

SOA 기반의 웹 서비스 컴포넌트 개발에 관한 연구

박동식[†], 신호준^{**}, 김행곤^{***}

요 약

웹 서비스는 인터넷을 통해 기업 상호간에 비즈니스를 연결할 수 있도록 하는 것으로, 서비스 구축의 비용 감소와 속도 증진이 가능하다. 또한, 다른 영역과의 통합이 쉽게 가능하며, 컴포넌트 기반으로 개발할 경우 업데이트나 수정이 재사용성과 대체성을 제공함으로써 용이하다. 본 논문에서는 서비스지향 아키텍처 상에서 공급자 측면에서 구현되는 웹 서비스를 신뢰성과 민첩성을 보장하기 위해 컴포넌트 기반으로 개발하고자 한다. 이를 위해서 구성되는 컴포넌트를 효율적으로 통합하기 위한 아키텍처를 제안하고, 개발 프로세스를 서술한다. 또한, 컴포넌트를 기반으로 웹 서비스를 개발하기 위해 웹 서비스의 구조를 논리적으로 계층화하여 각 계층에서의 기능을 정의하였으며, 논리적 계층을 기반으로 아키텍처를 제시한다. 이들 내부는 크게 Facade 와 Backside 컴포넌트로 구성하며, 웹 서비스 기능을 가지고 있는 Facade 컴포넌트를 개발하기 위한 프로세스와 메일링 웹 서비스를 사례로 제시한다. 이를 통해 웹 서비스의 생산비용과 개발시간의 절감을 기대할 수 있으며, 컴포넌트를 기반으로 웹 서비스를 구성하여 재사용성과 대체성에 대한 신뢰성 향상을 가져온다.

A Study on the Development Web Services Component Based Service Oriented Architecture

Dong-Sik Park[†], Ho-Jun Shin^{**}, Haeng-Kon Kim^{***}

ABSTRACT

Web service should be to connect business between enterprise through the Internet, promotion of construction speed and decrease of development expense of service construction are possible. Also, unification with other domain is possible easily, and update or correction is easy by offering reusability and replaceability through component based development. In this paper, We suggest development process to build architecture and this to integrate consisted component efficiently to develop web service that is embodied in supplier side on service oriented architecture(SOA). The suggest architecture to integrate component that is consisted for this efficiently, and describes development process. So that component develops web service to base structure of web service because do command stratification logically function in each hierarchy define, and presents architecture based on logical hierarchy. The web services consist of Facade and Backside component; The Facade component have web service functions. We describe process that develop to Facade component and present mailing web services as case study. It can be decrease production cost and development time. The web service based on component will improve reliability for reuseability and replaceability.

Key words: Web Services(웹 서비스), SOA(서비스 지향 아키텍처), CBD(컴포넌트 기반 개발), Facade Component(패케이드 컴포넌트)

※ 교신저자(Corresponding Author) : 박동식, 주소 : 경상북도 경산시 하양읍 금락리 330번지(712-702), 전화 : 053) 850-2743, FAX : 053)850-2740, E-mail : sabujak@cu.ac.kr
접수일 : 2004년 2월 20일, 완료일 : 2004년 5월 28일

[†] 준회원, 대구가톨릭대학교 대학원 컴퓨터정보통신 석사과정

^{**} 대구가톨릭대학교 컴퓨터정보통신공학부 연구원
(E-mail : component@cu.ac.kr)

^{***} 정회원, 대구가톨릭대학교 컴퓨터정보통신공학부 교수
(E-mail : hangkon@cu.ac.kr)

1. 서 론

IT 영역에서 여러 기업들이 수익성 확보를 위해 각 기업들이 협력을 하고자 노력하고 있지만, 각자의 기업 고유의 업무 프로세스 즉, 상이한 플랫폼으로 인한 통합의 어려움이 많다.

웹 서비스는 플랫폼, 특정언어, 특정 하드웨어로부터 종속성을 탈피한 모델이다. 즉, 공통된 프로세스인 비즈니스 로직을 일정한 공급자가 컴포넌트 모델로 구현하고 그것을 각각의 사용자가 표준 프로토콜, XML(Extensible Markup Language), SOAP(Simple Object Access Protocol), WSDL(Web Services Description Language), UDDI(Universal Description Discovery and Integration)를 통하여 서비스를 받음으로써 자연스럽게 기업 간에 협력 발생을 유도한다[1,2].

최근 웹 서비스는 인터넷을 통해 기업 상호간에 비즈니스를 연결할 수 있도록 하는 것으로 IT 전문가들이나 관련 업계에 한정되지 않고 많은 분야에서 이슈가 되고 있다. 가트너 그룹에서는 2005년까지 중견 및 대기업의 90%가 웹 서비스를 채택할 것으로 예상한다. 이것은 웹 서비스의 구축이 개발비용의 감소와 구축속도 증진이 가능함으로써, 기업의 빠른 요구사항 변경에 대한 민첩성 향상을 보장하기 때문이다.

