A Novel Methodology for an Efficient Embedding of Legacy codes in SOA the Service Web based workflows

Ms. T.Lavanya
Head, Department of Computer Science, Govt. Arts College for Women, Krishnagiri, India

Abstract: In this paper, we present a generic optimization of legacy codes in SOA workflows. To ease the embedding of Legacy-codes in the Service-based framework, an application independent service is required. To address the problem of exposing legacy code command-line programs as services, tool is described with the Legacy codes. This is hardly compatible with our approach, which aims at the whole application execution at run time. This embedding service highly simplifies application development because its enable to wrap any legacy code with a minimal effort. In this paper, we first discuss Service-based code invocation. We defined the benefits of the Service-based approach from a user point of view. The application-code in the Service-based framework at very little cost, allowing an easy composition of complex application as service-based workflows. Thus the idea of SOA and web service technology are also analysed in detail.

Keyword: SOA, Web Services, Service-based Workflow, Legacy codes.

I. INTRODUCTION

One of the vital reasons, the technological advancements and changes in usability also motivate reverse engineering to exploit new features and incorporate them in legacy software systems to achieve the goals of SOA. The platform put the focus on the integration of legacy applications. The main contribution of this paper is a description of the framework used to wrap legacy application to expose themselves as services and the methodology used to facilitate the replacement of these applications by more adequate services[1]. We then propose an application-independent service wrapper to ease the migration of existing application code in the legacy codes of service based framework. We propose a complete SOA of the system as a legacy codes. The code invocation is straightforward, through the legacy code command-line.

II. SERVICE-ORIENTED ARCHITECTURE

Service-Oriented Architecture (SOA) is an information technology advancement in which the already existing applications of an organization employed with the various services available in a network. These types of services which can be easily integrated with the existing one can be developed by using SOA. On the basis of analysing the advantages of SOA and web service technology in realizing heterogenous system integration, a fast and flexible enterprise information system integration scheme based on combining SOA and web service technology is proposed to solve these problems of poor information sharing capability and business adaptability. The services are loosely coupled to applications, so they are only used when needed. They are also intended to be easily used by software creators, who have to create application in a reliable way. SOA is to control the used of these services to avoid security problems.

Security in SOA spins heavily around the security of the separate components within the architecture, distinctiveness and authentication measures related to those components and security the actual connections between the components of architecture service-based software architecture are actually just modernized implementations of the SOA model. The software components are developed as services to be exposed via APIs as SOA. SOA principles have engaged us to the cloud and are supporting the most advances cloud software development system is use today.

SOA has capability to adjust or modify the different external environments and large applications can be managed easily. In recent years, the SOA has been widely applied in system integration. The SOA is used to implement the standard and loosely coupled application architecture by using the characteristics of web service. Zhang and zhang[7] proposed an integrated service model based on three loosely coupled perpectives according to the services, using a SOA.

Based on web service execution automation, the proposed appraoach is the gap between based integration and SOA by enabling transparent integration of information parlanti[8] proposed a Service-Oriented approach based on SOA, the service and application integration system for data integration.

III. SERVICE-BASED CODES

This approach was widely adopted for dealing with heterogeneous and distributed systems. In the Service-based approach, the code is embedded in a standard service shell. The standard defines an interface and an invocation procedures. The Service standard[2] supported by the W3C is the most widely available although many existing implementations do not conform to the whole standard yet.

The Service-based approach is also often referred to as meta computing. The main advantage of the service based approach is the flexibility that it offers. Clients can discover and invoke any service through standard interfaces without any prior knowledge on the code to be executed. This approach delegates to the server side the actual code execution procedure.

In the case of legacy code application, it is often not the case and an intermediate code invocation layer or some code reworking is needed to exploit this paradigm. Some tooling are available for helping in generating service interfaces but they can not be fully automated and they all require a developer intervention. In the case of legacy code, recompilation for instruments the code may be very difficult or even impossible.

Therefore, the only way to deal with legacy code in a user friendly way is to propose a Service-complaining code execution interface. In the Service-based approach conversely, the actual code invocation is delegated to the service which is responsible for the correct handling of the invocation parameters. The service is a black box from the user side and to some extent, it can deal with the correct parametrization of the code to be executed.

A service dynamically input as parameters. The Service-based approach enables discovery mechanisms and dynamic invocation even for a priori unknown services. This provides a lot of flexibility both for the user (discovery of available data processing tools and their interface). Services are adding an extra layer between the code invocation, on which jobs are submitted. In the Service-based framework, the code reusability is also improved by the availability of a standard invocation interface. Services are naturally well adopted to describe applications with a complex workflow, chaining different processing whose outputs are piped to the inputs of each other.

A. Workflow of Services

Services are naturally well suited for representing and chaining workflow components. The Service-based approach has been implemented in different workflow managers such as the kepler system [3] which can be standard Web-Services. Building applications by assembling legacy codes for processing and analysing data is very common. Its allows code reusability without introducing a tool high load on the application developers.

