

Distr.  
GENERAL

ECE/CES/SEM.54/8  
17 May 2006

ENGLISH ONLY

**UNITED NATIONS STATISTICAL COMMISSION and  
ECONOMIC COMMISSION FOR EUROPE  
CONFERENCE OF EUROPEAN STATISTICIANS**

**EUROPEAN COMMISSION  
STATISTICAL OFFICE OF THE  
EUROPEAN COMMUNITIES (EUROSTAT)**

**ORGANISATION FOR ECONOMIC COOPERATION  
AND DEVELOPMENT (OECD)  
STATISTICS DIRECTORATE**

Joint UNECE/Eurostat/OECD Seminar on the Management of Statistical Information Systems (MSIS)  
Sofia, Bulgaria, 21-23 June 2006

Topic (i): Changes in statistical processes

## **APPLYING WEB SERVICES TO GOVERNMENTAL ORGANISATIONS: THE CASE OF THE NATIONAL STATISTICAL SERVICE OF GREECE**

**Supporting paper prepared by the National Statistical Service of Greece and  
the Research Academic Computer Technology Institute, Greece**

### **Summary**

**Abstract:** *This work presents the development of a prototype module for the National Statistical Service of Greece in order to enable enterprises to use and integrate it with their commercial and accounting packages. The benefits arising from the use of the system will be the improvement of the system and reduction of the burden on the respondents, especially for providers with large amounts of declarations, as well as an overall improvement of data quality. We have implemented the prototype module as a Web Service API, since Web Services allow different applications from different sources to exchange data without the need of implementing custom codes and applications for this purpose. Due to the fact that all communication uses XML, the Web Service is not tied to any operating system or programming language. Additionally, Web services are being adopted in the marketplace as a mechanism for efficient process integration in the organizations within and across organizational boundaries.*

## **I. INTRODUCTION**

1. Web services are the new breed of Web applications. The Web services technology has already been established by the World Wide Web consortium<sup>1</sup>. Web Services [2] are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. In other words, web services are interoperable building blocks for constructing applications. As an example, we can imagine a distributed digital library infrastructure built on web

---

<sup>1</sup> [www.w3c.org](http://www.w3c.org)

services providing functionality such as distributed search, authentication, inter-library loan requests, document translation and payment. These web services would be combined by a particular digital library application to offer an environment for reaching information resources that is tailored to its particular user community. While the web enables users to connect to applications, the web services architecture enables applications to connect to other applications. Web services is therefore a key technology in enabling business models to move from B2C (Business to Consumer) to B2B (Business to Business).[1]

2. From a technological perspective the Web Services framework is the core for building well-formed and efficient Web Services. It consists of essentially three basic components: The Web Service Description Language (WSDL), a language to allow formal functional characterization of the provided functionalities, the Simple object Access Protocol (SOAP), a protocol that defines the format of the information interchange, and the Universal Description, Discovery and integration (UDDI), which is a catalogue of Web Service Descriptions. The Web Services framework will be discussed further in the technology section.

3. In this paper, we present a prototype module implemented for the National Statistical Service of Greece (NSSG) in order to enable enterprises to use and integrate it with their commercial and accounting packages. The NSSG Web Service embraces emerging standards for describing (WSDL), discovering (UDDI), and invoking (SOAP) Web services.

4. The NSSG Web Service is built around a collection of features that include a framework for composing services for online submitting and managing INTRASTAT declarations, optimized querying services, and preserving privacy.

5. The remainder of this paper is organized as follows. Section II gives the motivation that led to the development of the module. It shows the feasibility study that took place prior to the development presenting the drawbacks of current system and what the Web Service aims to do. Section III presents the technology background (Web Service Framework and Functional Model) while in Section IV we describe the architecture of the prototype module. Section V refers to the implementation of the case study and Section VI continues with its evaluation and the expected results. Finally, Section VII gives our concluding remarks.

## **II. MOTIVATION**

6. The monthly collection of INTRASTAT declarations by the enterprises is one of the most important, heavy and time-consuming operations performed in the NSSG.

7. Until just a few years ago, INTRASTAT data were submitted by the enterprises as an “additional” module of their VAT declarations (paper-based), then these completed forms were collected by the Tax Authorities, on behalf of NSSG and they were finally sent to the statistical institute for further processing (data entry, error correction, data analysis, production of final statistical data and dissemination of results). This whole process was extremely time consuming and error-prone.