본 논문에서는 웹 서비스 개발을 위해 서비스 지향 컴포넌트를 제공하고자 한다. 이를 위해 서비스지향 아키텍처 상에서 공급자 측면에서 구현되는 웹 서비스를 개발하기 위해서 컴포넌트를 효율적으로 통합할 수 있는 아키텍처를 제안하고자 한다. 이를 위해 웹 서비스의 구조를 논리적으로 계층화하고, 논리적 계층 구조를 기반으로 서비스 지향 아키텍처를 정의한다. 아키텍처 상의 컴포넌트를 웹 서비스를 제공하기 위한 Facade 컴포넌트와 이를 지원하기 위한 Backside 컴포넌트로 정의한다. 또한, 웹 서비스를 직접 지원하고 상호 운용되는 Facade 컴포넌트를 개발하기 위한 프로세스를 제안하며, 서비스 지향 분석, 설계 및 사례를 제시한다. 본고의 구성은 다음과 같다. 2장에서는 서비스 지향 아키텍처의 구성과 구성요소들 간의 역할에 대해서 살펴보고, 3장에서는 서비스 지향 아키텍처를 컴포넌트로 구성하기 위한 논리적 계층과, 계층에서 핵심이 되는 Facade 컴포넌트를 개발하기 위한 개발 프로세스를 기술한다. 4장에서는 제시된 개발 프로세스를 사용하여 메일링

웹 서비스의 구현을 통해 사용 예를 제시한다. 마지막으로 5장에서 결론을 맺는다.

2. SOA(Services Oriented Architecture)

SOA는 서비스 지향 활동을 위한 기술적 기반이며, 원격으로 객체나 서비스에 접근하는 능력, 서비스 검색, 동적 바인딩 그리고 낮은 결합력을 가지는 특징을 가진다.

그림 1과 같이 아키텍처 내부의 역할들은 서비스 요청자(Service Requestor), 리파지토리(Discovery Agencies), 서비스 제공자(Service Provider)로 나누어져 있다. 또 SOA에서는 Publish, Find, Bind 등의 세 가지 오퍼레이션을 정의하고 있다. 서비스 제공자는 서비스의 제공을 위해 Publish를 통해 리파지토리에 서비스 제공의 위치와 정보들을 게시한다. 서비스 요청자는 Find를 통해 리파지토리에서 원하는 서비스를 검색을 하고, 검색된 서비스는 Bind를 통해 서비스 제공자에게 서비스를 제공받는다[3,4].

SOA의 기본적인 구성요소에는 SOAP, XML, UDDI, WSDL이 있다. SOAP은 정보를 실어 전달하는데 쓰이는 플랫폼 중립적인 표준 메시지 프로토콜이며 메시지의 내용은 XML로서 표현된다. WSDL은 서비스 사용자가 서비스 제공자가 공개한 인터페이스의 호출 방법을 설명하는 문서로서 XML로 작성되어 있다. 이 문서는 UDDI에 등록되며, 문서 안에는 서비스 요청자와 서비스 제공자 양자간에 전달되는 파라미터의 이름들, 등록된 웹 서비스가 실제 존재하는 URL(Uniform Resource Locator) 등의 정보가 기술되어 있다.

UDDI는 서비스 제공자가 웹 서비스를 등록하고 서비스 요청자는 등록된 웹 서비스를 찾기 위한 프레

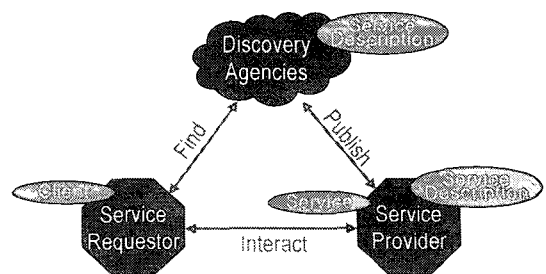


그림 1. Service Oriented Architecture

임워크이다. 그러나 꼭 UDDI를 거쳐야만 되는 것은 아니고 서비스 제공자의 URL을 이미 알고 있다면 서비스 요청자가 바로 서비스 제공자를 호출할 수도 있다.

3. 컴포넌트 기반 웹 서비스 시스템 개발

3.1 아키텍처

서비스 지향 아키텍처를 컴포넌트 기반으로 구성하기 위해서 사용자, 서비스, 운영, 지원 4개의 논리적 계층 구조를 그림 2와 같이 구성한다.

사용자 계층은 사용자가 사용하는 단말장치와 외부 시스템이며 사용자화면과 같은 프레젠테이션 부분이 포함된다.

서비스 계층은 웹 서비스가 이루어지는 부분이며, 사용자 계층을 관리할 수 있는 비즈니스 로직을 포함하며, 사용자 계층에게 서비스를 제공한다. 이 계층에 본 논문에서 개발하고자 하는 서비스 지향 컴포넌트인 Facade 컴포넌트를 위치시킨다. Facade 컴포넌트는 SOAP와 WSDL로 명세 되어 웹 서비스가 가능하다. 그림 3에서와 같이 기능적으로 Core와 Flow로 분류할 수 있다. Core 컴포넌트는 하위 컴포넌트 또는 다른 웹 서비스를 통합하여 하나의 웹 서비스를 제공하는 기능을 가졌으며, Flow 컴포넌트는 다른 서비스들과의 연동을 제공한다. Backside 컴포넌트는 그림 2의 논리적 계층에 해당하는 컴포넌트들로써 하나의 웹 서비스를 제공하기 위한 구성요소로 정의한다.

