Survey on Service Oriented Architecture To Access Bank Data

Samrat Dhengle[1], Asst.Prof. Hirendra Hajare[2]
[1]Dept of CSE
[2]Asst.Prof, Dept of CSE
[1, 2] Ballarpur Institute of Technology(BIT), Ballarpur

Abstract- *The growing and complex business processes and activities require banks to spearhead financial services or financial transaction flows to meet customer expectations. The task of integrating such system is quite complicated, especially the integration of the financial technology system. In addition to being integrated, the flow of data and business processes must effectively reach every part which has interests in the process and the data. Business productivity can be achieved if the required data flows smoothly and is accessible as efficient as possible. Similarly, a bank which usually uses more than one information systems, when it needs to combine financial statements from different bank systems must go through a data transformation mechanism to be accessible. As a response to those situations, in this research we proposed an application based on SOA (Service Oriented Architecture) which could be used for different bank transactions. It is expected that the results from this study could help the parties interested in SOA-based application with the understanding and description of SOA, such as: how it works, its benefits, and the role of SOA layers etc.*

Keywords- BPEL, Choreography, Error Handling, ESB, Services, SOA, SOAP, WSDL.

I. INTRODUCTION

SOA is a business centric information technology architectural approach that promotes integrated and reusable business processes or services. In SOA, service is a fundamental element that can be independently developed and evolved over time. Each service is a self-describing, composable, open software component. Business Process Execution Language for Web Services (BPEL4WS) was proposed for depicting interaction of web services in order to provide a process service. BPEL can compose various fine-grained services or business processes with different capabilities into a requested coarse-grained business process. Service composition refers to the interoperation of autonomous and heterogeneous web services. BPEL provides an ideal way to composite services within SOA into complete business processes. However, web services usually communicate over internet connections that are not highly reliable. Web services can raise exceptions due to logical and execution errors [5]. BPEL uses provisions for exception handling and detecting failures, however, the inclusion of such provisions is a tedious assignment for the business process designer. Unlike in monolithic applications, error handling becomes a significant step in the design of SOA applications as SOA applications integrate heterogeneous IT systems across the organizational boundaries, vendor and partner IT assets. Focusing on error handling analysis early in the analysis and design phases ensures that appropriate error handling standards/guidelines are put in place for modules in different platforms. In this topic identifies common error handling considerations such that architects and designers can address the issues while designing SOA Solutions.

II. AIMS

One of the important aspects of exception handling is propagating sufficient information to upstream nodes when an exception occurs. However, when an exception happens inside a Web service, the details of the exception and contextual information is available only within the Web service. Web service specifications provide a SOAP details element in the SOAP fault structure to carry the exception details, but it is not mandatory for the service to populate the details. Also, the format/schema for carrying the exception details is not defined. This may lead to services populating the SOAP details element with their own custom format or ignoring this element completely [9]. So, service consumers either do not get the exception details or they get this information in different formats from different services. For example, an application using three services would have to have complex exception handling logic to deal with three different formats of exceptions. Web services do not have the capability of maintaining stack trace information, which is very important for root cause analysis of any exception. The errors logged by services need to be traced back across each service node until the end consumer is reached to perform a root cause analysis. To solve these issues, common exceptions need to be converted to a standard, predefined exception message to promote consistency and prevent ambiguity to the service consumers. For example, HTTP errors like 404 Not Found,

401 Access Denied, 500 Internal Server Error, etc. can arise because of issues in accessing underlying services, even though the applications they are directly interacting with work without error. A user using a Web site may get an "Access Denied" error when he clicks on a button because an underlying service being used is denying access. This may confuse the user, as he might have been successfully authenticated by the application he is accessing.

III. REVIEW

Service-oriented architecture or known as SOA is an information system architecture that integrates some parts of application architectures as services [12]. In that case, SOA can be defined as an arrangement of software architecture rules which each element can be interconnected with different software designs and architectural styles. SOA is generally built based on web services [9, 10]. A web service is a collection of services provided through a web-based network with predefined standards capable of supporting interoperability. Interoperability in web services means that the services offered can be operated by different standards and platforms [11]. In the banking domain, there is a business architecture similar to SOA. The major banks in the world have implemented it, such as for Point of Sale transaction (POS) data communication systems for credit card payment in stores and Auto Teller Machine (ATM) for the transfer, payment, and cash withdrawal process. Some software-based component architectures, such as CORBA, DCOM have been developed for such banking transaction needs [12].

Currently, some banks are beginning to switch to implementing SOA. Various channels provided by the bank for integrated transaction processing: ATM, Internet Banking, SMS Banking, Phone Banking, Teller, etc. can be easily connected to a global network. With the implementation of SOA, transactional process in banking is compliance with the standard of Bank 2.0.

AlgirdasAvi_zienis, Fellow, IEEE, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, Senior Member, IEEE, "Basic Concepts and Taxonomy of Dependable and Secure Computing".