8. The continuously increasing workload, the important delays in the process and of course, the understanding of the burden put on the enterprises for accurate and timely data reporting has guided NSSG towards the development of a pilot web-based application for online data collection directly from the enterprises.

9. More specifically, NSSG offers the ability to enterprises to declare their products (imports, exports) via a web-based system for Intrastat statements' declaration. This electronic system has been operating successfully since January 2003 and it serves 32,398 registered enterprises and 3,356 accountants / accountant offices that have declared in total 366,017 statements. The most important characteristics of the system are the following:

- Use of SSL protocol for all tasks that concern control or treatment of personal elements
- Suppression of popup windows in all the levels of application
- Increment of records' limit for manual input up to 1.000 records per statement
- Increment of records' limit for file upload up to 50.000 records per statement
- Projection of statements' content up to 1.000 records with printing facility
- Simplification of customers' management for the accountants with global login
- Incorporation of confirmation processes in order to avoid common users mistakes

10. A company or its accountant has the ability to declare products, to modify declarations of products, to delete declarations of products.

11. The wide acceptance of this application by its users, their continuous demand for further improvements and the fact that most enterprises have access to Internet and use ready-made accounting software packages, in combination with the latest advances in the field of IT and communications technology has guided the system development further, in order to automate as much as possible the data collection process and make it practically "invisible" for the respondents, thus reducing the burden posed on them.

12. The National Statistical Service of Greece wanted to offer a new intelligent way of electronic declarations and decided to use web services (the Intrastat web service) for this implementation. This web service consists of several methods that have the same functionalities as the electronic system of Intrastat statements' declaration.

13. Within this framework, NSSG in collaboration with RA-CTI, which undertook the task of the S/W development, planned and designed the development of a prototype web service module for integrating the most popular commercial and accounting packages used by enterprises and improving the electronic data collection method, moving from the "push" towards the "pull" model. This action consisted of:

- A Feasibility study consisting of a market research to track down the most commonly used packages, visiting the software houses, defining the variables used for the Intrastat declaration, make the format of data, the validation checks needed for quality data etc.
- The development of a prototype module that enables the extraction and submission via the web of data directly from the commercial and accounting packages.
- The integration of the collection system with production system. Till now the uploading of data from the data base of the collection system (SQL Server) to the database of the production system (Oracle ver. 8i) is made manually (Extraction of a ASCII flat file from the collection data base and then upload to the production database using a loading form).

14. In brief, the main aim of the whole project and the idea underlying it was to improve the quality of the process in different parameters, namely:

- Reduction of the burden on the respondents, especially for providers with large amounts of declarations, and
- Improvement of the quality of the final statistical product (statistical data).

15. It is widely known that the quality of statistical data depends heavily on different facets of the statistical data production and especially the phase of data collection has a tremendous impact on it. Consequently, the minimisation of the errors occurring during the data collection phase (data entry errors), the elimination of logical errors through checks and controls embedded in the collection system, together with the minimisation of the time between collection and production and the ability to use readily available data directly from the source (IT systems of the respondent) are expected to improve significantly the accuracy and quality of the data transmitted to Eurostat and to all NSSG data users.

16. The feasibility study implemented at an early stage of the project demonstrated the benefit of using web services as the entry point for data upload, as well as the need of companies to do this procedure in the easiest way. The users of the service were involved in the process from an early stage, since they participated in the feasibility study by providing answers to the feasibility questionnaire transmitted to all potential users. The questionnaire (e-mail) gauged the level of interest among them in using this new way for INTRASTAT declarations. In the next step, the replies were analysed, in order to highlight their needs, identify existing problems and track down the most commonly used software packages. The number of companies that replied to our e-mails was 141, most of which were medium-to-big enterprises and thus “heavy” users of the service.

17. In appendix B an analysis of the received responses is presented.

### **III. TECHNOLOGY BACKGROUND**

18. The prototype module has been developed as a Web Service API, because Web Services allow different applications from different sources to exchange data without time-consuming custom coding. Additionally, since data transfer uses XML [3], Web services are not tied to any one operating system or programming language. The National Statistical Service of Greece Web Service API uses the SOAP and WSDL standards so enterprises are able to use and integrate it with the any commercial and accounting packages.

