

서비스 지향아키텍처 구축단계에 대한 연구

이상완* · 조재훈**†

*동아대학교 공과대학 산업경영공학과
**한국 아이비엠 소프트웨어그룹 기술영업

A study for deployment methodology of SOA(Service Oriented Architecture)

Sang-Wan Lee* · Jae-Hoon Cho**†

*Dept. Industrial Engineering, DongA University
**Technical Sales, Software Group, IBM Korea

The most fundamental method in IT innovation up to today is the SOA which has been accepted as the standard for the system integration and makes the business more flexible. Although it is important to classify the workflow of enterprise into the unit of services, there is no systematical research on this topic yet. This paper proposes a SOA deployment methodology, which composed of 6 steps. By utilizing this methodology, an enterprise can build up the SOA applied environment easily to achieve its business objectives and to make a linkage between services and business processes more effectively and efficiently, which is proved through Analytic Hierarchy Process. It is considered that Service Oriented Architecture can be the future standard for the IT integration beyond the individual application.

Keywords : SOA, System Integration, Service Science, e-Business Pattern

1. 서론

시장 변화에 대한 효과적이고 즉각적인 대응을 위하여 기업 내부뿐만 아니라 비즈니스 공급자와 수요자를 포함한 통합이 필요해지고 있다. 특히 통합된 각 시스템 간의 각 구성요소에 대해서 기술 독립적이며 융통성이 있는 시스템 간의 통합이 필요하다. 기업업무의 흐름은 수시로 변경되며, 복잡하고 변화해 가는 외부환경에 민감하게 대응할 수 있어야 한다. 그러기 위해서는 시스템의 통합뿐 아니라, 프로세서의 통합, 정보의 통합, 접근방법의 통합 등이 원활하고 수시로 변경 가능하여야 한다. 통합의 필수 요건 중 가장 중요한 요소는 공동 이용 가능성과 재 사용가능성이다. 통합의 대안으

로 EAI(Enterprise Application Integration) 솔루션의 도입은 내부적으로 약간 문제를 경감시키기는 하나 궁극적인 해결책은 아니다. 1996년 Gartner 그룹에 의해 소개된 서비스 지향 아키텍처는 최근에 많은 지지를 얻고 있다[2].

이제 서비스지향 아키텍처는 통합의 표준만이 아니라, 기업 업무 프로세서의 혁신 방안으로 받아들여지고 있다. 지금까지 SOA에 대한 체계화된 이론으로 그 구현 방법을 정의한 연구가 없어, 본 논문에서는 SOA 구현을 위한 단계를 연구 정리하였으며, 효율적인 방안으로서 e-비즈니스 패턴을 사용한 SOA구축 6단계 방안을 제시하고, AHP 의사결정 방안을 사용하여 유용성을 평가한다.

† 교신저자 jhcho@kr.ibm.com

2. SOA의 기본 개념

2.1 SOA의 정의 및 이점

IBM Mark Colan의 정의에 따르면 “서비스 지향 아키텍처란 서비스를 요청하거나 제공하여 응용 프로그램 기능이 가능하도록 할 수 있도록 하는 정책이나 관례 또는 골격이다”고 한다[3]. SOA는 서비스로 구성되어 있으며 개방된 표준을 사용하여 서비스간의 의사소통을 하는 아키텍처로 볼 수 있다. 현재 SOA를 구현하기 위한 기술로 가장 주목 받고 있는 것이 웹 서비스이다. 웹 서비스는 표준화 작업이 활발히 진행되고 있고 그 아키텍처가 SOA에 기반을 두고 있다는 면에서 SOA에 가장 근접한 것으로 이야기 되고 있다.

