Oracle Application Integration Architecture Foundation Pack for Oracle Applications: E-Business Suite, PeopleSoft, Siebel and JD Edwards

An Oracle White Paper
February 2009

ORACLE

AIA Foundation Pack for Oracle Applications: E-Business Suite, PeopleSoft, Siebel and JD Edwards

INTRODUCTION

This whitepaper will discuss how you can use Oracle Application Integration Architecture (AIA) Foundation Pack to integrate your Oracle and non-Oracle applications to build cross-functional business processes. It outlines how the Foundation Pack will help you overcome the most common yet critical application integration challenges by providing you with a set of pre-built 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 AIA Foundation Pack ensures that you are building and integrating your IT systems in line with enterprise SOA architecture practices that meet both immediate needs as well as anticipate your future business requirements. It provides an incremental result-based approach to strategic enterprise architecture that will transform IT from being tactical and responsive, to being more of a strategic and adaptable solution.

Business Trends

The pace of business is changing rapidly. Global markets have led to increased competition, changing customer demands, 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 while keeping costs under control. To accommodate the business, IT has to be able to not only support the changing business needs but also needs to do this rapidly and cost effectively. In this increasingly demanding global marketplace, a company's IT department can no longer afford to be just a cost center; IT must transform itself to a strategic business partner to deliver innovation and 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 which are owned by partner companies. Each of these applications was designed to solve a specific business problem (e.g. capture orders across multiple channels) rather than focusing on addressing an entire business process (e.g. order to cash). And most often, different technologies were used to develop these specific applications. Thus the integration challenge for IT is:

- How do I tie these different products together seamlessly, efficiently and rapidly?

- What happens when one of my applications gets upgraded?
- What happens if my business needs or processes changes?

Such integration projects are often very time consuming, costly, and difficult to maintain. IT departments often have to wire together multi-application environments using rigid point-to-point integrations, which makes their jobs very difficult when faced with new business requirements, evolving existing applications, introducing new applications, and simply keeping up with the maintenance and enhancements of existing integrations.

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

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

The Oracle Application Integration Architecture Foundation Pack heralds a new approach to integrating your application portfolio based on 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, pre-defined reference architecture, an integration governance infrastructure, proven methodology and best practice processes that significantly lowers your cost of ownership and provides a faster time to value for new composite business processes. The AIA Foundation Pack can be thought of as your very own "do-it-yourself kit" that includes the tools, templates and methodologies that you need to integrate, manage and evolve your applications and implementing best practices within your organization, without starting from scratch.

Using Foundation Pack, customers are able to design loosely coupled, adaptable, and sustainable integrations that meet their unique requirements, while at the same time delivering an infrastructure that can be modified quickly with less time and resources than traditional integrations.

Proven Methodology Mitigates Risks

The Foundation Pack delivers the same tools, templates and methodologies that are leveraged by Oracle when delivering its pre-packaged process integrations packs (PIPs) between Oracle's own Applications. Thus AIA Foundation Pack is proven and supported by Oracle. It mitigates your integration risks and costs by providing a set of proven best practices and standardized design principles to build 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 as well as non-Oracle applications, enabling you to leverage and

optimize existing investments. 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

"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

The concept of a 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 Enterprise Business Objects and Services, SOA governance tools, and methodology documents and templates provides a jumpstart to 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 on and address your current integration challenges while creating a smoother path to enterprise SOA 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 Process Models and Architecture
- Composite Application Framework

Enterprise Business Objects

The term Enterprise Business Object (EBO) refers to a SOA 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 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 define standard business level interfaces among 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, from development to run-time.

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 a web service and simulators to simulate responses from web service, 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 across technologies 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 Process Models and Architecture

Lack of skills/training is the leading SOA challenge

Infoworld SOA Report 2007

The reference process models are documented application independent business process models that act as a blueprint for mapping business processes from the task and activity levels down to the system level

The reference architecture consists of the Concepts and Technologies Guide, Business Process Guide, Enterprise Object Library Guide and an Integration Developer's Guide, four 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, develop and extend enterprise business processes, enterprise business objects and enterprise business services in your integrations. It also provides a list of best practices and design patterns that you can apply to your SOA based integrations.

Composite Application Framework

The Foundation Pack Composite Application Framework enables IT to deliver application independent, composite processes or user interfaces that are sustainable, repeatable and low-cost. These composite applications are loosely-coupled and insulated from changes in provider systems, enabling the end-user to

carry out all necessary tasks within the same application and context in an optimal manner.