19. A web service is a software application on a network that has an interface through which other programs can gain access. They are currently being used to help large and small entities get the most from their IT resources by allowing the integration of diverse software applications, from desktop programs to large enterprise-wide systems. Not only is this useful for day-to-day operations, it is especially helpful in integrating systems after a merger or acquisition.

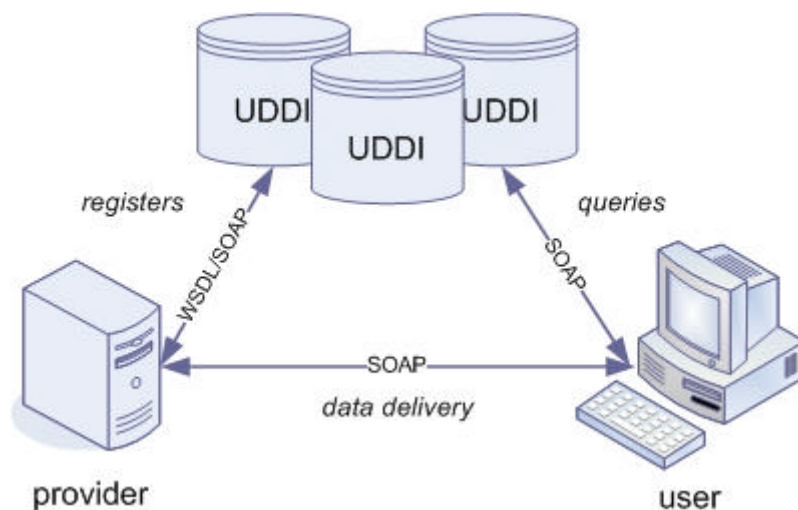
20. The key to making web services work is data, process, and communication standards. The communication protocol standard is the same as the Internet, TCP/IP. The data standard is XML, a set of syntax rules for adding meaning to data and for building other XML standards. The process standards are actually a set of evolving XML standards: SOAP (Simple Object Access Protocol) [4], for packaging messages from one software application to another, WSDL (Web Services Description Language) [5][6], for describing the web services processes in terms a software application can understand; and UDDI (Universal Description, Discovery and Integration) [7][8], for describing how to find and use an available web service.

21. The actual function of web services can be described from a provider's and a user's perspective. From a provider's perspective, a web service is created by using the data, process, and communication standards identified above to create a web interface to one or more software applications. Most of the web services described above provide data from a database in response to specific request parameters. In essence, a web service responds to a "get data" command by reading

the data from a database and sending it back to a software application on the Internet. To actually create such a web service, the provider uses WSDL to define the allowable read access "get data" commands that the database management software can understand. The web service also knows how to put the results in a SOAP envelope addressed to the requesting software application and how to send it via the Internet.

22. From a user's perspective, a software application must be able to issue the appropriate commands, put them in a SOAP envelope, and send them to the web service interface for processing. This usually requires downloading the WSDL and plugging it into a software application.

23. Figure 1 depicts the typical Web Services Functional Model. This model consists of three entities: the *provider*, the *user* that consumes the service and the *registry*. The provider registers its services to the UDDI registry using SOAP/WSDL and waits for service requests. A user process queries the registry for a Web Service, using SOAP messages and retrieves information again as SOAP messages.



**Figure 1. The Web Services Functional Model**

24. Implementing a Web Service for data interchange between two applications provides several advantages to the whole process. For example web services:

