

# 새로운 비즈니스 언어, SOA

## : Prologue



등장인물 소개

금융사업본부  
영업대표  
모하니

마케팅  
안양혜

제조사업본부  
영업대표  
김대강

IBM  
SOA맨







아키텍트의 분석 결과에 따르면 양사 시스템을 연결하려면 최소한 4가지 프로젝트를 동시에 진행해야 한다고 그러더군요.

- 1) MDM을 통한 고객 DB 통합
- 2) EAI를 이용한 양사 미들웨어 통합
- 3) 전자 포탈 도입을 통한 사용자 UI 통합
- 4) 그리고 이 모든 것에 대한 로드맵을 도출할 수 있는 컨설팅

\* MDM (Master Data Management) : 마스터 데이터 관리  
 \* EAI (Enterprise Application Integration) : 전자 어플리케이션 통합  
 \* UI (User Interface) : 사용자 인터페이스





# Chapter 1. SOA의 배경

## SOA

여기가 SOA센터..

실례하겠습니다.

안녕하세요.  
제가 바로 SOA맨입니다.  
기다리고 있었습니다.

어서오세요.

안망해 씨를 통해  
말씀 전해 들었습니다. 요즘  
일 때문에 고민이 많으시다구요.

아 그러셨군요. 그것 때문에  
SOA에 대해 알고 싶어서  
이렇게 찾아왔습니다.



음.. 잘 이해가 안 되는데요?

그럼 이렇게 생각해 보죠.  
기업이 비즈니스를 하기 위해서는  
반드시 IT 시스템이 필요하지요?



먼저 기업이 IT 시스템을  
개발하는 과정을 한 번 생각해 봅시다.

예를 들어 A은행이 방카슈랑스를  
하기 위해서 새롭게 보험 상품을 판다고  
생각해 봅시다. 어떻게 하지요?



보통 현업에서 IT 부서에 필요한  
요건을 이야기하면 이를 바탕으로 IT  
부서에서 개발하게 되는 것 아닌가요?

맞습니다. 현업이 새로운 보험 상품을  
개발하는데 필요한 요건을 제공하면 IT 담당자는  
이를 바탕으로 시스템을 설계하게 되지요.

이 과정에서 현업의 확인을 받는  
과정을 거치기도 하구요.



IT 담당자가 설계를 바탕으로  
어플리케이션 프로그램을 완성한 후, 개발된  
어플리케이션을 현업에서 테스트하고 문제가  
없으면 운영을 하는 단계를 거치게 됩니다.

그런데 보통 이 과정에서 현업의 요구 사항이  
제대로 반영되는 경우는 거의 없고 대부분  
몇 번의 수정을 거쳐서 완성됩니다. 이러한  
과정은 시간을 많이 잡아 먹죠.

짧게는 몇 개월에서 길게는  
몇 년까지 걸리니까요.



만약 경쟁하는 은행들이  
방카슈랑스 준비를 동시에  
시작했다고 칩시다.

A은행이 방카슈랑스 시스템을 1개월만에  
완성하여 고객에게 서비스를 시작하는데  
반해서, B은행은 6개월이 걸려  
완성했다고 하면,

이러한 경영 환경 변화에서 어느  
은행이 더 많은 혜택을 받을까요?



당연히 A은행이지요.  
고객들은 항상 1등만을  
기억하니까요.

그렇죠, 그것이 기업의  
경쟁력이라 할 수 있습니다.



A은행은 IT 시스템 때문에  
경쟁 상황에서 유리한 위치를  
차지하게 되었고

B은행의 경우는 좀 심하게  
이야기하자면 IT 시스템이 발목을  
잡았다고 할 수 있겠죠.

그렇다면 SOA는 급변하는  
경쟁 상황에 기업이 적절히  
대응하는데 필요한 유연한 IT  
시스템을 갖추는데 도움이  
되는 것이군요.

맞아요, 요즘처럼 경영 환경이 급변하고  
그 방향을 예측하기 어려울 때는 IT 시스템을  
미리 계획하여 개발하는 것이 아니고,  
어떤 변화도 수용할 수 있도록 IT 시스템의  
체질을 바꾸자는 것이지요

그렇군요, SOA가 왜  
나왔는지 이제 알 것 같네요.



