

SOA 기반의 Facade 컴포넌트에 관한 연구

박동식, 신호준, 김행곤
대구가톨릭대학교 컴퓨터정보통신공학부
e-mail:{s6047029, g98521002, hangkon}@amare.ac.kr

A Study on Facade Component based on Service Oriented Architecture

Dong-Sik Park, Ho-Jun Shin, Haeng-Kon Kim
*Dept of Computer Information Communication Engineering,
Catholic University of Daegu

요 약

최근 소프트웨어의 다양한 환경에서의 사용으로 다른 시스템과의 통합을 위한 요구가 증가하고 있으며, 소프트웨어 개발이 크고 복잡하게 됨에 따라 유지보수가 어렵게 되었다. 컴포넌트 기반 개발은 이런 문제점과 요구를 해결하기 위한 방법으로 대두되고 있으며, 웹 서비스와의 연동을 통한 기업의 요구사항 변경에 민첩하게 대응되는 방법으로 인식되고 있다.

본 논문에서는 웹 서비스 통한 소프트웨어 개발비용의 감소와 생산성의 향상으로 기업의 요구사항 변경에 대한 민첩성을 보장하기 위한 컴포넌트 기반의 웹 서비스 개발을 제안한다. 이는 컴포넌트의 재사용성과 대체성을 통해 신뢰성과 수정의 용이성을 보장하고자 한다. 이를 위한 웹 서비스 개발 컴포넌트기반의 서비스 지향 아키텍처를 정의하고, 서비스 지향 모델 기반 Facade-Backside 컴포넌트를 분석/설계를 통해 웹 서비스 구현 전략에 대해 논한다. Facade-Backside 컴포넌트에서 등록, 검색을 위한 웹 서비스 사례를 제시한다.

1. 서론

최근 웹 서비스는 인터넷을 통해 기업 상호간에 비즈니스를 연결할 수 있도록 하는 것으로 IT전문가들이나 관련 업계에 한정되지 않고 많은 분야에서 이슈가 되고 있다. 가트너 그룹에서는 2005년까지 중견 및 대기업의 90%가 웹 서비스를 채택할 것으로 예상된다. 이것은 웹 서비스의 구축이 개발비용의 감소와 구축속도 증진이 가능함으로써, 기업의 빠른 요구사항 변경에 대한 민첩성 향상을 보장하기 때문이다[1].

소프트웨어의 폭넓은 사용으로 인해 다른 영역과의 통합을 위한 요구가 증가하고 있다. 또 소프트웨어의 개발이 크고 복잡하게 됨에 따라 업데이트나 수정이 어렵게 되었다. 하지만, 컴포넌트 기반 개발을 사용하여 재사용성(reusability)과 대체성(replaceability)을 제공함으로써 이러한 문제를 해결하고 있다. 그러나, 성숙도에 따른 다양한 컴포넌트 리파지토리의 구축으로 재사용성을 제공하고 있지만, 소프트웨어 아키텍처에 의존성 등의 몇몇 이유로 만족스

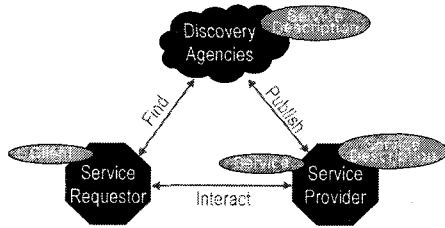
러운 결과를 유도하지 못하고 있다. 이러한 의존성을 웹 서비스를 통해 제거하고, 업데이트나 수정에 따른 배포를 제공한다.

본 논문에서는 신뢰성과 민첩성을 보장하는 웹 서비스 개발을 위해 서비스 지향 컴포넌트를 제공하고자 한다. 이를 위해 Facade, Backside 컴포넌트로 정의한 SOA(Service Oriented Architecture)를 제시한다. 또한, 웹 서비스를 직접 지원하고 상호 운용되는 Facade 컴포넌트 개발 프로세스를 제안하며, 서비스 지향 분석, 설계 및 사례를 제시한다.

