Applications and Service Integration of Decision Support Systems Using .NET Platform

Ilie Tamaş
Economical Studies Academy
Bucharest, Romania
ilie.tamas@gmail.com

Radu Bucea-Manea
Center for European Studies and
Mobility, Bucharest, Romania
radub_m@yahoo.com

Rocsana Ţoniş
Center for European Studies and
Mobility, Bucharest, Romania
evrika4you@yahoo.com

Abstract— **În cadrul acestui articol vom discuta despre suportul oferit de platforma .NET pentru arhitectura orientată pe servicii (SOA). Pentru început vom face o scurtă introducere a platformei .NET şi apoi vom aprofunda caracteristicile şi principiile de baza ale SOA contemporane. Scopul nostru este de a demonstra că arhitectura SOA este opţiunea perfectă în dezvoltarea de aplicatii e-business. Am subliniat, de asemenea, că sistemele de suport decizional şi soluţii de business intelligence sunt aplicaţii ideale pentru a fi implementate în SOA, printr-un studiu de caz elocvent.**

Cuvinte cheie *- .Net; SOA; SOAP; integrare; servicii Web; WSDL*

Abstract— **In this paper we discuss the support offered by the .Net platform for Service Oriented Architecture (SOA). First we give a brief introduction on service integration. An overview of the platform .Net is followed by the architecture, characteristics and primitives of contemporary SOA. Our purpose is to demonstrate that SOA architecture is the perfect option in developing the e-business applications. We also underline that decision support systems and business intelligence solutions are ideal applications to be implemented in SOA, though an eloquent case study.**

Keywords- .Net; SOA; SOAP; integration; Web services; WSDL

I. INTRODUCTION

The paper covers a wide view on Service oriented architecture (SOA), such as principles, characteristics, benefits and disadvantages of SOA. The web services architecture on .Net platform presentation is essential for the user to understand the case study of the article.

The authors studied the specialized literature and came with the conclusion that SOA is best suited for E-Business applications, and also for developing Business Intelligence (BI) tools. The fact is demonstrated in a relevant case study, of developing a BI tool which calculates the correlation between different fields in a table (such as turnover and net profit). Our BI tool is able to calculate and implement the linear regression model, or other complex data mining models. Having the coefficients of linear regression equation calculated, the user is also able to simulate different cases that are not included in the current data warehouse. This way, the authors synthesized existent specialized literature and present ways of SOA integration with business processes and BI.

SOA is a flexible set of principles of design, used during the development phase. SOA requires integration and loose coupling of services with operating systems and other technologies which underlie computer applications. SOA offers an integrated suite of services that can be used in several business areas by providing a way to connect on HTTP requests made by the service clients. XML is used for SOA interface in terms of protocols and functionality. SOA services are separate units or functions that developers make accessible over a network in order to allow users to combine and reuse them in production applications.

II. CONSIDERATIONS ABOUT SERVICE INTEGRATION

Each automation solution of the activities in an enterprise, regardless of platform, is a collection of features and functions designed to perform certain business process by attending one or more tasks. The requirements for such a system is built are generally well defined and relevant during the construction. But along the way, they may change. There are many causes that influence change in contemporary corporations. Here are some of the most common examples:

- Extension of the organization business field. When organizations are subject to periods of growth, their business often expands. These changes have an impact on the technological environment, responsible for the automation of the initial business process.
- Contraction of the business organization field. This contraction may be a response to changes in general economic or business climate change (such as the emergence of a new competitor in the market or loss of major clients). Whatever the cause, will require significant adjustments to the existing business processes, that can lead to major changes in the business automation logic. It is important to note that the contraction of business field is not always synonymous with a reduction in the size of the organization. Sometimes, organizations simply remove some areas of business to focus on others.
- Purchase of an organization by another organization. This may require incorporating the company's assets in an existing environment that requires coordinated integration over move levels. Business processes

affected are usually augmented or completely remodeled. Most problems of integration arise from the introduction of external data models that vary in design and content.

- The merger of two organizations. Requirements resulting from the merger may include direct integration or assimilation of automation solutions, but most likely will introduce a type of inter-organizational collaboration.

