

# A Hybrid Service Orchestration Framework based on BPEL4WS

Sung-Min Kim, *Member, KIMICS*

**Abstract**— As the value of B2B integration gets more importance and the business process becomes more complex, BPEL4WS standard has been focused for the solution of the problems which is based on web service technology. Many vendors implement the standard in their systems to integrate B2B and web service based inner-Enterprise Application(EA)s. But there is no way to use the standard to integrate non-web service based EAs, because the standard is designed only for Service Oriented Architecture (SOA) over web services. This paper proposes Hybrid Service Orchestration Framework to participate web services and also non-web service based EAs in the business process as a partner using BPEL4WS standard. The framework introduces Hybrid Service Architecture to represent non-web service based EAs as a service, so BPEL4WS could be used to integrate web services and non-web service based EAs. And the framework also introduces Hybrid Service Toolkit to support software level facilities. The paper also introduces Hybrid Orchestration System that is based on Hybrid Service Orchestration Framework.

**Index Terms**— Hybrid Service, Web Service, BPEL4WS, EAI, B2B Integration, Web Services Architecture

## I. INTRODUCTION

AS the value of B2B integration gets more importance and the business process becomes more complex, web service based on BPEL4WS[1] standard has been focused for the solution of the problems. The most recent practices of BPEL4WS are Oracle BPEL

Process Manager and IBM WebSphere Business Integration Server Foundation for the commercial use. ActiveBPEL Engine and Twister are the open source ones. And also EAI vendors implement BPEL4WS or its own specification for the web service integration facilities. The example of orchestration system is based on EAI[2].

Web Service is originally designed for B2B communications. In the same context, BPEL4WS is designed for B2B integrations, otherwise EAI is generally used in inner-EA integrations. But nowadays web service has been using as inner-EA and the boundary of B2B integration and EAI is breaking up for many EAI systems ([2], [3], [4]) are getting facilities of web service integration. So there is a need to integrate EAs and web services without boundaries. But the BPEL4WS, integration standard cannot support EAs for the standard is designed only for web services.

The paper proposes Hybrid Service Orchestration Framework to integrate web services and even EAs using standard integration specification, BPEL4WS. The framework involves Hybrid Service Architecture to represent EA as service in the business process and Hybrid Service Toolkit to support software level facilities that is to provide Hybrid Service to 3rd parties and to use Hybrid Services. The paper also introduces Hybrid Orchestration System that is based on standard BPEL4WS to integrate web services and EAs.

The next section introduces related technologies. Section 3 specifies Hybrid Service Architecture. Section 4 presents Hybrid Service Toolkit based on the architecture. Section 5 introduces Hybrid Orchestration System as a sample usage scenario of the Hybrid Service Orchestration Framework. Finally, the conclusion section presents some results and further developments.

---

Manuscript received August 8, 2009 ; Revised September 4, 2009.

Sung-min Kim is with the School of Computer Science and Engineering, Konkuk University, Seoul, 143-701, Korea (Email: Semih.Kim@gmail.com)

## II. RELATED TECHNOLOGIES

There are 3 types of systems to integrate web services. First one is web services integration system that does not based on BPEL4WS standard. Second one is pure implementation of BPEL4WS standard. Last one is also the implementation of BPEL4WS but based on EAI system.

### A. EAI with Web Services Integration

There is a type of systems to integrate EAs and web services. One of it is BEA Systems WebLogic Integration 8.1[3]. It supports EA integration but also web service. In fact it is a typical EAI system supporting web service protocol. As like the case of BizTalk 2004, it uses the other specification[6] to describe a business process that is non standard.

### B. Non-EAI based Orchestration Systems

An example of non-EAI based orchestration system is Oracle BPEL Process Manager[5]. The system is the BPEL4WS implementation running on J2EE platform. It fully supports BPEL4WS specification also extends some facilities. It is the ideal solution to integrate web services. However there is no way to integrate EAs.

### C. EAI based Orchestration Systems

An example of EAI based orchestration system is Microsoft BizTalk 2004[2]. It is an orchestration system because it supports BPEL4WS specifications. It also supports BPEL4WS facilities just like the above one. However EAs cannot be participated in the business process when the business process is described using BPEL4WS. It only integrates web services when using BPEL4WS. Because there is no way to represent EA as business partner of BPEL4WS business process. There is a way to participate web services and EAs in a business process. But it does not using BPEL4WS standard, it uses the other specification[6] instead which is not an integration standard.

## II. HYBRID SERVICE ORCHESTRATION FRAMEWORK

Hybrid Service Orchestration Framework involves Hybrid Service Architecture which is to represent EA as service and Hybrid Service Toolkit which is to support software level facilities. Web service and Java RMI is used as an example to explain the framework.

