AGRICULTURAL WEB SERVICES AS DECISION MAKING TOOL FOR FARMERS

Akhtar Ali Jalbani[1], Aneela Yasmin[2], Mansoor Hyder Depar[1] and Gordhan Das Menghwar[1]
[1]Information Technology Center,
Sindh Agriculture University, Tandojam, Pakistan
[2]Department of Biotechnology
Sindh Agriculture University, Tandojam, Pakistan
(akjalbani,aneelayasmin,mansoor.hyder,gdas)@sau.edu.pk

ABSTRACT: Most of the organizations pay high costs to the variety of applications used for exchange of information to the customers, business partners and even within organization. These all play vital role in the success of their business. Different problems can exist in information exchange, one of the common problems of data store and exchange in a dissimilar way. These problems may lead to decrease in the productivity of the organization. To overcome these problem's web services have been introduced, which are practical and cost-effective solutions for these types of problems. Web services have various advantages, for example, platform independent, and language independent, which were impossible in the past. In this paper, we have focused on agriculture-based web services and proposed different web services that can be useful for the agriculturist and the farmers. A simple weather forecast web service has been practically implemented and tested.

1. INTRODUCTION

In Web services are software components uses web technology standard such as HTTP and XML. These are designed in such a way that other applications can be accessed easily. Web service is currently a modern way to distributed computing application in which services are created and accessed in the form of client/ server application, or it is another distributed technology like EJB, RMI or CORBA[1]. The web services have certain advantages over other distributed computing technologies, for example; Web services are platform independent and language independent; this is because web services use XML standards. This provides great advantages that client accessing the web service need not be the same. The client using C++ under windows environment can access any web service running on Linux server, while the web services are programmed in Java or any other language. Another great advantage of using web services on the Internet is a usage of HTTP protocol for transmitting the service's request and response. This enables flexibility to use a web service [2] without troubling HTTP traffic unlike CORBA because internet proxies and firewalls do not create any troubles for HTTP traffic.

Web services use three core technologies based on XML; these are: WSDL, SOAP and UDDI [3,4,5]. WSDL document is first step to create a web service; this document describes service location and functionality the service provides. UDDI document should contain the entry of the service information, which is also called as UDDI registry. This allowed web service consumer to fetch and locate the required services [5].

a) XML (eXtensible Markup Language)

XML is a meta-language for describing data, and it is standardized by W3C (World Wide Web Consortium). Data is described hierarchically under text-based tags; hence these tags are the core syntax of the XML to make it text-based markup language. The tagging being used to describe the data for application independence, and also it is human readable. The XML becomes widely acceptable as a markup language because of its simplicity and interoperability. Hence, this has been adopted as the standard for exchanging information for applications mostly includes web services.

Due to the interoperability of XML, it provides a basis for the modern web services. As the core technologies of the web services such as WSDL, SOAP and UDDI are based on XML based messages that can be interpreted by any machine [6].

b) WSDL (Web Services Description Language)

WSDL is a web service description language which is also based on XML. It is also W3C standard language. It is used when a client wishes to access the web service, with this language one can read and interpret its WSDL document. With this language, one can easily point the location, the service and its operations available for further processing. WSDL is also called as an initial document or interface that presents all required information to interact with the web service. The WSDL XML document contains six main elements these are port type, port, message, types, binding and service. Port type describes the operations; port defines communication port; message contains the format of the message; types are the data types; binding defines the communication protocol, and services define the URL for accessing the service.

c) SOAP (Simple Object Access Protocol)

Another XML based protocol SOAP is standardized by W3C that is used for exchanging data over HTTP. It sends XML messages between applications. Web services use SOAP for communication between service and its clients. SOAP messages sent between applications without any platform or programming language issues. This is due to the XML language that facilitates platform and language independency and also the HTTP based communication, as HTTP is supported by all web servers and browsers. SOAP is also an XML document that contains following elements. 1) Envelop describes that XML documents is a SOAP message; 2) Header contains information relevant to the message; 3) Body contains all message's payload; 4) fault contains information about client or server error. The Header and Fault are optional for the SOAP document.

SOAP message is responsible for exchanging information between client and web service. The client and web service

uses request and response SOAP messages for the communication. The format of these messages is already defined in the WSDL contract when creating the SOAP message. Hence the messages will be compatible.

d) UDDI (Universal Description Discovery and Integration)

UDDI provides a uniform way for creating web services. It provides complete specification for creating an XML based registry that lists the information about the businesses and web services. IDDI can be implemented in a different way, for example, It can be used to describe services using WSDL and communicate via SOAP messages. UDDI registry is an optional step, when registering a web service. UDDI registries can be public or private. When UDDI is used with WSDL to look for a web service, A developer can query a UDDI registry to obtain WSDL for the web service. Developers can design their client web service application to receive automatic updates about any changes to a service from the UDDI registry.

In web service application, client program contact web service (server) and send a request for particular service for example weather information. The server returns the available forecast in the response to the server request. Hence request-response protocol exists in the web service for the client and server communication. This process has been illustrated in figure 1.