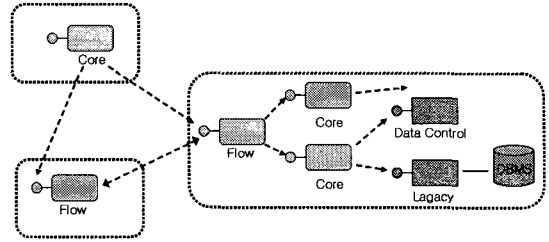


그림 3. Facade / Backside 컴포넌트의 개념

운영 계층은 실제 비즈니스 로직을 포함하는 부분이며, Legacy 시스템 또는 Backside 컴포넌트, 패키지 컴포넌트로 구성된다. Backside 컴포넌트는 CBD (Component Based Development) 방법을 통해 개발된 공용 컴포넌트 또는 COTS(Commercial off the shelf) 컴포넌트를 말한다.

지원 계층은 운영 계층의 비즈니스 로직이 요청하는 데이터에 대한 접근 또는 실제 적인 데이터를 제공한다.

그림 4는 본 논문에서 웹 서비스의 논리적 계층구조를 기반으로 구성한 컴포넌트 기반의 웹 서비스 아키텍처이다. 내부의 구조는 논리적 계층 구조와 매핑 된다. 시스템은 하나 이상의 어플리케이션으로 구성될 수 있다. 하나의 어플리케이션이 전체 시스템을 구성할 경우 서비스간의 흐름을 제어할 필요가 없으므로 Core Facade 컴포넌트를 통해 서비스를 제공한다. Flow Facade 컴포넌트는 시스템의 내부가 2개 이상의 어플리케이션으로 구성되었을 때, 이들 간의 통합과 흐름을 제어한다.

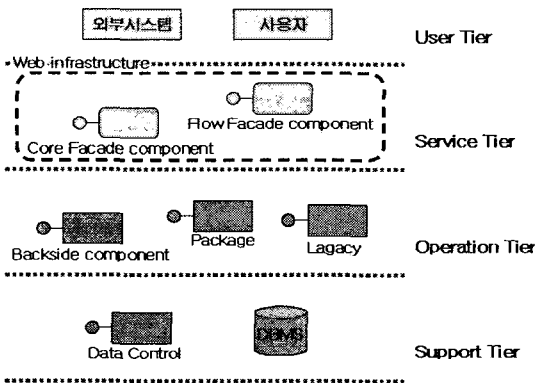


그림 2. 컴포넌트 기반 웹 서비스 아키텍처의 논리적 계층 구조

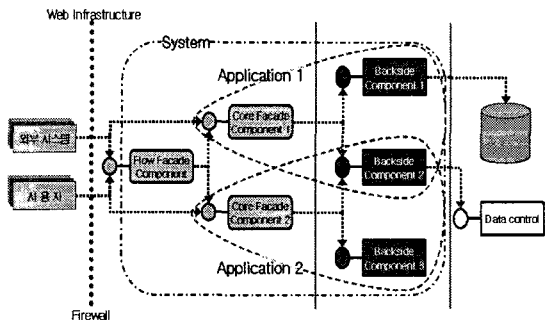


그림 4. 컴포넌트 기반의 웹 서비스 아키텍처

3.2 웹 서비스 지원 컴포넌트 분석 및 설계

앞에서 제시한 아키텍처 상의 웹 서비스 내부의

실질적인 웹 서비스기능을 담당하는 Facade 컴포넌트를 개발하기 위해서 모델 기반의 개발 방법을 사용하여 그림 5와 같이 웹 서비스 지원 컴포넌트 개발 프로세스를 제안한다.

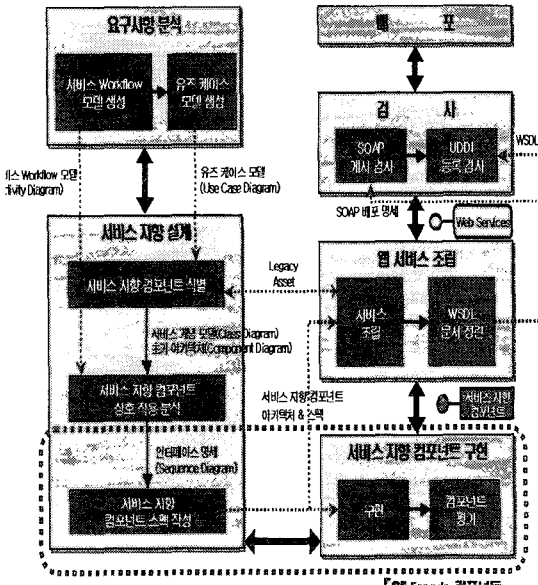


그림 5. 컴포넌트 기반 웹 서비스 개발 프로세스

3.2.1 요구사항 분석

요구사항 분석에서는 사용자 시스템의 요구사항에 대한 영역, 비전, 목표를 정의한다. 또한 프로세스와 프로세스에서 생성되거나 사용되는 자원을 중심으로 분석한다. 이를 통해 시스템을 구축하기 위한 프로세스, 자원, 관리조직을 정의한다.