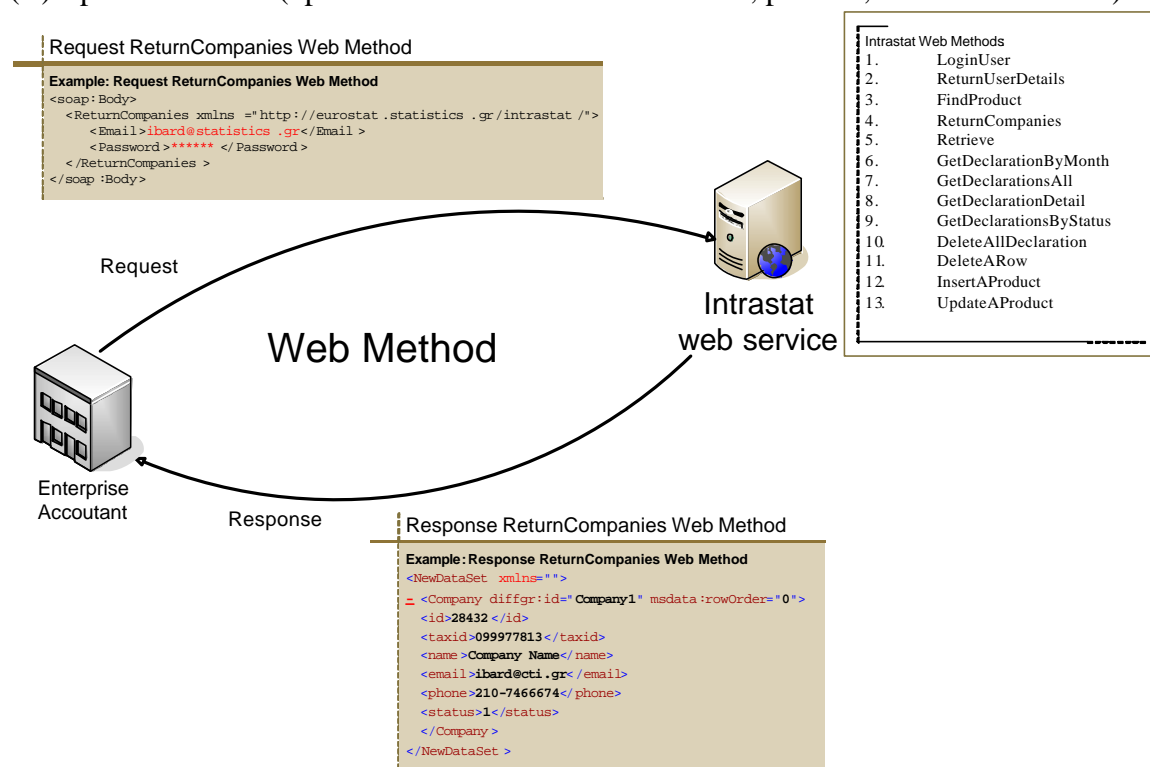
- provide interoperability between various software applications running on disparate platforms.
- use open standards and protocols. Protocols and data formats are text-based where possible, making it easy for developers to comprehend.
- can work, by utilizing HTTP, through many common firewall security measures without requiring changes to the firewall filtering rules.
- easily allow software and services from different companies and locations to be combined easily to provide an integrated service.
- allow the reuse of services and components within an infrastructure.
- can work through firewalls while other forms of Remote Procedure Calls may more often be blocked.

#### **IV. SYSTEM ARCHITECTURE**

25. The basic web methods that consists the Web Service are (Figure 2):

- (a) LoginUser (checks if the user has the rights to use the Web Service - Authentication Process. It also checks if the user is accountant of a company or the company itself)
- (b) ReturnUserDetails (returns the details for an authorized user)

- (c) FindProduct (returns a full description of a Product)
- (d) ReturnCompanies (presents a list of the companies that an accountant is responsible for)
- (e) Retrieve (returns the Details for a selected Declaration, for later modification of the Declaration)
- (f) GetDeclarationByMonth (returns the declarations - from history - that have been submitted by the company for a specific month and a specific year)
- (g) GetDeclarationsAll (returns all the declarations - from history - that have been submitted by the company)
- (h) GetDeclarationDetail (finds and returns the details for a selected declaration)
- (i) GetDeclarationsByStatus (returns a list with the declarations of a Company order by status)
- (j) DeleteAllDeclaration (deletes all the rows of a selected declaration - resets the declaration)
- (k) DeleteARow (deletes a row, product, from a selected declaration)
- (l) InsertAProduct (inserts a row, product, into a specified declaration)
- (m) UpdateAProduct (updates the values of a selected row, product, from a declaration)



**Figure 2. System Architecture**

26. For example, the ReturnUserDetails web method does not accept any parameters, although it requires the authentication and authorization of the user. This method returns a dataset with detailed description of an authorized user. If the user is a company, then the dataset is filled with details of the company. If the user is an accountant of ore than one company, then the dataset is filled with the details of the accountant. The dataset consists of the following fields:

- (a) Name: The name of the user.
- (b) Address: The address of the user.
- (c) City: The city of the user.
- (d) Zip: The zip code of the user.
- (e) Phone: The phone number of the user.
- (f) Fax: The fax number of the user.
- (g) Email: The email address of the user.
- (h) TaxID: The VAT Number of the user.
- (i) Taxoffice: The code of the Tax Office of the user.

