

Accelerate your Enterprise SOA
by Leveraging the Application
Integration Architecture
Foundation Pack

An Oracle White Paper
December 2007

Accelerate Your Enterprise SOA by Leveraging the Application Integration Architecture Foundation Pack

Introduction	3
Business Trend.....	3
The IT Challenge	3
The Solution: Application Interoperability using the Oracle Application Integration Architecture Foundation Pack.....	4
Reduce Total Cost of Ownership	4
Proven Methodology Mitigates Risks.....	4
Accelerate your Enterprise Service Oriented Architecture	4
Foundation Pack Components.....	5
Enterprise Business Objects	5
Enterprise Business Services	5
SOA Governance.....	5
Business Service Repository	5
Composite Application Validation System.....	5
Composite Application Error Management and Resolution	6
Reference Architecture	6
Powered by Fusion Middleware.....	6
Sample USE case	7
Service-oriented architecture Approach to Application Interoperability..	8
Standards: Enterprise Business Objects	9
Abstraction: Enterprise Business Services	10
Loosely Coupled Applications: Application Business Connector Services for Application-Specific Tasks	11
Extensibility.....	12
Extending an Enterprise Business Object	13
Extending a Business Process.....	13
Extending an Enterprise Business Service	14
SOA Governance	15
Service Visibility and Impact Analysis using Business Service Repository	15
SOA Quality	15
Error Management and Resolution	16
Diagnostics	16
Summary	17

INTRODUCTION

This whitepaper will discuss how you can use Oracle Application Integration Architecture Foundation Pack to integrate your heterogeneous application portfolio to build cross-functional business processes. It will outline how the Foundation Pack will help you overcome the most common yet critical application integration challenges by providing you with a set of prebuilt enterprise objects and services, an application integration management infrastructure and a proven application integration methodology. The Foundation Pack with its standardized enterprise business interfaces can be the core of your enterprise architecture strategy by letting you focus on doing business the way you want to do business. You can start using it today in your current integration projects to realize an immediate ROI. The Foundation Pack ensures that you are building your IT systems in line with enterprise architecture that is focused on meeting immediate needs in a way that will also meet your future requirements. It provides an incremental result-based approach to strategic enterprise architecture than will transform IT from being tactical and responsive to being more strategic and adaptable.

Business Trend

The pace of business is changing rapidly. Global markets have led to increased competition, a changing customer, and the need to bring new products and services to market quickly. As a result, companies have to change and adapt their business models quickly. IT has to be able to not only support the changing business needs but also be able to do this rapidly. In this increasingly competitive and global marketplace, a company's IT department can no longer afford to be just a cost center, but has to transform itself and be innovative to provide the business with a competitive advantage.

The IT Challenge

A typical IT environment today is characterized by a myriad of applications from different vendors. Businesses require that these applications work together to produce meaningful results. An example of this is an Order to Cash business process that has to bring together different applications like order capture, fulfillment, shipping, planning, and invoicing. In many cases, these business processes may also have to interact with third party systems from partners. Each of these applications were designed using different technologies and more often than not were intended to solve an entirely different business problem. Integrating these applications presents a tremendous challenge. Such integration projects are often very time consuming, costly, and difficult to maintain. This complexity presents a major challenge in dealing with new business requirements, evolving existing applications, and simply keeping up with the maintenance and enhancements of existing integrations.

Today these systems are often wired together with rigid point-to-point integrations that make it very difficult to respond and adapt to rapidly changing business conditions. Add to this the fact that the IT department has to maintain a tight grip

on costs. So how does IT transform itself to be agile and adaptable and yet deliver more with less?

THE SOLUTION: APPLICATION INTEROPERABILITY USING THE ORACLE APPLICATION INTEGRATION ARCHITECTURE FOUNDATION PACK

The Oracle Application Integration Architecture Foundation Pack heralds a new approach to integrating your application portfolio using composite business processes. It combines the power of Oracle's Fusion Middleware along with a set of best in class application Enterprise Business Objects and Services that form the building blocks for a new generation of composite applications that leverage your existing investments to fulfill mission critical business processes.