This paper gives the main definitions relating to dependability, a generic concept including as special case such attributes as reliability, availability, safety, integrity, maintainability, etc. This paper gives the main definitions relating to dependability, a generic concept including as special case such attributes as reliability, availability, safety, integrity, maintainability, etc.

IV. SERVICE-ORIENTED ARCHITECTURE

Service-oriented architecture or known as SOA is an information system architecture that integrates some parts of application architectures as services [1]. In that case, SOA canbe defined as an arrangement of software architecture rules which each element can be interconnected with different software designs and architectural styles.

SOA is generally built based on web services [2, 3]. A web service is a collection of services provided through a web-based network with predefined standards capable of supporting interoperability. Interoperability in web services means that the services offered can be operated by different standards and platforms [4]. In the banking domain, there is a business architecture similar to SOA. The major banks in the world have implemented it, such as for Point of Sale transaction (POS) data communication systems for credit card payment in stores and Auto Teller Machine (ATM) for the transfer, payment, and cash withdrawal process. Some software-based component architectures, such as CORBA, DCOM have been developed for such banking transaction needs [5].

Currently, some banks are beginning to switch to implementing SOA. Various channels provided by the bank for integrated transaction processing: ATM, Internet Banking, SMS Banking, Phone Banking, Teller, etc. can be easily connected to a global network. With the implementation ofSOA, transactional process in banking is compliance with the standard of Bank 2.0 [6].

Recently, the banking domain has become one of the most complicated branches in software market due to its need for advanced techniques and technologies for building and integrating systems in it. Basically, this complication comes from the need to develop various systems/modules to cover different needs. For example, each bank has a need for a branch automation system in order to perform most of daily tasks performed by tellers with their clients such as opening new accounts, transferring funds from one account to another, cash withdrawals and deposits. Another example, such as the need for loan origination system for defining, granting and promoting loans' offers to bank customers, and the need for a core banking system that allows keeping track of all transactions occurring in bank.

Such systems can be viewed as silos that consist of a set of different modules. These modules always use different dialects coming from different software vendors with varying technologies of implementation, architecture patterns, and hosting platforms, which make it hard to integrate them

effectively and efficiently. Also, these silo modules and systems always encompass redundant functionalities and business processes and this for sure adds extra costs for both implementation and maintenance issues.

The basic assumption of SOA is that there are many consumers that require services. In literature, consumers are also referred to as clients or customers. These terms are used interchangeably here. On the other side, there are many providers that provide services on the network. These two groups have to be linked together in a dynamic and adaptive way. This is usually done by a service broker [9], [10].

Service providers register their services at the broker (registry), service consumers request a service from the service broker, which returns a known provider for the requested service. Consumer and provider agree on the semantics. The consumer then binds himself to the service provider and uses the service. The structure of this architecture is shown in Figure. 1.

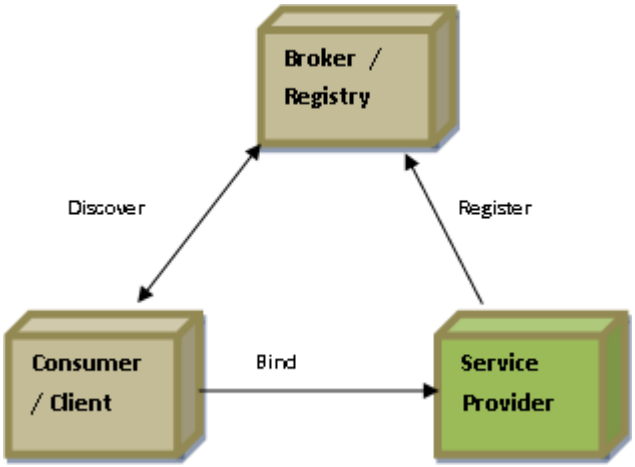

Figure 1: SOA Structure

SOA-specific Errors and Failures

In terms of fundamental concepts of dependability [11], threats for computer systems include errors, faults and failures. An error is that parts of the system state that may cause a subsequent failure: a failure occurs when an error reaches the service interface and alters the service. A fault is the supposed or hypothesized cause of an error. All faults are gathered into three major fault classes for which defenses need to be devised: design faults, physical faults, interaction faults. We proceed from the assumption that most of the errors and failures occur during service binding and invocation, messages transferring and requests processing by web service. In this paper we specified different types of SOA-specific errors and failures (Table 1).

Table 1: SOA-specific errors and failures

Type of error/failure	Error /failure domain
Error in Target Name Space	Client-side binding errors
Error in web service name	
Error in service port name	
Error in service operation's name	
Output parameter type mismatch	
Input parameter type mismatch	
Error in name of input parameter	
Mismatching of number of input service parameters	
Web service style mismatching	
Suspension of web service during transaction (getting into a loop)	Service errors and failures
System error during processing (like "Divide by Zero")	
Calculation error during processing(like, "Operand Type Mismatch	
Application error raising user exception(defined by developer)	
Network connection break-off	Network and system failures
Domain Name System (DNS) is down	
Loss of packet with client request or service response	
Host unavailable (off-line)	
Application Server is down	

V. PROPOSED SYSTEM

Unlike in monolithic applications, error handling becomes a significant step in the design of SOA applications as SOA applications integrate heterogeneous IT systems across the organizational boundaries, vendor and partner IT assets. Focusing on error handling analysis early in the analysis and design phases ensures that appropriate error handling standards/guidelines are put in place for modules in different platforms. This topic identifies common error handling considerations that architects and designers need to address while going through the SOA solution design.

