2011 International Conference on Computer Communication and Management
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

An Electronic Negotiation Coordinator for Software Development in Service-Oriented Environments

Amnart Pohthong [1] [+] , Chaiwat Phaiboonphuthiphong [2]

[1] Information and Communication Technology Programme, Faculty of Science,
Prince of Songkla University, Hat Yai, Songkhla, Thailand. 90110.

[2] Information and Communication Technology Center,

Office of the Permanent Secretary of Public Health, Nonthaburi, Thailand. 11000.

Abstract. Software development paradigms have been changed since service-oriented architecture (SOA) was introduced. A software service is used as a dynamically accessed process in a software system. This service is offered by any producer who publishes it in a service registry. In SOA environments, knowledge about services, providers, and clients becomes a crucial factor for building software applications. This knowledge is often incomplete and not enough for making decisions by stakeholders. Therefore, the research presented in this paper proposes a negotiation framework and development of a software system corresponding to this framework as an electronic negotiation coordinator. The system has been tested in our laboratory environment for a tourist application with hotel services and air ticket services. The results satisfy the objectives of the study.

Keywords: Electronic Negotiation, Coordinator, Service-Oriented Architecture

1. Introduction

In an ever changing world, information and communication technologies are widely used by most organizations in order to make their business competitive. Particularly, computer hardware and software plays a key role to support business operations. Software engineering methodologies should be flexible and have rapid processes to reduce time-to-market. Thus, with in the past few years, some new paradigms for software development have been introduced. Service-oriented computing (SOC) or distributed computing based on the so-called service-oriented architecture (SOA) is one of these paradigms that has gained popularity from software communities [1,2].

However, many service clients have several needs while many service providers have several offers and conditions. Knowledge about services and providers provided in a service registry is often incomplete and not enough for making decisions by clients. On the other hand, providers also lack knowledge about clients. These problems become obstacles to software development in SOA environment. Face-to-face negotiation also uses a lot of effort and can become bias. Thus, developing an electronic negotiation system offers an alternative to SOC.

2. Background

2.1. Service-Oriented Architecture

In the past, most customers and developers were familiar with traditional paradigms of software development. Customers requested developers to build whole software systems for them to meet their business's requirements. Later, software systems were composed from software components. Any selected component in the system could be replaced by a new one in order to meet new requirements. Since users' requirements have been rapidly changing, Service-oriented architecture (SOA) was introduced as an

[+] Corresponding author. Tel.: + 6674288697; fax: +6674288697.
E-mail address: amnart.p@psu.ac.th.

alternative software development paradigm. A software system using SOA, is composed of dynamically accessed components rather than static components. A service in service-oriented technology means a computing process that can satisfy requirements in a computing system. SOA can provide some key elements for building a computing system such as loose coupling, implementation neutrality, flexible configurability, persistence, granularity, location transparency, and teams [1,2]. SOA consists of three main parts as follows [3]:

(1) Service provider: a service provider should publish a service description in a service registry.

(2) Service client / customer: a service client finds a desirable service in a service registry and then requests to bind it with its provider.

(3) Registry and lookup service: this part serves as a repository that contains information about services, such as UDDI (Universal Description, Discovery and Integration).

2.2. Electronic Negotiation

In general, negotiation is a mechanism for making decisions for any benefit among stakeholders. Most negotiation consists of three main contents : process, behaviour, and substance [4].

Electronic negotiation has gained popularity since computers and the Internet technologies were introduced and adopted for many businesses. Some forms of electronic negotiation can be applied such as decision support systems, software agents, or mediation systems. A three-phase process for negotiating in a service-oriented environment: pre-negotiation, negotiation, and service delivery, was suggested in [5]. A requester process and a provider process were also suggested in [6].

3. Proposed Model and Process

3.1. Proposed Model

Our proposed model is based on a SOA model as shown as Figure 1. Knowledge about services, providers, and clients is a key factor for negotiation. However, in SOA environment, software systems are often dynamically changed that make this knowledge incomplete and not enough for making decision by both providers and clients. Therefore, a negotiation coordinator in our model can organize knowledge about services, providers, and clients and can support negotiation activities in SOA environment such as pre-negotiation, negotiation, and delivery.

