

# IMPLEMENTING BPEL COMPLIANT SOA ORCHESTRATION USING BIZTALK SERVER

Bayu Adhi Nugroho<sup>1</sup>, Riyanarto Sarno<sup>1</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya, 60111, Indonesia  
Email: nugroho09@mhs.if.its.ac.id<sup>1</sup>, riyanarto@if.its.ac.id<sup>1</sup>

## ABSTRACT

Service Oriented Architecture has promised better way in handling loose coupled component of software architecture. However, implementing SOA itself is a kind of art in software engineering practice and Business Process Execution Language (BPEL) is quite having a bright promise to handle such function. Being language to define SOA interaction it has met many challenges, broad debates both support and doubt have emerged, but its main feature as an executable document have made it successfully gain trust. BPEL actually have rivals, name Business Process Modeling Notation (BPMN) and Windows Workflow on it, each of them have their own plus compare to it. Examples need to be brought up to prove BPEL usage in daily business process basis. Bottom-up approach by creating web services as SOA implementation and managing them through orchestration inside a BPEL engine, resulting empirical usage of BPEL as a proven theory of integrating SOA using BPEL actually quite sufficient samples. Conclusion of SOA orchestration in BizTalk Server resulting BPEL documents would be represented briefly in this paper.

**Keywords:** SOA, orchestration, BPEL, BizTalk, software engineering.

## 1 INTRODUCTION

Web Services Business Process Execution Language defined by OASIS is currently in version 2.0, it contains revisions from previous, which are BPEL4WS (Business Process Execution Language for Web Services) 1.0 and 1.1 [23]. Due to BPEL restrictions on Microsoft BizTalk server [9], indeed there are incompatibility issues with Biztalk's native grammar and reasonable doubt of BPEL compliances on other engines would have raised rationally.

Founded by IT companies, OASIS is determined to demonstrate the highest level of commitment to open standard [16]. Having qualified contributors [15], BPEL should represent

mature behaviour to open systems. Being part of OASIS founder [16], Microsoft currently does not provide mature environment to BPEL [10]. Parasoft BPEL Maestro Engine [18] and other product [1] have promises in BPEL usage, ignoring language programming domain being used for SOA implementation [18], which is actually a BPEL's vendor-neutral openness's spirit.

## 2 MODEL, ANALYSIS, DESIGN, AND IMPLEMENTATION

### 2.1 Theory / Background

Service-oriented architecture (SOA) as paradigm for realization and maintenance of business processes that include large distributed systems which is based on three major technical concepts [14]:

1. Services
2. Interoperable by using enterprise service bus
3. Loose coupled

A service itself is a piece of self-contained business functionality which might be simple (storing or retrieving customer data), or complex (a business process for customers order) [14]. As mentioned in [14], services are concentrating on the business value of interfaces, they are bridges between business and IT. In the other hand, web services are widely regarded as the way SOA should be realized in practice [14]. As mentioned in [21], services are able being aggregated to satisfy more complex business requirements. Written in this paper terms web services and services are being used interchangeably.

Modeling workflow capability of business process nowadays is demanding executable notation and simplicity, while other model lack of them, such as: BPMN [17] is unable to provide executable environment and Windows Workflow [11] is being complicated by code-programming, such demands have rise executable grammars : XLANG [12], WSFL, BPEL4WS (shortly BPEL) [23]. The Web Services Business Process Execution Language (WS-BPEL) / BPEL, is a current primary industry specification which conducts standard of

orchestration in the Web Services context [13].As mentioned in [13]: "The BPEL orchestration

leverage the intrinsic interoperability provided by Web Services,

Table 1: Possible forms of loose coupling in SOA [14]

	<b>Tight coupling</b>	<b>Loose coupling</b>
Physical connections	Point-to-point	Via mediator
Communication style	Synchronous	Asynchronous
Data model	Common complex types	Simple common types only
Type system	Strong	Weak
Interaction pattern	Navigate through complex object trees	Data-centric, self-contained message
Control of process logic	Central control	Distributed control
Binding	Statically	Dynamically
Platform	Strong platform dependencies	Platform independent
Transactionality	2PC (two-phase commit)	Compensation
Deployment	Simultaneous	At different times
Versioning	Explicit upgrades	Implicit upgrades

conceiving orchestration itself as a service, specified in terms of a high level language and implemented through an engine". Engine itself will acts as Enterprise Service Bus/ESB. An enterprise service bus (ESB) as the infrastructure which enables highly interoperable among distributed systems of services therefore make it easier to distribute business processes throughout systems in different platforms and technologies [14].

