International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(A)**, January **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 243–254

BUSINESS PROCESS AUDITING ON AN SOA FOUNDATION

Shing-Han Li[1], Shih-Chih Chen[1], Chung-Chiang Hu[1]
Wei-Shou Wu[1] and Mark Hwang[2]

[1]Department of Information Management
Tatung University
No. 40, Sec. 3, Zhongshan North Rd., Taipei 104, Taiwan
{ shli; scchen; cchu; wswu }@ttu.edu.tw

[2]Department of Business Information Systems
Central Michigan University
Mount Pleasant, Michigan 48859, USA
mark.hwang@cmich.edu

Received August 2010; revised May 2011

Abstract. *A traditional audit technique manually controls internal data and processes. However, with the increase of the informatics transactions between global organizations, auditors have to provide a rigorous audit process. Some of them try to deal with software products in a complex environment by using service-oriented architecture (SOA) and web services (WS). These pose new challenges for management, reliability, change management, security and much more. In this paper, a comprehensive audit mechanism is proposed to provide a cross-platform solution in a heterogeneous software/hardware environment, satisfying interdepartmental and cross-organizational business demands. A prototype of the collaborative e-procurement system is developed to demonstrate how business activities can be properly audited in the SOA architecture. Some practical examples are also given to illustrate the control points designed for a successful audit process in the system.*

Keywords: Service-oriented architecture (SOA), Web services, Computer audit, Audit log, Business process

1. **Introduction.** In recent years, service-oriented architecture and web services (SOA/WS) are emerging as the next major wave of application architectures for IT-intensive enterprises. Organizations are looking to SOA/WS to improve the speed, flexibility and the cost of building and deploying applications for both internal and external uses. Auditors can use software to assist in auditing such applications, for instance, ACL® (Audit Command Language), IDEA® (Interactive Data Extraction and Analysis) and embedded auditing modules. However, the use of web services and SOA poses unprecedented challenges to auditors, such as interoperability, performance and security. Even more, the legacies of auditing tools and methods neither work in the new environments nor satisfy information security demands of users [8]. In some other systems, although audit software for financial reporting is available, process audit and operation logs are still foreign to information auditors [9]. The audit mechanism proposed for the accounting system provides assurance about specific business processes [10]. However, it does not deal with the errors in data transmission. The Business Process Management (BPM) infrastructure provides the run-time environment for public and private process models. It allows users to monitor the execution of individual processes and to analyze the overall behavior of a set of business processes.

For the above reasons, we propose a comprehensive audit mechanism which provides a cross-platform solution in a heterogeneous software/hardware environment, satisfying interdepartmental and cross-organizational business demands. It is developed based on prior research on computer assisted auditing and audit logs [11-14] to ensure that all business activities in the web services environment will be properly audited. A prototype of the SOA-based e-procurement with partner relationship management (PRM) abilities is built to demonstrate the feasibility of the audit mechanism.

2. Literature Review and Background Knowledge.

2.1. **Evolution of the business process.** In the early days of computing, applications were developed from scratch. Each of these applications hosted its own data in a database. However, there are multiple data dependencies between these disparate systems, and changes in one system had to be mirrored by changes in related systems. Against this backdrop, Enterprise Resource Planning (ERP) systems were developed, which integrated disparate databases. The idea behind ERP is that the software needs to communicate across functions and ERP systems support efficient exchange of relevant data regarding the production processes and their associated administrative tasks [15]. In fact, ERP advocates believe that ERP could combine both business processes in the organization and IT into one integrated solution. The basic architecture of an ERP system builds upon one database, one application, and unified interface across the enterprise [16].

An enterprise can operate under one application standard whereby all applications serving human resources, accounting, sales, manufacturing, distribution and supply-chain management are firmly integrated. With this capability, ERP software can accelerate decision-making, decrease operating costs, and control over a globally distributed business operation [17,18]. In order to realize collaboration between suppliers, the manufacturer, channel companies and buyers, activities have to be performed in a coordinated fashion. As a result, there are multiple interactions between the enterprise and its business partners, which are hard to manage both economically and organizationally. Today, a new trend is emerging for e-commerce. Companies are beginning to leverage their e-commerce platform to cost-effectively and efficiently move from a model of independent multi-channel sales processes to an integrated and unified cross-channel sales model. Therefore, business processes for commerce applications are of a very different nature than ERP and becoming more complex as interactions with customers span multiple channels and touch-points. A new approach and software architecture is required.

2.2. **SOA and web services.** In recent years, service-oriented computing has become one of the major trends both in business engineering and software technology. The main idea of service orientation is to capture business relevant functionality as a service and provide sufficiently detailed information so that customers can use it. In the enterprise computing environments, the functionality of application systems can be described and provided by services. Service composition is an important concept for application development based on existing services [19].

With enterprise services architecture, a business scenario can be sketched, whereby new applications are built to interact with other companies such as perform delivery, check credit rating and e-bank (i.e., the three “clouds” in Figure 1). These systems expose web services via standardized interfaces. The business process controls the execution of the composed service. This is done by invoking the web services according to process orchestration, as shown in Figure 1. Using web services, it is easy to change service compositions of the enterprise application and implement the changes at a low cost [20].

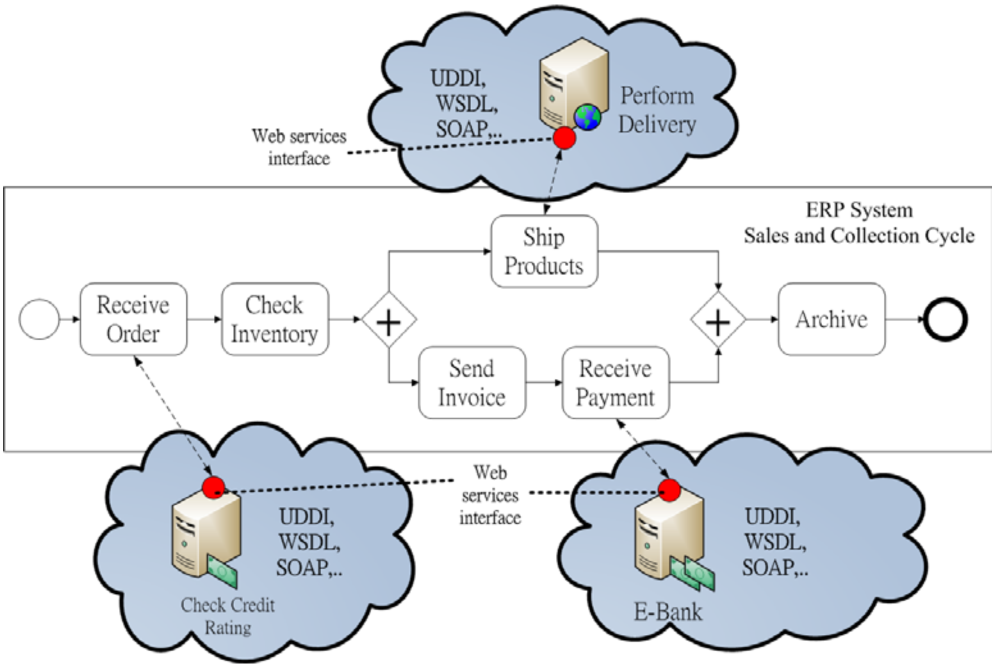


Figure 1. Enterprise services architecture

These features help the enterprise project developers quickly respond to the dynamic requirements of the business.