The globalization spreads more and more and lets us encounter these events more frequent. The organization has to adapt very fast to the situation, meeting the next requirements:

- Cross-platform interoperability - The ability of previously independent applications to be integrated with other applications that are hosted on the same platform and have been developed for different platforms of other vendors.
- Changes in cross-platform interoperability requirements – the capacity of existing channel to be expanded or replaced entirely, in response to technical or business requirements due to change.
- Logical abstraction - ability to make re-engineering on existing logic, even replace it entirely (often with completely new technologies).

SOA can be fully adapted to the construction of integrated architectures. SOA allows many benefits and new features like the ability to meet the challenges of market organization. Service oriented integration, therefore allows organizations to become highly responsive to change, based on SOA services foundation.

Business enterprise architecture defines the design of detailed tasks of the business process, the business policies (e.g. management of metadata), and the information technologies included in an IT infrastructure, based on the definitions of what firms does as a business. This infrastructure includes the integration of applications, databases, information, standards, and platform technologies behind the processes. An SOA includes the bulk of enterprise architecture, as in [1].

III. WEB SERVICES

Web services define connection standards, type of communication, description and discovery of components distributed over the Internet, as in [2]

The layers of a Web service are: transport layer provided by transport protocols such as HTTP, SMTP, FTP; XML messaging layer - managed by the SOA protocol; layer of services description through WSDL language; publication and integration layer - provided by UDDI. The WSDL document is itself an XML document. This begins with schema namespace definitions, which are included as a root element in the WSDL document that's using the <definitions> element, as in [3].

Modern web services are distributed on the Internet, as a process-oriented software architecture. All types of browsers facilitate access to services by sending the requests to web servers. Web Servers launches their messages to application servers in XDR-XML-RPC format or SOAP with performance parameters and the response is received in the same format.

SOA Protocol consists of three components: a header to describe the content of the message, a set of instructions for message encapsulation, a mechanism for distributed called management procedures.

Accessing a SOAP service and building a customer for this purpose, requires the WSDL associated file or XSD file and the service location. These resources specify public methods accessible through the service and type of required parameters to access these functions flawlessly. Many SOAP technologies, hybrid technologies, such as those based on RPC-XML or the pure, have demonstrated interoperability, even when a common technology is asked to publish and to access a web service.

Standard service oriented architecture, as in [4], is a model based on components, whose purpose is the coupling of software applications that interact with each other and provide services. The architecture allows the discovery services based on their descriptions. Loose coupling services brings advantages through the flexibility and extensibility of the architecture itself, and the internal structure and implementation services. This feature makes service-oriented architecture to be fit for dynamic business and based on explicit requests for services.

The benefits of SOA distributed architecture on Internet are [2]: comprehensible; integrating dispersed resources, effective implementation; updated automatically, allows reuse of resources; has advanced search mechanisms; has a strong virtual community.

The main disadvantages of SOA, in terms of the same author, are: low capacity in terms of representativeness, small number of instruments and facilities; impossibility of establishing complete ontology; inadequate management of the content transmitted, known security problems for open systems.

A. SOA and .NET platform

Most researchers and developers consider that SOA is a framework for integrating business processes and supporting IT infrastructure in terms of safety, through standardized components (services) that can be reused and combined to respond to changing business priorities. E-business applications perfectly illustrate the need for a service-oriented architecture.

Web services oriented architecture implements three types of operations:

- Publishing service: promotion of the service by the service provider;
- Discovery service: the service can be found based on WSDL description of a service and client search criteria. A WSDL file does not contain the full interface of a service. WSDL is simply a format for specifying the technical aspects of calling Web Services, as in [5] Semantic Web and Web 3.0 are successfully implemented modern paradigm in the currently searching and discovering services on the Internet. A customer may request a

particular type of service or provider characteristics or quality of service features, etc.

- Connect: Connect the service provider with the service client on a client-server pattern.

