Web Services for Environmental Informatics

Erick Arauco

Lorenzo Sommaruga

Web Services for Environmental Informatics

Erick Arauco [a] **and Lorenzo Sommaruga** [b]

[a] *University of Piura - Engineering Department ,Piura, Perú- earauco@udep.edu.pe*

[b] *University of Applied Sciences of Southern Switzerland - Innovative Technologies Department (DTI), Switzerland – lorenzo.sommaruga@supsi.ch*

Abstract: This paper presents the description of an open architecture for the management of environmental content using Web Services. The Web Services technology can be effectively exploited for integrating on one hand the needs for dissemination of analytical data about environment, such as air, noise, traffic, etc., and on the other hand the needs of different users concerning the accessibility requirements of their devices, distributed and heterogeneous systems, remote and mobile control access. The case study of this paper is based on the OASI (Environmental Observatory of Southern Switzerland) project that permits access to the air, noise and traffic measures for Southern Switzerland. Other than through traditional web pages, this access is also made possible thanks to the deployment of software applications based on Web Services. To this aim, a number of Web Services are defined using the UML analisys and developed. They identify a) a group handling the air information, b) a group implementing the access to the noise information and c) a group implementing the access to the traffic information using the OASI's database. In addition, another group allows users to manage his/her own information, e.g. e-mail, OASI's news, user's group, etc. In conclusion, the Web Services Technology could be a good solution for the management of environmental content because it provides open and mobile access to data, interoperability among different client-server nodes, easy extensibility for integrating any kind of device into the system.

Keywords: Web Services; mobile; J2ME; WAP; SOAP

1. Introduction

The Web Services technology [W3CWebServices, ApacheWebServices] can be effectively exploited for integrating on one hand the needs for dissemination of analytical data about environment such as air, noise, traffic, etc., and on the other hand the needs of different users concerning the accessibility requirements of their devices, distributed and heterogeneous systems, remote and mobile control access.

The main advantage of Web Services is to offer an open architecture for any type of client in a simple way. In fact, using Web Services any client can access the same environmental information independently from its platform, language, and above all device.

This paper describes relevant studies and experiments concerning a web services-based extension to the OASI (Environmental Observatory of Southern Switzerland) project [Andretta et al., 2004] in order to support the access of the air, noise and traffic measures for Southern Switzerland from any device. To this aim, the application context is firstly introduced, followed by a presentation of Web Services and a description of an open architecture for the management of the environmental contents using Web Services.

2. OASI Project

The OASI project originates from the risk of the environmental degradation caused by the increasing traffic or by the constructions of new buildings. In this situation, the Southern Switzerland local government (Cantone Ticino) needs tools to monitor and control this risk.

The main goals of the OASI project are:

- To control the effects on the environment within 20 years;
- To guarantee a management and a modern access to information;
- To offer in real time important information to the authorities and the population.

The system is based on the observation and collection in a database of information about traffic, emission, and their effects.

The previous solution is offering internet or intranet access to this information through an architecture consisting of a Java (stand-alone) or an Internet browser client, a web server and application server (Tomcat [Tomcat]), and an Oracle database, as shown in Figure 1.

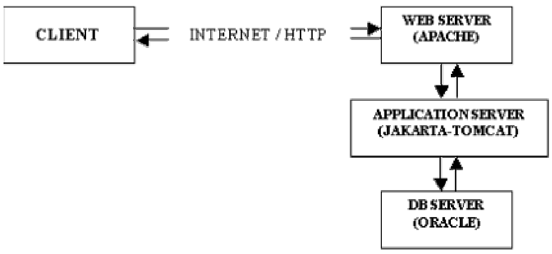

Figure 1. Initial OASI Architecture.

This architecture is presenting limitations on the data accessibility which depend on different user's devices, interoperability issues of various distributed and heterogeneous systems and the need for remote and mobile control access.

Within this context the Web Services technology has been considered.

3. Previous Works

Some research works related to this project support as well the demonstration of the feasibility of a Web Services based approach and the constant search of a standard architecture for client-server communication based on this technology.