In e-commerce, one of the most concerned issues is the security of transactions, for instance, how to ensure that there are no data interception, tampering or fabrication during data transmission? Security has become the main factor that determines whether e-commerce succeeds or not [21]. Presently, most web services applications use XML documents to exchange data; however, previous research on information security concerns mainly with XML documents [22] and less with web services process control and monitoring. The prototype developed in this paper addresses this new security challenge.

When relying on accountants to audit complicated e-commerce systems, new audit risks will increase substantially. Audit failures are likely to result, especially among information systems using the Internet and distributed database technology that auditors in general cannot easily comprehend. Potential audit risks can be raised because the auditors are ignorant of system design and internal control [23]. Therefore, the new challenge auditors are facing is how to choose the best e-auditing technique to reduce the risks generated in the Internet environment [24]. Web services allows the design of automated auditing module to banish manual auditing, which is easily manipulated and hard to monitor, reduce the cost, and raise the reliability of accounting and financial information [25].

In order to track the activities in e-business, it is necessary to employ an audit mechanism that maintains an ongoing data log of all service requests and responses. The data generated in e-business transactions include: logs, audit trails, source documents, communication logs, etc. Transaction logs show who did what, when and where, and the data include user id, passwords, and time and date stamps [26]. In an e-business environment, IT auditors need to review logs and records to perform additional tasks. For example, auditors need to look for failed attempts to access networks, unreasonably long logon sessions and dual logons, and attempts to make unauthorized changes to data. IT auditors need to make sure that trails exist in all e-business applications and that there are adequate internal controls to preserve them [27].

Applying audit logs to auditing had been proposed over a decade. Groomer and Murthy first proposed using database logs to audit data [28]. Recently, many researchers studied data security, user behavior and system efficiency through various audit logs [29-34]. Weber et al. distinguishes two types of audit logs: accounting audit logs and operation audit logs [35], which are described in the following. Accounting audit logs are maintained for system operations, such as data in ERP systems and database transaction logs. They focus on records of values, for example, the quantity is changed to 200 from 100. Operation audit logs record the use of system resources, such as event manager logs. They record the allocation of computer resources and exception events. Since the audit log is necessary for investigating security problems and diagnosing security weaknesses in the SOA, both types of audit logs are implemented in our system.

3. **The Proposed System Architecture.** The case study approach is a powerful research methodology that combines interviews with record analysis and observation. Following the paradigm, in our research, we first propose a comprehensive audit mechanism for SOA/WS applications and then develop a prototype of the collaborative e-procurement system. Finally, the system is verified by the case study to empirically confirm the improvements on accuracy and efficiency.

3.1. **System framework.** Figure 2 illustrates the architecture of our research framework. The framework distinguishes among components used by the end users, the service requestor and the service provider. The operational records about these components can be divided into four parts: host operation, network transmission, requestor operation and provider operation, each of which is described as follows:

(1) Host operation: it records user operations on Webs or applications.
(2) Network transmission: it records the information contents and the time of communication between the user and the service requestor.
(3) Requestor operation: it records the information about the call to the service provider by the service requestor, such as time, source IP, destination IP and service name.
(4) Provider operation: it records the information that the service provider provides, for instance, time, source IP, destination IP and service name.

3.2. **The planned audit log under SOA.** As mentioned previously, two types of audit logs are implemented in our system, which are:

(1) Operation audit logs: it records the usage status of system resources. In the system, the operation audit logs are divided into two types: the usage status of host resources and that of web servers. The first type includes the usage status of system resources, deployment of computer resources and exception events occurred. The second type includes access records of users and service logs of web servers. Note that, to control and monitor the provided services, operation audit logs also record information about the start/end time of the process and its associated applications. The logs are recorded in the event records of information devices provided by the service requestor. When the service requestor calls the service provider for services, the web server begins to record any accessed content.
(2) Accounting audit logs: it records system operations, such as data in ERP systems and database transaction logs. Two types of logs are provided here: web or applications operation logs and transaction logs. The first type of log stores users' activities in the applications, including time, operation, calling method, calling address and thread. The second type of log starts to record web/application operations when a user requests a web service. It stores all operations in every period into the service