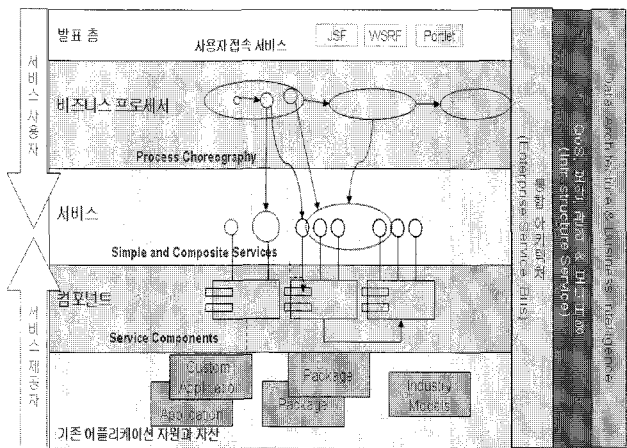
SOA를 적용할 경우 얻을 수 있는 기대 효과로는 크게 개발 및 유지비용의 감소와 IT의 비즈니스 환경 적용 속도 증가를 들 수 있다. SOA는 비즈니스를 여러 측면에서 변화시킬 수 있다. 비즈니스 구성방법, 어플리케이션을 구축하고 관리하는 방법, 조직이 이용하는 기술과 구입한 제품 등에 대한 모든 변화를 SOA를 통한 시스템 통합을 위해 효율적인 방식으로 이끌 수 있다. 서비스의 형태로 구성된 현재의 어플리케이션을 통해, 그 기능을 다른 비즈니스를 위해 재사용할 수도 있다. 따라서 어플리케이션의 생명주기를 늘리고, 새로운 기능을 개발하는 비용을 줄임에 따라 래저시 어플리케이션의 ROI(Return on Investment)를 증가시킬 수 있다.

2.2 컴포넌트와 서비스

서비스란 비즈니스의 논리적 단위이다. 이러한 논리적 구성요소 단위들 간에 흐름을 통제하거나, 자료 형식을 변환시키는 노력이 필요하다. 그래서 이들 서비스 컴포넌트들을 비즈니스 논리와 분리할 필요가 있다[4]. 기술적 관점에서 서비스가 이행되는 일련의 과정들을 생각해볼 때, 서비스들은 서비스 제공자와 요청자 간의 연속적인 네트워크 관계를 가진다. 이런 서비스가 이행되는 과정들의 네트워크 집합이 SOA라고 생각해 볼 수 있다[5].

서비스란 ‘특정 업무’를 통칭해 부르는 개념이라고 할 수 있다. ‘특정 기능’이 아니라는 점에 주목해야 한다. ‘화면을 어떻게 구성하고, 데이터를 어떤 식으로 처리하는가’하는 기술적 관점이나 사용자 인터페이스 개발이나 데이터베이스 관리 같은 솔루션 관점이 아니다. 서비스는 연말정산 서비스, 급여계산 서비스, 입출금 서비스 등 소프트웨어가 제공하는 궁극적인 기능에 초점을 둔 개념이다. 서비스는 좀 더 확장하여 인사관리 서비

스, 재무관리 서비스 등으로 확대할 수도 있다. 소프트웨어를 이 처럼 ‘서비스’의 관점에서 바라보는 것로부터 SOA는 출발한다. 객체지향 컴퓨팅이나 컴포넌트라는 개념은 SOA의 출발이라 할 수 있다. 늘 새로운 프로그램을 개발하는 비효율성을 탈피하고자, 재사용이 가능한 독립적인 프로그램 덩어리를 만들어, 이것들을 ‘레고 블럭’ 끼워 맞추듯 조립해 원하는 소프트웨어를 손쉽게 만들자는 것이 컴포넌트 개념이자 기술이다. SOA에서 말하는 서비스는 바로 ‘거대화한 컴포넌트’에 비유할 수 있다. <그림 1>은 컴포넌트와 서비스 및 비즈니스 프로세서와의 관계를 나타낸다. 서비스 제공자 입장에서는 어플리케이션 기능을 제공하기 위한, 표준화된 방법으로 개발된 단위 프로그램들의 집합이 컴포넌트이다. 서비스 사용자 입장에서는 이들 서비스들을 적절히 구성하여 그들의 업무 프로세서에 적용할 수 있다.



<그림 1> 컴포넌트와 서비스

3. SOA의 개발 단계

SOA 구현을 위하여 여러 가지 방안이 있을 수 있다. 물론 기업의 상황이나 여건에 따라 서로 상이한 구현 방안이 있을 수 있겠으나, 일반적으로 유효하고 정형화된 개발 단계가 있다면, 쉽고 빠르게 SOA를 구현할 수 있을 것이다. 상당한 구축기간과 비용이 필요한 SOA의 가장 큰 강점 중 하나는 IT자산의 유연한 흐름과 재사용을 통한 비즈니스 민첩성의 확보이다. 그러기 위해서는 보다 더 체계적이고 정리된 개발 단계의 적용이 필요하다. 기존에 제시된 개발 단계로는, 그랜드 센트럴의 David S. Linthicum이 제시한 일반적인 IT 환경에서의 SOA의 적용 12 단계[6]와 IBM이 제시한 SOA 구축 4단계[7] 등이 대표적이다. 그 외 Pierpalol은 이탈리아의 Ge-

nova 항구의 사례 연구로 3단계 접근법[8]을 제시하였고, Eric Knorrsmas가 SOA를 향한 10단계를 제시한바있다.

