

인터페이스 구현

(응용SW개발)

인터페이스 기능 구현 솔루션



한국기술교육대학교
온라인평생교육원

학습 내용

- 인터페이스 연동
- EAI, ESB, MCA

학습 목표

- 인터페이스 연동에 대하여 설명하고 분석할 수 있다.
- 인터페이스 기능 구현 솔루션인 EAI, ESB, MCA에 대하여 설명할 수 있다.

1. 인터페이스 연동

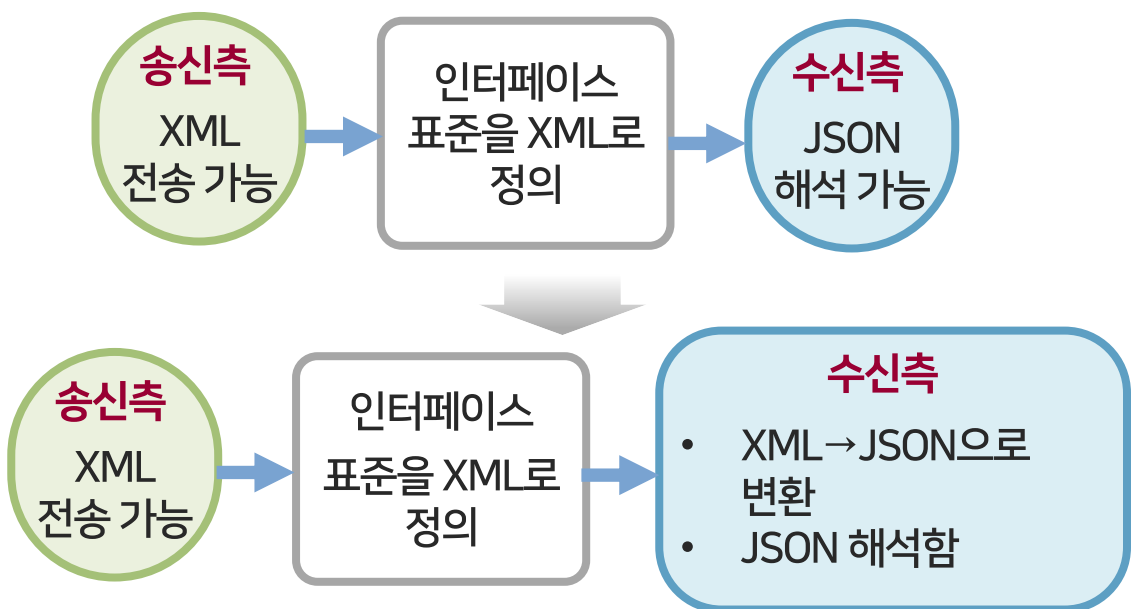
1 인터페이스 연동 개요

◆ 인터페이스 연동 개요(1대1)

인터페이스
연동

송신과 수신 단이 상호 통신을 통한 인터페이스 정보를 주고 받기 위해 서로 이해할 수 있는 데이터의 형식, 규칙, 전송 방법들을 정의한 **인터페이스 데이터 표준을 가지고 인터페이스를 연동하는 것**

- 송신과 수신 측에 인터페이스 표준을 정하여 사용하는 경우 송·수신 한쪽이 변환 또는 매핑하여 표준에 맞춰 처리하고 사용



1. 인터페이스 연동

1 인터페이스 연동 개요

◆ 인터페이스 연동 개요(N대1, 1대N, N대N)