### A. Hybrid Service Architecture

The concept of the web service is used to describe BPEL4WS business process. For example, URL of WSDL, name of PortType is used to describe partner link type and name of Message, Message Part is used to describe message property. W3C defines the concepts on Web Services Architecture (WSA)[7] and WSDL specification[8]. Hybrid Service Architecture is a subset of WSA and WSDL specification to represent web service and EA as a service on the BPEL4WS business process. There are two models in Hybrid Service Architecture for BPEL4WS as Fig. 1 and Fig 2 depicts.

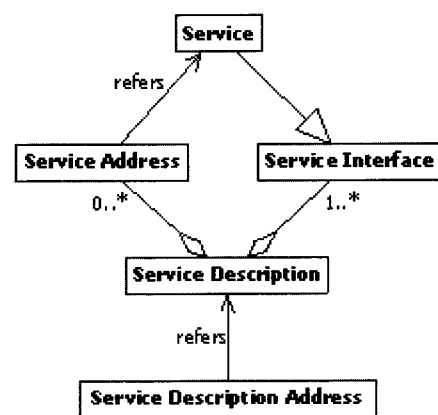


Fig. 1 Hybrid Service Oriented Model.

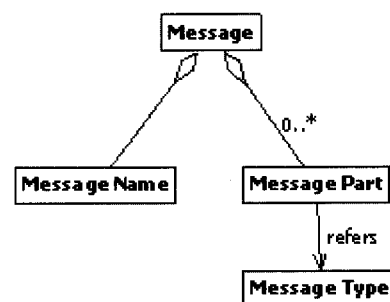


Fig. 2 Hybrid Message Oriented Model

### B. Hybrid Service Remodeling for Web Service and Java RMI

EA's architecture must be refined based on Hybrid Service Architecture to represent it as service. Redefining EA's architecture to Hybrid Service Architecture is called Hybrid Service Remodeling. There will be a remodeling per EA architectures. The example of web service and Java RMI remodeling is described on Table. 1.

In the case of Java RMI *Remote Object*, *Remote Interface*, *Method*, *Method Name* and *Method Parameter Types* are remodeled to *Service*, *Service Interface*, *Message*, *Message Name* and *Message Type*. But there is no concept like a *Service Description* and *Message Part*. So to remodel the Java RMI Architecture to the Hybrid Service Architecture policy for *Message Part* is defined and *Service Description* for Java RMI is defined.

Table 1 Hybrid Remodeling for Web Service and Java RMI

Hybrid Service Architecture	Web Service Architecture	Java RMI Architecture
Service	Web Service	Remote Object
Service Interface	Port Type	Remote Interface
Service Description	WSDL	N/A
Service Address	In WSDL	Implied
Service Description Address	Implied	Implied
Message	Message	Method
Message Name	Message Name	Method Name
Message Part	Message Part	N/A
Message Type	Message Type	Method Parameter Types

In the case of web service, *Message Part* acts like a overriding facility in Java RMI. So *Message Part* for Java RMI is used to distinct overridden methods.

**Example.java**

```
public interface Example extends Remote {
    int add(int v1, int v2)
        throws RemoteException;

    int add(int v1, int v2, int v3)
        throws RemoteException;
}
```

The first method's *Message Part* becomes 'int, int' and second one becomes 'int, int, int'. Because in Java specification, overridden methods are districted by its parameter types.

In the definition of Hybrid Service Architecture the *Service Description* must provide the information of *Service Address* and *Service Interface*. In the case of web service WSDL, a *Service Description* is like the below example. WSDL defines *Service Interface* as a human readable XML document and *Service Address* is in the address element.

**purchase.wsdl**

```
<definitions ... >
  <!-- omitted -->

  <portType name="purchaseOrderPT">
    <!-- omitted -->
  </portType>

  <!-- omitted -->

  <service name="purchaseOrderService">
    <port name="purchaseOrderPort"
      binding="tns:purchaseOrderBinding">
      <soap:address
        location=
        " http://manufacturing.org/wsdl/purchase"/>
      </port>
    </service>
  </definitions>
```

A simple example of *Service Description* for Java RMI is like this.

**ship.sd**

```
<ServiceDescription>
  <ServiceType>
    Java RMI
  </ServiceType>
  <ServiceInterfaceURL>
    http://subdueit.org/koc/rmi/shipping.jar
  </ServiceInterfaceURL>
  <ServiceURL>
    rmi://subdueit.org/ShippingOrder
  </ServiceURL>
</ServiceDescription>
```

Java RMI did not define *Service Interface* as like the web service do but defines Java object that is a binary format. So in the *Service Description* of the Java RMI *Service Interface* URL references *Service Interface*. And the *Service Address* uses **RMI** protocol rather than **HTTP** to represent *Service* is not a web service but Java RMI.

