

## 콜택시 웹 서비스를 위한 서비스 중개자 설계 및 구현

이춘오<sup>0</sup> 장우혁 한동수

한국정보통신대학교

{lco<sup>0</sup>, torajim, dshan} @tci.ac.kr

### Design and Implementation of Service Broker for Call-Taxi Web Services

Choon oh Lee<sup>0</sup> Woo hyuk Jang Dong-soo Han

Information and Communications University

#### 요 약

서비스 지향 구조 (Service-Oriented Architecture)는 독립적으로 구성된 웹 서비스(Web Services)들을 프로세스 중심적으로 연결시키기에 적합한 소프트웨어 시스템 구성 방식이다. 하지만 SOA의 규모가 커질수록 제공되는 서비스들의 숫자 또한 증가하게 될 것이며 그런 상황에서 사용자가 원하는 서비스를 직접 찾아 이용하는 것은 불가능하다. UDDI는 사용자에게 기술적인 정보만을 제공하기 때문에 이러한 문제를 적절하게 해결할 수 없다. 그에 비해 서비스 통합(Service Integration)은 서비스들이 상호 동작하여 사용자에게 통합된 서비스를 제공할 수 있기 때문에 UDDI의 한계를 극복할 수 있다. 하지만 서비스 통합의 장점과 영향들은 아직 분명하게 규명하지 못하였다. 이 논문에서는 서비스 통합의 한 형태인 서비스 중개자(Service Broker)를 구현하고 평가하였다. 구현된 중개자의 비즈니스 모델은 콜택시 서비스이며, 콜택시 서비스 중개자는 접근성, 확장성, 재사용성, 유연성, 안정성 등에서 향상된 기능을 제공하였다.

#### 1. 서 론

서비스 지향 구조 (Service-Oriented Architecture)는 웹 서비스(Web Services)와 같이 독립적으로 구성된 서비스들을 프로세스 중심적으로 연결시키기에 적합한 소프트웨어 시스템 구성 방식이다. 많은 연구자들과 개발자들은 분산 소프트웨어 개발 방식으로 SOA를 주목하고 있는데, 이는 SOA가 다른 분산 구조인 CORBA나 DCOM, EJB와 달리 몇 가지 흥미로운 특징들을 가지기 때문이다. SOA의 분산 서비스들은 느슨하게 묶여진 (loose-coupled) 상호 동작을 지원하고 그 외에 구성요소의 편재 가능성 (Component Ubiquity), 플랫폼 독립성 (Platform independency), 민첩한 비즈니스 지원 (Business Agility), 구성요소 재사용 가능성 (Component Reusability) 등의 특징들을 가지고 있다 [1].

SOA가 현실화 된다면 다양한 곳에서 제공하는 수많은 서비스들이 각자 다른 환경에서 그들의 기능을 제공하게 될 것이다. 앞서 열거되었던 특징들은 사용자들에게 유용한 점들로 작용하리라 기대된다. 하지만 그에 맞물려 제기될 수 있는 SOA의 단점은 SOA 규모의 확장성 (Scalability) 이다. SOA의 규모가 커질수록 제공되는 서비스들의 숫자 또한 기하 급수적으로 증가할 것이며 그런 상황에서 사용자가 원하는 서비스를 직접 찾아 이용하는 것은 거의 불가능하다.

현재 표준화 (Standardized)된 서비스 저장소인 UDDI (Universal Description, Discovery, and Integration)[2]는 위의 문제를 적절하게 해결할 수 없는데 이는 기본적으로 UDDI가 웹 서비스의 표현 언어인 WSDL (Web Service Description Language)[3]의 제공만을 목적으로 하고 있기

때문이다. 즉, 사용자는 UDDI로부터 기술적인 정보만을 담고 있는 WSDL을 제공 받아 WSDL이 가리키는 서비스를 직접 호출해야만 한다. UDDI에 의해 제공되는 서비스 정보가 부족하기 때문에 사용자는 원하는 서비스를 한 번에 검색하기 위해 많은 수고를 들여야만 한다. 특히 사용자가 여러 개의 서비스들이 필요로 하는 경우, 현재의 SOA는 사용자의 정확하고 편리한 서비스 이용의 보장하기 매우 어렵다.

서비스 통합(Service Integration)은 위와 같은 문제를 실용적으로 해결 할 수 있다. 서비스 통합은 독립적인 동, 이종의 서비스가 서로 협업하여 또 하나의 서비스를 제공하는 것을 말한다. 하지만 서비스 통합을 위한 연구들은 아직 표준으로 책정될 수 있을 만큼의 성과를 내지 못하였으며, 통합의 장점들과 영향들 또한 규명되지 못하였다.