Reduce Total Cost of Ownership

The Foundation Pack consists of a pre-built set of Enterprise Business Objects and Services, an integration management infrastructure, and a proven methodology that significantly lowers your cost of ownership and provides a faster time to value for new composite business processes. The Application Integration Architecture extensibility model, combined with the power of Oracle Fusion Middleware SOA Suite technologies enable customers to design loosely coupled, adaptable, and sustainable integrations that meet their unique requirements, while at the same time they can also be modified quickly with less time and resources than traditional integrations.

Proven Methodology Mitigates Risks

The Foundation Pack delivers the same methodology that Oracle uses to deliver its pre-packaged integrations between Oracle Applications. This methodology mitigates your integration risks and costs by providing a set of proven best practices and design principles to build these mission-critical business processes. Using the Foundation Pack to build your custom integration ensures that it is interoperable with other pre-packaged integrations from Oracle that may meet your business needs. So it's not a question of "build vs. buy," it is about providing customers with the power to choose what fits their business needs while still having a consistent approach to enterprise architecture that reduces their risks and costs.

Accelerate your Enterprise Service Oriented Architecture

The concept of Service Oriented Architecture is widely understood among business and IT circles and almost everyone agrees with the benefits of SOA—it just makes sense. However, the question everyone's asking is how do I get to SOA? Foundation Pack with its set of standardized pre-built business services, SOA governance tools and methodology document provides a jumpstart in your enterprise SOA projects. The path to enterprise SOA is evolutionary. The Foundation Pack provides an incremental results-oriented approach to your enterprise architecture strategy. You can start leveraging it in your current integration projects today to realize immediate ROI. This allows you to focus and

"It is apparently easier to say that a firm will adopt SOA than it is to make specific plans and follow through on them."

**Randy Heffner, "Planned SOA Usage Grows Faster Than Actual SOA Usage"
Forrester, March 2007**

address your current integration challenges and puts you on a smooth path to a more enterprise SOA architecture in the future.

FOUNDATION PACK COMPONENTS

Oracle Application Integration Architecture Foundation Pack consists of the following components:

- Enterprise Business Objects
- Enterprise Business Services
- SOA Governance Tools
- Reference Architecture

Enterprise Business Objects

The term Enterprise Business Object (EBO) refers to a data model consisting of standard business data object definitions and reusable data components representing a business object such as Sales Order, Party, Item, Customer etc. It is the best in class definition of your business data, rationalized across Oracle's entire application portfolio and industry standards. The Enterprise Business Objects are delivered as XML Schema Definition (XSD) files.

Enterprise Business Services

Enterprise Business Services (EBS) are the foundation blocks in the Oracle Application Integration Architecture. Enterprise Business Services represent the application independent web service definition for performing a business task. It is self-contained, that is, it can be used independent of any other services. In addition, it can also be used within another Enterprise Business Service. Enterprise Business Services are standard business level interfaces that can be implemented by the applications that want to participate in the integration.

SOA Governance

The Foundation Pack includes a set of tools to manage and govern your entire integration lifecycle.

Business Service Repository

The Business Service Repository (BSR) acts as a catalog of the objects, messages, and services that compose the integration scenarios in your Oracle Application Integration Architecture ecosystem. Business Service Repository facilitates the key functions of service visibility, reuse, and impact analysis in your service lifecycle.

Composite Application Validation System

The Composite Application Validation System (CAVS) enables you to test integration web services without the deployed participating applications in place. Using a framework that includes initiators to simulate calls to participating applications and simulators to simulate responses from participating applications, CAVS provides a system that can test integrations, while also eliminating the need

to set up deployments of all participating applications that are involved in the integration.

Composite Application Error Management and Resolution

This is a framework that provides customers with a consistent way to manage errors in your integration layer. It provides the ability to route the error back to the correct application and to the right application user. This way, the exceptions can be resolved quickly with less downtime and IT can meet the stricter service level agreements that businesses need to compete in the global marketplace.

Reference Architecture

The reference architecture consists of the Concepts and Technologies Guide and an Integration Developer’s Guide, two pieces of comprehensive documentation to assist in integration development. The guides outline the Application Integration Architecture methodology and provide step-by-step guidance on how to design and develop your integrations. It also provides a list of best practices and design patterns that you can apply to your custom integrations.

