Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 147-156

Using Web Services to Interoperate Data at the FAO

Andrea Zisman
Department of Computing
City University
Northampton Square
London EC1V 0HB,UK
a.zisman@soi.city.ac.uk

John Chelsom, Niki Dinsey
CSW Informatics Ltd
4240 Nash Court
Oxford Business Park South
Oxford OX4 2RU, UK
john.chelsom@csw.co.uk
niki.dinsey@csw.co.uk

Stephen Katz, Fernando Servan
Food and Agriculture Organization
United Nations - WAICENT
Viale delle Terme di Caracalla
Rome 00100, Italy
stephen.katz@fao.org fernando.servan@fao.org

Abstract

In this paper we present our experience of using Web services to support interoperability of data sources at the Food and Agriculture Organization of the United Nations. We describe the information bus architecture based on Web services to assist with multilingual access of data stored in various data sources and dynamic report generation. The architecture preserves the autonomy of the participating data sources and allows evolution of the system by adding and removing data sources. In addition, due to the characteristics of Web services of hiding implementation details of the services, and therefore, being able to be used independently of the hardware or software platform in which it is implemented, the proposed architecture supports the problem of existing different technologies widespread in the FAO, and alleviates the difficulty of imposing a single technology throughout the organization. We discuss the benefits and drawbacks of our approach and the experience gained during the development of our architecture.
Keywords: *XML, Web services, J2EE, .NET, Ontologies, RDF, Topic Maps, WAICENT.*

1. Introduction

The development of distributed computing and networking has provided the technological basis for remote access to data and applications. The development of different systems has increased the utility of these systems, but has not solved the problem of having a large number of applications interoperating with each other. The applications have not been built to be integrated, and therefore, they normally define different data formats, have their own communication protocols, and are developed on different platforms. Interoperability of distributed systems is still a challenge.

Nowadays it is important to allow interoperability of different types of information sources in a large company or community. Users and applications have a growing need to access and manipulate data from a wide variety of information sources. However, the data sources are generally created and administered independently, differing physically and logically. Other difficulties associated with such an environment include: heterogeneity and autonomy of database systems, conflict identification and resolution, semantic representation of data, location and identification of relevant information, access and unification of remote data, query processing, and easy evolution of the system.

An example of the above problem is found in the Food and Agriculture Organization of the United Nations (FAO). FAO is a specialized agency of the United Nations, which leads international efforts to defeat hunger. It helps developing countries modernize and expand agriculture, forestry and fisheries and ensure good nutrition for all. One of its most important functions is to collect, analyze and disseminate information to assist governments to fight hunger and achieve food security. Towards this effort FAO has established the World Agricultural Information Centre (WAICENT) for agricultural information management and dissemination.

Within the WAICENT framework, a large amount of data, represented in various distinct formats, in many different languages, and handled by several metadata structures, are generated every day and stored in different types of data sources. However, there are no standards for representing languages, metadata, and specific country information. People need to access and manipulate data distributed in the various sources from both inside and outside the organization. It is important to share data between systems quickly and easily, without requiring the systems to be tightly coupled. In simple terms, the existing systems need to "talk" to each other. Another main problem is related to the fact that within the organization the use of two different technologies (Microsoft ASP [5] and Java JSP/servlets [20]) is widespread and it is, therefore, very difficult to impose a single technology throughout the FAO.

In this paper we present an approach based on Web services [17] and eXtensible Markup Language (XML) [6] technology to allow interoperability of the different data sources in the FAO. It is a lightweight approach and is based on the use of an *information bus* to allow exchanged of data between various information sources implemented by using different technologies. The *information bus* supports multilingual access of data stored in various data sources, handles metadata in a generic way, and enables

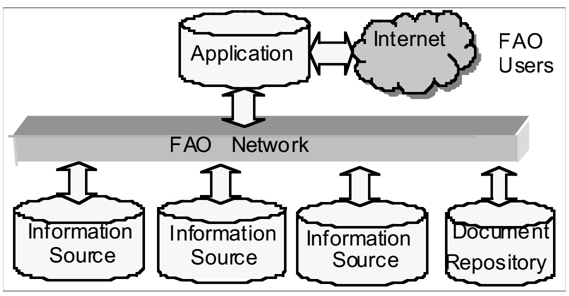

Figure 1. Existing information structure at FAO

metadata to be used as exchange models throughout FAO. The approach also supports dynamic report generation. A prototype tool has been implemented to demonstrate and evaluate the approach.

The remaining of the paper is organized as follows. Section 2 describes the problem in the FAO that is being tackled by our approach. Section 3 presents some related work. Section 4 outlines the *information bus* and the dynamic report generation. Section 5 illustrates our work through examples. Section 6 discusses the implementation of our prototype and evaluation issues. Finally, section 7 summarizes our experience and suggests directions for future work.

