

July 2013

Role of Web Services in migration of Legacy System to Service-Oriented Architecture

Vinay Goyal

Dept. of CSE, Jind Institute of Engineering & Technology, Jind., Haryana, India, vinayeq@yahoo.com

Amit Jain

Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India, amit_jainci@yahoo.com

Follow this and additional works at: <https://www.interscience.in/ijcsi>



Part of the [Computer Engineering Commons](#), [Information Security Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

Goyal, Vinay and Jain, Amit (2013) "Role of Web Services in migration of Legacy System to Service-Oriented Architecture," *International Journal of Computer Science and Informatics*: Vol. 3 : Iss. 1 , Article 8.

DOI: 10.47893/IJCSI.2013.1116

Available at: <https://www.interscience.in/ijcsi/vol3/iss1/8>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer Science and Informatics by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Role of Web Services in migration of Legacy System to Service-Oriented Architecture

Vinay Goyal¹ & Amit Jain²

¹Dept. of CSE, Jind Institute of Engineering & Technology, Jind., Haryana, India

²Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India

E-mail : vinayeq@yahoo.com¹, amit_jainci@yahoo.com²

Abstract - Managing the convolution of systems is considered as a key confront in computing and is transact with numerous expansion projected at enhanced computerization. Service-oriented architectures (SOA) are predicted to facilitate the prerequisite of a large quantity of varied and contending web services which the potential users will be capable to prefer dynamically, in the intend of receiving at every instance to the best possible assistance for their rationales. Web Services are set of software system designed to support interoperable machine-to-machine interaction over a network. To be adjustable, service arrangement should be performed by energetically picking and compiling the contributing services according to numerous eminence standards, so that the users persistently obtain best possible outcome.

I. INTRODUCTION

Service-oriented architecture is related to utilizing the service-orientation theory to produce an enterprise IT design; to tackle the business tribulations, the service identification, granularity, and construction need to be derived from the business process design. Service-Oriented Architecture should be open to allow quite a lot of services to put in and steer clear of inequitable choice. Openness entrust Service-Oriented Architecture to a distributed architecture for inflowing and parting resources that are bound to be decentralized.

With the increasing classiness about how and where modernism occurs, companies know that business flexibility is the driver. The flurry of new ideas comes from inside the business as well as from online resources like wikis, blogs, networking sites, clients, and even entrants. This world requires association to plead for the thoughts and suppleness to act in response to those ideas. The impending is that CEOs now articulate that additional ideas for innovation come from associates and patrons than from their own recruits. The fascinating harmony here is that all these new ideas originate from some sort of association, but to operate on those ideas, business flexibility must be a at the top of priorities.

As of now certain organizations are looking forward to newer technologies and in process replacing the older systems, still certain legacy systems are not subject to transform, since they often hold the most

significant business practices. The reality is that IT environment is a set of irreconcilable systems, technologies, and proprietary data classifications. Till now, generating a competent and business trustworthy IT background so that information can be cohesively common amongst systems and applications still seem impossible.

Service-Oriented Architecture (SOA) plays a vital role in effective integration concept to triumph over the complexities of building agile business systems. SOA is aimed at designing the framework with a theory of encapsulating business logic within services, so as to bind the information. It has an interface described in a format that machines can process. Other systems interact with the web-services in a manner prescribed by its description using SOAP messages, typically expressed using HTTP with XML serialization in union with other web-related standards.

In addition to these, Legacy programs, i.e. programs which have been developed with an outdated technology make-up for the vast majority of programs in several user applications in today's environment. It is these programs which actually run the information systems of the business world. Moving to a new technology such as service-oriented architecture is impossible without taking these programs along. Services that group to hold business processes within SOA are designed in such a way that different parts can operate autonomously of each another. As a result, any

feature can be altered without floating other specifications of the application. This makes companies that have adopted principles of SOA much more receptive to changing business necessities than those that rely on traditional software development procedures, with one feature change potentially stunning the whole application. The businesses that have adopted SOA technology function more proficiently than their competitors and acclimatize extra rapidly to altering business environment in their businesses.

II. INTRODUCTION TO WEB SERVICES

A service is basically a well summarized business utility with an apparent uniqueness and programmatic available interfaces. These services are usually executed using Web service technology, a standard recommended by W3C (The World Wide Web Consortium). The technology of Web services is the most likely correlation technology of service-oriented architectures. Web services basically use XML to build a robust connection. In the figure 1, service consumer at the left sending a service request message to a service provider at the right is shown. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined in some way that is understandable to both the service consumer and service provider. A service provider can also be a service consumer.

