

SOA에서 서비스 분류 기준에 따른 매칭기법 설계

최미숙*, 이서정**

요약

분산 컴퓨팅 기술이 발전하면서, 기존의 클라이언트/서버 기능을 기존의 개념에서 벗어나, 이 기종 환경사이에도 원하는 정보를 공유하기 위해 서비스 지향 아키텍처의 개념이 등장하였다. SOA가 성공하기 위해서는 비즈니스 계층의 서비스와 애플리케이션을 연결하는 기술이 중요한 이슈 중의 하나이다.

본 논문에서는 서비스를 종적 그리고 횡적으로 분류하고, 이에 따라 계층 사이의 서비스를 매칭하는 기법을 소개한다.

Design of Service Matching with Vertical and Horizontal Classification for SOA

Misook Choi*, Seojeong Lee**

Abstract

With the evolution to distributed computing technology, the concept of conventional client/server function has been advanced to Service-Oriented Architecture, which can share the information between heterogeneous systems as services. To success Service-Oriented Architecture, the conjunction between business layer and application layer is one of the important techniques. In this paper, we classify services into two views, vertical and horizontal, and introduce matching technique with service classification for SOA.

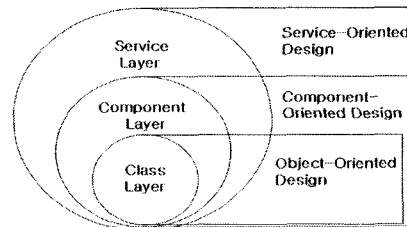
Keywords : Service-Oriented Architecture, Service Classification, Matching Technique between Services, Service Specification

1. 서론

SOA(Service Oriented Architecture)는 기존의 CORBA와 DCOM등 분산 컴포넌트 개념의 기술 기반으로 최근 몇 년간 큰 이슈가 되어왔다. 분산 컴퓨팅의 기본 조건인 개발 언어에 상관없이 서비스를 제공하고, 플랫폼 독립적인 컴포넌트를 제공하며, 쉽고 간단한 서비스의 유지보수를 실현하는 데 있다. 하지만, CORBA와 DCOM등은 플랫폼 독립적인 조건을 달성하는

데는 역부족이다. 이에 표준화된 인터페이스와 XML을 이용하여 개발 언어에서 독립적이고, 메시지 프로토콜을 이용한 약 결합으로 쉽고 간단한 유지보수를 제공하며, 플랫폼에 보다 유연한 웹 기반 개념으로 SOA가 등장하였다[1][2].

기존의 객체지향 개념과 컴포넌트 지향 개념을 거치면서 비즈니스 서비스의 개념은 정보시스템의 개발과정에 더 큰 비중을 차지하게 되었다. (그림 1)은 각 정보시스템 내부 계층별 요구되는 설계 개념과 범위의 차이를 보여준다[3].



(그림 1) 정보시스템 계층별 설계

※ 제일저자(First Author) : 최미숙

접수일자:2007년05월17일, 심사완료:2007년06월20일

* 우석대학교 연구교수

khc67_kr@hanmail.net

** 한국해양대학교 컴퓨터·제어·전자통신공학부 조교수

▣ 이 논문은 2006년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2006-353-D00029)

(그림 1)의 서비스지향설계와 컴포넌트지향설계 범위의 차이는 '서비스'에 있다. 즉, SOA는 비즈니스 수준의 서비스를 효율적으로 제공하기 위해 컴포넌트를 개발하고 관리하는 것으로, 기존 애플리케이션의 서비스를 조합함으로써 새로운 애플리케이션 서비스를 구현하는 아키텍처이다[4]. 이를 실현하기 위해 요구되는 가장 중요한 일 중의 하나는 이 기존 분산 환경에서 사용자들의 다양한 서비스의 요구에 대응하여 적합한 서비스를 제공할 수 있도록 서비스와 컴포넌트 계층간 갭(gap)을 최대한 줄이는 것이다[5].

