

University of East London Institutional Repository: <http://roar.uel.ac.uk>

This is an unpublished conference paper. Please scroll down to view the document itself. Please refer to the repository record for this item and our policy information available from the repository home page for further information.

Author(s): Draganova, Chrisina.

Article title: Web Services and PHP

Year of publication: 2004

Citation: Draganova, C. (2004) 'Web Services and PHP', In: *Proceedings of the Conference on Information Technology in Education (ITE'04)* Elizabethtown College, Elizabethtown, PA, September 2004, pp. 116 - 121.

Web Services and PHP

Elizabethtown College 2004

Chrisina Draganova

*Department of Computing, Communication Technology and Mathematics
London Metropolitan University
100 Minories, London EC3 1JY
c.draganova@londonmet.ac.uk*

ABSTRACT

Web Services are based on open standards and provide ways of communication between computer applications written in different programming languages and implemented on different platforms. Web Services technologies provide cost effective and comparatively easy ways of integrating heterogeneous software components and improving communications between different systems. Web Services are a major step forward in the computing field and, as such, they need to be addressed in the computing curriculum. In teaching computing we have to emphasize principles, as well as demonstrate them through specific technologies. Teaching Web Services is an excellent way to give the students knowledge about leading technologies and at the same time demonstrate fundamental computing concepts. This paper gives an overview of the main Web Services standards and the latest PHP implementations developed in this respect. It also considers how the Web Services topics are embedded in the computing curriculum at the Department of Computing, Communication Technology and Mathematics at London Metropolitan University.

1. INTRODUCTION

Web services are a new emerging set of technologies, which aim to enable the communication between heterogeneous software applications. Web services are based on open standards and many major organizations and software vendors are involved in the development of these standards including: W3C (World Wide Web Consortium), OASIS (Organisation for Advancement of Structured Information Standards, Web Services Interoperability Organisation, Sun Java Microsystems Inc, Microsoft, IBM, and Oracle.

Web Services technologies provide an easy way to expose the functionality of existing software components without the need to redevelop them. This provides cost effective and comparatively easy ways of integrating heterogeneous components and improving communications

between different systems. Web Services are a major step forward in the computing field and, as such, they need to be addressed in the computing curriculum.

This paper aims to share some ideas of how Web services topics can be embedded in the computing curriculum without the need of major redesign of existing modules or of introducing new ones. The paper is organized in six sections including the introduction. The next two sections give an overview of the main Web Services standards and the latest PHP implementations developed in this respect. The fourth section discusses how Web Services topics are taught at London Metropolitan University. The fifth section demonstrates some of the examples of PHP Web services used in the lab sessions for one of the Internet modules and the last section gives the conclusions.

2. OVERVIEW OF WEB SERVICES STANDARDS

The official definition of Web Services given by W3C organisation states that [22]:

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

In other words, Web services are software applications that may be part of a distributed system, which communicate with other applications by exchanging XML messages. XML (Exensible Markup Language) is the foundation for all Web Services standards. XML enables the description and exchange data in a format that is not specific to any particular platform. XML documents are easily parsed by computer programs and understood by humans. Sharing data in an XML format allows software applications from different vendors and platforms to communicate with each other, thus solving the interoperability issue of

the previous distributed technologies and enabling the construction of loosely coupled systems.

The key Web Services standards include: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration), which are all XML-based. SOAP provides the messaging mechanisms, WSDL gives a model for describing Web services, and UDDI enables publishing, searching and discovering Web services.

SOAP messages are the most common way of communication with Web services. SOAP provides: formal conventions for message construction such as envelope, header, body, fault part, attachment part; encoding rules for data types; and mechanisms for remote procedure calls and responses. SOAP messages can be exchanged using different Internet protocols as HTTP, SMTP, and FTP. SOAP supports Remote Procedure Call (RPC) and the document-oriented exchange of information. The SOAP messages can be transmitted synchronously or asynchronously depending on the requirements of the specific application. The current specifications of these standards could be found on the W3C web site [14].