As defined in [4], SOA is manageable by its nature of interaction and collaboration through two types of activities:

1. Choreography

Choreography tracks the sequence of messages that may involve multiple parties and multiple resources, each party participated in the process describe the part they play in the interaction role and it is associated with the public message exchanges that occur between multiple web services. [4]

2. Orchestration

By using orchestration, the process is always controlled from the perspective of one of the business parties, it differs from choreography in case it describes a process flow between services which is being controlled by a single party. [4]

By using Microsoft BizTalk, it is obvious we have agreed BizTalk is the only party responsible in managing and controlling messages, therefore we always mention it as BizTalk's orchestration.

BPEL stack as part of ESB implementation indeed competitive compare than other integration solutions such as Windows Workflow Foundation / WWF. Windows Workflow is designed to run as class libraries [3] as shown in figure 1, so it can be determined as vendor specific and code-oriented binding, while BPEL offers orchestration which has centralized configuration as shown in figure 2.

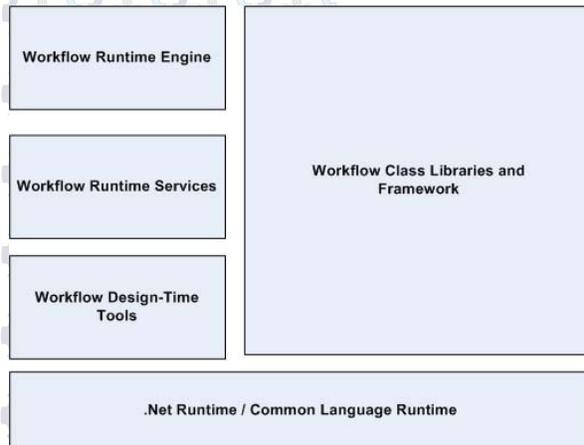


Figure 1: Windows Workflow Layer [3]

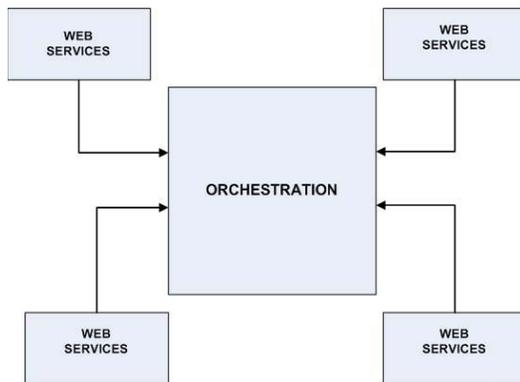


Figure 2: Orchestration Architecture [6]

BPEL would fulfill loose-coupled role compare than code oriented binding because its construction is in XML manner, and it helps the aim of IT as a catalyst for the sustainability of enterprise objectives [20].

The principles of abstraction, encapsulation, modularity, hierarchy, typing, concurrency, and persistence are encompassed by the object model [5]. As mentioned in [5]: "Object-oriented programming is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships". Three important portions of object-oriented programming definition [5]:

1. Objects are used as fundamental logical building blocks, not algorithms.
2. Each object is an instance of particular class.
3. Classes are related to one another through inheritance relationships.

An entity in object oriented domain usually is represented by a class using ORM (object relational mapping) technology, such as NHibernate [19]. This kind of entity will be used later in functional / logical process in computer programs. Those functions then will be arranged in form of services, those services need to be known by outside system and encapsulated as web methods. BPEL engines such as Biztalk [9] and ActiveBPEL [1] are able to manage those web methods using BPEL including create newly service and create messages to other engines.