시스템 내부의 프로세스들의 기능을 파악하기 위하여 UML의 Use Case Diagram을 생성한다. 각각의 프로세스는 하나 또는 그 이상의 흐름을 가지고 있으며, 서비스를 제공하고, 어떻게 서비스를 제공할지 정의한다. 이를 위해, 서비스 Workflow 모델 생성 단계에서는 프로세스의 흐름을 UML의 Activity Diagram을 통해 생성한다. 자원은 각 프로세스에서 생성되거나 사용되는 요소이다.

3.2.2 서비스 지향 설계

전체 개발 단계 중에서 가장 중요한 단계로써, 요구사항 분석 단계에서 생성된 산출물을 사용하여 웹 서비스를 설계하는 단계이다. 통합 구조, Facade 컴포넌트, Backside 컴포넌트와 같이 3가지 요소를 통

하여 서비스 지향 설계를 하게 된다. 통합 구조는 두 컴포넌트를 통합할 수 있는 아키텍처를 제공하며 이는 Component Diagram을 통해 초기 아키텍처 모델이 된다. Facade 컴포넌트는 일반적인 컴포넌트가 웹 서비스를 할 수 있도록 웹 서비스 가능한 인터페이스를 제공하는 컴포넌트이며, Backside 컴포넌트는 기존에 개발되거나 일반적인 컴포넌트 개발 방법론으로 만들어진 일반적인 컴포넌트이다.

(1) 서비스 지향 컴포넌트 식별

웹 서비스가 제공하는 기능을 파악하기 위해 생성된 Use Case 모델을 통하여 컴포넌트를 식별한다. Backside 컴포넌트는 실제적이고 독립적인 기능을 중심으로 식별하며, 보조적인 자원들과의 관계를 Class Diagram을 사용하여 보여준다. Facade 컴포넌트의 식별은 시스템에서 웹 서비스를 제공하기 위한 컴포넌트를 판단하고, 이는 Backside 컴포넌트를 통합하여 웹 서비스를 지원하는 인터페이스를 제공한다. 식별된 컴포넌트를 통합하기 위한 구조인 초기 아키텍처(Component Diagram)를 생성한다. 아키텍처 상의 컴포넌트를 서비스 지향 컴포넌트와 일반적인 컴포넌트로 식별하고, 서비스를 제공하기 위해 서비스 개념 모델(Class Diagram)을 생성한다.

(2) 서비스 지향 컴포넌트 상호 작용 분석

요구사항 분석 단계에서의 산출물인 서비스 Workflow 모델과 전 단계의 초기 아키텍처와 서비스 개념 모델을 사용하여 아키텍처상의 컴포넌트 사이의 상호작용을 분석하여 Sequence Diagram을 생성함으로써 컴포넌트 사이의 인터페이스와 서비스 지향 컴포넌트가 외부로 제공해야 하는 인터페이스를 식별한다.

(3) 서비스 지향 컴포넌트 명세 작성

Backside 컴포넌트는 기존의 개발된 컴포넌트 또는 일반적인 컴포넌트 개발 방법론을 사용하여 개발된 컴포넌트이므로 컴포넌트에 대한 명세를 작성할 필요가 없다.

서비스 지향 컴포넌트 명세 생성단계는 식별 단계에서 식별된 Facade 컴포넌트에 대한 명세를 작성하는 단계로써, 정련된 컴포넌트 사용 계약과 실제화 계약에 대한 명세를 정의한다. 사용계약은 인터페이스 명세인 Sequence Diagram을 사용하여 정의되며, 실제화 계약은 컴포넌트 명세로 정의된다. 컴포넌트

명세에는 인터페이스가 구현되는 방식에 대한 제약 사항을 포함한다. 또, 초기 아키텍처를 정련하여 서비스 지향 컴포넌트 아키텍처를 생성한다.

3.2.3 서비스 지향 컴포넌트 구현

서비스 지향 컴포넌트 아키텍처에서 식별된 Facade 컴포넌트를 구현하는 단계이다. Facade 컴포넌트와 Backside 컴포넌트와의 상호 작용은 전 단계에서 생성된 인터페이스 명세를 통하여 구현하며, 웹 서비스를 지원하기 위한 Facade 컴포넌트의 인터페이스는 웹 서비스 표준인 SOAP, XML을 사용한다. 웹 서비스를 개발은 크게 윈도우즈 기반의 Visual Studio .Net을 사용한 개발과 EJB(Enterprise Java Bean), CCM(CORBA Component Model) 등을 사용한 개발로 나눌 수 있다. 이들 모두 같은 표준 프로토콜과 데이터 형식을 사용함으로써 구현된 플랫폼에 상관없이 서로간의 접근이 가능하다.

3.2.4 웹 서비스 조립 단계

구현 단계에서 생성된 Facade 컴포넌트와 서비스 지향 컴포넌트 식별단계에서 식별된 Backside 컴포넌트를 서비스 지향 컴포넌트 아키텍처 상에 배치하여 조립하는 단계이다. 조립된 웹 서비스는 WSDL를 사용하여 웹 서비스를 게시하기 위한 웹 서비스 기술 문서를 생성한다. 이 문서에는 웹 서비스의 상호작용에 관한 정보를 XML 형식으로 정의하고 서비스 호출 메시지 및 응답 메시지에 관한 정보를 가지고 있다.

