2009 IITA International Conference on Services Science, Management and Engineering

Why Or Why Not Service Oriented Architecture

Ying-Hong Wang
Department of Computer Science & Information
Engineering, Tamkang University
Tamshui, Taipei County, Taiwan
inhon@mail.tku.edu.tw

Jingo Chenghorng Liao
Department of Computer Science & Information
Engineering, Tamkang University
Tamshui, Taipei County, Taiwan
jingo@cs.tku.edu.tw

***Abstract*—Service Oriented Architecture (SOA) is a new computation technology in these years. Though this architecture, business application can be wrapped as loosely-coupled component. Base on business needs, connecting different component into necessary services of real process, enterprise can utilize existing applications repeatedly, then integrate the independent applications into new services. This paper focused on the issues on which enterprise needs to consider before implementation of SOA. We have a further discussion on advantages and disadvantages for adopting SOA. Once SOA is implemented what we need to care about? It is helpful on your long term success if you can thoroughly understand the key points of this paper, then the enterprise can well evaluate if its information system needs to adopt SOA.**

Keywords- Service Oriented Architecture, SOA, Web Services, Advantage, Disadvantage.

I. INTRODUCTION

SOA is a new computation technology in these years. It's not only satisfied user's application needs, but also included service broker and service provider interaction. All these are related to standard SOA. In spite IBM and Microsoft have hyped the benefits of SOA, most of enterprises are currently on the very beginning step and there is a long way to go.

As SOA is service oriented, that will treat all existing applications or components as service models, and communication between services is by way of message exchange, and it solves the system integration problem by using of loosely-coupled models. This paper focused on the issues on which enterprise needs to consider before implementation of SOA, then we have a further discussion and introduction, enterprise can evaluate whether their information system can adopt SOA to come out a best solution.

This paper is organized as follows: Section 2 describes the related works for SOA existing technologies. Section 3 and 4 present why and why not SOA. And we have the key issues for speculating when applying SOA in section 5. The last part of this paper is our conclusion.

II. RELATED WORKS

This chapter is mainly described some SOA technologies. SOA is a kind of software architecture, and it transforms different function of software process into services. Through these services, it can define well sequence implementation and interface, so we called it loosely-coupled which has neutral interface. The requirement of loosely-coupled system, SOA will follow business application process, hence the requirement becomes more flexible to fit the variable environment.

To implement SOA, first of all, we need to understand the related technologies for SOA. The major technologies include: XML, SOAP, WSDL, UDDI and ESB.

III. WHY SERVICE ORIENTED ARCHITECTURE

The difference between SOA and traditional application architecture is SOA emphasizes interface, protocol, communication, coordination, working process, search, cooperation, publication. All these are through XML, SOAP, WSDL, UDDI and HTTP, by using of the common standard. That allows the development in different platform, and exchange data. In addition, SOA can use searching service on running period in Internet, and that's the data what we want. By using of the policy to dominant and coordination, and control various servers, the policy can also process additional development to designate more checkpoints and enforce on the policy, meanwhile, control during the whole procedure, to ensure reliability and accuracy of data. When the server environment is under distributed system, it can use ESB to perform mutual communication mechanism.

SOA has the following technical characteristics, such as distributed architecture, loosely-coupled interface, open standard and process centric and so on.

SOA component is combined by various systems which are distributed on the net, either LAN or WAN. By way of this, web services become the useful technology for all the platforms in Internet.

Three reasons are for "Why SOA", according to Christina Lau's description[1].

1) It's the popular field currently, not out-of-date.

2) Combination of tools, fundamental structures and standards can provide overall support for entire SOA lifecycle.

3) SOA is helpful on efficiency.

This paper comments on the above 3 viewpoints, you might not agree that SOA is the most popular field, because not everyone is using this technology. SOA practical technology is mostly using existing technology, and it's convenient for utilization and integration. These standard technologies enable us to look for reusable resources, not need to start from brand new. We can look for existing service, adjustment, utilization, share on certain services,

978-0-7695-3729-0/09 $25.00 © 2009 IEEE
DOI 10.1109/SSME.2009.126

then establish a co-existence system. Finally, we will come out more meaningful solutions.

