ISSN: 1839 - 0846

Service-Oriented Analysis and Design for Constructing the Online Sales Process Integration

Yung-Hsin Wang (corresponding author)
Department of Information Management, Tatung University,
No. 40, Chung-Shan North Road, Section 3, Taipei 104, Taiwan
ywang@ttu.edu.tw

Shih-Chih Chen
Department of Information Management, Tatung University,
No. 40, Chung-Shan North Road, Section 3, Taipei 104, Taiwan
scchen@ttu.edu.tw

An-Jung Tseng
Department of Information Management, Tatung University,
No. 40, Chung-Shan North Road, Section 3, Taipei 104, Taiwan

ABSTRACT

In recent years, enterprises try to develop the online sales systems in order to reach various customers in different places and expect to bring better profit or company exposure via the internet and mobile device channels. However, different platforms, technologies, transaction rules, policies and processes have resulted in data dispersion and business logic inconsistency that is hard for the company to audit and manage. This study presented a service-oriented analysis and design of the online sales process integration for our case company. We analyzed and modeled the processes using the service-oriented modeling and architecture (SOMA) approach to identify service candidates, and also utilized service component architecture (SCA) to create web services for activities required for the processes. Based on the function of services, business rules and logic, the corresponding service abstraction layer of the integrated online sales process is established. Finally, this study used the Business Process Execution Language (BPEL) to realize the service orchestration and composition, thus achieving service reusability. With the SOA solution, the case company can efficiently integrate resources and that the enterprise system become more flexible, agile and cost-effective.

Keywords: *Online Sales Process, SOA, BPEL, Web Service, Process Integration*

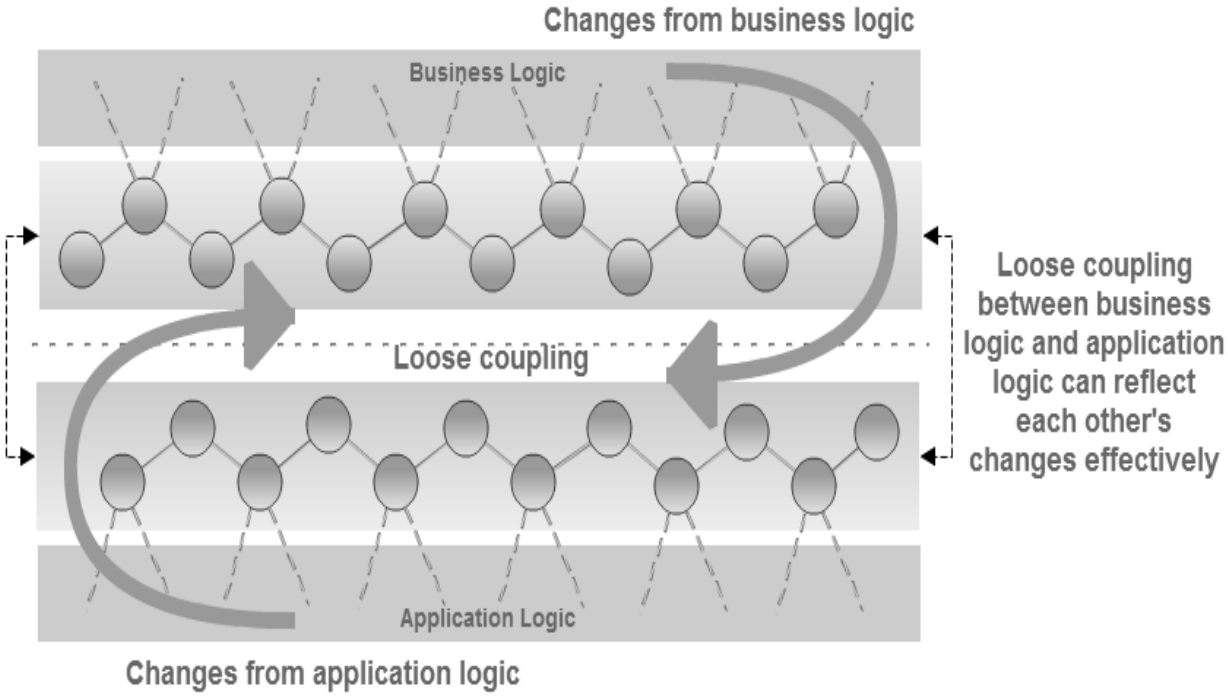
1. INTRODUCTION

In recent years, enterprises try to develop the online sales systems in order to reach various customers in different places and expect to bring better profit or company exposure via the internet and mobile device channels. However, different platforms, technologies, transaction rules, policies and processes have resulted in data dispersion and business logic inconsistency that is hard for the company to audit and manage. The case company of this study is specialized in professional design of information and communication products and is also an OEM. It has established branches in Asia, North America and Europe besides headquarter in Taiwan. Each branch sets up online sales platform, confirms the order, and approves application for goods return on its own due to different business targets and sale strategies. Also because of different staff organization and operating fashion in each branch, there exists different order confirmation process and goods return approval process, which leads to inconsistence in approval criteria, response time hard to control and low customer satisfaction. Therefore, integration and improvement on processes are in need of being made.