## Chapter 2.

# SOA의 개념



그런데 사용자 입장에서는 약간 다른 면을 발견할 수 있습니다.

다른 면이라면?

네, 예를 들어 은행에서 차세대 시스템을 구축한다면 일반적으로 banking 패키지를 도입하게 되는데요, 기존 시스템과 차세대 시스템을 비교하면 동일한 기능을 중복 개발한 모습이 되는 것입니다.

예를 들어 예금, 대출, 조회 등과 같은 기본적인 내용들은 플랫폼이 바뀌고 개발 언어가 바뀐다고 해서 기능의 내용이 바뀌는 것은 아니니까요.

그렇지만 기존 시스템으로는 새로운 경영 환경을 제대로 지원할 수 없어서 바꾸는 것이잖아요?

맞습니다. 기존 시스템에서 살릴 수 있는 것을 살리고, 없거나 필요한 것들만 개발한다면 빠르게 차세대 시스템 구축이 가능하겠죠. 사용자 입장에서 어떤 플랫폼과 언어를 사용했는지는 중요한 것이 아니니까요.

그럴 수도 있겠네요.

아무튼 SOA는 이러한 배경을 가지고 있습니다. 그러면 SOA에서 말하는 서비스는 무엇일까요?

그게 제법 어려워요. 서비스라는 말을 일상적으로 많이 사용하잖아요. 그런데 각각 그 의미가 달라서 혼란스럽기도 하네요.

Service  
Oriented  
Arch.

SOA에서 서비스는 IT와는 전혀 상관 없는 개념입니다. 우리가 은행에 가서 받는 대출도 서비스구요. 우리가 제주도 갈 때 비행기 타는 것도 서비스입니다

잘 이해가 안 가는데요?

이렇게 생각해 봅시다. 우리가 해외 여행을 한다고 할 때 일반적으로 패키지 여행을 많이 하지요?

패키지 여행을 가는 이유가 뭔가요?

우선 가격이 싸고, 그 나라 사정을 잘 모르니 겁도 나고, 어디를 가야 할지도 모르고

그렇죠. 하지만 패키지 여행을 다녀와서 사람들이 만족합니까?

만족하는 경우도 있지만 만족 못하는 경우가 더 많아요.

가이드가 원하지 않는 상점에 데려가기도 하고, 일정이 꼭 짜여져 있어서 피곤할 경우도 있구요.

만약 모대표가 태국에 관광을 간다고 생각해 봅시다. 패키지를 선택할 수도 있지만 본인이 직접 항공권도 예약하고

호텔도 인터넷에서 예약하고, 렌터카도 예약한 후에 자유롭게 가고 싶은 곳을 갈 수도 있겠죠? 약간 비싸고 귀찮겠지만 말이죠.

아. 그렇군요.

항공권 예매

렌터카 예약

HOTEL

호텔 예약

서비스라는 것이 바로 항공권 예약, 호텔 예약, 식당 예약, 렌터카 예약 같은 업무 단위를 말합니다.

다시 한 번 말하면 SOA에서의 서비스란  
구분 되어질 수 있는 업무 단위를  
말하는 거죠.



여기에는 IT의 기술적인  
내용이 하나도 없지요?

그래서 SOA는 현업에서  
시작한다 라는 말이  
나오는군요.



네. 기업에서 SOA를 시작하려고  
할 때 제일 먼저 해야 하는 것이  
기업 내의 업무를 서비스라는 단위로  
나누는 작업입니다.

그렇군요. 이제 서비스라는  
말을 이해하겠어요.

그러면 서비스 지향이라는  
말은 무엇일까요?

**Service Oriented**  
**Service Oriented**

글쎄요. 그건 아직 잘...

축하합니다. 서비스라는 말을  
이해했다면 SOA를 절반 정도는  
이해했다고도 볼 수 있어요.

서비스 지향이라는  
말은 정의된 서비스를  
서로 연결하여 특정한  
기능을 하도록 하는  
것입니다.