POWERED BY FUSION MIDDLEWARE

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

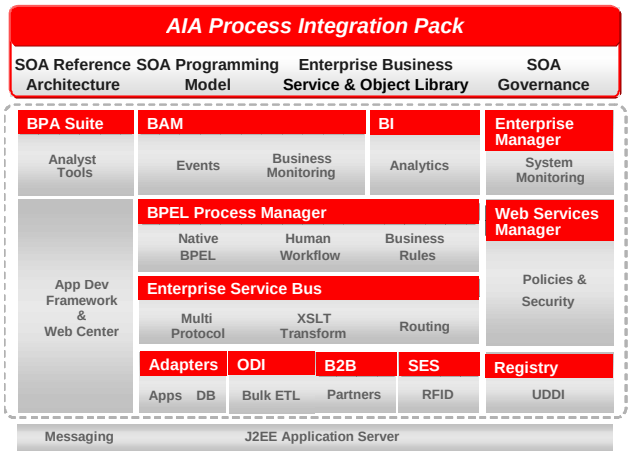

Figure 1: Oracle Fusion Middleware – Best in Class

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 for mediation and routing, Oracle Web Services Manager is used to secure your integrations, and Oracle Service Registry provides underlying UDDI implementation for business service repository. 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 pre-defined Enterprise Business Services and Objects based on 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.

ORDER CAPTURE USE CASE

Let us look at AIA concepts in some more detail. We will use the Order capture use case described below throughout this paper to illustrate how you can use Foundation Pack to build your own integrations that bring together applications within your Enterprise and outside, in a Service Oriented manner.

The figure below is an Order capture use case that involves propagating Orders created in a Siebel CRM system to finally create the order in Oracle E-business Suite Order Management system. This is a part of a larger Order to Cash implementation.

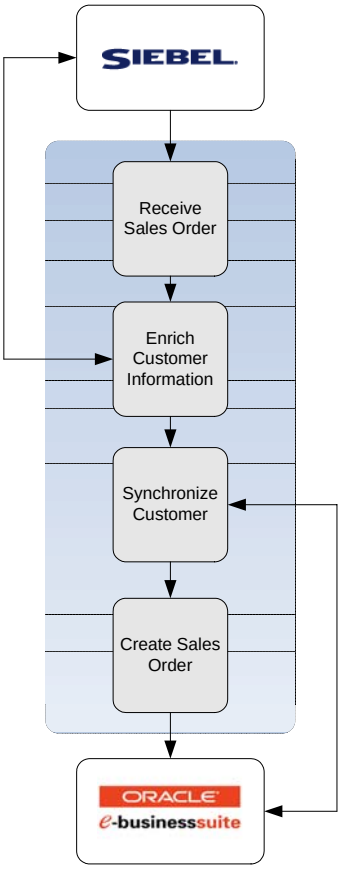

Figure 2: Order capture use case

The following steps describe the above integration steps in detail.

1. On submitting an Order in Siebel CRM, the Order details reach an Order processing flow which itself is implemented as a SOA artifact. This process flow orchestrates multiple service calls to achieve the desired integration.
2. This process flow aims to first synchronize the customer in the Order Management application creating the Order in that application. The current Sales Order information obtained from Siebel CRM contains only the reference to the Customer details. Hence there is a requirement to enrich the Customer data using the customer reference IDs present in the Sales Order message.

3. To achieve this, the process flow makes a call back to Siebel CRM and obtains the Customer information.
4. Now that the complete customer information is available, the process continues further to synchronize the customer information with E-Business Suite. This ensures that the customer context is established for the Sales Order that is to be created subsequently.
5. The Sales Order is now created in E-Business Suite for the customer that was just synchronized.

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.

The AIA is designed and developed on these first principles of SOA. Let's take our 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

The diagram below shows a technical representation of the Order capture implemented using the Application Integration Architecture Foundation Pack:

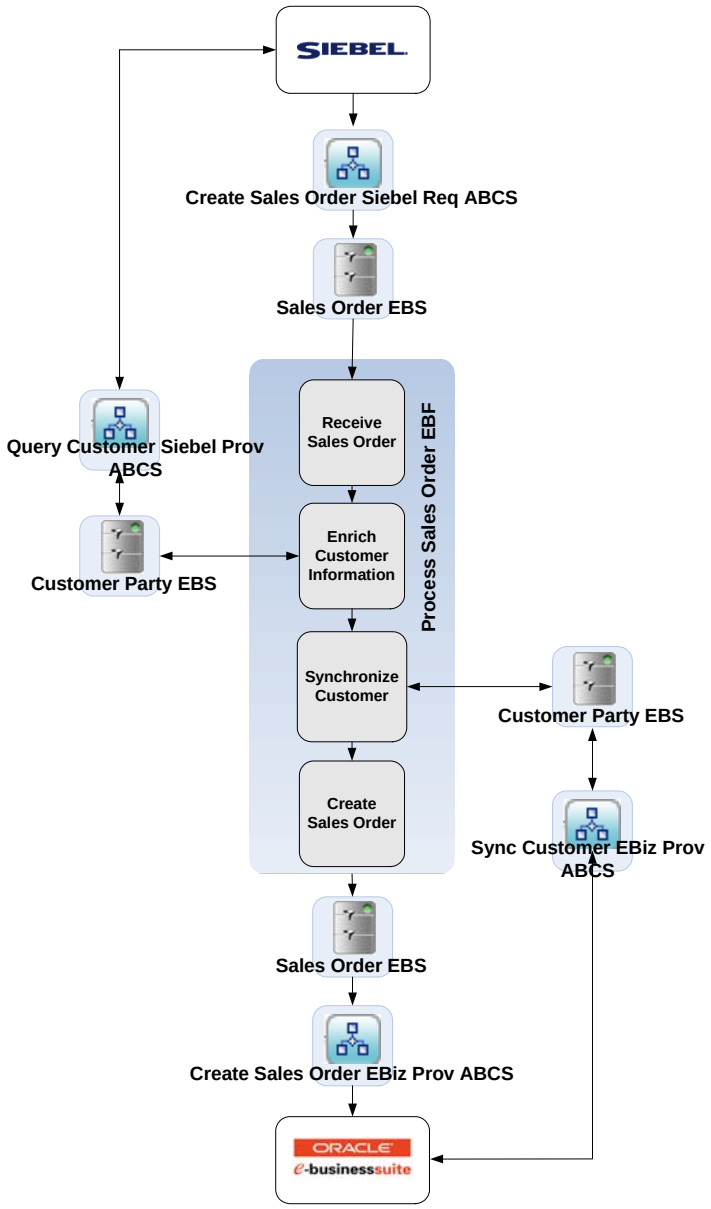

Figure 3: AIA Reference architecture for the use case

The following is the brief description of the AIA artifacts shown above.

Terminology	Description
EBS Note: The same EBS is shown twice in the figure for the sake of clarity	**Enterprise Business Service** – Application Independent, standards based, business level service interface. These services operate on Application agnostic Enterprise Business Messages (EBM) which are turn based on Enterprise Business Objects (EBO)

ABCS	**Application Business Connector Service** – Application specific business connecter service that bridges the application to the application agnostic layer
EBF	**Enterprise Business Flow** – Application agnostic flow that orchestrates several enterprise business services to achieve a larger business functionality

Although the use case above deals with Siebel and E-Business Suite applications, AIA is applicable to any participating application. For example in the same use case above, the CRM system could be PeopleSoft CRM instead of Siebel and the Order Management system could be JD Edwards instead of E-Business Suite. The AIA principles of constructing the integration would still remain the same.

We would now look at each of the AIA artifacts in detail.

Standards: Enterprise Business Objects

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

Gartner

One of the most common challenges in application interoperability is inconsistent business semantics among the different applications. Let us use the Sales Order business document from our use case. The definition of the Sales Order object in Siebel CRM and Oracle E-Business Suite can be different. 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.

The Foundation Pack provides a pre-built Enterprise Object Library that defines the best-in-class representation of business entities such as Sales Order, Purchase Order, Customer Party, 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 use case, Siebel CRM 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

An Enterprise Business Object is a logical representation of the business entity. An Enterprise Business Message on the other hand is the optimal data payload defined

based on the corresponding Enterprise Business Object and is passed among applications. 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 Create Order 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 routing, auditing, tracking and exception management.

Abstraction: Enterprise Business Services

The abstraction layer in Application Integration Architecture is provided by Enterprise Business Services. 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. The invocation to Enterprise Business Services is routed to an Enterprise Business Flow (EBF), which implements the best-practice business process, or to an Application Business Connector Service, which transforms the Enterprise Business Message (EBM) to the native Application Business Message (ABM). The architecture diagram illustrates the core artifacts and the message flow.