Bobby Woolf mentioned the following 4 conditions which are suitable for applying SOA[2]:

4) *When data* distribution is very high, you can use SOA.

5) When you need the function is highly available, you can use SOA.

6) When the individual parts of an application need to be developed, maintained or updated independently, you can use SOA.

7) When many applications need reusable function and data, you can use SOA.

In addition, according to Holt Adams[3]SOA development experience, he believes the following reasons brought us to adopt SOA:

1) Integration cost keeps increasing, and no slow down in spite of new opportunity generates real return on investment.

2) Merger and acquisition is core business for expanding market share and development opportunity.

3) Solution is integrated on business function of heterogeneous systems and programming models.

4) Business survival relies on the ability of fast adjustment on market changes and immediate reaction on competition.

5) Global economy effects request your company to develop business efficiently, and rely on business partners to provide non-core business.

6) The coordination efficiency of business partner is the key for raising revenue.

7) Your business asset value is decreasing, as it's unable to evaluate, then it can use on other places rather than original ones.

8) There is something wrong with your staff efficiency because they did not spend most of time on core functions and services of business models.

9) Your company is full of opportunity type business.

10) Your company begins to develop new applications.

IV. WHY NOT SERVICE ORIENTED ARCHITECTURE

SOA is popular because of its flexible and reusable architecture. But the service runs relatively slow under this kind of architecture. Since Gartner raised the idea of SOA[5][6]in 1996, SOA soon became the most famous product in view of its cost-saving feature and speed-up new business process. However, SOA needs to be additionally wrapped as a standard component through intermediate software. The extra process may slow down the whole run-time due to additional computation, and hence lower the performance.

According to news reports, as SOA getting popular, many experts raise 10 viewpoints of SOA[4]. For the system based on SOA architecture, application function is combined and structured by loosely-coupled and uniform interface components. SOA resulted huge influence in business software architecture. Though SOA can en-linear for application development, it can not solve all problems. We need to know the following 10 points about SOA.

1) SOA is only an architecture: It's a set of best practice, not technology. SOA can do what you want to do, not what you can buy.

2) SOA is not a panacea: SOA can reduce overall cost, enhance assets reuse, provide business transparency. But if your questions are not listed here, SOA may not be a right choice for you.

3) Business should treat SOA by using repeating ways: The certain step at certain time is the best, and every repeating will bring real business value, so you don't have to use SOA everywhere but gain success.

4) The problem of SOA is organization, culture and politics: People are unwilling to accept change and share resources, neither devote for other people's items.

5) The most technical challenge of SOA is service abstract establishment and maintenance: Business service hides the complexity of IT business and users. In fact, SOA increases its complexity.

6) Core architecture problem of SOA is control, quality and management: It's destined to fail without control.

7) No similarity on any two SOA applications: Different business selects different implementation solution according to individual requirement, so SOA's best solution is changed base on different condition.

8) The implement of SOA doesn't need additional intermediate software: Enterprises usually have sufficient intermediate software. SOA is for thoroughly driving the intermediate software, not for buying them.

9) Pure buying/selling doesn't need SOA: What enterprise's need is solution of business problem. SOA is the unique key of IT and solves business problem for enterprises.

10) SOA is still immature: We are still searching for the best practice, and specify the standards. You just need to know not only SOA's advantages, but also its disadvantages.

Bobby Woolf also mentioned the following 4 conditions which are not suitable for applying SOA, and need to be careful[2]:

1) When you need to develop as simple as possible, do not use SOA.

2) When you require the operating environment is as simple as possible, do not use SOA.

3) When the net is not reliable or the net speed is slow, do not use SOA.

4) When you process prototype design, do not use SOA.

The last but not the least, Bobby Woolf also mentioned the timing for alert on the issues. Frankly speaking, we need to keep alert anytime.

In addition, Holt Adams[3]also mentioned if we need to adopt SOA. In fact, it's usually influenced and limited to past decision. The following situation does not need to adopt SOA, and we will further explain SOA's marginal effects.

1) Your company just uses small part of IT budget in integration.

2) Most of your company's procedures are by manual works or documents, and no possibility of automation.

3) Most of your company's applications are using the same programming model.

