Evaluating an enterprise for suitability of developing composite business services

Prabhakar Mynampati

February 02, 2010

This article describes how an enterprise can be evaluated to adopt architectures that support Composite Business Services (CBS) and composite applications for solving enterprise business problems. We will focus on the four dimensions on which an enterprise can be evaluated, which includes: existing support for business architecture, application architecture, integration architecture and technology architecture.

Introduction

In this article we discuss the steps for an enterprise to plan and develop a CBS enablement strategy for transitioning from conventional enterprise architectures to reference architectures that support CBS. We discuss the methods for analyzing and evaluating an enterprise architecture alignment to different dimensions of business, application, integration and technology and their related key parameters. This will help us to understand an enterprise for its readiness to build solutions with Composite Business Services (CBS) and to bring out the existing gaps and fulfill the requirements in each dimension in which it is lagging.

Most organizations have incrementally automated their business process requirements by breaking them up as application use cases and then implementing the functionality as IT applications on a need-basis within budget constraints. Majority of these applications typically grow haphazardly with in the enterprise and at some point to meet new business process requirements these applications require integration with other applications. These applications may be internal as well as partner applications. This has created an increased demand for integration products and technologies. So many vendors are focusing in this space and making effort to be market-leaders in Enterprise Application Integration (EAI) domain. On the other side, different enterprise architectural blueprints like Zachman, TOGAF, TeA, and IAF (please refer to Resources section for details) are also looking to bridge the gap between business demands and implemented IT solutions. Many of these enterprise architectural methods look at implementing business process requirements through integration of these applications. This leads to more focus on integration efforts and fails to provide a clear picture on business processes at an enterprise level. The integration becomes challenging when these applications are themselves undergoing transformation to meet the new requirements from the business. There are additional demands on projects to deliver solutions with less developmental costs and in faster delivery time. Business-
driven development with SOA provides a viable solution to this problem in the form of composite application or composite business services. A Composite Business Service is a collection of Business Services that work together, along with a client's existing applications, to provide a specific business solution. CBS enable an asset model based on composition of loosely-coupled, distributed assets to deliver flexible reusable solutions. CBS may include legacy applications, packaged applications, and network delivered services. The architecture, design and development methodology of Composite Business Services solution helps us build reusable business services that are at a higher functional level and can deliver on the promise of the features of business driven development without any coding.

For an enterprise that has already embraced SOA, it is very quick and easy to move to composite application development by adopting proven industry content models. Industry content models provide service definitions, proven data models and common services based on industry standards and best practices. The only extra effort required is to re-align business architecture using these reusable models, re-evaluate business services for the right level of decomposition and granularity, and to enhance or modify the different functional features based on role or channel by which it is invoked.

If an enterprise has yet to adopt business driven development and SOA, and it wants to develop composite applications, it needs to study and evaluate the organization as-is enterprise architectures in practice to directly migrate to CBS reference architectures. The application and data architectures along with their integration methods alone are not sufficient in evaluation of enterprise SOA maturity as usually followed with Service Integration Maturity Models (SIMM). Apart from this, the enterprise support for business functionalities and technology architectures also need to be considered for evaluation.

**Enterprise Architecture evaluation methods**

There are proven qualitative and quantitative methods for evaluating enterprise architectures exist in practice. Qualitative methods try to help how well the enterprise architecture will address the requested requirements by examining the architectural decisions made during the design cycle. The results of this kind of evaluation derive qualitative conclusions on appraised goals. The Quantitative methods are more retrospective approaches and are based on quantitative measurements that are performed during the implementation phase. We describe them here:

**Qualitative methods**

The architecture of a solution is qualitatively evaluated by examining the system with the help of questionnaire and check-list based techniques. These methods are more suitable at the early stage of software development cycle (SDLC) before the Prototype models are being built. The qualitative assessment method of architecture can also be named as predictive evaluation methods. They try to anticipate how well the architecture will address requested requirements by examining the architectural design (decisions) made during early phase of SDLC. The result of this kind of evaluation provides qualitative conclusions on the appraised goals. Similar methods too can be applied on the existing as-is enterprise software architectures with the examination of questionnaire and check list based approaches without any quantitative measurements.
**Questionnaire based method:** If the goals of the software system are easy to be identified and characterized, it is possible to define a list of questions that can be applied to the overall architecture of the software system. These questions constitute the questionnaire to evaluate the architecture and they can deal with different aspects of architecture definition.