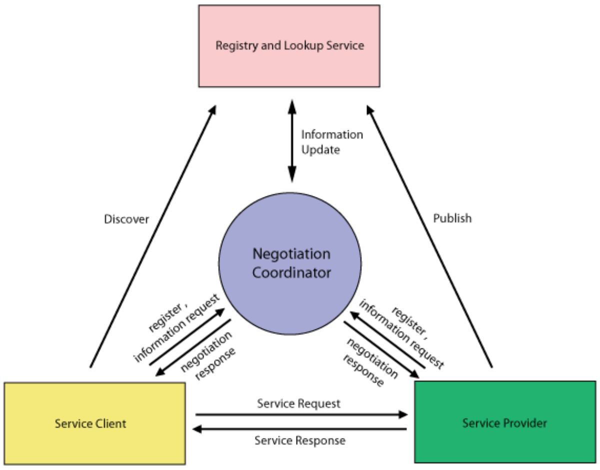


Fig. 1: A model of a negotiation coordinator for SOA.

3.2. Proposed Process

Our study began with the preliminary investigation of software development process in some standards such as CMMI, IEEE-1062, ISO/IEC 12207 in order to examine acquisition and supplier management. Then, software process in SOA environment was examined and a negotiation framework was suggested in [7]. From this framework and our proposed model, we have designed a negotiation process for software development with a negotiation coordinator in a SOA environment. The process is described into five phases as an activity diagram shown in Figure 2. These five phases are (1) requirements analysis phase, (2) selection phase, (3) proposal phase, (4) negotiation phase, (5) delivery phase.

