

# SOA에 관한 이해와 구현 핵심성공요인 도출

이상민\* · 박정선\*\*

\*한국정보통신진흥협회 · \*\*명지대학교 산업경영공학과

## A Study for Understanding and CSFs of SOA

Sang-Min Lee\* · Jeong-Sun Park\*\*

\*Korea Association for ICT Promotion

\*\*Dept of Industrial and Management Engineering, Myongji University

### Abstract

In this study, we reviewed SOA from the points of the concept, definition, technical components. Additionally, we derived seven critical factors which can lead SOA project to success: top level support, employees education, problem definition, service identification, appropriate development method, continuous development, start from small success.

**Keywords** : Service Oriented Architecture, Service, Loosely Coupled, CSF

## 1. 서론

### 1.1 연구배경 및 목적

현재 기업들은 독자적인 정보시스템(ERP, CRM, SCM 등)을 사용하고 있다. 이러한 정보시스템은 경쟁 업체들과의 경쟁에서 우위를 차지할 수 있게 해준다.

대기업부터 중소기업까지 회사의 발전과 자금사정으로 인해 인수합병(M&A)을 하는 경우가 많아졌다. 이러한 상황에서 독자적인 정보시스템은 하나의 회사로 만드는데 많은 어려움을 주고 있다.

이와 같은 이유로 새로운 정보시스템이 필요하게 되는 것인데 정보시스템을 구축하는 것은 비용적인 측면에서 큰 부담으로 다가올 수밖에 없다.

기업의 비즈니스 컨버전스 요구가 증대되고, 정보와 프로세스 통합의 문제가 발생하여 새로운 비즈니스 아키텍처 프레임워크가 필요한 실정이며, 변화에 따라 기업의 업무 흐름이 수시로 변화하고 고객, 공급자 등 모두를 고려하고 유연성과 민첩성을 만족시킬만한 통합 어플리케이션이 필요하게 되었다. 이러한 필요성에 의해 나

온 것이 서비스지향아키텍처(Service Oriented Architecture, 이하 SOA라 함)이다.

SOA는 비즈니스 모델의 애플리케이션을 서비스화하여 결합하는 것은 기업이 고민하고 있는 모든 비즈니스 문제를 해결해 줄 특효약으로 선전되기도 한다.

이러한 주장은 절대 실현될 수 없을 것이며, 비즈니스 문제는 특정 아키텍처나 특정 비즈니스 결정으로 해결될 수 있는 것은 아니다. SOA와 같은 기반 아키텍처를 구축할 때 얻을 수 있는 효과로는 다음과 같다.

- IT 활용의 잠재성을 실현하여 비즈니스 가속화 및 개선
- IT 비용과 자본 지출의 합리화
- IT 부문 외 담당자들에게 IT의 역할 및 운영방안 그리고 고유의 가치에 대한 인식 확대

현재까지 문헌연구는 SOA를 기술적으로 구현하는 방법인 웹 서비스, XML등에 대한 연구와 SOA 무엇인지, 구축 방법론 등에 대해 초점을 맞춘 연구들이 대부분이다. 따라서 본 연구에서는 SOA에 대해 연구하고 사례를 통해 핵심요소를 도출한다.

† 교신저자: 박정선, 경기도 용인시 처인구 남동 명지대학교 공학관 507호

M · P: 019-208-6453, E-mail: jspark@mju.ac.kr

2010년 10월 20일 접수; 2010년 12월 1일 수정본 접수; 2010년 12월 3일 게재확정

## 2. SOA 이해

### 2.1 SOA 개념 및 정의

기업 대부분은 생산성 향상과 고객서비스를 위해 정보시스템에 투자를 확대하면서 각각의 고유의 시스템을 도입하였다. 이기종 시스템 도입으로 시스템 간의 통신이 어려웠다. 이에 각 시스템이 같은 기능임에도 불구하고 자신의 시스템에 맞는 객체나 컴포넌트를 개발하였다. 이로 인해 재사용은 이루어지지 않았고 유지보수를 하기에도 어려움이 있다. 특히, 최근 들어 비즈니스 환경이 급격하게 변화하고 있는 시대에서는 이와 같은 이기종간의 시스템은 변화에 따라오지 못하고 있다. 이러한 문제점을 안고 있는 기존의 정보시스템으로 인해 비즈니스의 변화에 민첩하고 유연하게 대응할 수가 없었다. 이를 해결하기 위해 SOA가 발전한 것이다.

#### 2.1.1 서비스의 정의

IT 시장 전문가인 ZapThink에서는 서비스란 “애플리케이션 간 느슨한 결합(인터페이스와 구현의 분리)을 가지며 플랫폼에 독립적이며 재사용이 가능해야 한다.”라고 정의하고 있다.