## 2.2 Methodology

This research is trying to obtain actual results of SOA orchestration using Microsoft BizTalk Server, current version we were using was Microsoft BizTalk Server 2009 integrated with Microsoft Visual Studio 2008. By conducting

orchestrations we were expecting extraction results from any individual orchestration in BPEL form, thereby we can conclude whether the orchestration is BPEL compliant or not. Below here are steps to follow on implementing SOA orchestration using BizTalk:

### 2.2.1 Construction of Entity Classes

Entity-classes are important, since they are representing encapsulation of business domains which are available. Those entities contain attributes and methods to define business in information system domain.

### 2.2.2 Web Services Preparation

Web Services as SOA's implementation are interfaces between entities and orchestration, they act as gateways to inner entities which are available being consumed by orchestration.

### 2.2.3 Import Web Service

Web Services are readily consumed by outside system if they are exposed inside WSDL documents. WSDL document contains details of Web Service which are defined to be consumed by public, it represents how Web Service are consumable, what kind parameters and return types that pass through Web Service.

### 2.2.4 Orchestration Configuration

Orchestration is readily to be built after Web Services involved was already being set up.

### 2.2.5 Scheduler Configuration

Scheduler needs being set up to activate BizTalk's orchestration in such timely manner intervals.

## 2.3 Design and Analysis

For simple case, already known business processes below in goods inventory:

1. Newly goods arrived, this means to add goods inside particular store.
2. Remove goods from inventory.
3. Moving goods to other store.

This experiment was being implemented using C# and NHibernate platform to create object entities. Each of processes mentioned above will be represented as service, then will be sent to outer world as web-methods.

### 2.3.1 Construction of Entity Classes

To manage entities of goods, NHibernate is necessarily being set up. As shown in figure 3 is the

class-diagram represent objects' relationships which is being used, creation of the diagram was made using Enterprise Architect [22]. There are two entities:

1. Goods, this entity will act as primary item which is going to be manipulated by our services.
2. Store, this entity shows storage of goods which is available in each distinct store.

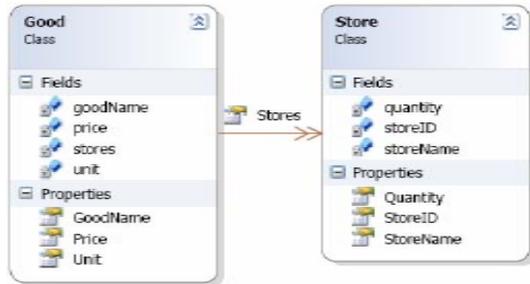


Figure 3 : NHibernate Entities

NHibernate have already being configured to generate SQL as a Data Definition Language in Microsoft SQL Server 2005, and it have worked well, below here is the SQL which was generated by NHibernate 2.1.2 :

```
if exists (select 1 from sysobjects where id =
OBJECT ID(N' [FK5C0EC3F79DF1EAD9]')
AND parent obj = OBJECT ID('Store')) alter table
Store drop constraint FK5C0EC3F79DF1EAD9 if
exists (select * from dbo.sysobjects where id =
object id(N'Good') and OBJECTPROPERTY(id,
N'IsUserTable') = 1) drop table Good if exists
(select * from dbo.sysobjects where id = object
id(N'Store') and OBJECTPROPERTY( id,
N'IsUserTable') = 1) drop table Store create table
Good ( Good Name NVARCHAR(100) not null,
Price NVARCHAR(255) null, Unit
NVARCHAR(255) null, primary key (Good
Name)) create table Store ( Store ID
NVARCHAR(255) not null, Store Name
NVARCHAR(100) null, Quantity
NVARCHAR(100) null, Good Name
NVARCHAR(100) null, primary key (Store ID) )
alter table Store add constraint
FK5C0EC3F79DF1EAD9 foreignkey (Good
Name) references Good
```

### 2.3.2 Web Services Preparation

As shown before, methods to expose attributes in classes which are available as raw data need to be created. Store class already being set up containing 3 distinct methods, those are :

1. Add

Add method is primarily used to append goods inside a particular store.

#### 2. Delete

Delete method is primarily used to delete goods inside a particular store

#### 3. Move

Move method is primarily used to move goods inside a particular store to another store

The interesting part that is available to be used as a simple test case is : although *move* method is already available inside the class, *the method can be generated inside the services instead, using composition of add and delete* method. Other method such as sum of goods in a particular store, is available through services only. This experiment was using .asmx web services rather than WCF based services preferred, our primary concern is that .asmx web services provide backward compatibility in SOAP based native protocol, rather than Microsoft-centric messaging such as WCF, and being used *NHibernate Best Practices with ASP.NET* [10] pattern to implement our code.