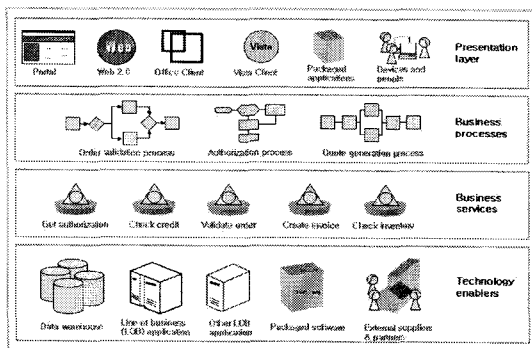
서비스는 재사용 가능한 기능과 이들 사이의 작업흐름(workflow)으로 구성된다. 작업흐름은 상황에 따라 변할 수 있고, 서비스도 포함하는 기능이 변화하거나 새로운 서비스가 생성될 수 있다. 이를 위해, 서비스를 실제 구현하는 컴포넌트 계층과의 갭을 최소화할 필요가 있으며, 이는 두 계층 사이 서비스와 컴포넌트의 매칭이 효율적으로 이루어지는가에 달려있다. 본 논문에서는 SOA 서비스 계층을 위한 서비스 분류 기준에 따른 논리적 매칭 기법을 설계한다.

2장에서 기존의 서비스 계층 정의와 계층간 서비스의 매칭기법을 살펴보고, 3장에서 서비스를 동적 및 횡적으로 분류하고, 분류에 따른 매칭기법과 효율적 매칭을 위한 서비스의 명세를 제안한다.

2. 관련연구

2.1 서비스 계층

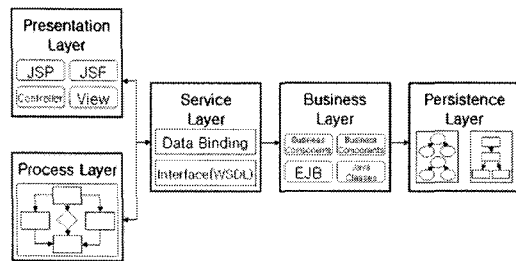
기존의 분산 시스템에서는 서비스의 제공을



(그림 2) 4-tier 아키텍처의 예

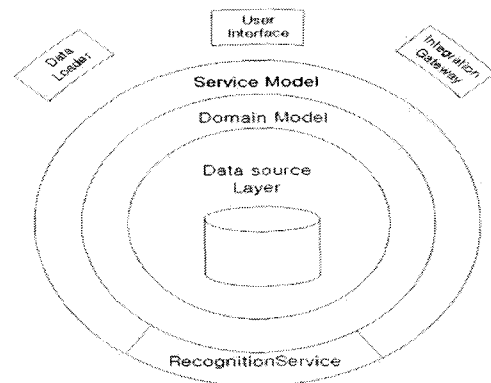
특정 브로커에서 취합하거나 서버에 클라이언트가 직접 요청을 하는 방식으로 제공되는 방식이라면, SOA에서는 서비스 계층을 별도로 구성한다는 점에서 차이가 있다. (그림 2)는 프리젠테이션계층, 비즈니스프로세스와 가용기술 사이에 비즈니스서비스를 두어 4-tier로 구성된 아키텍처이다[6].

한편, 서비스계층은 데이터 바인딩과 WSDL을 이용한 인터페이스 기능을 하여, 프리젠테이션계층과 프로세스계층과 연동하여 서비스를 제공하는 아키텍처를 구성하기도 한다. (그림 3)의 비즈니스계층에서는 비즈니스 컴포넌트, EJB 그리고 클래스를 포함하도록 설계한 경우다[7].



(그림 3) 서비스 지향 모델의 예

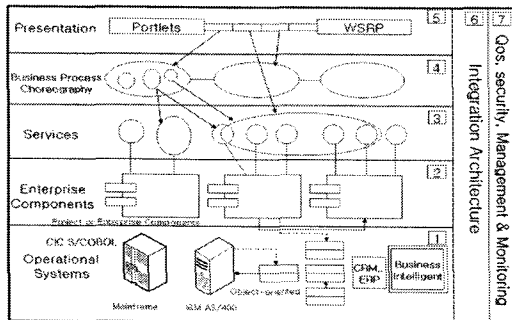
일반적으로 엔터프라이즈 응용 아키텍처에서 서비스 계층은 (그림 4)에서와 같이 도메인 모델과 사용자인터페이스, 게이트웨이를 통해 서비스를 제공하는 역할을 한다는 개념에서 비롯되었다[8].