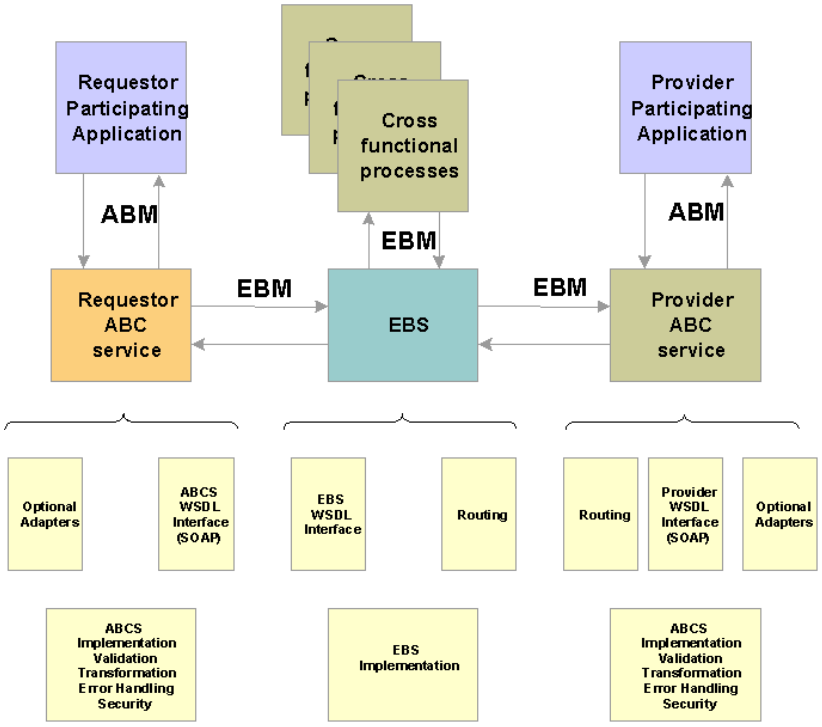

Figure 4: Application Integration Architecture Solution Artifacts

- 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 capture 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 CustomerPartyEBS service and the QueryCustomerPartyList operation in that service. Taking this approach ensures that if there is more than one 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:

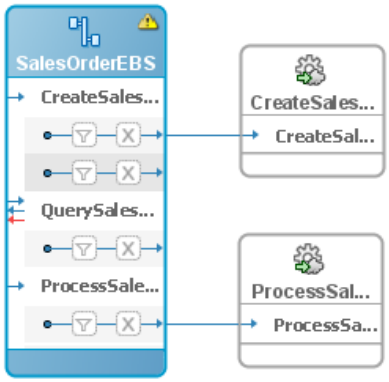

Figure 5: ESB implementation of the Customer Enterprise Business Service

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
SalesOrderEBS (SalesOrderEBO)	CreateSalesOrder	CreateSalesOrderEBM
	ProcessSalesOrder	ProcessSalesOrderEBM

CustomerPartyEBS (CustomerPartyEBO)	QueryCustomerPartyList	QueryCustomerPartyListEBM, QueryCustomerPartyListEBM
	SyncCustomerPartyList	SyncCustomerPartyListEBM, SyncCustomerPartyListEBM

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 or switch their applications 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 applications without a major rip or replace approach.

In AIA, the Application Business Connector Services (ABCS) bridge participating applications (such as Oracle E-Business Suite, PeopleSoft applications etc or any other application) to the Enterprise Business Services. Each ABCS is an implementation specific to a participating application.

As you can see in the above use case, both E-Business Suite and Siebel interface with the Order flow not directly but via Application Business Connector Services.

An ABCS in its simplest form is a service that performs transformation from the native application object to the canonical format and vice-versa. In addition, ABCS is also responsible for performing, any validations, data enrichment etc that may be required for the integration. Note that an ABCS does not aim to replace or modify native Application implementations. They only augment application capabilities and orchestrate native services if necessary so that these applications can successfully participate in the integration.

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

Application Business Connector Service	**Role**	**Participating Application**
CreateSalesOrderSiebelReqABCSImpl	Requester	Siebel CRM
QueryCustomerSiebelProvABCSImpl	Provider	Siebel CRM
SyncCustomerEbizProvABCSImpl	Provider	Oracle E-Business Suite
CreateSalesOrderEbizProvABCSImpl	Provider	Oracle E-Business Suite

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

From the naming, the following characteristics of ABCS can be observed. Like mentioned earlier, an ABCS is specific to an application. An ABCS can either have the requester or provider role in an integration scenario. For example, in our scenario, the Requestor ABCS CreateSalesOrderSiebelReqABCSImpl , transforms a request by Siebel CRM into the AIA integration system to process a new order. On the contrary the Provider ABCS CreateSalesOrderEbizProvABCSImpl is responsible for pushing the Order from AIA to E-Business Suite.