### 2.3.3 Import Web Service

The .asmx web services are importable into Biztalk 2009 engine using: *Project - Add Generated Items - Consume WCF Service - Metadata Files*, then choose *WSDL and XSD source type*, other option using MEX endpoint is usable primarily for WCF based services. Evaluating the steps before, our must have resource is a WSDL document.

### 2.3.4 Orchestration Configuration

Necessary orchestrations were created as shown. There were three orchestrations consist of :

1. Add Goods at Store orchestration
2. Remove Goods at Store orchestration
3. Move Goods orchestration

Web methods behalf on web service technology for doing such thing were being developed to be consumed by BizTalk, using .asmx technology.

### 2.3.5 Scheduler Configuration

After conducting research in BizTalk based technology, we assumed to make use of http protocol purely as a BizTalk orchestration activation was quite difficult. To deal with this problem, our own scheduler program was created. Other schedulers options available are by using Windows Task Scheduler or Scheduler Adapter [8], the Scheduler Adapter itself actually is a BizTalk's

custom adapter was made by Microsoft Codeplex Open Source Community [7]. Our scheduler program is mainly to drop an XML file to a specified folder in such any scheduled timely interval. The folder location itself will act as BizTalk's receive port which was being defined using BizTalk's File Adapter.

### 3 RESULTS

These are results in BizTalk testing, all tests were conducted using .asmx technology stack and NHibernate.

