Digital Innovations
& Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 13rd December, 2016
Review Type: Blind
Review/Acceptance Information Sent : 11th Jan, 2017
Final Acceptance:: 31st Jan, 2017
DOI Prefix: 10.22624

Article Citation Format
S.A. Robinson & R.A. Ailenoator (2017) Model For Web Application Component Reuse (PHP).
Journal of Digital Innovations & Contemp Res. In Sc., & Eng Vol. 5, No. 1. Pp 1-14

Model for Web Application Component Reuse (PHP)

[1]S.A. Robinson & [2]R.A. Ailenoator
[1]Department of Computer Science, University of Uyo, Uyo, Nigeria
[2]Department of Computer Science, University of Benin, Benin City, Nigeria
[1]samuelrobinson@uniuyo.edu.ng; [2]reubenailenoator@gmail.com

ABSTRACT

A web application or web app is any application software that runs in a web browser and is created in a browser-supported programming language (such as the combination of JavaScript, HTML and CSS) and relies on a common web browser to render the application. With the use of server side scripting languages like Php, perl, ruby and python, web applications can interact with databases online or on a network and provide dynamic and robust programs similar to desktop applications. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility. We live in times of cut-throat competition, where the delivery time of a product may decide the success or failure of a product or business. In such a scenario, rapid application development has gained a lot of attention and importance, especially in the software development field, where change is said to be the only constant. Conventional software development life cycle processes do not make for flexible web application development thus making rapid application development preferable these days. In this paper, effort is made to explain web application components which are featured in various web applications, the existing sources of web application components which are Computer Aided Software Engineering (CASE) tools, Internet Forums and Web Hosts are also discussed bringing out the issues with these sources and then proposing a model for better web application component reuse which would involve upload of usable web application components, transformation of such components from usable to reusable components by a team of programmers thus maintaining quality and retaining programmer flexibility.

Keyword: Web Application Component Reuse, Hypertext Preprocessor

1. INTRODUCTION

A component is an encapsulated piece of code that performs some function for an application. In a more practical sense, a component is an object that contains code to manipulate data from one form to another, and which provides access to that code through a well-specified set of publicly available services (Bauer, 1993 and Homer, 2013).This function could be the processing of a business rule, like the computation of a sales tax, or it could be the retrieval of some information from a database for an application. The key characteristic of a component is that when it is created for use, the code for the component, as well as the information associated with the component, are packaged together(Gaedke and Rehse, 2000).

In this way, if there are multiple versions of the same component in use at one time, each one keeps its information separate from the others. There is no danger of information in one polluting the information of another. In addition to the type of work that it performs, a component is also defined by its interface. A Web site consists of a set of related Web pages grouped together by some means. Generally, a Web site is all of the pages that exist on a server, or within a folder on that server. For example, all of the pages that are on the http://www.examplesite.com server are considered part of that Web site. The correlation between the pages on a site is maintained by the links within each page on the site. The links on each page in the site will take users to other pages within the site. In this way, the pages that make up the site internally maintain the hierarchy of the site (Gaedke and Rehse, 2000).

These pages are still subject to the restriction of the Web server architecture, in that the server does not maintain information about the series of requests from a particular client. So while this related set of Web pages that make up a Web site are beginning to look more like an application, there are still some missing components. Web Applications are applications in the traditional sense, -in that they provide a service to the user of the application. They are different in the way that they are created as well as in the components that make them up. A traditional application requires a special set of files during development, but distributes different outputs.

2. REVIEWED OF RELATED WORKS

Web applications evolved from Web sites or Web systems. The first Web sites, created by Tim Berners-Lee while at CERN (the European Laboratory for Particle Physics), formed a distributed hypermedia system that enabled researchers to have access to documents and information published by fellow researchers, directly from their computers (Schwabe, 2001). Documents were accessed and viewed with a piece of software called a browser, a software application that runs on a client computer (Homer, 2013). With a browser, the user can request documents from other computers on the network and render those documents on the user's display. To view a document, the user must start the browser and enter the name of the document and the name of the host computer where it can be found. The browser sends a request for the document to the host computer. The request is handled by a software application called a Web server, an application usually run as a service, or daemon that monitors network activity on a special port, usually port 80. The browser sends a specially formatted request for a document (Web page) to the Web server through this network port. The Web server receives the request, locates the document on its local file system, and sends it back to the browser.