Different technology languages and frameworks often cause difficulties in system collaboration and increase in maintenance cost. Besimi (2011) indicated that Small and Medium Enterprises had trouble in dealing with inconsistent B2B messages communication due to lack of IT experts and technical capabilities. Enterprises also are unable to provide appropriate services to customers of different scales and keep consistence in business rules at the same time (Zimmermann et al., 2005). Hence, the objective of this study is to form business automation being controlled and running collaboratively through architecture enhancement as well as achieve resources integration.

ISSN: 1839 - 0846

This paper illustrated firstly current business processes and improvement goals and then establishes an information system through service-oriented architecture (SOA) that combines business functions required with various service compositions, which has standard interface, features loose coupling and high degree of integration. As shown in Figure 1, the established service abstraction layer of the integrated process should promote loose coupling throughout the enterprise and achieve organizational agility to increase overall responsiveness to business change, allows enterprise to rapidly respond market needs and satisfy urgency of marketing department (Erl, 2005).



From above discussion, loose coupling allows business logic and application technology to effectively reflect changes of each other. In this study, the following goals will be achieved through analysis, design and integration of processes:

(1) With the Web services standardization all the dispersed participants, business processes and application systems can be integrated.
(2) The order confirmation process and sales return verification process are used to integrate multi-line processes and form a major business process with consistent interface.
(3) With our SOA solution the case company can efficiently integrate resources and the system is more flexible, agile and cost-effective.

2. RELATED CONCEPTS AND METHODS

2.1 Service-Oriented Architecture (SOA)

SOA is a multi-layered computing architecture, which helps organization in sharing business logic and data among multiple application programs and application modes. In order to achieve this, SOA needs to follow the characteristics and service-oriented principles such as standardized service contract, service loose coupling, service abstraction, service reusability, service autonomy, service granularity, service statelessness, service discoverability, and service composability (Erl, 2005; Valipour et al., 2009).

SOA enables function modulation and packs business into service. One service defines one interface being related to business function and constraints of the interface contract. Through efficient organization and composition, services will be effectively reused and deviated from particular technology or particular platform, making integration of dissimilar systems become easier.

2.2 Service Component Architecture (SCA)

Separation of interface with technology will provide a model being irrelevant to technology. The developers could apply unified interface to describe and store functions of other components. SCA provides a programming

ISSN: 1839 - 0846

model to establish SOA applications and solutions which supports a wide range of service execution technologies and binding modes. The SCA specifications define how to create components and how to combine those components into complete applications (Chappell, 2007; Bell 2008). In the SCA model, data and messages are exchanged in a Service Data Object (SDO) (OASIS SDO TC, 2007). The Service Component Definition Language (SCDL, XML-based format) is used to describe the components an SCA composite contains and specify how they relate to one another (Sosinsky, 2011).

2.3 Business Process Execution Language (BPEL)

BPEL (BEA et al., 2003; OASIS WSBPEL TC, 2007) is a model and language to describe the action of business process which interacts with other partners (via WSDL interface). It defines the basic actions for complex business processes, directs business processes to utilize web services to achieve the targets, and enables cross-platform application of business processes for different products through the web services provided.

2.4 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. The approach of service identification can be divided 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 users 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.

3. ONLINE SALES PROCESS INTEGRATION

Current sales platform in the case company which applies PHP and JAVA into development has been running for years. The improvement has been made on the user interface but not on integration of processes over these years. When striving for integration the information department made more efforts on .NET related part which they are more familiar with. If there is a set of unified business rules and control mechanism available for processes the probability of errors will be reduced, thus avoiding subsequent problems derived and increasing degree of customer satisfaction.

At present, after receiving orders placed by customers, each platform will carry out order confirmation process and the standard varies with judgment rules adopted by each platform. Every platform is independent which leads to a state that each branch often sacrifices risk control for profit pursuing. When receives application of goods return from customers, there is no quantified standard provided for judgment in the approval process.

The following major problems from the viewpoint of management proposed as follows:

(1) Business supervisor and financial supervisor in the headquarters are unable to control order or goods return. They rely on the supervisor in each branch to report so that processing urgency of the sales department cannot be satisfied.
(2) Due to frequent variation of credit situation of customers, each platform cannot acquire customer credit status in real time.
(3) Order approval criterion is not based on monetary amount or credit status of customers; the criterion varies and the process execution time is too long.