- (j) Password: The password of the user.
- (k) Responsible: The Responsible Person of the user.
- (l) Flag: The Type of the user (0 for company, 1 for accountant).
- (m) Status: The Status of the user (1 when access for the user is enabled, 0 when access for the user is disabled).

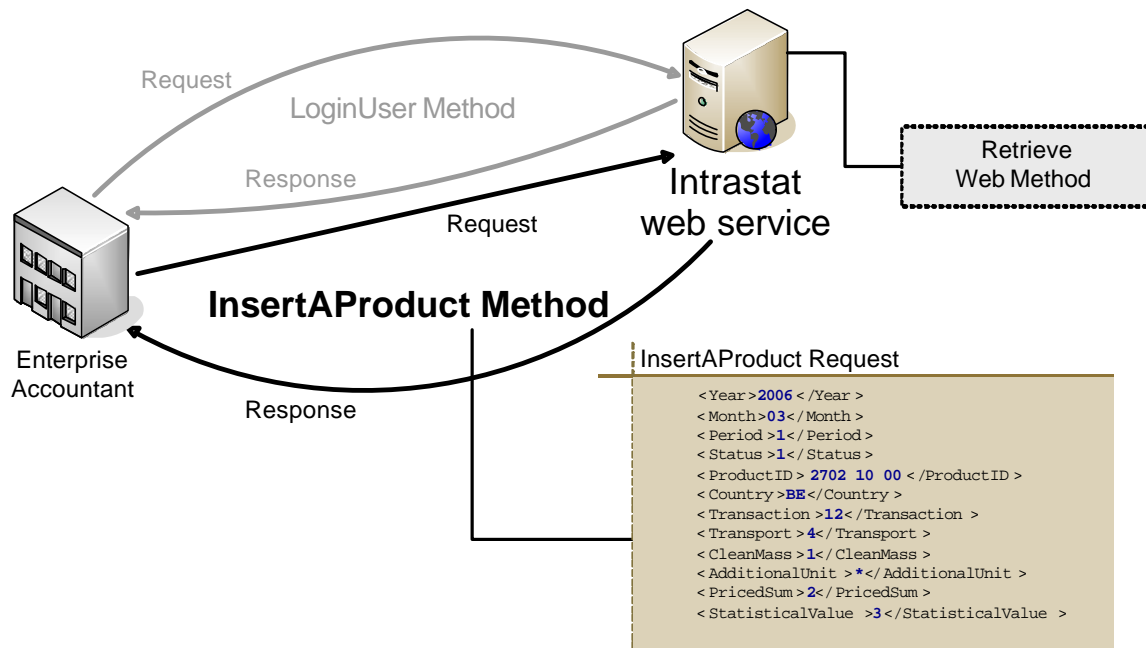
27. Below an example response message for the ReturnUserDetails web method is presented:

```
<?xml version="1.0" encoding="utf-8" ?>
- <DataSet xmlns="http://eurostat.statistics.gr/intrastat/">
- <xs:schema id="NewDataSet" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
- <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="el-GR">
- <xs:complexType>
- <xs:choice maxOccurs="unbounded">
- <xs:element name="User">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="name" type="xs:string" minOccurs="0" />
- <xs:element name="address" type="xs:string" minOccurs="0" />
- <xs:element name="city" type="xs:string" minOccurs="0" />
- <xs:element name="zip" type="xs:string" minOccurs="0" />
- <xs:element name="phone" type="xs:string" minOccurs="0" />
- <xs:element name="fax" type="xs:string" minOccurs="0" />
- <xs:element name="email" type="xs:string" minOccurs="0" />
- <xs:element name="taxid" type="xs:string" minOccurs="0" />
- <xs:element name="taxoffice" type="xs:string" minOccurs="0" />
- <xs:element name="responsible" type="xs:string" minOccurs="0" />
- <xs:element name="password" type="xs:string" minOccurs="0" />
- <xs:element name="flag" type="xs:unsignedByte" minOccurs="0" />
- <xs:element name="status" type="xs:unsignedByte" minOccurs="0" />
- <xs:element name="memo" type="xs:string" minOccurs="0" />
- </xs:sequence>
- </xs:complexType>
- </xs:element>
- </xs:choice>
- </xs:complexType>
- </xs:element>
- </xs:schema>
- <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-
microsoft-com:xml-diffgram-v1">
- <NewDataSet xmlns="">
- <User diffgr:id="User1" msdata:rowOrder="0">
- <name>Company Name</name>
- <address>Company Address</address>
- <city>??F?S??-?T??</city>
- <zip>14564</zip>
- <phone>210-8222222</phone>
- <fax>210-8222223</fax>
- <email>kmarkel@cti.gr</email>
- <taxid>094243464</taxid>
- <taxoffice>1159</taxoffice>
- <responsible> Accountant Name</responsible>
- <password>123456</password>
- <flag>0</flag>
- <status>1</status>
- </User>
- </NewDataSet>
- </diffgr:diffgram>
- </DataSet>
```