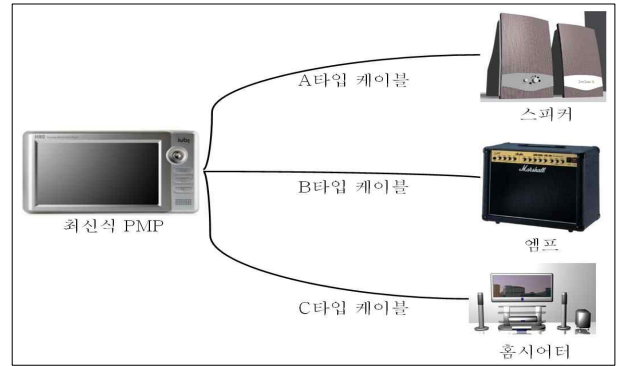
서비스는 비즈니스 프로세스를 수행하는 일련의 소프트웨어 컴포넌트를 말하고, 느슨한 결합(Loosely Coupled)을 가지며, 각각의 플랫폼에 대해 독립적이고, 재사용이 가능해야 한다. 느슨한 결합으로 서비스는 다른 서비스와 조합하여 또 다른 서비스를 만들어 낼 수 있어야 하고 재사용성을 높여야 한다. 또한, 서비스간의 통신을 메시지에 기반을 둔다.

즉, 간단히 말해 컴포넌트의 집합이며, 각 업무를 처리하는 절차를 하나로 묶어 서비스라 한다.

#### 2.1.2 SOA 개념

SOA는 변화에 적절히 대응할 수 있는 애플리케이션을 구축하는 개념으로, 애플리케이션을 구성하는 내부 업무 프로세스, 데이터와 기능을 각각 서비스라는 기본적인 기능 단위로 나누고, 이들 서비스를 연결하여 원하는 기능을 제공하는 것이다. 또한, SOA는 내부 애플리케이션과 서비스를 통합하는 것은 물론 협력사와 하도급 업체 등의 외부 어플리케이션까지 연계한다.

[그림 2.1]는 하나의 PMP를 가지고 소리를 크게 들을 수 있는 스피커, 앰프, 홈시어터에 연결하는 것을 말한다. 스피커, 앰프 그리고 홈시어터는 각각 다른 년도에 출시가 되었으며 서로 연결 되는 방식도 다르다.



[그림 2.1] SOA 개념의 예

연결 되는 선을 바꿔가면서 연결함으로써 인해 각각 독립적인 사운드 시스템에 유연성을 높인 것이다.

또한, 최신식 PMP에 오래된 앰프를 연결해 앰프의 재사용성을 높였다. SOA는 이와 같이 각각의 이기종간의 시스템을 느슨한 연결을 통하여 각각의 독립적인 시스템에 유연성 및 재사용성을 높인 것이다. 이와 같은 기능을 하는 것이 SOA이다.

#### 2.1.3 SOA 정의

SOA는 1996년 가트너 그룹에 의해 처음 소개된 개념으로 SOA라는 용어는 이미 10년 전부터 사용되어 왔다. 가트너는 SOA를 ‘기본적으로 Client/Server 방식의 소프트웨어 디자인 접근 방법이다. 전통적인 Client/Server와의 차이점은 서비스 컴포넌트 간의 느슨한 연결과 Client/Server를 분리하는 인터페이스라는 특징에서 기인한다.’ 라고 하였다.

SOA에 대한 정의는 표준화되어 정의되어 있지 않다.

그러나 각 그룹이 내린 정의는 공통적인 의미를 가지고 있다. SOA는 Client/Server와 비슷한 개념이지만 소프트웨어 컴포넌트들 간의 약 결합을 강조하고 개별적으로 독립된 인터페이스를 사용한다는 점에서 일반적인 Client/Server와는 다르다고 하고 있다.

즉, SOA란 서비스로 정의되는 애플리케이션 조각단위들을 느슨하게 연결해 하나의 완성된 애플리케이션을 만드는 아키텍처이다.

SOA를 기술적으로 살펴보면 SOA는 특정 비즈니스나 특정한 오퍼레이션을 수행하는 컴포넌트들로 구현된 서비스들로 소프트웨어를 정의한다.

## 2.2 SOA 요소

SOA는 컴퓨팅 환경 내에서 분산된 서비스를 유연하게 디자인, 개발, 배치, 관리하기 위한 소프트웨어 인프라 시스템으로 이기종 시스템간의 통합을 효과적으로

구현할 수 있는 방법론으로 기본적으로 서비스 사용자 (Service Consumer), 서비스 제공자(Service Provider), 서비스 중계자(Service Broker)로 구성된다.

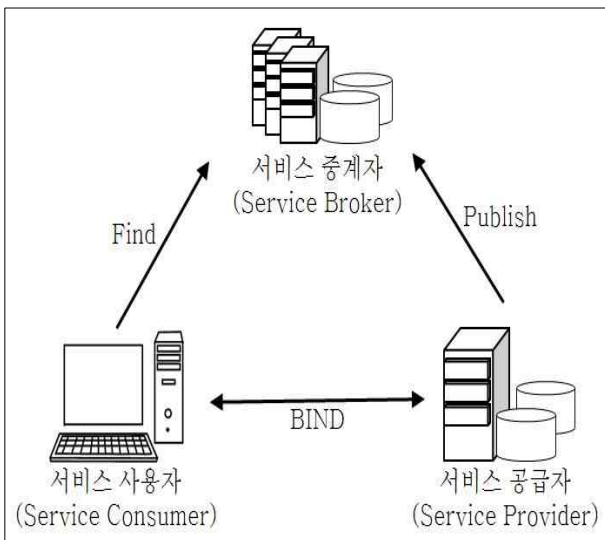
**2.2.1 SOA 기본구성**

SOA는 기본적으로 사용자, 공급자, 제공자로 구성되어 있다. 서비스 사용자는 필요에 따라 정보를 요청하게 되고, 공급되어지는 정보를 사용한다.