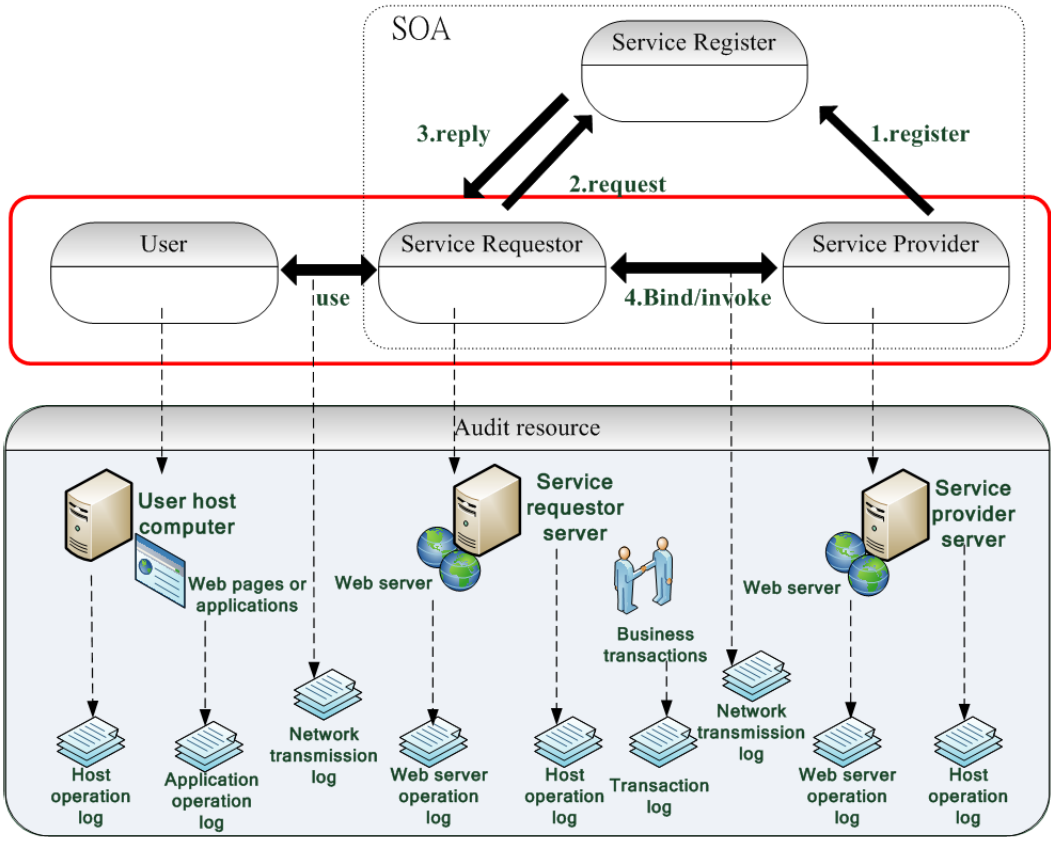


FIGURE 2. Research framework

requestor's devices until the transaction ends. Within web services, control points need to be built into various applications when there is a need to record collaborative information between different parties, such as records of calling services and services replying and transaction results.

The proposed audit mechanism monitors not only transaction logs but also application parameters of importance. In the procurement system, for example, the product name, quantity, access time, user accounts and passwords are required to be constantly monitored to ensure security is maintained as usage grows. For further analysis, the monitoring records are stored on the computers of the service requestor as log files, preventing from data hacking and transaction tampering during the transmission process. Additionally, if any party has an objection to any transaction results, the audit logs can provide proofs of these results so as to protect the rights and interests of the parties involved.

4. **System Implementation.** In our system, the operation audit logs and the accounting audit logs are implemented in the proposed audit mechanism to monitor and verify the e-procurement process. Each of them is summarized in Tables 1 and 2, respectively. Twelve different control points are carefully designed and established to ensure a successful audit on business processes. Figure 3 shows the twelve control points in the sequence diagram of SOA/WS-based applications. Figure 4 illustrates how the control points were deployed to audit the e-procurement process in the system.

The control points designed in the system are introduced as follows:

(1) The user asks the service requestor for information service.
   - Control point 1: the web server's IIS logs record access path and time of the user to the service requestor's website.

Table 1. Description of control points for operation audit logs

	Operation audit logs	
Type	Usage record of host resources	Usage record of Web servers
Control points	Event log	Log in Internet Information Services
Description	Record usage status and deployment of system resources, and start/end time of execution of applications.	Record user access to web resources/services.
Establishing method	Existed (established with process)	Existed (configured in IIS manager)
Information included	Time, source, classification, event, user	Time, IP, computer name, operation, port, browser information
Output	Warning messages written in application programs	IIS Log(*.txt)
Purpose	Monitor running status of processes	Inspect illegal access

Table 2. Description of control points for accounting audit logs

	Accounting audit logs	
Type	Operations of web pages/application programs	Transaction results
Control points	(1) Webpage Trace Log (2) AP Log (3) ASMX Trace	Transaction Result Log
Description	(1) Record user operations on the website (2) Record user operations on application programs (3) Record the contact information between services providers and services requestors when service connections are established.	Record parameters of importance in application programs
Establishing method	(1) Existed (setup in webpage) (2) Developed in .NET	Developed in .NET
Information included	(1) Time, IP, action ID, encoding, cookie, server variables (2) Loading process, address of using function, approach of using function, process threads, timestamp (3) Address of calling services, approach of calling services, calling and replying determination	Transaction sequence, time, user name, product name, quantity, IP
Output	(1) Axd file (2) AP Log(*.txt) (3) ASMX Log(*.txt)	Final Log(*.txt)
Purpose	Find out problems during collaboration	Find out abnormal transaction results

- Control point 2: web trace logs (axd files of trace function in webpages) analyze operations by the user at the services requestor's website.
- Control point 3: local event logs record process related events between the user and the service requestor.
- Control point 4: application operation logs record operations from the user's execution of the service requestor' applications.

(2) The service requestor asks the service provider for connection.

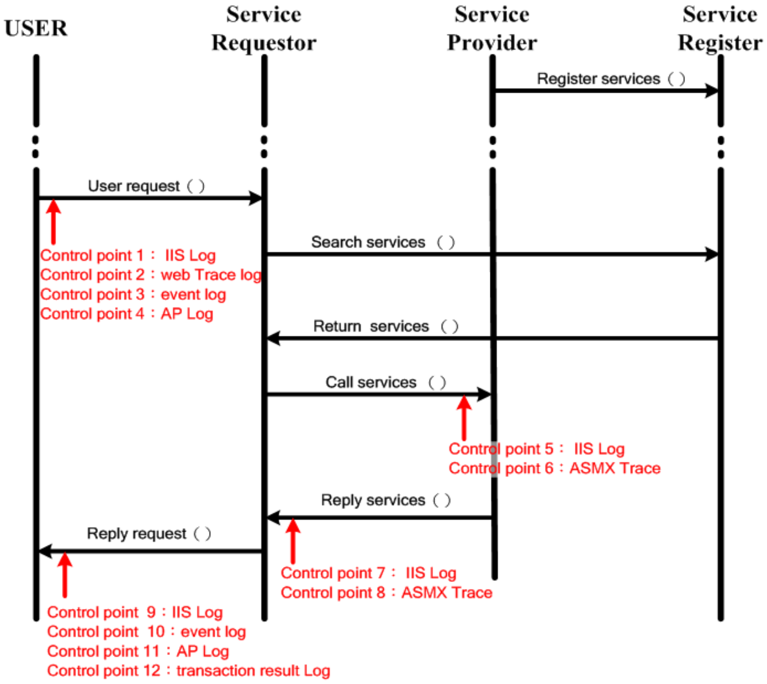

USER
Service Requestor
Service Provider
Service Register
Register services ( )
User request ( )
Control point 1 : IIS Log
Control point 2 : web Trace log
Control point 3 : event log
Control point 4 : AP Log
Search services ( )
Return services ( )
Call services ( )
Control point 5 : IIS Log
Control point 6 : ASMX Trace
Reply services ( )
Reply request ( )
Control point 7 : IIS Log
Control point 8 : ASMX Trace
Control point 9 : IIS Log
Control point 10 : event log
Control point 11 : AP Log
Control point 12 : transaction result Log

FIGURE 3. Audit mechanism in the sequence diagram of SOA

- Control point 5: the web server's IIS logs record resources access from the service requestor to the service provider.
- Control point 6: web service trace logs save calling records into the information facility of the service requestor and the service provider.