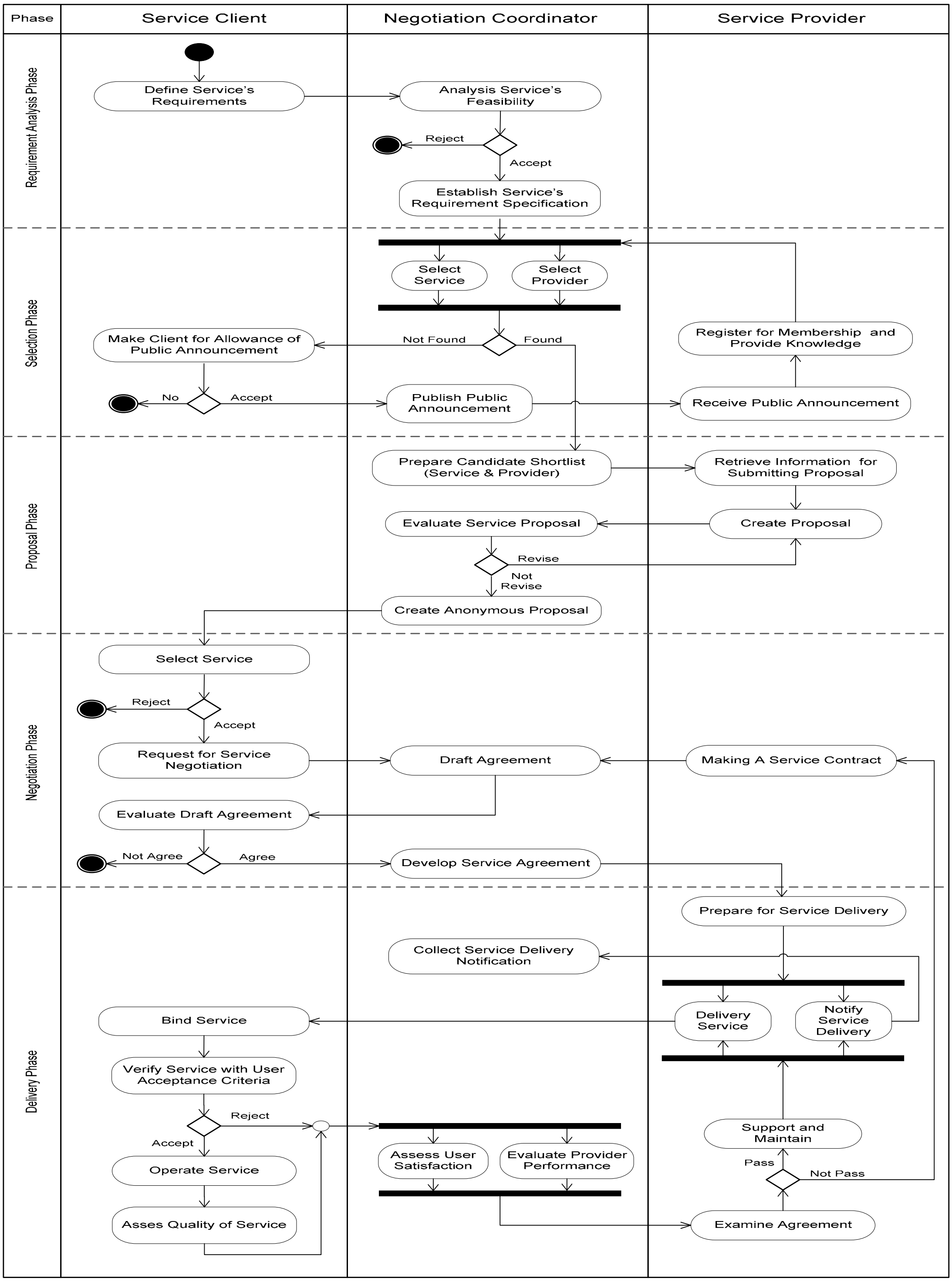

Phase
Service Client
Negotiation Coordinator
Service Provider
Requirement Analysis Phase
Define Service's Requirements
Analysis Service's Feasibility
Reject
Accept
Establish Service's Requirement Specification
Selection Phase
Select Service
Select Provider
Not Found
Found
Make Client for Allowance of Public Announcement
Register for Membership and Provide Knowledge
No
Accept
Publish Public Announcement
Receive Public Announcement
Proposal Phase
Prepare Candidate Shortlist (Service & Provider)
Retrieve Information for Submitting Proposal
Evaluate Service Proposal
Create Proposal
Revise
Not Revise
Create Anonymous Proposal
Negotiation Phase
Select Service
Reject
Accept
Request for Service Negotiation
Draft Agreement
Making A Service Contract
Evaluate Draft Agreement
Not Agree
Agree
Develop Service Agreement
Delivery Phase
Prepare for Service Delivery
Collect Service Delivery Notification
Bind Service
Delivery Service
Notify Service Delivery
Verify Service with User Acceptance Criteria
Reject
Accept
Operate Service
Assess User Satisfaction
Evaluate Provider Performance
Support and Maintain
Pass
Not Pass
Asses Quality of Service
Examine Agreement

Fig. 2: A proposed negotiation process for software development with a negotiation coordinator [7].

4. System Design

4.1. System Analysis

Activities and interactions among providers, clients, and service registry can be explained as a use-case diagram as shown in Figure 3.

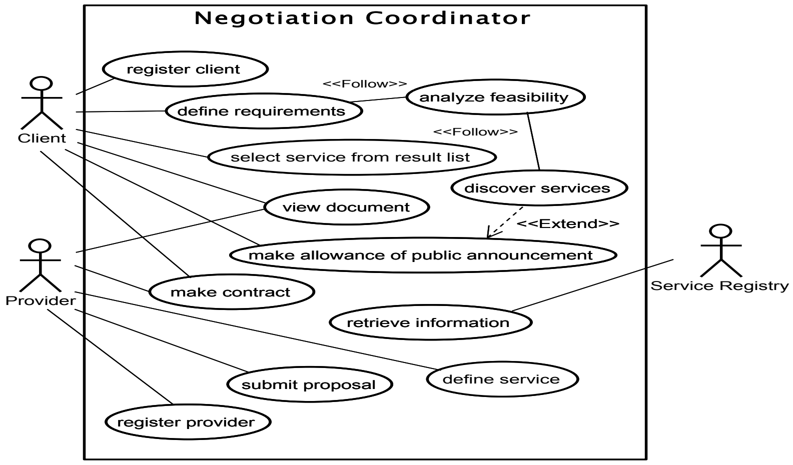


Fig. 3: A use-case diagram for a negotiation coordinator system.