2. 관련 연구

2.1 서비스 지향 아키텍처

SOA는 서비스 지향 활동을 위한 기술적 기반이며, 원격으로 객체나 서비스에 접근하는 능력, 서비스 검색, 동적 바인딩 그리고 Loose coupled를 지원하는 특징을 가진다[2].



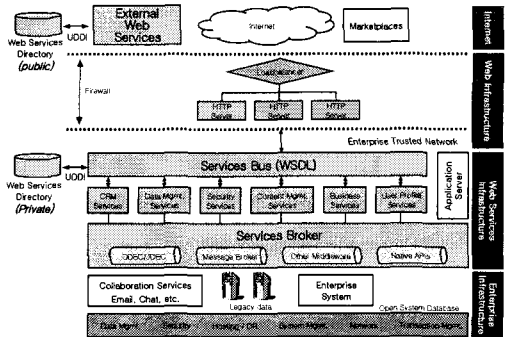
(그림 1) Service Oriented Architecture

(그림 1)과 같이 아키텍처내부의 역할들은 Services Requestor, Discovery Agencies, Service Provider로 나누어져 있다. 서비스의 명세는 XML 표준을 사용하며, 통신을 하기 위한 SOAP(Simple Object Access Protocol), 어떤 서비스를 제공하는지 기술하는 WSDL(Web Services Description Language), 서비스 브로커역할을 제공하는 UDDI(Universal Description Discovery and Integration)과 같은 표준을 사용한다. 서비스의 제공은 XML 기반의 메시지 흐름으로 이루어진다.

2.2 웹 서비스

웹 서비스는 SOA 기반이며, URI(Uniform Resource Identifier)에서 식별된 소프트웨어 시스템이다. 이들의 인터페이스와 바인딩은 XML을 사용하여 정의되고 표현한다. 웹서비스와 시스템은 네트워크의 IP프로토콜을 통해 XML 기반의 메시지를 사용하여 상호동작을 제공하는 하도록 규정되어 있다. 이러한 표준은 W3C(World Wide Web Consortium)에서 가지고 있고; Loose coupling은 공급자와 요청자 플랫폼간의 기술적 종속성을 제거, Self-defining은 XML 데이터의 네임 필드 또는 WSDL 파일에서 제공, Dynamic discovery는 웹 서비스의 가용성과 기능을 설계시간이 아닌 실시간으로 발견, Vender neutrality는 웹 서비스에 대한 업계 전체의 합의가 이루어짐, Lightweight는 애플리케이션의 상호운영성은 시간과 노력을 감소, Easy learning curve는 COM+, CORBA 심지어 RMI와 RPC보다 쉽게 이해하고 사용, Firewall-friendly는 내부 네트워크나 인터넷사이의 방화벽을 HTTP로 할당된 포트 80으로 접근하는 특징을 가지고 있다.

이런 특징들은 시스템간의 통신을 위한 높은 효과를 제공하며, 웹 서비스는 애플리케이션의 개발과 배포에 중점을 둔다. 다음 (그림 2)과 같이 웹 서비스는 웹 환경을 포함하고 있으며, 기업 내부의 웹 서비스와 외부의 웹 서비스들은 통신을 할 때 표준 데이터형식 XML과 프로토콜 SOAP를 사용함으로써 플랫폼간의 종속성을 제거한다.



(그림 2) Web Services Architecture

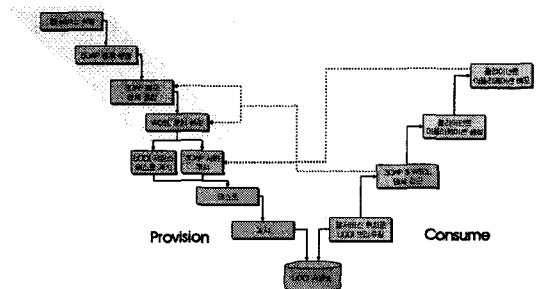
3. 컴포넌트 기반 SOA

웹 서비스의 개발은 크게 UDDI를 기준으로 (그림 3)과 같이 Provision과 Consumer 두 부분으로 나눈다. Provision은 UDDI에 웹 서비스 등록을 위한 개발 프로세스이며, Consumer는 등록되어 있는 웹 서비스를 이용하여 개발하는 부분이다.