(그림 4) 엔터프라이즈 애플리케이션 아키텍처

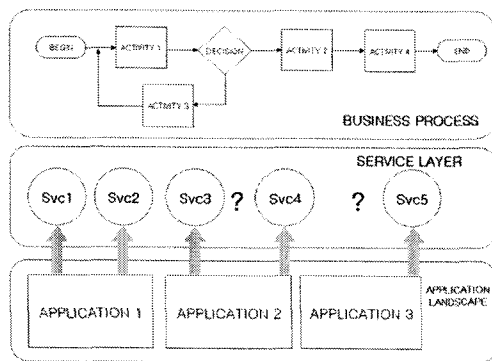
2.2 매칭기법

서비스지향 모델의 서비스는 비즈니스 프로세스와 컴포넌트를 서로 연결하여 생성한다. 즉, 어떤 비즈니스 프로세스를 어떤 컴포넌트가 수행할 것인가에 대한 매칭이 정의되어 있어야 한다. (그림 5)는 이러한 상황을 개념적으로 보여주는 서비스지향 아키텍처이다. 특정 비즈니스 프로세스를 수행하기 위해 하나의 컴포넌트가 매칭될 수도 있고, 둘 이상이 매칭될 수도 있다 [9].



(그림 5) 서비스지향 모델링의 예

(그림 6)은 비즈니스 프로세스와 컴포넌트의 매칭을 좀 더 구체적으로 보여준다. 서비스계층에서는 비즈니스 프로세스의 액티비티(Activity)와 디시전(Decision)을 수행할 수 있는 해당 애플리케이션에 대한 매칭을 정의한다. 새로운 비즈니스프로세스가 발생한 경우, 기존 애플리케이션에 매칭할 수 없으면 새로운 애플리케이션을 생성하는 방식이다[10].



(그림 6) 비즈니스프로세스, 애플리케이션 그리고 서비스의 관계

지금까지의 매칭 방법은 비즈니스계층의 단위 작업(chore) 또는 액티비티를 애플리케이션계층의 컴포넌트 인터페이스 또는 애플리케이션에 연결한다는 공통점을 갖는다. 이는 비즈니스계층의 작업흐름도(choreography)나 액티비티가 변경되어도 애플리케이션계층의 재사용성을 높일 수 있고, 필요한 경우에만 새로운 애플리케이션을 생성하므로 유지보수 비용도 줄일 수 있다. 하지만, 비즈니스계층의 단위작업이나 액티비티가 갖는 다양한 특성과 규모에 대한 구체적인 방법에 대해서는 많은 연구가 필요하다.

3. 서비스 분류기준에 따른 매칭 기법 설계

본 논문에서는 2장에서 소개한 비즈니스계층의 단위작업이나 액티비티와 애플리케이션계층의 컴포넌트와의 연결을 매칭대상으로 정의하고, 서비스의 분류기준에 따른 매칭을 개념적으로 보여준다.

3.1 서비스의 분류

서비스는 종적(vertical) 그리고 횡적(horizontal)으로 분류할 수 있다. 종적분류는 입자구성의 복잡성에 따라 분류하며, 횡적분류는 계층에 따라 분류한다.

종적분류를 위해서 [11]의 중개 서비스와 프로세스 서비스를 도입하여, 네 가지로 구분한다. [11]의 기본 서비스와 공개서비스는 본 논문에 적합하지 않다. 기본서비스는 의미는 비슷하지만 매칭의 개념에 차이가 있고, 공개서비스는 매칭대상이기 보다는 서비스의 특징이라고 할 수 있으므로 제외한다.

- 단위 서비스(unit service)
- 중개 서비스(intermediary service)
- 프로세스 서비스(process-centric service)
- 복합 서비스(complex service)

이들 서비스 유형 사이의 특징을 요약하면 다음과 같다.