따라서 본 논문에서는 서비스 통합의 한 형태인 서비스 중개자의 구현과 평가에 대해 다룬다. 서비스 중개자는 서비스 통합의 한 종류로서, 동종의 서비스를 통합하여 서비스 사용자에게 일관적이고 통일적인 서비스 인터페이스를 제공하는 역할을 한다. 서비스 중개자는 유용한 서비스들을 미리 모아 체계적으로 관리하고 그들의 호출을 전담하기 때문에 서비스 사용자들에게 UDDI보다 좀 더 편리한 서비스 호출의 방법을 제시할 수 있다. 그 외에 서비스 중개자는 SOA의 접근성, 확장성, 재사용성, 유연성, 안정성 등을 향상시킬 수 있도록 디자인되었다.

구현된 서비스 중개자의 비즈니스 모델은 콜택시 서비스이다. 구현된 콜택시 서비스 중개자는 콜택시 고객에게 안정되고 종합된 통합 콜택시 서비스를 제공하였

다. 또한 콜택시 서비스 중개자는 앞서 언급되었던 중개자의 디자인 목표들을 만족시키는 데 성공했다

2. 구조

서비스 통합 구조 (Service Integration Architecture)는 독립적으로 동작 가능한 서비스 제공자와 서비스 중개자, 서비스 사용자 등의 3부분으로 나뉠 수 있다.

서비스 제공자는 기본적인 웹 서비스를 게시하고 제공하는 서비스 종결점 (Service End-point)이다. 제공자의 서비스는 웹 서비스 기술을 사용해 구현되기 때문에 서비스의 사용자에게 충분한 서비스 개요(Service Abstraction)와 투명성(Transparency)을 제공할 수 있다.

서비스 중개자는 다른 제공자들의 서비스들을 통합한 또 하나의 웹 서비스를 제공하는 역할을 한다. 중개자는 사용자의 요청을 받아 사용자 대신 서비스 제공자들을 호출하고 그들의 결과를 종합, 정리하여 사용자에게 다시 전달한다.

마지막으로 서비스 사용자는 서비스 제공자들을 하나 하나 호출하는 대신 서비스 제공자의 대리 서비스를 이용한다.

이 구조의 가장 큰 장점은 여러 번 반복되어야 하는 서비스 호출의 수고가 줄어들다는 것이다. 보통의 SOA의 웹 서비스 통신들은 그림 1에서 묘사한 바와 같이 수많은 서비스 제공자들과 사용자들 사이의 상호 작용에 의해서 수행된다. 하지만 그림 1의 사용자들은 서비스를 호출하기 위해 필요한 모든 정보들과 소스 코드들을 항상 가지고 있어야 하고, 하나의 웹 서비스의 인터페이스가 아주 조금 바뀌어도 그것이 모든 사용자에게 반영되어야 한다. 이는 SOA 구조의 구축과 유지, 보수에 많은 비용을 낭비될 수 있음을 의미한다. 그와 반대로 중개자가 삽입된 그림 2는 명확한 상호 작용의 감소를 보여준다.

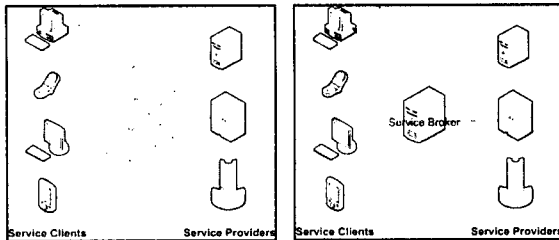


그림 1. 보통 SOA의 상호동작      그림 2. Broker가 삽입된 SOA

서비스 중개자로부터 기대할 수 있는 효과들은 다음과 같다.

- 접근성: 서비스 중개자는 서비스 제공자들의 인터페이스를 하나로 통일하기 때문에 향상된 접근성을 제공할 수 있다.
- 확장성: 서비스 중개자는 분산 (distribution) 과 중복 (duplication) 배치로 구조의 규모 확장을 효과적으로 지원할 수 있다.
- 재사용성: 서비스 중개자와 서비스 제공자들은 다른 서비스들에 의해 쉽게 이용될 수 있다.
- 유연성: 서비스 중개자는 서비스 제공자들의 인터페이스가 바뀌었을 경우 즉각 그에 반응하여 사용자의 불편을 최소화 시킬 수 있다.

- 안정성: 서비스 중개자는 서비스 호출의 실패가 사용자에게 영향을 미치지 못하도록 할 수 있다.