2. The problem

The Food and Agriculture Organization of the United Nations has approximately 200 systems supplying information for access on the World Wide Web, deployed using two different technologies: Microsoft ASP [5] and Java JSP/servlets [20]. These data sources need to share and exchange data between each other in an easy way. However, the use of the two technologies is already widespread in the organization and it is almost impossible to impose a single technology throughout the FAO. In addition, it is necessary to avoid rewriting of existing applications.

The existing information infrastructure is shown in Figure 1. It consists of information sources (database systems) containing different types of data including, but not limited to, different types of documents written in five official languages - English, French, Spanish, Chinese and Arabic (and some in Russian); electronic bibliography references; statistical data; maps and graphics; news and events from different countries; and web information.

Different people generate documents in different formats, which are inserted in the databases using web interfaces. The data is accessed from the databases in HTML format, through applications available on the Internet. Examples of these applications are WAICENT Information Finder (an online search tools), FAOBIB (an online catalogue of bibliography), FAO Virtual Library (a digital archive), and FAOSTAT (an online database about statistics of various areas). The FAO users are farmers, scientists, traders, government planners, and non-governmental people, both inside and outside the organization, that need to access and publish information.

Although the existing setting addresses some of the requirements of integrating disparate distributed systems, there are limitations involving budgetary or technical challenges, inflexibility, lack of standardization, and difficulty of scalability and extensibility. It is important to have a technology that is inexpensive, easy to implement, easy to maintain and based on open standards, to allow leverage of knowledge and existing resources without having to rewrite existing applications.

The technology needs to support interoperability of existing data sources and management of multilingual variants without changing the database structures. Currently, it is necessary to customize and add database structures for each different language. There is no standard way to manage language variants of documents or other data structures. This generates inconsistencies between applications in the way that they manage the different languages. In addition, the database models are not easily extensible when new data or language variants are added.

Other problems were related to the support of metadata representation and metadata exchange in a standard way, as well as use of standard ontology formats. In the FAO a document repository has been developed with the objective of storing and disseminating all publications electronically. It stores meeting notes, documents, metadata, and index data.

Different ASP interfaces have been created to allow searching the document repository by type, language, and subject. However, there is no standard way to manage language variants of documents or other data structures like specific country information and metadata. The multilingual Agricultural Thesaurus (AGROVOC) [2] from FAO has been applied to the web as a strategy to ensure some conformity in resource description/discovery. However, it falls short of being a complete tool for this purpose in view of a need for more specific subject terminology and richer ontological relations that are offered by traditional thesaurus.

3. Related work

The challenge of interoperating distributed systems, in particular database systems, has existed for a long time and has been extensively researched. Many approaches have been proposed to allow integration and interoperability of distributed systems developed in an independent way. These approaches have been proposed as outcomes of research work in both academia and industry.

We can divide the existing approaches into two main groups [34]. In the first group of approaches a global schema is used as another layer on the top of existing schemas which gives the users and applications the illusion of a single, centralized database system. Examples of these approaches include systems like DATAPLEX [9], DDTS [13], MULTIBASE [29], and PEGASUS [3]. However, the construction of a global schema is not a simple task, does not guarantee the autonomy of the participating database systems, and does not allow easy evolution of the system in terms of adding and removing of participating databases.

In order to overcome the problem of constructing a global integrated schema the second group of approaches has been proposed, in which 'partial' or 'no integration' is performed. Examples of these approaches include the federated architecture [19], five-level schema architecture [28], multidatabase architecture [22][23], the Jupiter system [18], and [33]. Within the approaches that do not use a global schema some of them proposed the use of mediators and wrappers. In these approaches data sources are encapsulated to make it usable in a more convenient manner by hiding or exposing the internal interface of the data sources, reformat data, and translate queries. Examples of systems that use wrappers and mediators are DIOM [24], DISCO [31], Garlic [27], and TSIMMIS [16].

In any of the above approaches and existing technologies the problems related to how to format data to be exchanged and how to transmit the data are still open problems. Regarding data format, there are almost no tools that can automate the process of translating data in different formats. Many systems use ASCII-based text files to represent their data. However, there is no standard way of formatting or describing the values in the files. The different systems exchanging data in ASCII format must have custom-built loading software to handle different file formats. Other systems exchange data via a specified file format, which does not scale well (e.g. Microsoft Excel).

On the other hand, data transmission has also been difficult to implement. The use of the File Transfer Protocol (FTP) facilitates file transfer, but this is not a tight, object-oriented approach to exchanging data. Electronic Data Interchange (EDI) has also been used for exchanging data. However, EDI is rigid, complex, and expensive to implement. More recently some technologies have been proposed to allow a more object-oriented and less expensive approach, based on Remote Procedure Calls. Examples of these approaches are DCOM [12] and CORBA/IIOP [10]. The problems with these technologies are that they are platform specific, do not easily integrate, and pose network security risks due to the requirement of having open ports to accommodate messages.

The existing approaches have contributed to alleviate the problems of sharing data between autonomous and heterogeneous data sources. However, the development of Web services [17], SOAP [8] and XML technologies support the problems of e-business by allowing the ability of representing data structures and describing these structures in an easy way to implement and maintain. In the next section we describe an approach that uses Web services.