From the viewpoint of technology, major problems are as follows:

(1) Functions are segmented and isolated. The same function is described and developed at the discretion of the developers making it hard to reuse.
(2) When demand occurs, it spends more time in evaluating approaches and possible impact as well as concerning labor investment.
(3) The long-term functional test, correction and introduction affected the schedule of demanding unit.

This study redesigned the order confirmation process and sales return verification process which had more impact on the operation cost of the case company that require prudent control. There are many overlapping procedures in the processes.

ISSN: 1839 - 0846

From the aspect of design and analysis, SOMA method is applied into identifying and establishing relevant services and produces SOA layered architecture (Erl, 2009) for the study case, followed by service design, BPEL process design and result validation. In this way, the business logic that is separately expressed by each branch will be integrated into loosely coupled and reusable business operating unit or service, and finally achieves a business demand driven integration mode.

4. SERVICE-ORIENTED ANALYSIS AND DESIGN

As has been indicated, using SOMA to identify and establish the related services include three stages: service identification, service specification and service realization.

4.1 Service Identification

In this stage, we analyzed the major processes separately in order to identify service candidates. After understanding business processes, business rules, related events and involved person, service candidates will be identified according to process execution efficiency and policy or demand change in combination with resources available. The classification of these service candidates should be judged too.

In the order confirmation process, each order information and amount is recognized by the order auditor, so we need to call the customer information and order information on two data files, these are reusability services. Order auditor used auto approval service to confirm, if orders are not approved by auto approval service, orders will review by business executives and chief financial officer.

As for the order confirmation process, candidate services are found as follows:

(1) Order confirmation process: a process service with process logics and no operations, which is initiated by the ordering system.
(2) Auto approval service: a business service used to judge whether orders need further review and treatment.
(3) Credit rules service: a business service.
(4) Sales approval service: a business service.
(5) Financial approval service: a business service.
(6) Customer information service: an application service with customer data inquiry function and high degree of reuse.
(7) Order information service: an application service with order data inquiry function.
(8) Credit information service: an application service with quality credit data and provided to each process to inquire credit status.

In the sales return verification process, order information and order amount are recognized by the return auditor, then it will go through the credit rules service, if credit level of customer is c, order will review by chief financial officer.

As for the sales return verification process, candidate services include:

(1) Return verification process: a process service meeting service requirements by composing services of lower layer, which is initiated by the canceling system.
(2) Credit rules service: a business service.
(3) Financial approval service: a business service.
(4) Customer information service: an application service.
(5) Order information service: an application service.
(6) Credit information service: an application service.

In above two processes, auto approval service, credit rules service, customer information service, order information service and credit information service are services provided by legacy systems. We moved them into application service layer and establish a set of auto approval service and credit rules service in business service layer to provide an integrated service. Figure 2 shows the service layer of the two processes, displaying the relationship of service properties identified and the composability feature. Figure 3 depicts the established SOA layered architecture.

