

비즈니스 통합의 표준으로서 “서비스 지향 아키텍처” 구축에 관한 연구 Implementing SOA (Service Oriented Architecture) as an upcoming IT standard for Business Integration

조재훈*, 최우용**, 이상완**

*한국 아이비엠, 소프트웨어 사업부, jhcho@kr.ibm.com

**동아 대학교 산업경영공학과, wychoi77@dau.ac.kr, swlee@dau.ac.kr

Abstract

To survive in rapidly changing business environment, the enterprise should adapt its business process to the changing environment. The system integration is an important issue to tackle this challenge. It is unavoidable direction that SOA (Service Oriented Architecture) is accepted as an upcoming IT standard to support flexible business processes and integration. In this paper, SOA is introduced, and the Web services and ESB (Enterprise Service Bus) are also introduced as a foundation technology to implement SOA-enabled solutions. The pilot project, which is introduced in this paper, will be a good reference for future SOA implementation.

Keywords: SOA, Web Services, ESB, System Integration

1. 서론

정보 시스템은 수작업을 줄이기 위한 목적으로 출발하였으나, 시스템 간의 통합이 원활히 이루어 지지 않아 오히려 많은 문제를 일으키고 있어, 정보 시스템의 통합은 필수 요소가 되고 있다. 기업의 사업 환경이 복잡해지고 변화함에 따라 기업의 지속적인 성장을 위해서는 업무 절차나 흐름을 수시로 변화시킬 수 있는 체제를 갖추어야 한다.

본 논문에서는 이러한 문제를 해결하기

위한 해결책으로, 전세계적으로 활발히 논의되고 있는 SOA에 대해서 여러 발표 자료를 바탕으로 소개하고, 이의 가장 중요한 기반 기술인 웹 서비스와 ESB에 대해 정리하고, SOA를 구축한 사례를 소개한다.

2. SOA (Service Oriented Architecture)

2-1 아키텍처 와 아키텍처 층(層)

아키텍처란 완성된 구조물의 전체 모양뿐 아니라 그 조각들이 어떻게 연계되어 있는지를 나타내는 개념이다. 소프트웨어 아키텍처는 구성 요소를 정의하고 이들 구성 요소의 기능, 연결 관계, 기술적인 구조, 제약 사항, 특징 등을 설명한다.[1]

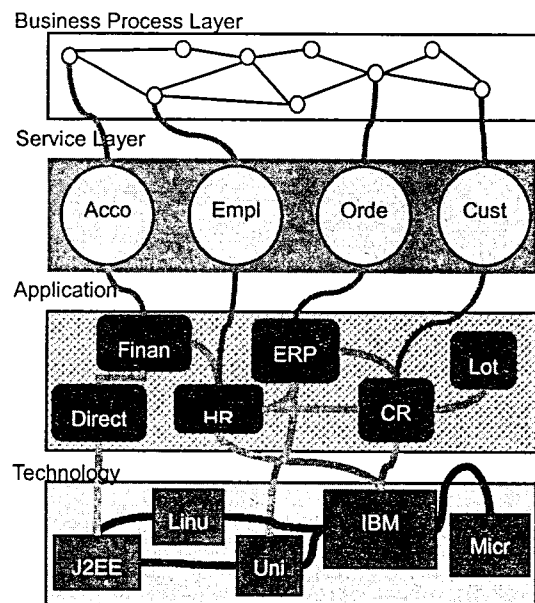


그림 1 아키텍처 층 [3]

통합 측면에서 IT 아키텍처는 그림 1과 같이 4단계 층으로 구분해 볼 수 있다. 통합은 모든 층에서 수행되지만 방식이나 내용은 다르다. 기술 층에서의 통합은 물리적인 기술의 연결이고, 응용 프로그램 층에서의 통합은 구현된 응용 프로그램이나 구성 요소의 연계를 의미하고, 서비스 층은 자원들로부터 독립적이다. 업무 처리 층에서의 통합은 서비스들만을 고려 하므로 어떤 변화가 생겨도 하위 층에서의 변화는 필요하지 않다. 아키텍처 층을 분리하는 가장 중요한 이유는 하위 계층의 변경에 무관하도록 변경에 따른 영향을 최소화하는 것이다.

2-2 SOA의 정의

Mark Colan에 의하면 SOA란 “서비스를 요청하거나 제공하여 응용 프로그램 기능이 가능하도록 하는 정책이나 관례 또는 골격이다” [7] SOA는 서비스로 구성되어 있으며 개방된 표준을 사용하여 서비스간의 의사 소통을 하는 아키텍처이다 현재 SOA를 구현하기 위한 기술로 가장 주목 받고 있는 것이 웹 서비스와 ESB이다.

