A Service Oriented Execution Model of Financial Online Transactions*

Hung-Ta Lu and Chua-Huang Huang
Department of Information Engineering and Computer Science
Feng Chia University
Taichung, Taiwan
{albert, chh}@pmlab.iecs.fcu.edu.tw

Abstract

A business system should provide appropriate services fast enough in order to satisfy customer's demand. It is important to integrate the resources inside an enterprise so that the development of transaction processes can be very flexibly. Service oriented architecture (SOA) is a concept which provides a methodology to deal with communications between heterogeneous distributed systems. SOA points out a new direction of the solution in system integration of existing legacy systems. The business process execution language (BPEL) can accomplish the business processes management based on SOA. The purpose of our research is to use the SOA technologies and BPEL to simplify the design of business transaction processes, especially, transaction processes of financial business. A software method to describe transaction processes of financial operations is proposed. Based on SOA and BPEL, we design a system that integrates heterogeneous platforms in a financial institution.

Keywords: web service, business process execution language (BPEL), service oriented architecture (SOA), software methodology, transaction process.

1: Introduction

The investment of financial information systems is quite large in a financial institution. Because of the change of laws, requirements, financial products, and information technologies of the financial market, a financial institution will accumulate and extend a variety of information systems gradually. Under this development model of financial information systems, most of the financial institutions have more than one business platform to deal with different kinds of business requirements such as core banking system, foreign exchange system, general fund purchase system, and so on. Therefore, some financial institutions adopt cross-platform integration architecture such as the enterprise application integration (EAI) architecture to manipulate cross-platform financial online transaction processes [7, 8]. However, the cost of establishing this kind of system is very expensive and the change of a transaction process must be predefined and is lack of flexibility. In this paper, we present a service oriented execution model of financial online transactions to improve the way of developing and executing cross-platform transaction processes. We use the web service technologies and BPEL to achieve the concept of service oriented architecture (SOA) in financial transaction processes.

Web service is designed to support the interoperability of software systems. It is build up with several standard technologies on the Internet and extensible markup language (XML) [2, 5]. The core standards of web service technologies are simple object access protocol (SOAP), web service description language (WSDL) and universal description, discovery and integration (UDDI) [1]. SOAP defines the XML message protocol to provide the basic service interoperability [9]. WSDL is the common grammar of describing the web services property [3]. The registry mechanism of UDDI provides the architecture of systematic publish and discovery web services [6]. Web service has the advantages of platform independent, interoperability and loosely couple in conceptions of technology.

Business Process Execution Language for Web Services (BPEL4WS) is a new technology specification presented by leading vendors of web service technologies such as IBM, Microsoft and BEA in August 2002 [4]. BPE4WS has combined web service flow language (WSFL) of IBM and XLANG of Microsoft. It provides a notation and semantic rules to describe the behaviors of business processes based on XML and web service structure. The business processes modeling and process engine are describing and executing in more abstract way to meet the goal of cross-platform application integration and BPEL is one of the most important description languages in business process management [7].

The organization of this paper is as the following. In Section 2, we will introduce some properties of financial transactions and in this paper we will focus on the financial online transactions. For designing a

*This work was supported in part by National Science Council, Taiwan, R.O.C. under grant NSC 94-2213-E-035-039.

service oriented financial online transaction model, we will analyze the behavior of financial online transactions in Section 3. We propose a service oriented financial online transaction model and design an execution platform in Section 4. Section 5 shows two main message formats we used in financial processes integration platform. Conclusion and future work are given in Section 6.

2: Financial Transactions

In financial institutions, financial transactions are usually deployed on heterogeneous business systems and have various types for different purposes. In this section we will introduce the financial transactions briefly including how we assort and manage these financial transactions.

2.1: The Classification of Financial Transactions

Batch operation. A batch operation is usually used to deal with a great quantity of data. It must be predefined and need not to be changed immediately. This kind of batch operation is started by a system operator. It usually uses to count and analyze the trade data. A batch operation also has fixed execution purpose, procedure, period and low timeliness.

