

Web Services Design and Implementation through C# .NET

Asst. lecturer. Mais Abd Khalil Fdheel

AL- Mustansiriyah University, College of Engineering, Computer & Software Eng. Dep.

Manarabd2@yahoo.com

Asst. lecturer. Izz Kadhum Abboud

AL- Mustansiriyah University, College of Engineering, Computer & Software Eng. Dep.

Izz1962@yahoo.com

Asst. lecturer. Abbas Saleh Hassan

AL- Mustansiriyah University, College of Engineering, Computer & Software Eng. Dep.

Abbasaltee10@yahoo.com

Abstract

The purpose of this paper is to give a practical introduction to web services and describe their increased importance in information technology world, also Explain everything about Web Services in .Net, web services benefits, explain what web service mean and how it differ from websites technology, also give practical examples to illustrate how to design and implement web services using .NET and C#, and how to lead auser to reach and consume (use) web service in network distributed systems and internet.

Keywords: XML, WSDL, HTTP, Disco, Proxy, RSS, ASP, C#.

تصميم وتنفيذ خدمات الوب من خلال استخدام لغة سي هاش دوت نت

م.م ميس عبد خليل م.م عز كاظم عبود م.م عباس صالح حسن
الجامعة المستنصرية / كلية الهندسة / قسم هندسة الحاسبات والبرمجيات

الخلاصة

هذا البحث يهدف إلى إعطاء مقدمة عملية إلى خدمات الوب وبيان أهميتها المتزايدة في عالم تكنولوجيا المعلومات، كذلك توضح كل شيء يتعلق بخدمات الوب في دوت نت فريم وورك، فوائد خدمات الوب، يوضح ماذا نعني بمصطلح خدمة الوب وكيف تختلف خدمة الوب عن تقنية مواقع الوب، كذلك إعطاء أمثلة عملية لتوضيح كيفية تصميم وتنفيذ خدمات الوب باستخدام دوت نت ولغة سي هاش، وكيفية جعل المستخدم يصل إلى هذه الخدمات والاستفادة منها سواء في أنظمة الشبكات الموزعة أو الوصول إليها والاستفادة منها عبر الانترنت.

1. Introduction

Most of the people does not understand that why we need a web service and make a wrong use of it. The main idea of a web service is to join two businesses together since they cannot join due to their graphical locations. Web Service is also used to link different systems

together. The best thing about this is that the systems can be in different in nature, meaning a web service enables a windows application to interact and communicate with the linux application, this is done by using [Extensible Markup Language](#)(XML) as the transferring medium. Since XML is understood by all the systems because it's

nothing but plain text that's why this is a perfect language to join systems together. XML web Service should never be used to transfer confidential data, not to be used in real time programming where time is the essence^[1].

Web services, which promote software reusability in distributed systems where applications execute across multiple computers on a network. A Web service is a class that allows its methods to be called by methods on other machines via common data formats and protocols, such as XML and HTTP. In .NET, the overall the network method calls are commonly implemented through the Simple Object Access Protocol (SOAP), an XML-based protocol describing how to mark up requests and responses so that they can be transferred via protocols such as HTTP. As increasing numbers of organizations worldwide have connected to the Internet, the concept of applications that call methods across a network has become more practical. Web services represent the next step in object-oriented programming-rather than developing software from a small number of class libraries provided at one location, programmers can access Web service class libraries distributed worldwide. A Web service can be defined as an interface to application functionality that is accessible using well-known Internet standards and is independent of any operating system or programming language ^{[2][3]}.

The standardization and the flexibility introduced by web services in the development of new applications translate into increased productivity and gained efficiency^[4].Web services can be viewed as remote objects, suitable for on-demand usage, which have to be published by a service provider in order to be subsequently discovered, selected, bound and used (web service functions invoked) by clients^[5].

Web services are loosely coupled, self-contained software modules that can be accessed programmatically using existing Internet technology, found and assembled dynamically to serve particular function, solve a specific problem, or deliver a particular solution to a customer^[6]. Examples of Web Service are mentioned below:

- **Weather Reporting:** You can use Weather Reporting web service to display weather information in your personal website.
- **Stock Quote:** You can display latest update of Share market with Stock Quote on your web site.
- **News Headline:** You can display latest news update by using news headline web Service in your website^[11].