3.2.5 검사 및 배포

조립된 웹 서비스가 제대로 동작하는지 검사하는 단계이다. 웹 서비스를 구성하는 컴포넌트들 간의 상호 작용이 제대로 이루어지는지를 검사하고, 웹 서비스를 검사를 위해 만들어진 테스트 서버에 WSDL 문서를 사용하여 웹 서비스를 등록함으로써 UDDI 등록 검사를 한다. 또, 등록된 웹 서비스의 호출과 응답 메시지가 SOAP 명세와의 일치여부를 검사한다. 또한, 시스템 요구사항에 대한 일치여부를 검사한다.

검사를 마친 웹 서비스를 UDDI 저장소에 등록한 후 웹 서비스를 제공하는 단계이다. 등록된 웹 서비스는 UDDI를 통해서 검색가능하며, UDDI에서 제공되는 WSDL문서를 통하여 실제 웹 서비스의 호출하여 사용한다.

4. 사례 연구

4.1 상품광고 웹 서비스

4.1.1 문제정의

본 논문에서 구현하고자 하는 웹 서비스는 상품을 팔고자 하는 판매자가 소비자에게 광고를 위한 메일링을 제공할 때, 광고의 효과를 높이기 위하여 상품에 관심이 있는 사용자에게만 상품 광고 메일을 전송하는 시스템이다. 판매자는 웹 서비스에게 소비자에게 보내고자 하는 메일의 내용과 그 상품에 대한 분류를 입력하면, 웹 서비스는 해당 분류에 관심이 있는 사람에게 메일을 전송한다.

4.1.2 상품광고 웹 서비스의 구조

메일링 서비스를 받는 내부 사용자와 개발된 웹 서비스를 사용하기 위해 접근하는 외부 사용자 및 다른 웹 서비스가 사용자 계층에 해당한다. 웹 서버는 웹 서비스 정의 및 게시 그리고 XML을 핸들링할 수 있어야하며 작업의 흐름을 제어하는 기능을 제공한다. 이는 서비스 계층에 해당하며, 메일링 시스템과 통합 가능한 시스템으로 확장될 경우 Flow Facade 컴포넌트가 위치할 수 있다. 메일링 웹 서비스 시스템은 논리적 구조에서의 서비스 계층과 운영 계층을 함께 포함하는 부분이다. 서비스 계층에서의 Core Facade 컴포넌트는 운영 계층의 Backside 컴포넌트를 기반으로 웹 서비스를 제공하기 때문에 이 두 부분은 하나의 시스템에 위치하게 된다. 마지막으로 지원 계층에서는 메일링 웹 서비스 시스템이 사용하는 메일링 리스트, 상품 리스트 등의 데이터 자원이 위치한다(그림 6).

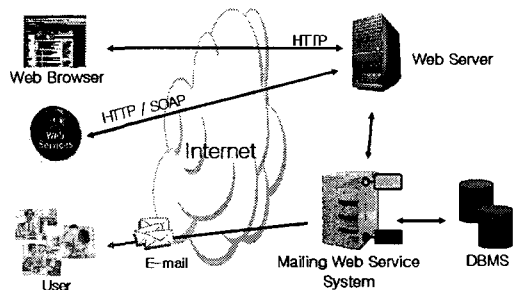


그림 6. 상품 광고 웹 서비스의 구조

4.1.3 요구사항 분석

문제 정의 단계에서의 시나리오를 바탕으로 그

림 7과 같이 시스템이 가져야 할 기능을 나타내는 유스 케이스 모델을 생성한다. 그림 8은 Seller 측면에서 시스템 기능들의 작업의 흐름을 보여주는 서비스 Workflow 모델로서 설계 단계의 서비스 지향 컴포넌트 상호작용 분석에서 입력으로 사용된다.

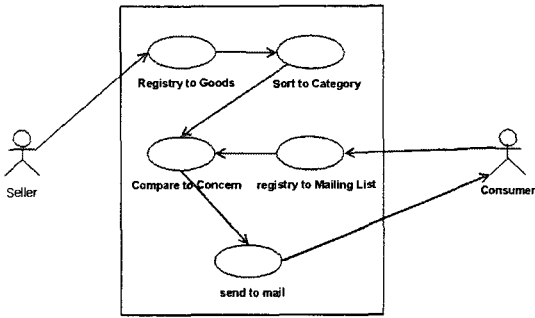


그림 7. 유스 케이스 모델

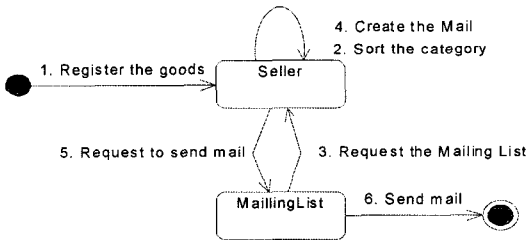


그림 8. 서비스 Workflow 모델

4.1.4 서비스 지향 설계

(1) 서비스 지향 컴포넌트 식별

이 단계에서의 생성물은 서비스 개념 모델과 초기 아키텍처이다. 서비스 개념 모델은 유즈케이스 모델로부터 식별한 시스템의 기능들을 클래스화하여 실제화 시킨 모델이며, 초기 아키텍처는 이들 클래스를 기능별로 묶어 패키지 또는 컴포넌트를 구성하여 작성되는 모델이다. 이를 통해 개발해야할 서비스 지원 컴포넌트와 일반적인 컴포넌트를 분류한다.