Lack of skills/training is the leading SOA challenge
 Infoworld SOA Report 2007

POWERED BY FUSION MIDDLEWARE

Oracle Fusion Middleware SOA suite is the underlying technology for building integrations using the Oracle Application Integration Architecture Foundation Pack.

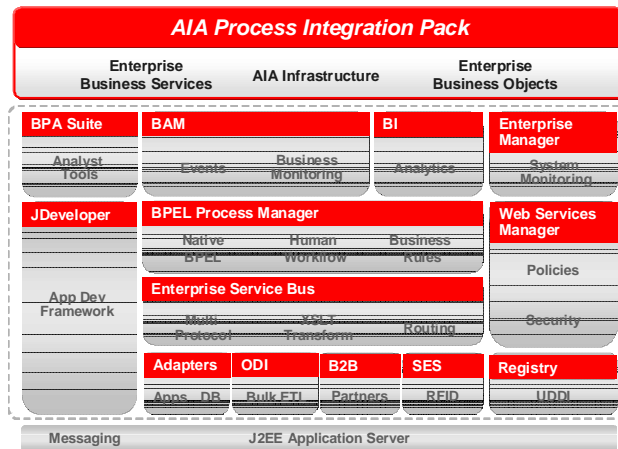


Figure 1: Application Integration Architecture runs on the best in class Oracle Fusion Middleware

The Fusion Middleware SOA Suite provides a best in class toolset to design and build your service-oriented integrations. For example, Oracle’s BPEL Process Manager is used for business process orchestration, the Enterprise Service Bus acts as the messaging backbone, the Oracle Web Services Manager is used to secure your integrations, and Oracle Business Activity Monitoring is used to monitor and optimize your business processes. This toolset forms the core of how Application Integration Architecture enables you to address your interoperability challenges. However, to build applications you not only need a rich toolset, but you also need a

methodology and a set of best practices that you can correctly apply to a given business problem. Oracle Application Integration Architecture provides you with a reference architecture that acts as your guiding principle.

SAMPLE USE CASE

It is quite easy to discuss terms in abstract, but the reality is unless you understand how a solution is designed and deployed, it can be difficult to conceptualize. As such we will use the following sample use case throughout this paper to illustrate how you can use the Oracle Application Integration Architecture Foundation Pack to build your own custom integrations to bring together a number of disparate applications, both internal to an enterprise and external in a Service Oriented way.

In this sample application, Global Company is a retail storefront for ordering electronic devices through a web-based client application. The following diagram outlines the company's business process for booking and fulfilling orders:

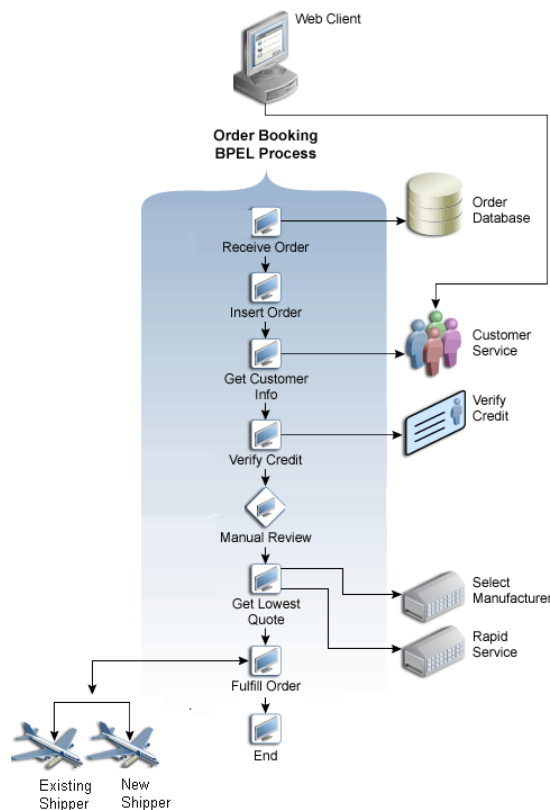


Figure 2: Order booking sample use case

When a new order is placed, the following occurs:

1. The web client sends a message to a JMS Queue which then routes the message to any service that has registered an interest in these messages. In this case, it is the order booking process.
2. This process sets the order to pending, and writes the order to the order database tables. The process then calls the customer service to retrieve

customer ID, name, address, and credit card information and checks the identified customer against the credit service to verify if the customer's credit card is valid. The credit service returns the relevant rating for the customer.