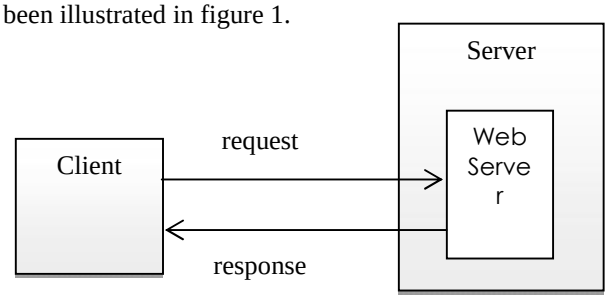

Figure 1: General Web Service Request and Response Flow.

Web service is not useful in some cases such as overhead in the sense that it is not a wise decision to transmit all data in XML. XML only provides portability [7]. Hence when deciding data to be transmitted using XML, we can win portability, but we can lose efficiency because transmitting binary data over the network is more efficient. It is not a too serious problem because you will never find any critical real time issues in web service application. Another disadvantage of using a web service could be the lack of versatility because web services only allow basic forms of service invocations. Other parallel solution such as CORBA offers a lot of supporting services to the programmers. Still work is needed to make web service's applications more versatile.

2. WEB SERVICE ARCHITECTURE FOR AGRICULTURE DOMAIN

Agriculture is one of the major domains, where people rely on the dynamic information, for example, weather information is a key attribute for agriculture domain. Farmer's production can depend upon the forecasts. Weather effects on the crops; hence dynamic and accurate information can help the farmers to predict and can make better arrangements to reduce the weather effects on their crops [8,9,10]. A typical weather information web service has been discussed in this paper. Figure 1 provides a basic flow of the request-response system of the web service. Before using a web service, it must be invoked. Figure 2 provide basic architecture of a web service; the details are described in the steps

Step-1: Client should have knowledge about the web service, which he wants to invoke. Hence the first step is to discover the web service that meets the customer requirements. For example, Farmer is interested to the coming month forecasts in specified city. This process of discovering web service will contact with the discovery service of the web service to identify the web service name and the required information about the service.

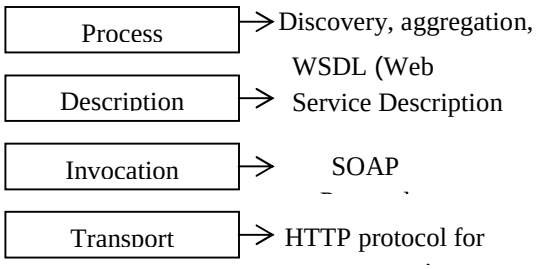

Figure 2: Web Service Architecture

Step-2: The discovery service will respond with a list of the services that matches the requirement of the farmer.

Step-3: we know the location of the service by providing the city, but he has no idea how to invoke it. The method to invoke the service will be like this for weather forecasting system i.e.; String getCityWeather(int postalcode).

Step-4: WSDL code will be transferred in the response of a reply.

Step-5: Finally, web service can be located, and invocation can be made itself by SOAP. SOAP then further processes our request to provide the web service response by providing farmers the coming month forecast of that particular city.

Web services are used in various decision making products, for example, weather information paly important role farmers. Agricultural web service means that farmers always have access to up-to-date information about the weather. Additionally, agriculture weather can also help farmers to take right decisions for sowing or spraying fields with pesticides [11,12,13].

Agricultural weather can include regular weather updates, forecast, weather statistics, positioning, one-month weather forecast and so on.

We have proposed web services for Agriculturist in Figure 3, in which farmers or agriculturist can take decision based on a different type of current data available to them using web services. In this paper, we have only Discussed Weather forecast as decision tool; other tools can be tentative crop outlook, and Agricultural advisory services in which different frequently asked questions will be answered using web service's data. Other forecasts necessary for farmers could crop yield forecast [14].

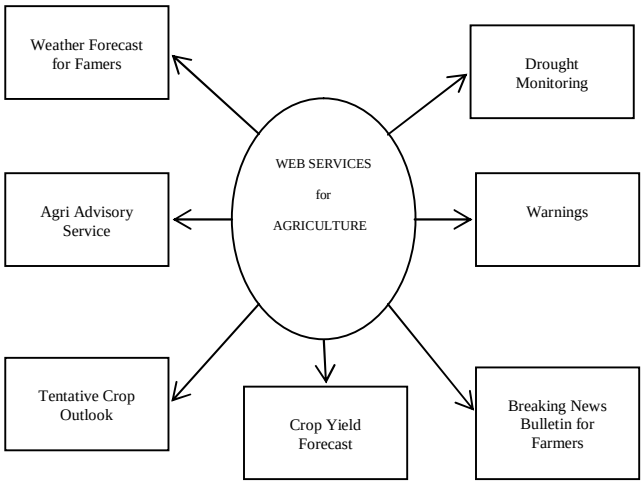

Figure 3: Proposed Web Services for Agriculture

The warning and drought monitoring can also be important part of the web service. Breaking news can be popped up or delivered to the farmers on their requests. These all makes integrated web services, which can be help for the farmers or agriculturists. Hence Information Technology used in agricultural as a decision tool in which real time knowledge is provided to the farmers.