3. 디자인과 구현

서비스 중개자를 검증하기 위해 구현된 서비스 통합의 예제는 콜 택시 서비스이다. 현재 콜 택시는 일반 개인 택시 기사들의 연합으로 구성되며, 사용자는 하나의 택시 연합에 전화를 걸어 택시 서비스를 받는 것이 일반적이다. 하지만 사용자는 어디까지나 '선택한 택시 연합 내에서 가장 가까운 택시'를 제공 받을 수 있을 뿐 '모든 연합의 택시들 중 가장 가까운 택시'를 제공 받을 수는 없다. 하지만 서비스 중개자가 모든 연합의 호출을 전담한다면 사용자는 큰 수고 없이 '가장 가까운 택시'를 제공 받을 수 있다.

이 논문은 콜 택시 연합을 하나의 서비스 제공자로 가정하고 이들을 통합하는 서비스 중개자 구조를 가상적으로 구현하였다. 택시 서비스 중개자는 사용자에게 가능한 모든 택시의 리스트를 제공하고 특정 택시를 호출할 수 있는 등의 기능들을 가진 단일 서비스를 제공한다.

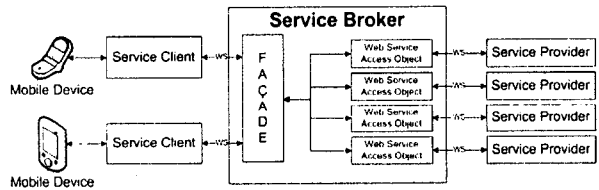


그림 3 택시 중개자 소프트웨어 구조

그림 3 은 택시 중개자 소프트웨어 (Taxi Broker Software)의 전반적인 구조를 그린 것이다. 이 소프트웨어는 중개자 구조에 입각하여 크게 복수의 택시 서비스 제공자들과 택시 서비스 중개자, 서비스 사용자들 등의 세 부분으로 나뉠 수 있다.

각각의 택시 서비스 제공자들은 지역 콜택시 연합의 온라인 택시 호출을 위한 서비스를 게시한다. 이 서비스는 크게 두 가지 함수를 제공하는데, 첫번째 함수는 고객의 위치를 받아 현재 호출 가능한 택시의 리스트를 제공해 주는 역할을 하며, 두번째로는 선택된 택시를 호출하는 역할을 하는 함수가 있다.

택시 서비스 중개자는 게시된 택시 서비스들을 호출하고 호출 가능한 택시 정보를 모두 수집, 재정리하여 사용자에게 보내주는 역할을 한다. 이 복잡한 프로세스 또한 하나의 웹 서비스로 게시되어 사용자에게 투명성과 편리성을 제공한다.

마지막으로 서비스 사용자는 중개자의 택시 호출 서비스를 하나의 단순한 웹 서비스로 보고 그를 사용하는 모바일 프로그램이다.

표 1은 세 부분의 상호 동작으로 도출된 택시 중개자 소프트웨어의 성공적인 실행 시나리오를 나타낸다.

순서	주 제	동 작
1	고객	모바일 장치의 택시 호출 프로그램을 시작
2	고객	호출 가능한 택시의 리스트를 요청
3	사용자 프로그램	고객의 현재 위치를 이용해 택시 서비스 중개자의 서비스를 요청

4	서비스 중개자	사용자 프로그램의 요청을 해석하고 택시 서비스 제공자들의 서비스를 대신 호출
5	서비스 제공자	서비스 중개자로부터 요청받음
6	서비스 제공자	고객으로부터 가장 가까운 빈 택시들의 리스트를 제작해 중개자에게 전달
7	서비스 중개자	많은 제공자들의 결과를 종합, 제정의, 정렬함
8	서비스 중개자	사용자 프로그램에게 종합된 결과를 전달
9	사용자 프로그램	받은 택시 리스트를 고객에게 보여줌
10	고객	리스트 중 하나를 선택하고 택시를 호출
11	사용자 프로그램	선택된 택시를 서비스 중개자에게 전달
12	서비스 중개자	선택된 택시를 제공한 서비스 제공자에게 선택된 택시의 호출을 요청
13	서비스 제공자	선택된 택시의 요청을 받아 택시 호출을 시도
14	서비스 제공자	서비스 중개자에게 호출 성공 여부를 알림
15	서비스 중개자	사용자 프로그램에게 호출 성공 여부를 알림
16	사용자 프로그램	고객에게 택시 호출 성공 여부를 보고

4. 결과 및 평가

실험 환경에는 두 개의 서비스 제공자들과 하나의 서비스 제공자, 그리고 하나의 서비스 사용자가 존재한다. 서비스 제공자들은 프로그램 시작 시 가상적으로 택시들을 만들어 내며 택시들의 위치를 임의로 바꾸어 택시의 이동을 반영한다. 서비스 제공자와 서비스 중개자는 J2EE의 웹 서비스 환경 위에서 구현되고 게시 되었으며, 실험을 위해 모두 로컬 컴퓨터 상에서 실행되었다.