3. If credit is not approved, the process cancels the order or else it will move to the next step.
4. If the order is approved, it is sent to two suppliers for their price quotes. The process collects the quotes and selects the lowest quoted price as the supplier to which to award the order. The process then invokes the fulfillment service which will select the appropriate fulfillment provider based on a business rule.
5. After the order is fulfilled, the BPEL process sets the order to complete, and starts a notification service, which sends the customer an email with the purchase order information.

SERVICE-ORIENTED ARCHITECTURE APPROACH TO APPLICATION INTEROPERABILITY

To understand the value of Application Integration Architecture, it's important to look at the key principles associated with Service Oriented Architecture

- Standards – Compliance to both common and industry specific standards is required for interoperability between heterogeneous applications.
- Abstraction – Reusable, modular coarse-grained business services provide rapid composite application development and easier maintenance.
- Loosely coupled – Minimal dependencies with other applications offer durability and agility.

Oracle Application Integration Architecture is designed and developed on these first principles of SOA. Let's take our order booking sample use case that we described earlier and see how that business process translates into a SOA based Application Integration Architecture.

Application Integration Architecture schematic of sample use case

This diagram shows a technical representation of the order booking sample use case implemented using the Application Integration Architecture Foundation Pack: case:

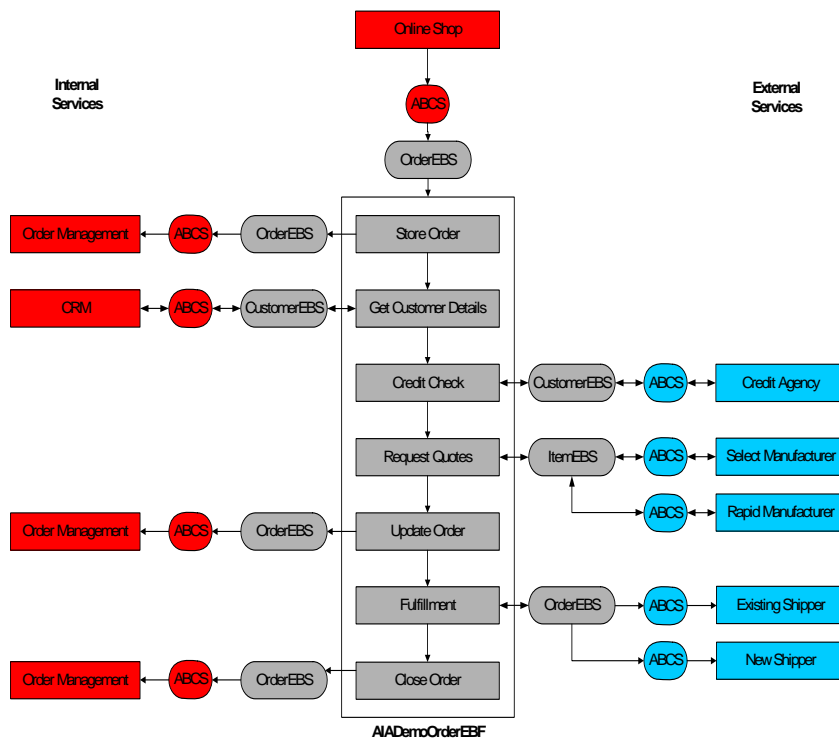


Figure 3: Order booking sample use case reference architecture

Standards: Enterprise Business Objects

One of the most common challenges in application interoperability is inconsistent business semantics among the different applications. Let's use the Sales Order business document from our sample use case. The definition of the Sales Order object in the order capture system, the online portal in this case, can be different from the way it is defined in the two fulfillment systems or the suppliers who may be using third party applications. The business objects can not only differ in terms of content but also in the naming rules and the meaning of the attributes in each of their systems. As the number of applications that you are integrating grows, this complexity grows exponentially.