Let us consider CreateSalesOrderEbizProvABCSImpl and focus on the minimum set of tasks that have to be completed to implement this ABCS. This ABCS is built using BPEL technology. According to the use case diagram, SalesOrderEBS invokes this ABC Service with the CreateSalesOrderEBM. The ABCS however has to finally invoke the E-Business Suite application. Hence a transformation activity is required in the BPEL process to transform this EBM to an E-Business Suite specific message (ABM)

Oracle E-Business Suite provides a PL/SQL API to create sales orders. An ESB is used as a transport adaptor that exposes this PL/SQL API as a service. So the ABCS BPEL process first transforms the EBM to the ABM (as required by the adaptor) and then invokes the end-point exposed through the adaptor. We could also have ABCS that invoke concurrent programs that perform the desired functionality.

In case there is a response provided by E-Business Suite (there is no response scenario in our use case), the response would have been returned back to the ABCS via the adaptor and this ABM will be transformed back as a CreateSalesOrderResponseEBM before sending it back to the EBS layer.

Note that there can be more than one invocation of an application from the same ABCS (chatty conversation). For example, this ABCS may want to ensure that the item for which the Order is being created is available in E-Business Suite before the Order itself is submitted. In this case, the desired PL/SQL.API would be invoked on the same lines as described above before invoking the order creation.

Enterprise Business Flows

An Enterprise Business flow is an application agnostic process flow that orchestrates several discrete tasks to achieve the desired business function.

In our use case, we have the ProcessSalesOrderEBF that has activities to perform an order creation. Notice that the EBF deals only with AIA canonicals (SalesOrderEBM and CustomerPartyEBM in this case). The EBF thus interacts only with EBS and hence there is a clear separation of the process definition from the actual implementations of individual functions in specific applications. The

presence of EBS ensures that new applications can be plugged into the integration with no impact on the process flow.

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 Oracle Application Integration Architecture Foundation Pack. Every single Application Integration Architecture component can be extended. Let's use our order capture use case to illustrate how extensibility works.

Extending an Enterprise Business Object

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

The SalesOrder Enterprise Business Object that is shipped by AIA has Supplier Name and Supplier Price attributes defined at the line level. However, let us imagine that Oracle E-Business Suite requires that these attributes be passed at the header level. To be able to integrate, 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.

