

웹 서비스 개발을 위한 Facade와 Backside 컴포넌트 추출에 관한 연구

박동식, 신호준, 김성원*, 김행곤
대구가톨릭대학교 컴퓨터정보통신공학부
*안양대학교 전기전자공학과

e-mail:{sabujak, component, hangkon}@cu.ac.kr, *swkim@aycc.anyang.ac.kr

A Study on Extraction of Facade and Backside Component for Web Service Development

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

*Dept. of Electrical & Electronic Engineering, Anyang University

요 약

최근 SOA(Service Oriented Architecture)와 웹 서비스에 대한 관심이 높아져가고 있다. 웹 서비스는 재사용 측면에서 현재 사용되고 있는 컴포넌트와 많은 공통점이 있다. 웹 서비스의 개발과 수정 및 확장을 위해서 신뢰성 있는 컴포넌트를 이용할 경우 개발 시간의 단축 등의 많은 이점을 가질 수 있다. 이러한 컴포넌트를 이용하여 웹 서비스를 개발할 경우 웹 서비스 도메인에 대한 이해뿐만 아니라 요구사항의 추출 및 식별은 매우 중요한 부분이다.

본 논문에서는 사용자 요구사항으로부터 컴포넌트를 식별하고 이들 컴포넌트를 서비스 지원 컴포넌트인 Facade 컴포넌트와 Backside 컴포넌트로 식별할 수 있는 방법을 제시하고자 한다. 이를 위해 서비스를 오퍼레이션과 데이터 요소로 정의하고, 정의된 요소를 사용한 모델을 제시한다. 이러한 컴포넌트의 식별은 현재 활성화되어 있는 컴포넌트를 사용할 수 있도록 함으로써 웹 서비스 개발의 생산성을 보장하며, 웹 서비스를 컴포넌트로 구성함으로써 컴포넌트의 기본적인 특징인 재사용과 대체성을 통해 웹 서비스의 신뢰성과 수정의 용이성 등을 얻을 수 있다.

1. 서론

최근 SOA와 웹 서비스의 관심이 높아져가고 있으며 이들에 대해 CBD(Component Based Development) 관점에서의 접근이 이루어지고 있다. 웹 서비스와 CBD에서의 공통적인 문제의 핵심은 재사용 가능한 컴포넌트 또는 서비스의 개발과 확보에 있다. 현재 컴포넌트의 개발은 성숙단계에 올라와 있으며, 컴포넌트 저장소를 통해서 컴포넌트를 가져와 사용할 수 있다. 하지만 웹 서비스는 아직 저장소인 UDDI(Universal Description Discovery and Integration)가 활성화되지 않았으며, 많은 서비스들이 개발 중에 있다[1][2].

현재 기존의 개발되어 있는 컴포넌트를 사용하여 웹 서비스를 개발하고자 하는 많은 노력이 있다. 이러한 웹 서비스의 개발에 컴포넌트를 사용하기 위해서는 컴포넌트의 식별단계에서 일반 컴포넌트와 개발

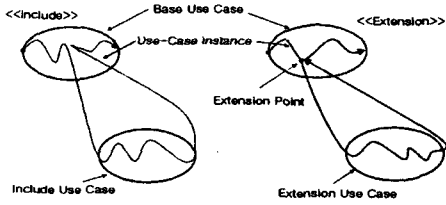
하여야할 서비스 지향 컴포넌트의 식별이 가장 중요한 부분이다.

본 논문에서는 사용자의 요구사항으로부터 서비스 지향 컴포넌트인 Facade 컴포넌트와 일반적인 컴포넌트를 식별을 위한 방법을 제시하고자 한다. 식별을 위해 오퍼레이션 요소와 데이터 요소와 같은 두 가지의 서비스 요소 제시한다. 또한, 이들 요소에 대한 모델 정의를 통한 컴포넌트의 분류를 통해 서비스 지향 컴포넌트 식별을 위한 방법을 제시한다.

