

SOA에서의 서비스 연결 특성 고찰

윤희진*, 최병주**

협성대학교 컴퓨터공학과*, 이화여자대학교 컴퓨터학과**

A Study on Connections of Services under SOA

Hoijin Yoon

Hyupsung University

E-mail : hjyoon@uhs.ac.kr

요 약

비즈니스 프로세스를 반영하는 엔터프라이즈 시스템 구현에 SOA가 주목받고 있다. 최근 많은 기업들에서 기존의 시스템을 SOA로 재구축하거나, 새로운 기능을 SOA기반으로 구축하려는 움직임이 활발하다. 그러나 이러한 프로젝트들이 현실적으로 실제 시스템으로 사용되기 어려운 점은 그에 대한 시험방법의 부재이다. 서비스 단위 테스트는 기존의 구현 코드 기반 또는 블랙박스 테스트 방법으로 접근 가능하나, 서비스들의 통합 테스트는 기존의 방법 적용과 더불어 SOA의 서비스 연결 개념을 정확하게 반영해야 할 필요가 있다. 서비스의 일대일 통합이 아닌, 전반적인 서비스들의 연결 구조를 분석하여 서비스 연결 부분에 대한 테스트 방안을 도출하여야 한다. 본 논문은 그 시작으로 SOA 패러다임을 적용한 서비스 연결 구조를 분석하는 방법을 제안한다. 이는 SOA가 주장하는 비즈니스 프로세스와 어플리케이션 로직과의 분리와 서비스들 사이의 느슨한 연결 규칙을 준수하는 연결 구조를 대상으로 한다.

1. 서론

Gartner Group에 의해 SOA(Service-Oriented Architecture)가 대두된 이후[1] 지금까지 많은 기업들이 자신들의 시스템을 SOA로 구축하는 일에 관심을 보이고 있다. 국내 시장에서도 지난해부터 서비스 기반 구조 파일럿 프로젝트가 기하급수적으로 늘고 있지만, 기대와는 달리 파일럿 프로젝트들은 연이어 본 프로젝트로 연결되지 못하고 있다. 그 이유는 SOA 시스템의 연결 테스트가 미흡하고 보안에 대한 투자가 함께 이루어지지 않았기 때문이다[2]. 결국 향후 SOA 개발에서 테스트가 문제가 될 것이며, SOA의 개념상 서비스 컴포넌트 사이의 연결 테스트가 매우 중요하다는 것이다.

SOA 도입으로 인한 주요 효과는 통합(integration)이 갖는 오래된 문제를 해결하는 것이다. 여러 분석에 따르면, 전형적인 IT 예산의 최고 30%가 통합 활동에 할당되고 있다[3]. 이때 말하는

통합이라는 것은 프로세스 통합, 기업체 통합, 그리고 M&A(Mergers and Acquisitions) 통합 등을 포함한다. 만일 이러한 통합에 쓰일 비용이 조금 줄어든다면, 남은 예산을 프로젝트의 전략을 위한 관리에 쓸 수 있게 된다. SOA는 이러한 프로젝트 전략을 지원한다. 이는 비즈니스 프로세스와 IT의 통합을 위한 전략을 의미한다.

이런 면에서, 비즈니스 프로세스를 반영하는 엔터프라이즈 시스템의 요구사항들을 SOA가 충족시키고 있다. 최근 많은 기업들에서 기존의 시스템을 SOA로 재구축하거나, 새로운 기능을 SOA기반으로 구축하려는 움직임이 활발하다. 그러나 이러한 프로젝트들이 현실적으로 실제 시스템으로 사용되기 어려운 점은 그에 대한 테스트 방법의 부재이다. 서비스 단위 테스트는 기존의 구현 코드 기반 또는 블랙박스 테스트 방법으로 접근 가능하나, 서비스들의 통합 테스트는 기존의 방법 적용과 더불어

어 SOA의 서비스 연결 개념을 정확하게 반영해야 할 필요가 있다. 서비스의 일대일 통합이 아닌, 전반적인 서비스들의 연결 구조를 분석하여 서비스 연결 부분에 대한 테스트 방안을 도출하여야 한다.

본 논문은 그 시작으로 SOA 패러다임을 적용한 서비스 연결 구조를 분석한다. 이는 SOA가 주장하는 비즈니스 프로세스와 어플리케이션 로직과의 분리와 서비스들 사이의 느슨한 연결 규칙을 준수하는 연결 구조를 대상으로 한다.