4.2. System Architecture

The system of a negotiation coordinator was prototyped into seven modules as shown as a system architecture in Figure 4. These seven modules are as follows:

(1) Presenting module: for managing user interfaces for service clients and service providers.

(2) Registering module: for managing membership registration.

(3) Retrieving module: for getting data from a service registry and a knowledge repository.

(4) Knowledge module: this module is for managing knowledge in the system.

(5) Selecting module: for searching and selecting a service that meets a client's requirements.

(6) Contracting module: for managing a contract between a provider and a client.

(7) QoS (Quality of Service) analysis module: for managing information about Quality of Service.

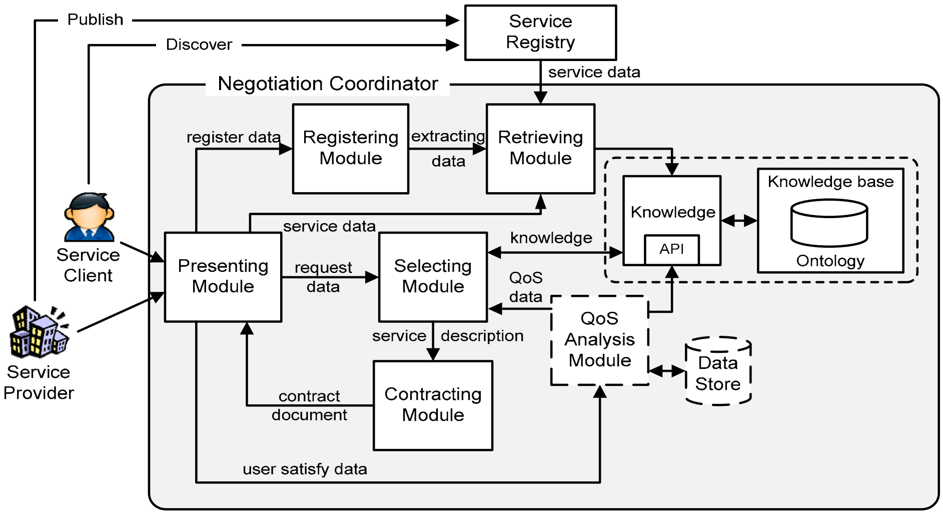
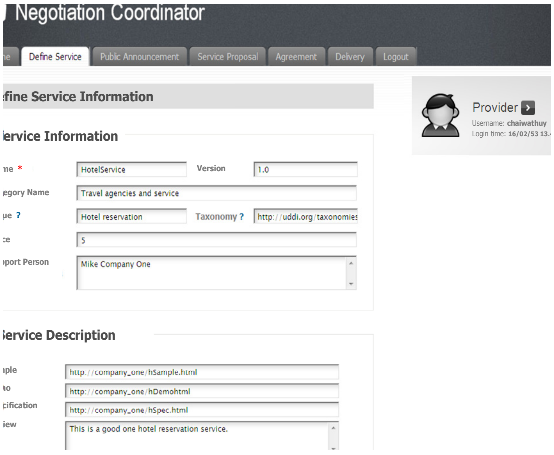


Fig. 4: The system architecture.