Three relevant projects can be mentioned within the context of the present study concerning environmental data management and Web Services. Dwyer and Clark [2002] in "Web Services Implementation: The Beta Phase of EPA Network Nodes" describe an introduction to XML and SOAP, and how they are used in the U.S. Environmental Protection Agency architecture to generate node services requests and to protect the data exchanges. The APNEE - TU project [Karatzas] has been designed to offer environmental information to different users. The information can be accesed from diverse channels like J2ME, WAP, SMS, internet, voice servers, etc. through an informatic portal that provides real-time information; and the MINNE project [MINNE] is a research project in the area of mobile computing for collecting, reporting, and delivering Environmental Information. The aim of the project is to find new ways to access the enviromental information, specially through mobile technology.

4. Web Services

Web Services are a technology that permits the integration of heterogeneous systems into a neutral platform. They can be considered as an interface which describes a set of operations made accessible on the network through standard XML messages [Sommaruga, 2003].

Their main characteristics can be summarized in the following points:

- Information interchanges with other web services
- Accessible across many protocols such as HTTP, SMTP, etc.
- Based on standard XML (Extensible Markup Language) languages
- Neutral platform, i.e. they do not depend on any platform
- Offering compatibility of heterogeneous systems.

Standard	Technology	Purpose
[XML]	Extensible Markup Language	Content's definition language
[SOAP]	Simple Object Access Protocol	Communication protocol based on XML
[UDDI]	Universal Discovery, Description and Integration	Public directory service that offers information about web services
[WSDL]	Web Services Description Language	Protocol that describe web service abilities

Table 1. Web Services main standards.

Other standards on which the Web Services technology is based are described in Table 1.

In particular, SOAP is a protocol based on XML that defines the message format. It is independent of the platform, programming language, and device, allowing in this way a large interoperability.

How a web service generally works is presented in the following Figure 2.

An application via a SOAP client retrieves service information by asking to UDDI registry for a necessary specific service; the UDDI registry searches for the required service and gives its description to the client (1).

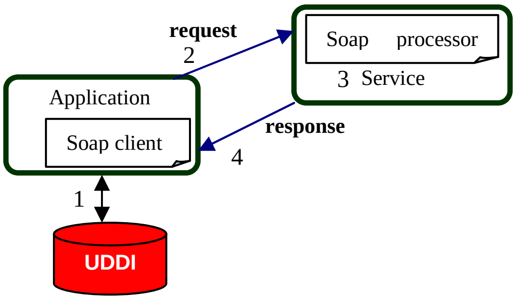

Figure 2. Web Services working flow.

On the basis of this information, the client can communicate with the SOAP service requesting by means of a SOAP message (2) the execution of an operation. Once the service is executed (3), the result is then passed back to the requesting client in the form of a SOAP XML message (4).

5. A Web Services based open architecture

One of the goals of the project was to offer a standard for the client-server communication and to extend its use on wireless devices.

To this aim, an architecture based on the web services and the tree-tiers model [Sun] has been adopted. In this architecture all the client-server communication is based on XML and SOAP. On this basis, any client that understands SOAP can be added as an interface (see below the Presentation level).

The basic principle is the possibility to wrap the functionalities the OASI system has to offer into web services and to transform each client application into a web services SOAP client.

In this way each actor in the system on the server and on the client, i.e. data providers and data consumers, can "speak" the same language, understanding each other and therefore interoperate in an effective way.

The three-tier model allows the functionalities of a distributed system to be separated into three layers or levels:

- *Data level*; where data are stored for instance in a database server.
- *Logical level*; where the web services are defined and operate. Here it is supported by an application server and a Web services SOAP listener.
- *Presentation level*; where all the user interaction occurs. It may consist of any client that needs to access the information. In our system it has been developed in the form of a WML (Wap) user client, a more advanced Java (J2ME) user interface, and an SMS client via a GSM line.

The separation of these levels is detailed in Figure 3. From this picture emerges the clear separation of the data from its processing and use in the logical level and particularly from a number of different clients which can independently access the data. It is worth noting that the access to the data is controlled exclusively in the logical level and it is interfaced to the client in a uniform way by means of SOAP messages.

For instance, the structure of a SOAP message for requesting some data from the "AirService" and getting a response are in a standard easy to read format, like in the following sample message excerpts where two measured values are returned (170 and 150), for the time 8:00:00 and 7:00:00 respectively as underlined in the messages.