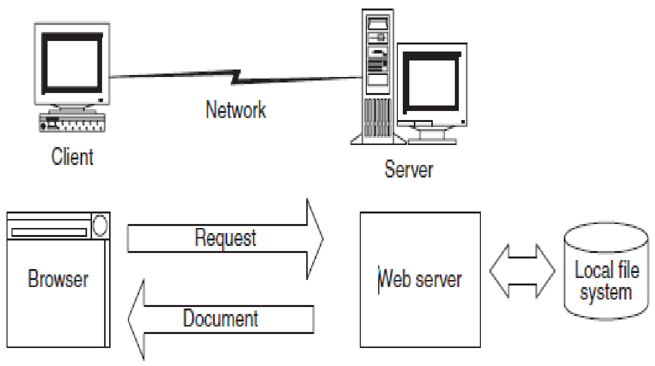

Figure 1: Web application architecture (source: www. investintech.com)

2.1 Web Applications Vs Web Sites

A standard website is generally content-centric (Bauer, 1993). That means it focuses on providing the web user with information, generally in a static layout with static links to other pages filled with static content. In a nutshell, there's not much to do aside from read each page. A web application means there is more to be done. It means some task can be accomplished, a goal can be attained or some expectation can be met. That is all rather esoteric so let's delve further into it with some sort of an example (Schwabe, 2001).

2.2 Hypertext Preprocessor PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers. Originally created by RasmusLerdorf in 1994, the reference implementation of PHP (the Zend Engine) is now produced by The PHP Group. While PHP originally stood for Personal Home Page,it now stands for PHP Hypertext Preprocessor, which is a recursive acronym (Homer, 2013).

PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable.

- PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP costs nothing, it is free to download and use. PHP is an amazing and popular language.
- It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)
- It is deep enough to run the largest social network (Facebook)
- It is also easy enough to be a beginner's first server side language!

2.3 Usability Vs Reusablity In Web Applications

Most web applications developed in php have usable components which are coded by the programmer to suit the specific function required (Tramontana, 2009). The components used which are present in other web applications are customized only. Usable codes in php are fully adapted to the web application being developed. Such usable codes would contain specific variable names. Consider the code segment below:

```
<?php
$sql = “select * from users where username = ‘$_POST[user]’ ”;
?>
```

Code segment 2.1 Usable Code Segment

The code above checks a database for a user using the username stored in the variable named $_POST[user]. Reusability would ensure that if this code is called, the programmer needs not know the details of the code before he uses it. A reusable code would then have generic names and extra lines of code to create dynamism. The reusable code format for the above code would thus be:

```
<?php
$username = $_POST['username']; //change to username textbox
$sql = "select * from users where username = '$username' ";
?>
```

Code segment

2.3.1 Reusable Code Segment

In the above code, an extra line of code was added to ensure that the variable is dynamic and comment was added to make it readable. The entire programmer needs to do when reusing this code is change the source of value of the variable to suit the new application.
It is worthy to note that reusability increases the lines of code and requires the use of more general terms as it aims for reuse.

2.4 Types of Web Application Component Reuse

In software engineering, there are three types of reuse: Application System Reuse which involves reusing an entire application by incorporation of one application inside another (COTS reuse)development of application families (e.g. MS Office); Component Reuse which involves the use of components (e.g. subsystems or single objects) of one application reused in another application and Function Reuse which means reusing software components that implement a single well-defined function (Griffiths, 2013).

2.5 Preparing Web Application Components for Reusability

In order to make components reusable, certain steps need to be taken. The first step would be to remove hard coding in components and make variables dynamic. Another requirement is the translation from procedural to object oriented programming. Security and robustness features should also be added to make the code meet the test of time in terms of quality and productivity. Documentation and testing are also important to make programmers understand the usage of the reusable components. The following sections discuss more on these preparation steps.