2. SOA에서의 서비스 구성

SOA는 통합이라는 측면에서, 다음 두 가지 중요한 특징을 갖는다. 첫째, SOA 통합은 기존의 강한 결합을 일으키는 방법이 아닌, 물리적으로 다른 공간에 존재하면서 RPC 통신 또는 데이터 통신을 SOAP을 통하여 하는 통합을 의미한다[4]. 이는 느슨한 결합을 이끌어낸다. 따라서 기존의 통합과는 다른 의미로서 "연결"이라는 용어로 표현하는 경향이 있다. 둘째, 연결은 비즈니스 서비스와 어플리케이션 서비스 사이에 존재한다. 동일한 종류의 서비스들, 즉 비즈니스 서비스들 또는 어플리케이션 서비스들, 사이에서는 연결이 존재하지 않도록 한다. 만일 어플리케이션 수준의 연결이 존재하였다면 그 연결단위 전체를 하나의 서비스로 정의해야 한다. 그렇게 함으로써 SOA의 연결 원칙을 지킬 수 있게 된다. 1장에서 언급한대로 SOA는 IT와 비즈니스 프로세스를 위한 요구사항 모두를 다루어야 한다. 따라서 SOA 조직은 두 가지 종류의 서비스를 가지게 된다. 하나는 비즈니스 프로세스를 반영하는 비즈니스 서비스이고, 또 다른 하나는 프로세스 기능 수행을 위해 비즈니스 서비스가 사용하는 다양한 IT 서비스들을 의미하는 어플리케이션 서비스이다.

SOA는 이들 서비스들의 특징에 따라 서로 다른 계층으로 배치하여, 이들의 연결 구조를 그림 1과 같이 구축한다[5, 6].

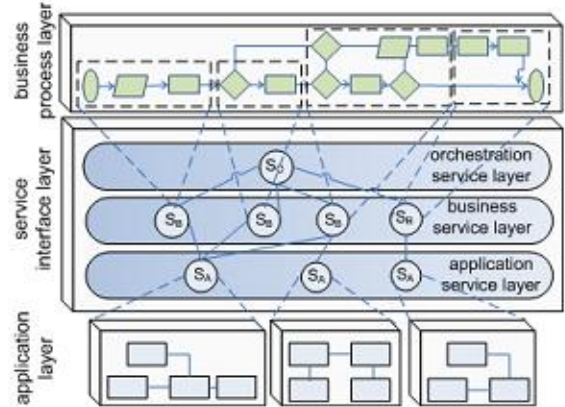


그림 1. SOA의 계층구조와 서비스들의 배치

그림 1에서 보듯이 비즈니스 서비스는 비즈니스 서비스 계층에 위치되어 있으면서 상위의 비즈니스 프로세스를 반영하고 있다. 어플리케이션 서비스 역시 자신만의 어플리케이션 서비스 계층에 위치하고 있으면서, 하단의 어플리케이션들을 참조하고 있다. 전체적인 비즈니스 프로세스를 수행하기 위하여 비즈니스 서비스들이 수행되고, 이 비즈니스 서비스들은 해당 어플리케이션에게 메시지를 전달함으로써, 구체적인 구현 단위인 하위 어플리케이션들을 구동시킨다. 이를 통한 수행 결과는 어플리케이션 서비스를 거쳐, 요청메시지를 처음에 보낸 비즈니스 서비스에게로 응답되어진다. 이러한 전반적인 흐름에 덧붙여, 조정(orchestration) 서비스 계층에는 여러 가지 프로세스 서비스들이 위치될 수 있다. 이는 기존의 제1세대 웹서비스 기술에 덧붙여 이후 제안된 WS-*들을 구현한 서비스들에 해당된다. 미들웨어 계층이라고 말할 수 있다. SOA 성숙도 모델[7, 8]에서의 성숙도가 높은 기업일수록, 그림 1의 모든 계층을 갖는 SOA를 구축할 가능성이 높다.

기존의 IT 중심의 웹서비스들을 WSDL, SOAP, UDDI등의 기술을 기반으로 엮어서 구현한 시스템들은 대부분 어플리케이션 서비스 계층만을 갖는다. 여기에 비즈니스 프로세스의 개념이 부가되어 비즈니스 서비스까지 구현되는 경우는 이전보다 좀 더 성숙한 SOA라고 평가될 수 있으며, 나아가 조정 서비스 계층에 하위계층 서비스들의 운영을 조정하는 WS-BPEL, WS-Coordination 등을 구현하는 프로세스 서비스를 갖도록 할 수 있다. 이 경우가 이상적인 SOA 구조라고 할 수 있다[3, 6].