송신과 수신 관계가
N대 1, 1대 N, N대 N

+

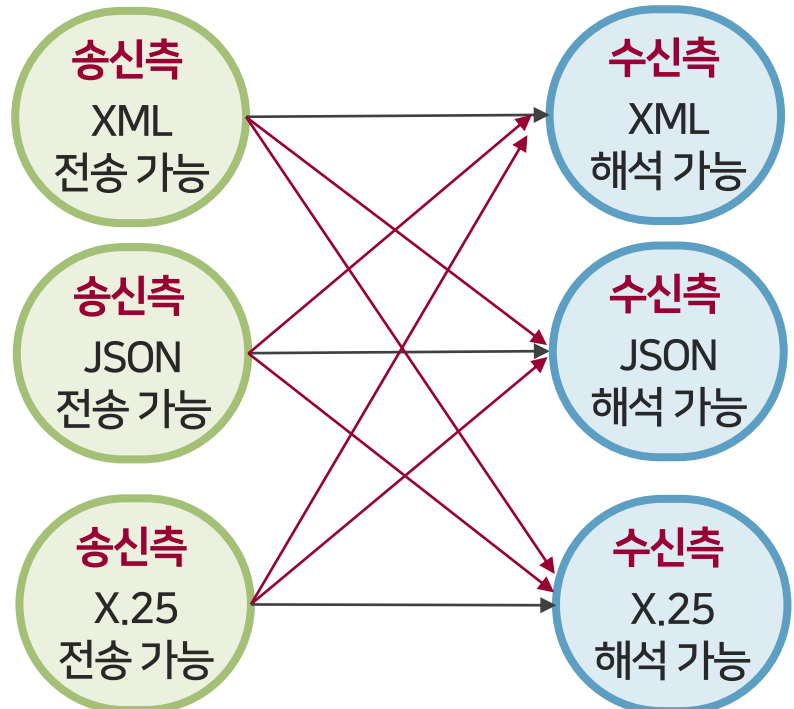
해당 송·수신이 전송하거나
이해할 수 있는 인터페이스
표준이 제각각인 경우

변환 또는 매핑 작업이 수시로 이루어져야 하는 경우가 발생

인터페이스
표준을
XML로 정의

→
인터페이스 표준이
일치하지 않아
변환이나 매핑 필요

→
인터페이스 표준일치



1. 인터페이스 연동

2 인터페이스 연동 방식

◆ 인터페이스 연동 방식

직접 연동 방식

중간 매개체 없이 송신 시스템과
수신 시스템이 직접 연계되는 방식

- ◆ 어댑터와 역할을 하는 부분이 인터페이스 표준 변환을 수행하는 방식
- ◆ 1대 1연계로 사용되는 경우 유용

「응용 프로그램 작성」으로 인터페이스 표준을 맞추는 작업을 수행함

간접 연동 방식

연계 솔루션과 같이 중간 매개체를
활용하여 간접 연계하는 방식

- ◆ 특정한 솔루션 매개체를 두고 해당 솔루션에서 다양한 인터페이스의 표준 변환 작업을 함
- ◆ 1대 N, N대 1, N대 N 방식에서 유용함

「연동 솔루션」으로 인터페이스 표준을 맞추는 작업을 수행

1. 인터페이스 연동

2 인터페이스 연동 방식

❖ 인터페이스 연동 방식

간접 연계 방식의 송신 시스템

- ❖ 운영 데이터베이스(DB: Database)에서 연계 데이터를 생성 및 변환
- ❖ 송신 로그(Log)를 모니터링하는 구현 대상 모듈과 솔루션(Solution)에서 제공하는 송·수신 엔진(Engine), 어댑터(Adapter)로 구성됨

간접 연계 방식의 수신 시스템

- ❖ 데이터를 수신 받아 인터페이스 테이블 또는 파일로 생성하는 송·수신 엔진(Engine), 어댑터(Adapter)와 수신된 데이터를 변환하여 운영 DB에 반영
- ❖ 데이터 수신 및 운영 DB 반영을 모니터링 하는 구현 모듈로 구성됨

1. 인터페이스 연동

2 인터페이스 연동 방식

◆ 인터페이스 연동 방식 별 장단점

직접 연동 장점

- 연계 및 통합 구현이 단순하며 용이함
- 개발 소요 비용 및 기간이 짧음
- 중간 매개체가 없어서 데이터 연계처리 성능이 대체로 좋음

직접 연동 단점

- 시스템 간의 결합도가 높아서, 시스템 변경에 민감함
(시스템 변경으로 인해 장애 발생 가능)
- 보안을 위한 암호·복호화 처리, 비즈니스 로직(Business Logic, 업무 처리 내용) 적용 등이 불가
- 연계 및 통합 가능한 시스템 환경이 제한적

간접 연동 장점

- 서로 상이한 네트워크, 프로토콜(Protocol) 등 다양한 환경의 연계 및 통합 가능
- 시스템 간 인터페이스 변경 시에도 장애나 오류 없이 서비스 가능
- 보안이나 비즈니스(Business, 업무) 처리를 위한 로직(Logic)을 자유롭게 반영

간접 연동 단점

- 연계 아키텍처 및 메커니즘이 복잡하고, 중간 매개체로 인해 성능 저하 요소 존재
- 개발 및 적용을 위한 테스트 기간이 상대적으로 장기간 소요됨