Online batch transaction. Online batch operation is started by a teller through a single online transaction. For example, printing the balance statement on customer's demand. After starting this batch operation through an online transaction, it triggers a batch operation. Its execution is on the background mode of the business system because it must manipulates many data and needs long execution time. Hence, an online batch operation is an asynchronous transaction mode.

Single online transaction. Single online transaction is a synchronous transaction model. Teller sends the request message and receives the response message immediately. Generally, the time difference of receiving the response message will not exceed a short period of time, e.g., one minute. We will take a single online transaction to be an atomic service (component service) in the service oriented execution model.

Online transaction process. Online transaction process is a kind of transaction execution process which connects a series of single online transactions according to the requirements on the banking business. Single online transactions may come from different business platform, so that an online transaction process should have the ability to cross various business platforms. In this part, many financial institutions handle the online transaction process by tellers manually or adopt expensive integration systems. An online transaction process is a composite service which consists of component services.

2.2: The Management of Financial Online Transactions

Batch operation. A batch operation has fixed execution period and specific purpose. So the name of a batch operation includes the information of execution type, operation type, business type and execution period.

Example: RBZM0010
R: execution type (random execution)
B: operation type (batch operation)
Z: business type (common type)
M: execution period (monthly)
0010: sequence number

Online transaction. An online transaction uses transaction code to represent an online transaction and manage transactions through a hierarchy code structure. A bank teller can use the transaction code to operate a financial online transaction through proprietary terminals. In our research, we also use the transaction code to identify transaction services.

Example: 02071
0: platform type (core banking system)
20: business type (savings business)
7: transaction type (inquiry transaction)
1: sequence number

Platform type represents which business platform the online transaction belongs to. Business type also tells us which kind of business the online transaction is, such as customer management or demand deposit. Transaction type identifies the feature of an online transaction such as several inquiry transactions can execute concurrently. Transaction type also decides the priority of transaction execution, such as the priority of deposit transaction is higher than withdraw transaction. The priority of online transactions will affect the composition of an online transaction process. Through appropriate online transaction management, we can use online transactions systematically and efficiently. We can also generate correct online transaction process automatically.

3: Behaviors of Financial Online Process

In a financial information system, a transaction includes main business manipulation logic, access of database, exception handling process, etc. The execution of a single online transaction is a synchronous progress. When a teller sends the request message from a terminal, and then receives the response message immediately. The transaction behavior model is simple, similar to the client-server architecture. We assume that a single online transaction will be encapsulated as a web service, called a component service. Component services can be the basis of a composite service. An online

transaction process is composed by several single online transactions.

The way of executing an online transaction process is divided into concurrency execution and sequential execution:

Concurrency execution. In financial business system, an online transaction not involving accounting account can be executed concurrently. For example, a balance inquiry transaction of each business platform has not necessary to concern about the interaction between other transactions when composing an online transaction process. It only need following the execution sequence assigned by teller.

Sequential execution. We must consider the execution priority between online transactions while executing an online transaction related to accounting. For example, a teller assigns the opening time deposit transaction first and then assigns the general fund redemption transaction. In fact, the redemption transaction will be executed before the opening account transaction because the priority of a deposit transaction is higher than a withdraw transaction. This policy can avoids the shortage of balance when doing the withdraw transaction.

3.1: Typical Financial Transaction Processes

In a small financial institution, it usually has no cross-platform infrastructure. Transaction executed on each business system is independent. They do not have a common transaction interface and most of the business systems have different communication protocols and information specifications. The interaction rules among business systems must be defined in each business system. If there is a new transaction process which must be across different business systems, it has to update the program code or the system configuration. It is time wasting and cost expensive. The change of systems also increases transaction risk. There almost has no concept of transaction process among these business systems. So many transaction processes are executed by teller manually. The upper part of Figure 1 describes this kind of transaction process model.

A large financial institution will construct a system structure (such as EAI) to integrate different business systems. EAI is a middleware which plays a role as a software router. It predefines message exchanging rules, message transmitting direction, transaction processes among business systems. A teller can finished a transaction process through a proprietary terminal of each business system or a common interface provided by the integration system. The lower part of Figure 1 describes this kind of transaction process model. But this kind of integrated system must predefine the connected business systems and transaction processes. It belongs to a passive integration way such that the system is lack of flexibility, high in its development cost, and difficult for maintenance.