서비스 공급자는 서비스 사용자가 요청 시 그에 해당하는 결과를 제공한다. 경우에 따라 서비스 공급자는 또 다른 서비스 공급자의 서비스를 사용하는 서비스 사용자가 될 수 있다.

서비스 중계자는 서비스에 대한 기술 정보를 저장하고 서비스 사용자가 어떠한 서비스의 공급을 원할 때 공급자를 찾고 그 정보를 사용자에게 공급한다.

[그림 2.3]에서 나타난 바와 같이 서비스 사용자, 서비스 공급자, 서비스 중계자를 SOA의 구성요소라 한다. 만약 서비스 중계자가 존재하지 않는다면 그와 같은 구성 방식을 클라이언트/서버 방식이라고 할 것이다. 따라서 서비스 중계자는 SOA의 필수적인 구성요소로 볼 수 있다. 서비스 중계자는 서비스에 대한 기술 정보를 정리하여 서비스를 필요로 하는 사용자가 원하는 서비스를 체계적으로 찾을 수 있는 근간을 제공한다. 또한 사용자는 공급자의 위치나 어떠한 언어로 작성되어 있는 상관없이 요청을 하게 된다. 만약 사용자가 두 시스템에서 모두 정보를 받아야 할 때는 두 개의 정보시스템이 서로 다른 언어로 작성되어 있어서 중간에 중계자를 통해 사용자 인터페이스에 맞추어진 정보를 받으면 된다.



[그림 2.3] SOA의 기본 구성요소

**2.2.2 SOA 특징**

SOA 구성요소를 통해서 SOA는 서비스의 발견과 동적 바인딩이라는 개념을 지원한다. 어떤 서비스를 필요로 하는 사용자는 작업 수행 중에 필요로 하는 서비스를 찾고, 그것을 사용할 수 있는 것이다.

또한, SOA에서 사용하는 서비스는 컴포넌트에서와 마찬가지로 독립적인 모듈이다. 각각의 서비스는 독립적으로 개발, 유지, 관리되며, 서로의 작동 자체에 큰 영향을 미치지 않는다. 이런 점은 특정 서비스를 수정했을 때 발생하는 과급효과를 최소화 시켜 결합도를 낮추는 중요한 특징이다. 각 서비스는 자신을 호출할 수 있는 인터페이스를 제공하고 있으며, 호출이 이루어지는 프로토콜과 호출 메시지의 포맷만 이해 할 수 있으며, 서로 다른 플랫폼 위에서 개발, 운영되는 서비스끼리도 통신이 가능한 플랫폼 독립적인 상호 운용성을 제공한다.

SOA는 서비스 사용자와 공급자를 느슨하게 연결해 결합도를 낮출 수 있도록 하고, 서비스 사용자는 중계자에게서 서비스를 찾아 바인딩 하기 전까지는 자신이 사용할 서비스의 정확한 위치를 알지 못하는 위치 투명성의 특징을 갖는다. 위치 투명성은 서비스를 제공하는 물리적인 위치가 변경 되더라도 DNS의 유지보수를 통해, 서비스 제공과 사용에는 큰 영향을 주지 않게 할 수 있는 것이 SOA의 핵심적인 특징이다.

**2.2.3 SOA 컴포넌트 기술**

SOA와 웹서비스를 같은 기술로 보는 견해도 있지만, SOA와 웹서비스는 차이는 디자인에 있다. SOA는 서비스가 특정 기술이나 솔루션에 의존적으로 상호작용하는 방식을 정확하게 정의하지 않는다. 단지 서비스들이 서로 인식하고 상호작용하는 방법만 정의 한다. 프로세스가 수행되는 방법 및 전략을 정의하는 것과 실제로 수행되는 방법 및 전략에는 차이가 있다. 반면 웹서비스는 서비스들 간에 메시지가 상호작용하는 방식에 대한 특정 가이드라인을 제시하고 있다. SOA 모델은 SOAP 메시지를 통해 일반적으로 구현된다.

**2.3 SOA 접근방식**

**2.3.1 하향식 접근방법**

하향식 접근은 SOA 적용을 전사적 범위로 추진하는 것으로 비즈니스 프로세스를 서비스화 하는 많은 사전작업이 필요하지만, 비즈니스와 애플리케이션 사이에 일치된 방향성을 확보할 수 있고 변화 요구에 반영을 단순화 할 수 있으며, 재사용이 높다. 부서별 요구사항 정의단계에서는 상위계층부터 부서별 사원까지 필요한

요구사항을 도출한다. 이렇게 정의된 요구사항은 서비스 식별 단계에 들어간다. 식별단계1을 거치면서 요구사항의 중복을 제거하고 식별단계2를 거치면서 재사용을 고려한다. 이와 같은 방법을 거치면 서비스가 될 것들이 무엇인지 나오게 된다. 그럼 서비스 정의 단계에서 각 서비스를 정의 한 후에 서비스 흐름을 정리한다.

그 후에 컴포넌트를 분석한다. 이 단계에서는 서비스 흐름을 통해 컴포넌트 논리 모델링을 작성하는 것은 서비스와 컴포넌트의 인터페이스를 맞추기 위한 작업이다. 마지막으로 구현 단계에서는 인터페이스를 구현한다.