A WSDL description is an XML document that provides information about what exactly a web service does, how to invoke it and where to find it. It includes details such as service operations, binding information, location, protocols and message formats. The current specification of WSDL standard is available on [23].

UDDI is a registry standard, which allows organisations to publish and discover Web services using standardised methods. UDDI is often compared to white, yellow and green pages, because they contain contact, categorisation and technical information. The UDDI specification [18] defines a set of data structures such as business entities, business services, specifications, service types and API operations for registering and finding Web services.

3. OVERVIEW OF PHP WEB SERVICES IMPLEMENTATIONS

PHP, short for PHP Hypertext Preprocessor, is a scripting programming language, designed for Web development. It is used in combination with Apache's Web server, which runs almost 70% of sites on the Web [21]. PHP is an open source product. The latest version is available free for download from [11]. PHP comprises a number of features that enable Web services development such as object oriented support, XML, DOM, XPath, XLink, SAX and XSLT support [5]. There are several PHP implementations that enable the development of Web Services including NuSOAP, XML-RPS, PHP-SOAP, PEAR and REST.

NuSOAP is a collection of PHP classes combined in a single file "nusoap.php", which provides implementation for SOAP and WSDL standards. It allows creating and consuming Web services without the need of additional PHP extensions. Download documentation and examples could be found in [9], [10].

XML-RPC is a Remote Procedure Calling protocol that works over the Internet. The body of an XML-RPC message is an XML document sent over a HTTP-POST request to a server, where a procedure is executed and the return value is wrapped in an XML message and sent back. XML-RPC is simpler than SOAP. The PHP implementation consists of a collection of PHP classes, which provide a framework for writing XML-RPC clients and servers in PHP. The classes that provide client and server XML-RPC functionality are included in the files "xmlrpc.inc" and "sxmlrpcs.inc", respectively. Download and full documentation can be found in [19], [20].

PHP-SOAP is an extension of PHP that implements a large subset of SOAP and WSDL specifications [15]. The code is developed in C with the aim of increasing speed. Full documentation for the PHP-SOAP extension is available on [12].

PEAR (PHP Extension and Application Repository) is a framework for building PHP applications, which includes Web services capabilities [13]. PEAR includes a "Server.php" library for the creation of PHP Web services using SOAP, a "Disco.php" library for generation of WSDL files necessary for exposing the services to clients, and a "Client.php" library for developing Web services clients [16].

REST (Representational State Transfer) is based on a different approach than XML-RPC or SOAP. It uses the standard HTTP methods such as GET, POST and PUT to send and retrieve XML data [8]. The cURL (Client URL Request Library) PHP extension combined with some PHP utilities for parsing XML documents are required to create Web services and clients with REST [17]. The drawback of this approach is that there are not so many available REST toolkits and frameworks, and developers need to write more code themselves.

4. TEACHING WEB SERVICES

The computing field is one of the most dynamic. Responding timely to new emerging technologies and changes is an important task of computing departments around the world. This is not easy, as we are aware of the issues that arise when introducing new material. Issues that need to be addressed include:

- maintaining the academic quality
- maintaining the cohesiveness of the curriculum

- providing the relevant resources

An important dilemma considered in [4] is whether we should wait until a new technology matures or whether we should introduce it at an earlier stage in the computing programs. In the first case students would be disadvantaged to some extent, as they will be lacking leading edge skills. In the second case we may find that a technology becomes quickly obsolete, and adopting it would be a waste of time and resources.