2-3 SOA에 대한 전망

Gartner에 의하면, SOA로의 변화는 과거 단말기 기반 아키텍처에서 Client/Server 아키텍처로의 변화에 견줄만한 사고 틀의 변화를 가져올 것으로 예측한다 다음은 SOA에 대한 Gartner의 예측이다. [4]

- 2008년까지, 새로운 개발 프로젝트의 80%는 SOA일 것이다. (0.8 probability).
- 2008년까지, SOA는 코드의 재사용 율을 100%이상 증가시킬 것이다. (0.8 probability).
- 2010년까지, 소프트웨어 매출 성장의 80%는 SOA에 기반한다 (0.7 probability).

3. 웹 서비스 (Web Services)

웹 서비스는 개방형 표준을 지향하는 기술로써, 인터넷 네트워크를 통해 다수의 비즈니스 시스템들을 표준화된 기술로 결합시켜 고객이 원하는 정보나 기능을 구현하는 기술이다.

그림 2는 웹 서비스의 기본 개념과 그 기술이다. 웹 서비스는 서비스 제공자, 요청자, 중개자간의 세 가지 오퍼레이션 (Publish, Find, Bind)을 통하여 정의된다. 이때 각각 WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration), SOAP(Simple Object Access Protocol)과 같은 표준 기술들이 사용된다. 웹 서비스는 어떤 어플리케이션 서비스를 중재자에 등록하고 등록된 서비스를 공급자로부터 요청자로 전달하는 과정을 통하여 구현된다.

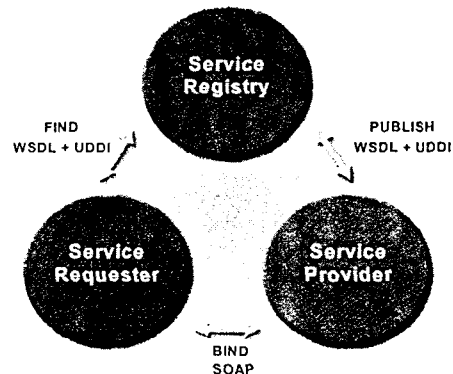


그림 2 웹 서비스 기본 개념과 기술 [7]

4. ESB (Enterprise Service Bus)

ESB는 SOA를 구현하기 위한 새로운 개념의 미들웨어로서, 여러 장소에 걸쳐 다양한 어플리케이션 유형간 최적의 방식으로 정보를 배포할 수 있는 아키텍처 패턴이다. [5]

4-1 ESB의 개념과 특징

ESB의 특징 중 중요한 하나는

어플리케이션 및 그 개발자가 인프라상의 서비스 또는 리소스의 위치에 신경 쓸 필요가 없다는 것이다. 즉 ESB가 있으면 이러한 서비스 및 리소스에 쉽게 연결할 수 있다. [2] ESB는 단순히 서비스들을 연결시켜주는 제품이 아니고, 개념적인 구성으로서, 서비스들이 연결되는 SOA 고속도로이다. 이 고속도로는 지역이나, 운송 방법이나 수단, 이 기종 플랫폼도 모두 연결 가능 하고, 다양한 프로토콜의 사용이나 전환이 가능하도록 한다. ESB를 개념적으로 살펴보면 그림 3과 같다.

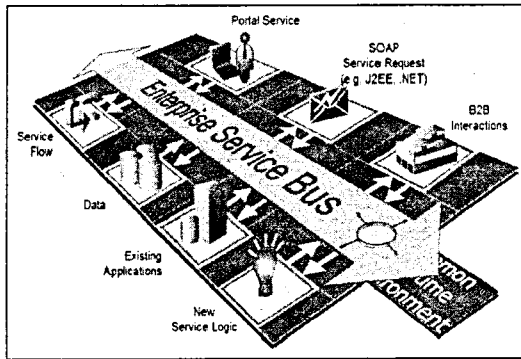


그림 3 ESB의 개념적 고찰 [6]

4-2 ESB에서 구현되는 것들

ESB는 엔터프라이즈 내부 또는 하위에서 전개되는 상황을 설명하고, 가용 리소스 및 흐름을 시각적으로 표현할 수 있어야 한다. 대부분의 기업에게 있어 통합을 더 쉽고 저렴하게 실현하는 것이 가장 큰 숙제이다. 어플리케이션 X가 어플리케이션 Y와 통신하게 할 수 있지만, X가 N개의 다른 어플리케이션과도 통신할 수 있어야 한다. 이러한 기능을 간단하게 구현하고 작동시키는 것이 ESB구현의 핵심이다. [5]

5. 사례 연구

본 사례는, 모 항공사의 어떤 업무에 SOA 개념을 적용시켜, 실 업무에 적용할

수 있는지 평가해 보기 위한 프로젝트이다.

5-1. 프로젝트의 배경과 범위

사례에서 선정된 SOA개념을 적용한 업무는 항공 램프 운영 시스템이다. 그림 4와 같이, 램프 운영 시스템은 비행기 운항에 필요한 비행기 기종 서비스, 고객 관련 정보, 비행 운항, 승무원 관리, 탑승이나 화물 등 비행에 관련된 모든 정보가 포함된다.