4. The approach

In order to tackle the problems described in Section 2 we proposed a lightweight approach based on Web services and related XML technologies. The approach was developed in a way that can be implemented on multiple vendor platforms, with minimal effort and disruption to existing systems.

The main goal of the approach is to create an environment where new web-based information systems can be developed quickly and easily, using any technology platform, by accessing information from any of the existing 200 information systems at the FAO, and supporting the multilingual characteristics of the institution in which documents are expressed in five official languages as well as Russian and other local variations. Other objectives included the implementation of dynamic report generator and development of an XML document repository to handle metadata and language variants in a generic way.

In the next subsections we describe the *information bus* approach proposed to support data exchange and dynamic report generation.

4.1. Information bus

Figure 2 presents an overview of the architecture of the *information bus* being proposed to support interoperability of various information sources. The approach consists of wrapping the various data sources with Web service interfaces in which information inputs and outputs are passed as XML structures.

The concept of the *information bus* is that all data passed through it is represented in standard XML formats. These formats can be imposed in a regulated fashion by publishing the XML schemas being used and validating instances of messages. Regardless of the formats used by the existing systems, the same XML syntax is used for input and output parameters on the Web services. For example, all data related to country, language or currency is represented in a single XML format, which uses (a) ISO 3166 country code (3 letter), (b) ISO 639-1 language code (2 letter), (c) ISO 4217 currency code, respectively. With Web services it is not necessary to re-engineer existing systems to new XML standards. However, it is necessary to enforce XML standards in the Web services interfaces. For example, the parameters for operations involving language codes always use the 2-character ISO 639-1 code.

The Web services were developed for systems containing information about statistics, documents, maps, news and events. These systems are:

- internal to the FAO, for which the development team had access to the application source code,
- internal to the FAO, but the development team had no access to the application source code, and
- external to the FAO.

The management of information, including handling of multilingual variants is also based on XML. We propose to move structured information out of database fields and represent them in XML documents to allow a more generic model, which is easier to administer and to extend to new languages (e.g. there is a growing need to support Russian, in addi-

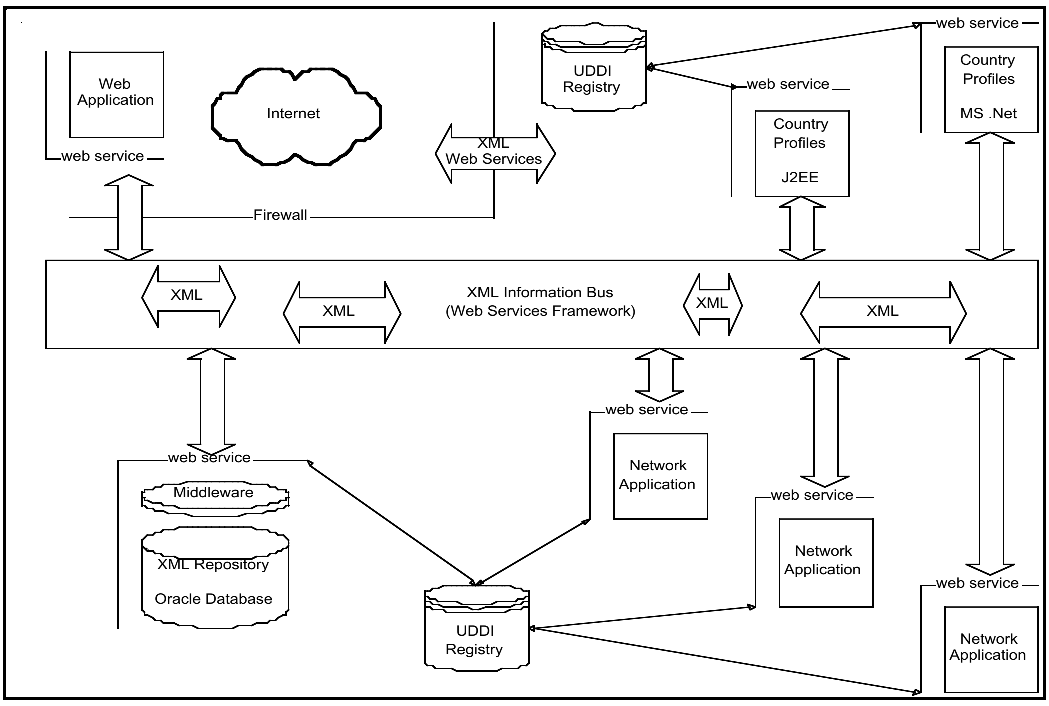

Figure 2. Information bus architecture

tion to the five existing official languages). Whereas existing systems use own (non-standard) database structures to model multilingual data, the XML approach provides a generic way to manage structured information to any schema. The XML documents are stored in an XML repository, as shown in Figure 2.