## V. CASE STUDY IMPLEMENTATION

28. Techniques have been implemented using the .NET v.1.1 framework technology and C# language [9] in particular. Generality is not lost as other Web Service supporting development platform may be utilized. The prototype is hosted on a MS Windows 2000 (Service Pack 4) Xeon at 3,2GHz using 2GB RAM. Quality database is designed and runs on an MS SQL Server 2000 sp4 instance. The particular development platform has been chosen mainly because of the authors' previous experience in the area.

29. The following usage scenario (Figure 3) describes how an enterprise or its accountant would make a declaration for a product, and how a developer would create a portion of a service.



**Figure 3. Overview of use case InsertAProduct Method using Intrastat web service.**

30. Companies have to follow the following steps in order to make a declaration that inserts a product using Intrastat web service:

- The Company requests for authentication using LoginUser web method.
- The response authenticates Company from the Intrastat web service (if the username and the password are right).
- The Company uses Retrieve web method in order to take some details (like year, month, period, status) of the declaration that is going to submit.
- The Company requests for using InsertAProduct web method.
- The response gives an answer “true” or “false”, if the procedure of this declaration completed successfully.

31. Using an SOAP request message for the InsertAProduct web method the schema is as follows:

```
POST /intrastat/webservice/intraservice.asmx HTTP/1.1
Host: eurostat.statistics.gr
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://eurostat.statistics.gr/intrastat/InsertAProduct"

<?xml version="1.0" encoding="utf-8"?>
<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>
    <InsertAProduct xmlns="http://eurostat.statistics.gr/intrastat/">
      <Year>2006</Year>
      <Month>03</Month>
      <Period>1</Period>
      <Status>1</Status>
      <ProductID> 2702 10 00 </ProductID>
      <Country>BE</Country>
      <Transaction>12</Transaction>
      <Transport>4</Transport>
      <CleanMass>1</CleanMass>
      <AdditionalUnit>*</AdditionalUnit>
      <PricedSum>2</PricedSum>
      <StatisticalValue>3</StatisticalValue>
    </InsertAProduct>
  </soap:Body>
</soap:Envelope>
```



```
</InsertAProduct>  
</soap:Body>  
</soap:Envelope>
```