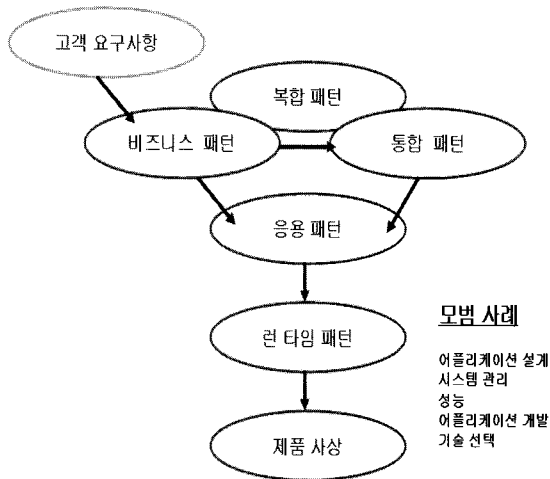
3.1 기존 SOA 개발 단계의 한계점

SOA 개발 단계에 대해서는 아직 학문적으로 완전히 증명되어 정착된 이론이 정립되지 못한 것이 현실이다. 앞서 제시한 방안들을 각 개발단계를 적용하여 SOA를 구현하기 위해서는 많은 한계점을 갖고 있다.

먼저 그랜드 센트랄이 제시한 일반적인 IT 환경에서의 SOA 개발 12단계는 SOA를 도입을 기획 하고자 할 때 예는 큰 도움이 될 것으로 보인다. 그러나 실질적으로 SOA를 구현하고자 할 때는 구체적인 구현 기술에 대한 언급이나 제시가 없어 현장에서 사용하고자 할 때는 많은 제약이 있다. 또 서비스나 프로세서를 어떻게 구분 할지에 대한 구체적인 제안을 하지 못하고 있다. IBM의 개발 4단계는 IT 기술 구현 측면에서는 잘 정리되어 제시하였으나, 비즈니스 측면에서의 서비스나 어플리케이션의 목표를 어떻게 명확화하고 비즈니스 프로세서를 어떻게 단위 컴포넌트나 서비스 등으로 구분할지에 대한 구체적인 제안이 없다. 또 구현 후 평가를 어떻게 하는 것이 좋을지 제시하지 못하고 있다.

3.2 e-비즈니스 패턴

e-비즈니스 패턴은 과거의 성공적인 경험과, 증명된 해결방법들의 모음이다. 이러한 구성 요소들의 재사용을 통하여 아키텍처들이 e-비즈니스 해결 방법을 쉽게 실행할 수 있게 한다. 패턴은 아래와 같은 구성요소들을 고려하여 <그림 2>와 같이 계층화된 구조로 되어 있다[9].



<그림 2> e-비즈니스 패턴 모형

e-비즈니스 패턴을 사용하는 목적은, 이미 검증되고 증명된 기술을 재사용함으로써, 시행착오를 줄이고 생산성을 높이기 위한 방법이라 하겠다[10].

- ① 비즈니스 패턴(Business Pattern) : 사용자, 비즈니스 및 데이터 사이의 상호 작용을 확인하는 패턴.
- ② 통합 패턴(Integration Pattern) : 단일 해결책이 단일 비즈니스 패턴에 기반 할 수 없을 때 복합 비즈니스 패턴으로 묶어서 나타내는 패턴
- ③ 복합패턴(Composite Pattern) : 비즈니스 패턴과 통합 패턴에서 일반적으로 발생하는 조합을 나타내는 패턴.
- ④ 응용패턴(Application Pattern) : 비즈니스 패턴과 통합 패턴 내에서 어플리케이션 컴포넌트와 데이터들이 어떻게 상호작용하는지를 설명하는 개념적 레이어아웃을 제공하는 패턴.
- ⑤ 런 타임 패턴(Runtime Pattern) : 응용 패턴을 지원하는 논리적인 미들웨어의 구조를 설명하는 패턴. 런 타임 패턴은 주요 미들웨어 노드와 역할들, 그리고 노드들 간의 인터페이스에 관해 묘사한다.
- ⑥ 제품 사상(Product Mapping) : 각각의 런 타임 패턴이 증명되고 테스트된 소프트웨어 수행임을 확인하는 것.
- ⑦ 모범 사례 : e-비즈니스 어플리케이션들의 설계, 개발, 배치, 관리에 관한 가이드라인.

