

## The convergence of SaaS and SOA magnify effect

Jin Xiaoye, Liu Xianhui\*

Engineering Research Center for Enterprise Digital  
Technology of Ministry of Education  
Tongji University  
Shanghai, China  
\*amjinxiaoye@yahoo.cn

Zhao Weidong

Computer Aided Design Research Center  
Tongji University  
Shanghai, China  
lxh@tongji.edu.cn

**Abstract**—SaaS is a software distribution model in which customers can receive their special types of software application services from the SaaS provider via the Internet, the specified software can be ordered according to customer’s demand. SOA is a valid architecture pattern to provide integrated systems, applications and processes through the linking of services. With the help of SaaS model, distributed networked software services can be seamlessly integrated and deliver “on-demand” applications to customers. However, the benefit of SaaS can only be fully exploited if they leverage the SOA strategic framework technology. This paper does researches about recent SaaS, SOA and web services and proposes a new model to show how to combine SaaS with SOA. This combination makes it possible to efficiently develop loosely coupled SOA-based SaaS application. Moreover, in reverse, SOA can be enriched by this convergence.

**Keywords**-SaaS; SOA; web service; convergence; semantic

### I. INTRODUCTION

Software as a service (SaaS) is a software application delivery model by which an enterprise vendor develops a web-based software application. Software and associated data are centrally hosted on the cloud. SaaS application is typically accessed by users using a thin client via a web browser [1]. They only pay monthly fees for using the software. Now, a large number of enterprise software vendors begin to move their application to SaaS, and more and more customers also show interest in SaaS model. Despite its increasingly growing adoption, SaaS applications still face challenges: agility, configurability, and extensibility. Recent SaaS vendors only deliver packaged version of software, which makes it difficult to meet radical different demand of customer [2].

Service-oriented architecture (SOA) is a strategic framework technology for designing and developing software in the form of interoperable services. However, well-structured SOA-based application lacks a convenient and efficient way for delivery, and the ability to expand services within an SOA is typically a painful and expensive process [3]. SaaS is accelerating the adoption of SOA by providing aspects of SOA on demand.

In order to solve different problems both SaaS and SOA may encounter, we propose a new model that let both draw from each other’s strengths. The goal is by constructing the SaaS platform to publish applications and services on the platform and rent those to users through Internet for use [4].

We can deliver SOA-based software application through SaaS platform, so that we customize qualified software according to users different business requirement to the maximum extent. Meanwhile, the SaaS provider and existing SaaS application also benefit SOA. Therefore, the convergence of SaaS and SOA can generate/produce magnifying effect.

### II. RELATED STUDIES ON SAAS AND SOA

#### A. Recent SaaS and shortages

SaaS has become a common delivery model for many business applications, including accounting, collaboration, CRM, MIS, ERP, invoicing, HRM, CM and so on. An increasing number of enterprise software companies migrate their business into SaaS platform. Therefore, some notable SaaS service providers emerge. Such as Salesforce.com, the top SaaS vendor in CRM; Google Apps offering office automation applications including Google Docs, Gmail, and Google Calendar; Amazon Web Services; Windows Azure.

As we can see in Fig.1, many SaaS providers offer SaaS software applications in form of editions, each having a set of features. The drawbacks are obviously, customers can not choose the features they actual need. Instead, they have to subscribe to a whole edition and pay for that. Since the main characteristic of SaaS is Multi-tenant, different users have their unique business model and requirements, so it’s unreasonable to pack the same software for all the customers. SaaS provider should offer custom-made services [5].

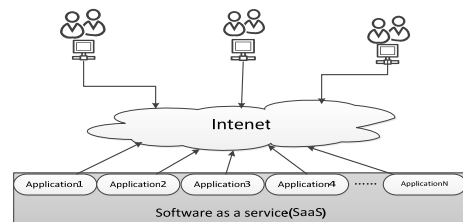


Figure 1. recent SaaS supply model

In order to achieve the essential purpose that delivered software can be configured and customized SaaS provider should not only rent common software applications, but also focus on abstracting some changeable features, then encapsulate them into components, so that the applications constructed by SaaS can be composed with a series of module according to multi-user’s needs. Since some SaaS companies like Salesforce.com have opened force.com

platform and AppExchange market to enable customize application, but it is still at the start stage and has limitations [6]. How to leverage the more developed SOA and already existing web services to enrich SaaS platform is the issue that we should do more research on. Note that SaaS should take full advantage of the characteristic of SOA services like statelessness, low coupling, modularity, and semantic interoperability [7]. Weaving SaaS delivery services into applications within service-oriented architecture can fully satisfy user's need.