Web services technologies are already in a stage where the basic standards are set. Major organizations and vendors are working on different specifications and implementations. The fundamental technology XML on which Web services are based is adopted almost everywhere in the industry. Thus introducing Web services technologies in the computing programs would provide valuable opportunities for students to [1]:

- reinforce the learning of fundamental programming concepts
- understand better the present Internet technologies
- understand interoperability issues in distributed computing
- understand business issues related to Web Services
- understand the importance of using open standards
- acquire knowledge of leading technologies
- gain marketable skills

At present Web services topics are introduced only in existing modules at London Metropolitan University. These modules are typically modules, which are related to Web development, Internet technologies, Networking and Distributed systems and are taught in the second or third year of study for various degrees in computing. XML topics are already included in several modules and this has made easier the adoption of Web services related material. The students are familiarized with the main Web services standards and a specific implementation. They are expected to gain understanding of these technologies and develop skills in implementing Web services using a specific platform. One of the challenges is choosing a particular Web services implementation for the practical lab sessions. Some major Web services platforms available at present are Sun Microsystems Inc. Java™ Web Services Developer Pack, Microsoft .NET, IBM WebSphere SDK for Web Services, Oracle9i support for Web Services, Mindelectric GLUE framework for Web Services and PHP implementations for Web Services. The main considerations taken in account in selecting the most suitable Web services implementation are: student's programming language skills, existing software support, ease of using and learning the respective implementation. A natural choice for the modules based on Java programming language is a

platform that is based on Java. In the last two years the Oracle9i and GLUE Web services implementations have been experimented with in these modules.

A PHP NuSOAP implementation of Web services capabilities is used in the "Internet technologies" second year module for "Foundation Degree of Computing". The selection of the PHP scripting language for this web development module is based on the nature of the foundation degree study. The "Foundation Degree of Computing" is a new vocationally oriented degree launched in the UK in 2001, which aims to develop work-specific skills, relevant to a particular sector of industry [3]. The students who finish the foundation degree study should be able to make immediate contributions in the workplace. PHP comes as a natural choice in this case, as a widely used web development language, which is also supported by our university.

Teaching topics on Web services has been perceived well by the students and has given them exposure to new emerging technologies and deeper understanding of fundamental programming concepts.

5. PHP WEB SERVICES EXAMPLES

This section includes some of the Web services related examples used in the lab sessions for the "Internet Technologies" second year module for "Foundation Degree in Computing". These examples are programmed using a PHP NuSOAP implementation of Web services. PHP NuSOAP does not require any special PHP extension or any special installation. PHP NuSOAP is a set of classes contained in a single file "nusoap.php" which has to be placed in a directory accessible by the PHP files. The first two examples based on [8] show how to build simple Web services provider and client respectively. The third example demonstrates how to develop a client to consume existing Web services using Google Web APIs [6], [7]. During the lab sessions the students are asked to try these examples and to modify some parts of them in order to add more functionality. By doing these examples, which are directly related to the Web services lecture session, students develop better understanding of the technologies and gain practical skills as well.

5.1 Developing a server

Developing a server involves several basic steps:

- include the NuSOAP library
`require_once('nusoap.php');`
- create a new server instance
`$server = new soap_server();`
- define a name space for the service: `$name_space = "http://meglos.lgu.ac.uk/~draganov/qf204/examples/week10/";`

- configure the WSDL description and name for the service


```
$server->configureWSDL('Calculator', $name_space);
$server->wsdl->schemaTargetNamespace = $name_space;
```
- register a method to be exposed for remote invocation with the relevant parameters, return values and name-space


```
$server->register('multiply', $param, $return, $name_space );
```
- define the method as PHP function


```
function multiply($x,$y)
```
- invoke the service


```
$server->service($HTTP_RAW_POST_DATA);
```
- save the code as “calc_server.php” and place it on the web server

The entire code for the example “calc_server.php” is:

```
<?php //calc_server.php
require_once('nusoap.php');
$server = new soap_server();
$name_space = "http://meglos.lgu.ac.uk/~draganov/qf204/examples/week10/";
$server->configureWSDL('Calculator', $name_space);
$server->wsdl->schemaTargetNamespace = $name_space;
$param = array('x' => 'xsd:string', 'y' => 'xsd:string');
$return = array('return' => 'xsd:string');
$server->register('multiply', $param, $return, $name_space );
function multiply($x,$y) {
    $z = $x*$y;
    return ' '. $x .' * ' . $y .' = ' . $z;
}
$server->service($HTTP_RAW_POST_DATA);
?>
```

The WSDL description of the service can be viewed on:
http://meglos.lgu.ac.uk/~draganov/qf204/examples/week10/calc_server.php?wsdl.