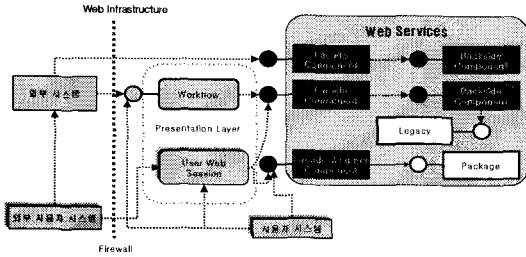
컴포넌트 기반의 웹 서비스를 만들기 위해서 고려한 부분은 전체 개발 프로세스에서 Provision에 대한 부분이다. 그 중에서도 테스트와 게시 부분은 생성된 웹 서비스에 대한 테스트와 UDDI등록에 관한 부분이다.

본 논문에서는 웹 서비스 식별에서부터 웹 서비스 생성에 대한 단계를 중심으로 개발 프로세스를 제안한다. 웹 서비스를 컴포넌트 기반으로 개발을 하게되면 기존 컴포넌트들을 재사용 함으로써 개발 시간과 비용이 감소된다. 또한, 컴포넌트의 대체를 통해 사용자 요구변경에 대한 민첩성을 가진다.

(그림 4)는 컴포넌트 기반으로 웹 서비스 아키텍처를 나타낸 것이다. 방화벽을 중심으로 외부시스템 혹은 외부사용자 시스템과 웹 서비스영역으로 나뉜다. 프레젠테이션 계층은 User Web Session과 Workflow가 해당된다.



(그림 3) 웹 서비스 개발 프로세스



(그림 4) 컴포넌트 기반 SOA

외부 사용자는 3가지의 방법으로 웹 서비스를 제공받을 수 있다. 첫 번째, 웹 브라우저를 사용하여 User Web Session에서 제공하는 웹 페이지의 형식으로 서비스를 제공받는다. 두 번째, 외부 시스템을 사용하여 Workflow 서비스를 통하거나 마지막으로, 직접 웹 서비스를 제공받을 수 있다. Workflow는 내부의 웹 서비스들에 대한 작업의 흐름을 제어하는 웹 서비스이다. 내부 사용자도 유사한 방법으로 서비스를 제공받을 수 있다. 단지 내부 사용자의 경우 방화벽 내부에 존재하기 때문에 방화벽을 통과하지 않아도 되고, 외부 시스템을 사용할 필요가 없다.

웹 서비스의 내부는 Backside 컴포넌트와 Facade 컴포넌트로 분류하였다. Backside 컴포넌트는 일반적인 비즈니스 로직 컴포넌트와 데이터 컨트롤 컴포넌트와 같이 주로 내부 로직을 담당하는 컴포넌트들로 구성이 된다.

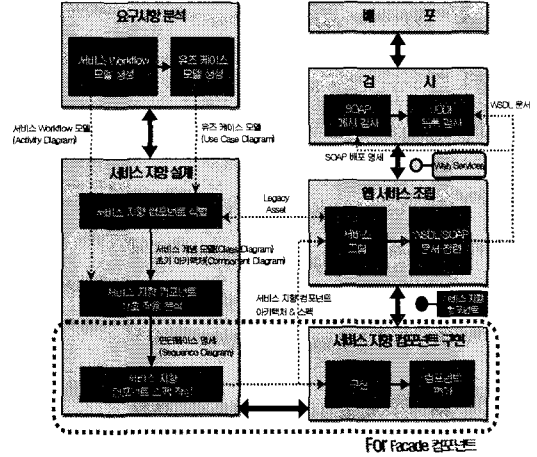
Facade 컴포넌트는 SOAP과 WSDL로 명세 되어 웹 서비스가 가능하도록 해준다. Backside 컴포넌트는 기존의 개발된 컴포넌트의 대체가 가능한 부분이라면, Facade는 본 논문에서 개발해야할 목표 컴포넌트이다. 이는 SOAP의 구현을 통해 실제적인 웹 서비스를 제공하고, Presentation Layer에 위치한 웹 서비스와 웹 기능 모듈과 상호 운용되며, 외부 시스템이나 외부사용자라도 통신이 가능하다.