Oracle Application Integration Architecture provides a prebuilt Enterprise Object Library that defines the best-in-class representation of business entities such as Sales Order, Purchase Order, Item, Invoice, etc. These definitions are rationalized across the entire Oracle application portfolio and industry standards.

This canonical definition of the object brings together disparate applications using a common language and drastically reduces the number of transformations required to integrate between different applications. In our sample use case, the online portal application now only has the responsibility to post the sales order message in the common Application Integration Architecture Enterprise Business Object format without worrying about how the order is represented in any of the other end systems, which are likely to change and evolve over time.

Enterprise Business Messages

Only 5% of the interface is a function of the middleware choice. The remaining is a function of application semantics

Gartner

An Enterprise Business Object is a logical representation of the business entity. An Enterprise Business Message on the other hand is the implementation of an Enterprise Business Object. The Enterprise Business Message is designed to be operation specific so that you don't have the overhead of passing the entire Enterprise Business Object to every service operation. For example, the CreateOrder service operation requires more attributes to be passed in the Sales Order EBM than a DeleteOrder operation which may require only a unique identifier. The Enterprise Business Message also contains an Enterprise Business Message header which has additional attributes that are used to provide robust auditing and exception management.

Abstraction: Enterprise Business Services

The abstraction layer in Application Integration Architecture is provided by the Enterprise Business Service. This service serves three key purposes:

- Hides service implementation complexity from the service requester. This way the service requesters can focus more on solving the business problem than dealing with integration challenges
- Exposes a service contract that is application independent. This means any application can invoke the service in the same way irrespective of who the service provider application is.
- Ensures contract first development which forces the organization to take a more strategic enterprise architecture approach than building a tactical point-to-point integration. It narrows the gap between IT and the business because the coarse-grained Enterprise Business Service is designed to meet business needs rather than be dictated by the complexity or nuances of the participating applications.

If we look at the architecture schematic for the order booking sample use case, you will notice that every service invocation is done through the Enterprise Business Service. So when the process needs customer details, it invokes the AIADemoCustomerEBS service and the GetCustomerParty operation in that service. Taking this approach ensures that if the legacy application that acts as a customer service provider today is upgraded to an enterprise CRM application tomorrow, the process doesn't have to change. The Enterprise Business Service interface remains the same. This enables customers to leverage their existing investments and evolve their IT systems at their own pace.

Following is an example of how an Enterprise Business Service is implemented using the Foundation Pack WSDL interfaces:

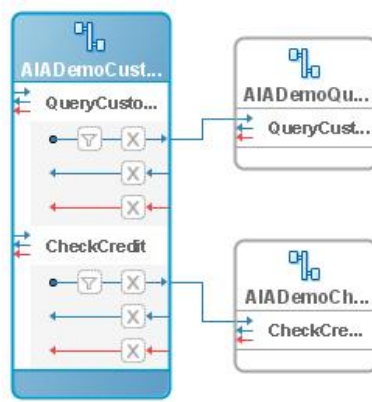


Figure 4: ESB implementation of the Customer Enterprise Business Service

The following table lists some of the EBOs and EBS used in the sample use case:

Enterprise Business Service (Enterprise Business Object)	Operation	Enterprise Business Messages
AIADemoCustomerEBS (CustomerPartyEBO)	QueryCustomerParty	QueryCustomerPartyReqMsg, QueryCustomerPartyRespMsg
	CheckCredit	CheckCreditReqMsg, CheckCreditRespMsg
AIADemoOrderEBS (SalesOrderEBO)	CreateOrder	CreateOrderReqMsg
	FulfillOrder	FulfillOrderReqMsg
AIADemoItemEBS (ItemEBO)	getSMItemPrice	getItemPriceReqMsg, getItemPriceRespMsg
	getRSItemPrice	getItemPriceReqMsg, getItemPriceRespMsg

Loosely Coupled Applications: Application Business Connector Services for Application-Specific Tasks

One of the key tenets of Service Oriented Architecture is to eliminate the dependency of service requesters on service providers. This means that businesses can change their systems without having to make reflective changes to other dependent systems. This provides customers with the agility to quickly bring new applications into the mix or upgrade to newer versions of existing systems without a major rip or replace approach.