1.1 Deference's between web service and website

Here are the main differences between a Web Service and a Web Site: Web Site has an interface Web Service has no interface Web Site is designed to interact with people Web Service is designed to interact with other applications Web Site is designed to work with web browser clients Web Service is designed to work with any type of client or device as you can see, a Web Service and a Web Application have almost the same role, they just go about fulfilling that role in vastly different ways . A Web Service uses HTTP (Hypertext Transfer Protocol) and SOAP (Simple Object Access Protocol) to transfer data to and from the clients. Data sent to and from the Web Service is rendered into XML so the Web Service can read it, then it is sent back to the client, which then renders the returned XML into the format it was expecting^[10].

1.2 Standards body

The World Wide Web Consortium (W3C) began the standardization of Web standards such HTTP, HTML, and XML. Actually the W3C is home to SOAP, WSDL, WS-Choreography, WS-Addressing, WS-Policy, XML Encryption, and XML Signature. The Organization for the Advancement of Structured Information Standards (OASIS) originally started to promote interoperability across Structured Generic Markup Language implementations. Actually OASIS is home to UDDI, WS-Security, WS-BPEL, WS-Composite Application Framework, WS-Notification, WS-Reliability, Web Services Policy Language, Web Services Distributed Management, and Web Services Resource Framework. Web Services Interoperability (WS-I) is a standards body which promotes Web Services interoperability across platforms, operating systems, and programming languages. WS-I produces specifications called profiles that provide a common interpretation of other specifications and provides testing tools to help Web services vendors ensure conformance to WS-I specifications. A profile is a set of guidelines for the use of Web services specifications beyond the core protocols. These guidelines are important for creating interoperable Web Services. Interoperability Profiles also resolve ambiguities in areas where the Web services specifications are not clear enough to ensure that all implementations process SOAP messages in the same way.

1.3 The Maturity of Web Services Specifications

The most common used standards in the Web Services world are:

1. XML Schema: For message data typing and structuring. It allows defining a common vocabulary that the sending and receiving party may understand for achieving the message interchange goal.
2. WSDL: For associating messages and message exchange patterns (logic interface) with service names and network addresses (endpoints acting as physical interface).

3. WS-Addressing: For including endpoint addressing and reference properties associated with endpoints. Many of the other extended specifications require WS-Addressing support for defining endpoints and reference properties in communication patterns.
4. WS-Policy: For associating quality of service requirements with a WSDL definition. WS-Policy is a framework that includes policy declarations for various aspects of security, transactions, and reliability.
5. WS-Security: For providing message integrity, authentication and confidentiality, security token exchange, message session security, security policy expression, and security for a federation of services within a system.
6. WS-Metadata Exchange: For querying and discovering metadata associated with a Web service, including the ability to fetch a WSDL file and associated WS-Policy definitions.

The former extensions are pretty intuitive to design and implement, because Web Services basic specifications (SOAP and WSDL) have an inherently support composition of new features, meaning existing applications can be extended instead of being changed.

When invoking a service, it's important to understand not only the data types and structures to send but also the additional qualities of service provided (if any), such as security, reliability, or transactions. If one or more of these features are missing from the message, it may prevent successful message processing^[7].

1.4 The Reasons for using Web Services

After buying something over the Internet, you may have wondered about the delivery status. Calling the delivery company consumes your time, and it's also not a value-added activity for the delivery company. To eliminate this scenario the delivery company needs to expose the delivery information without compromising its security. Enterprise security architecture can be very sophisticated. What if we can just use port 80 (the Web server port) and expose the information through the Web server? Still, we have to build a whole new Web application to extract data from the core business applications. This will cost the delivery company money. All the company wants is to expose the delivery status and concentrate on its core business. This is where Web Services come in^[8]. Adding a Web Service, (also called an Application Programming Interface, or API for short) to an existing web site or desktop application (of the client-server variety) is a great way to enable additional and innovative uses of the data it holds, and also extend its reach to different development platforms such as native Apple and Linux applications, or to native mobile device applications such as those on Apple's iPhone as shown in **Figure (1)** below^[9].



Fig .(1) Web Services or APIs can be used to easily extend existing applications onto new platforms.

1.5 Web Service Communication

Web Services communicate by using standard web protocols and data formats, such as (HTTP, XML and SOAP).