4. 웹 서비스 지원 컴포넌트 개발 프로세스

본 논문에서는 웹 서비스의 내부에서 실제적인 웹 서비스 기능을 수행할 Facade 컴포넌트를 개발하기 위해서 모델 기반의 개발 방법을 사용하여 다음 (그림 5)와 같이 웹 서비스지원 컴포넌트 개발 프로세스를 제안한다.

4.1 프로세스 행위

요구사항 분석 단계에서는 작업을 흐름을 파악하기 위하여 서비스 Workflow 모델을 UML의 Activity Diagram을 통해 생성한다. 또한, 작업의 흐름을 통하여 어떤 기능이 있는가를 파악하기 위하여 유즈 케이스 모델을 생성한다. 두 모델을 생성함으로써 요구사항 분석이 이루어진다.



(그림 5) 웹 서비스지원 컴포넌트 개발 프로세스

서비스 지향 설계 단계는 전체 단계에서 가장 중요한 단계이다. 전 단계의 산출물과 기존 소프트웨어 자산(기존의 컴포넌트, 레거시 시스템 등)에 관한 정보를 사용하여 컴포넌트 명세와 아키텍처를 생성한다.

- 서비스 지향 컴포넌트 식별 단계에서는 요구사항 분석 단계에서 산출물인 유즈 케이스 모델을 통해 컴포넌트를 식별한다. 초기 컴포넌트 아키텍처 모델(Component Diagram)과 서비스 개념 모델(Class Diagram)을 생성한다.
- 서비스 지향 컴포넌트 상호 작용 분석 단계에서는 식별 단계의 산출물과 서비스 Workflow 모델을 가지고 컴포넌트간의 상호 작용을 분석하여 인터페이스 명세를 Sequence Diagram을 통해 생성한다.
- 서비스 지향 컴포넌트 스펙 작성 단계에서는 정련된 컴포넌트 사용 계약과 실제화 계약에 대한 명세를 정의한다. 사용계약은 인터페이스 명세로 정의된다. 실제화 계약은 컴포넌트 명세로서 정의된다. 컴포넌트 스펙에는 인터페이스가 구현되는 방식에 대한 제약사항을 포함하며, 초기 아키텍처를 정련하여 서비스 지향 컴포넌트 아키텍처를 생성한다.

서비스 지향 컴포넌트 구현 단계에서는 서비스 지향 설계에서 식별한 웹 서비스의 기능을 수행할 수 있는 서비스 지향 컴포넌트(Facade 컴포넌트)를 서비스 지향 컴포넌트 아키텍처를 기반으로 구현한다.

웹 서비스 조립 단계에서는 서비스 지향 컴포넌트 아키텍처를 기반으로 서비스 지향 컴포넌트와 서비스 지향 설계에서 식별된 기존 자산을 조립하여 웹

서비스를 생성한다. UDDI등록을 위해 조립된 웹 서비스에서 WSDL 문서와 SOAP 배포 명세를 정련화한다.

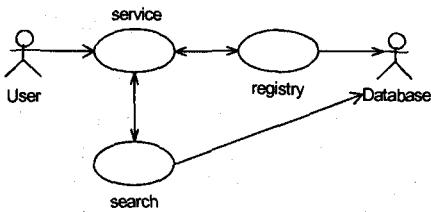
검사 단계에서는 웹 서비스에 대한 검사를 한다. 서비스간의 통신이 SOAP 배포 문서와 일치하는지 서비스 실행이 되는지를 메시지 전송을 통하여 검사한다. 또한, WSDL 문서를 사용하여 웹 서비스를 등록한 후 서비스 검색이 가능한지 검사한다.

배포 단계에서는 검사를 마친 웹 서비스를 UDDI 저장소에 등록한 후 웹 서비스를 제공하게 되며, 이를 사용자가 requestor 프로그램 작성을 통해 사용 가능하다.