3.3 SOA 구현 6단계 방안

SOA의 기술적인 구현을 위해서는 웹 서비스와 ESB 기술의 적용만으로도 충분하였으나, 컴포넌트와 표준 서비스 단위로의 구분이 용이하지 않기 때문에, 컴포넌트와 서비스 분리에 대한 용이함과 자원의 재사용이라는 SOA의 정의를 충분히 활용할 수 있도록 하는 새로운 방법이 필요하다. e-비즈니스 패턴은 검증된 아키텍처의 집합으로서, 복잡한 비즈니스의 문제를 단순화하고, 쉽게 접근 가능한 의미 있는 기능을 제공할 수 있기 때문에 복잡한 비즈니스 환경에서 e-비즈니스 패턴을 사용하면 SOA 구현이 용이하다. 개발 단계들의 가장 중요한 부분 중의 하나는 현재의 업무 프로세스를 어떻게 서비스 단위로 나눌 것이며, 이런 분류 과정에서 기존의 프로세스를 재사용할 수 있는 패턴을 도출해 내는 것은 상당히 유용한 방법이라고 할 수 있다. 패턴을 적용한 SOA의 구축 개발은 기존 업무 프로세스 재사용을 가능하게 한다. 또한 현재의 업무 프로세스를 서비스 단위로 분리하는 단계에서 기업의 새로운 서비스 생산에 따른 경제적 손실과 기술적 어려움을 줄일 수 있게 한다.

본 논문에서는, e-비즈니스 패턴을 사용하여, 다음과 같이 SOA 구축 6단계 방안을 제시한다[1].

3.3.1 제 1단계 : 비즈니스의 목표 이해와 성공요소 정의

기업에 있어 IT는 비즈니스 목표를 성취하기 위하여 존재한다. SOA를 적용하기 위한 첫 단계로 기업의 목표를 명확히 하고 성공을 위한 기업의 모든 목적들을 바르게 정의하는 것은 매우 중요하다. 이 단계는 기본적인 필요조건들을 모으는 단계로써 목표를 이해하고, 원활한 운영을 위하여 모든 문서와 서비스, 프로세스 등이 사람과 시스템 사이에서 조화를 이루도록 하며 설정된 기업의 목표와 문제해결을 위한 정확한 정의를 하여야 한다. 근본적으로 SOA 구축은 비즈니스와 IT간의 간극을 해소하기 위함이다. 그래서 비즈니스의 목표를 이해해야 하는 첫 번째 단계가 중요하다.

3.3.2 제 2단계 : e-비즈니스 패턴을 이용한 플리케이션과 프로세서 정의

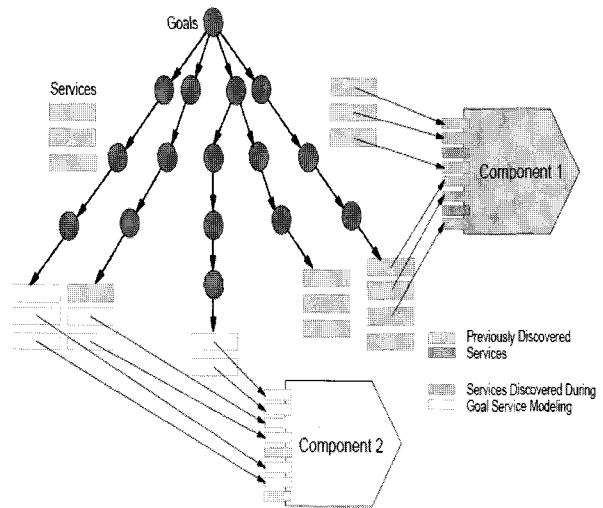
e-비즈니스 패턴은 솔루션 개발 방법론이 아니라 인터넷 기반 프로젝트에서 얻어낸 검증된 아키텍처의 모음이다. 그래서 e-비즈니스 패턴을 적용하면 어플리케이션과 프로세서를 쉽게 이해할 수 있다. 일반적으로 어플리케이션과 프로세서의 의미를 쉽게 이해하기 위해서는, 비즈니스 패턴과 통합패턴 및 어플리케이션 패턴의 적용만으로도 충분하다. 특정 비즈니스 프로세서에서 참조하는 어플리케이션의 언어나 문법을 이해하기 위해서는 어플리케이션의 의미론을 이해해야 한다. 이때 e-비즈니스 패턴은 어플리케이션 의미를 이해할 수 있게 한다. 서비스 인터페이스의 형태, 정보의 출처 이해, 프로세스의 이해, 다른 프로세스간의 공유 및 인터페이스를 이해하는데 e-비즈니스 패턴은 매우 유용하다.

