International Journal of Computer and Communication Technology

Volume 1 | Issue 4 Article 11

October 2010

Enabling Mainframe Assets to Services for SOA

Latha Sadanandam
Education and Research Dept. Infosys Tech Ltd. Bangalore, India, Latha_Sadanandam@infosys.com

Enabling Mainframe Assets to Services for SOA

Latha Sadanandam
Education and Research Dept.
Infosys Tech Ltd.
Bangalore, India
Latha_Sadanandam@infosys.com

Abstract- **Service-oriented architecture (SOA) is a mechanism for achieving interoperability between heterogeneous systems. SOA enables existing legacy systems to expose their functionality as services, without making significant changes to the legacy systems. Migration towards a service-oriented approach (SOA) not only standardizes interaction, but also allows for more flexibility in the existing process.**

Web services technology is an ideal technology choice for implementing a SOA. Web services can be implemented in any programming language. The functionality of Web services range from simple request-reply to full business process. These services can be newly developed applications or just wrapper program for existing business functions to be network-enabled.

The strategy is to form a framework to integrate z/OS assets in distributed environment using SOA approach, to enable optimal business agility and flexibility. Mainframe applications run the business and contain critical business logic that is unique, difficult, and costly to replicate. Enabling existing applications allows reusing critical business assets and leveraging the assets as a service to be invoked in heterogeneous environment.

Keywords: Service Oriented Architecture, Web services, Legacy applications, service broker, service consumer, service provider, CICS, etc.,

I. INTRODUCTION

Integration of ZOS systems with other system in distributed environment in SOA framework is to publish the assets such as application and data as a service [11].

Modernization by Integration using SOA technique can be categorized as

- Modernizing Presentation layer
- Business logic Integration
- Data Integration [15].

In SOA framework, the services can be categorized as service provider and service requester.

The service provider creates a Web service and publishes to the service registry the information necessary to access and interface with the Web service. The service registry publishes the Web service access and interface information to any service requester. The service requester binds to the service provider to invoke one of its Web services, published in broker registry.

Mainframe Assets such as data, batch and online programs/applications has to be used as either as service provider or requester to fit into SOA. The mainframe assets can be categorized as in the below figure. Each asset can be converted as service provider, requester or both.

The asset which is converted to services can be invoked by any distributed clients such as JAVA client, .Net client. The ZOS applications can act as requester to invoke the services in the heterogeneous environment.

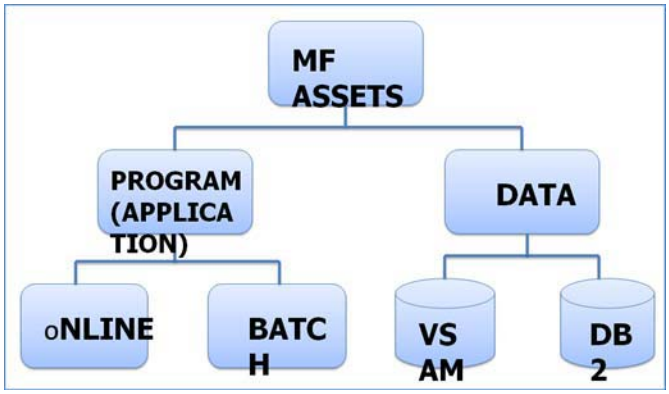

Fig 1. The mainframe assets categorization

I. ACCESSING VSAM DATA AS A SERVICE PROVIDER

To integrate VSAM data through Web Services is to expose the existing z/OS program that currently accesses the VSAM data as a Web service. If such program does not exist, a new program might need to be built. The Web service can then be invoked by multiple clients such as Java application, a .Net application, or any other application that is capable of using Web Services as a

protocol. The structure related to the VSAM record needs to be converted to a Web Services Description Language (WSDL) file The WSDL file can then be published to the Java client or .Net client application to invoke the Web service.

Benefits: Reuse of existing assets, robust data access through transaction servers and no data migration needed, because the data can remain inside VSAM files [7].

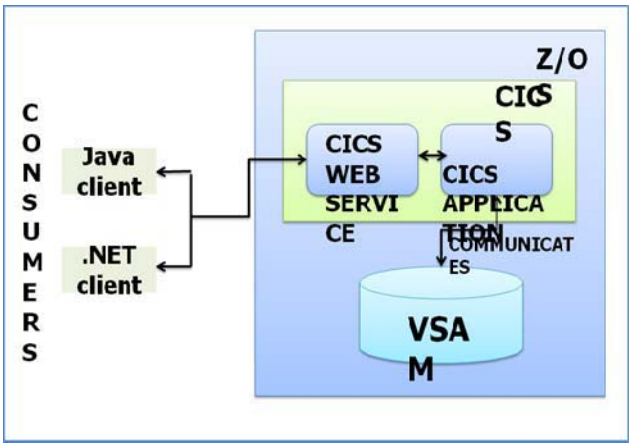

Fig 2. Accessing VSAM data as a Service Provider

II. Accessing DB2 Data as a Service Provider

Data Web Services (DWS) can be used to expose existing SQL statements as Web services[7]. The input for the SQL, that is host variables, becomes parameters of the Web service to be created. The Web service can be deployed on Websphere Application Server for windows or on Z/OS or any other Application Servers. The deployed Web service accesses DB2 using Java Database Connectivity (JDBC), either locally on the same logical partition (LPAR) using a JDBC Type 2 driver or on another LPAR using a JDBC Type 4 driver[5]. It works for both SQL queries and calling stored procedures.

