A Synthetic Solution Scheme for SOA Security Assurance

Bing Xu1, Tianbo Lu1,2, Xiaojin Wang3, Lingling Zhao1, Xiaoyan Zhang1, Wanjiang Han1
1School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, China
2Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin, China
3China Software Testing Center, Beijing, China
lutb@bupt.edu.cn

Abstract - Due to the changes of architecture, tradition security mechanism can’t fulfill SOA security requirements. So it is the high time to design a comprehensive security assurance system of models and solutions that fulfill SOA and SOA-based applications’ security requirement without hurting SOA’s loose coupling and high scalability features. Based on the in-depth research of tiered SOA security, this paper provides a comprehensive solution and analyzes SOA security assurance on three levels: strategy, service solution, testing. Firstly this paper proposes a new security assurance model for the overall architecture, and then proposes a new framework as a practical application solution for modeling and evaluating reliability on single service, service pool and service composition. At last, compared with traditional application-oriented system integration testing, this paper proposes the strategies of integration test and test responsibilities division for SOA systems.

Keywords- SOA security assurance; reliability; service pool; integration test

1 Introduction

With the continuous development of computer technology, environment confronted by modern enterprise becomes more and more complicated, where a loosely-coupled, cross-platform, and language-agnostic system is needed to coping with changes of the outside world rapidly. In order to solve such puzzle, more and more enterprises have noticed SOA (service-oriented architecture). SOA is a service-oriented architecture. In addition, it can strengthen agility of enterprise businesses and reduce development costs of enterprise information system. Major advantages of SOA can be summarized as: providing business centric better and faster, rapid flexibility, and reusability [1].

There is still not a comprehensive solution about SOA security. And such distributed computing methods as CORBA and DCOM can provide loose coupling to some extent, but they can’t realize SOA better because of various limitations. However, it has become the set of some technologies which are best fit for realizing SOA because of maturity of web service standards and popularity of application [2]. It seems that there is an understanding in SOA field which can’t be denied, namely it is unnecessary to change current software testing methods. However, it isn’t the case. Change of testing methods is in bad need, and it is necessary to change many other things at the same time. Integration testing of SOA is particularly important because most tests are on single web services rather than sets of web service. In addition, safety issues can’t be ignored, especially for e-commerce platforms.

How to build safe networked transaction environment to ensure high confidentiality of transaction information in the network transmission process and not being threatened by such safety issues as forging, manipulation, and stealing etc., is an issue which must be considered for e-commerce platforms. E-commerce based on SOA conducts information exchange and completes electronic transaction on each heterogeneous platform with the use of XML technologies. At the time of information exchange, the following safety issues shall be considered: how to confirm accuracy of information exchange, identity of trading partners, and non-repudiation of transaction on platforms trusted by both parties.

The rest of the paper is organized as follows. Section 2 makes a research on tiered SOA security. Section 3 provides a security assurance model for the overall architecture. Section 4 proposes a new framework for modeling and evaluating reliability in web service system. Section 5 summarizes safety precautions. Section 6 proposes the strategies of integration test and test responsibilities division for SOA systems. Finally we draw a conclusion of this paper in section 7.

2 Tiered SOA Security

SOA security is reflected in the data safety of the SOA environment, the data is always in a different security level and facing security threats at every level. It can be summed up as following six security levels: transport layer security, message layer security, application layer security, data layer security, metadata layer security, management layer security. Transport layer security refers to the security of data in the point to point transfer by both sides. It is relatively easy to implement and there is already a series of transport layer security mechanisms. Message layer security refers to the
security of passing data between service consumers and service providers. Now WS-Security protocols ensure security of data exchanging based on SOAP. Application security refers to the security of the data in the application logic processing. It can’t achieve security assurance by Infrastructure transparent in this level. Data Layer Security refers to the security of the data encapsulated in service. It is crucial to protect the application related data in the stationary state. However, other data should also be taken into account in order to build a comprehensive and integrated security solution. The metadata layer security refers to the security of the public information in service. It needs to protect the definition of the service (WSDL document), port information, binding and protocol information, etc. Management layer security refers to the security of data being regulated. It includes preventing attacks on the management of infrastructure, preventing leakage of the authentication information or authorization information, preventing the abuse of service management permissions, etc.