단위 서비스(unit service)는 SOA의 기반이 되는 서비스로서, 애플리케이션계층의 하나의 컴

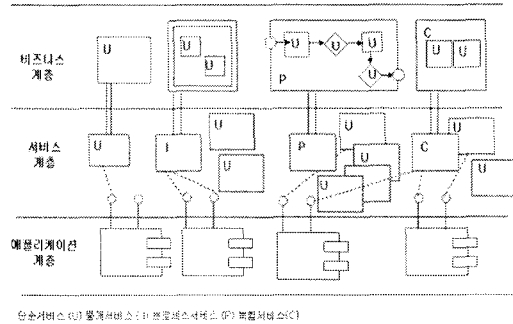
포넌트와 매칭된다. 중개 서비스(intermediary service)는 레거시 시스템과 연결하는 게이트웨이(gateway) 기능, 다른 서비스의 시그니처(signature), 어댑터(adapter), 하나 이상의 기존 서비스를 통합해서 다른 뷰를 제공하는 퍼사드(façade) 등의 기능을 제공하는 서비스이다. 프로세스 서비스(process service)는 비즈니스 프로세스(business process)를 캡슐화 한다. 복잡한 비즈니스 룰을 담당한다. 복합 서비스(complex service)는 두 개 이상의 컴포넌트를 결합하여 간단한 로직에 의해 조합한 결과를 제공하는 서비스이다. 프로세스 서비스처럼 복잡한 룰을 적용하는 것이 아니라, 포함된 컴포넌트에 대해 산술, 논리 및 관계 연산 등에 의해 생성된 결과를 제공하는 서비스이다.

횡적 분류는 비즈니스계층(business layer), 서비스계층(service layer) 그리고 애플리케이션계층(application layer)으로 분류한다. 비즈니스계층은 사용자에게 최종적으로 서비스를 전달하는 계층이고, 서비스계층은 애플리케이션계층과 비즈니스계층을 연결한다. 비즈니스계층의 서비스는 단위비즈니스이고, 서비스계층은 단위서비스, 그리고 애플리케이션계층의 단위서비스는 컴포넌트로 정의한다. 컴포넌트를 애플리케이션의 단위서비스로 취급하기 위해, 재사용 가능한 단위 컴포넌트로 정의한다. 계층의 분류는 기존의 연구와 비슷하지만, 매칭단위를 도입했다는 점에서 그 내용은 차별된다.

3.2 매칭 기법

비즈니스계층의 서비스는 컴포넌트계층의 하나 또는 둘 이상으로 수행할 수 있으며, 이는 서비스계층의 서비스를 통해 매칭된다. 비즈니스계층의 단위서비스는 그 종적 분류에 따라, 단위, 중개, 프로세스 및 복합 서비스로 분류하며, 서비스계층에서 (그림 7)과 같이 분리된다.

하나의 단위서비스는 서비스계층의 하나의 서비스에 매칭되고, 두 개 단위서비스를 포함한 중개서비스의 경우, 세 개의 서비스로 분리된다. 하나의 서비스가 증가되는 이유는 포함된 단위 서비스마다 하나씩 서비스를 생성하고, 중개를 위해 단위서비스를 묶는 서비스를 별도로 두기 때문이다. 이런 방식으로, 4개의 단위서비스의 프로세스로 구성되는 프로세스서비스의 경우, 다



(그림 7) 계층간 서비스 매칭의 개념

섯 개의 서비스로 분리된다. 복합서비스의 경우에도 포함되는 단위서비스의 수보다 한 개 많은 서비스로 분리된다.

3.3 서비스 명세

계층간 서비스의 매칭을 원활히 하기 위해서는 적절한 서비스 명세가 필요하다. <표 1>은 본 논문에서 제안하는 서비스 분류를 표현할 수 있는 서비스 명세로써, [12]의 서비스 명세를 참조했다.

<표 1> 서비스 명세

Name	findCreditRegistrationPerson
Service Class	NewAccountController
Usage	Finds Persons in the BKR system
Input	creditRegistrationPerson
Precondition	Collection of creditRegistrationPerson must have agreed on checking BKR
Postcondition	NA
Behavior	1. determine the necessary data 2. sends the data to BKR 3. receives the results from BKR 4. return the information
Service Involved	creditRegistrationPerson

서비스 명세의 각 항목을 간단히 설명하면 다음과 같다.

- Name: 서비스명
- ServiceClass: [단위|중개|프로세스|복합]
- Usage: 용도를 간단히 기입

- Input: 입력값
- Output: 출력값
- Precondition: 선수조건
- Postcondition: 후수조건
- Behavior : 용도의 사용예 또는 내부액션
- ServiceInvolved : [중개|프로세스|복합]의 경우, 포함하는 단위서비스

본 논문에서 제안한 서비스 명세는 서비스의 분류 및 서비스에 포함되는 단위서비스를 명세한다는 점에서 기존의 서비스 명세와 차이가 있으며, 소프트웨어 모델링도구에서 제공하는 환경으로 구현할 수 있다.