(3) The service provider replies to the service requestor's request and transmits information.

- Control point 7: the web server's IIS logs record access from the service provider to the service requestor.
- Control point 8: web service trace logs save responding records into the information facility of the service requestor and the service provider.

(4) The service requestor replies to the user's information request.

- Control point 9: the web server's IIS logs record access path and time of the user to the service requestor's website.
- Control point 10: local event logs record process related events between the user and the service requestor.
- Control point 11: application operation logs record operations from the user's execution of the service requestor's applications.
- Control point 12: monitoring logs of important application fields record transaction results with unencrypted transmission from the user to the service requestor's applications.

5. **System Evaluation.** One case study (detecting abnormal processes) is presented to illustrate the feasibility of our audit mechanism in providing proper control in the SOA environment.

During collaboration among business partners, active calling services and service responses indicate that some transactions are being conducted, but how can the auditor tell if any party is not following the business procedure in the transaction? Figure 5 shows the process of our audit mechanism to solve this problem.

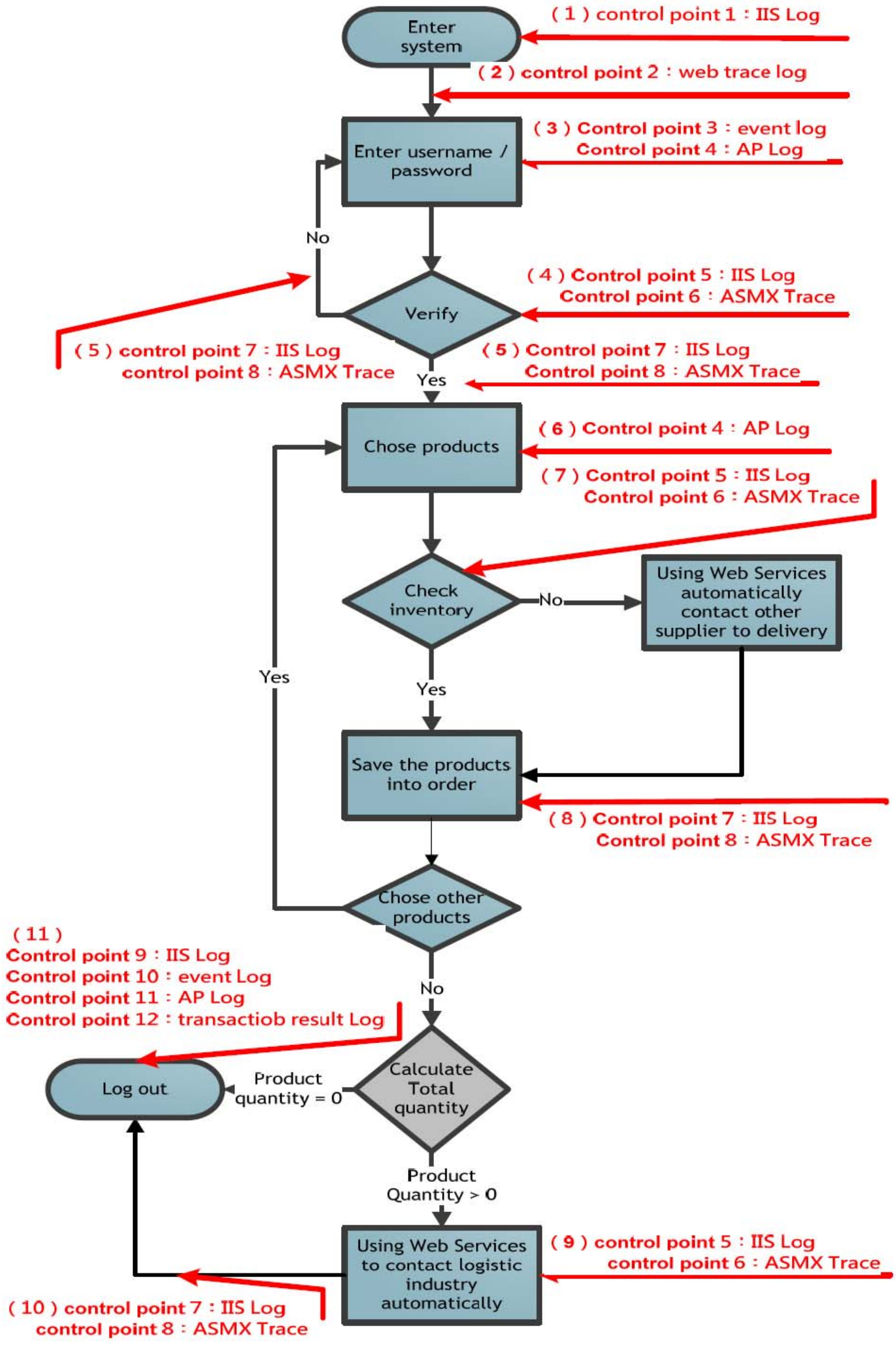

Enter system
(1) control point 1 : IIS Log
(2) control point 2 : web trace log
Enter username / password
(3) Control point 3 : event log
Control point 4 : AP Log
No
Verify
(4) Control point 5 : IIS Log
Control point 6 : ASMX Trace
(5) control point 7 : IIS Log
control point 8 : ASMX Trace
(5) Control point 7 : IIS Log
Control point 8 : ASMX Trace
Yes
Chose products
(6) Control point 4 : AP Log
(7) Control point 5 : IIS Log
Control point 6 : ASMX Trace
Check inventory
No
Using Web Services automatically contact other supplier to delivery
Yes
Yes
Save the products into order
(8) Control point 7 : IIS Log
Control point 8 : ASMX Trace
Chose other products
(11)
Control point 9 : IIS Log
Control point 10 : event Log
Control point 11 : AP Log
Control point 12 : transactiob result Log
No
Log out
Product quantity = 0
Calculate Total quantity
Product Quantity > 0
Using Web Services to contact logistic industry automatically
(9) control point 5 : IIS Log
control point 6 : ASMX Trace
(10) control point 7 : IIS Log
control point 8 : ASMX Trace

FIGURE 4. Control points for e-procurement process

We set up additional control points to detect abnormal processes more quickly. Now, we will focus on Control point 9, 10 and 11.