No matter where the data is in the SOA environment, it always faces the following security problems: confidentiality, integrity, auditability, authentication management, authorization management, identity management, security policy management. Confidentiality is such a property that the data could not be acquired by non-authorized user or process, i.e., an eavesdropper or a non-authorized user can not view the secret message content. It can be ensured by typical encryption technology. It should be noted that confidentiality needs is different in transport layer and message layer. Integrity refers to the data has not been altered or damaged. The data received should be the same with the data source in a single message exchange, and this is mainly achieved by data signature mechanism. In multi-party interaction of the message, the data should not be changed in the transmission process, and it depends on the reliability of the transmission of the message. Auditability is a capacity which makes historical events relate to the perpetrators. Different systems may require different audit level. Some may only need a simple record, while some systems need to achieve non-repudiation, that is to say the initiator of the transaction can be identified as a unique entity. Authentication management’s purpose is to prevent using the service by Illegals which includes process, module and external systems. This is the first line of defense in system security controls. Authorization management is assessed by identity information and access control policy information: Whether a user has permission to access specific categories of resources. Due to the distributed nature of the resources in the SOA environment, the distributed authorization management of accessing to services will be the key to success. Due to the openness of SOA, the traditional information systems security boundary is breaking. It should be taken into account that how to pass the identifier safely between different trust domains. The strategy refers to a series of rules to manage the entire computing system behavior. The strategy can be used to describe all aspects of security, and can be applied to the all the SOA entities, including service consumers, service providers and infrastructure. It should be considered that how to manage security and make a set of policies to achieve interoperability from different entities.

In addition to the above analysis of the safety issues, it also faces replacement messages, message replay, message injection, session hijacking and other security threats in the SOA message exchange. These security issues seriously affect establishing a reliable relationship between the cooperation partners, and therefore it needs to establish end-to-end security services between service providers and consumers.

3 S3R SOA Assurance Model

SOA is the architecture about how the services combine together, and software is an important component of the service. Currently, software assurance which focuses on security, insurance, reliability and survivability has become the core of security, forming a multidisciplinary discipline of software engineering and information security [3]. The concept of software assurance was first proposed by Marilyn S. Fujii [4] and received close attention of the U.S. government. U.S. Department of Homeland Security, National Security Agency, NIST and had some research results [5] [6] [7] [8] [9]. The study of the software assurance in China begun since 2007, and we proposed the S3R software assurance model first. Combined with the characteristics of SOA, we propose an improved model, as shown in Figure 1.

This model includes four dimensions: functions, SOA assurance (SOAA) services, SOAA measures, and SOAA time. There are five functions: transport, service communication, service description, service, business process, and each function can be divided by a fine granularity. Corresponding measures are different in different functions.
Transport: It is a mechanism used to transfer the service request from the service user to the service provider and the response from the service provider to the service user.

Service communication protocol: It is a negotiation mechanism through which the service provider can connect with the service user.

Service description: It is a negotiation mode used to describe the service, how to call the service and what data is needed for successfully calling the service.

Business process: It is a collection of services. In order to meet business requirements, it can be called in a specific order and a specific set of rules. There will be the concept that a business process can be composed by different granularity of services if the business process itself is seen as a service.

There is one key consideration for service users and providers that is the security of the processes of dealing with the service components during their development and distribution. Indeed, SOAA involves a shared responsibility among providers, service intermediaries, and customers containing three factors:

- **Security:** In the processes of the SOA design, development and testing, security threats are anticipated and addressed. The focus of both quality aspects (e.g., “free from writing off the end of the array”) and functional requirements (e.g., “identity card number must be written in the police systematic database”), should be demanded.

- **Integrity:** The processes of creating and delivering service contain controls to strengthen the confidence that the functions achieve the goal the providers required.

- **Authenticity:** The service is real, not counterfeit and customers can use some methods to be sure that they own the real service.