2.6 Procedural Vs Object Oriented Programming

PHP has support for both procedural and object oriented programming. This creates flexibility for programmers especially when scripts need to be mixed up with html codes but using procedural programming paradigm has been found to limit code reusability. Object oriented programming in php is best suited for building reusable components since it supports code re-use, portability, structure, meaning, control, separation of responsibility, time and reduces dependencies. It is worthy to note that object oriented programming does not affect the functionality of the produced web application but is focused more on the reusability of components (Hayes, 2008 and Quaint, 2014).

2.7 Making PHP Code Reusable

No one likes to write similar code over and over again and whenever one writes code one of the aims should be to avoid any repetition. Not only does it mean that writing more code could be avoided, but it'll help to make applications more efficient.PHP is well geared to enable you to do this by giving you the ability to create functions and classes, and there are a few simple things you can do to help you make your code more reusable.

2.8 Removing Hardcoding from Variable Names

In choosing variable names, many programmers make the repeated mistake of using hardcoded names. An example is in a developed a class which gets information from a social network account, and somewhere the programmer will need to send the social network an identifier for an account. Somewhere in the class will be a variable that defines that ID. Instead of having something like:

```
private $identifier = 'some_unique_identifier';
```

In order not to hardcode variables, it is preferable to use the code below:

```
private $identifier = null;
public function set_identifier($id = null)
{
if(!is_null($id))
   {
      $this->identifier = $id;
   }
}
```

2.9 Issues with Forums

One issue with forum is the lack of proper coordination of content. Code uploaded in forums often lack coordination and as such could be guess work or a try and error approach. Most codes uploaded are also mostly procedural giving a quick way out. The codes also lack security and robustness in most cases. A major issue is the lack of complete components or whole reusable systems since it is simply a forum for interaction among programmers. Internet forums have the following problems:

- Opens programmers up to hackers, spammers and predators. This could be of a very damaging effect to web applications developed with components gotten from such forums.
- Creating a successful forum is exhausting because the forum-master must present all the topics for discussion on a continuous basis for months, and the topics have to be interesting so that people start commenting. This means that the forum-master must also make sure that what is being posted is not spam and non-sense (all time-consuming aspects). Otherwise, all of the work and hard efforts will go to waste.
- Administrators can edit people's code, and they can ban/mute people if they put some ideas with which they don't feel comfortable with, even if the ideas may be useful and true to others. This would limit the amount of reusable code components that can be placed in such forums.
- People may give up posting due to the responses lacking body language, taking much longer to propose than verbal conversations, and being misleading and useless.

3. OUR APPROACH

Considering the problems with the existing methods discussed, we present a model that would help in making code reusability in PHP web applications more effective. This model is designed with a view to enable flexibility on the part of programmers while building web applications rapidly using PHP. This model would present programmers with reusable component on request which could be adjusted to suit development. In this model usable PHP web application components would be submitted to the internet site repository. The usable codes would then be translated from procedural to object oriented format if not already so. When programmers would request these reusable components, they are then given from the site and could make use of those components in their development. This model would allow for programmer flexibility as he has full access to the components unlike when using CASE tools which build the entire system following specified formats.

The programmer may choose to modify codes depending on need and would better create a dynamic system. This model would enhance standard in web applications and make them more secure and robust as these features would be added to usable components while transforming them to reusable components. Components of web application fully reusable could be channeled using this model to programmers requesting them.

3.1 Research Methodology

In carrying out this research, a case study was chosen - PHP which is a popular server side scripting language and also a general purpose programming language. Code segments, files and whole software systems built using PHP in areas of medicine, social networks, management systems and fuzzy logic systems were collected and examined with a view to identifying usable components. These usable components discovered such as login component, video select, file download, web cam view, database manipulation and sign-out components were converted from usable to reusable components by adding robustness, security, documentation, removing hard coded variables and statements and transforming procedural to object oriented programming paradigm which is supported by PHP. Observations were made during practical deployment of these reusable components and the findings were noted. CASE tools and forums were also examined with a view to finding flexibility and security properties as they also serve as sources of PHP components. A model was proposed to marry the patterns of transformation and the overall architecture of getting usable components and making them reusable and available to PHP programmers.

