-  
>closeCursor();  
    } catch(Exception $e) {  
        $greska->uspesno =  
false;  
  
        $greska->poruka =  
"GREŠKA>>".get_class($th  
is)."__FUNCTION__":\  
n\n". $e->getMessage();  
        $greska->stackTrace =  
$e->getTraceAsString();  
        $ret->greska = $greska;  
        $konektor-  
>reportError($e);  
    }  
  
    return $ret;  
}
```

#### 4. Application development

There are two basic parts of the application: front-end and back-end. They will certainly be split into separate Flex projects, which will produce (after compiling) separate SWF files. There are also three types of users in the administrative area (administrator, professor and student), and separate pages will also be used for them (as a separate Flex projects). Depending on the group to which the user belongs, after entering valid access parameters, it will be redirected to its page in the index file. However, these three types of users share functionalities according to the application description, so there is a dilemma whether to separate common parts into separate Flex projects. These projects will produce separate SWF files that will need to be clicked on the link on the screen of the appropriate group of administrator users to be load into Flash Player. This will cause delays in the application. Another solution would be to separate this common code into separate classes and embed it into each screen individually. However, then the initial screen start time will be larger. It has been decided to apply the first design method with one of the main assumptions, that is, the loading delay will be within reasonable limits. After this consideration, it can be noticed that the separate Flex projects will have the need for the same components and classes. To avoid repeating the code, we need to find out how to share it between projects.