**Checklist-based methods:** This method is similar to the questionnaire based but, generally, it focuses on specific qualities to be addressed by the architecture. The checklist based method requires a more mature evaluation practice with respect to the previous one.

**Quantitative methods**

The architecture of a solution is quantitatively evaluated by conducting some experiments on the existing system. These methods are more of retrospective approaches and they are based on quantitative measurements that are performed during implementation phase. The prototype models are built at the early stage of SDLC and quantitative measurements are made on these models and there by from these results the architecture is quantitatively assessed.

**Metric-based methods:** This method is quantitative analysis based on measurement of architecture components. The aim of this measurement is to find the places where there are problems in the overall architecture in order to introduce changes for improving the design.

**Proof-of-Concept (PoC) based methods:** This is when prototypes used for experiments and simulations are the artifacts that result from the development process. In this method, we actually test an implementation by considering a complex application use case which represents a model of the architecture. These prototype results are used to answer some critical architecture questions before the design and development happens in a large number of use-cases.

We can follow either the qualitative approach or a combined qualitative and quantitative approach for evaluation of enterprise architecture and its readiness for developing composite applications based on time availability and the organization support for evaluation. A clearly defined combined evaluation approach is shown in figure 1.

**Figure 1. Evaluation process approach**
Disciplines for CBS

An evaluation process is designed to assess enterprise architectures for purposes of alignment with IBM’s CBS foundation reference architectures. There are four dimensions to this, which are detailed below:

Business architecture

This discipline addresses the concerns of users, planners, and business managers, and focuses on the functional aspects of the system from the perspective of the users. It is primarily concerned with business performance, functionality, and usability. It has the following sub views (please see the link in the Resources section for a link to the Open Group, from which the following is taken):

- The **People view** focuses on the human resource aspects of the system. It examines the human actors involved in the system
- The **Business Process view** deals with the user processes involved in the system
- The **Business Function view** deals with the functions required to support the processes.
- The **Business Information view** deals with the information required to flow in support of the processes.
- The **Usability view** considers the usability aspects of the system and its environment.
- The **Business Performance view** considers the performance aspects of the system and its environment

Application and Data architecture:

This discipline describes architectures covering both data and application system domains. It includes application software inventories, diagrams and interfaces between applications (this includes events, messages, and data flows). The data architecture includes conceptual, logical and physical data models and its metadata models.

Integration architecture:

The integration architecture describes various aspects of integration in an enterprise which includes integrating people, systems and databases internally and externally. There are different aspects of integration being looked for developing flexible and efficient composite business services. These integration sub views include:

- The **Access/presentation integration view** deals with various ways of accessing the system functions and the support for various types of clients (portal, mobile, intranet, and devices like telephony, e-mail, PDAs etc). It deals with these through web service calls, web service remote portlets, and custom built interaction API calls.
- The **Application integration view** deals with integrating applications within the organization, or with business partners through enterprise applications. This allows applications to connect to one another so that they share and use information for better use at the enterprise level.
• The **Information (data) integration view** deals with diverse forms of business information that can be integrated across the enterprise. Integration enables coherent search, access, replication, transformation, and analysis over a unified view of information assets to meet business needs.

• The **process integration view** deals with changes in business, how it operates through modeling, automation, and monitoring of processes across people and heterogeneous systems, both inside and outside of the enterprise.