어플리케이션과 프로세서를 이해하기 위하여 e-비즈니스 패턴을 적용할 때는 먼저, 비즈니스 패턴과 통합패턴을 적용하여 영역에 대한 분해를 완성한 후, 어플리케이션 패턴을 이용하여 상세 시스템을 분석한다. 먼저 영역에 대한 분해와 비즈니스패턴 및 통합패턴 적용하고, 세부 시스템 분석과 어플리케이션 패턴 적용을 한다. 문제 영역에 대한 분해를 마친 다음에는 세부 시스템의 각 기능들이 비즈니스 영역의 각 가치사슬과 어떻게 반응하는지 정의한다. 비즈니스 사용자 정의를 시스템 사용자정의로 정리하는 것이 세부시스템 분석이라고 할 수 있다. 세부시스템은 고객, 주문, 제품과 같은 비즈니스 구성요소와 메시징, 보안, 기록과 같은 기술적인 구성요소로 이루어져있다. 비즈니스 구성요소를 정의하기 위해서는 업무의 흐름을 분석해야 하고, 기술적 구성요소를 정의하기 위해서는 기능적, 비 기능적 요구사항을 알아야 한다. 간단히 설명하면 어떤 서비스들을 묶어서 특정한 기능을 제공하는 집합으로 제공할 수 있다. 이상과 같이 비즈니스 패턴 과 통합패턴 및 어플리케이션

패턴의 적용으로 어플리케이션과 프로세서를 이해하였다. 이를 통하여 우리는 서비스 인터페이스, 정보의 출처, 프로세스간의 공유 및 인터페이스를 쉽게 이해할 수 있다.

3.3.3 제 3단계 : 현재의 비즈니스 프로세서를 세부 단위 업무로 분할

e-비즈니스 패턴을 이용하여 이해한 프로세서를 표준 단위 업무로 분할한다. 이때 상세 분류한 업무의 단위를 서비스라고 하고, 표준 단위 업무를 컴포넌트라고 생각할 수 있다. 각 요소의 비즈니스 기능, 사용되는 데이터, 데이터의 제공자 및 사용자 또는 구성요소들은 e-비즈니스 패턴 분석의 결과를 이용한다. 현재의 비즈니스 프로세서를 세부 단위 업무로 분할하기 위해서는, 전단계인 어플리케이션과 프로세서의 정의 결과로 얻은 단위 업무들을 서비스로 할당하고 컴포넌트의 명세를 정의하고 이를 구조화해야 한다.



<그림 3> 서비스 할당

e-비즈니스 패턴을 이용하여 제 2단계에서 영역에 대한 분해와 세부시스템 분석을 하였다. 이 과정을 거쳐 우리는 프로세서를 단위 업무로 분류할 수 있고 이를 단위 서비스들로 정의한다. 모든 서비스들은 비즈니스의 목표 및 하부 목표와 일치 시켜야 한다. 몇 개의 서비스들을 묶어 그들이 주어진 기능을 수행할 수 있으면 이를 컴포넌트라고 한다. 이를 그림으로 나타내면 <그림 3>과 같다. 세부 시스템 분석을 통하여 우리는 세부 시스템에 대한 연결 관계, 시스템 사용자 정의, 비즈니스와 기술적인 구성요소 및 그들의 연관관계와 흐름을 정의하고 서비스 및 각 컴포넌트들을 인식하였다. 다음 단계는 컴포넌트에 대한 명세를 작성하는 것이다. 컴포

먼트 명세에는 규정들, 데이터의 속성, 연관된 다른 컴포넌트, 기타 고려해야 할 사항들을 정의해야 한다.

이상의 과정을 거쳐 우리는 서비스 및 컴포넌트를 정의하였고 각 컴포넌트에 대한 명세를 정리하였다.