In Application Integration Architecture, the Application Business Connector (ABC) services shield the Enterprise Business Service from being tightly coupled to the underlying applications. The Application Business Connector service does all of the application specific processing. As you can see in the sample use case, all of the service end points have an Application Business Connector service that does the transformation from the native object to the canonical format and vice-versa. The

Application Business Connector service is also responsible for doing any validations and data enrichment that may be required.

Following are some of the Application Business Connector Services implemented for the Order Booking sample use case:

Application Business Connector Service	Role	Participating Application or Service
AIADemoBookOrderReqABCServiceImpl	Requester	Online Shop
AIADemoCheckCreditProvABCServiceImpl	Provider	Credit Check Agency
AIADemogetRSItemPriceProvABCServiceImpl	Provider	Rapid Service
AIADemogetSMItemPriceProvABCServiceImpl	Provider	Select Manufacturer
AIADemoQueryCustomerPartyProvABCServiceImpl	Provider	CRM

Based on the integration design pattern, an Application Business Connector Service can either be implemented as an ESB or a BPEL service.

An Application Business Connector service can either have the requester or provider role in an integration scenario. The requester Application Business Connector Service AIADemoBookOrderReqABCServiceImpl, for instance, is invoked by the online shop application to process a new order. It then transforms the order into the Enterprise Business Message containing the SalesOrderEBO structure and invokes the AIADemoOrderEBS. On the other hand, a provider Application Business Connector service like AIADemoCheckCreditProvABCServiceImpl is called by AIADemoCustomerEBS. It receives the request Enterprise Business Message (including the CustomerPartyEBO) to execute the credit check for a particular customer. It transforms this into the format that the credit check application understands. The application answers itself with its specific format and the Application Business Connector service must take care to convert it back to the response Enterprise Business Message that the calling Enterprise Business Service AIADemoCustomerEBS expects.

EXTENSIBILITY

Implementations always have unique requirements, either specific to their business or specific to the industry. The ability to build extensions that are sustained and preserved across new releases and upgrades is one of the key features of the Oracle Application Integration Architecture. Every single Application Integration Architecture component can be extended. Let's use our order booking sample use case to illustrate how extensibility works.

All extensions to Enterprise Business Objects & Services are preserved across patches or upgrades

Extending an Enterprise Business Object

The Enterprise Business Objects are delivered as a set of XSD files. For every Enterprise Business Object, Application Integration Architecture also provides a custom XSD file in which all customer extensions are stored.

The SalesOrder Enterprise Business Object that is shipped by Oracle has Supplier Name and Supplier Price attributes defined at the line level. However, the third party system used by the company's supplier requires that these attributes be passed at the header level. To be able to integrate with those systems, the company's IT department will have to extend the Sales Order Enterprise Business Object to add these two new attributes at the header level.

To achieve this, we will extend the custom schema CustomSalesOrderEBO.xsd to add the additional attributes. As we want to add the attributes on the order header level, the following part of the schema definition needs to be changed:

```
<xsd:complexType name="CustomSalesOrderType"/>
```

After adding the attributes, this section of the schema definition looks like:

```
<xs:complexType name="CustomSalesOrderType">
  <xs:sequence>
    <xs:element name="SupplierName" type="xs:string"/>
    <xs:element name="SupplierPrice" type="xs:double"/>
  </xs:sequence>
</xs:complexType>
```

Having done this, the SalesOrderEBO is now ready to carry the custom attributes.

Note that the extension of the underlying SalesOrderEBO also extends all Enterprise Business Messages that reference the SalesOrderEBO. In our case, these are the Enterprise Business Messages CreateOrderReqMsg and FulfillOrderReqMsg. As the Enterprise Business Messages are also extended, the extended message definition also extends all Enterprise Business Services and Application Business Connector Services that work with these Enterprise Business Messages.