It is foreseen with the project proposal for the handling of metadata also to be based on XML and stored in the XML repository, which can be used as the exchange model throughout the FAO. The metadata would be represented as RDF [21], RDF Schema [7], Dublin Core elements version 1.1 [11], and XML Topic Maps [26]. RDF would be used to specify metadata on resources, i.e. values of properties for the resources. RDF Schema would be used to define classes of resources and the properties that instances of each class can take. In addition, RDF Schema, Dublin Core, and XML Topic Maps would be used to define ontologies, which capture the relationship between classes, resources, and properties that compose a vocabulary. XML Schemas [14] would also be used to define vocabulary range of values contained in a property.

The assignment of constraint metadata would be based on standard ontologies published or developed in-house, also represented in XML. This would facilitate importing and exporting of all XML metadata held in participating systems.

The XML repository stores resources (documents) in a relational database, using a Java interface based on an extended version of the XML:DB API [32] that caters for document variants (e.g. different language variants of the same document) and metadata associated with documents. The repository is also wrapped as a Web service to allow access of documents by metadata and/or language.

The FAO currently has a web application called FAO Country Profiles [15], which draws information from a variety of systems on the internal network and presents an aggregated view, sorted by country. Within each country profile, information is structured according to the main functional areas of the FAO - sustainable development, economic situation, agriculture sector, forestry sector, fishery sector, technical cooperation. We developed an application for the Country Profiles using the *information bus* architecture (see Section 5 for an example).

The architecture can contain two Universal Discovery, Description, and Integration (UDDI) registries to support discovery of information. One UDDI registry is internal to the FAO and assists with share and exchange of information between the data sources internal to the organization. The other UDDI registry is used to support share and exchange of data between the data sources external to FAO. In the initial deployment of the architecture, only the internal registry was active.

An example of the XML structure passed in the *information bus* is shown in Figure 3. It consists of a

```
<soap:Envelope
 xmlns:xsi="http://www.w3.org/2001/
                              XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/
                              XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/soap/
                              envelope/">
 <soap:Body>
  <Query xmlns="http://tempuri.org/">
    <Country>SEN</Country>
    <Language>EN</Language>
   <Keyword>Forestry</Keyword>
  </Query>
 </soap:Body>
</soap:Envelope>
```

Figure 3. Example of XML structure passed in the information bus

SOAP [8] message enriched with metadata from ontologies represented in RDF [21]. In this example, the XML structure represents a query about documents containing information of *forestry* (Keyword), in *Senegal* (Country – *SEN*), written in *English* (Language – *EN*). The transformation from the standard XML representation used in as the input parameters of the Web service, to the native input parameters of the system is implemented in the Web Service code itself. This is achieved using mapping structures from the native input parameters of the application (strings, integers) to the ISO representations outlined in the *information bus*.

In the approach, we propose to use three different types of Web services based on their functionality, named: *support*, *relevance*, and *content*. The support Web service type contains utilities to return standard representations of countries, metadata categories, and language translations. An example of the information returned by the support service is shown in Figure 4. The relevance Web service type is used to identify the Web service that is related to a particular application context and the setting of parameters necessary to call the identified Web service, as illustrated in Figure 5. In this example Web service with ID 900 contains description of general maps and should be accessed by using parameters such as *Country*, *Language*, and *Category*.

The content Web services type is invoked to return XML content from existing information sources, through Web services interfaces with parameters for language, country, subject, and others. Figure 6 shows an example of the content service returned from the BBC News Online information source (external to the FAO).

4.2. Country Profile report

Our approach also supports dynamic report generation based on data extracted from the various information sources. The reports are assembled as XML

```
<FSCollectionChoices
xmlns="http://tempuri.org/CollectionChoices.xsd">
 <Country diffgr:id="Country231"
          msdata:rowOrder="230">
   <COUNTRY>Zimbabwe</COUNTRY>
   <FS_COUNTRYCODE>181
          </FS_COUNTRYCODE>
   <ISOCODE>ZWE</ISOCODE>
 </Country>
 <Item diffgr:id="Item1"    msdata:rowOrder="0">
   <ITEM>Abaca (Manila Hemp)</ITEM>
   <FS_ITEMCODE>809</FS_ITEMCODE>
 </Item>
 <Element diffgr:id="Element1"
          msdata:rowOrder="0">
   <ELEMENT>Seed</ELEMENT>
   <FS_ELEMENTCODE>111
      </FS_ELEMENTCODE>
 </Element>
 <Year diffgr:id="Year1"   msdata:rowOrder="0"
       diffgr:hasChanges="inserted">
   <YEAR>1961</YEAR>
 </Year>
</FSCollectionChoices>
```


Figure 4. Example of information returned from support service


```
<ServiceDetails
 xmlns:xsi="http://www.w3.org/2001/
                         XMLSchema-instance"
 xmlns="http://tempuri.org/ServiceDetails.xsd">
 <ServiceDetail
     d2p1:ServiceName="GeneralMaps"
     d2p1:ServiceID="900"
xmlns:d2p1="http://tempuri.org/ServiceDetails.xsd">
   <ServiceDescription>
     Description for General Maps
   </ServiceDescription>
   <Param d2p1:name="Country">
       <value>GBR</value>
   </Param>
   <Param d2p1:name="Language">
       <value>EN</value>
   </Param>
   <Param d2p1:name="Category">
       <value>16</value>
       <value>19</value>
   </Param>
 </ServiceDetail>
</ServiceDetails>
```


Figure 5. Example of information returned from relevance service

and rendered as PDF, by using XML Stylesheet Language: Formatting Objects – XSL:FO [1] and the open source FO Processor from Apache [4]. The reports are generated based on information content selected by the user.