ISSN: 1839 - 0846

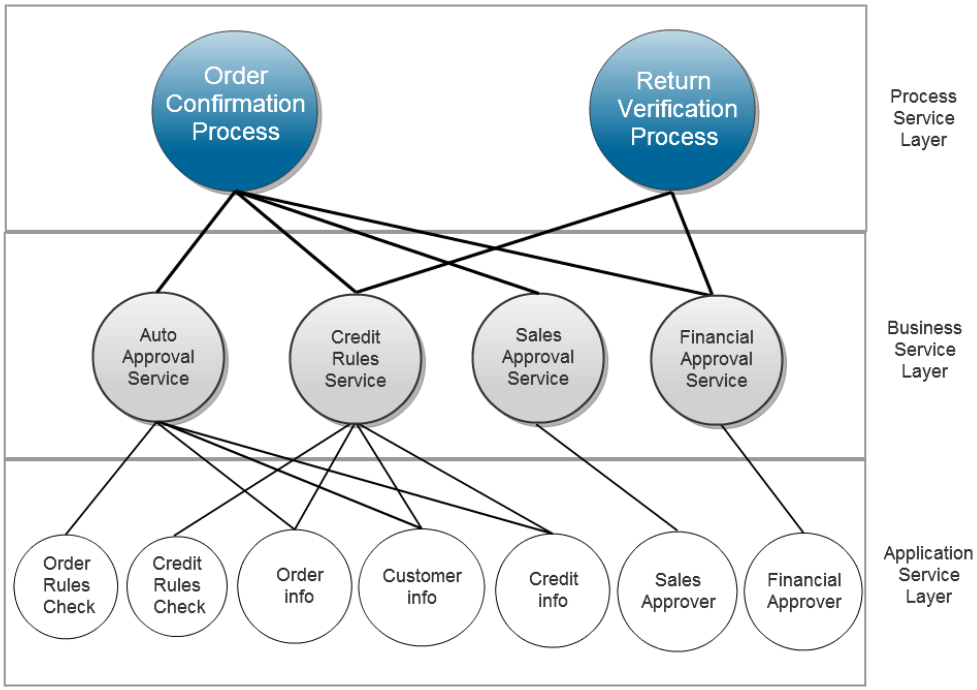


Figure 2. Service layer and composability

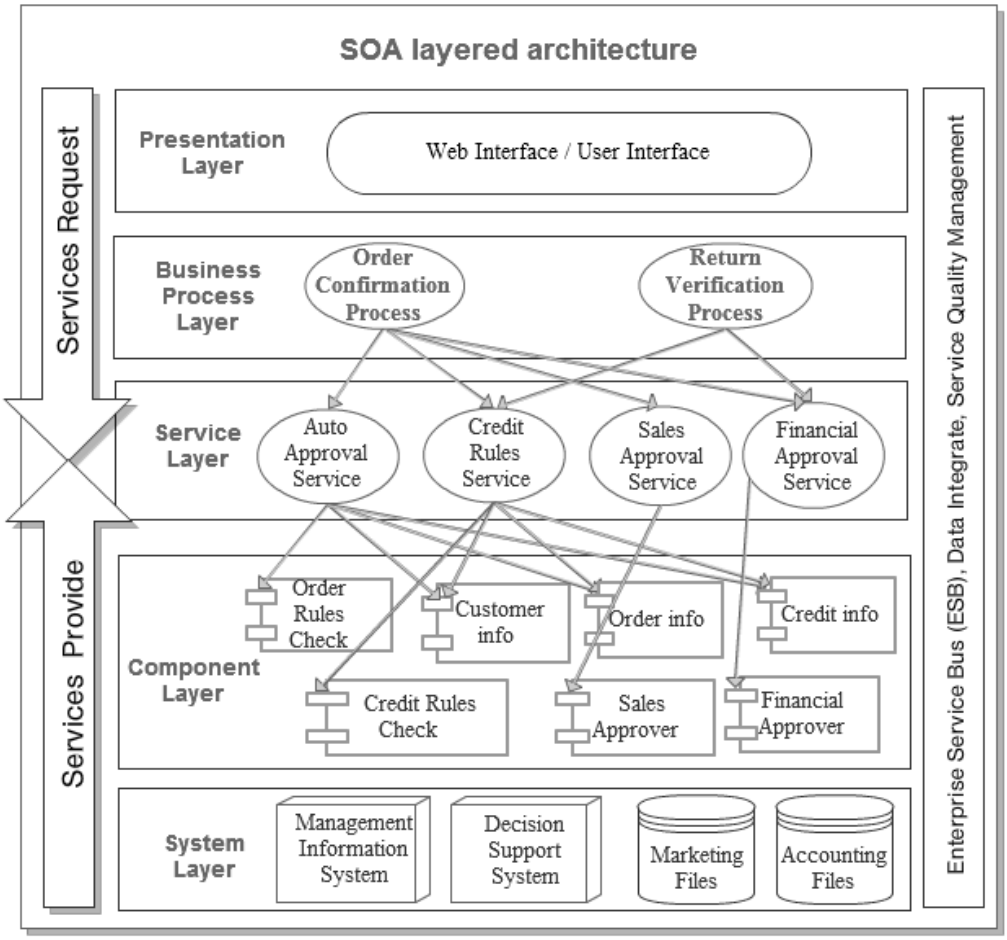


Figure 3. The SOA layered architecture of this study

ISSN: 1839 - 0846

4.2 Service Specification

In this stage, services identification has uniform interface and can interact with each other via the standard interface, thus achieving the process goals and facilitate the service realization.

(1) Definitions of service interface

Service-oriented architecture is to combine systems through interface standard. In order to ensure service availability, before service is launched to the public, it needs a set of standard specification. Therefore, this study provided definitions of business object and service interface in details (Table 1), allowing service to be interactive through well-defined interface.

(2) Business rules

After defined interface, rule logic will make logical judgment from data fields. The rules interface used are defined as Table 2, the results of judgment will make process go through different routes. This rule varies with changes of transaction and risk policy.

Table 1. Service interface definition

Service	Input	Output
Order confirmation process	Order info (OrderRequestBO)	ProcessingResult
Return verification process	Return application form (ReturnRequestBO)	ProcessingResult
Auto approval service	Order info, auto approval rules	CheckResult
Credit rules service	Credit info, Credit rules	CheckResult
Sales approval service	SalesApproverBO	Boolean
Financial approval service	FinancierApproverBO	Boolean

Table 2. Business rules definition

Auto approval rules	
Input	Order info, auto approval rules
Condition	IF OrderRequestBO.PurchaseAmount <= 50000 AND OrderRequestBO.OrderInspectorChecked == true THEN return CheckResult = true ELSE return CheckResult = false
Output	CheckResult

4.3 Service Realization

In this study, we utilized IBM WebSphere Integration Development (WID) v6.1 for development. It is a software for process integration development based on SOA. The graphic user interface of the software can help guiding the development. After the development, we adopted the IBM WebSphere Process Server (WPS) v6.1 for the deployment and test.