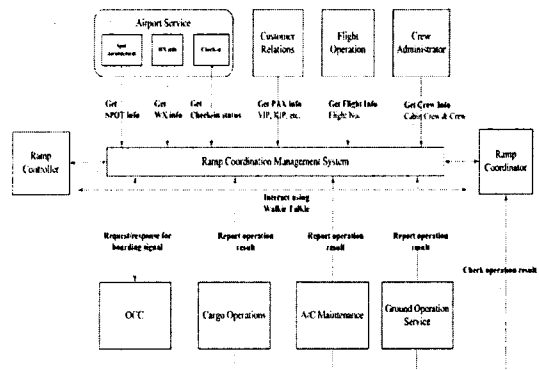


그림 4 항공 램프 관리 시스템. [6]

5-2. 프로젝트의 진행

프로젝트는 2005년 6월부터 8월까지 8주간 진행되었으며, 프로젝트의 진행 과정을 살펴보면 다음과 같다: ①고객의 요구 사항 분석, ②비즈니스 모델을 SOA가 요구하는 서비스 모델로 변경, ③각 서비스들을 비즈니스 목표와 연계, ④서비스 항목들 중 파일럿으로 구축할 항목 선정, ⑤참조 아키텍처 디자인, ⑥아키텍처 인프라에 들어갈 제품 선정, ⑦아키텍처 완성, ⑧ESB 및 웹 서비스 등 SOA관련 기본 기술들을 이용하여 실제 적용, ⑨완성된 결과물 전달 및 기술 전수.

위의 과정 중 주요한 내용만 정리하면, ①가장 중요한 고객의 요구 사항은 '램프 관리 시스템 관련 모든 정보를 실시간으로 공유하고, 기존 시스템에 대한 투자를 보호하라'는 것이었다. ②비즈니스 모델을

서비스로 분류하기 위하여, SOMA (Service Oriented Modeling Architecture)라는 체계화된 방법론을 사용하였다. ③비즈니스 목표와 하부 목표를 연계시키고, 하부 목표를 성능 요소와 연계시킨 후, 성능 요소를 서비스와 연계시키는 과정을 거쳤다. ⑥예를 들어, ESB 인프라를 구축하기 위하여 필요한 몇 가지 옵션들을 나열한 후, 각자의 장단점을 살펴 WAS 5.x + WBI Event Broker + MQ를 선정하였다.

프로젝트 진행을 살펴보면, 적용(설치 및 코딩) 자체 보다는, 이를 위하여 설계하는데 많은 노력과 경험이 필요했다.

5-3. 프로젝트 결과

프로젝트의 결과물로는 관련 서류와 운영 가능한 PoC (Prove of Concept)시스템이 전달되었다. 서류에는 아키텍처 설계 문서, 서비스 모델 문서, 구성 요소 설계 문서가 포함되었으며, PoC 시스템에는 선택된 파일럿 프로세서, 사용자 인터페이스 서비스, 응용 프로그램 서비스, ESB 등이 포함되었다.

프로젝트의 결과, 비즈니스의 목표에 잘 부합된 SOA 파일럿 시스템을 성공적으로 구축하였으며, 축적된 경험과 기술을 바탕으로 향후 다른 업무에 추가 적용할 수 있는 기반을 갖추었다. SOA를 구축하면, 외부 환경이나 시장의 변화에 민첩하게 대응할 수 있고, 작성된 서비스들은 재사용 가능하여 관리가 용이하고, 비용 효과적임을 알 수 있었다.

6. 결론

비즈니스 통합의 필요성 증가에 따라, 통합을 서비스 중심으로 생각하는 것은

목표 중심 (Object Oriented)으로 생각하는 것 보다는 분명히 앞서나간 개념이고 상당히 큰 변화임에 틀림없다. 사례 연구에서 살펴본 바와 같이, 기업의 인프라 환경이 발전하여도 서비스 기반 어플리케이션에 대한 신규 투자의 효과가 보존되고, 외부 환경의 변화에 민감하게 대처하고 비즈니스와 잘 연계된 SOA를 구축할 수 있었다.

개별 어플리케이션의 통합 개념을 뛰어넘는 서비스 지향 아키텍처가 향후 IT의 통합 표준이 될 수 있으리라 생각한다.

참고 문헌

- [1] Rirk Krafzig, Karl Banke, Dirk Slama, "Enterprise SOA; Service-Oriented Architecture Best Practice", PTR, 2004
- [2] Dave Chappell, "Enterprise Service Bus", Publisher: O'Reilly, June 2004
- [3] www.cbdiforum.com, "CBDI insight for Web Services & Software Component Practice", Tutorial: Service Orientation and Web Services
- [4] Simon Hayward, "Positions 2005: Service-Oriented Architecture Adds Flexibility to Business Processes", Gartner research, ID number:G00126409, 2005
- [5] Stephen Todd, Understanding Web Services on ESB, IBM ON Demand solution white paper, 2005. 03
- [6] Xin Sheng Mao, Lead Architect, Great China, manager of China SOA Design Center, IBM, Service Oriented Architecture Workshop
- [7] Mark Colan, "Service Oriented Architecture and Web Services", IBM, Technical Presentation Material