그림 4는 서비스 사용자의 웹 서비스 호출 프로그램의 성공적인 결과를 보여준다.

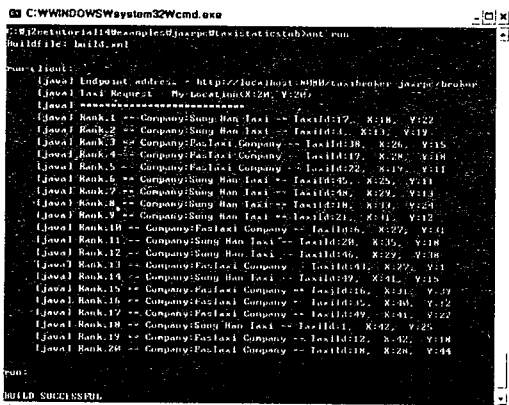


그림 4의 결과에서 보이는 바와 같이, 사용자 프로그램은 서로 다른 두 택시 회사의 택시들을 제공하였다. 택시들은 고객과 택시 거리의 올림 차순으로 정렬되었기 때문에 가장 맨 위에 위치한 Rank.1의 택시가 가장 고객과 가까운 택시라고 할 수 있다. 또한 택시의 고유 번호가 결과에 포함되어 있어 사용자는 고유 번호를 택시 호출에 사용할 수 있다.

이 실험을 통하여 서비스 중개자를 검증한 결과는 다음과 같다.

- 사용자 프로그램의 간략화: 웹 서비스의 호출이 복잡한 (자바 오브젝트를 주고 받는) 경우 사용자는 그만큼 복잡한 스텝(Stub)을 가질 수 밖에 없다. 중개자 구조의 사용자 프로그램은 아주 간단한 자바 어플리케이션 (Java Application)이며 오직 서비스 중개자를 호출하기 위한 스텝만을 가지고 있다. 모든 서비스 제공자들을 위한 스텝을 다 가질 필요가 없기 때문에 사용자 프로그램이 가벼워 졌다.
- 서비스 실패의 은닉: 사용자는 언제나 실패 없이 호출 가능한 택시 리스트를 보장 받을 수 있다. 한 서비스 제공자의 실패는 단지 전체 결과의 숫자만 줄일 뿐이며 택시 호출에 큰 영향을 미치지 못한다.
- 최선의 선택 보장: 중개자가 모든 택시 서비스 제공자의 결과를 하나로 종합하였기 때문에 고객은 제공자에 관계없이 자신과 가장 가까운 택시를 호출할 수 있다.

또한 이 논문은 중개자 구현을 통해 현 웹 서비스 구현 프레임워크 (implementation framework)의 한계점들을 지적하였다. 그 중 가장 큰 문제는 현재 웹 서비스의 표준의 구현을 제공하는 프레임워크들이 아직 서비스 통합과 협업을 지원하기 위한 효과적인 해결책을 완벽하게 제시하지 못했다는 것이다. 즉 다시 말해 현재의 표준은 서비스 요청자와 서비스 제공자 등의 2단 (two tiers)의 구조만을 지원하고 있으며, 서비스 중개자는 서비스 요청자의 특별한 형태로만 여겨지고 있다. 그 결과 서비스 중개자는 새로운 서비스 요청자를 실행 도중 추가하는 등의 동적인 특성을 가질 수 없다. 이는 서비스 통합의 중요성이 증대되어 표준과 구현 프레임워크에 차후 적용되어야 한다고 본다.

5. 결론

SOA(서비스 지향 구조)의 중개자 구조는 웹 서비스들이 신중하게 종합된 중개자 웹 서비스를 제공해 SOA의 서비스 호출과 이용을 좀 더 쉽고 편리하게 만드는 역할을 한다. 이 논문에서는 실제로 중개자 구조를 구현해 그 실용성을 검증하고자 하였으며 그 결과 서비스 중개자는 서비스 사용자와 서비스 제공자 간의 통신을 간단하고 편리하며 안전하게 만드는 성과를 거두었다.

6. 참고 문헌

- [1] Web Services Concepts - A technical overview, Hewlett-Packard Company, 2001
- [2] Luc Clement, Andrew Hatley, Claus von Riegen, Tony Rogers, UDDI Spec Technical Committee, OASIS, 2004
- [3] Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, Web Services Description Language 1.1, W3C, 2001
- [4] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
- [5] Sun Microsystems, Core J2EE Patterns - Data Access Object, Sun Developer Network, 2002