Sample Request Message

```
<SOAP-ENV:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
...>
  <SOAP-ENV:Body SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/">
    <GetInformation xmlns="urn:AirService" id="o0" SOAP-
ENC:root="1">
      <place xmlns="" xsi:type="xsd:string">1</place>
      <date xmlns="" xsi:type="xsd:string">21-07-03</date>
      <param xmlns="" xsi:type="xsd:string">PM10</param>
      <unit xmlns="" xsi:type="xsd:string">ug/m3</unit>
    </GetInformation>
  </SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Sample Response Message

```
 <soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
...>
    <soapenv:Body>
      <ns1:GetInformationResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/en
coding/" xmlns:ns1="urn:AirService">
      <ns1:GetInformationReturn xsi:type="soapenc:Array"
         soapenc:arrayType="xsd:anyType[2]"
     xmlns:soapenc="http://schemas.xmlsoap.org/soap/encod
ing/">
         <item href="#id0"/>
         <item href="#id1"/>
       </ns1:GetInformationReturn>
     </ns1:GetInformationResponse>
     <multiRef id="id1" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/en
coding/" xsi:type="ns2:ObjAirData"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="http://Oasi">
       <mcode xsi:type="xsd:string">A002</mcode>
       <time xsi:type="xsd:string">8:00:00</time>
       <value xsi:type="xsd:string">170</value>
```