2. 관련 연구

2.1 컴포넌트 추출 방법

컴포넌트 추출을 위해서는 시스템에서 제공해야 할 기능을 기능적, 계층적으로 구분하여 모듈화 한다. 이 때 구분된 모듈은 그 자체로 의미 있는 서비스들을 제공하는 하나의 실행 단위여야 하며, 다른



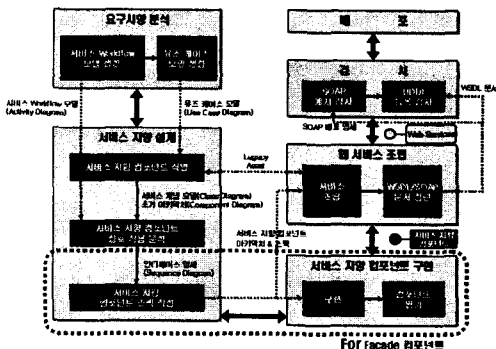
(그림 1) 유스케이스와 <<include>> 및 <<extend>> 관계

모듈과 독립적이어야 한다. 이런 하나의 작은 모듈을 컴포넌트라 한다[3].

컴포넌트의 추출은 (그림 1)에서 나타난 것과 같이 시나리오 분석 시, 하나 이상의 유스케이스에서 동일하게 반복되는 사용자와 시스템간의 상호작용이 발견 될 경우 이를 <<include>> 관계의 다른 유스케이스로 도출할 수 있다. 유스케이스를 <<include>> 관계로 사용하는 유스케이스들에서 제공하는 공통 서비스를 하나의 컴포넌트로 제공할 수 있다. 그리고 사용자와 시스템간의 상호작용이 아닌 어떤 조건에 의해서 수행 될 수 있는 상호작용이 발견 될 경우 이를 <<extend>> 유스케이스로 분리해 낼 수 있다. 따라서 베이스 유스케이스는 조건에 상관없이 반드시 수행되어야 하는 상호작용을 포함하고, <<extend>> 유스케이스는 베이스 유스케이스 수행 시 발생하는 조건에 따라 수행 할 수도 있는 상호작용을 포함한다. 이 경우 <<extend>> 유스케이스가 제공하는 선택적인 서비스를 하나의 컴포넌트로 제공할 수 있다.

2.2 서비스 지원 컴포넌트 개발 프로세스

서비스 지원 컴포넌트 개발 프로세스는 웹 서비스를 컴포넌트를 사용하여 개발하기 위한 개발 프로세스이다. (그림 2)와 같이 6단계로 구성이 되어 있으며, 전 단계의 산출물은 다음 단계의 입력으로 사용되도록 구성되어 있다[4].



(그림 2) 서비스 지원 컴포넌트 개발 프로세스

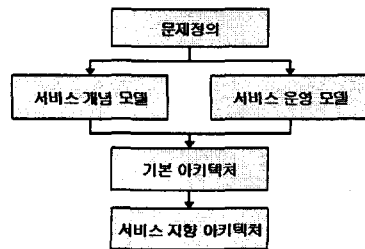
요구사항분석과 서비스 지향 설계 단계는 사용자의 요구사항을 분석하여 Facade 컴포넌트와 Backside 컴포넌트를 식별하는 부분과 식별된 컴포넌트를 조립하여 웹 서비스를 만드는 부분으로 나눌 수 있다. 여기서 Facade 컴포넌트는 웹 서비스를 지원원 위한 기능을 제공하는 컴포넌트이며, Backside 컴포넌트는 일반적인 방법론으로 개발된 컴포넌트로 정의하고 있다. 실제적인 Facade 컴포넌트의 개발은 For Facade 컴포넌트부분에서 이루어진다.

