

The Importance of Service Oriented Architecture

Applies to:

Business Experts

Summary

This document describes the need for Service Oriented Architecture (SOA) in an enterprise.

.

Author(s): Rohit Kumar Shukla

Company: HCL Technologies

Created on: 12th October 2006

Author Bio

Rohit Kumar Shukla is a SAP NetWeaver technical consultant working with HCL Technologies.

.

Table of Contents

Applies to:.....	1
Summary	1
Author Bio.....	1
Introduction to Service-Oriented and Event-Driven Architectures	3
Related Content	4
Disclaimer and Liability Notice.....	5

The Importance of Service-Oriented Architecture

Web services technology has now advanced so that functions within existing application programs and suites - as well as functions within ERP (enterprise resource planning), CRM (customer relationship management), SCM (supply chain management) and other packages - can easily and reliably be published to an intranet or the Internet for remote execution using SOAP, WSDL and UDDI. But what has been missing until now is an automated way to invoke available Web services based on business rules. This technology is now becoming available with business process management languages and tools.

Until now, the term "Service-Oriented Architecture" (SOA) has been synonymous with "web services." Use SOA more precisely: to invoke web services using business process management tools and languages. This is an important distinction. SOA is expected to make a significant contribution to the future of systems development technologies, as indicated in the following paragraphs.

Before SOA, systems development used workflow diagrams or systems flowcharts that were drawn and then printed so that relevant business logic could be coded by hand. These manually coded programs were laboriously tested and eventually deployed for execution. With SOA using business process management tools, this manual coding and testing step is bypassed. Instead, the diagrams are tested for correct logical execution using simulation methods. Once correct, these diagrams are then automatically generated as XML-based business process management languages for immediate execution.

This business process management technology is a major advance in the productivity of systems development; it is as significant as the development of high-level language compilers. It becomes easy to invoke web services anywhere in the world and to execute them based on business rules. When these rules do change, the relevant logic in the diagrams is changed: these diagrams are then automatically regenerated. This promises to totally transform the way we build systems in the future from slow, error-prone manual coding to an automated discipline. It will enable enterprises to implement changed business rules in minutes or hours, rather than in months or years. Enterprises will then be able to change direction rapidly.

Introduction to Service-Oriented and Event-Driven Architectures

Service-Oriented Architecture (SOA) is the term that has emerged to describe executable components - such as web services - that can be invoked by other programs that act as clients or consumers of those services. As well as the execution of web services, these services can also be complete modern - or even legacy - application programs that can be invoked for execution as a black box. A developer does not need to know how the programs work, only the input that they require, the output they provide and how to invoke them for execution.

The services are loosely coupled to the client program. They can be invoked based on decisions made by business rules. This means that developers can swap out one service and replace it with another service that is designed to achieve the same or enhanced result without having to worry about their inner workings. Today, standard parts in a car can be interchanged without having to strip down the whole car and rebuild it. With SOA we have similar flexibility, where existing services can be easily replaced by improved services without having to change the internal logic of monolithic application programs as was necessary in the past. Software categories that provide this SOA flexibility are called business process management or Business Process Integration (BPI) products.

A further term also describes these business process management and BPI execution environments: Event-Driven Architecture (EDA). This is an approach for designing and building applications where business events trigger messages to be sent between independent services that are completely unaware of each other. An event may be the receipt by the enterprise of a sales order transaction from a customer for processing. Or, it may be a change in a data value that requires a purchase order to be placed with a supplier when the available quantity of a product in the warehouse falls below a minimum balance threshold.

Because the services in an EDA environment are independent, they are decoupled - as distinct from the loosely coupled services of the SOA-based approach. An event source sends messages to middleware software, which matches the messages against the subscription criteria or business rules of programs or services that want to be notified of these events. Messages are typically sent using the publish-and-subscribe approach because this enables simultaneous delivery to multiple destinations.

Most business process management products to date have used proprietary methods to define process logic in workflow diagrams. To overcome these product-focused solutions, an open architecture approach has been defined for interoperability. Several XML languages have emerged: Business Process Execution Language for Web Services (BPEL4WS), Business Process Modeling Language (BPML) and the ebXML Business Process Specification Schema (BPSS) definitions, and I will address these in the following months.

Related Content

[ARIS for SAP Netweaver and SAP Enterprise SOA](#)

[Road to ESA – the EPI Model Part I](#)

[Road to ESA – the EPI Model Part II](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.