**B. Service-oriented architecture and principles**

Fig.2 shows us a clearer concept [8]. From bottom-up, SOA is our best approach to fixing the broken architectures by reusing the legacy system. Extracting useful loosely coupled functions and then packaging them into small units which allows other interested systems, inside and outside an enterprise, to expose and access well-defined internet-based services, as well as information bound to those services. while from the top-down side, in order to form diversity business logic to meet users' requirements, we use the collection of abstract services to composite applications for solution development in process layers, thus allowing us to deal with system changes by using a monitoring configuration layer rather than constantly having to redevelop new systems[9].

SOA focus on forming demand-led software to meet particular requirement. The key is a set of distributed services that can be configured and bound at delivery time. Those services are governed by following principles: reusability, abstraction, loose coupling, autonomy, encapsulation, granularity, composability, componentization, interoperability [10].

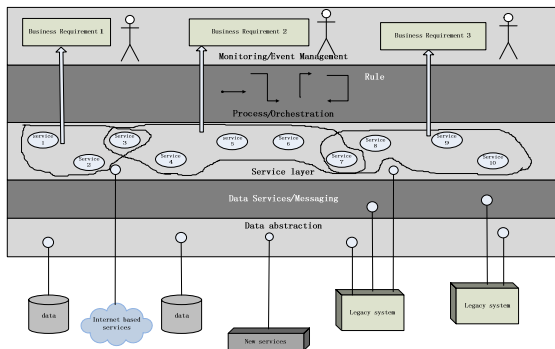


Figure 2. loosely coupled architecture of SOA

**C. Semantic web service approach**

Different from supply-led service model, demand-led SOA architecture has a integration layer to enable service management. We usually use WSDL to describe services. It's the foundation of mapping provider's offerings to the customer's needs, then customers can use service discovery (UDDI) to find the appropriate services. The recent widely used description language WSDL typically describes only data types, method, messages format, and URI of service on syntax level [11]. In order to describe the functional and

nonfunctional aspects of services in a semantically rich, ontology-based format, scholars begin to use OWL-S, which is a machine-readable language depicting web services logically, so that we can automate discover, invoke, composite, monitor web services. Due to traditional description method, UDDI's limitation is searching keywords only because lack of semantic description, which restricts model's dynamic discovery capabilities. By extending UDDI with OWL-S we can enrich its inferential capabilities. Meanwhile workflow-based integration step can also incorporate semantic information, making the whole process more accurate and intelligent.

**D. SOA benefits and Challenges**

The primary benefits of an Service-oriented architecture include leveraging remote application behavior as if it existed locally, reuse of the same functionality again and again, the ability to change business processes on the basis of existing services and information flows quickly and needful, to support a changing business, and exposing certain enterprise processes to other external entities for the purpose of enterprise collaboration or shared processes.

SOA-based environments can include many services that exchange messages to form a single application which may generate millions of messages. Managing and providing information on services interaction can become complex. Another challenge is about the way we use SOA-based application, how it can be more convenient for us to access on-demand software from Internet through pay-per-use [12]. We can address this serious problem by leveraging SaaS.

**III. SAAS PLUS SOA PROVIDE "SS" SERVICE**

The service in SaaS means operation service, which is a kind of profitable pattern, which needs a technology platform. SOA-related technology is the key part of the architecture, providing service management, scheduling, and maintenance, in order to construct more agile SaaS application. According to research, customers need not only software but also services in practice. So "S+S" means you needn't choose one of them [13], you can enjoy the magnified effect through the convergence of SaaS and SOA.

