Traditional core-banking system not only has high development cost but also acts like an information island and lacks a more flexible business model where its subsystems are independently established and hard to share data with each other. In this study we adopt Service-Oriented Architecture (SOA) to the core-banking system using foreign exchange import business as example and focusing on the SOA solution for the integration of the L/C (letter of credit) business process. The Service-Oriented Modeling and Architecture (SOMA) approach is used to analyze the issuing business process of the import L/C. The web services in the process are identified and then realized that enable reusability of the existing services through the exercise of service orchestration and composition. According to the established SOA-layered architecture of the L/C management, we then implemented a new foreign exchange import system. With the features of service reusability and service composability, the system development time and costs are reduced and the integration can be easily achieved. The SOA solution of this study improves the use of our case bank's internal information system and the developed prototype should be able to serve as a reference for other banks.

Keywords: Service-oriented Architecture (SOA), Web Service, Business Process, Core-banking System, Foreign Exchange Import, Letters of Credit (L/C)
in the past the financial sector used the functional information system where most of the information was dispersed on different hosts making data integration difficult.

Taiwan’s economy relies mainly on international trade, where importers and exporters primarily use letters of credit as a form of transaction, and these letters are issued and guaranteed through the bank. Along with the annual growth of total commercial trade value, the letter of credit business has grown in proportion to the total bank revenues. As the banks compete amongst each other for business, it has become a first and foremost issue as to how to provide customers with fast and customizable services.

From an information viewpoint, how to integrate the two aspects, service and technology, into the system has become an important question in the banks’ IT department. Firstly, the staffs are facing the increasingly complex IT infrastructure. More and more systems are developed independently, thus more and more systems need to be able to communicate with each other to complete the complex business functionalities. Secondly, the staffs urgently demand a more flexible business model. Coming from an increasingly competitive environment, there is an even greater need for a system that can handle a faster business and market it thereafter. At the same time, there is a need for an innovative business to attract more customers.

This objective of this study is to use Service-Oriented Architecture (SOA) as a method to solve the financial business systems integration problem. SOA provides an interoperable and integrative technical architecture which commonly uses the popular implementation technology of web services with distributed, loosely coupled and cross-platform features as well as the advantageous integration capability (Erl, 2005; Papazoglou, 2008). This can assist the financial and banking services in decreasing the overall complexity of the banking system and thereby optimize the operational efficiency of banks in facing the challenges that occur due to changes of the market, businesses, and rules.

Therefore, in this study we shall investigate the SOA solution in resolving bank’s letters of credit business of foreign exchange import, and hope to achieve the goals of providing a prototype financial service-oriented architecture and fast integration and development of the operational systems.

LITERATURE REVIEW

Service-Oriented Architecture (SOA)
SOA as a design concept is a multilayered computing architecture which helps organization in sharing business logic and data among multiple application programs and application modes. It is based on business process in mind with a set of service oriented design principles: Service Contract, Service Loose Coupling, Service Abstraction, Service Reusability, Service Autonomy, Service Statelessness, Service Discoverability, and Service Composability (Erl, 2005; Erl, 2008). The concept being that the use of an SOA provides agility, i.e., a better ability to change the IT infrastructure or business processes on top of existing services faster to adapt to changing needs of the business (Linthicum, 2009)
Business Process Execution Language (BPEL)

BPEL or WS-BPEL (OASIS, 2007) is a model and language for specifying business process behavior based on Web Services. It defines the basic actions for complex business processes and directs business processes to utilize web services to achieve the goals. Moreover, it enables applications for business processes to cross platforms using the web services provided. Processes in BPEL export and import functionality by using Web Service interfaces exclusively.

Service Component Architecture (SCA)

SCA (OASIS Open, 2011) provides a programming model for building SOA-based applications and solutions. Furthermore, this model supports a wide range of technologies for execution services and connectivity modes. SCA provides an open and technology-agnostic model and the interface is in accordance to the business function defined service. Software developers can easily use a unified interface to access the functions provided by the other components.

Service Data Object (SDO)

SDO (OASISP, 2007) is a framework for data application development that includes architecture and an associated Application Programming Interface (API). Its main purpose is to facilitate data programming so that developers can concentrate on the business logic instead of the underlying technologies. SDO unifies data representation across disparate data stores, simplifies the J2EE data programming model, and supports and integrates XML. Most important, it is the common data model for SOA because it provides a layer of abstraction (Iyengar et al., 2008).