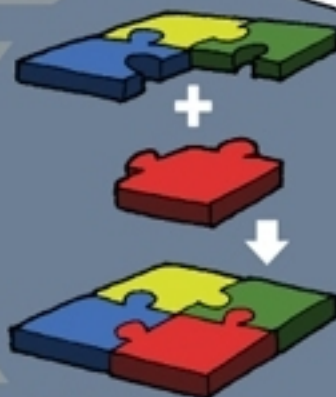
아까 예를 든 태국 여행에서  
항공권, 호텔, 렌터카 예약, 비행기  
탑승, 렌터카 수령, 호텔 체크인  
등과 같은 각각의 서비스를  
결합하면 태국 여행이라는 목적을  
달성할 수 있는 것과 같습니다.

태국 여행과 일본 여행에서  
항공권 예약, 호텔 예약 등의  
기능은 공통으로 사용되니까  
재활용을 하면 되고,

태국 여행 때 사용했던 전통  
마사지는 일본 여행에서는 후지산  
등반이나 온천 등의 대체 가능한  
서비스로 변경하면 되죠.



그렇군요. 제가 다음에 일본  
여행을 간다고 하면 그 중에서 항공권  
예약이나 렌터카 예약 등과 같은  
서비스를 다시 사용하면 되겠군요.



이것이 SOA가 보다 빨리  
비즈니스 환경변화에 대응하도록 해주는  
비밀이지요. 여기까지도 여전히 IT의  
기술적인 내용이 끼어들지 않아요.

알겠습니다만 이러한 것들은 모두 IT로 구현이 되어야 실제로 사용할 수 있는게 아닌가요?

맞습니다. 이 모든게 IT로 구현이 될 때 비로소 사용 가능하죠. 그러기 위해서 알아야 할 개념이 **컴포지트 어플리케이션** 입니다.

컴포지트 어플리케이션이요?

## Composite Application



보통 어플리케이션을 개발한다 라는 용어를 많이 쓰잖아요. SOA에서는 대신 **어플리케이션을 조립한다** 라는 용어를 많이 사용합니다.

그게 무슨 말인가요?

SOA에서는 서비스들을 미리 만들어 놓고 이들을 서로 연결하여 특정 기능을 구현할 수 있다고 설명했죠.

이들 기능이 실제로 작동하기 위해서는 서비스와 서로 연결되는 **컴포넌트들이** 필요합니다. 즉 EJB라든가 .NET 컴포넌트라든가 하는 것들이죠.



아까 예를 들었던 태국 여행에서 항공권 예약이라는 서비스를 생각해 봅시다. 항공권 예약은 하나의 기능이지만 이것이 실제로 작동하기 위해서는



여행사의 항공권 예약 어플리케이션을 이용해야만 합니다. 이 어플리케이션을 A항공사는 UNIX에서 C를 이용해 구축하였고, B항공사는 Java 기반의 EJB 어플리케이션으로 만들었다고 가정해 볼게요.

즉 항공권 예약이라는 업무 기능(서비스)은 실제로 까보면 B항공사의 Java 기반 EJB 어플리케이션을 사용하여 실제 일을 하게 됩니다.



그래서 정의된 업무 기능(서비스)과  
눈에는 안 보이지만 데이터베이스를 검색하고  
처리하고 다시 데이터베이스를

수정하는 작업을 하는 실제  
어플리케이션을 연결하고,

연결된 서비스들의 순서를 정의하여  
특정한 기능을 구현하도록 하는 것이  
컴포지트 어플리케이션입니다.

Service

Application

Composite  
Application

아. 여기서 IT 기술적인  
내용이 나오는군요.

사용자들은 사실 밑단의 어플리케이션  
컴포넌트가 무엇으로 되어 있는지 상관 없이  
정의된 서비스만을 알게 되면 원하는 기능을 쉽게  
개발할 수 있게 되는 것이지요.

사용자가 매우 쉽게  
특정 기능을 구현해  
낼 수 있겠군요.

네, 이를 위해서 업무를 기업 상황에 맞게 서비스로  
잘게 쪼개 정의해 놓는 작업과 사전 개발된  
어플리케이션 컴포넌트를 연결시켜 놓는  
작업이 선행되어야 합니다.