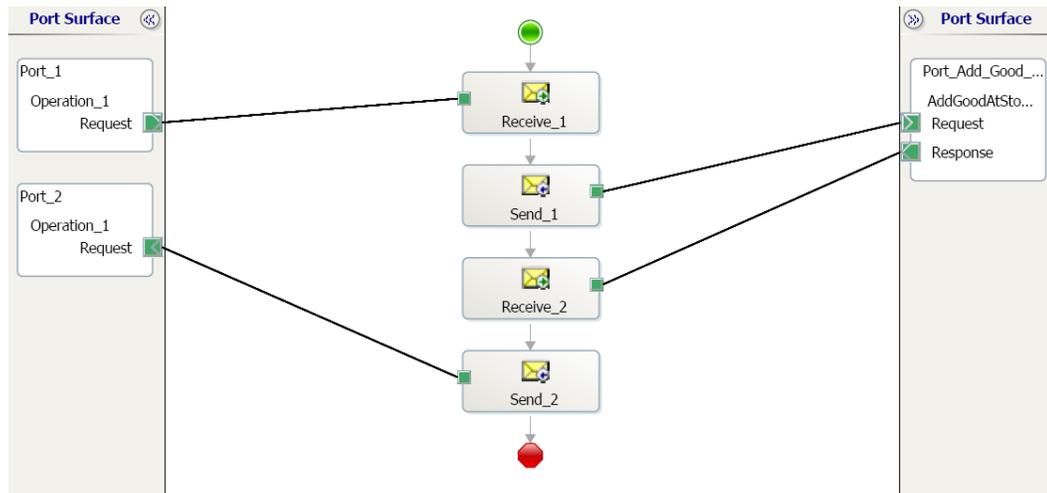


Figure 4 Add Good at Store Orchestration

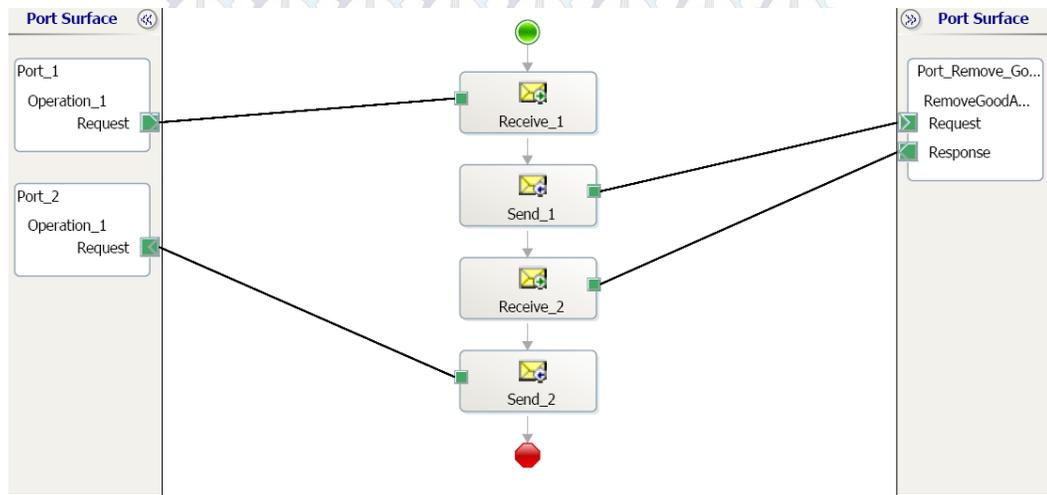


Figure 5 Move Good at Store Orchestration

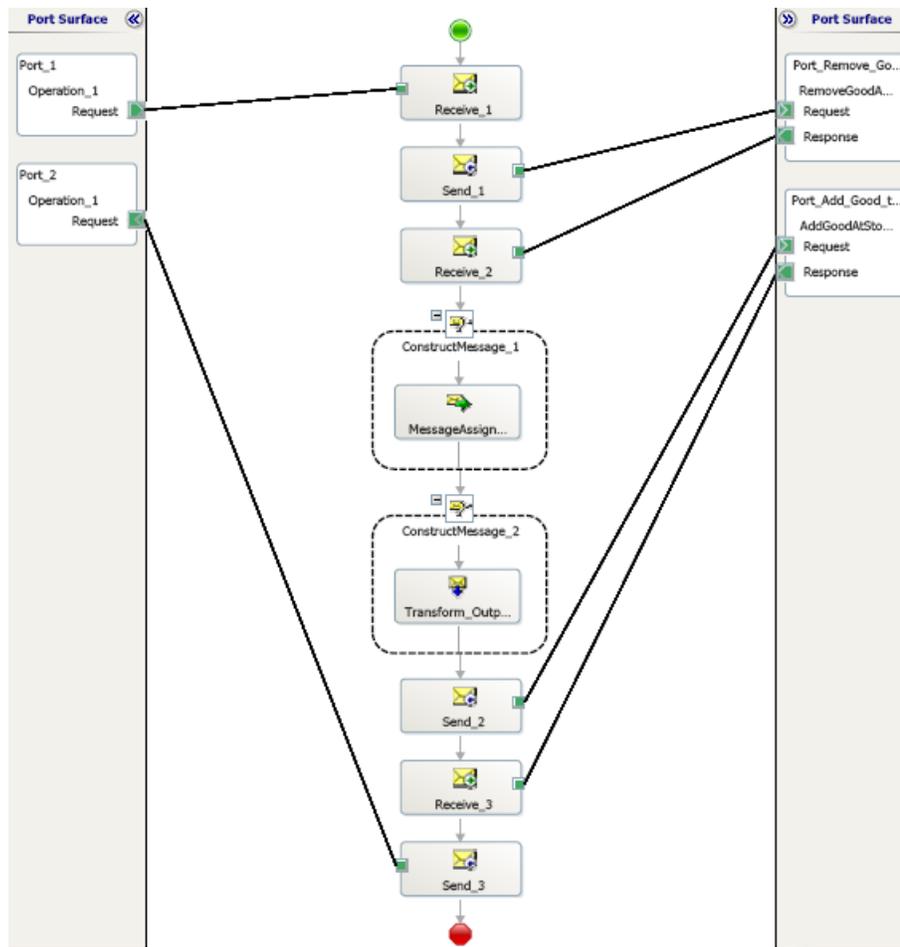


Figure 6 Remove Good at Store Orchestration

### 3.1 Result in Add Good at Store Orchestration

The Add Good at Store orchestration as shown in figure 4 were successfully doing its function, and written here is BPEL document which was exported and extracted as result of orchestration.

```
<?xml version="1.0"?>
<process xmlns:xsd=
"http://www.w3.org/2001/XMLSchema"
xmlns:q1="AddGoodModule"
xmlns:wSDL=
"http://schemas.xmlsoap.org/wSDL/"
name="AddGoodAtStore.
AddGoodAtStoreClient"
targetNamespace="AddGoodOrchestra"
xmlns=
```

```
"http://schemas.xmlsoap.org/ws/
2003/03/business-process/">
<partnerLinks>
<partnerLink name="Port_Add"
partnerLinkType="q1:Port_AddType"
partnerRole="portRole" />
<partnerLink name="Port_1"
partnerLinkType="q1:Port_1Type"
myRole="portRole" />
<partnerLink name="Port_2"
partnerLinkType="q1:Port_2Type"
partnerRole="portRole" />
</partnerLinks>
```

```

<variables>
<variable name="IN"
messageType=
"q1:AddGoodAtStoreMethodSoapIn"/>
<variable name="OUT"
messageType=
"q1:AddGoodAtStoreMethodSoapOut"/>
</variables>
<sequence>
<receive partnerLink="Port_1"
portType="q1:PortType_1"
operation="Operation_1"
variable="IN" createInstance="yes" />
<invoke partnerLink="Port_Add"
portType="q1:AddGoodAtStoreSoap"
operation="AddGoodAtStoreMethod"
inputVariable="IN"
outputVariable="OUT" />
<invoke partnerLink="Port_2"
portType="q1:PortType_2"
operation="Operation_1"
inputVariable="OUT" />
</sequence>
</process>

```

### 3.2 Result in Remove Good at Store orchestration

The Remove Good at Store orchestration as shown figure 5 were successfully doing its function. Below here is BPEL document which was exported and extracted as result of orchestration.

```

<?xml version="1.0"?>
<process xmlns:xsd=
"http://www.w3.org/2001/XMLSchema"