When the user chooses a country and language from the support Web services this sets the state of


```
<BBCNewsDS
 xmlns ="http://www.fao.org/waicent/cpmis/
                              BBCNewsDS.xsd">
 <BBCNews>
   <headline>Blair blasts green pacesetters
   </headline>
   <intro>In 1997 Labour undertook to be the
&amp;#34;first truly green government&amp;#34;,
but has that promise been fulfilled?</intro>
   <newsdate>23/10/2000</newsdate>
   <link>http://news.bbc.co.uk/hi/english/sci/tech/
         newsid_987000/987400.stm</link>
 </BBCNews>
 <BBCNews>
   <headline>Labour: A green government?
   </headline>
   <intro>In 1997 Labour undertook to be the
&amp;#34;first truly green government&amp;#34;,
but has that promise been fulfilled?</intro>
   <newsdate>23/10/2000</newsdate>
   <link>http://news.bbc.co.uk/hi/english/sci/tech/
         newsid_986000/986532.stm</link>
 </BBCNews>
</BBCNewsDS>
```


Figure 6. Example of information returned from content service

the client and the relevance Web service is used to define the information available to the user in that context. Then when the user chooses to generate a dynamic report they are presented with the option to invoke different Web services, depending on the context. These Web services create the different sections of the report, according to the preference of the user.

Once the user has chosen the services to invoke in the creation of the report, the report generator calls all the Web services simultaneously using multi-threading. The report is built in memory in an order that depends on which Web service returns results first; the final report, in the correct order is compiled and generated once the last Web service returns results. The whole process takes approximately 60 seconds from invoking the services to report generation; a normal report will involve between 30 and 50 different Web services.

5. Example

In this section we present an example of the Country Profiles application [15] used as a case study for our approach. Country Profiles is an application in the FAO that allows access to country-specific information without the need to search individual databases and systems. It is an information retrieval tool that groups in a single area the vast amount of information available at FAO based on the global activities in agriculture and development, and classifies the information by country. The application uses three categories to group information:

1) FAO's areas of expertise - sustainable development, economy, agriculture, fisheries, forestry, and technical cooperation,
2) FAO's priority areas for interdisciplinary action (PAIA)– ranging from biological diversity to trade in agriculture, fisheries, and forestry, and
3) AGROVOC - a metadata ontology with over 4000 terms breaking down the first two metadata categories to a lower level (i.e. Cattle Breeding). AGROVOC is mainly used in the Library applications at FAO.

We have developed the Country Profiles application by using Web service technology. Figure 7 presents the web page used as the interface to the application. In Figure 7 it is possible to see all the different services used in the application.

Firstly the three dropdown lists under the banner at the top of the page are invoked from the support Web services described above. These set the state of the application and are currently set to English (EN), Afghanistan (AFG) and FAO's Fields of Expertise for the metadata. The categories to the left of the page (General Information) are also populated from the same metadata support Web service. Slightly below is the fourth dropdown list, this is populated using the relevance service, which takes inputs from the above three services and generates a list of available Web services which meet the current state of the application. It also contains the exact parameters to be sent to each content service when the user chooses an application (see Figure 6 for an example). Finally in the main body of the screen you can see an example of an invoked content Web service, in this example the service returns News information about the selected country from a system names EIMS.

Figure 8 shows examples of another content service being invoked (in this case the information in derived from the General Mapping application). Images are received in the body of the XML response as Base64 encoded string, which is decoded and cached on the client application server for faster retrieval. Legend text is also sent with the encoded string and the colors are generated using hash codes (e.g. #FFFFCC). The second screenshot in figure 8 shows a generated report in PDF format (see subsection 4.2).