SOA analysis and design tasks are broadly classified into three major phases i.e. Service Identification, Service Specification and Service Realization as identified in Service Oriented Modeling and Architecture by Ali Arsanjani. Subsequent discussion of this topic is oriented around error handling considerations that apply to these three phases.

Error Handling during Service Identification

The goal of service identification is to come up with a candidate service portfolio that leads to identifying re-usable service portfolio. This phase involves analysis of business

artifacts package that includes key requirements, business goals, capability models, Business Process Analysis Model (BPAM), use cases, etc.

Error Handling during Service Specification

Service Specification phase consists of tasks defining inputs and output messages, service and operation names, schemas, service composition, non-functional requirements and other service characteristics such as sync/async, invocation style, etc. for the services that are marked as to be exposed.

Error Handling during Service Realization

Service realization phase is where the service model is mapped to service component and runtime/deployment model. This step typically involves designing service components, allocating the components to SOA stack layers choosing component interaction styles, runtime platforms and making architectural design decisions. Subsequent discussion of the subject will be focused around some best practices to implement error handling considerations in the three layers of typical enterprise SOA stack: business processes or choreography, mediation/BUS and component layers as highlighted in figure below

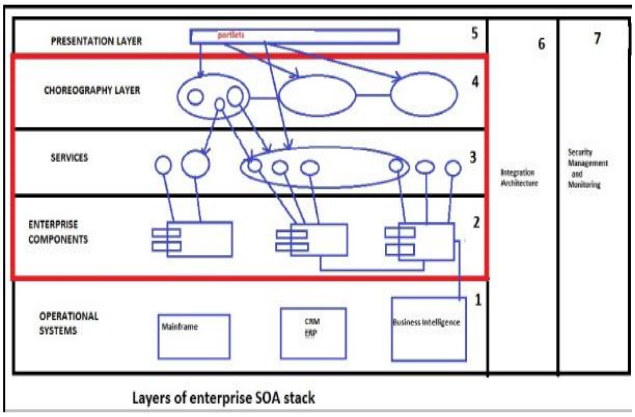

Fig. 2: SOA Enterprise Layer

VI. CONCLUSIONS

In this topic Financial Technology prototype which is designed to simplify, plan and execute for financial transaction services needs of our day-to-day activities by using the SOA approach. The integrated Enterprise Service Bus (ESB), will allow users to combine their bank transactions into a single platform. Besides that, the ongoing updates for different transactions will be handled by the integration of ESB.

REFERENCES

[1] ArdiSuryatmojo,Emil R Kaburuan, Ahmad NurulFajar, SepsellonaSutarty, Abba SugandaGirsang, "Financial Technology Integration Based on Service Oriented Architecture", Conference Paper · October 2018

[2] Mahmoud. Q , "The Road to Enterprise Application Integration (EAI) ", Service-Oriented Architecture (SOA) and Web Services published April 2005, Available: http://www.oracle.com/technetwork/articles/javase/soa-142870.html, [Accessed: 30 Apr 2014].

[3] Rozlog. M, "REST and SOAP: When Should I Use Each (or Both)", 01 Apr 2010 [Blog entry] Available:http://www.infoq.com/articles/rest-soap-when-to-use-each.

[4] Kareliotis Christos, Dr.Vassilakis Costas, D. G. P. "Enhancing BPEL scenarios with dynamic relevance-based exception handling". IEEE (2007).

[5] Poolla, H." Error handling considerations in SOA analysis and design". Enterprise Architecture (2010).

[6] Shukla, R. K. "Exception handling in service-oriented architecture". HCL Technologies (2006).

[7] Stefan Bruning, S. W., and Malek, M. "A Fault taxonomy for service-oriented architecture". IEEE (2007).

[8] Papazoglou, M.P., and Georgakopoulos, D. (2003), Service Oriented Computing: Introduction, Communications of the ACM, 46(10), pp. 25-28.

[9] Meredith, L. G., and Bjorg, S. (2003), Contracts and types, Communications of the ACM, 46(10), pp. 41-47.

[10] Ali Arsanjani, Ph.D (2004), Service-oriented modeling and architecture, pp 2.

[11] Josuttis, Nicloai M.(2007), SOA in Practice, O'Relly 2007, pp 210.

[12] Baskerville, Richard. (2005), Extensible Architectures: The Strategic Value of Service-oriented Architecture in Banking, Department of Computer Information Systems, Robinson College of Business, Georgia State University.

[13] King, Bret. (2010), Bank 2.0: How Customer Behaviour and Technology Will Change the Future of Financial Services – Marshall Cavendish Business.