**A. Learn from each other**

SaaS Model is a newly developed software service, developing software and hosting service which enables access via Internet, being an efficient Software delivery mechanism, it has infused new energy into the evolution of information technology. However, SaaS is not a cure-all model. Facing multi-tenant, SaaS platform eagerly needs some sort of architectural discipline with guiding principles to organize its architecture, thus structuring mess system into smaller module for reusability and flexibility by using SOA. At the same time, SOA can provide a compelling business proposition when combined with SaaS. In a word, service-rich SOA and SaaS provide a great deal of value when they work together.

**B. Offering SaaS application as SOA service**

Reference Ali et al proposed model [14], we can use six

steps to move existing SaaS application towards SOA services, and Fig. 3 shows the specified steps. After the proposed six steps, SaaS vendors can act as service provider as well as service requester. The new SaaS delivered business applications will be a collection of services that communicate with each other, different SaaS vendor's applications can also interoperate with each other [15]. Meanwhile, during the Software architecture procedure, SaaS provider can acquire additional services from other providers, including different SaaS vendor, software firms, ISV, and then add them to its own application.

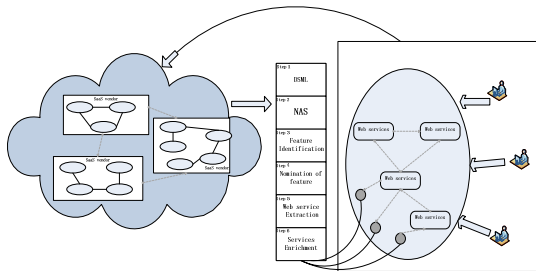


Figure 3. Figure 3 The Dream Team of SaaS and SOA

### C. SOA gets SaaS

SOA and SaaS goes hand in hand, refer to the proposed Fig. 3, we can consider SaaS the extension of SOA, the trick is to decide which functions should be abstracted within the existing or emerging SaaS platform and exposed to SOA, so that SOA can leverage SaaS resources as services and use them as if they were contained within the SOA. In this way big SOA was created. On the other hand, providing SOA service by the use of SaaS platform is a win-win solution, foreign media called this trend SOA gets SaaS [12]. We used to interpret as the convergence of SaaS and SOA.

## IV. NEW ARCHITECTURE PROPOSED BY THE CONVERGENCE OF SAAS AND SOA

In this part, we propose a new model showed in Fig. 4 to describe the way in which SaaS and SOA can leverage each other. Through the new architecture you can better mix and match SaaS services to meet the needs of the different business problems, making a more powerful SaaS platform as well as SOA.

a) *Business modeling*: Based on customer's demand analysis, we can use UML profile to define the business model. All rules, parameter and regulation are taken into consideration to differentiate differ from each customer's business requirement.

b) *Domain decomposition*: The goal of this step is to further investigate the defined model, The whole model will be divided into sets of loosely coupled subsystems, then maturity level of the subsystems that SaaS provider will provider to the specified customer will be measured, and nominated and suited system to be customized.

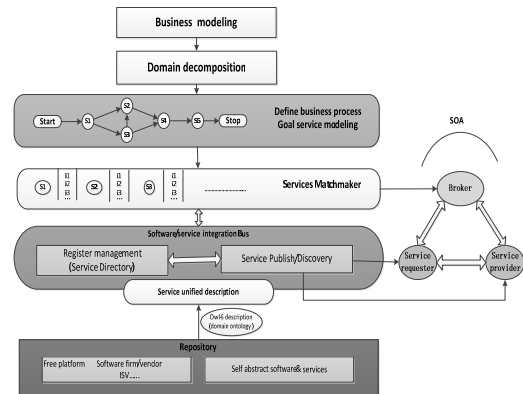


Figure 4. Figure 4 the combined architecture of SaaS&SOA

c) *Define business process/services*: in this step, we will detail UML diagrams which include two parts, one being the business process rules and workflow, the other constituting a collection of abstract services. Types of services are also well-defined. There are atomic services, composite service, process services, and external service.