During the service development, we followed the technical specifications of service component architecture (SCA), and encapsulated each service into service component.

(1) Assembly Diagram

As can be seen in Figure 4, this module includes two process components (OrderConfirmationProcess and ReturnVerificationProcess), two human tasks (SalesApprover and FinancialApprover), and two rule component (AutoApproval-Rules and CreditCheckRules). Each component has a service interface.

ISSN: 1839 - 0846

(2) Interfaces
By encapsulating related services into service components, each service component will have a service interface and need to define data of input and output; through the interface, it can input data and return results. Figure 5 shows the example interfaces.
(3) Data Types
Data type's name with BO (business object) is defined according to the SDO specification. Figure 6 shows an example.

(4) Business Logic
As to business logic, process is executed via "Assign" and "Invoke" service and rule components. Figure 7 represents the Order Confirmation Process. Resources used include interface partners, reference partners and process variables.

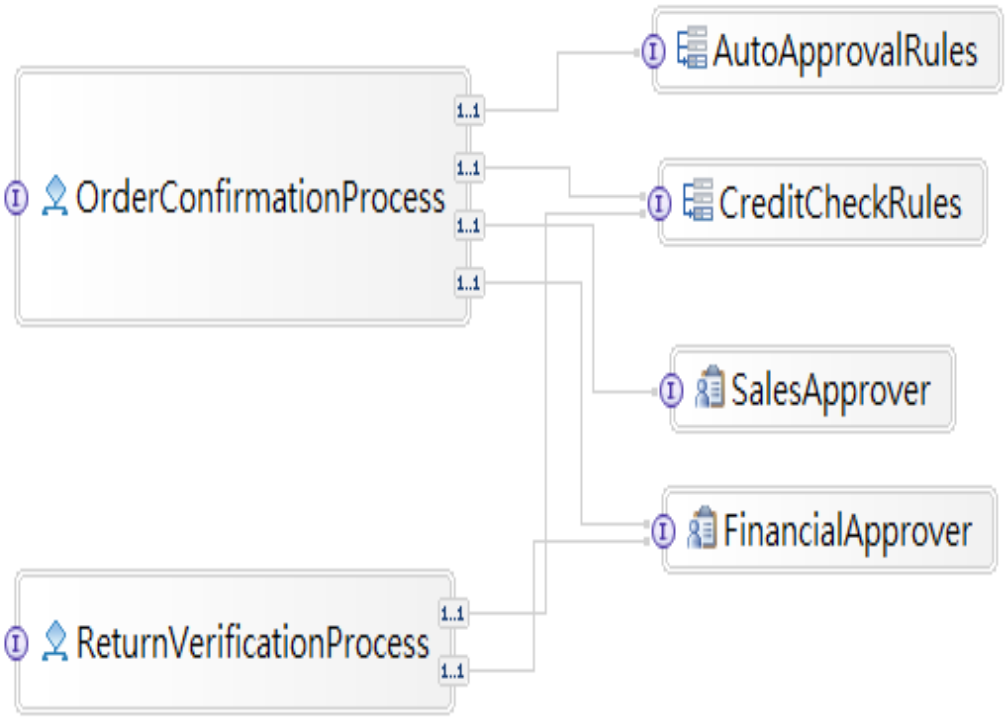


Figure 4. Assembly diagram

5. DISCUSSION

This study analyzed and designed the processes with SOMA trying to create uniform interface for services via service specifications, and the services can communicate with each other via the interface to make themselves the 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 services of lower layer can accommodate with the changes via the uniform interface, so as to achieve the features of loose coupling, reusability, flexibility and agility.