1. 인터페이스 연동

2 인터페이스 연동 방식

❖ 직접 연동 방식 예시

1

DB LINK

- 데이터베이스에서 제공하는 객체(오브젝트)를 이용
- 수신 시스템 Database에 송신 시스템에서 접근 가능한 DB-Link 객체를 하나 생성한 후 송신시스템에서 해당 DB-Link 명으로 직접 참조하여 연계



예 | 테이블 명@DBLINK명

2

DB Connection-Pool)

- 수신 시스템 WAS에서 송신 시스템 DB로 연결되는 Connection Pool 생성
- 프로그램 소스에서 WAS에 설정된 Connection Pool 명을 참고하여 구현



예 | 송신 시스템 데이터 소스(DS: Data Source)
(WAS에서 설정한 Connection Pool명)

1. 인터페이스 연동

2 인터페이스 연동 방식

📍 직접 연동 방식 예시

3**JDBC/ JNDI**

- 데이터 베이스 서버와 어플리케이션 서버간 인터페이스를 위해 제공되는 데이터베이스 인터페이스 연동 모듈



예 | JDBC-connection

4**URL 링크**

- 웹 애플리케이션에서 URL을 링크하여 데이터를 받음 (XML, JSON, REST-API 등)



예 | http://주소~

1. 인터페이스 연동

2 인터페이스 연동 방식

📌 직접 연동 방식 예시

1

EAI

- Enterprise **A**pplication **I**ntegration
- N대 N 형태의 다양한 송·수신 관계에서 송·수신 표준 인터페이스 변환, 송·수신 처리 및 진행 모니터링, 송·수신 로드 밸런싱 등을 EAI 솔루션 서버의 Adaptor 등으로 처리함

2

ESB/Web Service

- Enterprise **S**ervice **B**us
- ESB 솔루션에서 서비스 간 연동을 위한 변환 처리로 다중 플랫폼을 지원
- 웹 서비스(WSI, UDDI, SOAP) 표준을 이용한 시스템간 연계

1. 인터페이스 연동

2 인터페이스 연동 방식

❖ 직접 연동 방식 예시

3

MCI/MCA

- **M**ulti **C**hannel **I**ntegration/**M**ulti **C**hannel **A**rchitecture)
- 채널 시스템으로 변환하는데 중점을 두고,
채널 관점의 인터페이스 연동을 처리하는 솔루션
(EAI/ESB는 내·외부 시스템 간 인터페이스 연동을 지원하는데 중점)

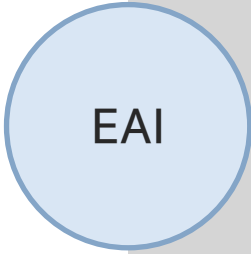
[계좌이체 서비스]

예 | 해당 업무는 동일한 Back-end 로 처리되고 있지만 해당 서비스를 인터넷(웹), 모바일, ARS 등 다양한 채널을 통해 제공해야 함

2. EAI, ESB, MCA

1 EAI, ESB

🔗 EAI(Enterprise Application Integration)