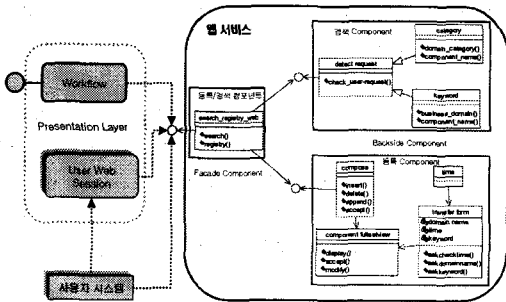
4.2 등록, 검색 웹 서비스 사례

다음 (그림 6)은 컴포넌트의 리파지토리에 컴포넌트를 등록하고 검색하는 간단한 사례를 제시한 것이다. 이는 요구사항 분석 단계에서 작업의 흐름을 통해 어떠한 작업이 있는가를 파악하기 위하여 생성된 산출물이다. 이 모델을 통하여 어떠한 기능이 컴포넌트화 될 수 있는지의 요구사항을 파악할 수 있다.

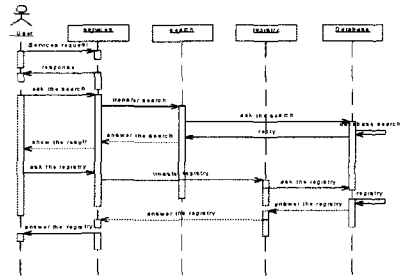
다음 (그림 7)은 요구사항 분석 단계에서 생성된 유즈 케이스 모델을 사용하여 서비스 지향 컴포넌트 식별 단계에서 생성된 산출물이다. 그림에서는 서비스 개념 모델과 초기 아키텍처 모델을 함께 표현하였다. 다음 (그림 8)은 초기 아키텍처 모델과 서비스 Workflow 모델을 가지고 서비스 지향 컴포넌트 상호 작용 분석 단계에서 생성된 산출물이다.



(그림 6) Use Case Diagram



(그림 7) Class/Component Diagram



(그림 8) Sequence Diagram

5. 결론 및 향후 연구

현재 인터넷을 이용한 웹 환경에서의 비즈니스가 확대되고 있다. 이러한 환경의 변화에 따라 서비스를 지원하기 위해 웹 서비스가 관심사로 떠오르고 있다. 또한 서비스의 통합으로 인하여 소프트웨어가 크고 복잡하게 되어 업데이트나 수정이 어렵게 되었다. 하지만 컴포넌트 기반 개발을 사용함으로써 재사용성과 대체성을 제공함으로써 해결이 가능하였다. 그러나 아키텍처상의 의존성과 같은 문제로 인하여 만족스러운 결과를 얻기 힘들었다.

이러한 문제를 해결하기 위해 웹 서비스 지원 컴포넌트가 필요하다. 웹 서비스는 개발비용의 감소와 구축속도의 증진으로 기업의 요구사항 변경에 대한 민첩성을 보장한다. 또한 컴포넌트를 기반으로 웹 서비스를 구성하여 재사용성과 대체성에 대한 신뢰성 향상을 가져온다. 향후 연구로 서비스 지향 아키텍처의 정련화와 이를 통한 구현 단계에서 Facade 컴포넌트와 Backside 컴포넌트의 구현 및 평가가 수반되어야 하겠다.

참고문헌

[1] 연구황, "비즈니스적인 웹 서비스(Web Services) 이해," 정보처리학회논문지, Vol. 9, No. 4, pp. 5-14, 2002.
 [2] John Cheesman, John Daniels, UML components A Simple Process for Specifying Component-Based Software, Addison-Wesley, 2001.
 [3] 석광진, 윤심, 시스템 통합 프로젝트에서의 웹 서비스 적용, 소프트웨어 공학회지, Vol 15 No 4, 2002. 12
 [4] Richard Veryard, "Modeling for SOA," CBDi Journal, pp. 11-18, February, 2003.
 [5] Ivica Crnkovic, Component-based Software Engineering - New Challenges in Software Development, Software Focus, December 2001.
 [6] 김민수, "웹서비스 표준화," 정보처리학회지, Vol. 9, No. 4, pp. 31-35, 2002.
 [7] 최하정, 김행근, e-비즈니스 컴포넌트 개발에 관한 설계 및 구현, 정보처리학회논문지D, Vol. 10-D, No. 1, pp. 85-100, 2002.