만약 필요로 하는 기능에 맞는  
어플리케이션 컴포넌트가 없다면  
미리 만들어 놓아야 현업 담당자가  
필요할 때 사용할 수 있겠지요.

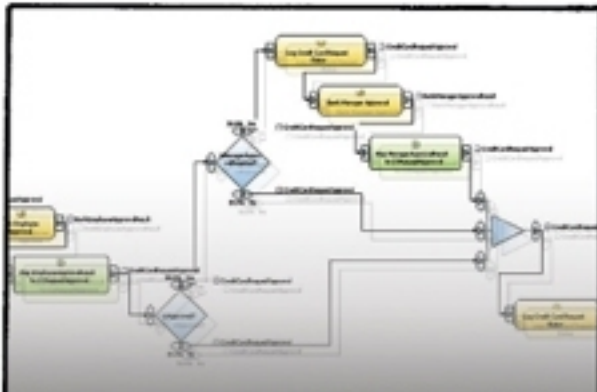
이제야 SOA를 좀  
이해할 수 있네요.

네. 그래서 SOA, 즉 서비스 지향  
아키텍처는 이렇게 서비스 지향성을  
가지고 IT 시스템을 구축하기 위한  
철학이자 사상이라고 말할 수 있습니다.



물론 어렵겠죠.  
서비스를 연결하기 위해서  
서로 어떤 식으로 연결해야  
하는지를 확인해서 가능한  
코드를 새로 만들어야  
하겠지요.

맞아요. 그렇게 된다면  
SOA가 추구하는 것, 즉  
변화에 빨리 대응하도록  
IT 시스템을 바꿀 수가  
없게 되겠지요.



그래서 개방 표준이 필요합니다. 어플리케이션  
컴포넌트 개발 뿐만 아니라 정의된 서비스의  
순서를 정의하는 것도 개방 표준을 따라야 하지요.

그렇게 된다면 서비스와  
서비스 연결 방법이 통일되어 쉽게  
연결할 수 있게 되는 것입니다.

그렇군요.  
그럼 어떤 개방 표준  
들이 SOA에서 사용될  
수 있습니까?

여러 가지 표준들을 다 사용할 수  
있지만 특별히 기억해야 할 것은 정의된  
서비스들을 순서를 정해서 연결할 때  
사용되는 BPEL이라는 표준입니다.

## Business Process Excution Language

그리고 XML을 기반으로 한 WSDL,  
SOAP 같은 웹서비스 표준도 SOA에서는  
중요하게 사용됩니다.

그렇군요. 개방 표준은 SOA에  
있어서 정말 중요한 내용이군요.  
잊지 말아야겠네요.

XML

WSDL

SOAP



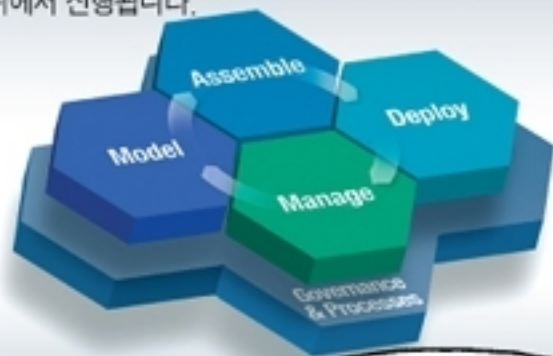


# Chapter 4.

## SOA와 라이프사이클



네. 모든 SOA 시스템은 SOA 라이프사이클을 따르고 있습니다. 즉 **모델→조립→운영→관리**의 단계를 밟아가는 것이지요. 이들 4가지 단계는 모두 SOA 거버넌스 위에서 진행됩니다.



좀 더 자세히 설명해 주세요.

**01. Model** SOA를 구현하는 첫번째는 모델 단계인데요. 비즈니스를 잘 아는 현업 담당자가 비즈니스 요건을 분석하여 어떤 서비스를 어떻게 조립할 것인가를 결정하고, 그 흐름을 정의하여 비즈니스 프로세스 모델을 만드는 단계입니다.