기업 내 또는 기업 간에 복수의 이질적 응용 프로그램들이 상호 융화 혹은 동기화 되어 동작하도록 하는 것

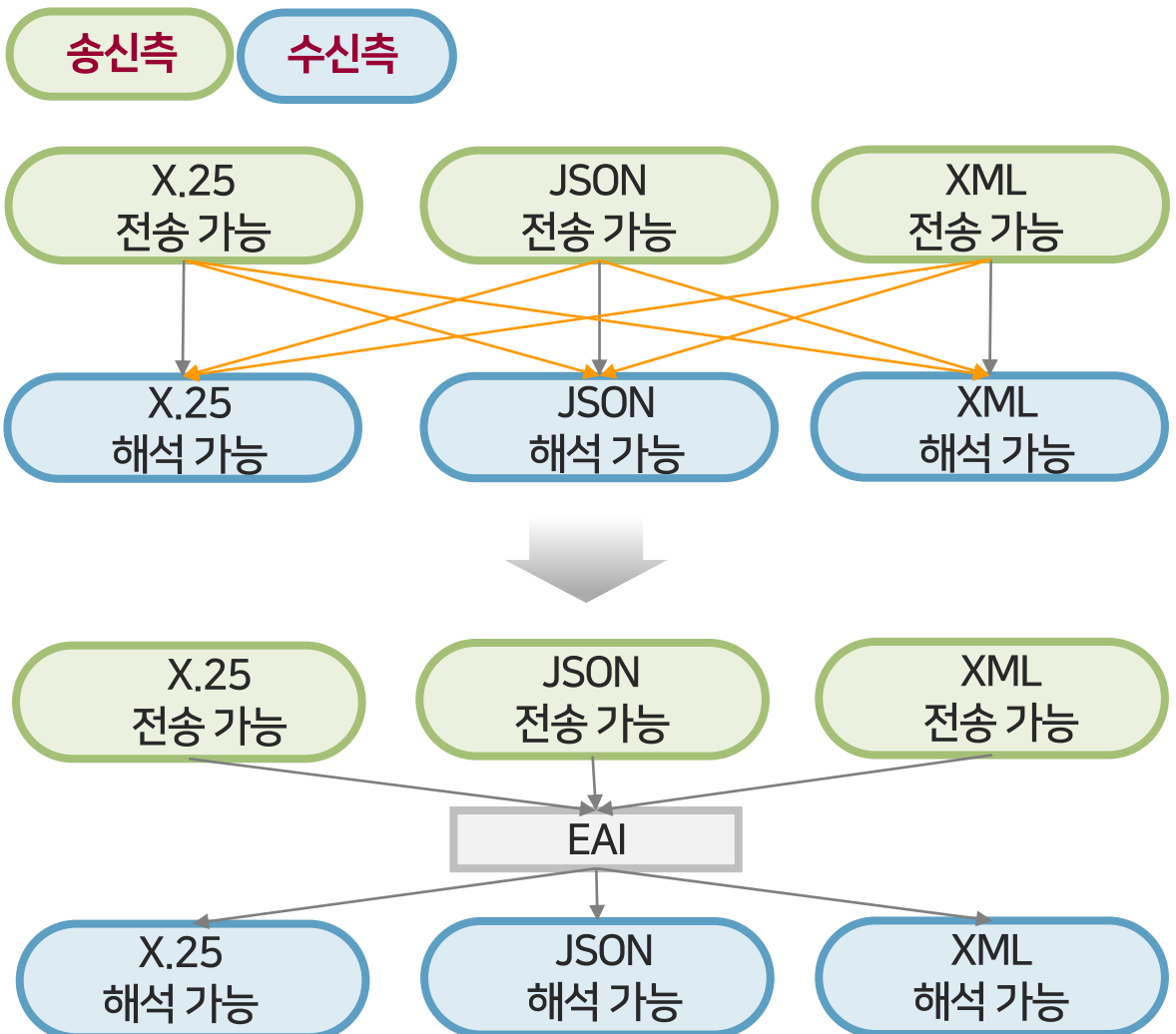
- 🔗 N대 N 형태의 다양한 송·수신 관계에서 송·수신 표준 인터페이스 변환, 송·수신 처리 및 진행 모니터링, 송·수신 로드 밸런싱 등을 EAI 솔루션 서버의 Adaptor로 처리함

2. EAI, ESB, MCA

1 EAI, ESB

❖ EAI(Enterprise Application Integration)

- ❖ Front-Office 시스템, 기존의 Legacy 시스템, Packaged 어플리케이션 등의 전사적으로 산재 되어 있는 어플리케이션을 프로세스 및 메시지(Message) 차원에서 통합하고 관리



2. EAI, ESB, MCA

1 EAI, ESB

❖ EAI 통합 패턴

EAI 시스템
구현 유형

중개

연합

- 동시에 일어나기도 함
- 같은 EAI 시스템은 다중 중계 시스템으로 유지됨
- 외부 사용자 요청에 대해 연합 서비스 제공 가능

중개

- EAI 시스템은 여러 응용프로그램 사이에서 **중개자** 또는 **브로커의 역할**을 수행
- 응용프로그램에서 어떤 상황이 발생하면, EAI 시스템의 통합 모듈에 통보됨
- 해당 모듈은 다른 관련된 응용 프로그램에게 전파됨

연합

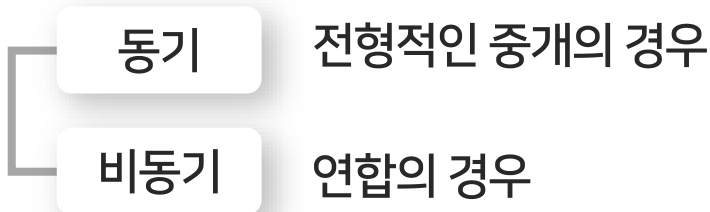
- EAI 시스템은 모든 응용프로그램들의 **최상위에 위치함**
- 모든 외부로부터의 접속은 EAI를 통해 이루어짐
- EAI 시스템은 외부로 적절한 정보와 인터페이스를 제공하기 위해 구성되고, 요청자를 위한 응용프로그램과 서로 작업을 수행하게 됨