### 2.3.2 상향식 접근방법

상향식 접근은 시스템 통합에 중점을 둔 방법으로 기존의 애플리케이션의 효과적인 연결을 통하여 상위 개념의 서비스를 만드는 방식이다. 기존의 부서나 팀 별로 사용하던 애플리케이션을 서비스로 전환하고, 이들 서비스를 기반으로 또 다른 서비스를 구현해 나가는 방법이다. 구현이 쉽고 빠르기는 하지만 얻을 수 있는 이득에 제약이 있고 전체적으로 비용이 많이 소요 될 수 있으며 재사용이 낮을 수 있다. 상향식 접근 방법은 처음 단계로 컴포넌트 분석을 한다. 기존의 쓰던 시스템 컴포넌트 분석을 하고 컴포넌트를 통합한다. 여기서 통합은 재사용을 고려한다. 다음 단계인 서비스 정의에서는 사용되는 서비스들을 정의하고 서비스 흐름 단계에서는 서비스들의 흐름을 정의하고 구현단계에서는 인터페이스를 개발한다.

### 2.3.3 절충식 접근방법

절충식 접근은 상향식 접근을 통해 전사 애플리케이션 통합 부분을 SOA 기술요소 뿐만 아니라 SOA 원칙을 모두 적용하여 우선적으로 설계하고 구현한다. 하향식 접근을 통해 전사 비즈니스 모델과 비즈니스 서비스를 정의 및 설계하고 구현하여 잘 정의된 프로세스 부분에 상향식 접근으로 개발한 통합서비스를 우선 적용하여 상위 수준의 비즈니스 서비스를 지원하도록 한다. 이 접근은 하향식 접근 보다 빠른 효과를 기대할 수 있고 장기적인 목표에 보다 잘 부합하며 상향식 설계보다 더 많은 효과를 제공한다. 절충식 접근 방법의 처음 단계는 공급자들 입장에서 컴포넌트를 분석하고, 사용자들 입장에서는 요구사항을 도출한다. 다음 단계로 개발자 입장에서 시스템을 통합하고, 사용자들 입장에서는 서비스 식별 1,2를 거친다. 서비스 정의 단계로 개발자 입장에는 기존의 사용하던 컴포넌트를 가지고 재사용을 높이고, 사용자 입장은 새로운 서비스를 창출해 낸다. 서비스 흐름 단계에서는 각 입장에서 정의한 서비스를 흐름별로 정리 하고, 마지막으로 구현 단계에서는 인터페이스를 개발한다.

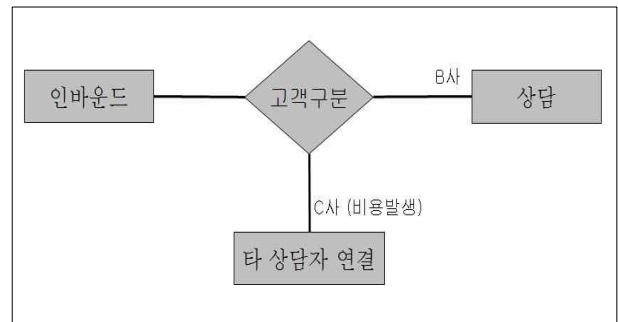
## 3. 서비스 기업의 SOA 도입

### 3.1 A보험사의 SOA 도입 사례연구

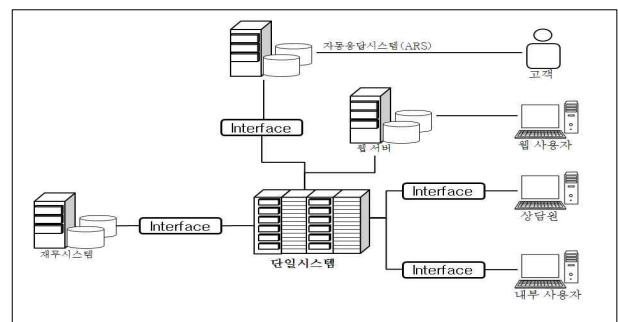
#### 3.1.1 SOA 도입 전 현황

A보험사는 미국의 보험회사중 하나이다. A회사는 B사와 C사의 합병으로 만들어진 회사이며, 기종이 다른 정보시스템으로 서로 다른 기업문화 속에서 절반쯤 독립적으로 운영되었고, 예산부족과 각종 규제로 모든 사람들의 손발이 묶여 있는 상태였다. 이러한 문제가 발생되게 된 것은 전혀 다른 방식으로 아키텍처를 설계했다. 그 결과 애플리케이션 통합과 신규 요건 실행엔진진이 안 되고, 실망스러운 결과만 계속되었다. 기존 고객이 전화를 걸어 상담을 요청하면 계약번호를 들고 B사의 고객인지 C사의 고객인지 구분을 하고 상담원의 담당이 아니면 전화를 돌리는 일이 발생하였다. 고객을 구분하고 전화를 돌리면서 고객의 만족도는 떨어졌고, 전화를 돌리는 추가적인 비용이 발생하였다. 고객의 상담 절차를 보면 [그림 3.1]과 같다.