Service-oriented Modeling and Architecture (SOMA)

SOMA (Arsanjani, 2004; Bieberstein et al., 2008) method for analysis, design, and implementation has three phases: service identification, service specification, and service realization as shown in Figure 1. The approach of service identification can be distinguished into three types: (1) top-down, that means via domain decomposition analyzing business process to find candidate business services; (2) bottom-up, that means via
existing system analysis to identify candidate services; (3) meet-in-the-middle, that means finding out business-aligned services with goal-service modeling. The service specifications are mainly used to describe the service models. They define the interface, operation, and I/O messages according to the reusability, business rules and process control of the identified service candidates, so as to enable interactions with other services. The service requesters know the details of services, and the service providers know the status of how services are used. The last phase is service realization, including realization decisions, allocation to components, etc.

**Letter of Credit (L/C)**

Letters of credit are documents issued by the bank in accordance with the request of the importer to the exporter (beneficiary). The bank issuing the said document is committed to the specified conditions thereon, and is responsible for the payment of the loan in the place of the importer. Therefore, letters of credit are documents that guarantee payment. So both importers and exporters must abide by the contract provisions stating the use of letters of credit as the form of payment. The importer must request the bank to issue a document, namely the letter of credit, L/C. The content of the L/C given to the exporter is based on the L/C application, which is applied by the importer and submitted to the bank for issuing the letter. When the bank receives certification of shipping, the issuing bank will notify the importer to pay. The L/C issuing process can be divided into four sub-processes: registration of issuing, verification by branch manager, electronic message operations, and verifying the electronic message (as can be seen in Figure 2).

1. Issuing registration: Customer provides the application form and transaction contract to the bank to apply for the letter of credit. The branch tellers will verify whether the documents are in order and ensure that the beneficiary is not blacklisted. If approved, the system will establish a case of the L/C and send it for the next processing, otherwise the system will withdraw the case and return it to the customer.
2. Branch auditing: The branch manager will verify whether the contents of the case are correct and generate account entries according to the amount of money. If the case is not approved the manager will note the reasons for rejection and return it to the original branch. The teller will then record again the contents for the case according to the reasons for rejection or return it to the customer.

3. Message operations: The commissioner when receives the verified case will decide the notification bank, the method of cargo delivery, and the documentary bills bank according to the contract as well as decide which electronic messages are to be sent to the partner bank. If there are problems with the transaction contract, it will be returned to the branch and the teller shall retype and EC (eliminate account transactions). Otherwise it is ready to be sent to the next stage.

4. Auditing message: The manager of the center will verify whether the contents of the electronic message are correct. The correctly produced electronic document will then be sent to the banks overseas.

**Challenges Faced**

1. There are complex and diverse financial information systems with various communication protocols and programming languages.

2. It is difficult to expand and integrate system that lacks of flexibility and agility.

3. The developed business functions cannot be reused resulting in a waste of resources.
SERVICE-ORIENTED ANALYSIS AND DESIGN

We will adopt the SOMA method for the service-oriented analysis and design. This includes business process analysis, service identification, service specification and service realization. The target would be the L/C business process focusing on the four sub-processes described above.

Service Identification

With the three methods of service identification, top-down, bottom-up and meet-in-the-middle, the required service candidates can be identified. In this study we use the top-down method of analysis, which is in compliance with the gradual improvement of bank business segmentation, and the bottom-up method to analyze the currently existing systems. Top-down domain decomposition analyzes the four major processes. The possible services identified are described below:

1. Decompose the business process: Take the L/C business process of Figure 2 and break it down into a series of granular process steps we identify several sub-processes and service candidates.

   1. Issuing registration: L/C issuing, Branch teller accepts importers application for L/C, Establish cases (take a number and create an L/C case), Search customer information, Search blacklist, Search bank information, Compute fees, Query branch unfinished transactions, Close case, Submit a case to the next stage, Check if the customer has sufficient quota.

   2. Branch auditing: Branch audit, Fetch case, Examine L/C content, Generate accounting item, Deliver electronic message 301 to appropriate company’s quota, Submit or return the case.

   3. Message operations: Fetch case, Import message template, Introduce commonly used message items, Check message with the SWIFT (Society for Worldwide Interbank Financial Telecommunication) format, Query unfinished transactions, reject case and EC (eliminate account transactions), Send case, Generate message.