2. EAI, ESB, MCA

1 EAI, ESB

❖ EAI 통합 패턴

❖ EAI는 동기과 비동기 접근을 모두 지원함



EAI 방식

각 터미널
집중 방식 모델
(Hub-and-Spoke)

- 중앙에 EAI 시스템이 존재하고 (HUB구조), 연관된 응용프로그램들은 연결대를 통해 상호 교류하는 것

버스 모델

- EAI 시스템이 버스가 되거나, 이미 구현되어 있는 메시지 버스에 내재되어 운영됨
- 메시지 지향 미들웨어 라고도 함

2. EAI, ESB, MCA

1 EAI, ESB

◆ EAI 구성요소와 구축 유형

EAI 구성요소

1 Adapter

데이터 추출 지원, 다양한 형태 존재
(파일 어댑터, DB 어댑터)

2 Message Hub

목표 시스템으로 데이터 분기 및 목표 시스템에
적합한 포맷으로 데이터 작성 수행

3 Workflow

프로세스 통제, 자동화 및 통합 기능

4 데이터 전송 미들웨어

시스템간 데이터 전송, 통신 프로토콜, 하드웨어,
네트워크 상태와 무관하게 전달

2. EAI, ESB, MCA

1 EAI, ESB

◆ EAI 구성요소와 구축 유형

EAI 구축 유형

1 Hub & Spoke

- 모든 데이터가 허브를 통해 전송
- 데이터 전송이 보장되고, 유지보수 비용이 절감됨

2 메세징 버스

- 데이터 전송하는데 버스를 이용하여 병목 현상 발생 가능
- 대량의 데이터 교환에 적합함

3 하이브리드

- Hub & Spoke 방식과, 메시징 버스 방식의 통합
- 유연한 통합 작업 가능

2. EAI, ESB, MCA

1 EAI, ESB

◆ EAI 주요기능

1 비즈니스 프로세스 관리기능

각 업무 시스템 및 APP 상호간에 데이터 교환과 더불어 각 업무 흐름을 어떤 시점 또는 어떤 이벤트에 따라서 어디에서 어디로 업무가 진행되어야 하는지를 정의하고 운용할 수 있는 기능

2 데이터 브로커 기능

App 상호간에 중개되는 데이터를 자동 변환하여 전달하고 데이터 소스에서 지정된 대상 시스템까지 연결하는 기능

3 APP 접근 기능

- 패키지 APP 또는 메인 프레임과 같은 이기종 시스템과의 접속을 위한 기능
- 해당 SW와 플랫폼 사이에 위치하며 데이터 중개 및 APP 연동의 인터페이스를 제공

2. EAI, ESB, MCA

1 EAI, ESB

📦 EAI 주요기능

4 데이터 접근 기능

- 데이터 통합을 담당하는 영역
- 데이터의 전송, 타입 변환, 데이터 정제 및 추출 기능

5 플랫폼 기능

- EAI의 기반이 되는 APP 서버 또는 미들웨어로 구성되어 있는 영역
- EAI를 안정성 있게 실행하고 EAI 모든 기능들이 정상적으로 동작할 수 있도록 하는 기능