- Control point 9: it records access from the user to the service requestor. The IIS log will indicate if there were responses from the service requestor to the user. However, Control point 9 can only be used to confirm that there was a transaction but cannot tell whether the transaction followed normal procedures or not. To work on this issue, we have to review the business process involved.

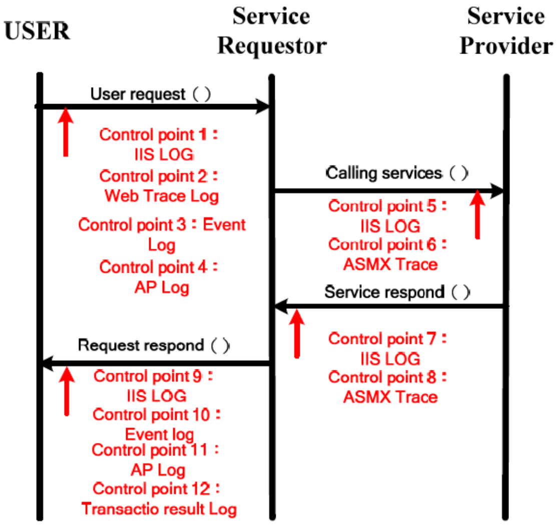
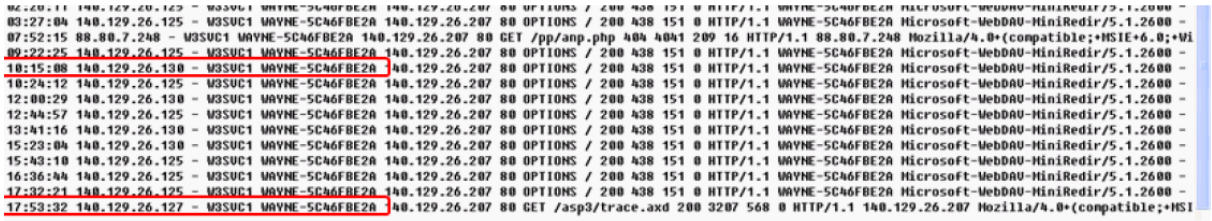

USER
Service Requestor
Service Provider
User request ( )
Control point 1 : IIS LOG
Control point 2 : Web Trace Log
Control point 3 : Event Log
Control point 4 : AP Log
Calling services ( )
Control point 5 : IIS LOG
Control point 6 : ASMX Trace
Service respond ( )
Control point 7 : IIS LOG
Control point 8 : ASMX Trace
Request respond ( )
Control point 9 : IIS LOG
Control point 10 : Event log
Control point 11 : AP Log
Control point 12 : Transactio result Log

Figure 5. Audit mechanism process



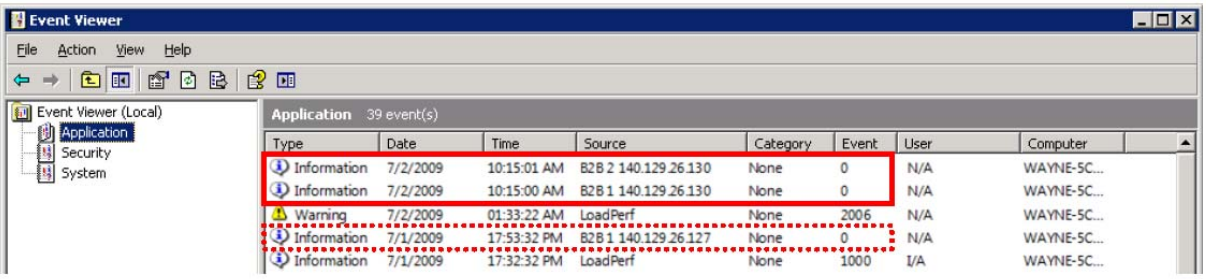
```
02:20:11 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
03:27:04 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
07:52:15 88.80.7.248 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 GET /pp/anp.php 404 4041 209 16 HTTP/1.1 88.80.7.248 Mozilla/4.0+(compatible;+MSIE+6.0;+Wi
09:22:25 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
10:15:08 140.129.26.130 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
10:24:12 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
12:00:29 140.129.26.130 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
12:44:57 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
13:41:16 140.129.26.130 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
15:23:04 140.129.26.130 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
15:43:10 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
16:36:44 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
17:32:21 140.129.26.125 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 OPTIONS / 200 438 151 0 HTTP/1.1 WAYNE-5C46FBE2A Microsoft-WebDAV-MiniRedir/5.1.2600 -
17:53:32 140.129.26.127 - W3SVC1 WAYNE-5C46FBE2A 140.129.26.207 80 GET /asp3/trace.axd 200 3207 568 0 HTTP/1.1 140.129.26.207 Mozilla/4.0+(compatible;+MSI
```

Figure 6. IIS log in the case

As shown in Figure 6, the marked circles show access results from IP-140.129.26.127 and IP-140.129.26.130 to the service requestor. It can be seen that the service requestor did reply to the user's requests, but it cannot be verified whether normal procedure is being followed.