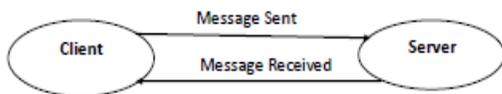


Fig. 1 : Communication between Client & Server

In addition to this, Web 2.0 facilitates the collaboration aspects, and SOA enables the infrastructure for flexibility. A great example is a retailer deciding whether to issue a credit card to a customer. It could use the technology to tap different sources and pull together information on a customer’s creditworthiness and buying habits. A bank can use the same computing services to handle account transfer requests, whether they are coming from a teller, an ATM, or a Web application, avoiding the need for multiple applications. A manufacturer could measure more closely what is happening in its production process and then make adjustments that feed back instantly through its chain of suppliers.

SOA enables profitability through revenue growth and cost cutting. SOA enables innovation through collaboration and flexibility.

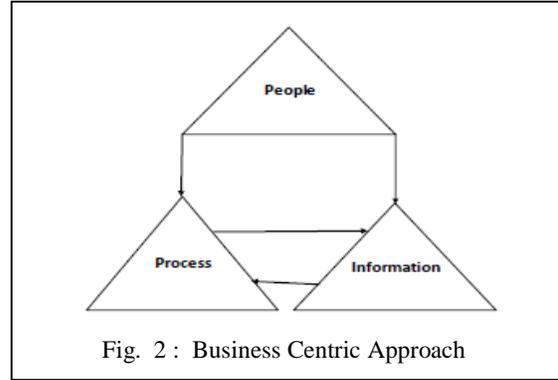


Fig. 2 : Business Centric Approach

Though in its formative years now, these SOA entry points promise to unleash capability similar to what the Internet—the prior technology evolution of comparable magnitude—already did. Companies employing SOA entry points countenance more than just technical challenges— there are process challenges and cultural issues too.

III. BENEFITS OF MIGRATING LEGACY SYSTEMS

The goal to incorporate a legacy system into the SOA is to make it a part of a rapid and flexible IT infrastructure. As a result, more business processes can utilize a legacy system without duplicating similar functionalities. It also leverages existing assets to have cost control and better management. From a technical perspective, the transformation improves interoperability with systems inside and outside of the organization. Utilities don’t need to discard existing legacy systems in order to support new technology.

The benefit of integrating legacy systems within the SOA can be summarized as:

- Revealing useful business logic and data – As mentioned earlier, certain legacy systems carry out the majority of significant business processes. Since technologies used to realize legacy systems didn’t provide well-defined interfaces for association with other systems, valuable business logic and data are not competently shared.

Through service enabled interfaces, legacy systems are capable to depict helpful information with a typical and interrelated strategy.

- Functionality Reprocessing – Unlike conventional proprietary point-to-point amalgamation approach, service facilitates a customary and open way to put together legacy systems within the SOA. This significantly promotes the functionality recycling which refer to utilizing the similar functionality continually without reworking the system.

- Capability to uphold the varying business – In the ever changing market the business process can change frequently. With the help of optimum utilization of services, legacy systems can familiarize themselves to the altering business quickly.

Though SOA plays a vital role in several perspectives, but still, there are numerous areas where many issues need to be determined, in case of integrating legacy systems within the SOA.

- Managing Proprietary data – Majority of legacy systems have proprietary data definitions. In the beginning, these data explanation were intended to persuade the internal system prerequisite and not projected for data exchange and interoperability. The gist of these data definitions might be only explicable by their owners. The semantic gap between legacy systems and other applications needs to be linked.
- Service recognition, categorization, and comprehension – Nowadays, business logics and functionalities within legacy systems are generally firmly united. There is a requirement for careful consideration of the components designed as services and their exposure to service-oriented architecture. Granularities of every service too deserve careful plan because it may affect the liveliness of the business as well as the intricacy of executing these services. Interaction between services facilitates legacy systems and other services inside the SOA are an additional design group. It has noteworthy influence to how services are ultimately realized.
- Effect on Effectiveness – One of the prime apprehensions concerning the service-enabled approach is the effectiveness. Certain legacy systems carry out enterprise's the most critical business process and generally have high effectiveness requirement. Making some of the legacy functionality service enabled may affix operating cost to it. The effectiveness shall be well understood and minimized.

Thus, Service-oriented architecture is a way to amalgamate business with a set of associated services so that the standard of unfastened pairing can be realized.

IV. RESOURCES OF WEB SERVICES

The main objective of service-oriented architecture is to create the software functionality accessible to all who require it and who are approved to utilize it. Also, they should be capable to mingle the functionality in whichever mode they consider fitting, i.e. to implant it as phenomenon in their business processes. By invoking

the methods offered by the service-oriented architecture they can fulfill the functions related to within their business process language (BPEL) measures without having to code and analyze them. The value for that is modeling the advancement of legacy systems to the WSDL interface, by citing the factors according to the interface requirement.