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

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="CustomSalesOrdreType"/>
```

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 are defined based one the type of SalesOrderEBO. In our case, it is the CreateSalesOrderEBM. 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 integration. Assume that the Organization added another Order Management System, which is a PeopleSoft Application, based on a recent acquisition. The following section illustrates how IT can bring another service provider into the mix with minimal impact to their existing order capture process.

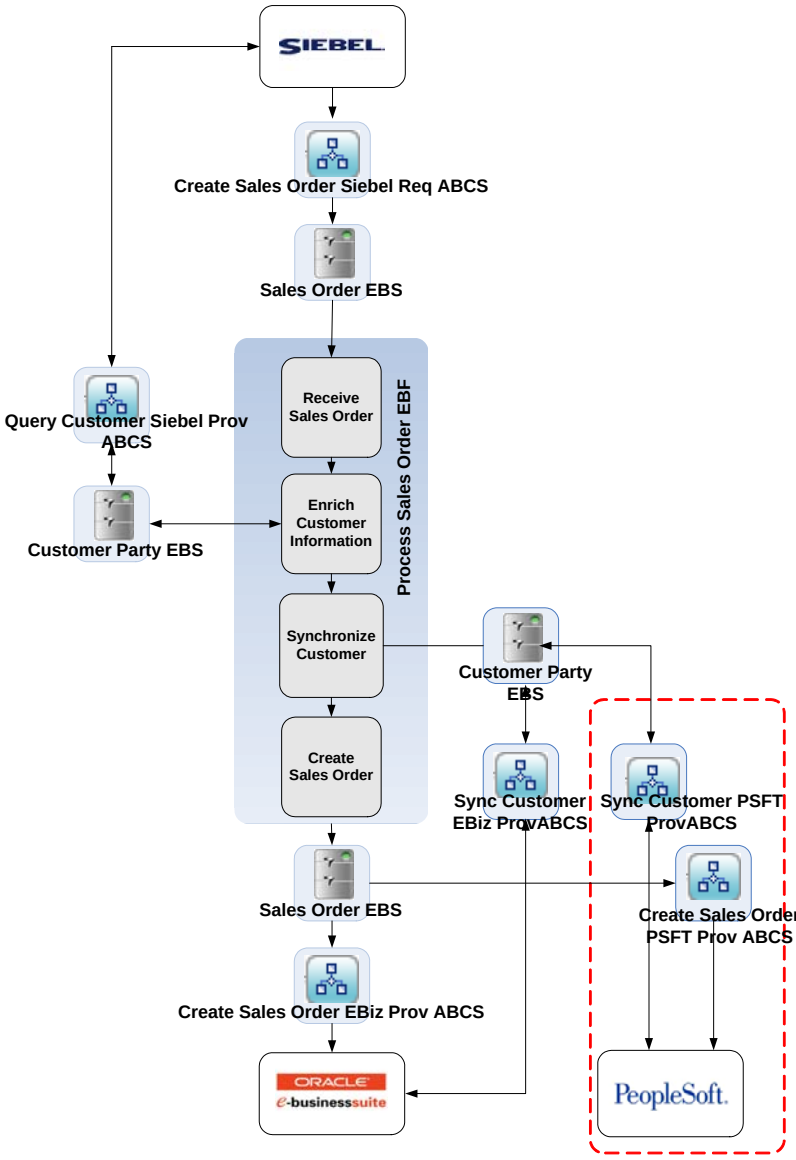

Figure 6: Bringing a new application into your existing integration flow

When adding a new application to the integration, that application specific business connector is the only additional implementation required. The rest of the integration stays intact.

So in the above case, **SyncCustomerPSFTProvABCS** and **CreateSalesOrderPSFTProvABCS** are to be implemented to make sure that the PeopleSoft application takes part in the integration

These ABCS would be built on the same principles as described earlier for E-Business Suite. Of course the ABCS would be calling PeopleSoft service operations on the Integration broker and depending on the capabilities of PeopleSoft chatty conversation may or may not be required.

Besides adding a new application or replacing an existing application for certain process activity, the Enterprise Business Flow itself can also be extended with extensibility points to invoke customer 'add-in' code. In the Order Capture use case, the author of ProcessSalesOrderEBF may add an extensibility point for executing a customized logic before creating the Order for that person. The author of ProcessSalesOrderEBF defines the interface of this extension operation and provides a dummy implementation. After deployment, the implementer has the option of replacing the dummy extension operation with customer 'add-in' operation to select the supplier based on certain special business requirements. The extensibility points in an Enterprise Business Flow are preserved during the upgrade time.

By extending EBF, the customer specific business process can be adapted by the integration solution after the deployment. The extensibility points are upgrade safe.

Extending an Enterprise Business Service Routing Rules

The next step is to wire the ABCS that were created to the existing SalesOrderEBS and Customer PartyEBS. To achieve this, we need to add new routing rules to these EBS. 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 Sales Orders to PeopleSoft if the Order amount is over $500. If it is less than that it will reach the Oracle E-Business Suite. These criteria can be built using JDeveloper's Expression builders or even rules engines.

PeopleSoft */CreateSalesOrderEBM/DataArea/CreateOrder/Base/TotalAmount >=500*

Oracle E-Business Suite:
/CreateSalesOrderEBM/DataArea/CreateOrder/Base/TotalAmount <500

Similar approach can be followed for the Customer Party EBS as well.

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

Extending an Application Business Connector Service

An Application Business Connector Service, regardless of whether it is requestor or provider, has the ability to invoke customer code with extensibility points. The implementation of the extending an Application Business Connector Service is the same as the one for extending an Enterprise Business Flow. Thus, the extension of ABCS is straightforward and upgradeable as described in the section of 'Extending a Business Process'.