4) Your company's operation is managed by one or two CRMs and ERPs, and no requirement of integration.

5) Your company's current technology container has big difference with the technology container of SOA infrastructure.

6) No further benefits on business requirement or opportunity from SOA.

7) New business service brings negative effects on current status.

8) The business partner is adopting different priority on flow automation between companies.

9) Your major business development involves huge affairs which are highly demand synchronously and instantaneously.

The above illustration is an example only to explain if SOA is your best choice. Every contract or item has sole requirement, so when you need to adopt is based on your business condition. SOA's value is attractive, however, when your company will adopt SOA needs to consider the real situation of business environment. Adopting of SOA is proceeded gradually, no necessary to have a big step. First you need to find the item for using SOA concept and principle, then you use the major function target to measure the value. It's a good way that benefits all of us.

V. Key Issues for Speculating When Adopting Service Oriented Architecture

This chapter is mainly describing the key issues to be noted for adopting SOA.

First of all, there are SOA's 11 misconceptions[7], which usually resulted from software vendor's hype. We aim to clarify these misconceptions rather than to disappoint anyone. Meanwhile, we wish enterprises can much understand the expected outcome when adopting SOA.

1) SOA provides the complete architecture for a system.

2) Legacy systems can be easily integrated into an SOA environment.

3) SOA is all about standards and standards are all that is needed.

4) SOA is all about technology.

5) The use of standards guarantees interoperability among services in an SOA environment.

6) It is easy to develop applications based on services.

7) It is easy to develop services anybody can use.

8) It is easy to compose services dynamically at runtime.

9) Services can only be business services.

10) Testing applications that use services is no different than testing any other application.

11) SOA can be implemented quickly.

Finally, the above intent is not to discourage organizations from adopting SOA, but to caution them about some important issues and risks to consider while creating their SOA strategy. We believe SOA may be the best approach available for achieving several critical interoperability, agility, and reuse goals.

Although SOA is a software architecture, it can be composed by web service components of the different functions. However, some people treated web services as distributed object technology. Werner Vogels used to seriously mentioned "Web Services are not distributed objects"[8], to clarify several widely held beliefs about the technology that are partially or completely wrong. We briefly describe the paper as follows:

1) Web Services Are Just Like Distributed Objects.

2) Web Services Are RPC for the Internet.

3) Web Services Need HTTP.

4) Web Services Need Web Servers.

5) Web Services Are Reliable Because They Use TCP.

6) Debugging Web Services Is Impossible.

When enterprise wants to adopt new technology, it will consider how to replace some legacy system resources, but they usually suffer from some unexpected situation. If enterprise wants to adopt SOA to legacy system, it should note related issues and problems. Here we will describe the main concept of Xiaofeng Wang et al. paper[9].

1) In respect of a system, data representation needs to be realized by whole system, as the whole system requires an interface to access it.

2) How to identify the services of legacy system? And to determine the application of service granularity and construct services.

3) When a legacy system is integrated to SOA, performance is always the major concern. Especially when using web services, data translation and payload transportation are always discussed.

Major description of the above paper is that developer must completely understand legacy system's architecture and workflow, and transform existing functional components into service components, then integrate required legacy system components and resources. In addition, it also mentioned efficiency issues after integration.

Furthermore, we will discuss the key elements we need to pay attention if we want to implement SOA successfully[10]:

1) SOA is a long-term strategy for enterprise scope. The team is responsible for formulating internal standards, blueprint reference architecture, design model, template and some shared standard services, and also has experts in specific fields as part of the team.

2) To formulate implementation roadmap just is like any other good plans. A common strategy is starting from creating portfolio for current status and future status, and these portfolios make things easier for reviewing whole condition and realize system's interoperation.

3) Only the sound architecture fundamental can achieve SOA advantages. From the viewpoint of SOA, the whole architecture and planning design in every aspect is very important.

4) The classification includes deeper understanding field, vertical technology and process, business technology standard, emerging technology and trend, compiling requirement, knowledge of developing platform, model, best practice, item management and test.

5) As item and technology portfolios have different satisfaction and professional knowledge on technologies, every organization's delivery model is different, so the model should keep consensus in every task and item. And then when developers transfer between different items, they will have similar experiences and minimize learning curve.