xmlns:q1="RemoveModule" xmlns:wSDL=
"http://schemas.xmlsoap.org/wSDL/"
name="RemoveBz.
RemoveGoodAtStoreClient"
targetNamespace="RemoveOrchestra"
xmlns="http://schemas.xmlsoap.org/
ws/2003/03/business-process/">
<partnerLinks>
<partnerLink name="Port_Remove"
partnerLinkType=
"q1:Port_RemoveType"
partnerRole="portRole" />
<partnerLink name="Port_1"
partnerLinkType="q1:Port_1Type"
myRole="portRole" />
<partnerLink name="Port_2"
partnerLinkType="q1:Port_2Type"
partnerRole="portRole" />
</partnerLinks>

```

```

<variable name="IN" messageType=
"q1:
RemoveGoodAtStoreMethodSoapIn"/>
<variable name="OUT" messageType=
"q1:
RemoveGoodAtStoreMethodSoapOut"/>
</variables>
<sequence>
<receive partnerLink="Port_1"
portType="q1:PortType_1"
operation="Operation_1"
variable="IN"
createInstance="yes" />
<invoke partnerLink="Port_Remove"
portType="q1:RemoveGoodAtStoreSoap"
operation="RemoveGoodAtStoreMethod"
inputVariable="IN" outputVariable="OUT" />
<invoke partnerLink="Port_2"
portType="q1:PortType_2"
operation="Operation_1"
inputVariable="OUT" />
</sequence>
</process>

```

### 3.3 Result in Move Good orchestration

It was unsuccessful thing happens in Move Good orchestration which is shown in figure 6. The orchestration was unable in doing its function properly. We were tracing the message through its orchestration, and we found out the message was stopped at Remove Good at Store step, but it was able to produce output message from the Web Service / SOA *Remove Good* which was being consumed and then stopped. No BPEL was able being produced from this orchestration, since BizTalk raised error message while we were trying to export the orchestration to BPEL.

## 4 CONCLUSION AND DISCUSSION

After our experiments were being conducted, we decide to collect BPEL's tags / syntaxes, they would be useful for future reference. And we concluded Microsoft Biztalk's Orchestration is not 100 % BPEL compatible, it is being designed for XLANG.