3. 서비스 연결 분석

SOA 시스템 구축에 대한 여러 방법들이 상향식 또는 하향식 접근등으로 제안되고 있다[6]. 이 두 가지 방법 모두 비즈니스 서비스 계층과 어플리케이션 서비스 계층을 구성하며, 이들 계층에 오는 서비스들은 다음의 특성에 맞게 구현되어야 한다. 표1은 그림1에 등장한 계층을 이루는 단위들에 대한 설명이다.

표 1. SOA 구성단위들의 분석 결과

SOA 구성 단위	의미
비즈니스 프로세스	일련의 작업(work)들의 집합
비즈니스 서비스	연관된 작업들의 모임
어플리케이션 서비스	연관된 오퍼레이션들의 모임
어플리케이션	일련의 오퍼레이션들의 집합

표1을 살펴보면, 비즈니스 서비스는 비즈니스 프로세스를 이루는 작업들로 이루어지며, 이는 해당 작업을 수행하기 위하여 어플리케이션 서비스에게 요청 메시지를 전송하게 된다. 어플리케이션 서비스는 비즈니스 프로세스 또는 그의 작업과는 전혀 무관하며 단지 어플리케이션들이 갖는 오퍼레이션들을 모아서 구성한 서비스 단위이다. 비즈니스 서비스로부터 요청 메시지를 받게 되면 어플리케이션 서비스는 그와 연결된 오퍼레이션을 수행한다. 이때 실제 오퍼레이션의 구현 객체는 또 다른 제3의 장소에 있을 수도 있다. 그렇다면 이때도 물론 SOAP을 통한 메시지 전송으로 요청 메시지가 전송되며, 수행된 결과가 응답 메시지로 되어 해당 어플리케이션 서비스에게 보내진다.

따라서 하나의 트랜잭션에 대하여 비즈니스 서비스가 어플리케이션 서비스에게 요청메시지를 보내고 어플리케이션 서비스가 하위계층의 어플리케이션들로부터 결과를 받아 비즈니스 서비스에게 응답메시지를 보내는 연결 구조가 존재한다. 이렇게 함으로써, 비즈니스 프로세스에 대하여 추상화 정도가 다르도록 계층을 구성하게 되며, 이것이 SOA가 적용될 B2B등에서의 요구사항과 일치한다.

지금까지 설명한 서비스 연결 특성에 따르면, 각 서비스들 사이의 연결은 구체적으로, 한 비즈니스 서비스의 작업과 그 작업을 위한 어플리케이션 서비스의 오퍼레이션 사이의 연결로 정의할 수 있

다. 따라서 연결은 한 쌍의 (비즈니스 서비스의 작업, 어플리케이션 서비스의 오퍼레이션)으로 정의된다. 다음 장에서는 특정 회사의 SOA 서비스 연결 구조를 분석한다.

4. 적용 및 결론

RailCo은 에어 브레이크를 판매하는 회사로 비효율적인 비즈니스 프로세스를 자동화하는 데 구식 기술을 활용하고 있었고, 그로 인해 고객들은 B2B 솔루션을 통해 동일 제품을 더 효율적이고 더 저렴하게 제공하는 경쟁사로 이탈해 갔다. 주요 고객인 TLS사 또한 경쟁사들과 온라인 관계를 형성을 시작하게 되면서 RailCo은 TLS와 거래를 위해 TLS만을 위한 웹 서비스를 서둘러 구축했고 그로 인해, 다른 고객들과 거래하기에는 적합하지 않았으며 SOA로의 설계에서 많은 계획에 차질이 생겼다. 우선 RailCo은 상향식 접근법을 택했고, 조정을 제공하는 미들웨어에 투자할 예산이나 기술이 없었으므로, 조정 서비스 레이어가 비즈니스 로직을 관리하지 않도록 결정했고, 대신 태스크 중심 비즈니스 서비스가 애플리케이션 서비스들을 컨트롤할 수 있는 컨트롤러가 되어, 각 비즈니스 프로세스를 수행하도록 했고, 장차 테스트 중심 비즈니스 서비스들을 오케스트레이션 서비스 레이어로 대체할 계획이다[6]. 그림 2는 RailCo가 갖는 서비스의 계층 배치도이다.