그림 9에서는 서비스가 제공하는 기능을 Class Diagram을 통하여 나타내었으며, Send_mail, Receive_mail, SMTPmail은 Seller 컴포넌트로, MaillingList와 ListDB는 MaillingList 컴포넌트로 식별된다. 그리고 이 둘을 통합하는 Facade Class는 Facade 컴포넌트로 식별된다.

그림 10에서는 MaillingList 컴포넌트와 Seller 컴포넌트의 인터페이스를 Facade 설계 패턴에 의해 하나의 통합된 인터페이스를 Facade 컴포넌트에서 제공하며, 이 Facade 컴포넌트는 웹 서비스를 가능하도록 XML 기반의 SOAP을 사용하여 구현하며, UDDI등록과 웹 게시를 위하여 WSDL 문서를 생성한다.

(2) 서비스 지향 상호작용 분석

그림 11은 인터페이스 명세로써 컴포넌트들 간의

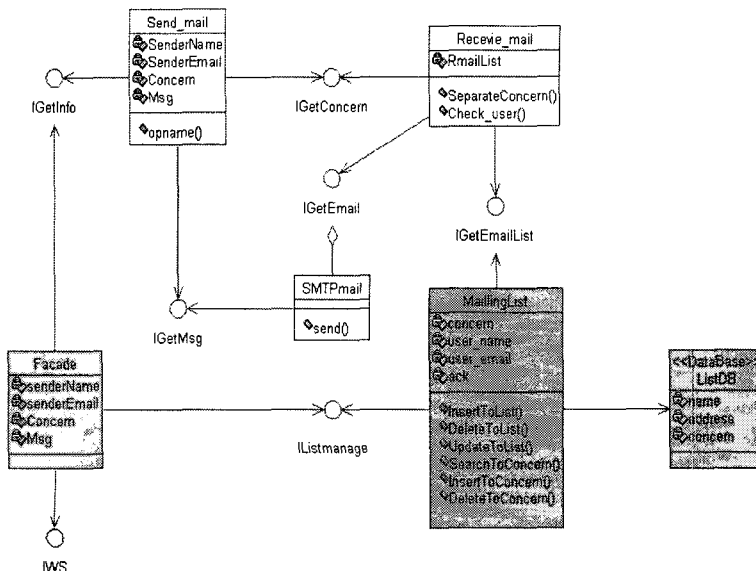


그림 9. 서비스 개념 모델

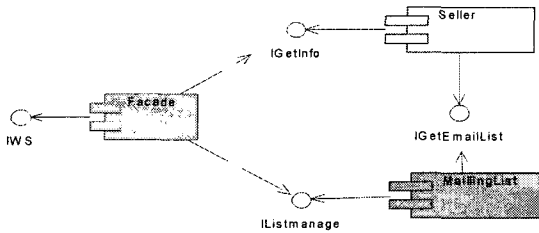


그림 10. 초기 아키텍처

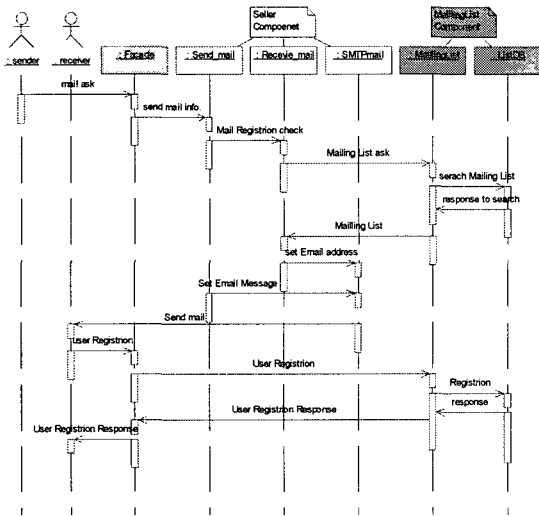


그림 11. 인터페이스 명세

상호작용을 나타낸 것으로, 서비스 지향 컴포넌트 식별 단계에서 식별된 컴포넌트와 내부 클래스의 인터페이스를 식별하고, 서비스 Workflow 모델의 서비스가 가지는 기능의 절차를 실체화한다. 사용자는 웹 서비스기능을 제공하는 Facade 컴포넌트의 인터페이스를 통하여 서비스 요청을 한다. Facade 컴포넌트는 일반적인 인터페이스를 사용하여 내부 컴포넌트와 상호 작용을 한다.

(3) 서비스 지향 컴포넌트 스펙 작성

컴포넌트 식별 단계와 상호 작용분석 단계에서 생성된 모델을 문서화하는 단계로써, 컴포넌트의 배포 또는 재사용을 위한 정보를 작성한다(그림 12). 차후 웹 서비스 등록을 위한 WSDL 문서 정련을 위한 입력으로 사용된다.