### IV. HYBRID SERVICE TOOLKIT ARCHITECTURE

Hybrid Service Architecture is for represent EAs as service. But Hybrid Service Toolkit is for use EAs as service. Hybrid Service Toolkit uses J2EE Connector Architecture (JCA)[9] to support common API to a client.

#### A. Hybrid Service Toolkit Usage Patterns

The JCA provides a Java technology solution to the problem of connectivity between the many EA servers and enterprise information systems (EIS). And it also defines standard contracts that allow bi-directional connectivity between EAs and EISs. JCA is not a SOA based technology but it is the best choice to integrate applications for so many vendors already supports JCA Adapters for their own systems and it also be used to access web services.

Hybrid Service Toolkit proposes two usage patterns. First one is called Service Usage Pattern for it supports to access Hybrid Services. And second one is called Service Providing Pattern for it supports to provide Hybrid Services to 3rd party. Fig. 3 and Fig. 4 depict the patterns.

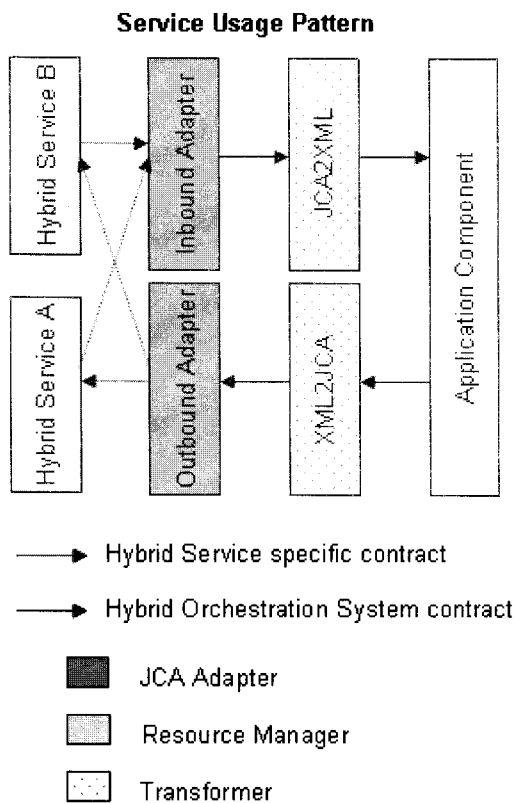


Fig. 3 Hybrid Service Toolkit Usage Patterns – Service Usage Pattern

#### Service Providing Pattern

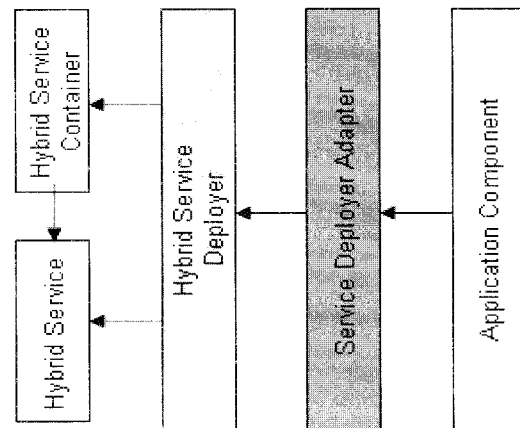


Fig. 4 Hybrid Service Toolkit Usage Patterns – Service Providing Pattern

#### B. Service Usage Pattern

Service Usage Pattern is a pattern that is to use Hybrid Service with Hybrid Service Toolkit. *Application Component* accesses Hybrid Services by *Outbound Adapter* using Common Client Interface (CCI) and it receives asynchronous messages by *Inbound Adapter*. Hybrid Service Toolkit also transforms outgoing, incoming records with *JCA2XML* and *XML2JCA Transformer* for Hybrid Service Architecture defines a Message as a XML formatted message. Vector supported JCA Adapters can be used directly in Service Usage Pattern. Hybrid Service Toolkit currently provides Java RMI and Web Service JCA Adapter.

#### C. Service Providing Pattern

Service Providing Pattern is a pattern that to provide Hybrid Service to 3rd parties with Hybrid Service Toolkit. *Application Component* requests to create and deploy a service to *Service Deployer Adapter* and *Hybrid Service Deployer* actually create and deploy the service. *Application Component* is responsible for service's logic. Hybrid Service Toolkit defines *Service Deployer Adapter's* Interaction API and provides Java RMI and Web Service Deployer Adapter.

### V. HYBRID ORCHESTRATION SYSTEM

This section introduces a Hybrid Orchestration System that is based on Hybrid Service Orchestration Framework.