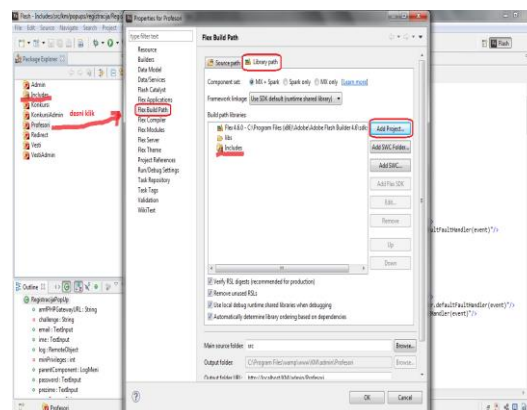


Figure 5. Split the shared code into a shared Flex Library project

Below are the characteristic screens of the web application developed in the previously described way:

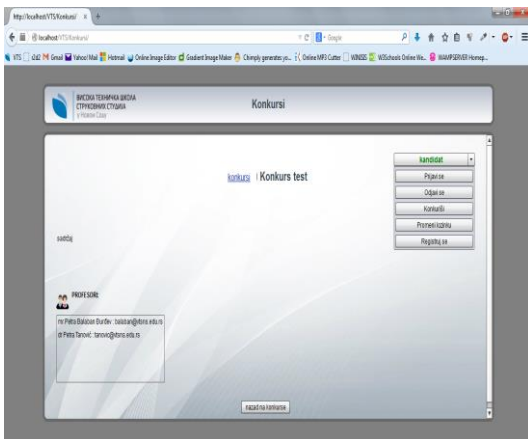


Figure 6. Displaying information about the contest for students

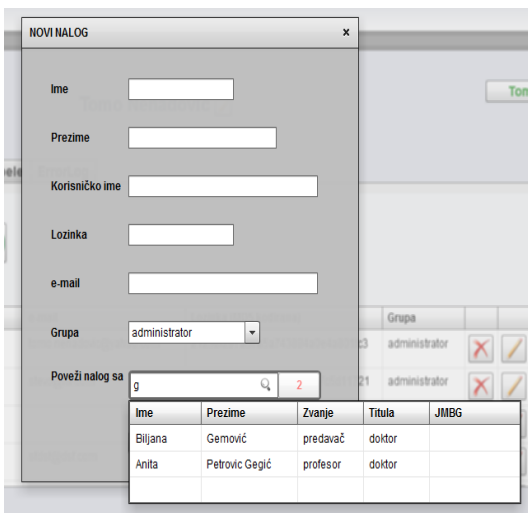


Figure 7. Advanced search

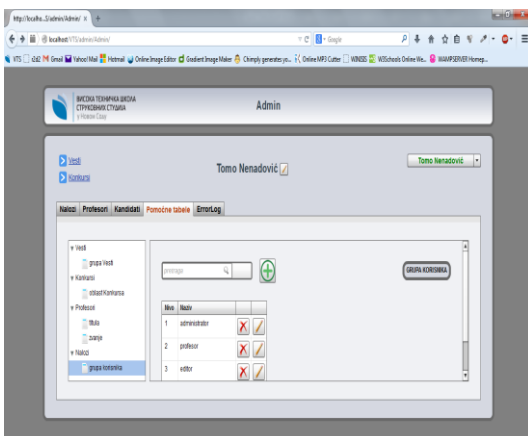


Figure 8. Admin Page

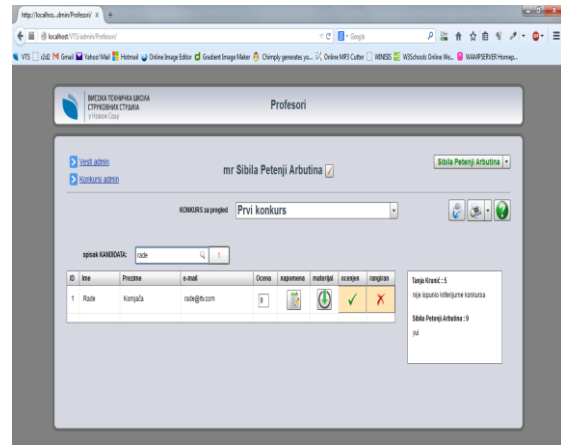


Figure 9. One of the professor's screens

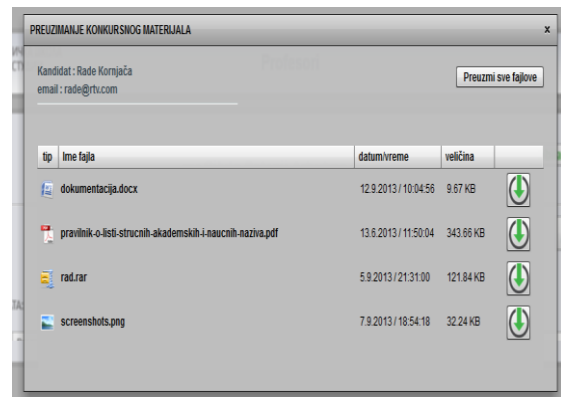


Figure 10. Downloading the work that the candidate has uploaded

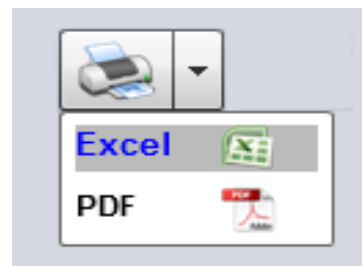


Figure 11. Printing Button

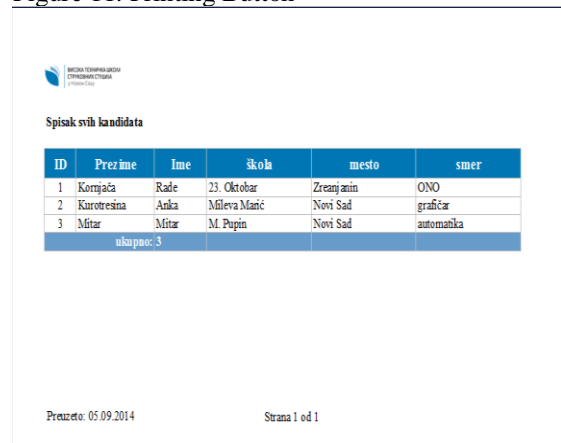


Figure 12. Printed documents

## **5. Discussion of the solutions**

The FLEX software development package has a rich collection of graphical components for creating user interfaces. The component that needs to be significantly improved to meet the needs of CMS is RichTextEditor for formatting text input. The current version of this component provides only the most basic features. On the other hand, some components have proved not to be too useful, such as ToggleButton and PopUpAnchor. One of the richer parts of the graphics package are the tabular display components. We massively used the DataGrid table for our application, but FLEX also offers AdvancedDataGrid and OLAPDataGrid[7]. For better arrangement of larger amounts of components FLEX offers advanced container components such as TabNavigator, Accordion, ViewStack, and Tree, which also have found their place in the application.