**Technology architecture**

The technology architecture is essentially the infrastructure that includes hardware and software components. It includes enterprise servers, data servers, firewalls, application infrastructure, security, monitoring and middleware. This also describes the programming languages and operating systems used in the enterprise. This discipline also evaluates how the developed software components have leveraged open technical standards.

**How to assess Enterprise Architecture readiness for CBS**

The first step in assessing enterprise architecture is to first complete a Request for Information (RFI) that addresses in it the four disciplines mentioned above. This should be sent to the client. After obtaining the response from the organization, a checklist-based template is prepared; responses are verified against this template. The template final results qualitatively assess their existing architecture in relevance to the CBS reference architecture and development of CBS services. If an organization is qualitatively qualified for all four disciplines being evaluated, then as a second step for quantitative evaluation, the organization is requested to prepare a description for developing a prototype model based on various scenarios. The description on how the prototype model is to be designed, developed based on scenario and metrics to be evaluated are presented in the section, "Scenario based PoC evaluation". Both qualitative and quantitative approaches with respect to CBS alignment are shown in figure 2.

**Figure 2. Qualitative and quantitative iteration**
Business architecture alignment

A qualitative evaluation can start from the following questionnaire. An organization is evaluated based on the business domain or functional area on which it is providing solutions. Firstly the infrastructural aspects of an organization are looked at to support their business needs. Some of the important points (please see the article, "Exploring Business Process Management Systems and the impact of BPM on developers" which is noted in the Resources section) to consider are as follows:

- Does the organization have a well-defined Business Process Management System (BPMS) to define, maintain measure, analyze, and continuously improve their business processes?
- Does the enterprise have tools like a business process modeler, executable process modeler, process execution engine, business activity monitors, process administration portal and others that support full lifecycle management of BPMS?
- Does the organization have a BPM Center of Excellence (BPM-COE) center set up for practicing such frameworks, tools and methodologies in order to efficiently transform the business requirements into IT system?
- Does the organization have process governance which helps to ensure corporate directions are realized at an operational level?

The business requirements section in an RFI needs to include the predefined business sub-functionalities in the business domain on which the enterprise is engaged. We can consider a customer banking self-service portal as an example. There will be sub-functionalities which include account opening, account viewing, service requests for check book or ATM duplicate PIN, bill payments, funds transfer, and credit card services. The organization needs to present their business solutions in and around these sub-functionalities. From the RFI response obtained from the enterprise, the following points are considered for the organization business alignment.

- How many business sub-functionalities are currently supported by the organization?
- How many business sub-functionalities need to be modified based on predefined functionalities?
- How many business sub-functionalities need to be developed from green field?
- How many business sub-functionalities are there which are not currently supported but have a clear roadmap for supporting those services within in a stipulated timeframe?

The business architecture evaluation section also contains a questionnaire on realization of business sub-functionalities with the business process models and business services. The following points are to be considered in evaluating their business services realization.

- Does the organization adopt any industry specific business process models?
- Do they use their own custom built models? If yes, how flexible are these custom built models to absorb changes in business requirements?
- Do their business services support industry specific data models like ACCORD, HiPAA, and SWIFT to exchange data among other services?
- Does the organization follow any standard method or technique for identification of business services like RUP for SOA, SOMA, etc?
• Do the realized business services provide flexible and adaptable behavior based on business policy and user context?
• Are the business services provisioned through multiple communication channels?
• Is the business service realized from disparate IT systems? If yes, is it from a Silo format, is it integrated, or is it through componentized process integration?

All the responses for the above questionnaire are studied and analyzed towards alignment with development of CBS services and ultimately a qualitative assessment chart is prepared against this section.