3.3.4 제 4단계 : e-비즈니스 패턴으로 정의한 요소들을 서비스화

서비스화의 의미는 서비스의 사용자와 소비자를 정의하고, 웹 서비스 기술을 적용하여 새로 개발하거나 이미 사용 중인 업무의 단위 요소들을 웹 서비스로 변환시킨다. 웹 서비스를 사용하지 않고, 대신 메인프레임 트랜잭션 또는 객체 지향 시스템을 사용하는 SOA도 존재할 수도 있다. 그러나 SOA가 제공하고 또한 제공해야 하는 비즈니스 가치, 진정한 재사용 가능성 및 유연성을 갖추어 진정한 SOA로 간주되기 위해서는 웹 서비스 기술을 사용하는 것이 효과적이다. 그래서 본 4단계에서는 웹서비스 기술을 사용하여 서비스화 하는 경우에 대해서만 논의한다. 웹서비스는 표준기술인 XML, WSDL, SOAP, UDDI등의 표준을 준수하도록 코딩을 해야 하나, IBM등과 같은 벤드에서 제공하는 툴을 이용하여 우리가 작성한 프로그램을 표준기반에 맞도록 변환해주는 툴을 사용하는 것이 일반적이다. 이때 모든 서비스는 표준화된 XML 메시지를 이용하여 WSDL 인터페이스로 생성된다[12]. 예를 들어, IBM의 Rational Web Developer for WebSphere는 WS-I 호환 웹 서비스를 개발, 구축, 테스트, 배치 및 게시하는데 필요한 툴을 포함하고 있어, 처음부터 완전히 웹 서비스를 개발하거나 기존 어플리케이션을 WS-I 호환 가능한 웹 서비스화 할 수 있게 한다. 이를 사용하면 웹서비스를 설명하고 검사에 대비하는데 필요한 기존 자산으로부터 WSDL파일을 자동으로 생성하거나, WSDL파일 디자인 및 Java 코드를 생성하게 한다. WSDL 편집기 및 웹 서비스 탐색기를 사용해 WS-I와 호환 가능한 상호운영 가능 웹 서비스를 손쉽게 생성, 검증 및 발견하게 한다. SOAP의 이용한 입출력 메시지 인코딩이나 별도의 코딩 작업 없이 웹서비스 테스트클라이언트의 생성, UDDI, V2 레지스트리에 배치 등을 할 수 있게 한다[13].

3.3.5 제 5단계 : 서비스들을 ESB등과 연결

SOA 구현을 위해서는 기존 어플리케이션 인프라스트럭처에서 제공되는 서비스를 통합 관리하고, 느슨하게 연결 해 줄 수 있는 ‘서비스 인프라스트럭처’가 필수적이다. 이런 서비스 인프라스트럭처는 엔터프라이즈 전반에 걸쳐 서비스 제공자와 소비자의 네트워크를 더욱 관리하기 쉽고, 유연하게 만들어 주는 새로운 기능 모음을 제공해야 한다. SOA 구현의 핵심이라 할 수 있

는 ‘서비스 인프라스트럭처’는 이기종 플랫폼에 구축된 서비스를 별도 코딩 없이 구성만으로 쉽게 서비스를 연결해주고 표준 기반의 연결성을 제공할 수 있어야 한다. 이 ‘서비스 인프라스트럭처’를 ESB(Enterprise Service Bus)라고 할 수 있겠다. ESB가 설치되어 있지 않다면, 설치하여야 한다. ESB는 IBM등에서 제공하는 여러 제품의 기능 등을 검토하여 적절히 선정하면 된다. 이때 ESB는 서비스 요청자의 위치, 수송 방법, 위치나 시간 또는 형식 등에 상관없이 신뢰성 있는 메시지를 전송 및 중계하게 한다. 이 때 개발한 서비스들을 ESB에 직접 연결하기도 하고, 경우에 따라 IBM 등과 같은 업체에서 제공하는 어댑터 등을 사용하기도 한다.

3.3.6 제 6단계 : 테스트와 평가를 실시하여, SOA를 완성

타당한 테스트를 보증하기 위해, 테스트 계획은 특정 공간 내에서 시행되어야 한다. 테스트 계획의 세부적인 토론은 가이드의 범위 안에서 진행되며, 세부적으로 단계별로 수행되어 SOA의 테스트가 완료되어 진다. 테스트는 짧은 기간 내에 끝나야 한다.

4. 유용성 평가과정과 결과

본 논문에서 제시한 SOA구축 단계를 평가하기위해서는 SOA특성상 다중의 목표에 대한 분석과 정성적인 요인을 고려해야만 한다. 그래서 T.L.Satty가 제안한 비구조적인 문제 의사결정방법인 AHP(Analytic Hierarchy Process)를 적용하였다. 평가과정 및 결과는 아래와 같다[1].

4.1 대안의 선택

비교 대안은 가장 대표적인 SOA구축 단계로 받아들여지고 있는 그랜드 센트럴의 David S. Linthicum이 제시한 일반적인 IT 환경에서의 SOA의 적용 12단계와 IBM이 제시한 SOA 구축 4단계 및 본 논문에서 제시한 6단계 방안을 대안으로 선정하였다.