.NET is a proprietary solution for code running and for development, designed for use with Windows operating systems and servers. .NET platform can be used to provide a variety of applications, from desktop and mobile systems to distributed Web solutions and web services. An important .NET component for SOA is ASP.NET environment, used to provide the level of web technology in SOA (and subsequent additions and the Web Services Enhancements (WSE) extension.

Architecture components: .NET provides a design environment for delivery of various types of distributed solutions. Listed below are often associated with application components. NET Web-based: ASP.NET web forms, Web Services ASP.NET, assemblies, runtime environments, programming languages, APIs, service providers, client service, service agents.

Assemblies - a set / package is the standard unit of processing logic in the environment. NET applications. An assembly can contain several classes that share more code using principles of object-oriented code. The .NET application logic related to a web services is usually contained in an assembly.

Runtime environments - Architecture components described above are based on implementing a common language named common language runtime (CLR) provided by .NET Framework. CLR provides a collection of running / performance agents, offering a range of .NET applications management services, including support for multiple languages, to centralize data, the life cycle of objects and memory management.

Service providers - .NET Service providers are Web services, called ASP.NET Web Services. An URL will contain the extension „.asmx" to call an ASP.NET web service. Instant server applications are waiting constantly to for the client request and then provide a response.

Client Service - Client Service can be achieved with other technologies than the .NET, platform and can run at distance on different client computers simultaneously. Creating client service entails the use of proxy classes, plus the logic of the application service requested, class that copies the service provider interface.

Services agents - carried out various activities of performing to tasks, system tasks such as authentication, authorization and states management..

B. Support for SOA primitives

.NET Framework provides native support for SOA primitives through its runtime environment and development tools, as explained below.

Encapsulation service - applications are built on components that are then integrated and encapsulated by the ASP.NET Web Services.

Flexible assembly - services are accessed via public service interfaces as WSDL descriptions that provide the SOAP messaging framework

Messaging - components communicate with one another through messages, more specifically through the .NET extension, Microsoft Messaging Queue (MSMQ).

C. The characteristics of contemporary SOA

Service oriented architecture standard is defined by architecture and technology-based Web services. Web service-based architecture has the following characteristics, as in [6]:

- Is based on XML technologies;
- Is independent of transport protocols (HTTP, SMTP);
- Used for exchanging XML messages between applications and services interact. These messages allow expansion and adaptation to different requirements for reliability, security, etc.. The exchange is done using SOAP protocol (Service Oriented Architecture Protocol)
- Allows publication of functional capacity, quality services, using the description language WSDL (Web Services Description Language).

SOA is evolving, but we can talk about some characteristic of SOA current mature version, in terms of [7].

C1. It is based on open standards: .NET includes a large library of namespaces which assists first-generation industry standards for Web services specifications.

C2. Facilitate vendor diversity: .NET platform can integrate other web services developed in other technologies and interoperability will continue as long as all services are exposed in accordance with common standards (as dictated by the Base Profile, for example).

C3. Intrinsic interoperability - .NET (Version 2.0.) with Visual Studio 2005 provides native support for WS-I Basic Profile.

C4. Promote federation - BizTalk Server platform is a .NET extension that promote federation in various business environments.

C5. Architecture on components - .NET is an example of a component programming model, functionally independent.

C6. Extensibility – .NET platform has many extensions that allow expansion of the application logic (implemented through combinations with service-oriented design classes).

C7. Assisted modeling service oriented business modeling concept-oriented business assistance services can be implemented. NET by creating a standard one, based on entities and tasks. Orchestration layer requires a layer manager, able to execute the process definition that centralizes business logic.

C8. Abstracting the logical level -.NET SOAs can position ASP.NET web services and the layers services to the abstract logic on different levels. Legacy and new application logic and abstract formulation can be embedded entirely through appropriate interface design services.

C9. Organizational ability and flexibility of assembling to business layer - Since assembly. NET assist developing industry standards for Web services, proper positioning and application service layers allows the creation of levels of abstraction necessary to promote basic skills.