```
</multiRef>
<multiRef id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/en
coding/" xsi:type="ns3:ObjAirData" xmlns:ns3="http://Oasi"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
>
<mcode xsi:type="xsd:string">A001</mcode>
<time xsi:type="xsd:string">7:00:00</time>
<value xsi:type="xsd:string">150</value>
</multiRef>
</soapenv:Body>
</soapenv:Envelope>
```


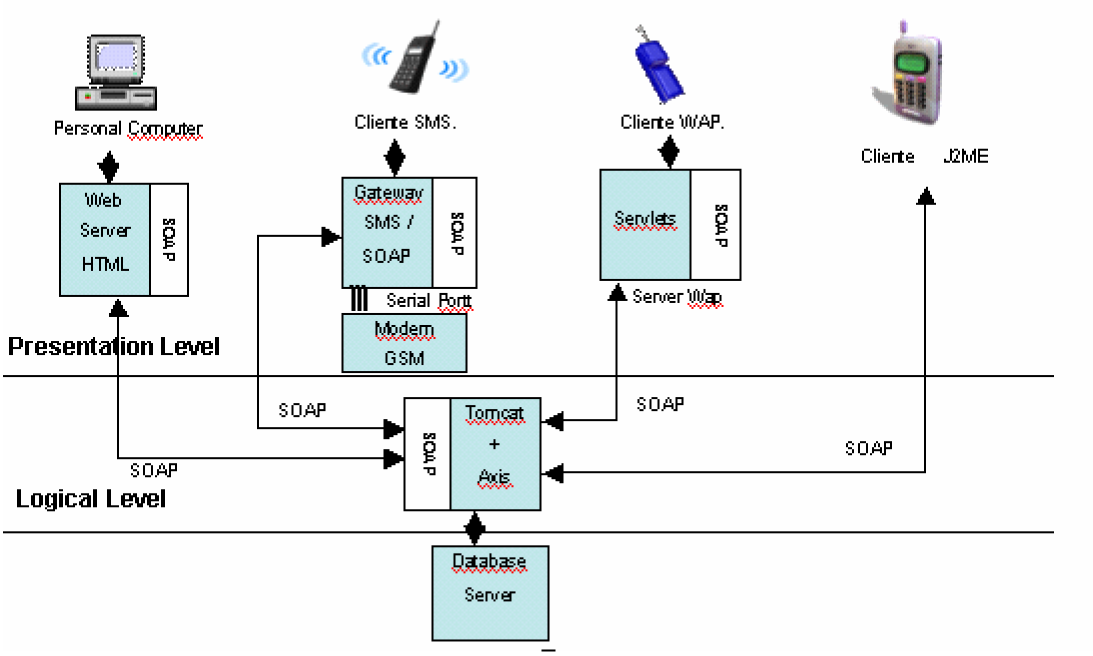

Figure 3. Web services based open architecture.

The graphic corresponding to these request and response messages could be elaborated and displayed in client, such as a Java enabled mobile device as later presented in Figure 5.

6. Implementation

This section describes how Web Services has been implemented in the OASI project prototype. The project has been primarily a feasibility experiment within the main OASI project for testing the use of the web services technology in order to provide more mobility and accessibility to its environmental data.

A UML analysis has been firstly carried out for the specification of Web Services. This analysis allowed us to identify how the logical level can function. In this point, a number of Web Services are defined and developed. They identify a) a group handling the air information, b) a group implementing the access to the noise information and c) a group implementing the access to the traffic information using the OASI's database. In addition, another group allows users to manage his/her own information, e.g. e-mail, OASI's news, user's group, etc.

In a second phase, it has been developed and implemented using real devices that supports J2ME and WAP technology.

J2ME [J2ME] is a Sun's Java developing platform for programming applications dedicated to mobile devices which support the java virtual machine. Such typical devices include many mobile phone models of the last generation.

The J2ME language is based on the midlet concept. A Midlet is a portable programming code that consists of a particular java class. This code can be downloaded via a data-phone connection into the mobile device and can be run as a local application by the local virtual machine.

In order to simplify the development phase, emulators for J2ME and WAP clients have been exploited in our project. A final testing has been accomplished on real mobile devices such as Nokia Java enabled phones.

All users that have a J2ME phone or WAP phone can have access to the information via Web Services, although it is important to note that these services may have a cost, and these costs depend on the mobile operators in relation to the bytes transmitted.

The present project can be helpful because it shows how the same information can be delivered on diverse formats (text, graphic, etc).

Currently, this experimental phase has been completed and has allowed the architectural and technological feasibility to be validated. In a future phase, a plan for the real deployment and integration of the Web Services approach into the OASI environment will be evaluated by the main project's responsible (Canton Ticino).

The specific languages and technologies used for the system implementation are here summarized.

A database Server Oracle 9i was used for retrieving the data in the *Data* level. The web services of the *logical* level were developed using the Soap Server Axis 1.1 on Apache Tomcat 4.1.27 as the application server (i.e. servlet container) on Java SE 1.4.03 and also as the Wap Server. In the *Presentation* level, the Java MicroEdition (J2ME) and its extension for supporting Mobile Information Device Profile (MIDP 2.0) have been used on the Java enabled mobile clients additionally supported by KSoap on J2ME for the Soap Client. For the other type of clients considering Wap devices, the Wireless Application Language (WML) was used supported by Axis 1.1.

The next pictures show samples of the interaction of both a J2ME and a WAP client within the OASI system.

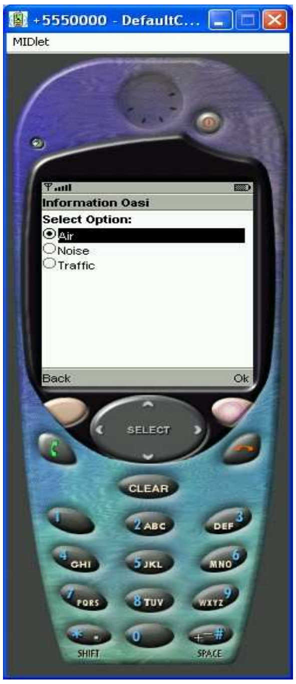

Figure 4. Selection of the type of data to be shown in a J2ME Client.

Figure 5. Example of Air graphic in a J2ME client.

In Figure 4, a menu in a J2ME Client is offering the possibility of a selection of the type of data to be shown. The result of this selection, and a following definition of a time range, is shown in Figure 5, where it is possible to observe a statistic graphic which permits users to analyze the air's measurements in the Canton Ticino – Switzerland.

Figure 6. Air information measures displayed in a J2ME Client.

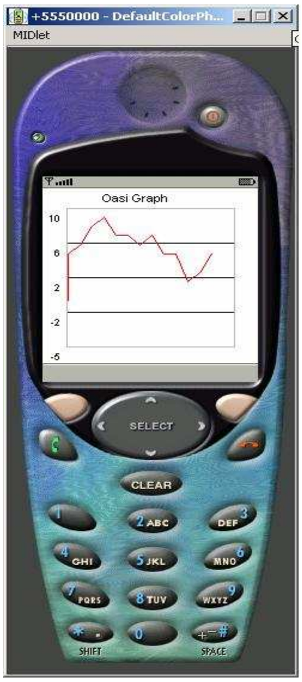

Figure 7. Example of PM10 (ug/m^3) Air pollution variation in a J2ME Client.

Figure 6, presents the graphic interface of a J2ME client for requesting air measurements in a given place at a specified date.

An example of PM10 (ug/m^3) Air pollution variation over one day, previously specified, with a time interval of 1 hour, is displayed in a J2ME Client (Figure 7).

Figure 8. One of the OASI functionality menus for the interaction in a Wap Client.