ISSN: 1839 - 0846

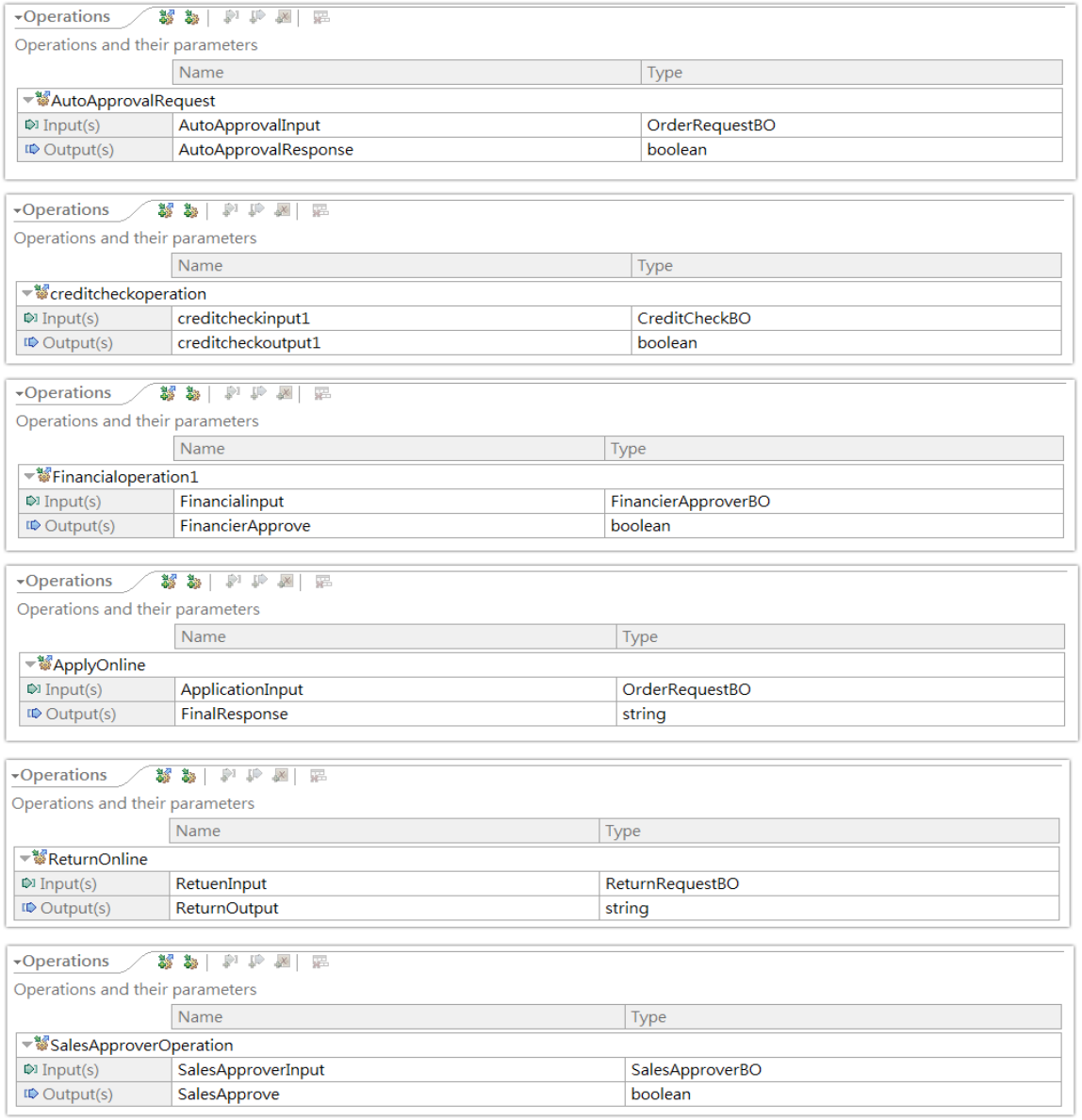


Figure 5. Example interfaces

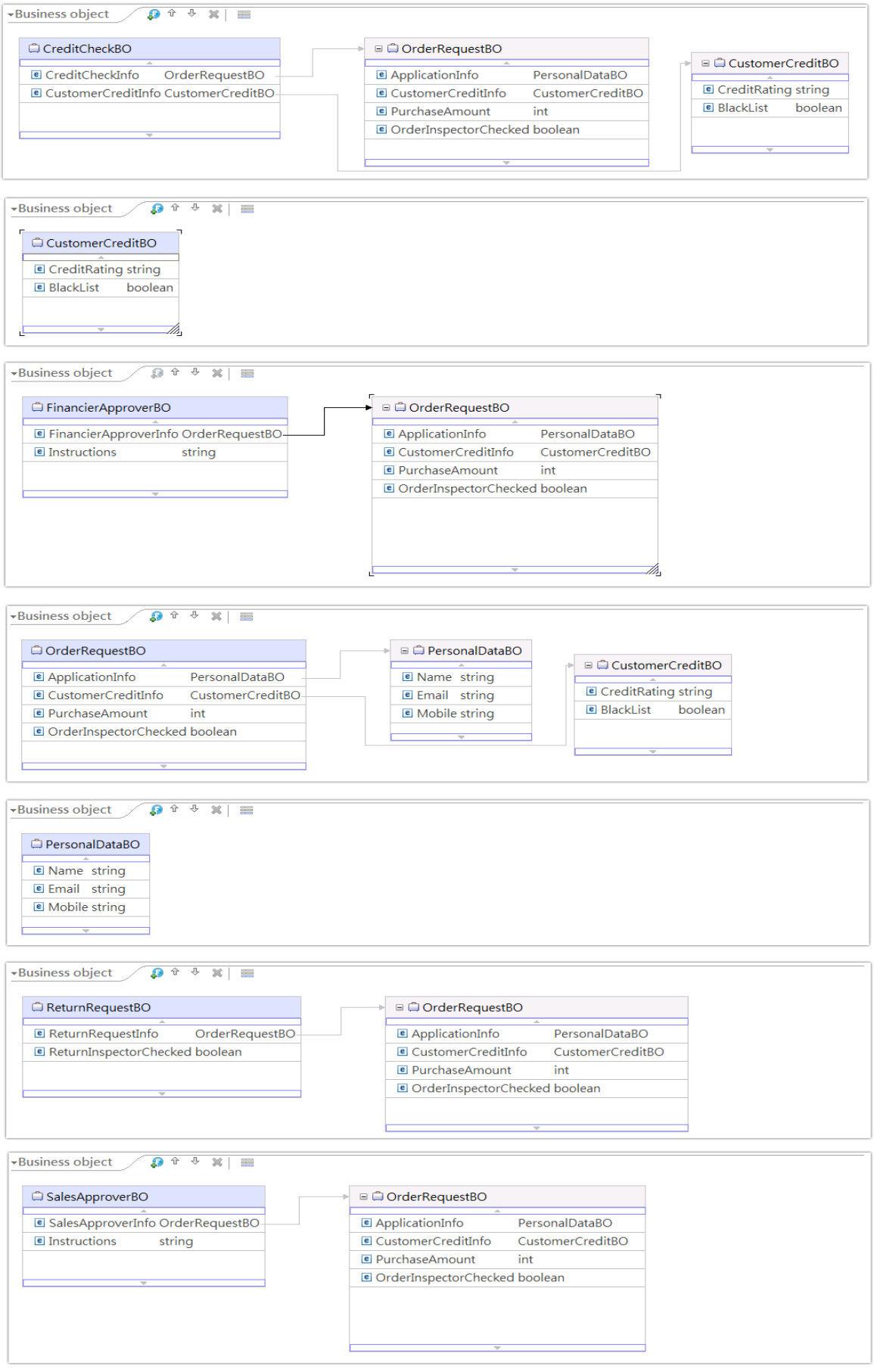
6. CONCLUSION

This study has taken the advantages of service-oriented architecture to solve the process integration problems of the case company, in which services are loosely coupled, reusable and composable. We have made some improvements as follows:

(1) By integrating different business logic and data from dispersed systems into available resources in business processes, there is no need to repeatedly develop same functions.
(2) Data has a standard specification and definition that can be easily managed and reused.
(3) The entire information system features loose coupling, which is not bound to particular platform technology.
(4) Business rules or business processes can respond quickly to accommodate with the changes to match company's policy.

In business environment nowadays, enterprises had faced same problems with limited resources and time pressure. With SOA concept, we achieve process improvement and reduce the cost while facilitating integration. The solution provided in this study may serve as reference and alternative for the enterprises facing similar issues.