3. METHODOLOGY

Web services can allow various applications to communicate with each other even from different sources without wasting time to custom source code. XML is the main language which is used in the web service that provides platform independent facility. In this paper, we provide a simple weather information web service for farmers to make any decision for upcoming crops or even spraying the fields based on the atmospheric temperatures.

To build any web service, we need a web server such as apache, tomcat or WAMP, XAMP and so on and for client any web browser is required to send requests to the server. We have created web service using visual studio, hence we explain how we have made agriculture weather information as a web service.

Step-1 create a new project in the visual studio.

Step-2 Add web service URL from which we need to import the service data.

Step-3 compose a web service i.e composition of the web service.

Step-4 Design & code in XML

Web service Composition can be done as follows:

```
Namespace WebServiceComposition
{[Activity(Label="AgriWeather, MainLauncher=
true….]

        public class Activity1: Activity
        protected override void
        onCreate(Bundle bundle)
        {……..……..……..…..}
                                    ,
```

Above source is automatically created by tool, in this code, we should add location for which we need the current weather updates and the description of the web service.

Some of the code snip of WSDL document is list here:

```
<wsdl:message
name="GetWeatherInformationSoapIn">
  <wsdl:part name="parameters"
element="tt:GetWeatherInformationResponse"/>
</wsdl:message>
```

4. CONCLUSION

A prototype web service for agriculture farmers based on weather information was developed in the visual studio. The developed prototype gets real-time information form weather information service and provides SMS based delivery of current updates about storms, rainfall and increase or decrease in temperature.

The updated information about weather helped farmers to take decisions about what type of crop should be considered and when spray should be done. These decisions improve the confidence level among the farmers to increase their productivity.

Our proposed framework of web services for agriculturist and farmers is an integrated framework where more than one web services are proposed for farmers. In this paper, we have implemented only weather forecast web service as a prototype. The results of the prototype web service were encouraging, and the weather web service can be implemented on the real mode in our proposed framework.

Additionally, the web services crop yield forecast, warnings, breaking news, drought monitoring, tentative crop outlook and agri advisory service can be implemented to provide a bulk of web services to the farmers. The combined information of this integrated framework definitely will help to the agriculturist and farmers to make quick and reliable decisions for their crops.

REFERENCES

[1] W. Emmerich, N. Kaveh "Component Technologies: Java Beans, COM, CORBA, RMI, EJB and the CORBA Component Model". *In Proceedings of 24th International Conference on Software Engineering*, 2002.

[2] Query, B. Lee. "Web Services: Program Integration across Application and Organization boundaries". *http://www.w3.org/DesignIssues/WebServices.html*

[3] J. Clabb "Web Service Gotchas". *http://www.ibm.com/developerworks/webservices/library/ws-gotcha.html*.

[4] F. Cohen "Myths and misunderstandings surrounding-SOAP". *http://www.ibm.com/developerworks/webservices/library/ws-soap.html*

[5] J. Snell, D. Tidwell, P. K." Programming Web Services with SOAP. *O'Reilly & Associates*, 2002.

[6] A. Zdanowicz, R. Kaschek , K. Schewe, Be. Thalheim"Context-aware Web Information Systems" *Proceedings of the first Asian-Pacific conference on Conceptual modeling*, 2004.

[7] A. Pashtan, R. Blattle, A. Heusser, P. Scheuermann, "CATIS: A Context-Aware Tourist Information System", *International workshop of mobile computing*, 2003.

[8] B. Soukkarieh, and F. Sedes, "Towards an Adaptive Web Information System Based on Web Services," *4th International Conference on Autonomic and Autonomous Systems ICS*, 2008

[9] S.Fountas, D. Wulfsohn,B.S. Blackmore, H.L Jacobsen., and S.M Pedersen. "A model of decision-making and information flows for information-intensive agriculture". *Agricultural Systems*, **87(2),** 192-210, 2006.

[10] D.C Jochinke, B.J Noonon, N.G Wachsmann, N.G., and R.M Norton." The adoption of precision agriculture in an Australian broadacre cropping system--Challenges and opportunities". *Field Crops Research*, **104(1-3)**,68-76, 2007.

[11] N.R Kitchen. "Emerging technologies for real-time and integrated agriculture decisions" *Computers and Electronics in Agriculture*, **61(1)**, 2008.

[12] R. Nikkilä,I. Seilonen, and K. Koskinen." Software architecture for farm management information systems in precision agriculture". *Computers and Electronics in Agriculture*, **70(2)**,2009.

[13] C.G Sørensen, L. Pesonen,S. Fountas, P. Suomi, D. Bochtis,P. Bildsø,. "A user-centric approach for information modeling in arable farming". *Computers and Electronics in Agriculture*, **73(1)**.2010.

[14] J. Wolfert, C.N Verdouw, C.M Verloop, and A.J.M Beulens, "Organizing information integration in agri-food--A method based on a service-oriented architecture and living lab approach." *Computers and Electronics in Agriculture*, **70(2)**, 2010.