### 4 Application Solutions

Figure 2 reflects a typical implementation frame of web service system. Actual physical structure is on the right side, including service portfolio flows on the top layer or logical layer and service (sets) (layers of middleware are omitted) on the bottom layer; basic implementation mechanism is on the left side.

Figure 2. Typical Framework of Implementing a Web Services System

Modeling and evaluation process of SOA reliability are filled among the whole lifecycle, so a corresponding interlayer shall be added in the basic framework above as the support [10], as shown in the figure 3. The following are some newly-increased steps when compared with Fig.2:

![Figure 3. A New Frame Including Reliability Modeling and Evaluating Process](image-url)
(1) Deliver constructed application model to the module charging reliability modeling of service portfolio. The module builds corresponding reliability models automatically or under manual intervention, which shall be stored in reliability model library. However, when application model changes, it may strike renewal of corresponding reliability models at the same time. Relative reliability models will be inserted when a portfolio of services are inserted into other service portfolios as a single service, and this can refer to Grassi’s work [11];
(2) Look up services and invoke service testing functions in service intermediaries with application builder, whose transfers shall be charged by the service-pool supervisor;
(3) Service-pool supervisor is responsible for building and maintaining a service pool for each component in the service portfolio and submitting them to reliability modeling modules through composing them into conditions, and reliability can be calculated synchronously;
(4) Service-pool supervisor invokes service monitor to complete service seeking;
(5) Service monitor looks up services required in service intermediaries according to requirements of service-pool supervisor, and at the same time, service monitor shall check whether service intermediaries have renewed relevant services at regular intervals;
(6) Service-pool supervisor invokes testing processes for found services and stores testing results into testing results/operational profile library. The service test is an external support technology. It started earlier and there are many results can refer to [12][13][14][15][16];
(7) Service monitor surveys running conditions of all services, which will submit monitoring and statistical results to testing results/operational profile library for conservation;
(8) After the above tasks are completed, reliability evaluator can be invoked manually or triggered automatically, which can acquire reliability models and reliability data from reliability model library and testing results/operational profile library, so as to complete reliability assessment of single services, service pools, service portfolios, and even the whole system, and assessment results shall be fed back to application modeling as bases for application modeling and service assembly.

5 Safety Precautions

Generally, there are several aspects of safety precautions aiming at safety issues confronted by each layer, including precautions in technology, management strategy, and educational training and improvement of safety awareness. Precautions in management strategy are supplementing defects discovered after design of safety system. For example, if there are defects in technology implementation, management strategy shall supplement them. However, the system may possess some potential threats with supplementary of management strategy, and educational training and improvement of safety awareness, the last layer of precautions, to is needed for supplementary. In this paper, precautions in technology are mainly concerned, so safety precautions in technology will be introduced mainly in the following, rather than other two layers of safety precautions.

In order to guarantee safety of SOA environment, it is necessary to adopt some new safety elements besides traditional safety technologies, such as data encryption technology, information signature technology, authentication technology, and licensing technology etc.

(1) XML Signature and Encryption

In order to guarantee safety of message exchange in SOA environment, confidentiality, integrity, and non-repudiation of SOAP information shall be considered, and mainly XML signature and encryption and WS-Security can solve safety technology of problems in these three aspects at present. Safety of traditional security mechanism communication layer is guaranteed with the use of SSL/TLS protocol, but it shall be implemented in SOA environment with the use of composite services units, which involves conditions of multiparty delivery of communication information. Furthermore, different parts of information contents may aim at different service providers, so it is inevitable to deal with information partially, which are solved by XML signature and encryption and WS-Security well.

(2) Single Sign-on

Generally, SOA environment is multi-domain, which is more complicated when compared with single-domain environment. A prominent problem in multi-domain heterogeneous environment is numerous users and widespread sources, especially service-oriented application. Widely researched single sign-on technology can solve identity authentication and identity management in the SOA environment well.

(3) Policy-based Access Control

Traditional access control frame couples safety logic and business application logic closely, with poor system flexibility, while service-oriented application needs a flexible way to carry out authorization management. Policy-based access control mechanism solves authorization management and strategic management in the SOA environment well. Flexible authorization policies can be formulated with the use of policy-based access control mechanism, so as to meet various access control demands of SOA environment.