Application and Data architecture alignment

In this section we will go through how an organization's applications and data architectures are evaluated for alignment with CBS reference architectures. The overall application architectural maturity can be assessed from points like closeness to CBS reference architecture, IBM's e-business patterns, enterprise application architectural patterns, and development with model driven architectures tools. The architectural principles like loose coupling among tiers, MVC patterns followed, layered concepts practiced and scaling capabilities of the application are strictly evaluated in this section. Key architectural layers from their application architectures and the key evaluation points (please see the link to the article, "Evaluating Service Oriented Architecture" in the Resources section) include

• Channels and Presentation layer
• Business process and choreography layer
• Services or exposed functions
• Business rules
• Service registry layer
• Data and data access layer

The details on each topic is covered in the below sections.

Channels and Presentation layer:

The architecture of the application or system is evaluated as how it is related to the channel and presentation layers. The composite application needs to service multiple clients from a shared, common hosting environment. The channels and presentation layer is assessed from the following points.

• The presentation layer should support open standards frameworks like STRUTS, JSF and Dot Net UI and must be easy to extend or modify to build custom presentation tier frameworks.
• The presentation layer should also be flexible enough to add new channels like PDA clients, forms and email etc.
• If the organization is using any kind of proprietary frameworks, their association with open source frameworks should be evaluated.
• Check for headless invocation of system functions through web service interfaces.
• Check whether the presentation layer is loosely coupled with the present system / application or not.
• What are all the different types of physical devices / channels supported with system? And how flexible in adding a new physical device to the existing system.

Business process and choreography layer:

The architecture of the application is evaluated with respect to business process and choreography function. One should look at the organization and see whether they adopt any business process layer and runtime engine for orchestration of their business services / application functions. The following points evaluate this architectural layer.

• If any proprietary process flow or workflow layer is used, check whether it is in compliance with BPEL standards by porting to external BPEL design tools and runtime engines. Identify the steps and procedures which need to be followed to run such proprietary workflows on open standards runtime engines.
• Whether the organization has any business process modeling tool which automatically generates BPEL run time code for deployment?
• Check how BPEL compliant runtime engine is implemented as a full-fledged scalable product with compensation, business and technical exception handling and has business metrics, transaction volumes monitoring features.
• Check whether current process flows have support for invoking human tasks, selectors, business rules and ESB.
• Look at how the BPEL process flow and service interaction are implemented; are they tightly coupled or loosely coupled and can the BPEL process itself can be exposed as a service with open standards?

Services and exposed functions:

Here we will look at how to evaluate the architecture of the system / application as it relates to services or exposed functions as interfaces and API. The maturity level of services is determined from the following points:

• How can the services be accessed? Is it through open technical standards like web services or SCA interfaces?
• How the services are implemented with underlying systems; are they tightly coupled or loosely coupled?
• Do the organization's boundary services follow industry standards like ACCORD, HiPAA for enterprise data sharing and accessing?
• Are the services implemented with the right level of decomposition and granularity?
• Do the services support both synchronous and asynchronous invocation?

• Do the services support both exception handling and fault recovery?
• Do the services support both authentication and authorization?
• Do the services have provision for publishing in registries both at design time and at run time?
• Is service versioning supported both at design time and run time?
• How are the technical services organized and how does the application services or business service interact through these technical services in realizing business transactions?
Business rules:

This section evaluates the architecture of an application as it relates to business rules. How are the business rules implemented? Are they tightly coupled to the system and cannot be externalized? Though some implementations are loosely coupled they still cannot be externalized and code level modifications are required in order to change the rule. Some implementations are loosely coupled and externalized, but use a proprietary rule engine or proprietary programming framework. Some business rules are loosely coupled and externalized as well, their programming model follows standards like JSR94 and rules can be easily changed with business requirements. The following points evaluate the strength of the business rules adopted in the architecture of the solution.

- How the rule engine is built with? Is it with plain Java classes or EJBs? Is it implemented as a full fledged scalable product with online editing and complete life cycle management support?
- Does the existing rule engine allow for a third party rule engine to connect either to add new rules or transfer the existing rules to third party engine?
- Check whether or not this rule component can be exposed as a web service or SCA service either to orchestrate from external BPEL process flows or call from third party clients.