2. EAI, ESB, MCA

1 EAI, ESB

🔹 EAI 통합 절차

1단계 데이터 통합

어플리케이션에 필요한 데이터를
요구하는 형태로 변환

- 데이터 소스 변환
- 가장 많이 사용
- 리스크 적고 비용 저렴



2단계 Object 통합

공통 Object의 통합 및 Mapping

- 통합 메시지, API 등 같은
어플리케이션 입·출력에 의해 수행



2. EAI, ESB, MCA

1 EAI, ESB

📦 EAI 통합 절차

3단계 프로세스 통합

기업내 업무 규칙에 따라
Application 간 메시지 교환

- 3단계로 통합
 - 가능 솔루션 규정
 - 솔루션 시뮬레이션
 - 비즈니스 프로세스 최적화
- 프로세스 자동화 방식,
워크 플로우 자동화



4단계 비즈니스 통합

기업간 업무 프로세스들의
메시지 라우팅에 의한 통합

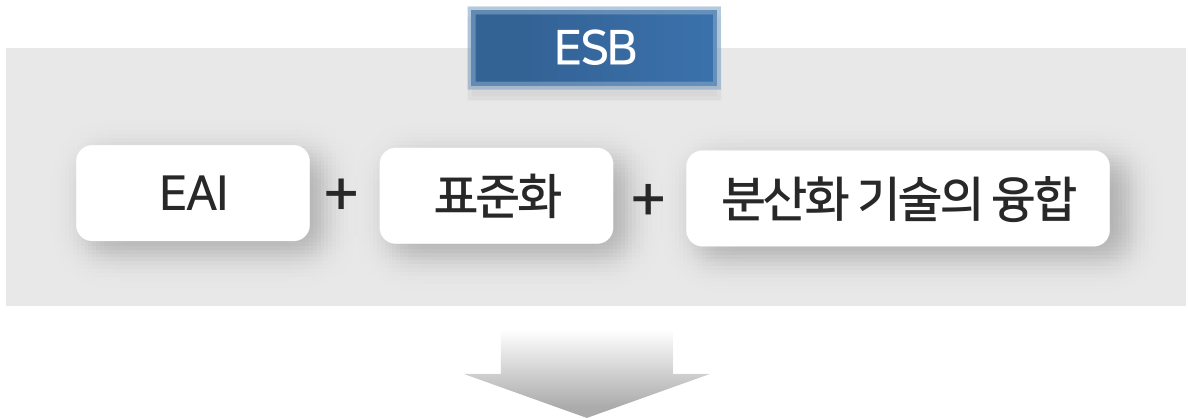
- 중개/메시지 교환 개념
 - 데이터 공유
 - 비즈니스 규칙
 - 제한사항 및 실시간 정보공유

2. EAI, ESB, MCA

1 EAI, ESB

ESB(Enterprise Service Bus)

- ESB 솔루션에서 서비스 간 연동을 위한 변환 처리로 다중 플랫폼을 지원
- 웹 서비스(WSDL/UDDI/SOAP) 표준을 이용한 시스템간 연계
- 실제로 ESB 솔루션은 웹 서비스를 지원하는 진보된 EAI 솔루션 정도로 이해하는 경향이 있음
- SOA를 지원하는 서비스와 어플리케이션 컴포넌트간의 연동을 지원하는 미들웨어 플랫폼



ESB(Enterprise Service Bus)가 주목 받은 이유는?

- EAI에서 Hub and Spoke 통합 방식의 한계점 발견
- SOA의 핵심 필요 기능으로 중요성 증대

2. EAI, ESB, MCA

1 EAI, ESB

ESB(Enterprise Service Bus)

- 느슨하게 결합되었거나 결합되지 않은 구성 요소들 간에 중재적인 관계와 직접 통신을 지원하는 웹 서비스가 가능한 인프라 = Gartner Group

ESB란 표시의 의미는?

그 제품이 MOM과 웹 서비스 프로토콜 모두를 지원하는 일종의 통합 미들웨어 제품이라는 뜻

- ESB는 벤더의 독자 기술을 판매하기 위한 일종의 마케팅 전술
채펠 앤 어소시에이트 CEO

2. EAI, ESB, MCA

1 EAI, ESB

ESB 구성요소

1 어댑터 형태의 레거시 연동 컴포넌트

- ESB는 기본적으로 다양한 표준 프로토콜을 어댑터(Adapter) 형태로 지원해야 함
- 통합 개발 환경에서 플러그블(Pluggable) 할 수 있도록 간단한 형태여야 함

2 메시지의 변환, 가공

- 데이터 포맷과 형태 등을 통합 개발 환경을 통해 자유롭게 변화하고 가공 할 수 있음

3 BPM

- **B**usiness **P**rocess **M**anagement
- 단순한 인터페이스 연동 뿐만 아니라 기업의 비즈니스 절차 관리와 연동되어 처리할 수 있는 서비스를 제공함