6. Implementation aspects and evaluation

An operational prototype tool has been implemented in the period of three months in order to demonstrate and evaluate the approach. The case study used in the development was the Country Profiles application, as illustrated in Section 5. The prototype allows (a) generic XML-based information infrastructure to support multilingual information in an extensible and standard way, (b) application integration structure based on Web services to allow interoperability of FAO systems and information sources for delivery through web portals, (c) use of Microsoft .NET [25], (d) standard XML representations for handling metadata and multilingual documents, and (e) dynamic country profiles report generation. In addition, the prototype has also demonstrated the ability to combine information in multiple languages together in the same pages, to develop new web-

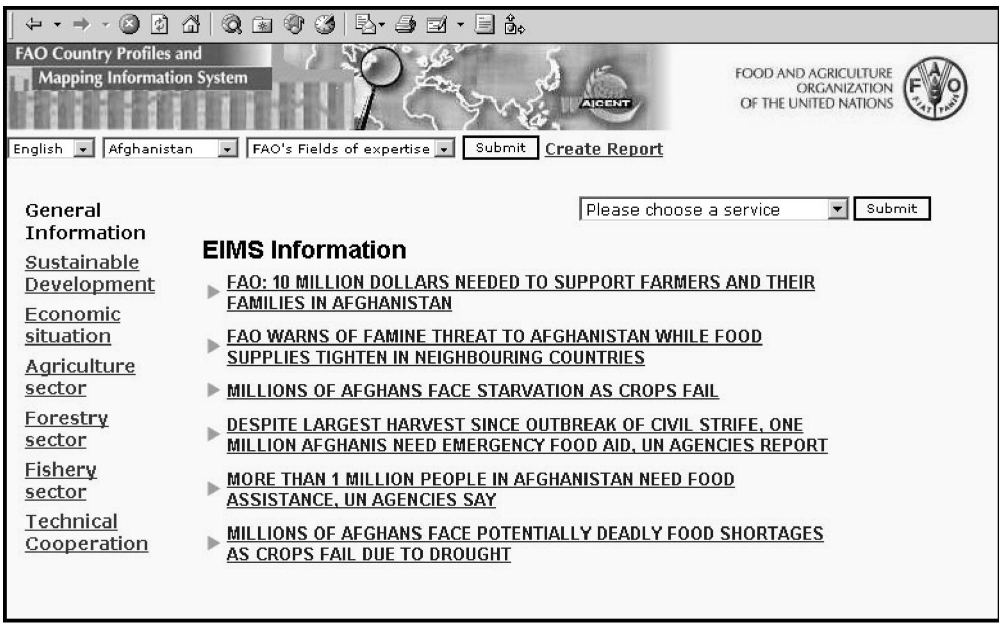

Figure 7. Example of the web page for the Country Profile application

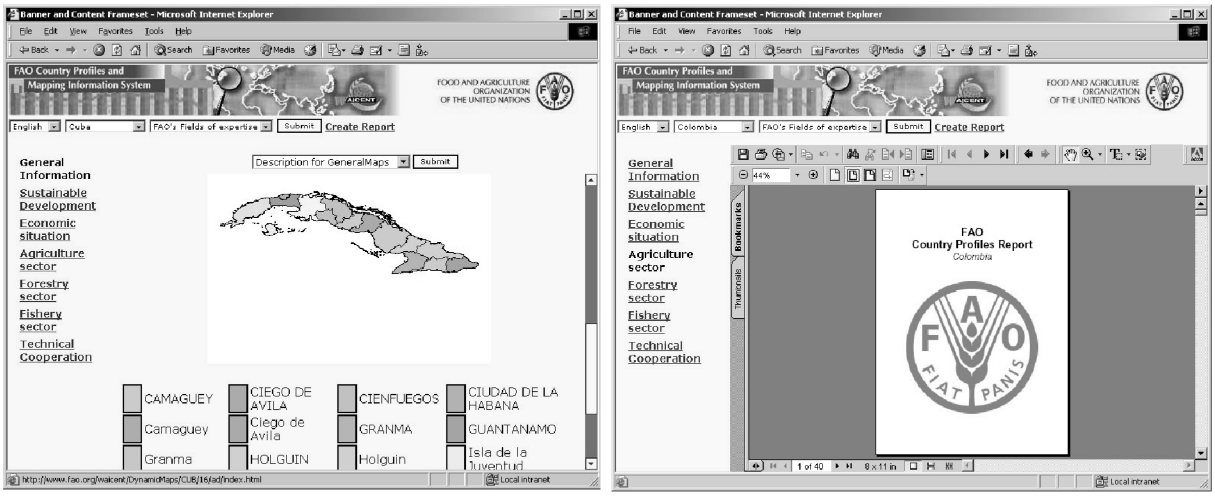

Figure 8. Screenshots of the Country Profile application

based information systems quickly and easily, and to combine and compare statistics from different countries.

The main activities in the development of the prototype consist of the (a) creation of Web services wrappers to existing data sources so that they could be accessed by using the XML *information bus*, (b) implementation of a new XML document repository that allows structured data to be stored for different languages in a generic and extendible manner, and (c) implementation of the Country Profiles application.