2. Further analyze the processes to identify service operation candidates and candidate service compositions, and to abstract orchestration logic:

   1. Which contains the process logic belongs to process service: L/C issuing, Branch auditing, Message operations, Auditing message.

   2. Manual operations are eliminated from the service candidates: Branch teller accepts importers application for L/C, Branch supervisor examines L/C content, Commissioner in the center enters message data, Supervisor in the center sends the case to the next stage and waits for a foreign bank reply.

   3. Related operations are combined into a service:

      i. The four operations that branch teller sends a case to the supervisor for next process, the supervisor of the branch
office sends the case to the managing center, center commissioner sends the case to the manager in charge, and supervisor in the center sends the case and waits for a foreign bank reply are combined into a “Send case” service.

ii. Query of branch unfinished transactions and query of center unfinished transactions are combined into a “Query to-do items” service.

iii. Each the case returned to the previous business unit is defined as a “Reject case” service.

iv. The three operations that branch supervisor, center commissioner and center supervisor fetch cases from the foreign currency import platform were combined into a “Fetch case” service.

4. Abstract the process (orchestration) logic from the business logic: Service “Start ACK/NAK and wait for reply” contains process logic and is abstracted to form a new “Message wait for reply” process service and a new “ACK/NAK wait for reply” business service.

5. Abstract the application logic from the business logic: Service “Start ACK/NAK and wait for reply” starts SWIFT message system and calls “Generate message” service. Thus its application logic is abstracted into the “SWIFT” component.

3. After elimination and abstraction, the services are categorized according to their businesses:

1. Import L/C: Take a number, Create case.

2. Customer information search: Search
customer information, Search blacklist.


4. Quota information search: Calculate service charge, Check quota, Appropriate quota.

5. To-Do items: Query to-do items.


7. Accounting information: Generate account, EC.

8. SWIFT: Import message template, Introduce commonly used items, Check message data, Generate message, Send message, ACK/NAK wait for reply.

Figure 3 shows the service layer of the L/C process after identification where the service composability is achieved based on their functionalities, service attributes, and business logic. Note that the application services in the bottom layer are fulfilled with components. The SOA layered architecture of the import L/C can then be established as shown in Figure 4.

**Service Specification**

In order to pave the way for later service realization, during the service specification the business and technology used must be aligned. In this phase the services identified will have uniform interface and can interact with each other via the interface, thus achieving the process goals and facilitate the service realization. Table 1 defines a variety of service interfaces including the service input and output.

**Service Realization**

In this study we utilize IBM WebSphere Integration Development (WID) v6.1 for development. It is an integrated software for process development based on SOA. The graphic user interface of the software can help guiding the development. After the development, we then use IBM WebSphere Process Server (WPS) v6.1 for the deployment and test. During the service development, we follow the technical specifications of SCA and encapsulate each service into service component.

The foreign exchange import L/C service module (ImportIssue) includes Assembly diagram, Business logic, Data Types, and Interfaces.

1. **Assembly Diagram:** Services are encapsulated into components; as can be seen in Figure 5, the four service components on the left represent respectively the four sub-processes of the foreign exchange letter of credit import service: Issuing Registration, Branch Auditing, Message Operations and Auditing Message, which reference the 19 service components as defined in Table 1.

2. **Interfaces:** Since service components encapsulate related services, every service component will have a service interface and have the data input and output defined. Figure 6 shows an example interface, Fetch Specific Case Interface, which contains an operation named fetch Specific Case Service.

3. **Data Types:** Data type’s name with BO (business object) is defined according to the SDO specification. Figure 7 shows some examples.