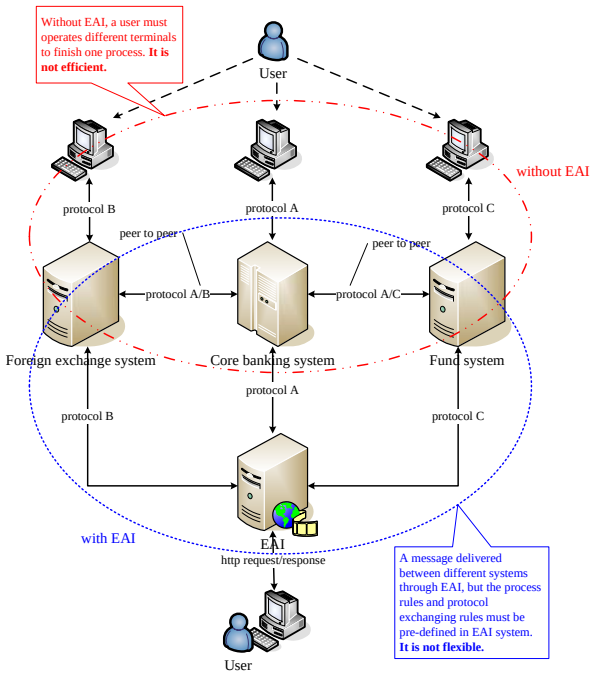

Figure 1: Typical financial transaction processes

3.2: Service Oriented Financial Transaction Processes

According to the maturity of web service technologies, the concept of SOA is being proposed. In order to meet the requirements of integration and services in financial industry, we use the technologies of web service to propose a concept of service oriented financial transaction processes. Service oriented financial transaction processes can improve the weakness of typical financial transaction processes and build an execution model that can change the transaction processes to accomplish real business system integration flexibly.

As shown in Figure 2, we present a Financial Processes Integration Platform (FPIP). FPIP includes several models such as UDDI registry, BPEL process engine, BPEL document generator, etc. We define a set of common message interfaces and encapsulate the financial transactions on each business system in web services. WSDL documents of these web services are published to the UDDI registry in FPIP. A teller can use the operation interface provided by FPIP to select a service and generate the BPEL document of the transaction process. The BPEL process engine executes the transaction process in compliance with the BPEL document and invokes the web services on each business system which is defined in the BPEL document. Finally, FPIP will summarize the execution results and display the outcome to the teller on a display monitor.

The service oriented financial transaction process can simplify the teller operations by using a common operation interface, utilizing the flexibility of BPEL to

construct a financial transaction process, and then completing the execution of the transaction process through the web service operation mechanism. The scope of integration includes the business systems of subsidiary companies in enterprise. The SOA structure also gives FPIP extension capability to combine with external business systems.

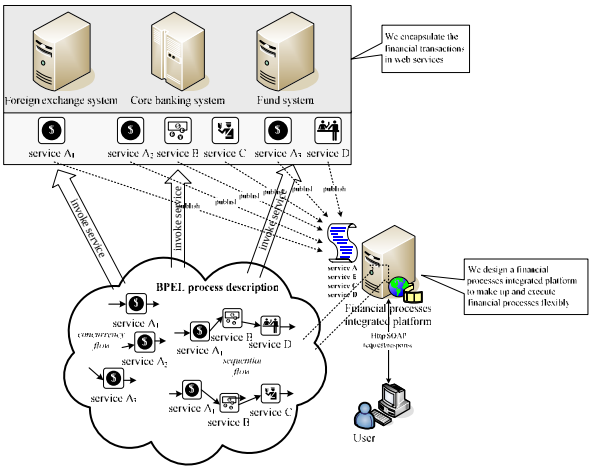

Figure 2: Service oriented financial transaction processes

4: Financial Processes Integration Platform

The structure of FPIP is based on the open sources on the Internet. The main purpose of our research is to build a transaction execution platform that can adjust the transaction processes between different business systems flexibly so that tellers can finish a cross-platform transaction process via a simple operation interface to select transaction services. The system structure and design model of FPIP is shown in Figure 3, consisting of three core modules:

Process composing module. According to the services selected by tellers, the process composing module will determine the execution sequence and translate it into the standardized BPEL document, and then deploy the BPEL document on process engine to be executed.