4.2 평가 기준 및 평가 항목의 선택

대안을 평가하기 위한 기준은 관련논문 및 연구보고서 등의 문헌검토를 통해 1차적으로 추출하였고, AHP의 초점이 집단 의사결정에 관한 것이므로 SOA 구축에 참여한 인원을 포함한 IBM의 SOA 전문가들을 대상으로 설문조사 및 토의를 거쳐 4개의 항목을 선정하였다. 선정된 4개 항목은 개발기간, 비용, 품질, 재사용성이다.

4.3 대안의 선택

각 평가기준에 대해서 항목 간에 어느 쪽이 얼마만큼 중요한가를 AHP에서 사용하는 일대일 비교치 목록을 기준으로, 쌍대 비교하여 중요성의 정도에 따라 중요도를 수치로 부여하고, 각 평가기준에 대한 대안들 간의 중요도를 수치로 부여하였다. 개별 구성원들의 평가 자료를 종합하는 방법으로는 기하평균법을 이용하였다[11].

그 결과는 다음과 같다. 각 평가기준에 대한 쌍대비교행렬과 중요도가 <표 1>과 같다.

<표 1> 각 평가기준에 대한 쌍대비교행렬

	개발 기간	비용	고객 만족	재사용성	기하 평균	중요도
개발 기간	1	2	1	6	1.861	0.383
비용	1/2	1	1/2	4	1.000	0.206
고객 만족	1	2	1	4	1.681	0.346
재사용성	1/6	1/4	1/4	1	0.319	0.066
				합계	4.862	1.00

<표 1>에 대해 일관성 검사를 수행하면 평가항목 간 쌍대비교행렬의 일관성 비율 CR이 0.017로 0.1보다 작아 일관성에는 문제가 없다고 할 수 있다.

다음은 각 평가기준에 대한 대안들 간의 중요도를 수치로 부여하고, 일관성검사를 수행하여, 일관성에 문제가 없음을 확인하였다. 각 대안들 간의 중요도는 아래 표와 같다.

<표 2> 개발기간에 대한 대안들 간의 중요도

개발 기간	12단계	6단계	4단계	기하 평균	중요도
12단계	1	1/5	1/2	0.464	0.122
6단계	5	1	3	2.466	0.648
4단계	2	1/3	1	0.230	0.230
			합계	3.804	1.00

<표 3> 비용에 대한 대안들 간의 중요도

비용	12단계	6단계	4단계	기하 평균	중요도
12단계	1	1/4	1/3	0.437	0.124
6단계	4	1	1/2	1.260	0.359
4단계	3	2	1	1.817	0.517
			합계	3.514	1.00

<표 4> 고객만족에 대한 대안들 간의 중요도

고객 만족	12단계	6단계	4단계	기하 평균	중요도
12단계	1	1/2	1	0.794	0.240
6단계	2	1	3	1.817	0.550
4단계	1	1/3	1	0.693	0.210
			합계	3.304	1.00

<표 5> 재사용성에 대한 대안들 간의 중요도

재사용성	12단계	6단계	4단계	기하 평균	가중치
12단계	1	1/7	1/2	0.415	0.094
6단계	7	1	5	3.271	0.740
4단계	2	1/5	1	0.737	0.167
			합계	4.423	1.00

각각에 대하여 일관성 검사를 수행하면 쌍대비교행렬의 일관성 비율 CR이 각각 0.003, 0.093, 0.016, 0.012로 0.1보다 작으므로 일관성에는 문제가 없다고 할 수 있다.

4.4 대안의 종합 평가치 및 결과

각 대안의 종합 평가치는 평가항목의 가중치 <표 1>과 각 요소의 가중치를 곱하여 합산한다. 그 결과가 아래 <표 6>에 나타나 있다.

<표 6> 각 대안의 평가결과

	개발기간 (0.383)	비용 (0.206)	고객만족 (0.346)	재사용성 (0.066)	종합 평가치
12단계	0.047 =0.383* 0.122	0.026 =0.206* 0.124	0.083 =0.346* 0.240	0.006 =0.066* 0.094	0.162
6단계	0.248 =0.383* 0.648	0.074 =0.206* 0.359	0.190 =0.346* 0.550	0.049 =0.066* 0.740	0.561
4단계	0.088 =0.383* 0.230	0.107 =0.206* 0.517	0.073 =0.346* 0.210	0.011 =0.066* 0.167	0.279

종합평가 가중치를 살펴보면 6단계가 0.561로 가장 높으므로 다른 대안에 비해 6단계 방안을 선택하는 것이 최적의 대안이 된다.