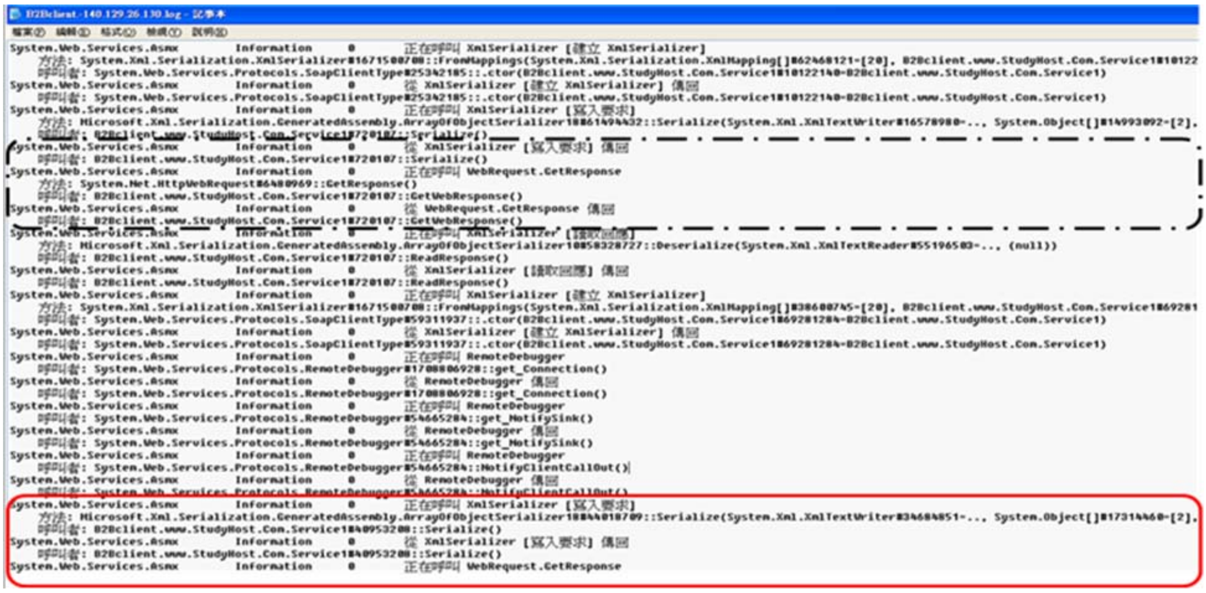
- Control point 10: it is an alert message generated by the system. When a process is finished, the system will generate a record in the event log of the service requestor. For example, an event log is generated after the supplier confirms the acceptance of its order to the purchaser. The log includes the time, the user, event serial number, and so on. When the shipping address and the date of delivery are confirmed by the shipping company, another log is generated subsequently. We can determine whether every step of the transaction is consistent with the business process; in other words, we can verify the accuracy and legitimacy of the transaction through control point 10.

As shown in Figure 7, the upper circle shows the operation logs from IP-140.129.26.130. The second log (B2B_1) was generated from order confirmation and the first log (B2B_2) was generated from delivery. The lower circle shows the log from IP-140.129.26.127. The log for order confirmation was existed but that for delivery was missing, indicating that an abnormal process has occurred.



Type	Date	Time	Source	Category	Event	User	Computer
Information	7/2/2009	10:15:01 AM	B2B 2 140.129.26.130	None	0	N/A	WAYNE-5C...
Information	7/2/2009	10:15:00 AM	B2B 1 140.129.26.130	None	0	N/A	WAYNE-5C...
Warning	7/2/2009	01:33:22 AM	LoadPerf	None	2006	N/A	WAYNE-5C...
Information	7/1/2009	17:53:32 PM	B2B 1 140.129.26.127	None	0	N/A	WAYNE-5C...
Information	7/1/2009	17:32:32 PM	LoadPerf	None	1000	N/A	WAYNE-5C...