Building services-oriented solutions with .NET usually involve the creation of service providers as ASP.NET Web Services (assisted by processing logic in the implementation of business components as assemblies) and client services using auto-generated proxy classes. Service developer code is transformed in Common Language Runtime and then executed. Libraries of .NET classes are the basic elements of .NET Framework, which are organized as namespaces to which it refers. A subset of the characteristics of contemporary SOA can be achieved with .NET native features, while others can be achieved through product extensions.

IV. CASE STUDY

A web service consists of two major components: service provider and service client. The source code will be compiled into a. asmx file and will be saved in an virtual directory of the IIS server, where it can be invoked by the client.

A. Service provider implemented in C#

The service provides a connection to a data warehouse Oracle 10g, enables customers to choose between many types of queries Rollup and Cube and return the query result into a string variable. Service source code is presented below.

Further, we intend to develop a Business Intelligence tool to calculate the correlation between the turnover and net profit. The r correlation coefficient formula is:

$$r_{y/x} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{\left[n\sum x_i^2 - \left(\sum x_i\right)^2\right]\cdot\left[n\sum y_i^2 - \left(\sum y_i\right)^2\right]}} \quad (1)$$

where $-1 \le r_{y/x} \le +1$.

In this formula *x* will be the turnover, *y* the net profit and *n* the number of companies. *N* is in fact the number of records in table. Our data warehouse contains the *Financiar* table with the turnover, net profit, and number of employees.

The calculus of *r* implies previous steps to calculate x^2 and y^2, stored in the virtual table *tmpFin.* In a second virtual table, *tempFin,* are stored the intermediate results (sums) presented in the formula. In the third query the correlation coefficient is calculated. Our queries are:

select CA, PN, NrSal, CA*PN As XY, (CA*CA)As XX, (PN*PN)As YY into tmpFin from Financiar, where CA is turnover, PN is net profit, and NrSal is number of employees.

select count(CA)as n, sum(CA) as sumX, sum(PN) as sumY, sum(XY)as sumXY, sum(XX)as sumXX, sum(YY)as sumYY, sum(CA)*sum(CA) as sumX2, sum(PN)*sum(PN) as sumY2 into tempFinR from tmpFin

select (n*sumXY-sumX*sumY) / sqrt((n*sumXX-sumX2) * (n*sumYY-sumY2)) as CoeficientCorelatieR into tmpR from tempFinR

The equation for linear regression is Y=aX+b, where Y=PN and X=CA. The formulas for the linear regression model coefficients are:

$$a = \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \quad (2)$$

$$b = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \quad (3)$$

Having the coefficients calculated we may introduce its to our service.

OCM.asmx page will contain only one directive indicates the environment. NET to export a web service, written in C # and implemented by the COM class. The code itself is written in the page App_Code / Service.cs and is presented below:

```
<%@ WebService Language="C#" CodeBehind="~/
App_Code/Service.cs" Class="OCM" %>
```

In the first step the software libraries are included in the project. They will be called as namespaces within the service. With the directive “using” are attached the assemblies, too, in order to create the web service that allows the users to access the database.

```
using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data;
using System.Data.OleDb;
```

The developer establishes an explicit namespace [WebService (Namespace = "urn: OCM ')], not to be selected one by default (http://tempuri.org/) by. NET. WebMethod stresses that the OCM class method is part of the web service, and is available to customers, as in [8].

The developer creates a public class, called OCM, and its method with a parameter of string type called sayOCM. The alternative *switch()* instruction, implemented in the function, allows the customer to chose of a query.

```
[WebService(Namespace = "urn:OCM")]
public class OCM
{
  [WebMethod]
  public string sayOCM(string name)
```

```
{

    switch (name)
    {
      case "r coefficient":
        return myquery ("select * from tempFinR ");
        break;
      case "a coefficient":
        return myquery ("select (sumY*sumXX–sumXY
*sumX) /(n*sumXX-sumX2) from tempFinR");
        break;
      case "b coefficient":
        return myquery ("select (n*sumXY-sumX*sumY)/
(n*sumXX-sumX2)  from tempFinR");
        break;
      case "clients bills":
        return myquery (" SELECT Facturi.NrFactura,
dbo.Facturi.DataFactura, Facturi.ValoareTotala, SUM
(ArticoleFacturate.PretUnitar) FROM ArticoleFacturate
INNER JOIN Facturi ON ArticoleFacturate.FacturiID =
Facturi.FacturiID GROUP BY CUBE(Facturi.NrFactura,
Facturi.DataFactura), CUBE (Facturi.ValoareTotala)");
        break; ……..……
       default:
        return "OCM";
        break;
    }
}
```