1.5.1 SOAP

SOAP are remote function calls that invokes method and execute them on Remote machine and translate the object communication into XML format. In short, SOAP are way by which method calls are translate into XML format and sent via HTTP.

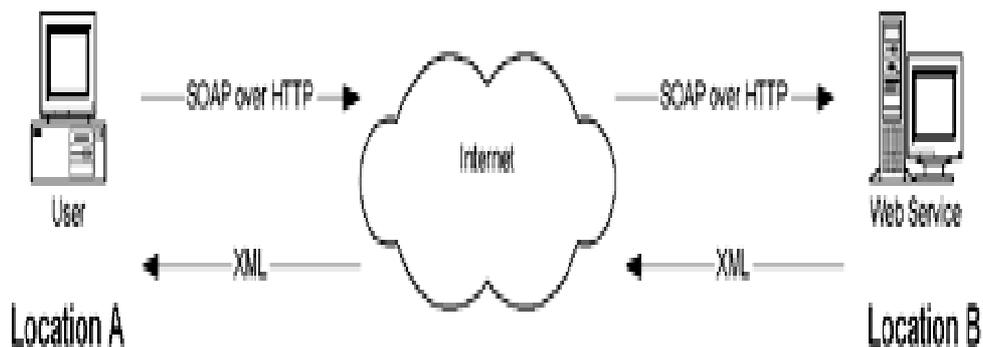


Fig .(2) SOAP with remote function calls.

1.5.2 WSDL

1.5.2.1 WSDL stands for Web Service Description Language, a standard by which a web service can tell clients what messages it accepts and which results it will return.

1.5.2.2 WSDL contains every details regarding using web service:

- a) Method and Properties provided by web service.
- b) URLs from which those method can be accessed.
- c) Data Types used.
- d) Communication Protocol used.

1.5.3 Discovery or .Disco Files

- a. Discovery files are used to group common services together on a web server.
- b. Discovery files .Disco and .Vs Disco are XML based files that contains link in the form of URLs to resources that provides discovery information for a web service.

1.5.4 Vs Disco File (dynamic)

A dynamic discovery files are dynamic discovery document that are automatically generated by VS.Net during the development phase of a web service.

1.5.5 Proxy Class

- a. A proxy class is code that looks exactly like the class it meant to represent; however the proxy class doesn't contain any of the application logic. Instead, the proxy class contains marshalling and transport logic.
- b. A proxy class object allows a client to access a web service as if it were a local COM object.
- c. The Proxy must be on the computer that has the web application.

2. Web Service Description Language Tool (wsdl.exe)

- 2.1** This tool can take a **WSDL** file and generate a corresponding proxy class that you can use to invoke the web service.
- 2.2** Alternate of generating Proxy class through **WSDL.exe** is you can use web reference. Web Reference automatically generates a proxy classes for a web service by setting a web reference to point to the web service.
- 2.3** Advantage of using **Web** Reference as compare to using **WSDL.exe** Tool is you can update changes done in web service class easily by updating web reference, which is more tedious task with **WSDL.exe** tool.

3. RSS on Steroids

RSS (originally [RDF](#) Site Summary, often dubbed *Really Simple Syndication*) is a family of [web feed](#) formats used to publish frequently updated works—such as [blog](#) entries, news headlines, audio, and video—in a standardized format^[13]. An RSS document (which is called a "feed", "web feed"^[14], or "channel") includes full or summarized text, plus [metadata](#) such as publishing dates and authorship. RSS feeds can be read using [software](#) called an "[RSS reader](#)", "feed reader", or "[aggregator](#)", which can be [web-based](#), [desktop-based](#), or mobile-device-based. The user subscribes to a feed by entering into the reader the feed's [URI](#) or by clicking a [feed icon](#) in a web browser that initiates the subscription process^[15].

3.1 Testing a Web Service

You can test web service without building an entire client application. With Asp.net you can simply run the application and test the method by entering valid input parameters. You can also use .Net Web Service Studio Tool comes from Microsoft^[11].