d) *Service Matchmaker*: Fig. 4 shows us that each abstract service has a number of service instances which are from different service sources. For example s1 contains some instances like i1,i2,i3... In detail, matching step consists of two parts described in Fig. 5: first depend on the integration bus, search for request services which are existing in specified SaaS provider scope, if the semantic similarity of the returned maximum match value is below our pre-defined threshold, then we should go for SOA-based broker, take advantage of the vast services provided by multiple vendor. Meanwhile, we re-describe returned web service by projecting WSDL to OWL-S, then save it back to inner directory for knowledge collaboration.

e) *Integration bus*: millions of services were collected from bottom repository layer, before registering them into service directory, each of service will be enriched by the use of owl-s to form a unified description. During this transfer, domain ontology will be used. Through this step, a formal, semantically rich platform was created. As we can refer to in Fig. 4, services published/discovery mechanism should interoperate with service catalogue and its upper layer, which can enable SaaS provider be either the SOA provider or requester by exposing inside SaaS service to SOA and leveraging richer services outside within SOA.

f) *Service repository*: This is a big melting pot consisting of two major recourses: one is a free platform provided by SaaS vendor, so that external software firms, independent software vendors or other developer who are interested in it can use this platform to deploy its own application services or small functional granule. The other is self services which are extracted from existing SaaS applications through the six step depicted in previous chapter, offering SaaS application as SOA services.

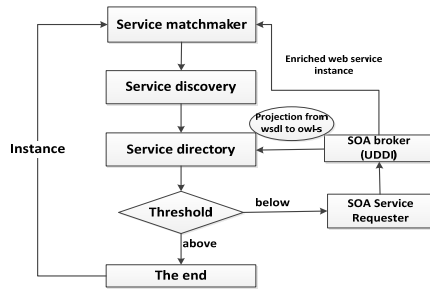


Figure 5. Discovery and knowledge collaboration of service

V. REAL-LIFE CASE STUDY

In order to clearly illustrate our proposed new model, we study a real-life scenario. With the closer cooperation between enterprises, a mold industry MoldA is starved of rebuilding its information and management system, so MoldA goes for a SaaS provider called TopSS to purchase their new software needs. Through the analysis of business requirement TopSS delivered a coordinated manufacturing system using our proposed model. The process can be divided into the following steps:

- Step1: Analyze MoldA’s business demand, define business model by using UML profile.
- Step2: Decompose the whole system, evaluate its maturity level, divide the system into several parts, mould design subsystem, manufacturing subsystem and sales software.
- Step3: Focus on the manufacturing and sales business entity, define business process. Business flow is showed in Fig. 6. Usually, customer login on the sales management system to search for their needed mould type. After customer submits specified order, if MoldA’s inventory is enough for their need, reach the deal. Otherwise flow to manufacturing process. In addition, to perfect the main flow, some other services may include as the picture shows.
- Step4: According to step3, TopSS can extract a set of abstract services, such as SaaS services={login service},{search service},{check service},{CRUD operation service},{order service},{quotation service},{premium service}..., TopSS should find the most suitable service instance of each abstract services for composition.

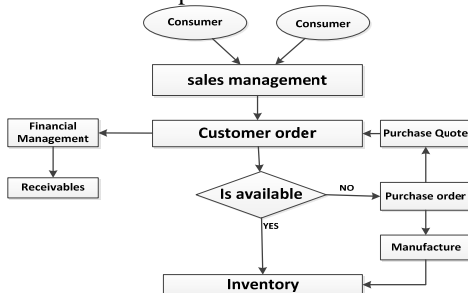


Figure 6. business flow of sales software

- Step5: Before the real searching we should do some preparation work, take {search service} for example, semantic description is a prerequisite. Here, we take advantage of OWL-S to describe input and output and function of {search service} which customers are most satisfied with, so that semantics-based matching can be used in the coming step instead of key-word searching, which can greatly improve accuracy.
- Step6: After all the preparation, TopSS first reach inner service directory for discovery, which will return returning the maximum similarity and corresponding service. If the similarity is above our predefined threshold, then it’s the service that we want to catch, other external query mechanism will be started, at this point TopSS will be a SOA service requester searching for distributed service that is based on service oriented architecture. The returned SOA-based service is usually described by WSDL, so before we register it in inner directory, the service description should be transformed from WSDL to OWL-S.
- Step7: After TopSS collects the set of services needed in the business flow, the last integration step can be carrying out.