Service registry layer:

A service registry provides registration of services, management of metadata and automation services. This layer is evaluated based on answers from the following questionnaire:

- Is a registry being used? If not, how do various parties using shared services know about the availability and capability of services? How is service information maintained to avoid unnecessary duplication?
- What policies are in place to ensure the proper use of registries?
- How is service metadata defined and managed within and outside of the registry? Are long-term considerations of future possible needs factored into the design?
- For which phases of the SOA application life cycle (inception through decommissioning) is the registry being used?
- How are service access controls and change management policies governed? Are proper controls in place to balance security, modifiability, and compliance with IT and other standards?
- Is the registry being used for the dynamic routing of service calls (e.g., for failover, load balancing, and application partitioning)? If so, is the registry installation a single point of failure? Does it meet performance and failover time requirements?
- Is the registry public or private? Does the registry implementation properly handle the differentiation of internal and external services?

Data and data access layer:

This section explains the architecture of the application as it relates to data and data access. The following points are considered in evaluating this section.
• How robust and flexible is the data model? Does follow mature industry standards? Can new data elements be easily added?
• What is the data access layer implemented with? Is it tightly coupled and does it use proprietary frameworks? Is it loosely coupled and does it follow matured frameworks like open source service data objects?
• Does the organization leverage any object relation mapping tools like toplink, hibernate or iBatis?
• If a data repository is distributed across the enterprise, what kind of mechanism is followed to allow access to the applications?
• To support "Information As a Service" what kinds of tools or products does the organization need to leverage?
• How does the enterprise data architecture help in processing the transformation of transactional data into analytical data with reduced data latency?
• How does the enterprise data architecture help in analytical processing of data to deliver information to business users on demand?

Integration architecture alignment

This section evaluates the application architecture as it relates to the integration of applications, components and services including third party and legacy systems. The following points are considered in evaluating the maturity of the integration layer.

The sample evaluation questions that need to be asked about the integration layer are

• How robust is the integration layer? Is it being implemented as a full-fledged scalable product? Or is it implemented with an open source API or with connectors and adapters on an as-needed basis?
• What are the integration architectural patterns that are supported? Will it use ESB, hub and spoke, or point-to-point?
• What are the functionalities supported by the integration layer like message routing, transformation of data formats, centralized security gateway for all services? Will it support the publish and subscribe message model and message aggregation?
• How tightly or loosely coupled is the integration layer from the rest of the system or application?
• What are the different types of integration specifications/standards/frameworks the organization currently supports? For example, does it support RPC, RMI, SOAP/JMS, SOAP/HTTP?
• Does the integration layer support auxiliary functionalities like exception handling, event management, auditing, logging and support for access control?
• Does the presently followed application architecture provide a provision for introducing this integration layer in between tiers that have mature solutions with integration architectures?

We exclusively capture information on how the legacy application integration is taking place in the organization by getting more information on the following:

• What mechanisms have been adopted for the integration of new and legacy systems? The ones we look for include screen scrapers, web service calls, ESB with adapters for a legacy business services
platform, messaging systems, direct legacy software API calls, technology specific gateways and bridges.

- How has the selected mechanism been compared in terms of complexity and cost of implementation?
- Has the selected mechanism met the performance of the system in terms of expected number of calls, desired response times?
- Have security requirements like access control and data privacy been met in both existing and legacy systems

**Technology architecture alignment**

Let’s take a look at the software infrastructure and how it is intended to support the deployment of core, mission-critical applications. Enterprise servers, application servers, process servers, database servers, security servers, notification servers and their deployment configurations fall into this category. The following topics are covered in this technology architecture evaluation.