이러한 상황에서는 두 개의 정보시스템을 하나처럼 인식 시키고, 새로운 인터페이스를 개발 할 수 있는 SOA를 도입하기로 결정하였다. [그림 3.2]와 [그림 3.3]은 기존의 B사와 C사의 레거시 시스템이다. [그림 3.4]는 합병 후에 두 개의 레거시 시스템을 연결시킨 통합 시스템이다.



[그림 3.1] 합병 후 전화 상담 연결 절차



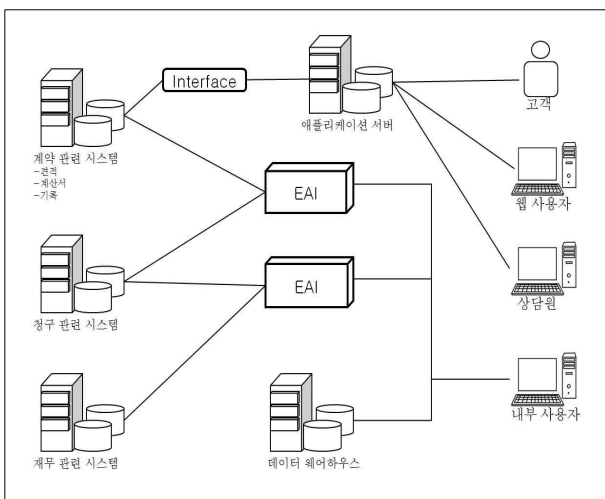
[그림 3.2] B사의 레거시 시스템

[그림 3.2]에서 보듯이 B사는 단일 메인프레임 플랫폼을 중심으로 IT인프라를 사용하고 있다. 단일 시스템에서 보험 업무인 계약, 견적, 청구, 인수 등을 다루었다. 보조 시스템인 재무시스템, 자동응답시스템, 웹서버는 인터페이스를 통해 연결되었다. 고객이 ARS를 통해 문의를 할 경우에는 자동응답시스템으로 연결되어 단일시스템하고 인터페이스로 연결된다. 웹 사용자는 웹서버를 통해 단일시스템에 연결되며 계약, 견적 등의 내용을 웹을 통해 알아볼 수 있다. 내부 사용자와 상담원은 인터페이스를 통해 단일시스템과 연결되었고 재무시스템은 보조시스템으로 따로 보관이 되고 있다.

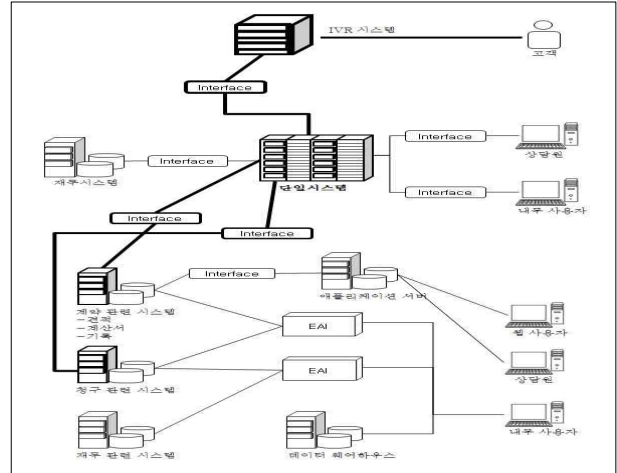
[그림 3.3]을 보듯이 C사의 레거시 시스템은 B사와 다르게 분산되어 있다. 이러한 분산시스템에서 재무 관련 시스템, 청구 관련 시스템, 계약 관련 시스템의 공급업체도 다양하여 계약, 청구, 재무 관리를 EAI 시스템으로 연결 하였다. 내부 사용자들은 클라이언트 시스템으로 EAI와 데이터 웨어하우스에 접근하였고, 상담원, 고객 그리고 웹 사용자들은 애플리케이션 서버를 통해 시스템에 접근하였다. 애플리케이션 서버는 계약 관련 데이터를 인터페이스를 통해 연결되었다. C사는 EAI 솔루션을 사용하여 계약, 청구, 재무 시스템 간의 상호 운영을 한다.

A사는 고객과 내부직원들의 요구사항을 서비스화 하고 시스템의 통합을 같이 시행하였다. 이러한 절충식 방법을 통해 효율의 극대화를 원하고 있다.

[그림 3.4]에서 보면 합병 이후에 통합시스템에서는 B사의 단일 메인프레임과 C사의 계약 관련 시스템, 청구 관련 시스템을 인터페이스를 통해 연결하였고, B사의 ARS시스템을 IVR(Interactive voice response)시스템으로 업그레이드를 하였으며, B사에 웹 서버 서비스를 중단하고, C사의 애플리케이션을 통한 웹 서비스를