3. 웹 서비스 지원 컴포넌트 추출

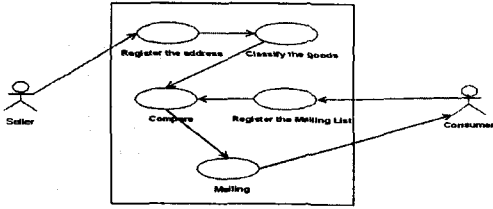
사용자의 요구사항으로부터 컴포넌트 식별을 하는 방법은 전체 서비스 지원 컴포넌트 개발 프로세스의 서비스지향 컴포넌트 식별단계에서 가장 중요한 부분을 차지한다. 신뢰성 있는 웹 서비스의 개발과 개발 시간의 단축 그리고 웹 서비스의 수정 및 확장을 위해서는 컴포넌트를 사용할 필요성이 있다. 즉, 컴포넌트를 사용함으로써 컴포넌트의 가장 큰 특징인 재사용성과 변경에 대한 용이성을 보장할 수 있기 때문이다. 이를 위해서는 사용자의 요구사항으로부터 일반적인 컴포넌트인 Backside 컴포넌트와 웹 서비스가 가능하게 해주는 Facade 컴포넌트를 식별을 위한 방법이 요구된다.

본 논문에서는 컴포넌트를 식별할 수 있도록 서비스 요소를 정의한다. 정의된 요소를 통하여 서비스 개념 모델과 서비스 운영 모델을 생성하며, 이들을 통합하여 Facade 컴포넌트와 Backside 컴포넌트로 식별된 서비스 지향 컴포넌트 아키텍처를 생성한다.

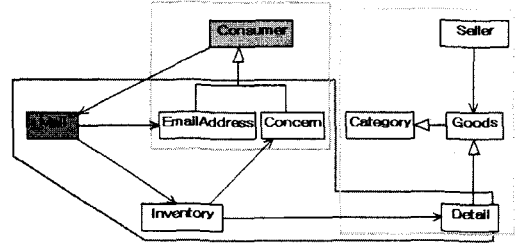
웹 서비스의 기능은 구체적인 서비스와 이를 지원하는 프로세스로 구성되며, 서비스는 크게 2가지 요소로 나눌 수가 있다. 첫 번째는 프로세스를 구성하는 오퍼레이션 요소이며, 두 번째는 데이터 요소로 오퍼레이션에서 사용되는 데이터이다. 이 2가지 요소를 중심으로 웹 서비스를 지원하는 컴포넌트를 추출하는 단계는 (그림 3)과 같다.



(그림 3) 컴포넌트 추출 절차



(그림 4) 유스케이스 모델



(그림 6) 서비스 개념 모델

3.1 문제 정의

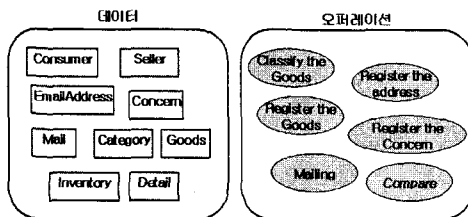
상품 광고를 위해 판매자가 소비자에게 소비자가 관심을 가지는 상품을 분류하여 광고 메일로 보내는 시스템을 대상으로 적용한다. (그림 4)는 사용자의 요구사항을 유스케이스 모델로 나타낸 것이다.

각 유스케이스에서 제공하는 오퍼레이션들은 프로세스 요소의 오퍼레이션을 식별하는데 사용되며, 이들 오퍼레이션에서 생성 또는 사용되는 데이터들은 데이터 요소로 식별한다. 예를 들어 '상품등록한다'라는 것은 '상품등록한다'라는 오퍼레이션과 상품을 등록하고자 하는 'Seller', 등록하기 위한 'Goods'란 데이터 요소를 식별할 수 있다. (그림 5)는 (그림 4)의 유스케이스 모델에서 식별된 요소들이다.

3.2 서비스 개념 모델 생성

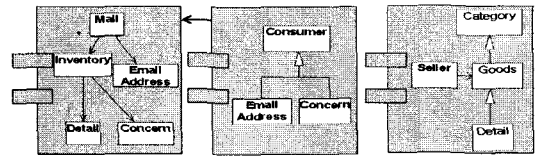
서비스 개념 모델은 데이터의 의존성 및 독립성을 식별하기 위해 사용되는 모델로서, 앞에서 식별된 데이터 요소를 사용하여 생성한다. 데이터 요소들간의 관계를 클래스 다이어그램을 사용하여 나타내며, 이 모델을 사용하여 데이터의 독립성과 의존성을 파악하여 핵심 데이터와 일반 데이터로 데이터를 구분할 수 있다. 이렇게 구분된 핵심데이터를 기준으로 컴포넌트를 식별할 수 있다.