The Web services wrappers were created using Microsoft .NET for the following systems (those already accessible from the web are listed with their URL):

- Statistics
  a) FAOSTAT - an internal FAO statistics system (http://apps.fao.org) and
  b) World Bank Statistics – an external system
- Documents:
  a) FAOBib - an existing internal FAO bibliography system (http://www4.fao.org/faobib)
- EIMS (Electronic Information Management System) - an existing internal FAO repository of full text documents
- RAP - a new internal XML document and metadata repository
- Maps
  a) General Maps – an internal FAO application of maps
  b) GeoNetwork – an internal FAO geographic application (http://www.fao.org/geonetwork)
- News and Events
  a) NEMS (News and Events Management System) - an internal FAO news application
  b) BBC News Online - an external new service

For EIMS, RAP, and NEMS systems the development team had access to the applications source code and respective databases. For FAOBib, GeoNetwork, General Maps, and BBC News the development team did not have access to the source code and the information was accessed by HTTP on scrape HTML data. For FAOSTAT and World Bank Statistics the access to the data was through batch and cache.

Our experience has been very positive. We have found that it was easy to develop the wrappers around the data sources. Some of the activities have been implemented in hours, instead of days, as it was previously thought.

A major advantage of using Microsoft .NET framework was the ease with which Web services wrappers could by created. However, the integration of these Web services with the J2EE platform had some problems due to the difference in handling complex data types and inconsistencies in the use of Web Services Description Language (WSDL).

One problem is related to the fact that .NET uses Document-style Web services by default, whereas the J2EE implementation (Apache Axis) uses RPC-style invocation. To alleviate this problem in .NET we used the SoapRpcService() [30] property to indicate that the .NET Web service was RPC-style. However, there were further problems because Axis did not yet implement support for multi-dimensional arrays or for generating complex type definitions in WSDL, which were created automatically by .NET. To alleviate these problems, and to allow developers to create Web services quickly and easily from existing Microsoft applications (of great importance to FAO, to encourage all departments to make their applications available as Web services), a second tier of Web services was created that automatically made the transformation from the data types generated by .NET to XML arrays that could be used by both .NET or J2EE Web services.

The prototype has shown that it is possible to integrate different information sources (internal and external to the organisation) by preserving their autonomy and that the system can evolve in an easy way by adding and removing data sources. In addition, it has also demonstrated that it is possible to avoid the problem of imposing the use of a single technology in an organization like the FAO. The Web services framework used in our approach allows a platform that is stable, flexible, extensible, and high performance.

The work presented in this paper has provided new opportunities for the FAO. Examples of these opportunities include, but are not limited to, standardization on the way information is shared within FAO and with external parties, provision of new information services within FAO (e.g. ontologies, statistics presentations), provision of an environment to efficiently develop, deploy, and maintain new information services, leading of a next generation information dissemination methods to assist with the aims of the FAO.

7. Conclusion and future work

In this paper we have presented an *information bus* approach to interoperate different data sources in the Food and Agriculture Organization of the United Nations. The data sources are both internal and external to the FAO and are used to help developing countries to modernize and expand agriculture, forestry and fisheries by collecting, analyzing and disseminating information which can be used to fight hunger and achieve food security. The different data sources contain information related to different types of documents written in five official languages, statistical data, electronic bibliography references, maps and graphics, and news and events.

The approach is lightweight and based on Web services and XML technologies. It preserves the autonomy of the existing systems and allows evolution by adding and removing data sources. The approach allows the creation of an environment where new web-based information systems can be developed quickly and easily, supports the multilingual characteristics of an institution like FAO, provides dynamic report generation, and handles metadata and language variants in a generic way.

Before development of a full implementation of the prototype throughout the FAO, we are extending the prototype to support XML configuration files, generic report configuration tailored by subject area, ontology services in which based on a data item a list of related data is identified, and public Web services for ontologies, countries, and codes. We are investigating the application of Dublin Core to represent metadata information and, therefore, extending its current use on supporting metadata associated with documents in the XML repository. We also plan to implement the Country Profiles application by using J2EE Web services technology and compare this technology with Microsoft .NET.

Acknowledgement

We would like to thank all members of the WAICENT/FAOINFO Dissemination Management Branch team at the Food and Agriculture Organization of the United Nations. Giorgio Lanzarone, who acted as the project coordinator in FAO, and Marta Iglesias, Nick Waltham and Anne Aubert for participating in the requirements specification phase and developing the Web services wrappers in the data sources of FAO. We would also like to thank some members of CSW Informatics. Mavis Courname for participating in the requirements specification phase and Stephen Horwood responsible for the implementation of the country profiles applications, Web services client wrappers, and XML repository.

References

[1] Adler, S., Berglund, A., Caruso, J., et al., 2001. Extensible Stylesheet language, Version 1.0, W3C Recommendation, October 2001, http://www.w3.org/TR/2001/REC-xsl-20011015.

[2] AGROVOC. http://www.fao.org/agrovoc.

[3] Ahmed, R., Albert, J., Du, W., Kent, W., Litwin, W. and Shan, M.C., 1993. An Overview of Pegasus. *In the 3rd International Workshop on Research Issues and Data Engineering: Interoperability in Multidatabase Systems,* pages 273-277, Vienna, Austria, April 1993. IEEE Computer Society Press.

[4] Apache. Apache Project. http://www.apache.org.