**02. Assemble** 두번째 단계는 모델 단계에서 만들어진 프로세스 모델과 실제 실행되는 컴포넌트를 연결하여 컴포지트 어플리케이션을 만듭니다



**03. Deploy** 세번째로는 만들어진 컴포지트 어플리케이션을 실제로 서버에서 운영하는 단계입니다.



**04. Manage** 마지막으로 운영되는 컴포지트 애플리케이션의 운영 상황을 실시간으로 모니터링하여 적절한지 혹은 개선점이 없는지 판단합니다.



만약 개선점이 있다면 다시 모델 단계로 가서 작성된 모델을 수정한 후 조립, 운영 및 관리 단계로 순환됩니다.



SOA 거버넌스는 관리 단계와 어떻게 다른지 궁금합니다?

SOA 거버넌스는 기업이 SOA를 구현하는데 있어서 보다 효율적으로 구축 관리할 수 있도록 해줍니다.

SOA 거버넌스의 과제는 사람들이 의사 결정 권한을 행사할 수 있도록 책임, 권한 및 의사 소통의 근거를 만들고

사람들이 역할과 책임을 수행할 수 있도록 평가 기준, 정책 및 관리 통제 매커니즘을 만드는 것입니다.



**SOA Governance**



그러면 SOA 거버넌스와 IT 거버넌스는 어떻게 다른지 궁금합니다?

IT 거버넌스는 IT와 관련된 의사 결정 권한 수립과 의사 결정 방식의 평가, 통제를 위한 매커니즘과 정책을 구축하는데 비해서, SOA 거버넌스는 범위를 SOA에 한정한다는 것입니다.

아, 그렇군요.



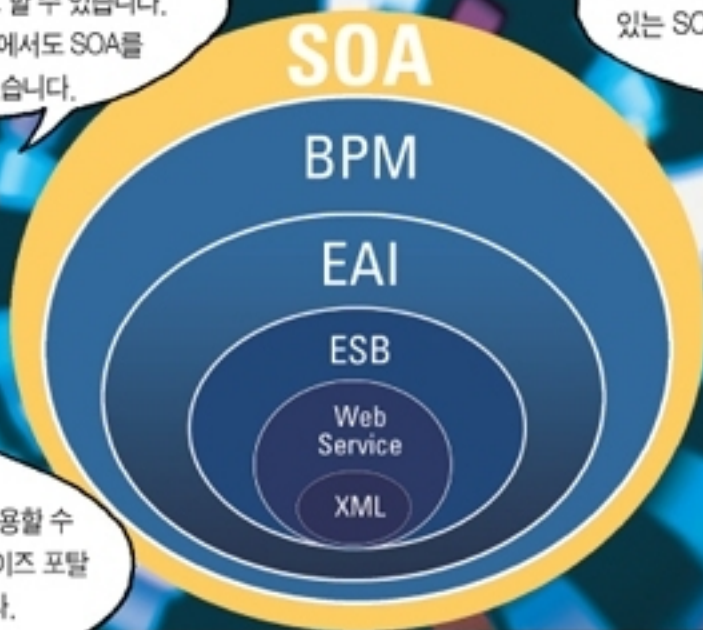
# Chapter 5.

## SOA 적용분야



맞습니다. SOA는 IT시스템 구축을 위한 철학 또는 사상이라고 할 수 있습니다. 그래서 어떠한 부분에서도 SOA를 적용할 수 있습니다.

그렇군요. 요사이 많이 이야기 되고 있는 SOA 기반 BPM 프로젝트 등이 그런 것이군요.



기본적으로 SOA는 비즈니스 프로세스를 가지는 곳에 적용할 수 있기 때문에 BPM, EAI 또는 엔터프라이즈 포탈 같은 부분에 적용할 수 있습니다.

네, SOA는 기업의 IT 시스템 어느 곳에도 적용할 수 있지만 SOA를 적용하면 효과가 큰 분야가 있습니다.