The summary of steps in using this centralized model are presented below:

- Identifying usable Components
- Upload usable components on centralized system
- Transform components from usable to reusable adding all required features
- Make component available for download

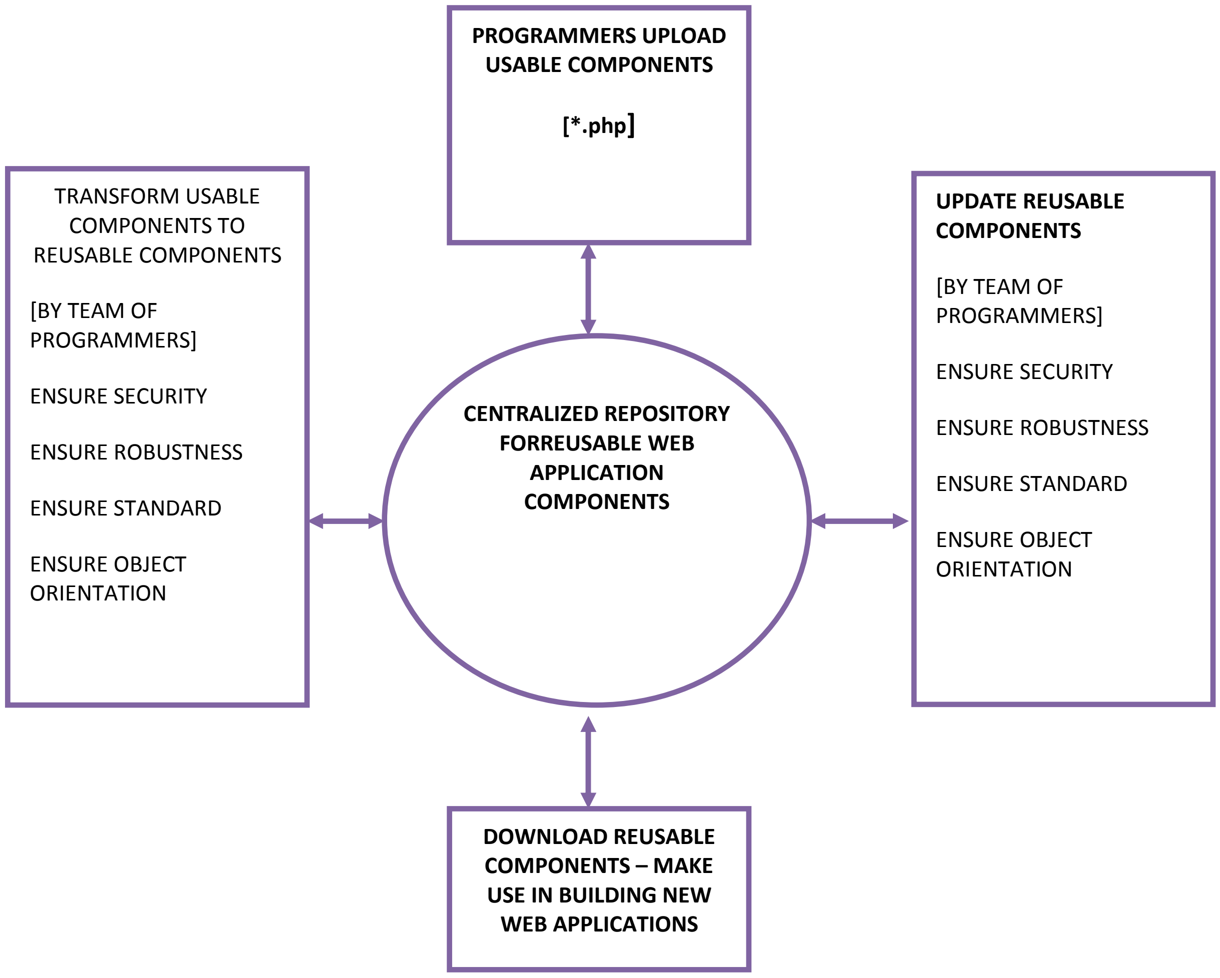
PROGRAMMERS UPLOAD USABLE COMPONENTS
[*.php]
TRANSFORM USABLE COMPONENTS TO REUSABLE COMPONENTS
[BY TEAM OF PROGRAMMERS]
ENSURE SECURITY
ENSURE ROBUSTNESS
ENSURE STANDARD
ENSURE OBJECT ORIENTATION
CENTRALIZED REPOSITORY FORREUSABLE WEB APPLICATION COMPONENTS
UPDATE REUSABLE COMPONENTS
[BY TEAM OF PROGRAMMERS]
ENSURE SECURITY
ENSURE ROBUSTNESS
ENSURE STANDARD
ENSURE OBJECT ORIENTATION
DOWNLOAD REUSABLE COMPONENTS – MAKE USE IN BUILDING NEW WEB APPLICATIONS