32. An example of SOAP response message for the InsertAProduct web method follows:

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<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>  
    <InsertAProductResponse xmlns="http://eurostat.statistics.gr/intrastat/">  
      <InsertAProductResult>  
        <xsd:schema>schema</xsd:schema>xml</InsertAProductResult>  
      </InsertAProductResponse>  
    </soap:Body>  
  </soap:Envelope>
```

33. The developer uses the identifier to retrieve a WSDL. The WSDL defines the message formats, data types, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent. It also specifies one or more network locations at which a provider agent can be invoked, and may provide some information about the message exchange pattern that is expected. In essence, the service description represents an agreement governing the mechanics of interacting with that service.

34. Then the developer creates an implementation of the service based upon the WSDL, he tests it and deploys the implementation (makes a declaration that inserts a product) at the Intrastat web server.

## VI. CONCLUSIONS AND FUTURE WORK

35. During the time this paper was prepared, the web service development was completed and the module was in the technical testing phase. As soon as this stage is over, the pilot operation phase will be launched, which will also involve the users (enterprises) that participated in the feasibility study.

36. The feedback from the pilot phase will provide valuable input for the further improvement of the web service and will prepare the ground for the full-scale operation.

37. The future steps include the identification of possible areas of expansion of the use of the service and the deployment of additional similar facilities for other statistical sectors, in order to support the NSSG policies for e-Government services to the citizens.

## **APPENDIX A**

### **THE WEB SERVICES FRAMEWORK**

The Web Services framework consists of the following basic components and standards:

#### **SOAP**

Short for Simple Object Access Protocol, a lightweight XML-based messaging protocol used to encode the information in Web service request and response messages before sending them over a network. SOAP messages are independent of any operating system or protocol and may be transported using a variety of Internet protocols, including SMTP, MIME, and HTTP.

#### **WSDL**

Short for Web Services Description Language, an XML-formatted language used to describe a Web service's capabilities as collections of communication endpoints capable of exchanging messages. WSDL is an integral part of UDDI, an XML-based worldwide business registry. WSDL is the language that UDDI uses. WSDL was developed jointly by Microsoft and IBM.

#### **UDDI**

Short for Universal Description, Discovery and Integration. It is a Web-based distributed directory that enables businesses to list themselves on the Internet and discover each other, similar to a traditional phone book's yellow and white pages.

#### **API**

Abbreviation of Application Program Interface, a set of routines, protocols, and tools for building software applications. A good API makes it easier to develop a program by providing all the building blocks. A programmer puts the blocks together.

#### **XML**

Short for Extensible Markup Language, a specification developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

**APPENDIX B**

The analysis of the responses are summarized in the following table:

<b>Statistical Values of the Companies (Number of Companies:141)</b>								