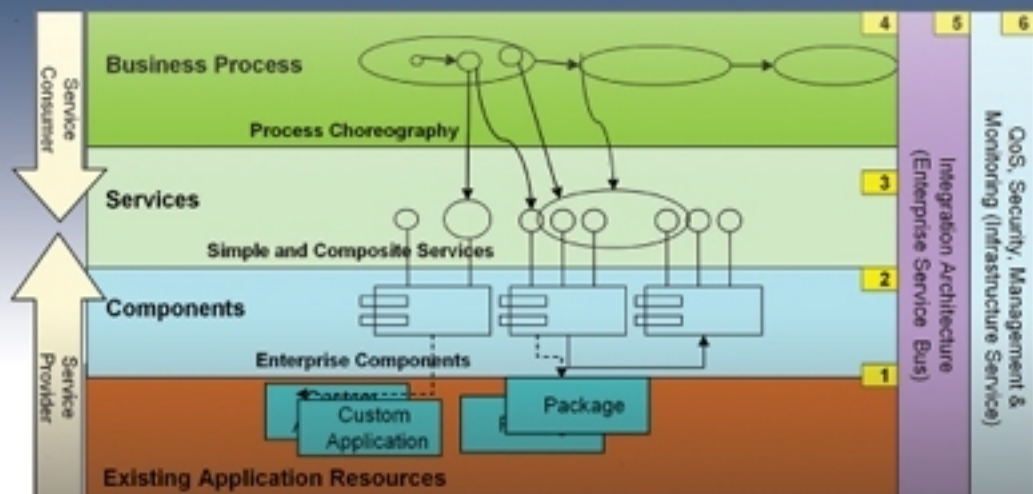
그게 어떤 거지요?

먼저 기존 시스템을 확장하고 연결하는데 SOA를 적용할 수 있습니다. 기존 시스템이 SOA 기반이 아닌 경우가 많은데 이를 SOA로 확장하여 전체 SOA 시스템에 포함될 수 있습니다.



이는 각 시스템의 연결 로직을 표준화하여 쉽게 SOA로 전환이 되게 합니다.

두번째는 SOA 기반 BPM입니다. 기존 BPM의 경우는 현업 사용자가 비즈니스 프로세스를 조합하기가 어려운 구조로 되어 있는 반면에 SOA 기반 BPM의 경우는 IT 전문가가 아닌 현업 담당자가 본인이 가장 잘 아는 비즈니스 프로세스를 손쉽게 시스템에 구현할 수 있어서 효과가 아주 큼니다.



그렇군요.

정보 통합 부분에 SOA를 적용할 수 있습니다. 정형 데이터를 저장하는 데이터베이스 뿐만 아니라 비정형 데이터를 저장하는 저장소 및 메일과 파일 시스템 등에도 정보가 저장되는데 이들 정보가 서로 통합되어 사용자에게 제공될 때 경영층이나 현업 담당자가 보다 효율적으로 의사 결정을 할 수 있도록 합니다.

그밖에 어떤 부분에 적용할 수 있나요?

정보 통합에도 사용되는군요.

네, 정보를 활용한다는 측면에서 보면 각 정보가 저장된 곳에서 어떻게 정보를 가져와서 어떻게 조합하는지 등에 관련된 프로세스가 존재하기 때문에 SOA를 적용하기가 좋습니다.

엔터프라이즈 포탈에도 적용이 가능하고 멀티 채널 통합 같은 것도 SOA 적용이 가능합니다.

그렇군요.

## Chapter 6. SOA 제품

all about the  
**SOA**



그러면 SOA 제품은 어떻게  
구성되어 있나요?



일반적으로 SOA는  
모델, 조립, 전개, 관리 부분으로 크게  
나누어지며, 이 모든 것을 통제하기 위한  
거버넌스 영역이 추가적으로 요구됩니다



모델, 조립, 전개, 관리에 대해  
세부적으로 알려 주세요



모델 영역을 위해서는  
비즈니스 모델 작성을 위한  
WebSphere Business Modeler와  
UML 구현을 위한  
Rational Software Architect를  
사용할 수 있습니다



조립 영역은 비즈니스 모델 영역에서 나온  
모델링 결과를 근거로 하여 IT에서  
사용할 수 있는 코드를 만드는 작업인데  
WebSphere Integration Developer가  
대표적인 제품입니다. 이 외에도  
Rational Application Developer 등이  
그 영역에 해당됩니다.



전개 영역은 크게 인프라 구현과 프로세스 통합으로 분리됩니다.