5. 결론

서비스 지향에 대한 개념이 처음 소개된 것은 약 10년 전이었지만 그 동안 기반 기술의 발전 부족과 시장

환경의 미성숙 등으로 널리 인식되지 못하였다. 그 동안 웹 서비스와 같은 표준기술의 발전, 외부 환경의 급격한 변화에 대응하기 위한 기업의 혁신 필요성 때문에 최근에는 그 중요성이 부각되었다. SOA는 단순한 기술이 아니며 프로세스와 사람을 필요로 하는 IT 기술과 비즈니스의 결합이다. SOA는 그 자체만으로는 독립된 기술이라고 하기는 어려우며 이를 지원하는 많은 기술들과 표준들과의 연계에 의해서만 이루어 질 수 있다. Gartner Research 연구 보고서나 최근 시장의 동향으로 미루어 볼 때, 많은 기업들이 적극적으로 SOA 도입을 추진할 것으로 보이며, 정보시스템의 측면에서뿐 아니라 비즈니스 측면에서도 SOA는 대단히 중요한 화두가 될 것이다. 그러므로 SOA는 향후 IT 업계의 비즈니스 통합을 위한 표준으로 자리 잡을 것임이 분명해 보인다.

많은 기업들이 SOA 개념을 적용한 비즈니스 프로세스를 실제 업무에 적용하고 있다. 그래서 기존 투자를 보호하면서 SOA를 구축할 수 있음을 확인하였고, 재사용 가능한 서비스의 활용으로 기업은 외부 환경의 변화에 민감하게 대처하고 향후 쉽게 확장 가능하며, 비즈니스 목표에 잘 연계된 정보 시스템을 구축할 수 있었다. e-비즈니스 패턴을 이용한 SOA 구축 6단계 방안을 적용하면, 보다 더 쉽게 비즈니스의 상황을 이해할 수 있으며, 효율적으로 SOA를 구축할 수 있음을 AHP분석을 통하여 확인 하였다.

본 논문은 기존의 업무를 서비스 단위로 분류하는데 있어서 그 체계적인 연구가 없었으나 패턴을 이용하여 서비스 단위로 분류하고 이를 적용하고자 한 시도에 중요한 의미가 있다. 본 논문에서 제시한 e-비즈니스 패턴을 이용한 SOA 구축 6단계가 향후 기업들의 SOA 구축에 도움이 되기를 기대한다. 비즈니스 통합의 필요성 증가에 따라, 통합을 서비스 중심으로 생각하는 것은 분명히 앞서나간 개념이고 상당히 큰 변화임에 틀림없다. 개별 어플리케이션의 통합 개념을 뛰어넘는 서비스 지향 아키텍처가 향후 IT의 통합 표준이 될 수 있으리라 생각한다.

참고문헌

- [1] 조재훈, "e-비즈니스 패턴을 이용한 효율적 SOA 구축방안 연구", 동아대학교, 박사학위논문, 2006.
- [2] Roy, W., Schulte, Yefim, and Natis, V.; "Service Oriented Architecture," Gartner Group, SSA Research Note SPA-401-068, 1996.
- [3] Daniel Sabbah, "2005 IBM Executive SOA Summit," IBM presentation material, 2005.
- [4] Bieberstein, et al.; "Service-Oriented Architecture Compass : Business Value, Planning and Enterprise Roadmap," Person Education, 2006.
- [5] <http://www.cbdiforum.com>; "CBDI insight for Web Services & Software Component Practice," Tutorial: Service Orientation and Web Services.
- [6] David, S. Linthicum; "12 Steps to Implementing a Service Oriented Architecture", Grand Central, 2005.
- [7] Bieberstein, et al.; "Service-Oriented Architecture Compass : Business Value, Planning and Enterprise Roadmap," Person Education, 2006.
- [8] Pierpaolo Baglietto, Massimo Maresca, Andrea Parody, and Nicola Zingirian; "Stepwise deployment methodology of a service oriented architecture for business communities," *Information and Software Technology*, Italy, 47 : 427-436.
- [9] IBM patterns for e-business; <http://www-128.ibm.com/developerworks/patterns>.
- [10] Martin, K.; "Patterns: Implementing an SOA using an Enterprise Service Bus in WebSphere Application Server V6," IBM Redbook, 2005.
- [11] Hannan, E. L.; "An Eigenvalue Method for Evaluating Contestants," *Computer and Operations Research*, 10(1) : 41-46, 1983.
- [12] O'Neill, M.; "Web Services Security," McGraw Hill, 2003.
- [13] Thomas, E.; "Service-Oriented Architecture : A field guide to integrating XML and Web Services," Prentice Hall, 2004.