4. Main aspects of C# Language

C# is a hybrid of C and C++, it is a Microsoft programming language developed to compete with Sun's Java language. C# is an object-oriented programming language used with XML-based Web services on the .NET platform and designed for improving productivity in the development of Web applications. C# boasts type-safety, garbage collection, simplified type declarations, versioning and scalability support, and other features that make developing solutions faster and easier, especially for Component Object model plus and Web services^[12]. A Web Service Example in the following example we will use ASP.NET to create a simple Web Service that converts the temperature from Fahrenheit to Celsius, and vice versa, also web service to calculate the area of any circle according to it's radius. First of all to create the service we Start a New Web Site in visual studio, then choose ASP.NET Web Service.

The code written for the application which return area of circle and temperature conversion (Service.cs). This service contain a Circle Area method which calculate the area of circle also Fahrenheit and Celsius methods for the conversion. This document is saved as an asmx file. This is the ASP.NET file extension for XML Web Services.

```
%@WebServiceLanguage="C#"CodeBehind="~/App Code/Service.cs"Class="Service"  
""%
```

The next lines import the namespace "System. Web. Services" from the .NET framework:

```
using System;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml.Linq;
```

By default,ASP.Net Web Services use <http://tempuri.org/> unique namespace to distinguish it from other services on the Web:

```
[WebService(Namespace = "http://tempuri.org/")]
```

The next line defines that the "Service" class is a Web Service class type:

```
publicclassService : System.Web.Services.WebService
```

The next steps are basic C# programming. This application has three functions. One to convert from Fahrenheit to Celsius, and one to convert from Celsius to Fahrenheit, also to calculate circle area.

The only difference from a normal application is that this function is defined as a "**Web Method()**"

```
[WebMethod(Description = "This Service To calculate Area of Circle")]
publicdouble CircleArea(int Radius)
    { return 3.14 * (Radius * Radius); }

    [WebMethod(Description = "This Service To convert temprature from
Fahrenheit To Celsius")]
publicdouble Fahrenheit(double fhr)
    { return (((fhr) - 32) / 9) * 5); }

    [WebMethod(Description = "This Service To convert temprature from
Celsius To Fahrenheit")]
publicdouble Celsius(double cls)
    { return (((cls) * 9) / 5) + 32);}
```

Publish (execute) the.asmx file on a server with .NET support or type link: <http://localhost:61088/CircleArea/Service.asmx>, and you will have your first working Web Service as explained in the following **Figures(3,4,5,6,7):**