4.2 실행에

그림 13에서는 웹 서비스 게시를 위해 SOAP 호출 테스트를 하는 화면이다. 호출 메시지는 웹 서비스에

Item	Description
Service Name	MailingSystem
Port name	MailingSoapPort
Type	SenderMailConcernMsg
Message	IWS

그림 12. 서비스 지향 컴포넌트 스펙

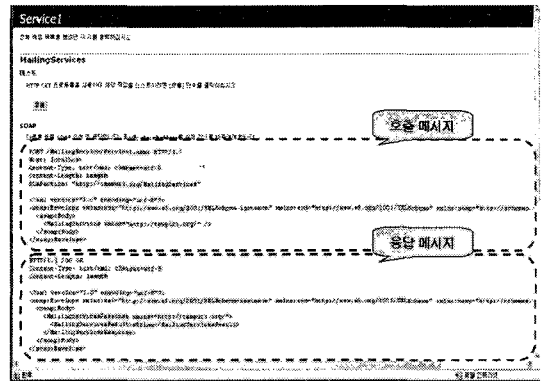


그림 13. 웹 서비스 호출 테스트

게 서비스를 제공받기 위하여 서비스를 호출하는 메시지이고, 응답 메시지는 웹 서비스가 호출에 대한 응답을 보내는 메시지이다.

그림 14는 생성된 웹 서비스를 게시하고 이를 사용하기 위한 SOAP을 사용한 클라이언트이다. 클라이언트를 통해 웹 서비스에 소비자에게 보내고자 하는 내용과 상품의 분류를 전송한다. 상품 광고 웹 서비스는 상품의 분류 항목에 따라 소비자에게 광고 메일을 발송한다. 그림 15는 클라이언트가 서비스 요청에 대한 결과를 나타내는 응답 메시지이다.

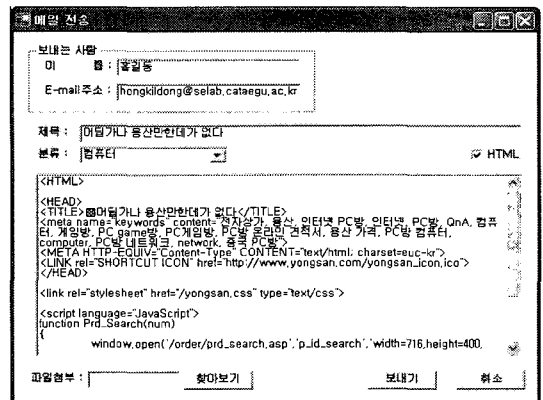


그림 14. 웹 서비스 클라이언트

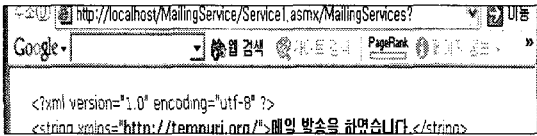


그림 15. 서비스 요청에 대한 응답 메시지

5. 결론 및 향후연구

다양한 벤더의 플랫폼을 위해 작성된 애플리케이션들을 하나로 통합할 경우 웹 서비스가 유용한 기능을 제공한다. 일반적으로 웹 서비스는 네트워크를 통해 XML 포맷의 메시지를 보내서 액세스될 수 있는 소프트웨어이다. XML은 특정 프로그래밍 언어나 OS에 제약받지 않는다. 따라서 이종 시스템간 정보 교환용 프로토콜로 쓰일 수 있다. WSDL은 웹 서비스가 제공하는 인터페이스를 기술하는데 사용된다. 이러한 웹 서비스 기술은 소프트웨어의 폭넓은 사용으로 인해 다른 영역과의 통합을 위한 부분에 효과적으로 사용될 수 있는 기술이다.

본 논문에서는 신뢰성과 민첩성을 보장하는 웹 서비스 개발을 컴포넌트를 기반으로 하기 위해, 웹 서비스의 4계층의 논리적 구조로 분할하였다. 이를 통해 컴포넌트 기반의 서비스 지향 아키텍처를 제시하였으며, 내부는 크게 Facade와 Backside 컴포넌트로 구성하였다. 아키텍처를 컴포넌트로 구성함으로써 컴포넌트의 특징인 재사용성과 신뢰성을 제공할 뿐만 아니라, 컴포넌트 기반으로 구성된 기존의 시스템을 웹 서비스로의 변경이 Facade 컴포넌트만을 개발하여 추가함으로써 웹 서비스를 제공할 수 있기 때문에 변경에 대한 용이성뿐만 아니라 기존의 인터페이스 또한 그대로 사용할 수 있다. 따라서, 웹 서비스를 직접 지원하고 상호 운용되는 Facade 컴포넌트 개발하기 위한 개발 프로세스 제시하였다. 제시된 아키텍처와 프로세스를 적용하여 상품 광고 웹 서비스를 구현하였으며, 기존에 만들어진 컴포넌트를 사용함으로써 재사용성을 보장하고, Facade 컴포넌트만을 개발함으로써 웹 서비스의 구축이 가능하였으므로 웹 서비스 개발의 민첩성을 보장하였다. 향후 연구로써 Flow Facade 컴포넌트를 통한 어플리케이션의 통합과, 다른 웹 서비스와의 성능에 대한 평가가 요구된다.