FIGURE 7. Event log of abnormal process in case 3


B2Bclient-140.129.26.130.log - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [建立 XmlSerializer]
    方法: System.Xml.Serialization.XmlSerializer#16715007#0::FromMappings(System.Xml.Serialization.XmlMapping[]#62468121-[20], B2Bclient.www.StudyHost.Com.Service1#10122
    呼叫者: System.Web.Services.Protocols.SoapClientType#25342185::.ctor(B2Bclient.www.StudyHost.Com.Service1#10122140-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       從 XmlSerializer [建立 XmlSerializer] 傳回
    呼叫者: System.Web.Services.Protocols.SoapClientType#25342185::.ctor(B2Bclient.www.StudyHost.Com.Service1#10122140-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [寫入要求]
    方法: Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer10#61494432::Serialize(System.Xml.XmlTextWriter#16578980-..., System.Object[]#14993092-[2],
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::Serialize()
System.Web.Services.Asmx        Information     0       從 XmlSerializer [寫入要求] 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::Serialize()
System.Web.Services.Asmx        Information     0       正在呼叫 WebRequest.GetResponse
    方法: System.Net.HttpWebRequest#6480969::GetResponse()
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::GetWebResponse()
System.Web.Services.Asmx        Information     0       從 WebRequest.GetResponse 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::GetWebResponse()
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [讀取回應]
    方法: Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer10#58328727::Deserialize(System.Xml.XmlTextReader#55196583-..., (null))
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::ReadResponse()
System.Web.Services.Asmx        Information     0       從 XmlSerializer [讀取回應] 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::ReadResponse()
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [建立 XmlSerializer]
    方法: System.Xml.Serialization.XmlSerializer#16715007#0::FromMappings(System.Xml.Serialization.XmlMapping[]#38600745-[20], B2Bclient.www.StudyHost.Com.Service1#69281
    呼叫者: System.Web.Services.Protocols.SoapClientType#59311937::.ctor(B2Bclient.www.StudyHost.Com.Service1#6928128#-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       從 XmlSerializer [建立 XmlSerializer] 傳回
    呼叫者: System.Web.Services.Protocols.SoapClientType#59311937::.ctor(B2Bclient.www.StudyHost.Com.Service1#6928128#-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       正在呼叫 RemoteDebugger
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#17008069#8::get_Connection()
System.Web.Services.Asmx        Information     0       從 RemoteDebugger 傳回
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#17008069#8::get_Connection()
System.Web.Services.Asmx        Information     0       正在呼叫 RemoteDebugger
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#54665284::get_NotifySink()
System.Web.Services.Asmx        Information     0       從 RemoteDebugger 傳回
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#54665284::get_NotifySink()
System.Web.Services.Asmx        Information     0       正在呼叫 RemoteDebugger
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#54665284::NotifyClientCallOut()
System.Web.Services.Asmx        Information     0       從 RemoteDebugger 傳回
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#54665284::NotifyClientCallOut()
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [寫入要求]
    方法: Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer10#44018709::Serialize(System.Xml.XmlTextWriter#34684851-..., System.Object[]#17314460-[2],
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#40953200::Serialize()
System.Web.Services.Asmx        Information     0       從 XmlSerializer [寫入要求] 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#40953200::Serialize()
System.Web.Services.Asmx        Information     0       正在呼叫 WebRequest.GetResponse
```

FIGURE 8. Manufacturers A (140.129.26.130) AP log

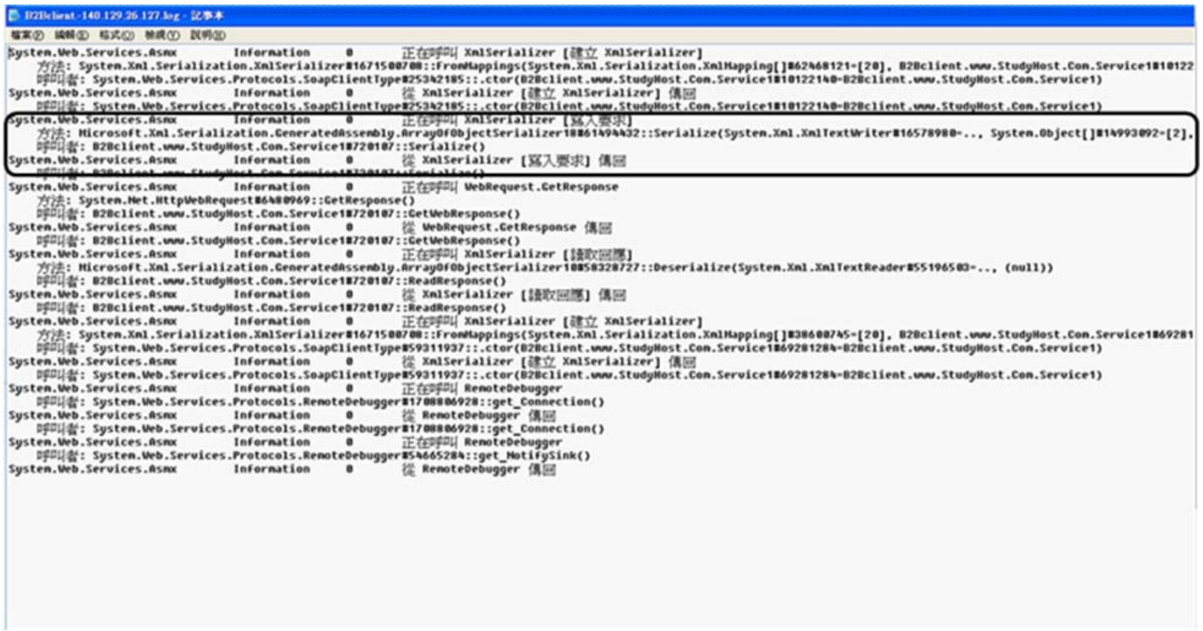
- Control point 11: it records user operations in the AP logs of the active applications of the service provider. The log file will be maintained by the service requestor, and it includes the user name, timestamp and thread. Auditors can quickly find out when the problems occurred through cross-comparison; furthermore, they can trace back the control point logs to ascertain the time when and at which side the problems occurred.

As shown in Figure 8, manufacture A's IP is 140.129.26.130. The upper circle shows the calling/response records when the user confirms the order, and the lower one indicates that the system is ready to transmit the necessary information to the shipping company. As shown in Figure 9, manufacture B's IP is 140.129.26.127. It is obvious that the user did not input shipping information, making the system unable to connect with the shipping company. This finding is consistent with what is found in the log of control point 10. Thus, both control points allow effective detection of abnormal processes.

6. Conclusion.

Service-oriented architectures eases the integration of the heterogeneous IT environments found in many organizations through the use of standard protocols, such as web services. It can support an infinite variety of business processes, but simultaneously challenges the way that IT organizations manage, audit and secure applications and data.

In this paper, a comprehensive audit mechanism is proposed to provide a cross-platform solution in a heterogeneous software/hardware environment. Our objective is to monitor


B2Bclient-140.129.26.127.log - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [建立 XmlSerializer]
    方法: System.Xml.Serialization.XmlSerializer#16715008::FromMappings(System.Xml.Serialization.XmlMapping[]#62468121-[20], B2Bclient.www.StudyHost.Com.Service1#10122
    呼叫者: System.Web.Services.Protocols.SoapClientType#25342185::.ctor(B2Bclient.www.StudyHost.Com.Service1#10122140-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       從 XmlSerializer [建立 XmlSerializer] 傳回
    呼叫者: System.Web.Services.Protocols.SoapClientType#25342185::.ctor(B2Bclient.www.StudyHost.Com.Service1#10122140-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [寫入要求]
    方法: Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer1#61494432::Serialize(System.Xml.XmlTextWriter#16578980-..., System.Object[]#14993092-[2],
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::Serialize()
System.Web.Services.Asmx        Information     0       從 XmlSerializer [寫入要求] 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::Serialize()
System.Web.Services.Asmx        Information     0       正在呼叫 WebRequest.GetResponse
    方法: System.Net.HttpWebRequest#6480969::GetResponse()
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::GetWebResponse()
System.Web.Services.Asmx        Information     0       從 WebRequest.GetResponse 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::GetWebResponse()
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [讀取回應]
    方法: Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer10#58328727::Deserialize(System.Xml.XmlTextReader#55196503-..., (null))
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::ReadResponse()
System.Web.Services.Asmx        Information     0       從 XmlSerializer [讀取回應] 傳回
    呼叫者: B2Bclient.www.StudyHost.Com.Service1#720107::ReadResponse()
System.Web.Services.Asmx        Information     0       正在呼叫 XmlSerializer [建立 XmlSerializer]
    方法: System.Xml.Serialization.XmlSerializer#16715008::FromMappings(System.Xml.Serialization.XmlMapping[]#38600745-[20], B2Bclient.www.StudyHost.Com.Service1#69281
    呼叫者: System.Web.Services.Protocols.SoapClientType#59311937::.ctor(B2Bclient.www.StudyHost.Com.Service1#6928128-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       從 XmlSerializer [建立 XmlSerializer] 傳回
    呼叫者: System.Web.Services.Protocols.SoapClientType#59311937::.ctor(B2Bclient.www.StudyHost.Com.Service1#6928128-B2Bclient.www.StudyHost.Com.Service1)
System.Web.Services.Asmx        Information     0       正在呼叫 RemoteDebugger
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#17088069::get_Connection()
System.Web.Services.Asmx        Information     0       從 RemoteDebugger 傳回
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#17088069::get_Connection()
System.Web.Services.Asmx        Information     0       正在呼叫 RemoteDebugger
    呼叫者: System.Web.Services.Protocols.RemoteDebugger#5466528::get_NotifySink()
System.Web.Services.Asmx        Information     0       從 RemoteDebugger 傳回
```

Figure 9. Manufacturers B (140.129.26.127) AP log

and safeguard business activities in the SOA/WS environment so as to avoid interruptions of business processes that may lead to losses of the enterprise, its partners and customers. In the prototype system, the study case shows that the audit mechanism aggregates logs generated by existed tools to audit business processes and transaction results under different environments. It has three distinct benefits over other solutions:

(1) It provides a more efficient and effective audit. Traditional auditing methods cannot detect process errors and forged results in the web services environment. Our audit mechanism provides enterprises with digital evidence of internal control, in addition to finding out abnormal and illegal events in the business process.
(2) It saves time and cost in deployment. Our audit mechanism uses some available logs in the system. Therefore, implementing our audit mechanism does not require additional investments, and it does not have any negative effects on the audited system as well.
(3) It reduces errors in message transmission. Because the control points in our audit mechanism are interrelated, the probability of producing erroneous digital evidence is relatively low. Therefore, auditors will be less likely to make wrong judgments on the audit logs so that misunderstandings among a variety of departments or enterprises will be reduced correspondingly.