Extending a Business Process

The only business constant is change. Be it mergers and acquisitions, response to competitive threats, or changing business models, existing business processes and systems have to undergo change. Oracle Application Integration Architecture is designed to manage such changes with minimal disruptions to your existing infrastructure.

Let's take the example of our global company. Assume that they make a business decision to use another fulfillment provider, called New Shipper, along with their Existing Shipper. The following section illustrates how IT can bring another service provider into the mix with minimal impact to their existing order booking process.

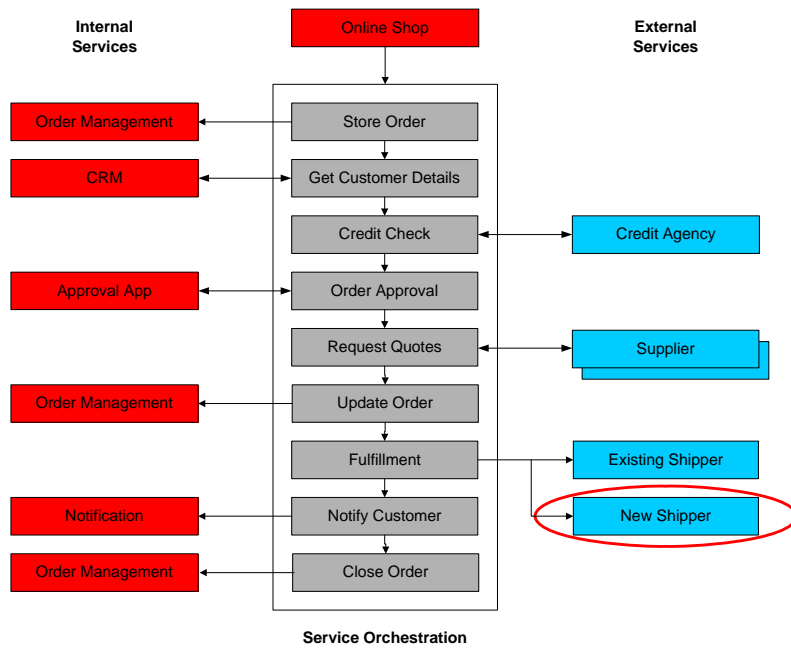


Figure 5: Bringing a new application into your existing integration flow

First, a new Application Business Connector Service will have to be implemented for the New Shipper service called AIADemoNewShipperProvABCImpl. The next step is to wire the AIADemoOrderEBS to the new application as shown below:

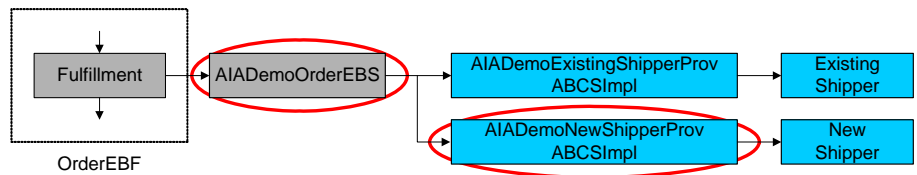


Figure 6: Adding an additional service provider (New Shipper) into the flow

Extending an Enterprise Business Service

The next step is to wire the AIADemoNewShipperShipmentProvABCImpl into the business flow. To achieve this, we need to extend the respective Enterprise Business Service, which is AIADemoOrderEBS in our case.

Enterprise Business Services are implemented with the Oracle Enterprise Service Bus (ESB). It is a fairly easy task to add a new target into the message routing.

The following simple XPath filter criteria will now route the orders to Existing Shipper if the amount is over \$500.

Existing Shipper: `/FulfillOrderEBM/Data.Area/FulfillOrder/Base/TotalAmount >=500`

New Shipper: `/FulfillOrderEBM/Data.Area/FulfillOrder/Base/TotalAmount <500`

This shows how easy it is to bring new applications into your composite business process design using the Application Integration Architecture with minimal disruptions to your existing integrations.

SOA GOVERNANCE

Service-Oriented Architecture provides a powerful way to implement your IT infrastructure. However, it also increases the number of inter-dependent moving parts in your IT system. To be successful with SOA, it is imperative that you have a way to manage and govern the entire lifecycle of the services from concept and design, to deployment and change. Application Integration Architecture Foundation Pack provides an application integration management framework that enables you to manage the entire lifecycle of your integrations.