5.2 Developing a client

Developing a client to consume the web service that was created in 5.1 requires the following steps:

- include the NuSOAP library


```
require_once('nusoap.php');
```
- instantiate the NuSOAP client object and pass the WSDL web service description to the client


```
$client = new soapclient($wsdl, 'wsdl');
```
- create an array of parameters to pass to the Web service


```
$param = array('x' => 3.6, 'y' => 4.5);
```
- call the remote method


```
$result = $client->call('multiply', $param);
```

The entire code for the client is:

```
<?php //calc_client.php
```

```
require_once('nusoap.php');
$wsdl = "http://meglos.lgu.ac.uk/~draganov/qf204/examples/week10/calc_server.php?wsdl";
$client = new soapclient($wsdl, 'wsdl');
$param = array('x' => 3.6, 'y' => 4.5);
$result = $client->call('multiply', $param);
echo $result;
?>
```

5.3 Developing a client using “Google” Web services APIs

The example in this section demonstrates how to develop a search engine based on the Google Web APIs [6] using PHP and NuSOAP. This example requires similar steps to those in 5.2. The remote method `doGoogleSearch` takes a number of parameters set in the array variable `$param`. A full description of the search parameters and format of results of the method `doGoogleSearch` is available in [7]. The WSDL description of the Google Web services, given at <http://api.google.com/GoogleSearch.wsdl> is passed to the `call` method of the SAOP client. The result of the call is assigned to the variable `$results` which is an associative array. The element `$results['estimatedTotalResultsCount']` corresponds to the estimated total number of results that exist for the query. The element `$results['resultElements']` is an array of `resultElement` items which corresponds to the actual list of search results. The function `display` introduced for convenience, renders a single search result passed through the array parameter `$result`. Using the associative array `$result` the URL, title and the snippet are displayed by `$result['URL']` (the URL of the search result, returned as text, with an absolute URL path), `$result['title']` (the title of the search result) and `$result['snippet']` (the text excerpt from the results page that shows the query in context as it appears on the matching results page), respectively. The full code for this example is:

```
<?php //search_google.php
//include the NuSOAP library
require_once('nusoap.php');

// set parameters
$params = array(
    'key'=>'licence key', //Google licence key
    'q' => 'Web services', //search query
    'start' => '0', // start from result 0
    'maxResults' => '10', //display max 10 results
    'filter' => 'false', //filter similar results
    'restrict' => '', // Restricts to topic 'safe-Search' => 'false', // adult content filter
    'lr' => '', // Language Restrict
    'ie' => '', // Input Encoding - deprecated
    'oe' => '' // Output Encoding - deprecated
);

// create a new SOAP client
```

```

$soapclient = new soapcli-
ent('http://api.google.com/GoogleSearch.wsdl','ws
dl');

// call the service method
$results = $soapcli-
ent->call('doGoogleSearch',$params);

// display results
if ( is_array($results['resultElements']) ) {
    print '<p>Your Google query for <b>Web
services</b> found '
    . $results['estimatedTotalResultsCount'] . "
results, the top ten of which are:</p>";
    foreach ( $results['resultElements'] as
$result ) {
        disply_result($result)
    }
}
else { //no results
    print "No results from this query";
}

function display($result){
    "<p><a href='" . $result['URL'] . "'> " .
    ($result['title'] ? $result['title'] :
    'no title' ) .
    "</a><br>" . $result['URL'] . "<br>" .
    ($result['snippet'] ? $result['snippet'] :
    'no snippet' ) .
    "</p>";
}
?>

```

Figure 1 shows the results after executing the script search_google.php.