There are four possible extensibility points for each ABCS, injecting customer code for ABM/EBM validation, enrichment, and transformation based on certain special business requirements.

The following is the suggested BPEL flow for extending ABCS.

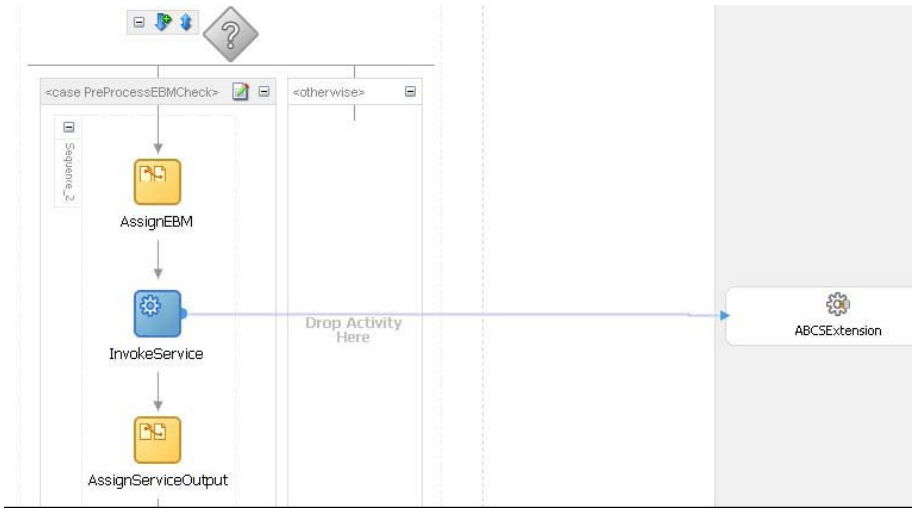

Figure 7: ABCS extensibility point BPEL flow – Switch, Assign, Invoke, Assign

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 (BSR) 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?
- What are the triggering event and data payload passed among applications through 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 definition and relationship of Enterprise Business Objects and the Enterprise Business Services not only from technical aspect, but also from functional aspect. 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 artifacts, including Enterprise Business Services, Enterprise Business Objects, Enterprise Business Flow, Requester and Provider Application Business Connector Services, and Participate Applications in that flow. This provides an enterprise architect visibility into how a particular integration has been designed, deployed and executed. From Business Process Analysis (BPA) Suite, a business process can be drilled down to the relevant BSR page for browsing the detail attributes of the relevant artifacts. This allows the business people to easily switch high-level functional process design view to the detailed object and service technical view. Furthermore, JDeveloper, Oracle Java IDE, is also linked to BSR for developers to publish and share the objects and services created from JDeveloper at the design time.

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 complicate the physical infrastructure required for development and 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. Thus, it can be used for both unit tests at the development time and for system regression test at the QA, deployment or upgrade time.

Let's take an example of how we can use the composite application validation system to test the Order Capture service without actually having to call the E-Business Suite.

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.

For this we create two sets of **Simulators** to simulate the behavior of the Order and Customer service providers of E-Business Suite. In AIA Composite Application Validation System UI, 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.

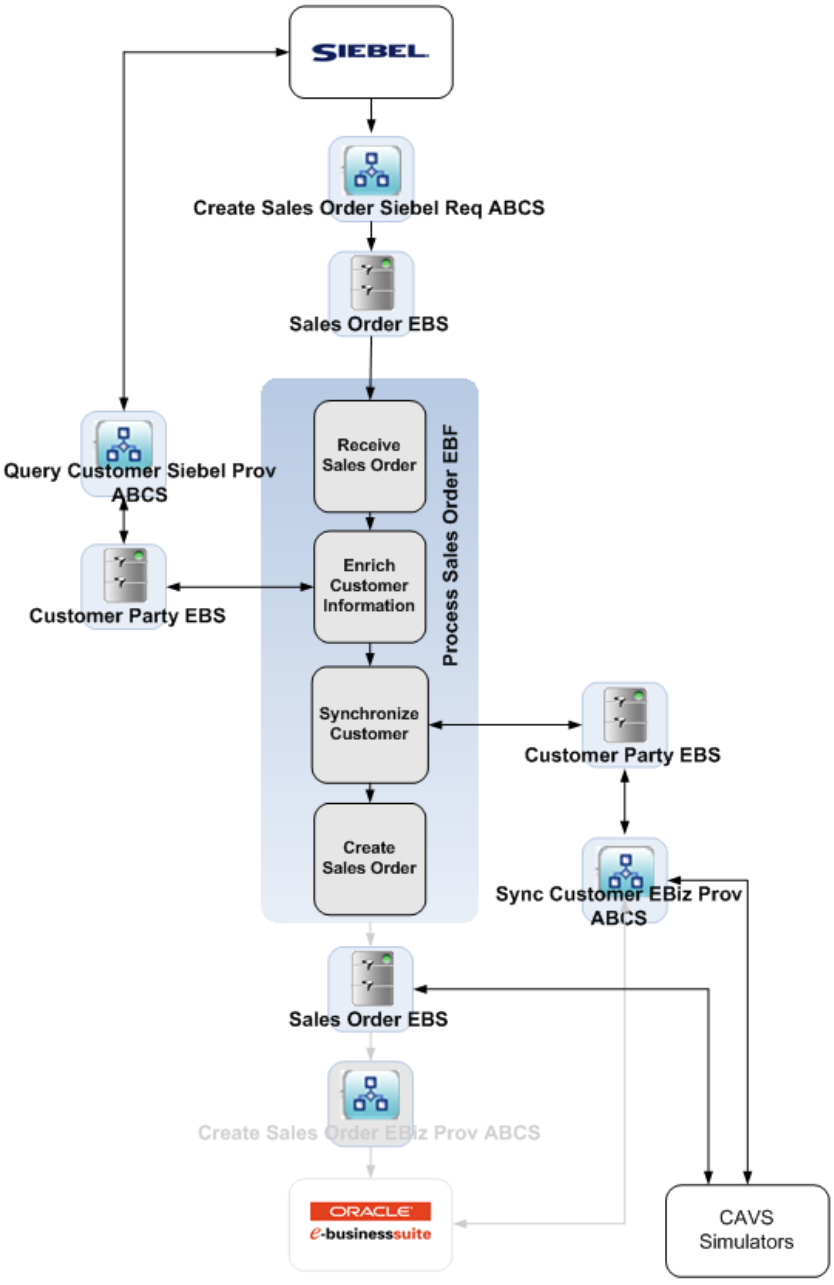

Figure 8: Simulation Scenario in Composite Application Validation System

In the case of Customer Party EBS, we see that message flows through the ABCS. However instead of reaching the Oracle E-Business Suite, a CAVS simulator is

reached. Simulators have a predominantly static behavior with no functional implementation. Hence multiple cases have to be coded for multiple data so that the simulator responds appropriately.

All of the Application Integration Architecture services, including requester ABCS, EBS, Provider ABCS and EBF are designed to route the messages to a simulator or to the actual service based on a configuration parameter. In the Sales Order creation of the above scenario we see that the ABCS itself is bypassed to reach the simulator.