The connection to the data warehouse is done by obtaining a Connection object (myOleDbConnection) initialized via a default constructor having as parameter the connection string - provider: Oracle, Express Edition XE, the user ID and its password.

```
public string myquery (string qry_str)
{
    string connectionString;
    OleDbConnection myOleDbConnection;
    OleDbCommand myOleDbCommand;
    OleDbDataReader myOleDbDataReader;
    DataTable schemaTable;

connectionString = "Provider=OraOLEDB.Oracle;Data
Source = XE; User Id=system; Password = pass";
myOleDbConnection = new OleDbConnection
(connectionString);
myOleDbCommand = myOleDbConnection.Create
Command();
    myOleDbCommand.CommandText = qry_str;
    myOleDbConnection.Open();
```

Once the connection was made the CommandText property of the myOleDbCommand object is initialised with the string representing the SQL query to be executed. It is created a DataReader object to return the records from the data warehouse:

```
myOleDbDataReader = myOleDbCommand.ExecuteReader();
```

The SchemaTable allows the application to returned the field names of the tables required by the query. The fields are displayed with the repeating statement *foreach ()* resulting the table header:

```
    schemaTable = myOleDbDataReader.GetSchemaTable();
    string str = "<table border=1><tr>";
    int i;
    foreach (DataRow myField în schemaTable.Rows)
    {
      str+= "<td>"+myField[0].ToString()+"</td>";
    }
    str += "</tr><tr>";
```

To return the table contents the developer uses a repetitive nested parsing of the code. Thus, the columns associated to the myOleDbDataReader vector will be merged, line by line, into a string variable (str).

```
    while (myOleDbDataReader.Read())
    {
      for(i=0;i<myOleDbDataReader.FieldCount;i++)
     str +=<td>"+Convert.ToString(myOleDbDataReader[i])
+ "</td></tr>";
    };
    myOleDbDataReader.Close();
    return str;
```

The service runs in the execution environment. NET and can be called by a client in a SOAP request sent using HTTP protocol, in order to:
- display service interface, automatically generating its WSDL description
- invoke the public methods provided by the web service via SOAP 1.1 or SOAP 2.0 (operation of selection queries)
- show a method provided by the service, as in [8]

After starting the .NET Silverlight server by clicking Run, the result can be viewed in the browser (http://localhost/OCM.asmx). If using a version of IIS web server it is called the OCM.asmx file, depending on the relative path to the public directory (c:\inetpub\wwwroot\).

SayOCM link opens the window to call the method query by providing the parameter requested by the alternative instruction *switch(),* presented above. Also there is a detailed description of the invocation manner of sayOCM method using SOAP 1.1., SOAP 1.2 or POST requests. The Service Description link opens the WSDL file associated to the service.

The flexibility and the security of the service provided are emphasized when the user accesses another server than that containing the service provider. We simulate this situation by creating a PHP client that is hosted on a server Apache.

B. *Service client implemented in PHP5*

OCM.php page code is generated automatically by the WSDL2PHP tool. This tool is developed and managed by [9] and use a combination of SOAP and DOM extensions in PHP5 (http://www.urdalen.no/wsdl2php/ manual.php).

The invocation of service uses its associated WSDL file. So the public member function can be called directly as a method of the $var object, that is an instance of the OCM class that extends (inherits) SoapClient class.

```
<?php
class sayOCM {
  public $name; // string
}
class sayOCMResponse {
  public $sayOCMResult; // string
}
class OCM extends SoapClient {
  private static $classmap = array(
          'sayOCM' => 'sayOCM',
          'sayOCMResponse' => 'sayOCMResponse', );

  public function OCM($wsdl = "OCM.wsdl", $options = array()) {
    foreach(self::$classmap as $key => $value) {
      if(!isset($options['classmap'][$key])) {
        $options['classmap'][$key] = $value;
      }
    }
    parent::__construct($wsdl, $options);
  }
  public function sayOCM(sayOCM $parameters) {
    return $this->__soapCall('sayOCM', array($parameters), array( 'uri' => 'urn:OCM', 'soapaction' => '' )
      );
  }
}
?>
```

A low level call can be done through the method *__soapCall*.