Hybrid Orchestration System Architecture is depicted Fig. 5. This system uses standard BPEL4WS to integrate web services and also EAs. The system is based on

Hybrid Service Toolkit to integrate Hybrid Services and to provide Hybrid Service to 3rd parties.

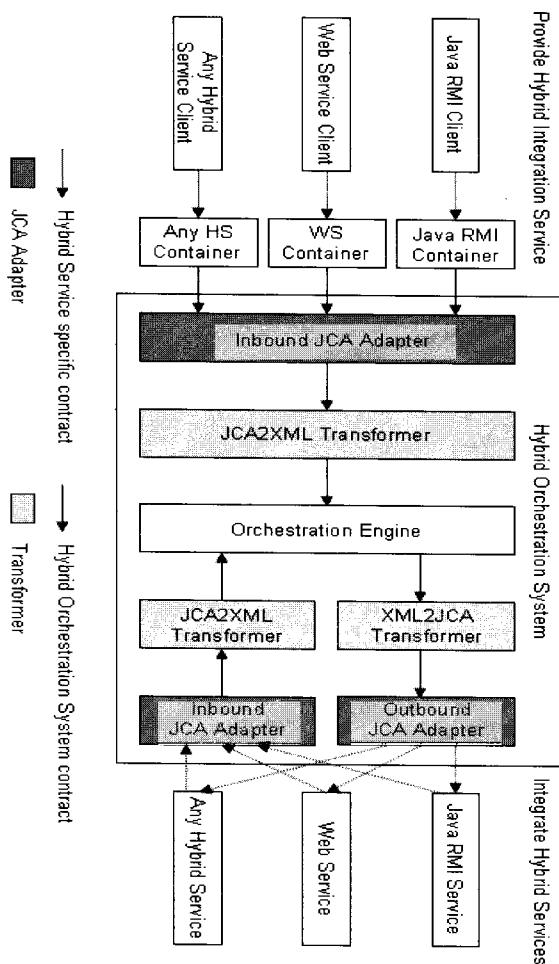


Fig. 5 Hybrid Orchestration System Architecture

## VI. CONCLUSIONS AND FUTURE WORKS

This paper proposes Hybrid Service Orchestration Framework which consists of Hybrid Service Architecture and Hybrid Service Toolkit. Using Hybrid Service Architecture, it's possible to represent EA as a service. That means every EAs could be treated like web services, because the EAs are represented as services. And Hybrid Service Toolkit is the software enabler to use EA as a service. With Hybrid Service Orchestration Framework, web services and non-web service based EAs could be participated in a BPEL4WS business process which is the standard of application integration.

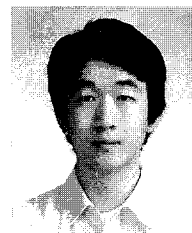
Hybrid Service Architecture is a subset of WSA and WSDL specification. And it is used to represent EAs

as service. To represent EA as service, EA architecture is remodeled to Hybrid Service Architecture which is called Hybrid Service Remodeling. And remodeled EA can be participated in BPEL4WS business process as partner.

Hybrid Service Toolkit provides common interfaces and some adapters to use and provide Hybrid Service. It uses inbound/outbound JCA Adapter to service type independent communications. And it also defines Hybrid Service deployment Interface to provide a service to 3rd party. At the last of the paper introduced Hybrid Orchestration System that is based on Hybrid Service Orchestration Framework. It can integrate various services and provide the integration services to 3rd party.

## REFERENCES

- [1] IBM et al., "Business Process Execution Language for Web Services version 1.1", <http://www-128.ibm.com/developerworks/library/specification/BPEL4WS/>
- [2] Microsoft, "BizTalk 2004", <http://www.microsoft.com/biztalk/>
- [3] BEA Systems, "BEA WebLogic Integration 8.1", <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/integrate/>
- [4] IBM, "WebSphere MQ", <http://www-306.ibm.com/software/integration/wmq/>
- [5] Oracle, "Oracle BPEL Process Manager", <http://www.oracle.com/technology/products/ias/bpel/index.html>
- [6] Satish Thatte, "XLANG", [http://www.gotdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm)
- [7] W3C, "Web Services Architecture", <http://www.w3.org/TR/ws-arch/>
- [8] W3C, "Web Services Description Language (WSDL) 1.1", <http://www.w3.org/TR/wsdl>
- [9] Sun Microsystems, "J2EE Connector Architecture Specification Version 1.5", 2003.



**Sung-min Kim**

received the B.S., M.S. in computer science and engineering from Konkuk Univ., Seoul, Korea. His research interests include Semantic Web and Future Internet.