(그림 6)은 서비스 개념 모델로써 앞 단계에서 식별된 데이터 요소로 생성된 클래스 다이어그램을 이용한 서비스 개념 모델이다. 핵심 데이터 요소로 Mail, Consumer, Seller를 식별하였고, 이들 데이터 간의 의존성으로 3개의 컴포넌트를 식별하였다.



(그림 5) 오퍼레이션과 데이터 요소

(그림 7)은 서비스 개념모델에서 식별된 컴포넌트를 알기 쉽도록 컴포넌트 다이어그램으로 나타낸 것이다. (그림 6)에서 핵심 자원 Consumer와 Mail은 서로 연관성을 가지므로 컴포넌트 다이어그램에서도 표시를 해주었다



(그림 7) 핵심 데이터를 중심으로 일반 컴포넌트 식별

3.3 서비스 운영 모델

서비스 운영 모델은 서비스가 제공하고자 하는 기능을 파악하기 위해 생성되는 모델로서, 서비스 요소에서 정의된 프로세스 요소를 중심으로 모델을 생성한다. 유스케이스 모델에서의 각 케이스에서 사용되는 오퍼레이션을 묶어 프로세스로 식별할 수 있으며, 식별된 프로세스들은, 각 프로세스간의 연관성에 의해 서비스로 묶여진다. (그림 8)은 식별된 오퍼레이션요소를 묶어 프로세스를 식별하고 이들의 연관성을 통해 서비스를 식별할 수 있는 모델이다.

상품을 등록하고 분류하는 오퍼레이션은 Seller란 하나의 프로세스에서 기능을 제공할 수 있으며, 다른 프로세스와는 연관성이 없으므로 하나의 서비스로 나타나며, Consumer와 Mail은 서로 연관을 가지므로 두 프로세스가 하나의 서비스로 식별된다.

Services	Process	Operation
Seller Service	Seller	상품을 Category 별로 분류한다
		상품을 등록한다
Consumer Service	Consumer	Mail 주소를 등록한다
		Concern을 등록한다.
	Mail	Concern과 Category를 비교한다
		메일을 전송한다

(그림 8) 서비스 운영 모델

3.4 기본 아키텍처 모델

기본 아키텍처 모델은 앞에서의 서비스 개념 모델과 서비스 운용 모델을 통합한 모델로써, 서비스 개념 모델에서 식별된 컴포넌트와 서비스 운용 모델에서 식별된 서비스의 연관성을 사용하여 생성된다. 모델의 상위에는 서비스가 위치하며 하위에는 컴포넌트들이 배치되는 구조를 가진다.

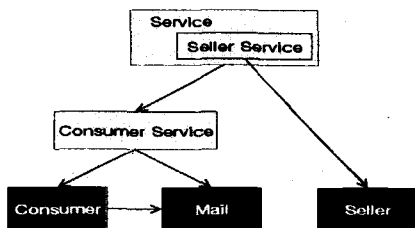
(그림 9)는 앞에서 생성된 모델을 통합하여 생성된 기본 아키텍처이다. 서비스 운용 모델에서 식별된 Consumer Service와 Seller Service가 상위에 위치하며 하위에는 서비스 개념 모델에서 식별된 컴포넌트들이 위치한다. 서비스 운영 모델에서의 프로세스는 개념 모델에서 식별된 컴포넌트와 매핑이 되며, 프로세스에서 가지고 있는 오퍼레이션은 컴포넌트와 인터페이스로 매핑 된다. Service는 내부에 Seller Service의 기능을 포함하고 Consumer Service와 상호작용을 할 수 있게 하여 전체를 하나의 서비스로 구성하였다.