Acknowledgment. This work is partially supported by the Science Council of Taiwan (NSC 100-2221-E-036-036) and Tatung University (B99-N05-067).

REFERENCES

[1] P. P. Michael, *Web Services Principles and Technology*, Addison Wesley Pub., 2008.
[2] R. Uleman, Service oriented architecture unveiled, *Geospatial Solutions*, pp.30-33, 2006.
[3] L. Walker, IBM business transformation enabled by service-oriented architecture, *IBM Systems Journal*, vol.46, no.4, pp.651-667, 2007.
[4] M. R. Brenner and M. R. Unmehhopa, Service-oriented architecture and web services penetration in next-generation networks, *Bell Labs Technical Journal*, vol.12, no.2, pp.147-160, 2007.
[5] F. Leymann, D. Roller and M. T. Schmidt, Web services and business process management, *IBM Systems Journal*, vol.41, no.2, pp.198-211, 2002.
[6] Y. V. Natis, *Service-oriented Architecture Scenario*, Gartner, 2003.

[7] H. Deitel, P. Deitel, B. Duwaldt and L. Trees, *Web Services a Technical Introduction*, Prentice Hall, 2003.

[8] L. Wang, X. Ming, J. You, F. Kong and D. Li, A new design methodology for metadata of master data management system, *International Journal of Innovative Computer, Information and Control*, vol.6, no.7, pp.3219-3228, 2010.

[9] F. Valles-Barajas and W. Schaufelberger, A proposal for the software design of control systems based on the personal software process, *International Journal of Innovative Computer, Information and Control*, vol.6, no.8, pp.3451-3466, 2010.

[10] U. S. Murthy and S. M. Groomer, A continuous auditing web services model for XML based accounting systems, *International Journal of Accounting Information Systems*, vol.5, no.2, pp.139-163, 2004.

[11] S. H. Li, Using generalized audit software to assist computing auditing in ERP system, *International Conference on Accounting and Information Technology*, 2005.

[12] H. H. Chen, S. H. Li, S. M. Huang and Y. C. Hung, The development and performance evaluation of a continuous auditing assistance system, *International Journal of Electronic Finance*, vol.1, no.4, pp.460-472, 2007.

[13] S. H. Li, S. M. Huang and Y. C. Lin, Developing a continuous auditing assistance system based on information process models, *Journal of Computer Information Systems*, pp.2-13, 2007.

[14] S. H. Li and D. Yen, Using database log assist auditors to do compliance test, *International Conference on Accounting and Information Technology*, 2007.

[15] L. L. Thomas, The role of organizational factors in realizing ERP benefits, *Information Systems Management*, vol.19, no.4, pp.21-42, 2002.

[16] M. Weske, *Business Process Management*, Springer Pub., 2007.

[17] M. Kim and D. Kim, Fault-tolerant process debugger for business process design, *International Journal of Innovative Computer, Information and Control*, vol.6, no.4, pp.1679-1687, 2010.

[18] D. Kim, N. Lee, S. Kang, M. Cho and M. Kim, Business process version management based on process change patterns, *International Journal of Innovative Computer, Information and Control*, vol.6, no.2, pp.567-575, 2010.

[19] M. Dumas, W. M. P. Aalst and A. H. M. Hofstede, *Process-aware Information Systems*, Wiley Pub., 2005.

[20] E. Newcomer and G. Lomow, *Understanding SOA with Web Services*, Pearson Education, Inc., 2007.

[21] K. G. Anup, *E-Commerce Security: Weak Links, Best Defenses*, Wiley, 1998.

[22] D. Blake, *XML Security*, McGraw-Hill, 2002.

[23] B. Mark, B. Christoph, Z. Michal, F. Tim, N. H. Michael, P. Amit and W. Stuart, A semantic web service architecture, *IEEE Internet Computing*, vol.9, no.5, pp.72-81, 2005.

[24] A. H. James and S. Tommie, *Information Technology Auditing and Assurance*, South-Western Pub., 2006.

[25] E. Koskivaara, Artificial neural network models for predicting patterns in auditing monthly balances, *The Journal of the Operational Research Society*, vol.51, no.9, pp.1060-1069, 2000.

[26] Z. Gu, D. Wang and D. Yue, Fault detection for continuous-time networked control system with non-ideal QoS, *International Journal of Innovative Computer, Information and Control*, vol.6, no.8, pp.3631-3640, 2010.

[27] J. E. Hunton, S. M. Bryant and N. A. Bagranoff, *Core Concepts of Information Technology Auditing*, Wiley Pub., 2003.

[28] S. M. Groomer and U. S. Murthy, Continuous auditing of database applications: An embedded audit module approach, *Journal of Information Systems*, vol.3, no.1, pp.53-69, 1989.

[29] D. Thomas and S. Eugene, Identification of host audit data to detect attacks on low-level IP vulnerabilities, *Journal of Computer Security*, vol.7, no.1, pp.3-35, 1999.

[30] S. Bruce and K. John, Secure audit logs to support computer forensics, *ACM Transactions on Information and System Security*, vol.1, no.3, pp.159-176, 1999.

[31] L. V. Orman, Database audit and control strategies, *Information Technology and Management*, vol.2, no.1, pp.27-51, 2001.

[32] R. T. Mercuri, On auditing audit trails, *Communications of the ACM*, vol.46, no.1, pp.17-20, 2003.

[33] G. H. Bodnar, SQL: An internal audit technology, *Internal Auditing*, vol.19, no.1, pp.34-38, 2004.

[34] W. M. P. Aalst, W. Ton and M. Laura, Workflow mining discovering process models from event logs, *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.9, pp.1128-1142, 2004.

[35] G. L. Weber, J. P. Michael and B. Daniel, Machine learning for user modeling, *User Modeling and User-adapted Interaction*, vol.11, no.1-2, pp.19-29, 2001.