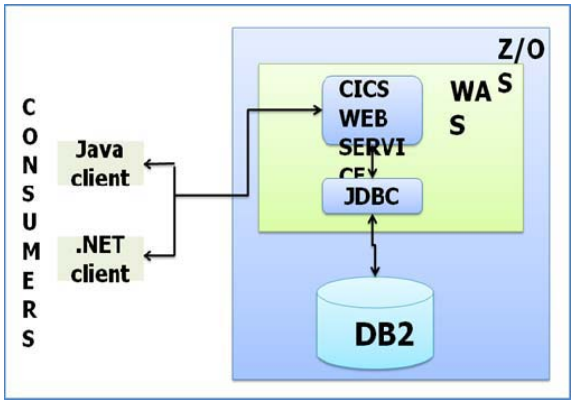

Fig 3. Accessing DB2 data as a Service Provider

The advantages of this solution: it supports static SQL and stored procedures to be converted as web service. Existing complex queries can easily be reused without any change.

III. Accessing Z/Os Online (CICS) Programs as a Service Provider

The Z/OS online programs (CICS/IMS) generally have their presentation and business logic layers segregated. The applications contain several programs with business logic of high demand which can be reused. This framework is to enable the business logic programs accessible for a client programs in a structured way. The performance of the application plays a vital role for the existing applications.

Web service enable selected CICS business logic programs and convert it to Web Services Description Language (WSDL). The WSDL can be published for the requestors. After a CICS Web service is implemented using CICS Web Services support, it is accessible from a Web service consumer using Web Services protocols. The consumer can exchange SOAP messages with CICS over HTTP.

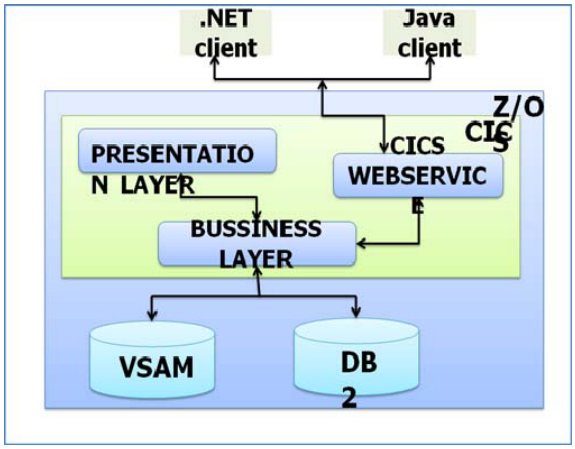

Fig 4. Accessing Z/OS online programs as a Service

Benefits: Existing programs, which have been proven, secure, reliable, and scalable, can be retained and it uses standard protocols for communication.

IV. Z/Os Online System (CICS) as a Service Requestor

The legacy CICS applications strongly benefit from having access to external services. The new functionality that is required for legacy applications resides in other environments or on other platforms [4]. New service-oriented architecture (SOA) technology and features in CICS make it easy now to reuse functionality outside CICS.

The Web Services Description Language (WSDL) file of the existing Web service is used to convert the XML interface structure of the distributed service into a language copybook structure using CICS Web Services Assistant (WSA) utility. WSA create both the request and response structures in the form of a language copybook. The service requestor program and Wsbind file are deployed to z/OS. The deployed service can be invoked from CICS application.

V.Z/OS BATCH SYSTEM AS A SERVICE REQUESTOR

Batch systems in legacy system are schedule weekly, monthly or daily. Many of these jobs perform functions in which they can benefit from accessing certain services directly. Basically, batch applications access data directly and do the process and writes output to file, printer, or databases. Batch applications are carefully designed to fit a certain processing window, outbound service calls need to be carefully evaluated and designed.

External services can be invoked from existing batch applications. This can be achieved using Message Queue (MQ) / Enterprise service Bus (ESB) / Java Messaging Service (JMS) [6].

VI.CASE STUDIES

Case study 1

Background

A bank has been maintaining customer data over the last few decades in VSAM clusters in Mainframe Environment. The business logic was written using legacy language COBOL. After some time the bank was acquired by another leading bank. The leading bank runs their applications predominantly on windows environment. The idea was to replicate the data in VSAM to the database in Windows environment.

The study here would be to discuss on, provided this situation which would be the best of the above mentioned techniques to access data from the acquired bank.