SayOCM.php page code implements an HTML form that contains a combo box and a button tools to invoke the request to the service provider.

```
<html><body>
<form method= "GET" action="sayOCM.php">
Choose one of the ROLLUP or CUBE queries:<br>
<select name="args" onchange="javasript: document.forms[0] .submit()">
<option value = "r coefficient">Return the R Correlation Coefficient </option>
<option value="a coefficient"> Return the a coeficient. </option>……..
</select>
<input type="submit" value="Send">
</form>
```

Business architecture perspective can be conceived of a service-oriented business alignment mechanism to facilitate effective business integration, as in [10].

CONCLUSIONS

Modern companies make frequently online transactions, exchanging documents, information, and knowledge. The best decisions have to be taken in the global environment. This is the reason users have to access online the DSS and BI in a secure mode, with low consumption of memory resources and time. Here comes the SOA that allows the integration of services and applications from different platforms, with different architectures. SOA improves the web service discovery through the WSDL description and implements the principles of semantic web. In SOA the service client, implemented in PHP5, accesses an instance of the provided service, implemented in .NET platform. This fact secures the communication. The inheritance is easy to be implemented because different components of a web service, designed in accordance with the Base Standards can be reorganized inside other web services. In other words SOA organizes, secures the internet and allow free communication between companies, helping them to adapt to a changing environment.

In their approach to success the modern companies answer to the globalization through Business Intelligence tools. Business Intelligence focuses on analyzing data and information, analyzing human behavior and analyzing the performance of the business, as in [11]. The last two key aspects can be implemented by pervasive BI, with service oriented architecture. Our case study demonstrates how to build a BI tool on SOA principles.

REFERENCES

[1] J. P. Lawler, H. Howell-Barber, Service-Oriented Architecture: SOA Strategy, Methodology, and Technology, Boca Raton: Auerbach Publications, 2008, FL., USA, pp. 6.

[2] E. Turban, J. Aronson, T. Liang, R, Sharda, Decision Support and Business Intelligence Systems. 8th ed, New Jersey: Pearson Prentice Hall, 2007, pp. 695–696.

[3] J. Hasan, M. Duran, Expert service-oriented architecture in C# 2005, Berkley:Apress, 2006, p. 16.

[4] H. Hao, „What is Service ¬ Oriented Architecture”, 2003, available on-line at: http://Webservices.xml.com/lpt/a/ws/2003/09/30/soa.html.

[5] N. M. Josuttis, SOA in Practice: The Art of Distributed System Design, Sebastopol: O'Reilly, 2007, p. 224

[6] A. Gălătescu, R. Iliescu, V. Popa , A. Cosmescu, T. Greceanu, C. Neicu, „Arhitecturi funcționale pentru proiectarea şi integrarea serviciilor bazate pe semantică”, 2006, available on-line at: http://www.ici.ro/RRIA/ria2006_4/art10.html.

[7] T. Erl, Service Oriented Architecture – Concepts, Technology and Design, Crawfordville: Prentice Hall, 2006, pp. 688–702.

[8] L. Alboaie, S. Buraga, Servicii Web – Concepte de baza şi implementari, Iasi: Polirom, 2006, pp. 224-228.

[9] K. Urdalen, available on-line at: (http://www.urdalen.no/wsdl2php/manual.php).

[10] M. Bell, Service-Oriented Modeling: Service Analysis, Design, and Architecture, New Jersey:Wiley, 2008, p. 231.

[11] G. Curtis, D. Cobham, Business Information Systems: Analysis, Design and Practice, Harlow: Pearson, 2008, England, p. 229