인프라 구축을 위해서는 WebSphere ESB라고 하는 제품을 중심으로 기술 어댑터 및 패키지 어플리케이션 어댑터를 활용할 수 있고, 프로세스 통합을 위해서는 WebSphere Process Server 라는 제품이 사용됩니다. WebSphere Process Server 내에는 WebSphere ESB가 포함되어 있기 때문에 인프라 구축 영역의 기능을 모두 사용할 수 있습니다. 그 외에도 사용자 통합을 위한 WebSphere Portal이나 정보 통합을 위한 WebSphere Information Server 등을 필요에 따라 추가하여 활용할 수 있습니다



관리 영역은 IT 관리와 비즈니스 관리로 나누어지는데 IT 자산의 관리를 위해서는

Tivoli Composite Application Manager가 사용됩니다. 반면 KPI 등을 측정하고 프로세스를 모니터링하는 비즈니스 관리를 위해서는 WebSphere Business Monitor가 사용됩니다



그렇다면 거버넌스를 구성하는 제품은 따로 있나요?

SOA 거버넌스의 경우 특정 제품만으로 완벽하게 구현 가능하다 라고 이야기하기는 힘듭니다.

고객이 SOA 자산의 관리를 어떤 식으로 구현할 것인가에 따라 각각 달라질 수 있는 서비스 영역이 포함되어 있기 때문이죠.

다만 일반적으로 거버넌스를 계획하고 정의하고 동작하고 평가하는 4단계 방법을 주로 사용하며, 이를 위해 해당하는 다양한 제품의 조합을 제공합니다.

전개 영역은 크게 인프라 구현과 프로세스 통합으로  
분리됩니다.

인프라 구축을 위해서는 WebSphere ESB라고 하는  
제품을 중심으로 기술 어댑터 및 패키지 어플리케이션  
어댑터를 활용할 수 있고,  
프로세스 통합을 위해서는 WebSphere Process Server  
라는 제품이 사용됩니다. WebSphere Process Server  
내에는 WebSphere ESB가 포함되어 있기 때문에 인프라  
구축 영역의 기능을 모두 사용할 수 있습니다.  
그 외에도 사용자 통합을 위한 WebSphere Portal이나  
정보 통합을 위한 WebSphere Information Server 등을  
필요에 따라 추가하여 활용할 수 있습니다



관리 영역은 IT 관리와 비즈니스 관리로 나누어지는데  
IT 자산의 관리를 위해서는

Tivoli Composite Application Manager가  
사용됩니다. 반면 KPI 등을 측정하고 프로세스를  
모니터링하는 비즈니스 관리를 위해서는  
WebSphere Business Monitor가 사용됩니다



그렇다면 거버넌스를 구성하는  
제품은 따로 있나요?

SOA 거버넌스의 경우 특정  
제품만으로 완벽하게 구현 가능하다  
라고 이야기하기는 힘듭니다.

고객이 SOA 자산의 관리를 어떤 식으로  
구현할 것인가에 따라 각각 달라질  
수 있는 서비스 영역이 포함되어  
있기 때문이죠.

다만 일반적으로 거버넌스를  
계획하고 정의하고 동작하고  
평가하는 4단계 방법을 주로 사용하며,  
이를 위해 해당하는 다양한 제품의  
조합을 제공합니다.

일반적으로 계획 및 정의 단계에서는 Rational Method Composer, Rational Portfolio Manager 를 많이 사용하며

동작 단계에서는 WebSphere Service Registry & Repository 를 이용하고,



평가 단계에서는

WebSphere Business Monitor, Tivoli Composite Application Manager for SOA 등을 활용합니다.

네, 정말 다양한 제품들이 사용되는군요.



적용분야, 제품에 이르기까지에 대해 간략하게 설명드렸습니다. 많은 도움이 되셨는지요?



네. SOA는 기업의 유연한 IT 환경을 위해, 선택이 아닌 필수적인 방법론인 것 같습니다.

이제 어느 정도 SOA에 대한 자신감이 생긴 것 같습니다. 정말 감사합니다.



네. SOA를 통해 앞으로 기업의 비즈니스에 많은 도움을 주시길 부탁드립니다.

그럼요, 더 열심히 해야죠 다음에도 많은 가르침 부탁드립니다.



END