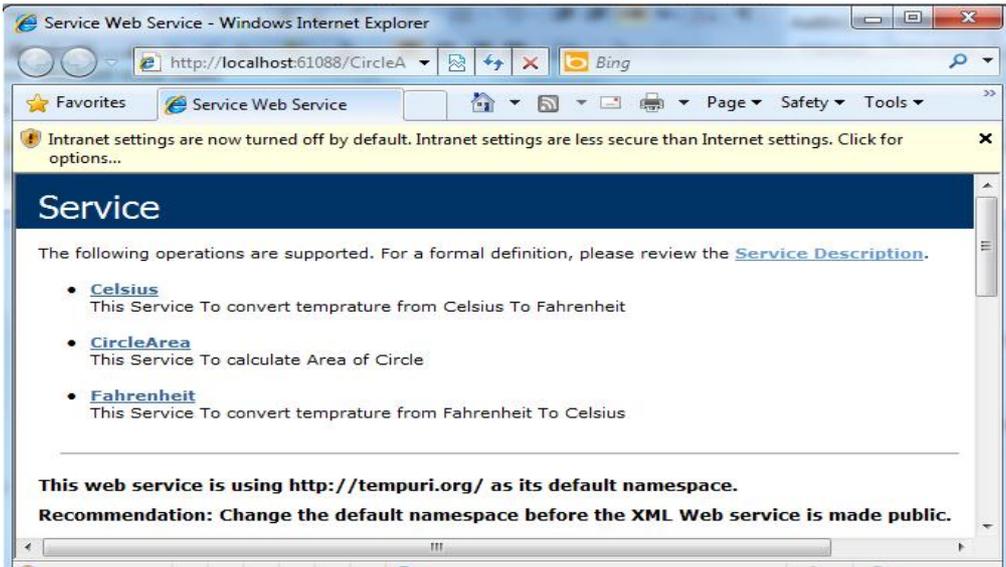


Fig .(3)Publish Web service on the Internet

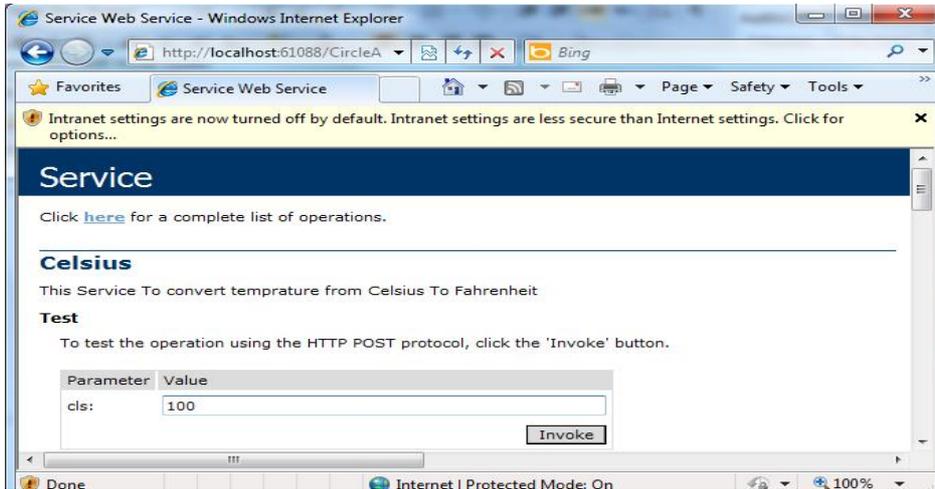


Fig .(4) calling Celsius Web method in web service.

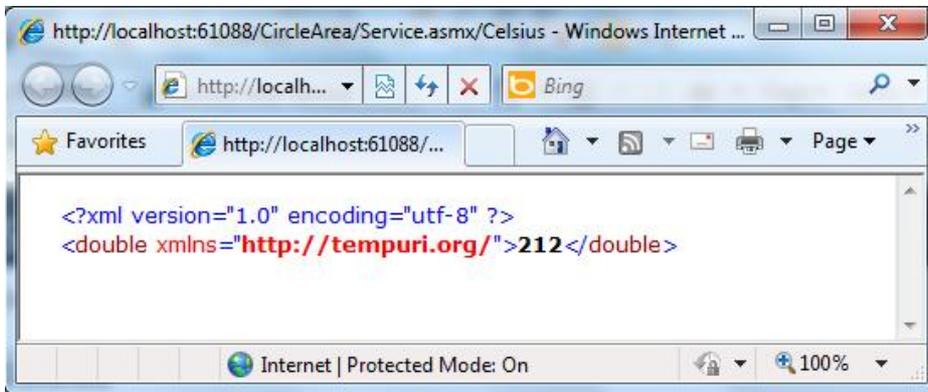


Fig .(5) get the Fahrenheit temperature over Web Service.

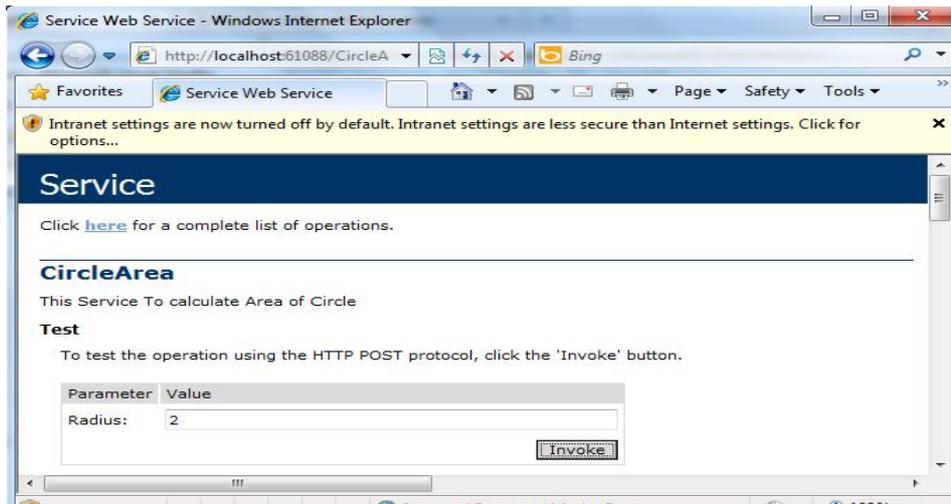


Fig.(6) calling CircleArea Web method in web service.