6 SOA Integration Testing Strategy

Some modules in traditional testing can work alone, but whether they can work normally after connection is not been guaranteed. Issues that can’t be reflected in partial process
may be exposed on the whole situation, which may influence function realization. That is to say, the following issues shall be considered:

- Whether data passing through module interface will lose when each module is connected;
- Whether the predicted father function can be achieved through combining sub-functions;
- Whether functions of a module can have adverse effects on functions of another module;
- Whether global data structure has some problems;
- Whether errors of single modules can be enlarged through accumulation, to reach up to an unacceptable degree.

Therefore, it is necessary to conduct integration testing upon completion of unit testing, so as to find and to remove the above issues that may be occurred on module connection and to compose required software subsystem or system finally. It is the same to SOA testing. Unit testing of SOA conducts functional testing respectively towards each web service. At this stage, codes can be tested directly with the use of white-box testing, so as to find mistakes in programming inside modules and to guarantee that each web service can work normally. SOA emphasizes interoperation of web services in the framework, so integration testing shall be conducted upon completion of unit testing. The key to conducting integration testing towards SOA is testing information and communication errors that may be initiated among web services. Such communication errors may include: deletion, copying, delay, reordering, or false rumor of information.

### 6.1 SOA Integration Testing Contents

Integration testing of SOA mainly contains three role functions of SOA, namely release, binding, and finding, and asynchronous communication ability among web services, including functional testing and performance testing. In the following, functional testing and performance testing of SOA are analyzed generally.

Functional testing means testing functions of SOA system, mainly checking information errors possibly triggered at the time of interaction among web services, namely checking whether there are invalid operations in the calling process and web service can be invoked repeatedly. Invoking results of each service invocation includes success and failure. Generally, “failure” state is ejected through being translated into “abnormal” state in the system design and implementation where object-oriented thought is used widely. Customers shall deal with them further according to “abnormal” type and information after customer capture server ejects “abnormal” state. Abnormal state of web service system includes “abnormal system” and “abnormal customized definition”. The former indicates abnormal core ejection of web service system, such as abnormal network and abnormal HTTP Server; the latter indicates abnormal customized definition in specifically implemented web services, which has nothing to do with the core of web service system. Only these specific services and customers using such services share common “abnormal” meaning. Therefore, three different service invocation result states shall be tested at the time of functional testing, as shown in the table.

<table>
<thead>
<tr>
<th>Plan No.</th>
<th>Target Test Results</th>
<th>Client Input</th>
<th>Expected Output</th>
<th>Explain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Service call is successful</td>
<td>Correct input parameters</td>
<td>Correct service results</td>
<td>No abnormal</td>
</tr>
<tr>
<td>2</td>
<td>Abnormal user-defined</td>
<td>Wrong input parameters</td>
<td>Capture abnormal user-defined</td>
<td>System core normal, abnormal specific Web Service</td>
</tr>
<tr>
<td>3</td>
<td>Abnormal Service call</td>
<td>Correct input parameters</td>
<td>Capture system abnormality</td>
<td>Abnormal system core</td>
</tr>
</tbody>
</table>

Performance testing means the operation capacity of testing software in the integration system. It is impossible to provide software meeting functional requirements but not meeting performance requirements in real-time system. Performance testing can be generated in all steps during testing, but only when all compositions of the whole system are integrated, real performances of a system can be checked. One aspect of performance testing is pressure testing, mainly testing SOA system on several machines through operating several virtual users.

### 6.2 Integration Testing Strategy in SOA Development

(1) Integration Testing Strategy of Traditional Software

Three kinds of integration strategies are generally adopted in traditional software testing, namely top-down integration, bottom-up integration, and hybrid integration. Compile corresponding driver modules and stub modules as required and integrate step by step according to selected testing strategies. Complexity of software integration is greatly increased with the introduction, encapsulation, inheritance, and polymorphism of object-oriented development technology, and the above integration strategies are not proper again. What’s more, integration strategies based on usage scenario are mostly applied in the integration testing of object-oriented system.