5. Implementation and Results

5.1. System Implementation

The system was implemented with object-oriented programming in a Model-View-Controller (MVC) style, corresponding to the system architecture. Thus, the system was divided into three parts : (1) data and knowledge management as a model, (2) operation processing as a controller, (3) presentations and user interfaces as a view. The XML (Extensible Markup Language) schema was adopted for knowledge representation. Figure 5 and 6 show some examples of menus in the user interface for defining a service by a provider and defining requirements by a client respectively.



fine Service Information

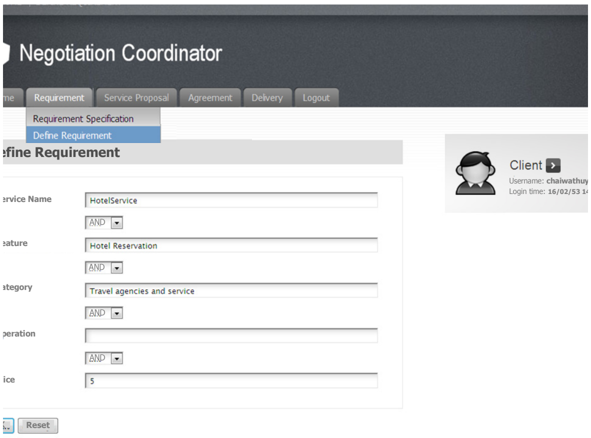
ervice Information

me * HotelService
Version 1.0
egory Name Travel agencies and service
ue ? Hotel reservation
Taxonomy ? http://uddi.org/taxonomies
ce 5
port Person Mike Company One

ervice Description

iple http://company_one/hSample.html
no http://company_one/hDemohtml
cification http://company_one/hSpec.html
iew This is a good one hotel reservation service.

Fig. 5: The window for providers.



efine Requirement

ervice Name HotelService
AND
eature Hotel Reservation
AND
ategory Travel agencies and service
AND
peration
AND
ice 5

Fig. 6: The window for clients.

5.2. Case Study and Evaluation

A tourist application was selected as a case study. Two providers were assigned as Company-1 and Company-2 to provide services for this application. Company-1 provided two services: (1) HotelService and (2) AirTicket while Company-2 also provided two similar services: (1) BookingHotel and (2) FlightBooking. A service registry was developed by using an open-source Apache jUDDI that conforms to the web-service standards.

The system was evaluated in our laboratory environment corresponding to the proposed model in Figure 1. Firstly, service requests from Company-1 and Company-2 were tested via SOAP requests. Then, the service registry was tested and evaluated in terms of readiness of use, correctness of data recording and searching. Finally, the system was tested and evaluated whether it could work as an electronic negotiation coordinator as the proposed process in Figure 2. The results satisfy the objectives of the study.

6. Conclusion

In this paper, the model and process for an electronic negotiation coordinator in service-oriented environments were proposed. Then, the system was designed, implemented, tested and evaluated in our laboratory environment. Future work, we should look for more details about ontology for knowledge representation as well as the issues about security.

7. References

[1] M.N. Huhns and M.P. Singh. Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing. January-February 2005, pp. 75-81.

[2] M.P. Papazoglou. Service-Oriented Computing: Concepts, Characteristics and Directions. Proc. Of the 4th International conference on Web Information Systems Engineering (WISE'03). 2003.

[3] R.L. McIntosh. Open-Source Tools for Distributed Device Control with A Service-Oriented Architecture. Journal of the Association for Laboratory Automation. 2004, 9(6): pp. 404-410.

[4] J.L. Rosa, N. Hormazabal, S. Aciar, G. Lopardo, A. Trias, and M. Montaner. A Negotiation Style Recommender Based on Computational Ecology in Open Negotiation Environments. IEEE Transactions on Industrail Electronics. December 2009, pp. 1-13.

[5] A. Elfatatry and P. Layzell. Negotiating in Service-Oriented Environments. Communications of the ACM. August 2004, 47(8): pp. 103-108.

[6] J.D. Bruijn, D. Fensel, M. Kerrigan, U. Keller, H. Lausen, and J. Scicluna. Modeling Semantic Web Service Modeling Language. Springer: Springer-Verlag Berlin Heidelberg. 2008.

[7] C. Phaiboonphuthiphong and A. Pohthong. A Negotiation Framework for Service-Oriented Architecture by Enhancing CMMI. *Proc. of the 2nd National Conference on Information Technology(NCIT2008), Bangkok, Thailand*. 6-7 November 2008, pp. 490-499.