Service Visibility and Impact Analysis using Business Service Repository

The business service repository allows you to answer the following three key SOA governance questions:

- How can I reuse a Service?
- What is the impact on my IT infrastructure if I change an existing service?
- How do I get visibility into how my integration process is implemented so that I can ensure full fidelity with my business process?

The Business Service Repository provides visibility into all of the Application Integration Architecture services in your IT ecosystem. You can view the Enterprise Business Objects and the Services that these objects are used in. The Business Service Repository also stores what is called an integration scenario. An integration scenario depicts the end-to-end integration process, and lists all of the service providers and requesters in that flow. This provides an enterprise architect visibility into how a particular integration has been deployed and all of the services that are used as part of that process.

SOA Quality

The most challenging aspect of composite applications is executing an end-to-end system test. There are different applications involved, they are distributed in nature, and have different interfaces. In many cases, these participating applications are unavailable for testing as you are building the integrations, which complicates the physical infrastructure required for testing.

The Composite Application Validation System (CAVS) is a tool to manage your SOA Quality. With Composite Application Validation System, customers can define test cases to test your entire integration flow or just parts of the integration. It provides a mechanism to simulate participating applications and does message level validation at each service level.

Let's take an example of how we can use the composite application validation system to test the Order Fulfillment service at design time without actually having

The Composite Application Validation System provides a quick way to test your integrations without the need to have all of the participating applications in place.

to direct a call to the external partners. To do this, let's define two test cases on the fulfillment operation within the AIADemoOrderEBS. The first order is for \$100 and will be shipped by New Shipper. The second order is for \$2000 and must be shipped by Existing Shipper according to the sample use case.

Two simulators are defined to simulate the behavior of the service providers Existing Shipper and New Shipper. In Composite Application Validation System, we can define a simulator by simply assigning it the expected incoming message. Additionally, we can define a result message (only if it simulates a synchronous service) and also XPATH expressions. These XPATH expressions are validated during execution in order to check if the incoming message follows expected rules.

This diagram shows a test and simulation scenario in Composite Application Validation System.



Figure 7: Test and Simulation Scenario in Composite Application Validation System

All of the Application Integration Architecture services are designed to route the messages to a simulator or to the actual service based on a configuration parameter. In this example, the Application Business Connector service will route the message to a simulator which will perform validation on each attribute in the message and generate a report file.

Error Management and Resolution

The operation and maintenance of a distributed composite application requires a robust error management and resolution system. The Application Integration Architecture Foundation Pack includes a unified and consistent approach for error handling and resolution across applications, technologies, and integration patterns. The Enterprise Business Message is enhanced with attributes that allow an error message to be routed to the correct application and application user. The error message is also published to a queue which enables a customer to kick off their own error management system if they have one.

Diagnostics

To ensure the integrity of an Application Integration Architecture based integration, Application Integration Architecture provides a rich set of diagnostic tests. These tests provide an efficient way to troubleshoot a problem in your integration ecosystem. These kinds of tests allow you to quickly check the consistency of the integration infrastructure, particularly after upgrades or patches to applications or to the integration infrastructure itself.

Application Integration Architecture Diagnostics lets you validate the consistency of your whole integration ecosystem at any time.

SUMMARY

The Application Integration Architecture Foundation Pack delivers everything what organizations need to implement state-of-the-art integrations to better support their business processes. Leveraging service-oriented concepts along with a sophisticated governance model provides the means to achieve and keep the agility and adaptability businesses require today and tomorrow.



Accelerate Your Enterprise SOA by Leveraging the Application Integration Architecture Foundation Pack
December 2007

Author: Sameer Phatarpekar
Contributing Authors: Gerhard Drasch

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only
and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to
any other warranties or conditions, whether expressed orally
or implied in law, including implied warranties and conditions of
merchantability or fitness for a particular purpose. We specifically
disclaim any liability with respect to this document and no
contractual obligations are formed either directly or indirectly
by this document. This document may not be reproduced or
transmitted in any form or by any means, electronic or mechanical,
for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.