[그림 3.3] C사의 레거시 시스템



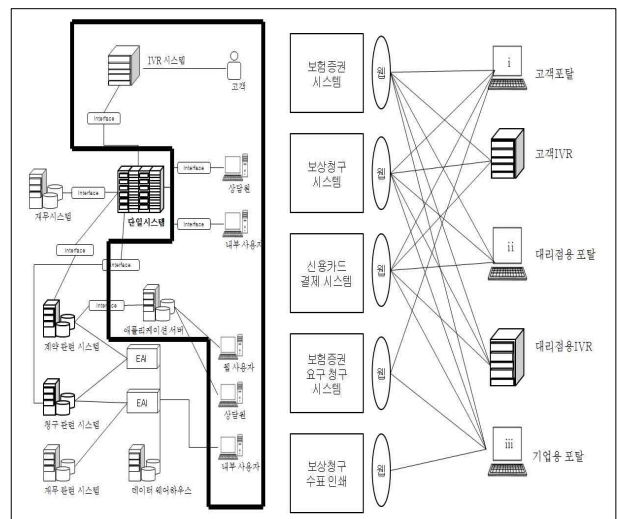
[그림 3.4] A사 합병 이후 통합 시스템

연결하였다. 이와 같은 시스템으로 두 개의 레거시 시스템을 통합했지만, 문서들 양식과 통신규약이 다르기 때문에 완벽히 하나의 시스템으로 만들지 못하였다.

재무부서에서도 회사 전체의 보고서를 적시에 받지 못하였다. B사와 C사는 비용과 수익을 관리하는 방식이 다르기 때문에 재무보고서를 일일이 대조한 뒤에는 CEO가 재무자료를 볼 수 있다.

### 3.1.2 SOA 도입 후 현황

A사의 시스템과 SOA의 상위 개념 관점에서 보면 [그림 3.5]과 같은 모델이 나온다. A사는 3가지의 포털과 2개의 IVR 시스템을 개발했다. 이러한 인터페이스와 시스템이 연결되기 위해 우선 제거되어야 할 시스템을 구분한다. 기존에 사용하던 IVR 시스템이나 애플리케이션 서버는 SOA로 도입으로 무용지물이 되므로 제거한다.



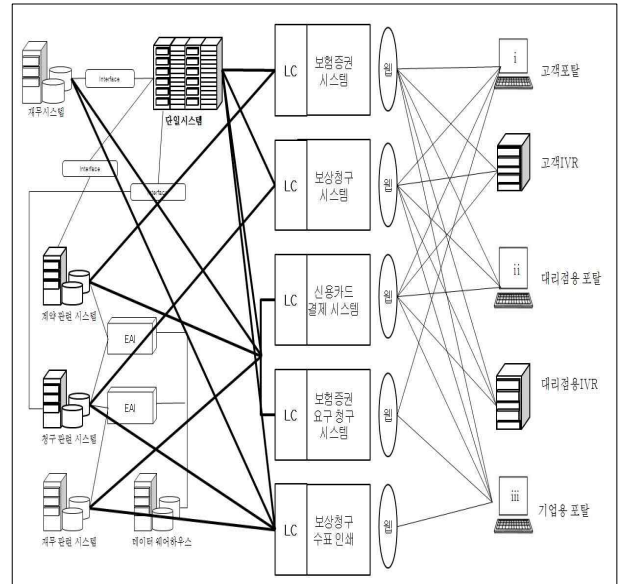
[그림 3.5] SOA 도입 후 제거될 시스템

A사는 [그림 3.6]과 같이 SOA 모델을 완성했다. 기존의 시스템들이 새로운 인터페이스랑 연결이 가능하도록 LC(Language Converter)의 개념을 도입하였다.

LC는 기존의 자체적인 언어로 작성된 것들을 XML로 바꿔주는 단계이다. LC를 도입함으로써 이기종간의 호환이 가능한 것이다. XML로 교체된 언어들은 웹을 통해 각각의 포털과 IVR에 필요한 정보가 전송된다.

보험증권 시스템을 사용하기 위해 B사의 단일시스템과 C사의 계약관련 시스템을 연결하였고, 보상청구 시스템은 단일시스템과 청구관련 시스템을 연결하였다.

신용카드 결제 시스템과 보험증권 요구 청구 시스템을 연결하기 위해 B사의 단일시스템, 재무시스템, C사의 계약관련 시스템, 재무관련 시스템을 연결하였다. 그리고 보상청구 수표 인쇄를 위해 B사의 단일 시스템, C사의 청구관련 시스템, 재무관련 시스템을 연결하였다.



[그림 3.6] A사의 SOA 모델

### 3.1.3 SOA를 통한 개선효과

A사는 SOA 도입으로 성공적인 프로젝트를 수행 중이며 다음과 같은 효과를 얻을 수 있었다.

- 비즈니스와 IT 간의 격차 해소
- 중복된 IT 비용지출 개선
- 시스템 간 중복된 기능 및 노력 제거
- 조직의 응집력 강화
- 시스템 교체에도 유연성 있게 대처 가능
- 신규 서비스 재작성을 위한 비용절감
- 보험 증권 계약, 보험 청구 계약 비용 절감

## 3.2 성공적인 SOA 핵심요소

A보험사의 사례를 통해 성공적인 SOA 도입을 위해 필요한 핵심요소 7가지를 제시하고자 한다. 핵심요소는 다음과 같이 정의 하였다.