참고 문헌

- [1] Roy, Ramanujan, "Understanding Web services," IT Professional, Vol. 3, No. 6, pp. 69-73, Nov. 2001.
- [2] Uche Ogbuji, "The Past, Present and Future of Web Services," <http://www.webservices.org/index.php/article/articleview/663/4/61/>, 2002.
- [3] Litoiu, M, "Migrating to Web Services-latency and scalability," Proceedings of Fourth International Workshop on Web Site Evolution, pp. 13-20, Oct. 2002.
- [4] Michael Champion, Chris Ferris, Eric Newcomer, Iona and David Orchard, "Web Services Architecture: W3C Working Draft," <http://www.w3.org/TR/ws-arch/>, 2002.
- [5] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
- [6] Koichi TERA, Noriaki IZUMI and Takahira YAMAGUCHI, "Coordinating Web Services based on business models," Proceedings of the 5th international conference on Electronic commerce, pp. 473-478, Sep. 2003.
- [7] Raul Silaghi and Alfred Strohmeier, "Integrating CBSE, SoC, MDA, and AOP in a Software Development Method," Seventh International Enterprise Distributed Object Computing Conference (EDOC'03), pp. 136-146, Sep. 2003.
- [8] Jian Yang, "Service-oriented computing: Web service componentization," Communications of the ACM, Vol. 46, No. 10, pp.35-40, Oct. 2003.
- [9] John Cheesman, John Daniels, UML Components A Simple Process for Specifying Component-Based Software, Addison-Wesley, 2001.
- [10] 석광진, 윤심, "시스템 통합 프로젝트에서의 웹 서비스 적용," 한국정보처리학회 소프트웨어공학논문지, 제 15권, 제 4호, 2002.

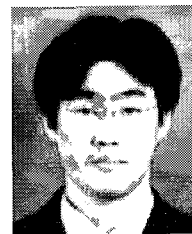
- [11] Richard Veryard, "Modeling for SOA," CBDi Journal, pp. 11-18, Feb. 2003.
- [12] Ivica Crnkovic, Component-based Software Engineering - New Challenges in Software Development, Software Focus, Dec. 2001.
- [13] 김민수, "웹 서비스 표준화," 한국정보처리학회지, 제 9권, 제 4호, pp. 31-35, 2002.
- [14] Al-Tahat, Sembok, Bin Idris, "Using design patterns in the development of a planner-based courseware system," Proceedings of IEEE International Conference on Electrical and Electronic Technology, Vol. 2, pp. 873-876, Aug. 2001.
- [15] Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, Prentice Hall, 1998.
- [16] OMG, OMG Unified Modeling Language Specification Version 1.5, <http://www.omg.org/uml>, 2003.
- [17] Kilgore, R.A, "Simulation Web services with .Net technologies," Simulation Conference, 2002. Proceedings of the Winter, Vol. 1, pp. 841-846, Dec. 2002.
- [18] Gunter Preuner, Michael Scherfl, "Integration of Web Services into Workflow through a Multi-Level Schema Architecture," Proceeding of Advanced Issues of E-Commerce and Web-Based Information Systems, Fourth IEEE International Workshop on, pp. 51-60, Jun. 2002.
- [19] 최하정, 김행곤, e-비즈니스 컴포넌트 개발에 관한 설계 및 구현, 정보처리학회논문지, 제 10-D권, 제 1호, pp. 85-100, 2002.



박 동 식

2003년 대구가톨릭대학교 컴퓨터정보통신(공학사)
2003년~현재 대구가톨릭대학교 컴퓨터정보통신(석사과정)

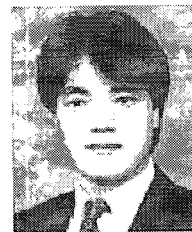
관심분야 : CBSE, 서비스지향 개발 방법론, 임베디드 소프트웨어 개발 방법론



신 호 준

1998년 경일대학교 전자계산학과(공학사)
2000년 대구가톨릭대학교 전산통계학과(이학석사)
2003년 대구가톨릭대학교 전산통계학과(이학박사)

관심분야 : CBSE, AOSE, 서비스지향 개발 방법론, 임베디드 소프트웨어 개발 방법론 및 툴, 제품개발 생산기술



김 행 곤

1985년 중앙대학교 전자계산학과(공학사)
1987년 중앙대학교 대학원 전자계산학과(공학석사)
1991년 중앙대학교 대학원 전자계산학과(공학박사)
1978년~1979년 미 항공우주국

직원 연구원

1987년~1989년 한국전기통신공사 전임연구원
1988년~1989년 AT&T 직원 연구원
2001년~2002년 Central Michigan University 교환교수
1990년~2000년 대구효성가톨릭대학교 컴퓨터공학과 부교수

2000년~현재 대구가톨릭대학교 컴퓨터공학과 교수
관심분야 : 컴포넌트기반 소프트웨어공학, 서비스지향 개발 방법론, 임베디드 소프트웨어 개발 방법론 및 툴, 프로덕트라인 공학