예 | 은행의 이자율이 변함에 따라 인터페이스 변경이 처리될 수 있도록 솔루션이 도움을 줌

- 다양한 시스템 데이터와 애플리케이션과의 연동을 통해 하나의 서비스로 제공

2. EAI, ESB, MCA

1 EAI, ESB

ESB 구성요소

4 컨트롤과 모니터링

- 여러 과정들은 통합적인 조작과 모니터링이 가능해야 함
- 모니터링의 대상
 - 어댑터 단에서 진행되고 있는 프로세스의 상태 및 데이터 값

5 통합개발환경

- 통합개발환경은 ESB에서 필요한 요소 중 하나
- 인터페이스 연동 시 편리한 개발환경을 제공해야 함

2. EAI, ESB, MCA

1 EAI, ESB

ESB와 EAI의 비교

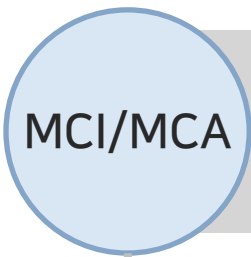
실제로 ESB 솔루션은 웹 서비스를 지원하는
진보된 EAI 솔루션 정도로 이해하는 경향이 있음

특징	전통적인 EAI	ESB
통합의 종류	어플리케이션 통합	서비스 통합, 서비스 호스팅
통합 방안	시스템 별 어댑터 사용으로 복잡성 증가	표준 기술 사용한 단순 통합
표준	벤더 별 전송기술 상이	개방형 표준(웹서비스)
통합형태	단단한 결합(Static, 1:1 결합)	느슨한 결합(Dynamic, 1:N 결합)
비용	통합대상 시스템별 어댑터 구입 또는 개발로 지속적 비용 발생	동일 표준 기반이므로 추가 개발비용 절감, 비즈니스 로직 재사용을 통한 비용 절감
구현 아키텍처	집중형(Hub & Spoke)	분산형(Distributed)

2. EAI, ESB, MCA

2 MCA

❖ MCA(Multi Channel Integration / Multi Channel Architecture)



- 채널 시스템으로 변환의 중점을 두는 솔루션
- 채널 관점의 인터페이스 연동을 처리하는 솔루션



EAI/ESB는 내·외부 시스템 간의
인터페이스 연동을 지원하는데 중점

계좌이체 서비스에서, 해당 업무는 동일한 Back-end로
예 | 처리되고 있지만 해당 서비스를 인터넷(웹), 모바일, ARS 등
다양한 채널을 통해 제공해야 함

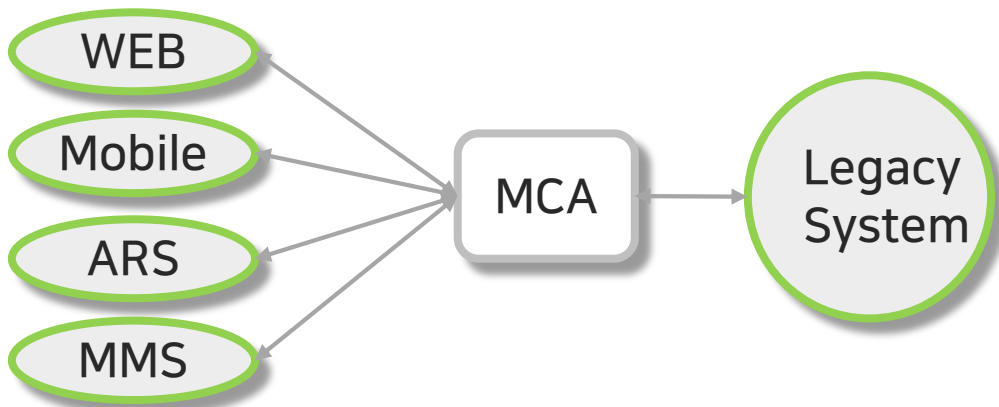
2. EAI, ESB, MCA

2 MCA

❖ MCA(Multi Channel Integration / Multi Channel Architecture)



다양한 채널(단말, 자동화기기, 콜센터, 모바일 등)을 통합하는 기술 기반을 구축하여
업무시스템의 채널 의존성을 제거하고,
모든 채널이 정보를 공유하는 채널 통합 시스템



2. EAI, ESB, MCA

2 MCA

❖ MCA의 필요성

1 업무의 채널 종속성

- 시스템 개발 시 모든 채널을 고려하여 업무 프로그램을 구현
- 채널의 변경이나 추가에도 업무 프로그램의 변경 필요

2 채널 기능의 중복 개발

- 각 채널은 모든 업무 시스템에 대한 채널 기능을 중복 개발
- 업무시스템이 변경되면 모든 채널을 같이 변경해야 하는 역 의존성이 발생

3 채널 기능의 중복 개발

- 각 채널은 모든 업무 시스템에 대한 채널 기능 중복 개발
- 업무시스템이 변경되면 모든 채널을 같이 변경해야 하는 역 의존성이 발생

4 인터페이스 중복 개발

- 각 채널은 모든 업무 시스템에 대한 인터페이스를 중복 개발
- 업무시스템 추가 시 채널 인터페이스 추가 개발 필요

2. EAI, ESB, MCA

2 MCA

◆ MCA의 주요기능

1 거래 처리

- **프로토콜 요청 접수**: HTTP, TCP/IP 프로토콜
- **메시지 변환 기능**: XML방식 메시지 정의, 다양한 포맷 지원, 메시지 변환, 코드 변환
- **메시지 라우팅**: 메시지 전송/조립, 거래 맵 상의 스크립트지원 등을 통한 룰의 동적 변경 지원