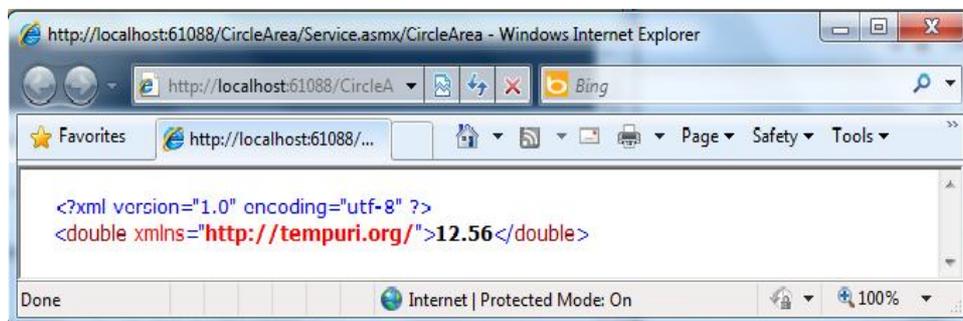


Fig .(7) get the Area of Circle over Web Service.

5. Weather Forecasting

Every time we wanted to use a weather forecast for our site, we found it too confusing. Each method we found had problems - some of the weather forecast Web services did not work properly. Eventually we noticed that Yahoo! has an RSS for its weather forecast, so we decided to make a program to use that in our site and we created a class for it! we hope it will be useful for others who want to have weather forecasts in their programs.

5.1 Yahoo! Weather RSS feed

The Yahoo! Weather RSS feed enables you to get up-to-date weather information for your location. You can save this feed in your Yahoo! or your favorite feed aggregate. You can also incorporate the RSS data into your own Web site or client application. The Yahoo! Weather RSS feed is a dynamically-generated feed that is based on zip code or Location ID.

5.1.1 RSS Request

The Yahoo! Weather RSS feed request follows simple HTTP GET syntax: start with a base URL and then add parameters and values after a question mark (?). Multiple parameters are

separated by an ampersand (&). There are two parameters for Weather RSS: *p* for location, *u* for degrees units (Fahrenheit or Celsius).

5.1.2 RSS Response

The Yahoo! Weather RSS feed is an XML document that conforms to RSS 2.0 specifications. RSS is an XML document containing: a single channel element representing the feed, several elements containing metadata about the feed itself, and one or more item elements describing individual items within the feed. As it is represented in an XML document, we can convert it to classes by XML Serialization in .NET.

6. Understanding XML Serialization in the .NET Framework

XML Serialization enables you to convert common language runtime objects into XML documents or streams, and vice versa. XML Serialization makes it possible to convert XML documents into such a meaningful format that all programming languages can process the converted documents without any difficulty. You don't need to know much about XML Serialization to use this code, but it can help you to change or customize it yourself.

C# Code of the Program:

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

[assembly: AssemblyTitle("YahooWeatherWinCSharp")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("Borna Rayaneh")]
[assembly: AssemblyProduct("YahooWeatherWinCSharp")]
[assembly: AssemblyCopyright("Copyright © Weather Web Service 2012")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace YahooWeatherWinCSharp
{
    static class Program
    {
```

```
[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```

After publishing(excuting)code over the Internet,will get the following Figures(8,9,10):



Fig .(8) Forecast Weather Home Page.



Fig .(9) Full Forecast at Yahoo Weather (Baghdad Weather).