4. **Business Logic:** As to business logic, process is executed via “Assign” and “Invoke” service and rule components. Figure 8
Figure 4: The SOA layered architecture of L/C issuing
<table>
<thead>
<tr>
<th>Service</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Take a number</td>
<td>IsDbu, branchCode</td>
<td>RemitNumberDocBO</td>
</tr>
<tr>
<td>2 Create case</td>
<td>RemitNumberDocBO</td>
<td>CaseBo</td>
</tr>
<tr>
<td>3 Search customer information</td>
<td>Id, Is799, isTerrorists</td>
<td>CustomerInformationBO</td>
</tr>
<tr>
<td>4 Calculate service charge</td>
<td>Ccy, amount, rate</td>
<td>Fee</td>
</tr>
<tr>
<td>5 Search bank information</td>
<td>BankCode</td>
<td>BankBo</td>
</tr>
<tr>
<td>6 Query to-do items</td>
<td>TellerId, branch, stage</td>
<td>ToDoItemBO</td>
</tr>
<tr>
<td>7 Close case</td>
<td>TellerId, branch, stage</td>
<td>True/False</td>
</tr>
<tr>
<td>8 Send case</td>
<td>TellerId, stage, CaseBo</td>
<td>True/False</td>
</tr>
<tr>
<td>9 Check quota</td>
<td>Id, account</td>
<td>Limit</td>
</tr>
<tr>
<td>10 Fetch case</td>
<td>TellerId, stage, caselId</td>
<td>CaseBo</td>
</tr>
<tr>
<td>11 Generate accounting account</td>
<td>caselId, stage</td>
<td>True/False</td>
</tr>
<tr>
<td>12 Appropriate quota</td>
<td>Id, ccy, account, account</td>
<td>True/False</td>
</tr>
<tr>
<td>13 Reject case</td>
<td>TellerId, stage, CaseBo</td>
<td>True/False</td>
</tr>
<tr>
<td>14 Message template</td>
<td>MessageType, template</td>
<td>SwiftTempletBO</td>
</tr>
<tr>
<td>15 Commonly used message items</td>
<td>MessageType, tag</td>
<td>messageItem</td>
</tr>
<tr>
<td>16 Check message data</td>
<td>ImportSwiftContentBO</td>
<td>True/False</td>
</tr>
<tr>
<td>17 Generate message</td>
<td>caselId, stage, ImportSwiftContentBO</td>
<td>True/False</td>
</tr>
<tr>
<td>18 Send message</td>
<td>caselId, stage, messageId</td>
<td>True/False</td>
</tr>
<tr>
<td>19 ACK/NAK wait for reply</td>
<td>caselId, messageId</td>
<td>ACK/NAK</td>
</tr>
</tbody>
</table>

represents the four processes of L/C Issuing Registration, Branch Auditing, Electronic Message Operations, and Auditing Message. The resources used include interface partners, reference partners and process variables.

**SIMULATION RESULTS AND DISCUSSION**

After analysis and design with SOMA, we created uniform interface for services via service specifications thus the services can communicate with and be linked to each other via the interface to make themselves reference partners of the process services. The process services arrange and compose the services in BPEL-compliant way according to the business logic. During the execution, the needed service data is encapsulated, assigned, and transmitted with business objects. Services are invoked according to process logic. The business goals are achieved via this series of activities. In addition, where there are changes in the arrangement of process services, the subordinate services of

---

This article can be downloaded from http://www.ijmrbs.com/currentissue.php
Figure 5: Assembly Diagram

Figure 6: Fetch Case Service Interface

This article can be downloaded from http://www.ijmrbs.com/currentissue.php
Figure 8: Business Processes
lower layer can accommodate with the changes via the uniform interface, so as to achieve the features of loose coupling, flexibility and agility.

CONCLUSION

In the current complex and intensely competitive financial market, the banking and financial service sectors are facing serious challenges. Banks not only need to be flexible in response to the challenges brought upon by globalization, but they need to face numerous difficulties of service localization. Only through active participation in the fierce competition of the emerging markets can they become invincible, reveal themselves and retain their distinctive qualities and form unique core competitiveness. For this sake, the financial products of banks must be approachable and remain innovative. The banking and financial systems must be efficient at integrating their services in order to provide different products and services for various customers.

In this study, the SOA practice uses the bank L/C service integration system as example, in which services are loosely coupled, reusable and composable. The spirit of SOA is not to abandon the old systems and rebuild new ones, but to identify services according to the process logic of the existing systems and establish the SOA layer. With the BPEL-compliant service orchestration engine the processes can easily compose business and application services and reuse resources to achieve the business goal.

From above discussion, using service-oriented architecture to design and improve the letter of credit business process has the following benefits:

(1) The problems of system expansion and integration difficulty, less flexibility and low agility which resulted from diverse technologies, systems and programming languages are solved.

(2) With our SOA solution, the business functions (services/components) can be reused thus reducing resources waste of IT establishment, man-power, time and maintenance cost.

REFERENCES


Process Server, BPEL, SCA and SOA. Indianapolis, IN: IBM Press.