ISSN: 1839 - 0846



ISSN: 1839 - 0846

Figure 6. Data types

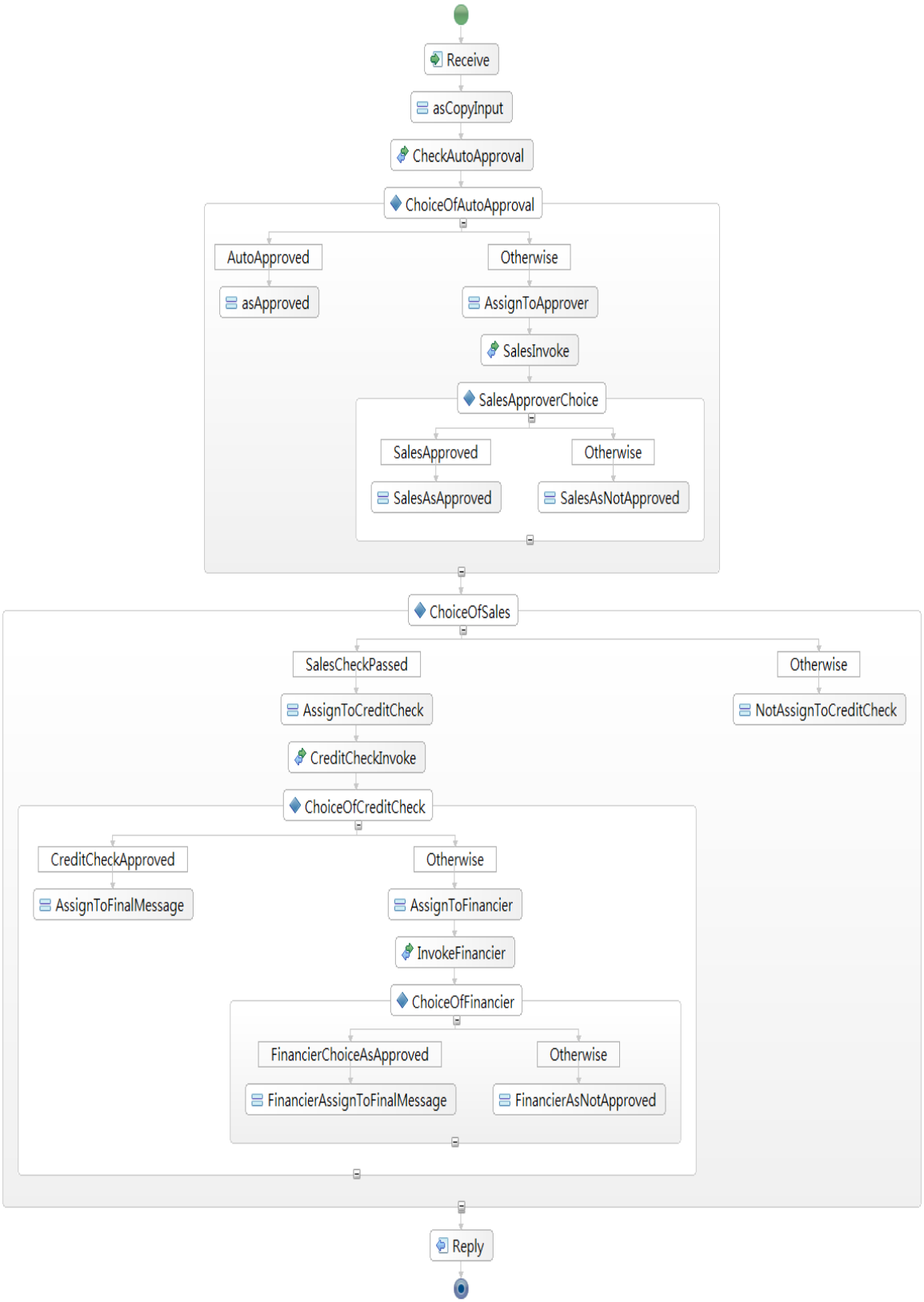


Figure 7. The confirmation process of orders

ISSN: 1839 - 0846

REFERENCES

1. Arsanjani, A. (2004). Service-Oriented Modeling and Architecture: How to identify, specify, and realize services for your SOA. Retrieved April 25, 2012, from http://www.ibm.com/developerworks/library/ws-soa-design1/.
2. BEA, IBM, Microsoft, SAP AG, & Siebel Systems. (2003). Business Process Execution Language for Web Services. Retrieved April 28, 2012, from http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf.
3. Bell, M. (2008). *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Hoboken, NJ: John Wiley & Sons, Inc.
4. Bieberstein, N., Laird, R. G., Jones, K. & Mitra, T. (2008). *Executing SOA: A Practical Guide for The Service-Oriented Architect*. Upper Saddle River, NJ: Pearson plc.
5. Besimi, A. (2011). B2B process integration using Service Oriented Architecture through Web Services. *International Journal of Computers, 5*(3), 396-403.
6. Chappell, D. (2007). INTRODUCING SCA. Retrieved April 28, 2012, from http://www.davidchappell.com/articles/Introducing_SCA.pdf.
7. Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ: Pearson Education.
8. Erl, T. (2009). *SOA Design Patterns*. Upper Saddle River, NJ: Pearson Education.
9. OASIS SDO TC. (2007). Service Data Objects (SDO) OASIS Open CSA. Retrieved May 15, 2012, from http://www.oasis-opencsa.org/sdo.
10. OASIS WSBPEL TC. (2007). Web Services Business Process Execution Language Version 2.0. OASIS. Retrieved May 15, 2012, from http://docs.oasis-open.org/ wsbpel/2.0/OS/ wsbpel-v2.0-OS.pdf.
11. Sosinsky, B. (2011). *Cloud Computing Bible*. Indianapolis, IN: Wiley Publishing, Inc.
12. Valipour, M. H., Amirzafari, B., Maleki, K. N., & Daneshpour, N. (2009). A Brief Survey of Software Architecture Concepts and Service Oriented Architecture. *2nd IEEE International Conference on Computer Science and Information Technology*, August 8-11, Beijing, China, 34-38.
13. Zimmermann, O., Doubrovski, V., Grundler, J., & Hogg, K. (2005). Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario: Rationale, Concepts, Lessons Learned. *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, October 16-20, San Diego, California, 301-312.