[5] ASP. Microsoft Active Server Pages. http://www.microsoft.com/asp.

[6] Bray, T., Paoli, J., Sperberg-McQueen, C.M. and Maler, E., 2000. Extensible Markup Language (XML) 1.0, Second Edition, W3C Recommendation, October 2000. http://www.w3.org/TR/2000/REC-xml-20001006.

[7] Brickley, D. and Guha, R.V., 2002. Resource Description Framework (RDF) Schema Specification 1.0, March 2002. http://www.w3.org/TR/rdf-schema.

[8] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H., Thatte, S. and Winer, D. Simple Object Access Protocol (SOAP) 1.1. http://www.w3.org/TR/SOAP.

[9] Chung, C., 1990. DATAPLEX: An Access to Heterogeneous Distributed Databases. *Communications of the ACM*, 33(1): 70-80, January 1990.

[10] CORBA/IIOP. Common Object Request Broker Architecture. http://www.omg.org/technology/documents/formal/corba-iiop.htm.

[11] Dublin Core. http://dublincore.org.

[12] DCOM. Microsoft Distributed Component Object Model, http://www.microsoft.com/ com/tech/DCOM.

[13] Dwyer, P.A. and Larson, J.A., 1987. Some Experiences with a Distributed Database Testbed System. In *Proceedings of the IEEE*, volume 75, pages 633-648, May 1987.

[14] Fallside, D.C., 2000. XML Schema Part 0: Primer. http://www.w3.org/TR/2000/WD-xmlschema-0-20000407.

[15] FAO. FAO Country Profiles and Mapping Information System. http://www.fao.org/countryprofiles/.

[16] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Vassalos, V. and Widom, J., 1995. The TSIMMIS Approach to Mediation: Data Models and Languages. In *Next Generation Information Technologies and Systems (NGITS)*, Naharia, Israel, June 1995.

[17] Graham, S., Simeonov, S., Boubez, T., Davis, D., Daniels, G., Nakamura, Y. and Neyama, R., 2002. Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI. SAMS Publishing, 2002.

[18] Grimson, J. and Murphy, J., 1995. The Jupiter Approach to Interoperability with Healthcare Legacy Systems. In R.A. Greenes, H.E. Peterson, and D.J. Protti, editors, *MEDINFO 95*, pages 367-371, IMIA 1995.

[19] Heimbigner, D. and McLeod, D., 1985. A Federated Architecture for Information Management. ACM Transaction on Office Information Systems, 3(3): 253-278, July 1985.

[20] Java. Java JSP/Servlet. http://java.sun.com.

[21] Lassila, O. and Swick, R.R., 1999. Resource Description Framework (RDF) Model and Syntax Specification. February, 1999. http://www.w3.org/TR/REC-rdf-syntax.

[22] Litwin, W., 1988. From Database Systems to Multidatabase Systems: Why and How. In W.A. Gray, editor, *In Proceedings of the 6th British National Conference on Databases (BNCOD 6)*, British Computer Society Workshop Series, pages 161-188, July 1988.

[23] Litwin, W., Mark, L. and Roussopoulos, N., 1990. Interoperability of Multiple Autonomous Databases. ACM Computing Surveys, 22(3): 267-293, September 1990.

[24] Liu, L. and Pu, C., 1997. Dynamic Query Processing in DIOM. Bulletin of Technical Committee of Data Engineering, 20(3): 30-37, September 1997.

[25] .Net. Micosoft ASP .Net. http://www.asp.net.

[26] Pepper, S. and Moore, G. XML Topic Maps (XTM) 1.0. http://www.topicmaps.org/xtm/1.0.

[27] Roth, M.T. and Schwatz, P., 1997. Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. *In Proceedings of the 23rd International Conference on Very Large Database Bases*, Athens, Greece, August 1997.

[28] Sheth, A.P. and Larson, J.A., 1990. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3): 183-236, September 1990.

[29] Smith, J.M., Bernstein, P.A., Dayal, U., Goodman, N., Landers, T., Lin, K.W.T. and Wong, E, 1981. Multibase – Integrating Heterogeneous Distributed Database Systems. In *National Computer Conference*, volume 50 of *AFIPS Conference Proceedings*, pages 487-499, 1981.

[30] SOAP-RPC. http://www.soaprpc.com.

[31] Tomasic, A., Raschid, L. and Valdueriz, P., 1996. Scaling Heterogeneous Databases and The Design of Disco. In *Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS)*, pages 449-457, Hog Kong, May 1996. IEEE Computer Society Press.

[32] XML:DB. http://www.xmldb.org.

[33] Zisman, A. and Kramer, J., 1999. An Approach to Interoperation between Autonomous Database Systems. *Distributed Systems Engineering Journal*, 6, 1999.

[34] Zisman, A., 1998. Information Discovery for Interoperable Autonomous Database Systems. PhD Thesis, Department of Computing, Imperial College of Science Technology and Medicine, University of London, UK, 1998 (http://www.soi.city.ac.uk/~zisman).