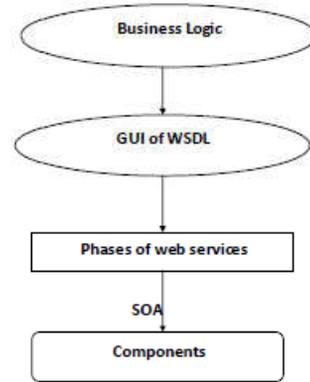


Fig. 3 : Resources of web services

Big client organization can have a unique improvement assembly of individuals to generate such common services, just as was the case with the familiar subroutine libraries and the familiar class libraries. There is no real difference here, only the interface languages changes. As an alternative of processing factors lists or linkage sections, the developers now have to transact with WSDL schemas. Besides developing the services, the user also has to analyze them. This could be an impediment to a lot of customers who are not comfortable in testing phenomenon. The testing of a WSDL interface is even further challenging than testing a GUI. The GUI can be formed and authorized visually at assessment time. The WSDL is invisible. The interface has to be produced by a program and sent to the intentional service via a middleware fabrication. The outcomes have to be acknowledged by the agenda; verified and authenticated against the predictable outcome. Thus each & every aspect call for skills and skilled testers, amazingly which numerous enterprise lack in.

The web services can also be created with the aid of existing software. Every user organization which has been utilizing information technology for any length of time will have accumulated a significant amount of legacy software. Some of this software will be tightly coupled to the environment for which it was developed, in particular the presentation software which is presenting maps or GUIs. Other parts of the software will be tightly coupled with a particular database system, namely the data access software. In so far as the same database is used for web applications, this

software can be reused. Also, the application software will be devoted to processing the business logic, which has to be extracted from the existing code.

Tracing and recovering such business-oriented software is related to software recycling. It is alike to recovering important building blocks from the wreck of an older construction in order to reclaim them in a new structure. The expertise has been in existence; since the mid 1990 s and has been well enclosed in the reengineering literature. The concept of reutilizing the old code blocks as web services in service-oriented architecture is new phenomenon altogether.

V. IDENTIFICATION OF WEB SERVICES

With reference to the marketing director of the Software A.G., the core functionality of most public administration offices is buried deep in their existing application software. [14] It is ineffective to endeavor to replicate it in another form. The only realistic elucidation is to enclose it and make it available as a general public service. What is not mentioned here is that this functionality must first be recovered and carried out from the deepness in order to reprocess it. E-government has become a chief contender for the methods of software reprocessing. The same applies to the functionality in business administration. There are several organizations specific tasks unique to every enterprise. Classical examples are the methods of payment, the conceding of credit, the calculation of interest charges and the management of fortunate clients. Traditional business processes have plenty of user centric solutions which have been developed and refrained over several years. They are a crucial element of the business procedures. The dilemma is that this customized business logic is not actually accessible.

Clients desirous of migrating to a service-oriented architecture must formulate a portfolio study of their existing applications and then jot down the necessary business rules. In that attempt, it will be necessary to split the multifaceted regulations down into basic rules which are self-contained logical units. All of these basic rules are contenders for web services. They have partial parameters, i.e. input variables and a single compound

result. As a result, it can be fitted suitably into a few business processes. The next step is to review the business worth of these reprocessing aspirants. The key to defining suitable web services is the granularity of the services. They must be broken down to a level of granularity where each service performs a single well-defined alteration or calculation upon a narrow set of parameters to provide a particular outcome.

VI. CONCLUSION

Service-oriented architecture is implemented by Web services and other technologies, but the terms and concepts have recently gained popularity as a result of Web services. Existing traditional management applications can be loosely coupled with additional service management applications according to processes of a service provider. If there exists a method for monitoring the whole quality of service-oriented software, the service selector component of the architecture, which has the responsibility of choosing the best candidate service at runtime, can be aware of the quality state of the whole software at any moment of decision.

REFERENCES

- [1] Axel Hochstein, Rüdiger Zarnekow, Walter Brenner: Evaluation of Service-Oriented IT Management in Practice, Proceedings of the International Conference on Services Systems and Service Management ICSSSM 2005, Volume 1, 2005.
- [2] Ali Arsanjani, Bernhard Borges, Kerrie Holley: Service-Oriented Architecture, Published in DM Direct, November 2004.
- [3] Litoiu, M.: Migrating to Web Services a performance engineering approach Journal of Software Maintenance and Evolution, Vol. 16, Nr. 1, 2004, p. 51
- [4] Warren, Ian: The Renaissance of Legacy Systems, Springer Pub., London, 1999, p. 2