2 인터페이스 및 대용량 관리

- Channel 별 인터페이스 방안 제공
- 호스트 및 서버 인터페이스 (SNA, TUXEDO 등 프로토콜제공)

3 통합 개발환경 및 운영 환경 지원

- Message 및 Workflow Designer
- 사용자 관리(채널 시스템 관리자)
- 관리자에 대한 권한 관리
- 파일 송·수신 기능
- Transaction 분석 및 제한

2. EAI, ESB, MCA

2 MCA

◆ MCA의 주요기능

4 장애 복구 서비스

- 멀티 Clustering Server 환경에 가용한 장애 메커니즘
- 보안 인프라 연계
- 채널 서비스(거래 제어, 승인, 저널링 등)

5 채널 서비스

- 채널 세션을 관리하는 기능
- RULE 및 CONFIG DATA 저장소 관리
- 로그, 에러 정보 관리

2. EAI, ESB, MCA

2 MCA

📦 MCA의 구현 목적

1 채널 개발 비용 절감

- 프로그램 내 분산, 중복 개발된 채널 기능을 통합함으로써 중복 기능을 제거하고 채널 비용을 절감

2 고객 중심 채널 인프라 확보

- 고객의 채널 선택과 무관하게 동일한 수준의 서비스를 제공하여 채널의 제약 사항에 따른 고객 불편 감소

3 비즈니스 경쟁력 강화

- 채널 독립적이고 비즈니스 중심적인 업무시스템 구축으로 고객의 채널별 접촉 이력을 수집
- 확장된 마케팅 자료를 제공하여 고객 확보 가능

2. EAI, ESB, MCA

2 MCA

❖ MCA의 구현 목적

4 고객 서비스 품질 향상

- 채널 통합으로 다양한 채널에서 동일 정보의 공유 및 정보 전달의 유연화
- 현장업무 지원의 가속화 및 고객의 정보를 통합 제공함으로써 Cross Sale 및 Up sale의 기회가 확대되어 영업 생산성 향상

5 업무시스템 Interface 통합

- 채널 별로 분산된 업무 시스템 인터페이스를 통합, 단순화하여 채널을 경량화
- 시스템 도입 및 운영 비용을 최소화하여 TCO를 절감
- 각 채널 별로 배치해야 했던 IT 요원의 수요를 최적화하여 효율적인 인력 운용 가능

핵심요약

인터페이스 연동

- 송신과 수신 단이 상호 통신을 통한 인터페이스 정보를 주고 받기 위해 서로가 이해할 수 있는 데이터들의 형식이나 규칙, 또는 전송 방법들을 정의한 인터페이스 데이터 표준을 가지고 인터페이스를 연동함
- 1대 1, 1대 N, N대 1, N대 N과 같은 형태가 있을 수 있음
- 인터페이스 연계 방식
 - ① 직접 연동 방식의 예
DB link, DB Connection Pool, JDBC, URL link
 - ② 간접 연동 방식의 예
EAI, ESB/Web Service, MCI/MCA가 있음

핵심요약

EAI, ESB, MCA

- N대 N형태의 다양한 송·수신 관계에서 송·수신 표준 인터페이스 변환, 송·수신 처리 및 진행 모니터링, 송·수신 로드 발랜싱 등을 EAI 솔루션 서버의 Adaptor를 통하여 처리
- **ESB 솔루션**은 웹 서비스를 지원하는 진보된 EAI 솔루션 정도로 이해하는 경향이 있음
- **ESB**는 SOA를 지원하는 서비스와 어플리케이션 컴포넌트간의 연동을 지원하는 미들웨어 플랫폼
- **EAI/ESB**가 내·외부 시스템 간의 인터페이스 연동을 지원하는데 중점을 두었다면 **MCI/MCA**는 채널 시스템 변환에 중점을 두는 솔루션