An initial interaction in a Wap Client concerning the possibility of selection of various functionalities is presented in Figure 8. This selection permits the presentation of environmental information and graphics, user data management (profile information, login, and email information and settings), and some news.

7. Conclusions

A presentation of an innovative application of the Web Services technology to the environmental field has been presented. An open architecture for the management of the environmental contents using Web Services has been introduced. This architecture presents a number of advantages with respect to traditional systems and solutions, including:

- The system can be accessed from anywhere, i.e. open and mobile access to data, both for accessing and for administering them.
- The possibility of achieving a standard way for client-server communication based on SOAP and XML.
- The possibility of integrating any kind of device into the architecture without touching or modifying the underlying *logical* and *data* level.

This architecture could be expanded and successfully applied to other similar domain problems, where it will be possible to easily integrate heterogeneous systems via Web Services abstractions. Using Web Services any client can access the same environmental information independently from its platform, language, and above all device.

The architecture proposed was implemented in particular for mobile clients that support J2ME or WAP technology, where the environment data can be displayed and presented to the end user in a textual or graphical format according to the specific device profile.

The development of this system architecture can be summarized in three steps according to the three-tier levels:

1) *Data* level: representing the information, i.e. the environmental data, for instance using a DB, as in our case.

2) *Logical* level: building the main web service(s) for interfacing the data and exposing the accessible functionalities (WSDL's <operations>) to be used by the clients.

3) *Presentation* level: developing the various kinds of client, wherever and whenever necessary, as we did for J2ME, Wap and SMS.

A number of issues also emerged in the project, including low capacity for processing the information or limited display in the mobile devices. However, we hope that in the future these technical problems will be resolved by the rapid evolution of technology.

In conclusion, the OASI's Mobile Web Services have demonstrated that the OASI's team may use the Web Services Technology to exchange environmental data. Future research will allow us to extend the project to other devices like SMS phones through a gateway implementation.

6. Acknowledgements

The authors wish to thank the University of Applied Sciences of Southern Switzerland and Canton Ticino for their kind collaboration and for providing us with all available data.

Special thanks to RETECA Foundation which allowed Engineer Erick Arauco to complete his Masters in Advanced Computer Science studies.

The work described in this project has been accomplished in partial fulfilment of Arauco's Master Diploma in Advanced Computer Science at SUPSI during year 2003 [Arauco].

7. References

ApacheWebServices, Apache Web Services Project, URI: http://ws.apache.org

Arauco, E., Servizi Mobili per il Monitoraggio Ambientale del Progetto Oasi. University of Applied Sciences of Southern Switzerland Manno, Switzerland, Oct. 2003.

J2ME, Java 2 Platform, Micro Edition (J2ME), 2003 URI: http://java.sun.com/j2me/

Andretta, M., G. Bernasconi, G. Corti and R. Mastropietro, OASI: An Integrated multidomain Information System, *iEMSs 2004, The International Environmental Modelling and Software Society Conference*, 14-17 June 2004, University of Osnabrück, Germany, URI: http://www.iemss.org/iemss2004/

Dwyer, C., Clark Chris, Nobles M., Web Services Implementation: The Beta Phase of EPA Network Nodes. *International Emission Inventory Conference "Emission Inventories - Partnering for the Future,"* Atlanta, GA, April 15-18, 2002, URI: http://www.epa.gov/ttn/chief/conference/ei11/datamgt/dwyer.pdf

Karatzas, K., Moussiopoulos, N., Providing timely and valid air quality information via human – centered electronic services: The Apnee Project. *Aristotle University of Thessaloniki, Greece*, URI: http://www.edie.net/library/features/ENB046.HTML

MINNE Mobile Environmental Information Systems project, University of Oulu, Finland, URI: http://www.minne.oulu.fi

SOAP, Web services SOAP, URI: http://ws.apache.org/soap/

Sommaruga, L., Web Services and SOAP, Postgraduate Course, University of Applied Sciences of Southern Switzerland, 2003 URI: http://www.macs.supsi.ch/

Sun, Web-Tier Application Framework Design, 2002, URI: http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html

Tomcat, The Jakarta Site – Apache Tomcat URI: http://jakarta.apache.org/tomcat

UDDI, UDDI.org URI: http://www.uddi.org/

W3CWebServices, W3C's Web Services Activity, URI: www.w3.org/2002/ws/

WSDL, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 200, URI: http://www.w3.org/TR/wsdl

XML, Extensible Markup Language (XML)URI: http://www.w3.org/XML/