## REFERENCES

- [1] Activebpel (2008) Activebpel [Online]. Available at: <http://www.activebpel.org/download/download.php> [Accessed: February 2010].
- [2] Billy M (2008). Nhibernate best practices with asp.net, 1.2nd ed [Online]. Available at <http://www.codeproject.com/KB/architecture/NHibernateBestPractices.aspx> [Accessed: February 2010].
- [3] Bruce B (2008) WF Windows Workflow in .NET 3.5. Apress: Berkeley, CA
- [4] Chris P (2003) Web services orchestration and choreography [Online]. Available at <http://soa.sys-con.com/node/39800> [Accessed: February 2010].
- [5] Grady B (1998) Object-oriented Analysis and Design with Applications, vol. 2nd Edition. Addison-Wesley: Santa Clara - California.
- [6] Karelitis C, Vassilakis C and Georgiadis P (2007) Enhancing bpel scenarios with dynamic relevance based exception handling. In ICWS 2007. Proceedings of IEEE International Conference on Web Services on Salt Lake City, UT, 9 – 13 July 2007, *ieee.org*, pp. 751 – 758.
- [7] Microsoft (2006-2010) Codeplex - open source project hosting [Online]. Available at <http://www.codeplex.com/> [Accessed: February 2010].
- [8] Microsoft (2008) Biztalk scheduled task adapter [Online]. Available at <http://biztalkscheduledtask.codeplex.com/>, [Accessed: February 2010].
- [9] Microsoft (2009) How to export bpel4ws [Online]. Available at [http://msdn.microsoft.com/en-us/library/aa559882\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/aa559882(BTS.10).aspx), [Accessed : February 2010].
- [10] Microsoft (2009) Xlang/s language [Online]. Available at [http://msdn.microsoft.com/en-us/library/aa577463\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/aa577463(BTS.10).aspx), [Accessed : February 2010].
- [11] Microsoft (2010) Windows workflow foundation [Online]. Available at <http://msdn.microsoft.com/en-us/netframework/aa663328.aspx>, [Accessed: February 2010].
- [12] Microsoft (2009) Xlang/s to bpel4ws type conversions [Online]. Available at [http://msdn.microsoft.com/en-us/library/aa577842\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/aa577842(BTS.10).aspx), [Accessed: February 2010].
- [13] Mirko V and Enrico D and Alessandro R (2007) Engineering a bpel orchestration engine as a multi-agent system. Elsevier - Electronic Notes in Theoretical Computer Science 66: 226-245.
- [14] Nicolai M. (2007) SOA in Practice - The Art of Distributed System Design. OReilly: Sebastopol, CA.
- [15] OASIS (-) Oasis contributors [Online]. Available at <http://www.oasis-open.org/about/contributors.php>, [Accessed: February 2010].
- [16] OASIS (-) Oasis foundational sponsors [Online]. Available at [http://www.oasis-open.org/about/foundational\\_sponsors.php](http://www.oasis-open.org/about/foundational_sponsors.php), [Accessed: February 2010].
- [17] Object Management Group (2010) Bpmn information home [Online]. Available at: <http://www.bpmn.org/>, [Accessed : February 2010].
- [18] Parasoft (2010) Bpel engine toolkit: Bpel maestro - parasoft [Online]. Available at <http://www.parasoft.com/jsp/products/bpel.jsp>, [Accessed: February 2010].
- [19] Red Hat (2009) hibernate.org - nhibernate for .net [Online]. Available at <https://www.hibernate.org/343.html>, [Accessed : February 2010].
- [20] Riyanarto S and Anisah H (2010) Developing information technology policies for enterprise resource planning to improve customer orientation and service. IJCSNS International Journal of Computer Science and Network Security 10: 82 – 93.
- [21] Riyanarto S and Anisah H (2010) A service portfolio for an enterprise resource planning. IJCSNS International Journal of Computer Science and Network Security 10: 144 – 156.
- [22] Sparx S (2000-2010) Uml tools for software development and modelling - enterprise architect uml modeling tool [Online]. Available at <http://www.sparxsystems.com/>, [Accessed: February 2010].
- [23] Thomas E (2007) Ws-bpel 2.0 primer [Online]. Available at <http://www.oasis-open.org/committees/download.php/23964/ws-bpel-v2.0-primer.htm>, [Accessed: February 2010].