3.5 서비스 지향 컴포넌트 아키텍처

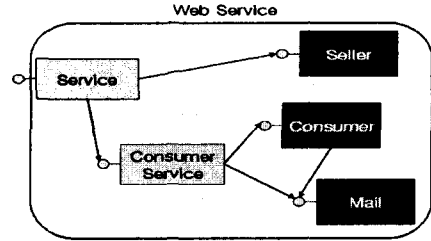
서비스 지향 컴포넌트 아키텍처는 기본 아키텍처 모델에서 웹 서비스를 구성하기 위해 컴포넌트를 배치하여 생성된다. 기본 아키텍처 모델에서 상위에서 서비스를 지원하는 Service와 Consumer Service는 Facade 컴포넌트로 식별을 할 수 있으며, 하위의 Consumer, Mail, Seller는 Backside 컴포넌트로 식별할 수 있다. (그림 10)은 기본 아키텍처 모델로부터 생성한 서비스 지향 컴포넌트 아키텍처이다. 하나의 웹 서비스를 제공하고 있으며 내부는 Facade 컴포넌트와 Backside 컴포넌트로 되어있다.

4. 결론 및 향후 연구

소프트웨어개발에서 CBD는 이미 개발 패러다임으로서 주류를 형성하였다. 현재는 독립적인 소프트웨어 개발 방법론으로 존재하기보다는 기본적인 기술 역량으로 간주되고 있다.



(그림 9) 기본 아키텍처 모델



(그림 10) 서비스 지향 컴포넌트 아키텍처

이러한 CBD 관련 제품들이 웹의 특성을 반영함으로써 웹 서비스로 자연스럽게 전이되는 추세이다. 하지만 기존에 개발된 웹의 특성을 반영하지 못한 컴포넌트들이 많이 존재하고 있다. 이러한 컴포넌트는 소프트웨어 아키텍처에 대한 의존성과 같은 문제들로 인해 만족할 만한 결과를 얻지 못하고 있다. 따라서 웹 서비스로 개발을 하기 위해서는 많은 시간과 노력이 요구될 것이다.

본 논문에서는 기존의 일반적인 컴포넌트를 사용하여 웹 서비스를 개발하기 위해 사용자의 요구사항으로부터 Facade 컴포넌트와 Backside 컴포넌트를 식별하기 위한 방법을 제시하였다. 이를 위해 서비스 요소를 오퍼레이션 요소와 데이터 요소 두 가지로 정의하였으며, 이들 요소를 사용한 모델을 정의하였다. 이들 모델을 통합하여 기본 아키텍처를 생성하며 여기서 식별된 컴포넌트들은 서비스 지향 아키텍처에서 웹 서비스의 구성을 Facade 컴포넌트와 Backside 컴포넌트로 하였다. 향후 연구로는 식별된 Facade 컴포넌트의 구현과, 일반적인 방법론으로 개발된 웹 서비스와의 평가가 수반되어야 할 것이다.

참고문헌

- [1] 첨단 SW 공학센터 SW Reuse 팀, CBD 동향보고서, 삼성 SDS IT REVIEW, 2003.
- [2] Jia Zhang and Jen-Yao Chung, "A SOAP-oriented component-based framework supporting device-independent multimedia Web services," Multimedia Software Engineering Proceedings, Vol. 50, No. 1, pp.40-47, 2002.
- [3] 강금석, 컴포넌트 추출 방법에 대한 연구, 삼성 SDS IT REVIEW, 2002.
- [4] 박동식, 신호준, 김행곤, "SOA기반의 Facade 컴포넌트에 관한 연구," 한국정보처리학회지, 10권, 1호, pp.1721-1724, 2003.
- [5] Richard Veryard, "Modeling for SOA," CBDi Journal, pp.11-18, February, 2003.
- [6] Stearns and Piccinelli, "Managing interaction concerns in Web-service systems," Proceedings of 22nd International Conference on Distributed Computing Systems Workshops, pp.424-429, 2002.