Figure 2; Centralized Repository for Web Application Component Reuse

3.2 Transformation from Usable to Reusable Code

The following sub sections show transformation from procedural to object oriented components using a few example components. Some of these transformation feature documentation, security, generalization and object oriented programming. Transformations are made with the aim of making the components reusable (Lim, 1994).

3.3 Transforming Login Component (Security and OOP)

The code sample below shows the typical usable php login :

```
$username = strtolower($_POST['username']);
$password = md5(strtolower($_POST['password']));
$qry = mysql_query("select * from users where username = ('$username') and password =
('$password') ") or die(mysql_error());
$no = mysql_num_rows($qry);
if ($no == 0 ){ $error = 'Wrong login details';s}
```

Code segment 4.2 Sample Login Procedural component

The transformed procedural code would look this way in object oriented format:

```
class users
{
public function login($username,$password)        {
            $username  = mysql_real_escape($username);
            $password  = mysql_real_escape($password);
            $qry = mysql_query("select * from users where username = ('$username') and password =
('$password') ");
            if (mysql_num_rows($qry) > 0) {return true;}
            else {return false;}
            }
}
//usage
$user = new users();
$login = $user->login($_POST['username'], $_POST['password']);
if ($login)
{
//next action
}
```

Code segment

3.3 Object Oriented Login component

From the code segments on login (procedural and object oriented), there are extra lines of code but the output is more structured. Programmers argue that codes are functional even though not structured and could be reused this way. This issue is further discussed in section

Transforming Data Base Manipulation Code (Security and OOP)

The code segments below show the common components used in inserting, updating and deleting data in database table. It is entirely usable and meets the functionality demanded in the web application.

```
<?php
$result=mysql_query("insert into users values
('$_POST['name']','$_POST['email']',NULL)");
?>
```

Code segment 4.4 Database Insert Component

```
<?php
$id = $_POST['id'];
$result=mysql_query("update users set username = '$_POST['name']', email = '$_POST['email']
where id = $id ");
?>
```

Code segment 4.5 Database Update Component

```
<?php
$id = $_POST['id'];
$result=mysql_query("delete from users where id = $id ");
?>
```

Code segment 4.6 Database Delete Component

```
<?php
$id = $_POST['id'];
$result=mysql_query("select *  from users where (regno  LIKE '%$search%')");
?>
```

Code segment 4.6 Database Search Component

```
<?php
$con= mysql_connect('localhost','root',password');
$db = mysql_select_db('school',$con);
?>
```