After going through the seven steps, SaaS vendor TopSS leverage the new SaaS and SOA combined architecture to configure “on-demand” software according to customer’s need. In the process of assembly, TopSS enjoys the abundant services that not only exit in its own repository but also in millions of distributed services provided by SOA service provider. The proposed new architecture can greatly improve the quality of SaaS vendors’ delivery software, meanwhile provide a more agile method for SaaS vendor to assemble software.

VI. CONCLUSION

Along with the rapid development of cloud computing, SaaS model will be adopted by more and more customers. Some architecture problems may prevent us promoting SaaS mode. Fortunately, SOA approaches provide a great way to decompose the architecture to collections of functional behaviors, callable primitive and rebuild it as a process model; this approach provides SaaS the ability to leverage SOA resources and the ability to change the application as needed by business.

At present, the author proposed a novel architecture to combine SaaS with SOA. Application leased to users could be made up of plenty of services hosted in a dozen of different locations. In return, SOA can leverage SaaS services and use them as if they were contained within the SOA. SaaS and SOA provide magnified value when they work together. Thus, the SaaS vendors as well as service consumers can harvest the benefit of SaaS and SOA to create a win-win situation.

## ACKNOWLEDGMENT

This work was supported by Key Projects in the National Science & Technology Pillar Program during the Twelfth Five-year Plan Period of China (Grant No. 2012BAF12B11).

## REFERENCES

- [1] [http://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Software_as_a_service).
- [2] Ali Bou Nassif and Miriam A.M.Capretz, "Offering SaaS as SOA Service 2013," unpublished.
- [3] S. Linthicum David, "Cloud Computing and SOA convergence in Your Enterprise," Boston, MA 02116,2009.
- [4] H.Hong, "Applications deployment on the SaaS platform," ICPCA, 2010 5<sup>th</sup>. Maribor, Slovenia, 2010, pp. 232-237.
- [5] H.Liao, "SaaS business model for software enterprise," ICIME,2010 2<sup>th</sup>. Chengdu, China,2010,pp. 604 – 607.
- [6] W. Kima, Lee JH, C. Hong, C.Han ,H. Lee,B. Jang,"An innovative method for data and software integration in SaaS," Comput Math Appl,Vol. 64, September 2012, pp.1252–1258 , doi:10.1016/j.camwa.2012.03.069.
- [7] J. Jing and J. Zhang, "Research on Open SaaS Software Architecture based on SOA,"International Symposium on Computational Intelligence and Design, Hangzhou, 2010, pp. 144- 147.
- [8] The Linthicum Group ,SOA meta-model, 2007.
- [9] Y. Xi, H. Xue, C. Tingbin, "Knowledge Collaborative Supply Chain Management System Based on SOA under SaaS model", ICCSIT, 2010 3rd IEEE International Conference,Vol. 1,July 2010, pp. 327 – 330.
- [10] T. Erl," Service Oriented Architecture: Concepts, Technology and Design." Pearson Education, Inc., London ,2005.
- [11] Turner M, Budgen D, Brereton P," Turning software into a service,"Computer 36(10): 38–44,2003.
- [12] Yu, W.D. and C.H. Ong, "A SOA based software engineering design approach in service engineering", International Conference on e-Business Engineering, 2009 IEEE pp.409-416.
- [13] H. You, "Analyze living situation of SOA and SaaS in China, "Net Game Magazine ,Feb. 2012.
- [14] Ali Bou Nassif and Miriam A M. Capretz, "Moving from SaaS Applications towards SOA Services," 2010 6<sup>th</sup> World Congress ons services, pp. 187-188, 2010.
- [15] Ritu Sharma, Manu Sood, Divya Sharma. Modeling Cloud SaaS with SOA and MDA.2011
- [16] S. Ritu,S. Manu and S. Divya, "Modeling Cloud SaaS with SOA and MDA," In Proceedings of the International Conference on Advances in Computing and Communications (ACC 2011), Communications in Computer and Information Science, 2011, Vol.190, Part 5, 511-518, DOI: 10.1007/978-3-642-22709-7\_50.