Many workflow representation formats execution managers have been proposed in the literature with very difficult properties. When dealing with workflows, the Service-based paradigms exhibit new fundamental differences. Based on the static description of such a workflow many different optimization strategies for the execution have been proposed [4]. An example of the flexibility offered by the Service-based approach is the ability to define different data composition strategies over the input data of a service. Workflow services are WCF-based services that are applied using workflows. Workflows that use the messaging actions to send and receive windows communication foundation. As applications develop increasingly distributed discrete services convert responsible for calling other services to offload some of the work. Workflows are a usual way to express the coordination of asynchronous exertion; particularly calls to external services are also active at representing long-running business processes. It is these qualities that make the workflow to great asses to building services in a distributed environment. The data is represented as data contracts and message contracts. The services itself exposes metadata to describe the operators of the service.

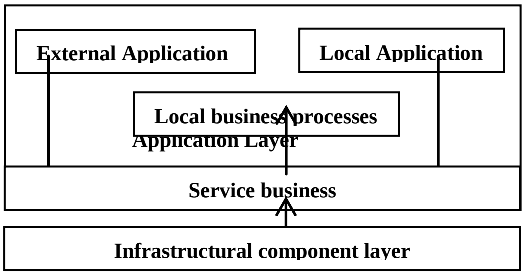

Figure.1. The Hierarchy Based on web service and SOA

The implementation of service uses infrastructural components. The hierarchy based on web services and SOA is shown in Fig.1 Application layer includes the local application and external application. A business process is consisted of a series of activities. A basic activity correspond a service. Since business process becomes a service flow.

Then the basic services must meet the requirements of definition and function singularity in the SOA. SOA is a key to design and hypotheses the loosing coupling software system for commercial functions in the way of programmable and available services. To a certain extent, the SOA is a model to be used for designing, developing, deploying and managing the discrete logic units in computer. This paper suggests a loosely coupling data application combination stage based on SOA and web service in order to improve the reusability and extension capacity.

IV. LEGACY CODE

To ease the embedding of legacy codes in the services based framework, an application-independent service is required. We briefly review systems that are used to wrap legacy code into services to be embedded in service-based workflows. The interfaces have been widely adopted for the wrapping of legacy codes into services. All of the reviewed existing wrappers are static, the legacy code wrapping is done offline, before the execution. It enables the execution of any legacy executable through a standard service interface. Legacy code refers to an application system source code type that is no longer supported. Also refer to unsupported operating systems, hardware and formats. Usually means that the major bugs have been fixed on worked around, and often it means that it was written, needs refactoring to bring it into modern standards.

CONCLUSION

This generic service highly simplifies application development because it is able to wrap any legacy code with a minimal effort. In a SOA it is preferable to split these services for several reasons. The web services represents legacy code wrapping services, the code wrapper service is responsible for dynamically generating and deploying application services. To expose legacy codes as autonomous web services respecting the main principles of SOA. A web service which wraps the submission of legacy code to the generic service wrapper. The composite web service is of the similar type than any even legacy code wrapping service. In this paper, we first described an application independent legacy code wrapper that works at runtime. We finally introduced SOA compliant architecture of whole system that fully automatizes the legacy code wrapping and the grouping strategy procedures. Thus the idea of SOA and web service technology are also analysed in detail. So a fast and flexible scheme of enterprise application integrated based on SOA and web services is proposed to improve the poor information sharing capability and business adaptability in this paper. The SOA is used to implement the standard and loosely coupled application architecture by using the characteristic of web service.

References

[1] Web Services Architecture requirements, w3c web services architecture working draft, available online at http://www.w3c.org/tr/2002/wd-wsa-reqs-20021114

[2] World Wide Web Consortium, Web Services Description.

[3] B.Ludascher, I.Altintas, C.Berkley, D.Higgins, E.Jaeger, M.Jones, E.A.Loe, J.Tao and Y.Zhao,

Scientific workflow management and the kepler system. Concurrency & Computation 2005.

[4] T.Delaitre, T.Kiss, A.Goyeneche, G.Terstyanszky, S.Winter & P.Kacsuk. A Running legacy code Applications, Journal of Grid Computing(JGC), 2005

[5] Y.Huang, I.Taylor, D.M.Walker and R.Dacies Wrapping Legacy Codes to Grid Based Applications. IEEE Computer Society 2003.

[6] G.Kecskemeti, Y.Zetuny, T.Kiss, G.Sipos, P.Kasuk, G.Terstyansky & S.Winter, Automatic deployment of interoperable legacy code services. In UK e-science. Sep 2005.

[7] L.J.Zhang and J.zhang, "An integrated service model approach for enabling SOA", IT Professional vol.11, No.5,(2009), pp 28-33.

[8] D.Parlanti, F.Paganelli, and D.Giuli, "A Service-Oriented approach for network centric data integration and its application", IEEE systems Journal(2011).

[9] G.Lewis, E.Morris and D.Smith, "Service-Oriented migration and reuse technique" proceedings of the 13th IEEE International workshop of software Technology and Engineering.(2005)

AUTHOR'S BIOGRAPHY

T.Lavanya is presently working as Assistant Professor and Head, Dept of Computer Science, Govt. Arts College for Women Krishnagiri, India. Her research interest includes Digital Electronics, Computer Architecture, Computer Networks, Web Services, Data Mining and Wireless Access Protocol. She has twenty years of teaching experience in the field of Information Technology and Computer Science. She has attended many conferences and several papers to her credit.