### 1) SOA의 관리는 상위계층에서 해야 한다.

SOA는 리더십이 필수적인 프로젝트이다. 능력이 뛰어난 직원들도 SOA 프로젝트를 담당할 수 있지만, SOA가 성공하기 위해서는 상위계층에서 관리를 하는 것이 성공의 확률을 높일 수 있다. 웹서비스 작업이 역효과를 초래하는 것을 막고 SOA를 향한 건설적인 단계를 밟기 위해서이다. 상위 경영진의 적극적인 참여가 없으면 프로젝트는 최초의 계획과는 다른 방향으로 발전할 가능성이 크다. 특히, 사례연구를 보면 합병을 한 회사의 경우 문화적으로 서로 다른 두 회사가 만난 것이 때문에 직원들의 인식 자체가 다르다. 이러한 경우

에는 상위계층의 직원들이 길을 만들어야 하위 직원들이 쉽게 따라올 수 있고, 상부의 지원 부족 같은 잠재적인 문제들을 해결할 수 있다.

### 2) SOA는 직원교육이 필수적이다.

SOA 개발에 있어 중요한 요소는 직원교육이다. 새로운 아키텍처에 관련되어 있거나 직접적인 영향을 받을 모든 직원의 관심을 유도하는 것이 우선적으로 요구된다. SOA는 사람이 개입해야 하는 프로젝트라는 인식을 계속 유지하는 것이 필요하다. 업체로부터 구입할 수 있는 하나의 프로젝트나 소프트웨어 패키지가 아니다. SOA 개발을 위해 긍정적이고 지속적인 학습기반의 접근방식으로 직원을 계속해서 끌어들여야 변화로부터 일어나는 어쩔 수 없는 도전 과제를 극복할 수 있게 된다. 인력요소에서 성공하지 못한다면 SOA는 사람들의 손에 의해 실패로 끝날 가능성이 크다. 이러한 직원 교육은 일시적이 아닌 지속적으로 이루어져야 한다.

### 3) 현재의 문제점을 충실히 파악해야 한다.

SOA는 문제점을 찾고 그 문제점의 해결 방안을 찾는 기법이다. 기존의 정보시스템은 고정결함으로 프로세스에 맞추어 비즈니스를 개발하고 문제를 해결해 나가는 방식이지만, SOA는 문제점을 찾아내고 유연한 결함으로 비즈니스 프로세스를 문제 해결에 맞추는 방식이다. 현재의 문제점을 정확히 분석하고 서비스로 전환될 항목들을 명확히 구분해야 개발 단계에서 문제점이 생기지 않는다.

#### 4) 서비스 식별은 SOA의 성공을 좌우한다.

본 연구에서는 서비스 식별 단계를 두 단계로 나누었다. 첫 번째 단계는 서비스 중복을 제거하는 단계이고, 두 번째 단계는 재사용을 고려하는 단계이다. 하지만 이보다 더 먼저 선행되어야 할 것은 요구항목들을 서비스화 하는 것이다. 이러한 요구항목을 서비스로 만들고 그것에 대한 중복을 제거해야 한다. 중복된 것을 제거하지 않으면 개발 단계가 끝나고 인터페이스에서 중복이 나오기 때문에 혼란을 일으킬 수 있다. 또한 재사용을 고려하지 않으면 불필요한 서비스도 인터페이스에 들어가게 되는 경우가 있다. SOA는 서비스를 가지고 느슨한 연결을 하는 애플리케이션인데 서비스가 제대로 식별되지 않았다면 무엇보다 큰 문제가 발생하고 SOA 프로젝트를 시작도 못해보고 덮어야 하는 경우가 발생할 수 있다.

#### 5) SOA는 적절한 접근방식을 사용해야 한다.

서비스를 생성하는 본격적인 작업을 시작하기 전에 SOA 설계자는 서로 다른 두 가지의 접근방식을 어떻게 조화를 시킬 것인지 고민해야 한다. 하향식 접근방법은 일반적으로 서비스를 완전히 새롭게 구축하는 경우에 활용되는 반면, 상향식 접근방법은 기존 애플리케이션을 SOA 환경에 통합하는 경우에 많이 사용된다.

일반적으로 SOA 구현 시 두 가지 방법 중에 하나만을 선택하는 방법은 성공을 보장하지 못한다. 모든 종류의 서비스에 대해 전략을 혼합하여 사용하는 것이 바람직하다. A사의 사례도 이와 같이 절충식 접근방법을 통해 개발되었다. 새로운 서비스 구현 시 재활용할 수 있는 가능성을 충분히 고려하여 접근방법을 선택하는 것이 바람직하다고 볼 수 있다.

#### 6) SOA는 지속적인 개발이 필요하다.

SOA의 중요한 장점 중 하나로 웹 서비스 개발자의 작업 결과가 단일 프로젝트에 한정되지 않는다는 점을 들 수 있다. 호환성이 보장된 환경이라면, 다른 프로젝트 초기 개발 단계에서 기존에 개발된 서비스를 연결하여 활용하는 것이 가능하다. 하지만 그러기 위해서는 사전 준비가 필요하다. 소프트웨어를 제한적으로 활용할 계획만 가지고 있다 하더라도 미래의 요구사항에 대비할 필요가 있다. 이러한 미래의 요구사항에 대비하기 위해 지속적으로 개발자는 개발을 해야 하고 지속적인 개발이 이루어진다면 그 속에서도 새로운 비즈니스가 창출될 수 있다.