Code segment 4.6 Database Connect Component

Transforming these components to object oriented format, we have:

```
<?php
Class Database
{
        Public function connect($db,$host,$user,$pass)
        {
        try {
        $mysql = new PDO ("mysql:dbname=$db;host=$host,$user,$pass");
        } catch (PDOException $e)
        { return 'Connection failed: '. $e->getMessage(); }
        }

        Public function insert($table,$colums,$values){
        $qry = mysql_query("insert into $table values ($values)") ;
        }

        Public function delete($table,$id,$value){
        $qry = mysql_query("delete from $table where $id = $value") ;
        }

        Public function update($table,$id,$idval,$colums,$values){
        $columns = split(',',$columns);
        $values = split(',',$values);
        $sizeCol = sizeof($columns);
        $items = '';
        For ($i=0;$i<$sizeCol;$i++)
        {$items .= $columns[$i].'='.$values[$i];}
        $qry = mysql_query("update $table set $items,  where $id = $idval") ;
        }
}
//example usage
// $database = new Database();
// $database-> connect('dbname','myhost','myuser','mypass');
// $database-> insert('users','username,password','test1,pass1');
// $database-> update('users','username,password','test1,pass1');
// $database-> delete('users' ,1);

?>
```

Transforming SMS Sending Code (OOP)
Sending SMS from web applications designed using php can be achieved using the code segment below. The SMS services provider is xwireless.net.

```
$thestring=
'http://SMSc.xwireless.net/API/WebSMS/Http/v1.0a/index.php?username=marstext.com&password
=Switbai.2014_rome&sender=Switbai&to='.$tempmobile.'&message='.$msg;
$url = $thestring;
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$contents = curl_exec ($ch);s
curl_close ($ch);
//////end of sending SMS
```

Code segment Object Oriented component for sending SMS

Transforming Mailing Code (Documentation and Robustness)
The php mailing component commonly used in contact forms and automated mail responses is shown below:

```
// send mail
mail("webmaster@example.com",$subject,$message,"From: $from\n");
```

This usable code lacks robustness and reusability. The code segment below shows the more robust form:

```
<?php
require 'PHPMailerAutoload.php';

$mail = new PHPMailer;

$mail->isSMTP();                                      // Set mailer to use SMTP
$mail->Host = 'smtp1.example.com;smtp2.example.com';  // Specify main and backup SMTP servers
$mail->SMTPAuth = true;                               // Enable SMTP authentication
$mail->Username = 'user@example.com';                 // SMTP username
$mail->Password = 'secret';                           // SMTP password
$mail->SMTPSecure = 'tls';                            // Enable encryption, 'ssl' also accepted

$mail->From = 'from@example.com';
$mail->FromName = 'Mailer';
$mail->addAddress('joe@example.net', 'Joe User');     // Add a recipient
$mail->addAddress('ellen@example.com');               // Name is optional
$mail->addReplyTo('info@example.com', 'Information');
$mail->addCC('cc@example.com');
$mail->addBCC('bcc@example.com');

$mail->WordWrap = 50;                                 // Set word wrap to 50 characters
$mail->addAttachment('/var/tmp/file.tar.gz');         // Add attachments
$mail->addAttachment('/tmp/image.jpg', 'new.jpg');    // Optional name
$mail->isHTML(true);                                  // Set email format to HTML

$mail->Subject = 'Here is the subject';
$mail->Body    = 'This is the HTML message body <b>in bold!</b>';
$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

if(!$mail->send()) {
echo 'Message could not be sent.';
echo 'Mailer Error: ' . $mail->ErrorInfo;
} else {
   echo 'Message has been sent';
}
```

From the code sample, it is seen that transforming the component from usable to reusable component makes for more lines of code and documentation. This would make future use easier and give the programmer more flexibility and features to manipulate.

Transforming Sign Out Code (Generalization)
The code segment below is used in carrying out a signout operation in web applications using php.

```
<?php
session_start();
session_destroy();
header ('Location: index.php');
?>
```

Transforming this code would aim at making it more general thus:

```
<?php
session_start();
session_destroy();
$destination = $_GET['destination];
header ('Location: $destination);
?>
/* usage
Signout.php?destination=destination.php
*/
```

From the code segment, effort was made to make the code reusable by making the destination generic after sign out has being completed. This same logic could be applied to the login form.