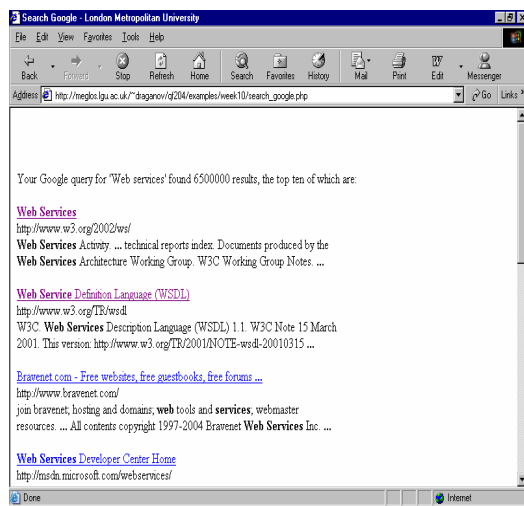


FIGURE 1. Google search results using search_google.php

6. CONCLUSIONS

One of the main principles in designing a computing curriculum emphasized in [2] states that there must be an ongoing review process of the curriculum as a response to the rapid evolution of computer science and the constant changes in technology. Including Web services in the

computing curriculum as a new major development in the computing field demonstrates this principle in practice. Teaching Web Services topics give students exposure to leading edge technologies and reinforce the understanding of fundamental concepts in computing such as data models, encapsulation, remote procedure calls, APIs, network protocols, distributed systems and security.

REFERENCES

- [1] Draganova, C. "Web services and Java", *JICC 7 Proceedings*, London Metropolitan University, 2003
- [2] Final Report of the Joint ACM/IEEE-CS Task Force on Computing Curricula 2001 for Computer Science <http://www.computer.org/education/cc2001/final/index.htm>
- [3] Foundation Degrees in the UK <http://www.foundationdegree.org.uk/>
- [4] French, T. and D. Stephens, "Teaching Java, XML and e-Commerce: Some reflections of past and future approaches from the 'front-line' at Luton University 1999-2002", *JICC 7 Proceedings*, London Metropolitan University, 2003
- [5] Fuller, J., Egervari, K. and H. Fuecks, et al, *Professional PHP Web Services*
- [6] Google Web APIs <http://www.google.com/apis/index.html>
- [7] Google Web APIs Reference <http://www.google.com/apis/reference.html#searchrequest>
- [8] Jean-Luc, D., *Creating and Consuming Web services with PHP*, 2004, <http://www.xml.com/pub/a/ws/2004/03/24/phpws.html>
- [9] NuSOAP Download <http://dietrich.ganx4.com/nusoap/>
- [10] NuSOAP API documentation <http://dietrich.ganx4.com/nusoap/APIDoc/>
- [11] PHP <http://www.php.net>
- [12] PHP-SOAP Documentation <http://www.php.net/manual/en/ref.soap.php>
- [13] PEAR <http://pear.php.net/>
- [14] SOAP specification <http://www.w3.org/TR/SOAP/>
- [15] Stogov, D., *PHP SOAP Extension*, 2004, <http://www.zend.com/php5/articles/php5-SOAP.php>
- [16] Stone L, *Building PHP Web Services with PEAR*, 2004, <http://www.devchannel.org/article.pl?sid=04/02/11/1432220>
- [17] Trachtenberg A., *PHP Web Services Without SOAP*, 2003, http://www.onlamp.com/pub/a/php/2003/10/30/amazon_rest.html
- [18] UDDI organization web site, <http://www.uddi.org>
- [19] XML-RPC Documentation <http://xmlrpc.usefulinc.com/doc/support.html>
- [20] XML-RPC Download <http://xmlrpc.usefulinc.com/php.html>

[21] Welling, L. and L. Thomson, *PHP and MySQL Web Development*, 2/e, Sams Publishing, 2001

[22] Web Services Architecture <http://www.w3.org/TR/ws-arch/>

[23] WSDL Specifications <http://www.w3.org/TR/wsdl>