The styling of components using skins turned out to be very flexible, but also too complicated for the simplest cases in which the older architecture of the development package and the use of CSS would be much more convenient.

Connecting usernames to remote servers is possible in a simple way using the RemoteObject class. These classes support processing the events according to the DOM3 standard, as well as the entire FLEX development package.

Finally, from Flash Player 11, JSON is supported with standard parse functions. However, support for hedging algorithms such as MD5 or SHA1 is still lacking.

Several new components have been developed for the application needs. In their utility in all similar applications, ComboBoxGroup and SearchField stand out. The development of these components was not trivial, but the effort paid off.

During application testing in the development phase, the assumption that separate SWF files will load at a sufficiently small time interval has reached the lower limit of the certificate. In the final version of the application, few separate entities should be made. Also, during the development phase of the AMF remote

service, the speed of service has been questioned. After testing the speed of operation of all other communication options, and considering the architecture of the existing application, a better solution might be the implementation of the JSON-RPC service

The PHPEXCEL library proved to be a valid solution for generating reports. If an optimal report generation rate is to be desired, the fromArray function for filling in the table and applyFromArray[8] for styling should be used. Otherwise, the file generation speed increases linearly with the number of rows of the table.

## **6. Conclusion**

Apache Flex proved to be a suitable software development package for creating a CMS system with a rich user interface. The development package offers a very rich library of graphic components for creating user screens. Component styling is possible to the smallest details using skins. The attractiveness of the user interface can be upgraded using transitions and animations. Linking user pages to the server layer of the application is possible for almost every mode of communication[9].

Adobe Flash Builder enables accelerated application development and provides a complete set of advanced features. In the development of the application, Network Monitor, code refactoring and finding of the declarations in Workspace were particularly useful. In addition to these advanced functionality to test the best way of communication between Flex and PHP, the development environment is provided by Profiler. If AMF remoting is selected, creating a Value Object class without using the code generation tool based on a class diagram takes a long time.

Aptana Studio IDE responded to the needs of creating a remoting service, but in some cases the lack of refactoring has proven to be a major drawback. Also, this development environment does not provide support for designing WSDL when creating a web service. If any of the future versions of the application will be used by WSDL, another development environment must be used, such as Eclipse for PHP developers with the installed WTP (Web Tools Platform) plug-in.

In the case of this application, PHP has proven to be a good solution for creating the server part of the application. At the moment, AMF does not have enough good and free support in the form of extensions, but there are packages like AMFPHP with the help of which remoting services can be created. In some cases, with open source software, there are problems of non-existence of comprehensive solutions. Such a case is with the PHPEXcel library that only allows you to generate Excel and PDF files.

### **References**

[1] Lovreković, Z., Sukić, Ć., Šabanović, M., (2010), Desktop, web and web2.0 application comparing on the aspect of the realisation possibility the complex user interface, TTEM Journal, Vol. 5, No. 4, pp. 678-688

[2]<http://www.devx.com/RichInternetApps/Article/35208>

[3]<http://flex.apache.org/community-3rdparty.html>

[4]<http://www.silexlabs.org/amfphp/>

[5]<http://www.silexlabs.org/amfphp-plugin-baguette-amf/>

[6] <http://php.net/manual/en/pdo.drivers.php>

[7][http://help.adobe.com/en\\_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf69084-7be0.html](http://help.adobe.com/en_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf69084-7be0.html)

[8]<https://github.com/PHPOffice/PHPEXcel/blob/develop/Documentation/PHPEXcel%20developer%20documentation.doc>

[9] Lovreković, Z., Lovreković, T. (2014.), Is HTML5 Right Technology for the Development of Business Software Applications? , TEM JOURNAL, Vol. 3, No. 1. pp.88-95