4. 결론 및 향후과제

본 논문은 SOA를 위한 서비스를 종적 및 횡적으로 분류하고, 이에 대한 매칭의 개념과 서비스 명세에 관한 연구이다. 종적으로 단위 서비스(unit service), 중개 서비스(intermediary service), 프로세스 서비스(process-centric service), 복합 서비스(complex service)로 분류하며, 횡적으로는 비즈니스계층(business layer), 서비스계층(service layer) 그리고 애플리케이션계층(application layer)으로 분류한다. 계층의 분류는 기존의 연구와 비슷하지만, 매칭단위를 도입했다는 점에서 그 내용은 차별된다. 서비스의 명세는 본 논문에서 제안한 서비스 명세는 서비스의 분류 및 서비스에 포함되는 단위서비스를 명세한다는 점에서 기존의 서비스 명세와 차이가 있다.

향후 과제로는 서비스계층과 서비스 명세를 소프트웨어 모델링도구에서 제공하는 환경으로 구현하고, 실제 응용환경에서 매칭을 시도하여 제안한 기법을 보완하고 확장하는 연구를 진행할 예정이다.

참고문헌

[1] 권수갑, SOA개념과 동향, 전자부품연구원, 2005
 [2] Roy W. Schulte, Yefim V. Natis, "Service Oriented Architecture," Gartner Group, SSA Research Note SPA-401-068, 1996
 [3] Olaf Zimmermann, Pal Krogdahl and Clive Gee, "Elements of Service-Oriented Analysis and Design," I

BM http://www-128.ibm.com/developerworks/web_services/library/, 2004

[4] Thomas Erl, Service-Oriented Architecture: A field Guide to Integrating XML and Web Service, Prentice-Hall, 2004
 [5] Sjoerd Michels, "SOA Suite build, Depolymnt and Test Automation," 2007, <http://technology.amis.nl/blog/?p=1645>
 [6] Moin Moinuddin, "Building Distributed Applications -An Overview of Service-Oriented Architecture in Retail," Microsoft Corporation, 2007
 [7] Debu Panda, "An Introduction to Service-Oriented Architecture from a Java Developer Perspective," O'Reilly, 2005
 [8] Martin Fowler, "Patterns of Enterprise Application Architecture," Addison Wesley, 2002
 [9] Ali Arsanjani, "How to identify, specify, and realize services for your SOA," SOA and Web services Center of Excellence, IBM, 2004
 [10] Srikanth Inaganti & Gopala Krishna Behara, "Service Identification: BPM and SOA Handshake," BPTrends, 2007
 [11] Kragfzig 외, Enterprise SOA, Service-Oriented Architecture Best Practices, Prentice Hall, 2005
 [12] Alcedo Coenen, "An SOA Case Study: Agility in Practice," SOA Magazine, 2006

최미숙



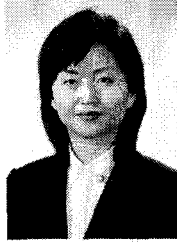
1990년 :전북대학교(이학사)
 1994년 :숙명여자대학교 일반대학원 컴퓨터학과(석사)
 2002년 :숙명여자대학교 일반대학원 컴퓨터학과(박사)

1995년~1999년 : 나주대학 소프트웨어 개발과 전임교수

2004년~2006년 : 우석대학교 컴퓨터공학과 초빙교수

2006년~현재 : 우석대학교 연구교수

관심분야 : CBD 방법론, 소프트웨어 메트릭, 서비스 지향 아키텍처



이서정

1989년 : 숙명여자대학교 전산학과
(학사)

1991년 : 숙명여자대학교 대학원
전산학(석사)

1998년 : 숙명여자대학교 대학원
전산학(박사)

1998년~2003년 : 동덕여자대학교 정보학부 강의교수

2004년 : 숭실대학교 연구교수

2005년~현재 : 한국해양대학교 컴퓨터·제어·전
자통신공학부 조교수