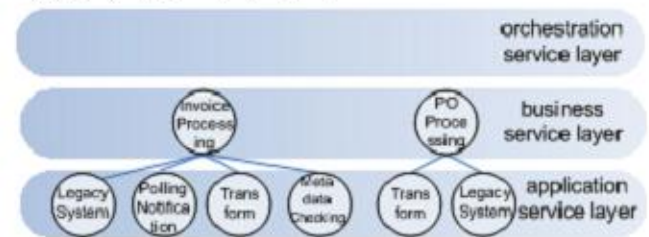


그림 2. RailCo의 서비스 계층 구조

RailCo에서 비즈니스 서비스와 어플리케이션 서비스의 연결 분석을 위하여 우선 그림 2의 비즈니스 서비스들을 작업단위로 분할하고, 어플리케이션 서비스들을 오퍼레이션 단위로 분할하였다. 그리고 난 후 어떤 작업과 어떤 오퍼레이션이 서로 연결 관계를 갖고 있는지를 표시하였다. 그 결과가 표2에 나타나있다.

표2에 등장하는 여러 연결들 가운데 하나를 예로 들어보면, "send an electronic invoice to Service" 라는 비즈니스 서비스의 하나의 작업은 "check if it is time to verify TLS metadata; if required, perform metadata check"라는 어플리케이션 서비스의 오퍼레이션과 메시지 전송을 통하여 연결된다는 것을 알 수 있다. 표2의 분석 결과를 이용하여 서비스 연결 테스트의 테스트 요구사항 추론에 사용될 수 있으며, 이를 통하여 보다 연결 구조에 충실한 연결 테스트 데이터를 생성할 수 있는 기본 방안을 마련하였다.

본 연구는 이어서 조정 서비스 계층에 존재하는 프로세스 서비스들과의 연결 관계를 분석하여 성능도가 높은 이상적인 SOA 구조에 적합한 연결 구조를 분석하고 있다. 이는 연결 테스트의 테스트 요구사항 생성에 적용될 것이며, 이를 통하여 궁극적으로 1장에서 언급한 본 논문의 진행 계기였던 SOA 서비스들 사이의 연결 테스트 필요성을 만족시키는 방법을 개발하고 있다.

[참고문헌]

[1]Gartner, "'Service Oriented' Architectures, Part 1," SSA Research Note SPA-401-068, April 1996

[2] "HP, BPM를 등 자동화 솔루션 도입 증가 예상," 디지털데일리, 8 March 2007

[3] Eric A. Mark, A Planning and Implementation Guide for Business and Technology, John Wiley and Sons, 2006.

[4] Jeff Offutt and Juan Luo, "Testing Web Services by XML Perturbation, Wuzhi Xu," *Proceeding on IEEE International Symposium on Software Reliability Engineering*, pages 257-266, November 2005, Chicago Illinois.

[5] Thomas Erl, Service-Oriented Architecture - A Field Guide to integrating XML and Web services, Prentice Hall, 2004

[6] Thomas Erl, Service-Oriented Architecture - Concepts, Prentice Hall, 2005

[7] Kunal Mittal, *Service Oriented Architecture Maturity Model*, e-book on <http://www.kunalmittal.com/html/soamm.shtml>, 2006

[8] Theo Beack, "SOA Maturity Model: Compass on the SOA Journey," SOAInstitute.org, Apr. 2006

표 2. 서비스 연결 분석표

		S ₄₁ - Legacy System		S ₄₂ - Polling Notification		S ₄₃ - Transform Accounting Documents		S ₄₄ - MetadataChecking		Application Service
		S ₄₁₁ - export document to network folder	S ₄₁₂ - import and forward document to work queue	S ₄₂₁ - poll folder for new documents	S ₄₂₂ - if documents arrive for which there are subscribers, issue notifications	S ₄₃₁ - transform XML documents to native format	S ₄₃₂ - transform native documents to XML	S ₄₄₁ - check if it is time to verify TLS metadata; if required, perform metadata check	S ₄₄₂ - if metadata check fails, issue notification	Operation
S ₆₁ - Invoice Submission Service	S ₆₁₁ - issues messages compliant with the accounts payable WSDL	v								
	S ₆₁₂ - request message	v			v	v			v	
	S ₆₁₃ - send an electronic invoice to Service	v		v				v		
S ₆₂ - Order Fulfillment	S ₆₂₁ - TLS Purchase Order Service		v				v			
Business Service	Work									