6) Having a sound management model can ensure service architecture, and gradually grow. It is important for management to ensure no repeated action and item is depart from SOA.

7) One of the enterprise architecture team missions is continuously checks. Enterprise architecture team and business experts need to ensure SOA steps always follow strategic business target if the short-term business target conflicts with long-term business target. And then the team should consult with senior management to find out the best solution.

8) If no good communication channel, architecture could be worse. Hence continuing communication is very important, then the enterprise should hold cross-team meeting to find the successful way.

9) Supporting from senior management is important, not only for control of conflicts, but also for raising of capital. SOA includes pervious investment, unless IT has support from senior management, otherwise, it's difficult to go further.

10) SOA is not a model for one time use. On the contrary, it includes continuous development and re-design. Following the pace, we will undergo several replace. For every replace, the feedback returns to service and has further refined.

VI. Conclusion

Service Oriented Architecture is an emerging model of system architecture. It is a software component which is composed based on enterprise requirements. The components includes software component, service and workflow. When enterprise is facing external requirement, the workflow will define the process steps. While the service includes all components of specific steps, and the component is responsible for the specific work, SOA has become important technology for today's software development. It simplifies the integration of heterogeneous systems, then the reuse rate of program is getting higher. The developer is not necessary to develop or own all components. Instead, the developer can compose the best services on the net base on enterprise needs. By no limitation on function and platform of the certain vender's products, it helps to achieve the real openness.

The way to SOA is just like savings for retirement, and it's a long-term investment. All of SOA development team may experience short-term panic, but it finally will get rewards. In spite SOA is not panacea, it helps enterprise to get the best answer when facing problems. SOA is not only technology, but also re-arcitechture of working process. Thoroughly to realize this paper's description will make sure your long-term success. Currently, many enterprises are considering of information system design and upgrade, before adopting SOA, they need to concern about some issues. They need a further discussion and introduction, including advantages and disadvantages of adopting SOA, and once SOA is adopted, what should be concerned? As mentioned on the above, we hope you can thoroughly realize this paper's description, and you will make sure your long-term success.

References

[1] Christina Lau, "No Myth Existing Here", http://www.ibm.com/developerworks/cn/webservices/ar-itio1 (Chinese webpage)

[2] Bobby Woolf, "Suitable and Non-Suitable Situation and Key-Points That You Need to Pay Attention", http://www.ibm.com/developerworks/cn/webservices/ar-itio1 (Chinese webpage)

[3] Holt Adams, "When to Adopt SOA or Not to Adopt SOA", http://www.ibm.com/developerworks/cn/webservices/ar-itio1 (Chinese webpage)

[4] Na Bu Han, "10 Viewpoints You Need to Know about SOA", http://webservices.ctocio.com.cn/comment/314/7612314.shtml (Chinese webpage)

[5] Gartner, "Service Oriented Architectures, Part 1", SSA Research Note SPA-401-068, 12 April 1996.

[6] Gartner "Service Oriented Architectures, Part 2", SSA Research Note SPA-401-069, 12 April 1996.

[7] Grace A. Lewis, Edwin Morris, Soumya Simanta, Lutz Wrage, "Common Misconceptions about Service-Oriented Architecture", Commercial-off-the-Shelf (COTS)-Based Software Systems, ICCBSS 2007. Sixth International IEEE Conference on Feb. 26 ~ March 2 2007, Pages 123~130.

[8] Werner Vogels, "Web Services Are Not Distributed Objects", Internet Computing, IEEE. Date Nov.~Dec. 2003. Volume 7 , Issue 6 , Pages: 59~66

[9] Xiaofeng Wang, Shawn X.K. Hu, Enamul Haq and Harry Garton, "Integrating Legacy Systems within The Service-oriented Architecture", Power Engineering Society General Meeting, IEEE, 24-28 June 2007, Pages 1~7.

[10] Anant Kadiyala, "10 Elements for Successfully Implement of SOA - Accurate SOA Development Requires Re-Design of Software Development", http://dev2dev.bea.com.tw/techdoc/01wp/01wp_041105.htm (Chinese webpage)