Process execution module. The process execution module includes a process engine in charge of executing BPEL document and replying the executed results to end users.

Service definition module. The transaction services provided by business systems are published to the UDDI registry through service definition module and referenced by the process composing module to generate BPEL documents.

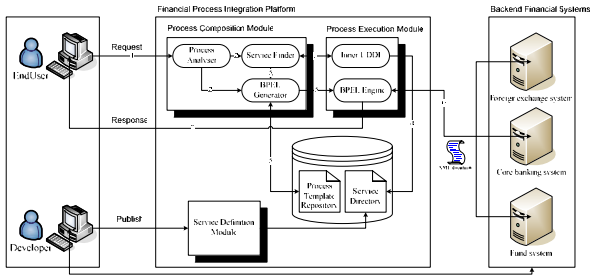

Figure 3: Structure of FPIP

4.1: Process Composing Module

Process composing module is in charge of generating BPEL documents and packing the BPEL documents in a file so the process engine can identify, deploy, and execute defined business processes. Process composing module is made up of three parts as the followings:

Process analyzer module. When a teller operates the user interface via a browser and the transaction services and the execution sequence of process are decided, the process analyzer module interprets the request message into the transaction service part and the flow control part. The selected transaction services are delivered to the service finder module and the message of process control is delivered to BPEL document generator module.

Service finder module. As soon as the service finder module receives a transaction message from process analyzer module, it will search for the WSDL document of each service from the inner UDDI registry sequentially. After the inquiry transaction terminates, it sends the result to BPEL document generator module.

BPEL document generator module. BPEL document generator generates a BPEL document following the BPEL specifications and the files used for deployment after it receives the process control message and WSDL document. Once the BPEL document generator module finishes the encapsulation of deployment file, the deployment file is forwarded to the process execution module. The way to encapsulate the deployment file depends on the type of BPEL process engine.

Process composing module is the most important module in FPIP. It provides a mechanism to dynamically select services, compose transaction processes and make the process execution more flexible.

4.2: Process Execution Module

The process execution module is the main execution environment of financial transaction processes and is in charge replying the result to tellers. There are three submodules in process execution

module. It uses application server as the main execution platform. The functionalities of these submodules are stated as the followings:

Inner UDDI registry. Inner UDDI registry is in charge of service registration. Every financial business system can publish their services in the inner UDDI registry from where the service finder module can find the transaction services automatically. This module can be replaced by other simple mechanism such as assigning the location of transaction service directly. The financial transaction service provider may create a locator just to tell clients about the location of the financial transaction service.

BPEL process engine. A process engine is in charge of executing financial transaction processes because BPEL document is an abstract description document of the financial transaction process based on the XML standard. FPIP adopts an open source BPEL process engine to build the execution environment. We can replace this submodule by another product.

Process template repository. The process template repository is an auxiliary submodule for storing the financial transaction process definitions. When service finder module find a process template in process template repository as same as the teller's request and then sends the process template to BPEL document generator directly. It needs not inquire the information of the financial transaction service from inner UDDI registry. Hence, this submodule can speed up the process execution efficiency and it is optional.

4.3: Service Definition Module

Service definition module provides a financial transaction service deployment interface. A developer can use this interface to manage the financial transaction services via a web browser directly. We can also use other development tools to meet this purpose.

5: Message Formats

In FPIP, we use two kinds of message format to construct a financial process document. The messages are specified using the XML standard. We use document object model (DOM) and simple API for XML (SAX) to control and manipulate all the XML documents. This section will explain these two message format structures.