Constraints

The main challenge in approaching is to keep the data synchronous and maintain data consistency. Another constraint is there must be minimal change in business logic and no language change.

Solution suggest

The acquired bank has to replicate the data. Replication of data happens in a time independent manner. Since the replication is asynchronous, data will not be sync with the master data base. Hence data consistency is lost. Integration using SOA is solution; here the VSAM data can be exposed as a web service. The web service can be invoked by multiple clients which supports SOAP protocol.

Benefits

- No replication of Data
- Data consistency can be maintained
- Robust access of data through transactions

Case Study 2:

Background

An organization had a requirement to “migrate” their existing mainframe functionality and its processing capabilities to a distributed system of workstations using client-server architecture.

They believed they could forego a formal requirements elicitation and validation process, and just concentrate on the “requirements delta.” In their thinking, the delta corresponded to a few new features they wanted to add, along with the processing differences stemming from migrating from a batch-oriented system to an interactive one.

The method they adopted to address this requirement was RE-ENGINEERING. This case study discusses on the impacts and effects of re-engineering used as a method of legacy modernization.

Constraints

The re-engineering comprises of two phases – Reverse engineering to get a better understanding of the existing system and second phase is of forward engineering to develop the same in new environment. Time involved in both of these phases is huge. Both the existing system and new system has to work parallel until new system is stable enough to work perfect. This increases the cost of the application. These processes are also subject to lot of risks and failures.

Solution suggest

This incremental approach to eliciting requirements was rendered more challenging by lack of documentation. There was no user’s guide, and the minimal system and software documentation that did exist was quite out of date due to years of software modifications and a legacy of changes to the system.

The solution for approaching this requirement can be Integration using SOA, where in the existing application can be exposed as a Web service, which can be invoked in heterogeneous environment.

Benefits

- Time and cost is moderate
- Reuse Existing applications
- User experience is New, since the legacy screens are converted to user friendly screens
- No change in Business process
- Business Logic resides in secured Mainframe environment.

Factors considered for analysis of performance of various techniques in case studies

FACTOR	REHOSTING	RE ENGINEERING	INTEGRATION
COST	LOW	HIGH	MODERATE
TIME	MODERATE	HIGH	MODERATE
USER EXPERIENCE	SAME	NEW & RICH	NEW
BUSSINESS FUNCTIONALITY	SAME	SAME	SAME
SKILL SET	SAME	NEW	MIXED
RISK	MODERATE	HIGH	LOW
BUSINESS PROCESS CHANGE	NO	YES	NO
DEPENDENCY	NIL	NIL	EXISTS

VII. Conclusion

SOA offers significant potential for modernizing interface to existing legacy assets, as well as exposing legacy functionality to a greater number of users. The SOA approach to systems development helps to achieve platform and language independence, reuse of legacy assets through loose coupling, and easy service upgrade. Further work can be carried out in enabling stored procedure as a service provider and enabling DB2 to invoke external services. CICS provides multiple way of securing the web service [4],

Acknowledgements

The authors would like to thank Mr. S V Subrahmanya (SVS), Vice President, Education & Research, Infosys Technologies ltd. for his guidance and constant encouragement.

Thanks are extended to Mr. Sarma KVRS for his critical review and suggestions for improvement. Thanks are extended to Mr. Bhargava Chandrasekara Sastry Kugur along with all the members of the Mainframe Education Delivery team for the help extended during this work.

The authors would also like to acknowledge and thank the authors and publishers of referenced papers and textbooks, for making available their invaluable work products which served as excellent reference to this paper. All trademarks and registered trademarks used in this paper are the properties of their respective owners / companies.

References

[1].http://www.mainframe-upgrade.com/article-soa-service-orientated-architecture.php

[2].http://www.soa.com/products/sola/sola_mainframe_soa/

[3]. http://it.toolbox.com/blogs/the-soa-blog/a-mainframe-soa-strategy-9792

[4].http://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.doc/InformationRoadmaps/ws_roadmap.html

[5].http://www.gtsoftware.com/content/mainframe-web-services

[6].ww.redbooks.ibm.com/abstracts/sg247206.html

[7].www.redbooks.ibm.com/abstracts/sg247669.html

[8].http://searchdatacenter.techtarget.com/news/article/0,289142,sid80_gci1250214,00.html

[9]. http://www.soamag.com/I36/0210-1.php

[10].http://www.ibm.com/developerworks/systems/library/es-soaonz/

[11].http://www.looselycoupled.com/opinion/2005/gilb-mainframe-eai0802.html

[12].http://www.ebizq.net/topics/legacy_integration/features/3548.html

[13]. http://adtmag.com/articles/2004/06/01/web-services-and-the-mainframe.aspx

[14].http://www.itanalysis.com/content.php?cid=10009

[15]. “Legacy Modernization- Modernize and Scale”- Venkataraghavan N Iyer-White paper