<b>Year</b>	<b>Flow</b>	<b>Value (via Tax Departments)</b>	<b>Value (via Web)</b>	<b>Total Value</b>	<b>Value of logistic Packages</b>	<b>% of Total Values</b>	<b>% of Values (via Web)</b>	<b>Number Company firms (via Web)</b>
2004	1(ARRIVALS)	8,124,396,169	16,112,843,894	24,237,240,063	2,933,902,199	12%	18%	8,699
2004	2(DISPATCHES)	2,743,357,513	3,732,913,201	6,476,270,714	251,423,939	4%	7%	1,967
2005	1(ARRIVALS)	6,402,628,028	15,776,795,358	22,179,423,386	2,979,046,317	13%	19%	9,592
2005	2(DISPATCHES)	2,477,939,191	4,191,257,266	6,669,196,457	383,870,654	6%	9%	2,543

**Notes**

- Year: Year of the declarations.
- Flow: Flow the declarations.
- Value (via Tax Departments): The statistical values of companies' declarations, when they use the Tax Department for their declarations.
- Value (via Web): The statistical values of companies' declarations, when they use the Intrastat (via Web) for their declarations.
- Total Value: The total statistical values of companies' declarations
- Value of logistic Packages: The statistical values of companies' declarations. These companies replied to our e-mails.
- % of Total Values: The percentage of the statistical values of companies' declarations (the companies replied to our e-mails) and the statistical values of all companies' declarations.
- % of Values (via Web): The percentage of the statistical values of companies' declarations (the companies replied to our e-mails) and the statistical values of companies' declarations (these companies make their declarations via Web).
- Number Company firms (via Web): The number of companies that make their declarations via Web.

In the next figures, we can see the percentage of the statistical values of companies' declarations (the companies replied to our e-mails) and the statistical values of all companies' declarations. Additionally, you can see the percentage of the statistical values of companies' declarations (the companies replied to our e-mails) and the statistical values of companies' declarations (these companies make their declarations via Web).

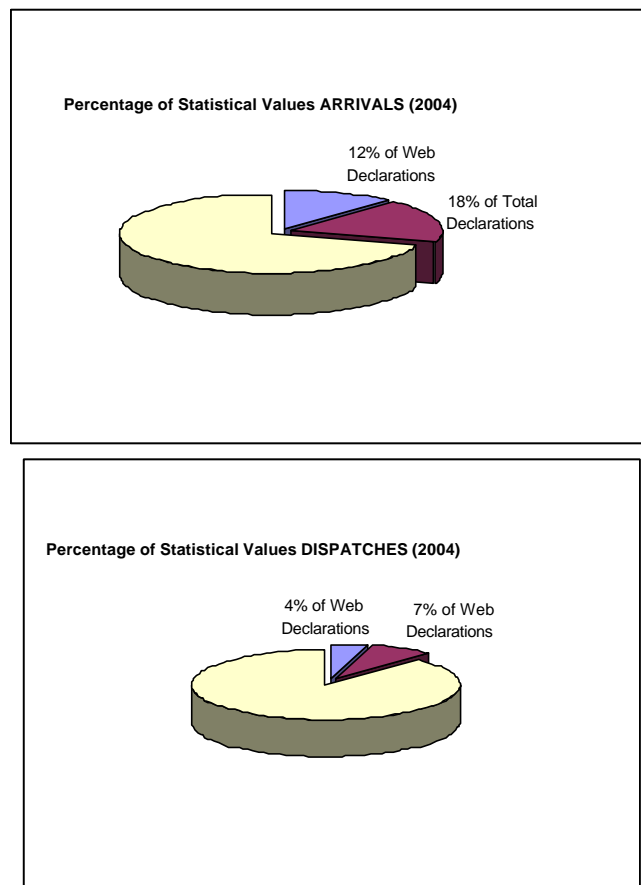


Figure 1: Percentage of Statistical Values Arrivals- Dispatches (2004)

We can observe that a great percentage of the market place (18% for 2004 and 19% for 2005) use Web for their declarations, which means that we should develop a new Service to make their declarations.

Additionally, after visiting the software houses and talk with developers, we decided the format of data, the validation checks needed for quality data that we should use in Web Service of Intrastat.

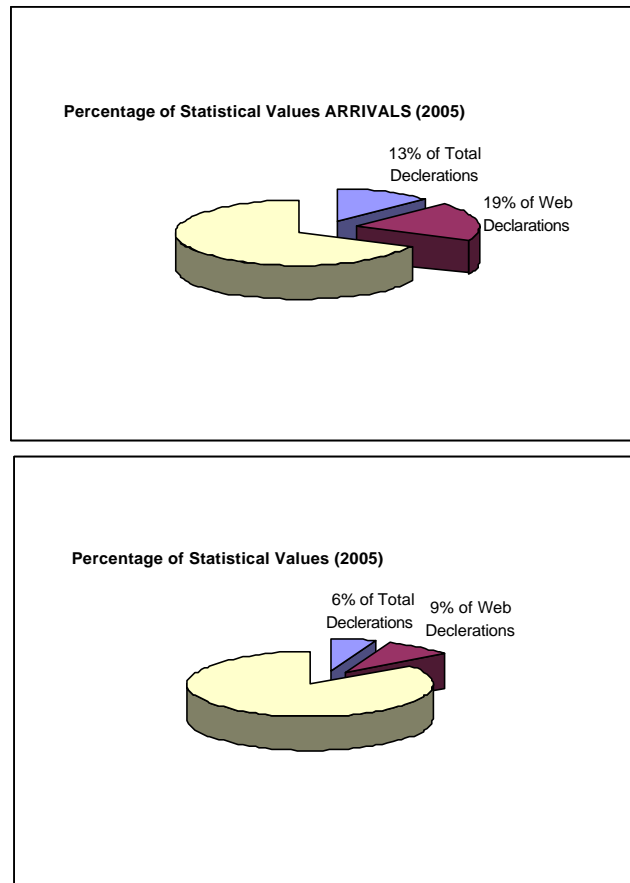


Figure 2: Percentage of Statistical Values Arrivals- Dispatches (2005)

## References

- [1] <http://www.ariadne.ac.uk/issue29/gardner/>
- [2] T.Ewald 2003. Understanding XML Web Services: The Web Services Idea. <http://msdn.microsoft.com/webservices/understanding/readme/default.aspx>
- [3] Extensible Markup Language (XML) <http://www.w3.org/XML/>
- [4] D. Box, D.Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H.F. Nielsen, S. Thatte, and D. Winer: May 2000. Simple Object access Protocol (SOAP) 1.1. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>
- [5] R. Data 2003. WSDL Tutorial. <http://www.w3schoolscom/WSDL/default.asp>
- [6] Web Services Description Language (WSDL) 1.1 <http://www.w3.org/TR/wsdl>
- [7] UDDI COMMITTEE 2002. UDDI Version 2.03 Data Structure Reference. <http://www.UDDI.org/pubs/DataStructurev2.htm>
- [8] Microsoft UDDI Business Registry Node – Test Site. <http://test.UDDI.microsoft.com/>
- [9] A. Banerjee, A. Corera, Z. GreenVos, A. Krowczyk, B. Maiani, C. Nagel, C. Peiris, T.Thangarathinam 2001. C# Web services: Building Web Services with .NET remoting and ASP.NET. Birmingham: Wrox Press, 255-275.

-----