#### 7) SOA는 작은 성과를 통해 큰 성과를 얻어야 한다.

SOA를 고려하는 기업은 장기적인 SOA 전략을 구현

하고, 조금씩 목표를 달성해 나가는 접근법을 사용해야 한다. 웹 서비스를 통해 가능한 모든 문제를 한꺼번에 해결하려는 방법보다는, 장기적으로 중요성을 갖는 핵심 전략 서비스를 확인하고, 이러한 서비스를 SOA 환경에 우선적으로 구현하는 전략이 바람직하다. 우선적으로 기업의 마인드를 제고하고 SOA의 적용범위에 대한 이해를 도모하고서 작은 성과를 얻어가다 보면, SOA 접근에 대한 인식이 점차 확산되면서 큰 성과를 거두게 될 것이다.

## 4. 결론

SOA는 경쟁력 우위를 확보하고 신속한 비즈니스 환경을 창출하면서, 정보시스템에 대한 불만을 줄이거나 제거할 수 있는 프로세스, 조직 그리고 기술적 방법의 집합이다. 기존의 정보시스템은 컴포넌트들의 집합인데, SOA는 프로세스의 절차들을 서비스로 정의하여 이러한 서비스를 느슨한 연결을 하여 하나의 완성된 애플리케이션을 만드는 아키텍처이다.

본 연구에서는 SOA가 무엇인지와 성공적인 도입을 위한 핵심요소를 사례분석을 통해 알아보았다. SOA의 개념과 정의, 기술요소와 접근방법에 대해 알아보았고, 또한 국내외의 동향을 알아보았다. 그리고 두 보험회사가 합병을 통하여 이기종간의 시스템을 통합하고, 기존의 컴포넌트를 서비스로 식별하여 완전히 하나의 시스템으로 통합하는 SOA 사례를 분석하고 성공할 수 있는 핵심요소 7가지를 도출하였다. 7가지 핵심요소는 'SOA의 관리는 상위계층에서 해야 한다', 'SOA는 직원 교육이 필수적이다', '현재의 문제점을 충실히 파악해야 한다', '서비스 식별은 SOA의 성공을 좌우한다', 'SOA는 적절한 접근방식을 사용해야 한다', 'SOA는 지속적인 개발이 필요하다', 'SOA는 작은 성과를 통해 큰 성과를 얻어야 한다.' 이다.

향후 연구에서는 SOA의 핵심요소를 가지고 타 업종(통신, 제조, 공공부문)에 도입하여 기존의 핵심요소와 추가적으로 발생하는 핵심요소의 검증에 대한 연구와 SOA의 성과측정을 어떻게 할 것인가에 대한 연구가 필요할 것이다. 또한 본 연구를 통해 나온 결과물이 실제적인 프로젝트 진행을 위해 C언어, COBOL 등과 같은 언어에서 XML로 변환 할 수 있는 알고리즘이 필요하다.

## 5. 참고 문헌

- [1] 김동우(1999), 「XML/EDI 시스템 개발에 관한 연구」, 명지대학교 석사학위논문.
- [2] 김명호(2006), 「서비스지향 아키텍처를 통한 기업

비즈니스 혁신 전략에 관한 연구」, 연세대학교, 석사학위논문.

- [3] 김유경, 윤희란(2005), 「SOA를 위한 서비스지향 개발 프로세스」, 한국전자거래학회지 제12권 제2호.
- [4] 구중역, 이용봉(2007), 「SOA 기반 웹서비스의 Library 2.0 적용방안에 관한 연구」, 정보학회지 제38권 제3호.
- [5] 노갑수(2007), 「SOA기반의 IT 혁신 전략」, 한국지역정보개발원 KALI 지역정보화 vol.43.
- [6] 박재년, 김유경, 윤희란, 이혜선, 송영자, 이은정, 오현목(2005) 「웹서비스 기반의 서비스지향아키텍처(SOA) 구현방법 연구」, 한국전산원 연구개발 결과 보고서.
- [7] 심우혁(2009), 「사용자 관점의 SOA 서비스 모델링 기법」, 숭실대학교, 석사학위논문.
- [8] 안민정, 이홍철, 진혜진(2007), 「SOA 기반의 자재관리시스템 설계」, 한국경영과학회/대한산업공학회 춘계공동학술대회 pp.1148~1153.
- [9] 이상완, 조재훈(2007), 「서비스 지향아키텍처 구축 단계에 대한 연구」, 산업경영시스템학회지 제30권 제4호 pp.96~102.
- [10] 이현주, 최병주, 이정원(2008), 「서비스 지향 아키텍처를 위한 컴포넌트기반 시스템의 서비스 식별」, 정보과학회논문지, 소프트웨어 및 응용 제35권 제2호 pp.70~80.

## 저 자 소 개

### 이 상 민



명지대학교 산업공학과에서 합사, 석사를 취득하였으며, 현 한국정보통신진흥협회에서 방송통신 이용자보호 정책에 관한 연구를 진행중에 있다. 관심분야는 SOA, 방송통신 정책연구, 이용자보호관련 등이다.

주소: 서울시 서초구 서초동 1678-2 동아빌라트 2타운 3층

### 박 정 선



서울대학교에서 학사, 한국과학기술원에서 석사학위를 취득하였고, 미국 텍사스주립대학교 경영학박사를 취득하였으며, 한국전산원에서 선임연구원을 거쳐 현재는 명지대학교 산업경영공학과 교수로 재직 중이다.

주소: 경기도 용인시 처인구 남동 명지대학교 공학관 507호