- Infrastructure services
- Security architecture
- System management and support services
- Open technical standards
- Operational model or deployment architecture
- Performance
- Other NFRs, Availability and Reliability

- Does the presently followed application architecture provide a provision for introducing this integration layer in between tiers that have mature solutions with integration architectures?

We exclusively capture information on how the legacy application integration is taking place in the organization by getting more information on the following:

- What mechanisms have been adopted for the integration of new and legacy systems? The ones we look for include screen scrapers, web service calls, ESB with adapters for a legacy platform, messaging systems, direct legacy software API calls, technology specific gateways and bridges.
- How has the selected mechanism been compared in terms of complexity and cost of implementation?
- Has the selected mechanism met the performance of the system in terms of expected number of calls, desired response times?
- Have security requirements like access control and data privacy been met in both existing and legacy systems

**Infrastructure services:**

We look at various infrastructural components (please see the link to the article, "SOA Practitioners guide part 2 SOA reference architecture" in the Resources section) required for reuse for application development or reuse at the enterprise level. If these services are reusable at all levels of the enterprise, it shows that the organization is consistent and has a uniform approach
for architecting solutions with proven services. It is easy to decide whether the organization is able to meet the service level agreements or not with the historic data available with the solutions that were previously built with such services. The evaluation is made based on the various services that are available in the organization. To determine how to best set up the infrastructure services, we consider the following:

- What are the common components / services that are available in the organization for developing custom applications / packaged applications? These services might include data services, logging services, fault handing services, audit, search, notification, and session management services.
- What are the different types of portal services available for reuse and consistent look and feel? These services include personalization, reporting, localization, and Web traffic monitoring services.
- What are the different types of enterprise infrastructural services available? We would look for things like LDAP, email, and collaboration (chat/IM/whiteboard), and content management.
- What are all the different master data management services available? Customer data integration services and product master fall into this category.

**Security architecture:**

It's important to understand the current security model, user roles, permissions and capabilities of the application. The following points helps in evaluating the maturity of security architecture:

- What are the different IT security services implemented in the organization?
- Identify whether IT security can be implemented in all application tiers.
- How easily can the security architecture be changed and upgraded?
- Find out if the security architecture has been implemented with a protocol firewall, domain firewall and enterprise firewall positioning.
- Does the application support single sign-on (SSO)? Is SSO at both the application and web services levels?
- Does the organization have security policy management frameworks in place?

**System management and support services:**

In this section we'll evaluate the architecture of the application as it relates to application management and support services. Some application architectures do not have support for system management services at all. Some applications are architected and designed well with full life cycle service support /application management such as governance, access, authorization and monitoring.

- Check whether system monitoring and management services are implemented with open standards and APIs like JMX, open SNMP APIs.
- Check whether all these management services or open standards products used are fulfilling the requirements of monitoring business and IT key performance indicators?
- Find out whether monitoring data is helping the management architect in fine tuning the infrastructure, and helping the business analyst in redefining the optimized business processes.
Deployment architecture:

Here we look at various middleware servers that are engaged in supporting the solutions implemented with specified application architectures. Typically, the organization presents a detailed deployment model of the solution.

- Check whether any standard e-business deployment architectural patterns followed in freezing their topology architecture?
- Check at the operational model or topology architecture of the system that shows hardware nodes and versions of software components that would be running in the nodes in a typical production environment. Check whether the model is complete and clear and provides detailed level of information about the zones, hardware, software and connection specifications or details.
- Look at the other aspects like whether the solution is virtualized, and if the solution grid allows you to take advantage of clustering and workload balance.

Performance:

Evaluate the performance of the application by looking at performance metrics results furnished by the organization for low, medium and complex use cases. Get the information as to the scalability of the system in terms of number of users and number of transactions with supported hardware configurations. Most of the organizations are not mature enough to provide performance bench marks at the service level. Insist on service level performance metrics which help predict the end-to-end response times and capacity planning of servers when the composite application is built. Look at the other aspects like

- Does the organization have any software framework components or products which improve the performance of the solution in terms of transactional response time and throughput?
- Does the organization have performance modeling and capacity planning tools? Does the present solution take into consideration a user workload growth plan for the next 2 to 3 years?
- During the Software Development Life Cycle process of the solution stage, we want to see if performance engineering lifecycle methodology / tools have been followed or applied?