Adding Enhancement Features
Kelleher (2010) showed that in order for the web application components to meet the test of time, features such as security, robustness and standard must be inputted into the reusable components. Security standards in connecting to databases involve the use of PDO (php data objects) instead of the normal procedural unsecure connection methods adopted by programmers. The code segment below shows a reusable PDO database connection component. It could be concluded that transformation from usable to reusable components require effort to make the component robust, secure and well documented.

```
<?php
try {
$mysql = new PDO ('mysql:dbname=example;host=localhost','root','password');
} catch (PDOException $e)
 {echo 'Connection failed: '. $e->getMessage();}
//usage
$user = new User('testuser','password',$mysql);
?>
```

From the code segment, such security feature could be added during the transformation from procedural to object oriented programming. Transformation to reusable components should also include some form of robustness and generic variable usage (Tramontana, 2009).

4. CONCLUSION

Web application component reuse means making use of usable web application components by transforming such components from usable to reusable components. Such transformations would involve removing hard coding, making procedural components object oriented, adding documentation and carrying out testing.

Although there are CASE tools and forums for obtaining components, a centralized system for component reuse, which would solve the problem of programmer flexibility while retaining standard and quality of applications developed, was proposed. This model would allow for uploads of usable components which are transformed by a team of programmers to reusable components and made available for future reuse. In conclusion, web applications have come to stay and in the near future, we would need only computers with web browsers and an access to the internet in order to make use of applications formerly installed on the computers operating systems. In order to enhance the development of such web applications, component reuse through an efficient centralized platform would ensure the production of quality web applications while retaining programmer flexibility.

REFERENCES

1. 1 Bauer, D. (1993). "A reusable parts center". IBM Systems Journal, 32(4):620-624. C++ Report magazine.
2. Gaedke, M.& Rehse, J. (2000). "Supporting Compositional Reuse in Component-Based Web Engineering". In SAC '00: Proceedings of the 2000 ACM symposium on Applied computing, pages 927-933, New York, NY, USA. ACM Press.
3. Ganesan, D. & J. Knodel, (2010). "Identifying Domain-Specific Reusable Components from Existing OO Systems to Support Product line Migration". Proceedings of the First International Workshop on Reengineering Towards Product Lines, 27-36.
4. Gill, N. (2012)."Reusability Issues in Component-Based Development".SigsoftSoftw. Eng. Notes, Vol. 28, pp. 4-4.
5. Griffiths, B. (2013). "How to make PHP code reusable".Retrievedfromhttp://www.appliedorder.com/how-to-make-your-php-code-reusable/
6. Hayes, L. (2008). "Reusability vs. Usability: Where to Draw the Line?"
7. Homer, A. (2013). "Components and Web Application Architecture", http://quainttech.blogspot.com/2007/03/why-you-want-object-oriented.html
8. http://www.stickyminds.com/article/reusability-vs-usability-where-draw-line IEEE Software, 11(5):23(8). IEEE Software, 11(6):17(9).pp 11-16.
9. Jones, C. (1993). "Software return on investment preliminary analysis."
10. Kelleher, C. (2010), "Toward Transforming Freely Available Source Code into
11. Lim W. (1994).Effects of reuse on quality, productivity, and economics.
12. Maxim, B. (2011). "Software Reuse and Component-Based Software Engineering".In: IEEE International Conference on Information Reuse and Integration, IRI 2007,August 13-15, 2007, pp. 323–328.
13. Quaint, T. (2014) "Why you want Object Oriented Programming in PHP". Retrieved from Retrieved from http://technet.microsoft.com/en-us/library/bb727121.aspx
14. Rouse, M. (2011). "CASE (computer-aided software engineering)".
15. Schmidt, D. (1999). "Why Software Reuse has failed and How to Make It Work for You"
16. Schwabe, D. (2001). "Engineering Web Applications for Reuse".JournalIEEE MultiMedia archiveVolume 8 Issue 1, Page 20-31. Software Productivity Research, Inc.
17. Tramontana, P. (2009). "Identifying reusable components in web applications".Proceedings of the IASTED International Conference on Software Engineering, SE, pp 526-531.