Apart from simulating a web service, CAVS can also mimic a web service client to initiate a service request for testing the components in the integration flow. We can define **Test Initiators** in CAVS for this purpose with the preset request message and expected response message. For example we can have CAVS replace Siebel with the rest of the flow intact (with or without simulator). During the execution of the test initiator, CAVS sends service requests to the desired service. The rest of the integration flow continues. If there is a response expected, the CAVS test initiator waits to receive the response which can be validated against an expected response message to conclude the test to be a success or failure.

SOA Security

Due to the nature of integration, SOA security and policy management need to take consideration of disparate security models and standardized exchange of authentication and authorization information across applications to achieve end-to-end security, instead of point-to-point security. By leveraging Oracle Web Service Management (OWSM), Application Integration Architecture provides the option of decoupling security logic from service development by configuring Webservice Security declaratively using OWSM during deployment. According to security policies set by developers and implementers, Application Integration Architecture is capable of identifying a user through authentication and validating digital signature to avoid message tampering. It supports both transport-level security through SSL full data encryption and message-level security through full or partial business message encryption.

The Foundation Pack provides a security mechanism to integrated participating applications with different security representations in a standard way by propagating standard security context end to end. If the requester application and/or provider application do not support the standard security context, the requester ABCS transforms application security context to standard security context while the provider ABCS performs the reverse transformation. The standard security context is designed compliant to industry standard eXtensible Access Control Markup Language (XACML)

Error Resolution and Logging

A centralized error handling approach across applications, technologies, and integration patterns is key to ensure SOA Quality.

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 proper people based on the error type and the application system which generate the error. . The error message is also published to a log file with view from management GUI, and to a queue, which enables a customer to kick off their own error management system if they have one.

Diagnostics

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

To ensure the integrity of Application Integration Architecture based integration, the Foundation Pack provides a rich set of diagnostic tests. These tests provide an efficient way to troubleshoot a problem in your integration ecosystem. The diagnostics frame is also extendible for running customer-defined diagnostics. These diagnostic 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.

SUMMARY

The Oracle Application Integration Architecture Foundation Pack delivers everything organizations need to implement state-of-the-art integrations to better support their continuously changing 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.

Application Integration Architecture Foundation Pack for Oracle Applications: E-Business Suite, PeopleSoft, Siebel and JD Edwards White Paper
February 2009
Author: Qiming Huang, Arvind Srinivasamoorthy
Contributing Authors: Rimita Bewtra, Angie Wong

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