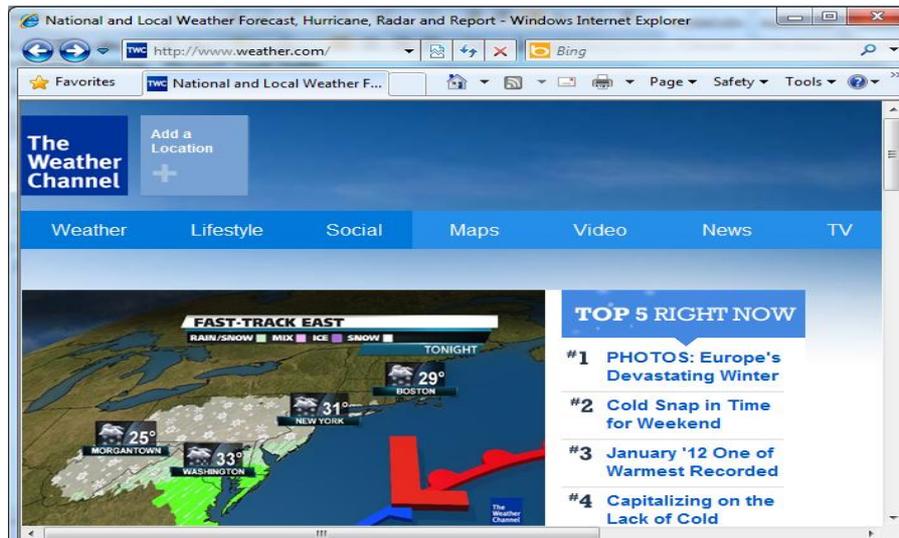


Fig .(10) The Weather Channel.

7. Conclusion

It is obvious from this paper the importance of web services and they playing important role in facilitating the concept of distributed computing between different systems running different programming languages. Web services facilitate collaboration and allow businesses to grow, companies can spend less time developing new applications. E-businesses can use Web services to provide their customers with enhanced shopping experiences.

Web Service is(Language Independent, Protocol Independent, Platform Independent and it assumes a stateless service architecture).

Web Service messages are formatted as XML, a standard way for communication between two incompatible system. And this message is sent via HTTP, so that they can reach to any machine on the internet without being blocked by firewall.

Each Response from web service is a new object, with a new state.

Web Service are asynchronous because the request object from the client application and the response object from the web service are unique SOAP envelopes that do not require shared connection.

This allow client application and the web service to continue processing while the interaction is ongoing.

Instead of a user interface, it provides a standard defined interface called a contract.

Web services are actually implemented using Remoting. Web services offers higher level of abstraction and is easier to use than Remoting, but Remoting is more efficient. So if both ends of the communication are within your control, use Remoting. If the other end is written in other platform, for the sake of interoperability, use web services.

References

1. by Sheikh M. Haris, Creating Web Services with C# , Developers Mania, May 27, 2010.
2. by Deitel & Associates, Visual C# 2005 How to Program, second Edition, pages: 1535, 2006 .
3. Chaitali Gupta, Rajdeep Bhowmik, Michael R. Head, Madhusudhan Govindaraju, Weiyi Meng Department of Computer Science, State University of New York (SUNY) at Binghamton , A Query-based System for Automatic Invocation of Web Services, IEEE International Conference on Web Services (ICWS 2007), 2007.
4. Carlo Blundo, Emiliano De Cristofaro, Clemente Galdi, Giuseppe Persiano, Validating Orchestration of Web Services with BPEL and Aggregate Signatures, IEEE , 2008.
5. Li-Qun Kuang, Yuan Zhang, Xie Han, Design and Implementation of Network Test System Based on Web Services, IEEE, 2009 .
6. Zhendong Ma, Christian Wagner, Thomas Bleier Safety & Security Department, Model-driven security for Web services in Government system: ideal and real, IEEE, 2011.
7. By John Charles Olamendy, The evolution of Web Services and .NET 2.0, c# corner, June 07, 2006
8. By Chris Peiris, Creating a .NET Web Service, codeguru Developer.com site, 2012
9. By: Andrew Freemantle, Securing ASP.Net Web Services with Forms Authentication, DotNetSlackers.com, Published: 30 Apr 2010
10. PhycoCoder, Introduction to Web Services with C#, Web Development, Jan 20, 2012.
11. Microsoft.net, All About Web Service in .Net, September 23, 2007
12. Free CSS Templates, C# Tutorials, DeepSearcher.com, 2008.
13. Libby, Dan (1999-07-10). "RSS 0.91 Spec, revision 3". Netscape Communications. Archived from the original on 2000-12-04. <http://web.archive.org/web/20001204093600/http://my.netscape.com/publish/formats/rss-spec-0.91.html>. Retrieved 2007-02-14.
14. "Web feeds | RSS | The Guardian | guardian.co.uk", The Guardian, London, 2008, webpage: GuardianUK-webfeeds.
15. From Wikipedia, RSS feeds, Wikipedia® , 27 December 2011.