5.1: Simple Process Document

Simple process format describes the execution sequence of selected transaction services. The message body includes the financial transaction services which will be executed. The root element is <transactions>. It is the container of <transaction> elements. The value of the name attribute of <transactions> element consists of the name of each financial transaction service. Each <transaction> element represents a financial transaction service. A <transaction> element includes two attributes, name and execution priority, and a child element <wsdl> of the financial transaction service. The name attribute of <transaction> represents the name of a financial transaction. It consists of host ID, business type and transaction type. The priority attribute represents the execution priority. We can use the priority attribute to adjust the execution sequence in a financial transaction process. The <wsdl> element contains the location information of the WSDL document of a financial transaction service. According to this specification, we can get the WSDL document of each financial transaction service and then construct the BPEL document of the financial transaction process follow these WSDL documents. The example of a simple process format is as the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<transactions flow="c" name="0207013070">
 <transaction name="02070" priority="0">
  <wsdl>
   http://localhost:8080/TestWS/services/
   BalanceQueryImpl?WSDL
  </wsdl>
 </transaction>
 <transaction name="13070" priority="0">
  <wsdl>
   http://localhost:8080/t13070/services/
   T13070?WSDL
  </wsdl>
 </transaction>
 ...
</transactions>
```

5.2: Transaction Process Document

The transaction process document is a BPEL document following the BPEL4WS standard. We use this BPEL document to execute the financial transaction process constructed by the selected financial transaction services. Each transaction service in a transaction process is encapsulated by a <sequence> activity. The financial process document maps to the simple process document and follows the execution sequence defined in simple process format. The example of a transaction process document is as the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<process name="p0207013070" ...>
 ...
 <sequence name="main">
  ...
  <scope name="MainScope">
   <flow>
    <sequence>
```

```
      <assign name="assign1_02070">
       <copy>
        <from variable="input_00000"/>
        <to variable="input_02070"/>
       </copy>
      </assign>
      <invoke inputVariable="input_02070"
       operation="getBalance"
   outputVariable="output_02070"
   partnerLink="PL_02070"
   portType="BalanceQueryImpl"/>
    </sequence>
...
   </flow>
   ...
  </scope>
 ...
 </sequence>
</process>
```

6: Conclusions and Future Work

SOA has the properties of interoperability, location transparency, loosely coupled and reusable. These characteristics make designing a business process more flexible.

The typical financial business platform and online transactions are tightly coupled and financial online processes are executed by tellers manually. We use the concept of SOA to simplify the design and automate execution of financial online processes. But how to integrate and manage the complex online transaction interface such that this execution model can apply to wide transaction range remain important issues.

Web services is a way to practice the SOA concept at present. To find a better way that can accomplish the SOA concept in specific domain, like financial industry, is our research goal in the future.

References

[1] A. Bonifati, S. Ceri, and S. Paraboschi. Active rules for XML: A new paradigm for E-services. *The VLDB Journal*, pages 39–47, 2001.

[2] M. Brandner, M. Craes, F. Oellermann, and O. Zimmermann. Web services-oriented architecture in production in the finance industry. *Informatik-Spektrum*, 02:136–145, 2004.

[3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (WSDL) 1.1. Technical report, W3C Web Site, 2001.

[4] C. Emig, J. Weisser, and S. Abeck. Development of soa-based software systems - an evolutionary programming approach. In *Special interest tracks and posters of the 14th International Conference on World Wide Web*, pages 1132–1133, 2006.

[5] A. Lazovik, M. Aiello, and M. Papazoglou. Associating assertions with business processes and monitoring their execution. In *Proceedings of the 2nd international conference on Service oriented computing*, pages 94–104, 2004.

[6] L.Clement, A. Hately, C. v. Riegen, and T. Rogers. UDDI version 3.0.2. Technical report, OASIS, 2004.

[7] R. Sharma, B. Stearns, and T. Ng. *J2EE Connector Architecture and Enterprise Application Integration*. Addison Wesley Longman, Inc., 2002.

[8] J. Sutherland and W. J. Heuvel. Enterprise application integration and complex adaptive systems. *Communications of the ACM*, 45(10):59–64, 2002.

[9] W3C. SOAP version 1.2. Technical report, W3C Web Site, 2003.