Other Non Functional Requirements (availability and reliability):

Check the system availability under critical conditions like the following:

- When the system is hacked with unauthorized and unformatted messages
- When the system is overloaded
- During maintenance periods
- During software version changes

Check for the system reliability under failures and recovery for:

- Transactional process state
- Maintaining the same data after the recovery
The above mentioned questionnaire in each discipline helps to assess enterprise architecture with qualitative attributes such as low, medium and highly aligned with IBM CBS reference architectures.

For better understanding of the concepts a quantitative assessment for alignment of CBS architectures, a sample scenario based PoC evaluation in the application architecture discipline is discussed in the below

**Scenario based PoC evaluation method**

We should evaluate the architectural disciplines mentioned above quantitatively by building scenario-based PoCs. The business architecture is evaluated by generating functional test cases as per the functionalities defined by the enterprise. These test cases are run on the deployed solution and verified with committed functional features. The quantitative evaluation is made based on the number of test cases cleared during functional testing. Similar quantitative evaluations are made for information, integration and technology architectural sections based on an evaluation scenario. For example, we consider a typical scenario from an application architecture discipline on which an organization is evaluated during the architectural transformation phase to CBS reference architectures.

**Scenario:**
Can existing application services and components be directly consumed for developing a composite application?

The quantitative assessment is made based on the following pre-defined set of points. Each assertion point is defined in such a way that it has a discrete level of distinction from its ideal alignment. Look at the data points given in descending order and that are deviating from the CBS service alignment and hence the evaluation score against each point is reduced gradually.

1. The organization has services / components that are exposed directly as web services and are being consumed from the BPEL process. These services are published in UDDI or some equivalent registry (score : 100%)
2. The organization has services / components that are exposed directly as web services and are being consumed from the BPEL process. But these services are not published in UDDI or some equivalent registry (score : 75%)
3. The organization has a services / component that are exposed indirectly as web services through some architectural framework component (gateway service) but able to consume through BPEL process (score: 50%).
4. The organization has services /components that are exposed as web services but not able to call the from external clients due to the fact that a SOAP address binding URL specification is missing due to non-compliance of WSDL.( score : 25%)
5. The organization has a service / component which has been implemented and exposed as an EJB interface. (score : 0%).

As per this scenario, a small PoC is built with WebSphere Integration Developer by importing a web service to its assembly environment and invoked through a constructed BPEL process.
with direct as well as indirect (through UDDI) end-point URL look up for a web service. If the web service of the type specified in condition 4, then this type of WSDL cannot be allowed to import in the WID itself. Based on these PoC executions and observations, the quantitative assessment is made for this scenario. Similar PoC models are built based on scenarios in integration and technology architecture disciplines too and quantitatively their as-is architectures are assessed.

**Conclusion**

In this article, we looked at enterprise architecture through RFI response obtained from an organization. Preliminary qualitative evaluations are made on their business, application and data, integration and technology architecture alignments with respect to CBS solution reference architectures based on the points presented in the previous sections. Since the evaluations are made based on information furnished by the enterprise, it is quantitatively evaluated by conducting a PoC on-site and thereby you are able to determine the state of the enterprise for its readiness for leveraging existing assets as they might relate to composite business services. The final PoC assessment report explains the gaps the organization needs to fulfill and further go ahead with building composite business services. If an organization is not sufficiently aligned with the requirements of CBS solutions, an enablement strategy needs to be prepared and submitted to the organization.

© Copyright IBM Corporation 2010
**Trademarks**
(www.ibm.com/developerworks/ibm/trademarks/)