(2) Duty Allocation of SOA System Testing Duties
In the SOA system, web services of system-based basic modules are provided by service providers but SOA system is charged by web service users, namely developers of the SOA system, and they belong to different organizations under this circumstance, so its partition in testing responsibilities is different from traditional software testing. Specific partition in testing responsibilities is shown in the figure.

[Image: Duty Allocation of SOA System Testing Duties]

Basic modules of SOA system are developed by web service providers, and codes are maintained by them, so unit testing shall be charged by web service providers naturally. However, web service users (developers of SOA system) are responsible for functional verification of web service. In addition, integration testing and system testing of SOA system are charged by the development team of SOA system. Acceptance testing is the same as that of traditional software development, which shall be solved by final users or personnel employed by final users.

(3) Preparations before Integration Testing – Test Single Web Service

Functional verification of web service can be conducted according to traditional black-box testing method, namely verify and confirm functions of single web service with the use of equivalence partitioning, boundary value analysis, cause-and-effect diagram, decision tables, orthogonal experimental design, and work flow according to web service’s WSDL. Web service to be applied in the system may come from different web service providers, whose development level and coverage degree of unit testing are unexpected. Therefore, it is necessary to verify and to confirm single web service. And at the same time, in the testing on single web service, testers test a great quantity of illegal inputs not meeting WSDL requirements in detail with the use of black-box testing method, so that developers have relatively in-depth understanding on such web service, which lays solid foundation on positioning and ejection of issues in the following integration process. Testing single web service fully is also in favor of increasing developers’ confidence on used web service, which simplifies the number of test cases in integration testing.

(4) Integration Testing Strategy

Among integration testing case design of SOA system, we choose testing strategies of scenario-based and top-down integration, whose design flow is shown in the figure.

[Image: Integration Testing Case Design Process of SOA System]

The integration strategy based on usage scenario can be adopted for the integration of SOA system. Each web service shall be integrated into the system successively according to different usage scenarios of users towards SOA system and framework of SOA system. Top-down strategy is adopted for the design of integration testing cases. SOA system is regarded as a black box, which provides certain functions for external users or other outside systems. Users input some parameters in the system interface, and the system may respond differently according to input parameters and show reaction results to external users through the system. In the design of integration testing cases, firstly, the above black-box testing methods (such as boundary value analysis method or equivalence partitioning method and so on) are comprehensively applied in the top layer of SOA, user input and system input are distinguished, and relevant testing cases are designed. Secondly, shielding of black box shall be removed layer by layer according to architecture design layer of the system, so as to show invocation relationship among web services layer by layer and to refine and divide corresponding testing cases further, till the bottom of web service. It has been found that testing cases designed with the use of such method are relatively complete, which covers various aspects of system function, tests invocation among web services completely, and is convenient for checking issues at the time of issues generated in the integration process.

7 Conclusion

As a new generation of application software architecture, SOA develops rapidly and is applied in various
fields with loose coupling and high scalability features. It provides great convenience for application integration within the enterprise and between enterprises. But there is still a lack of a comprehensive solution for SOA security problems. This paper provides a comprehensive solution and analyzes SOA security assurance on three levels: strategy, service solution, testing. Firstly this paper proposes a new security assurance model for the overall architecture, and then proposes a new framework as a practical application solution for modeling and evaluating reliability on single service, service pool and service composition. At last, compared with traditional application-oriented system integration testing, this paper proposes the strategies of integration test and test responsibilities division for SOA systems.

Nowadays dynamic and uncertainty of software development is greatly improved, especially for service-oriented architecture. So it's important to work on empirical studies. The further research is to develop a security assurance tools integration platform to fulfill the requirement of test.

8 Acknowledgment

This work is supported by the following programs: the National Natural Science Foundation of China under Grant No.61170273; Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (NO.CAAC-ITRB-201201); 2010 Information Security Program of China National Development and Reform Commission with the title “Testing Usability and Security of Network Service Software”.

9 References