

# 통합 구현

## 1. 연계 데이터 구성

### (1) 연계 요구사항 분석

#### ▼ 개념

- 서로 다른 두 시스템, 장치, 소프트웨어를 이어주는 중계 역할을 하는 연계 시스템과 관련된 요구사항 분석 과정

#### ▼ 분석 기법

- 인터뷰 - 사용자 면담
- 체크리스트 - 시스템 운영 환경, 성능, 보안, 데이터 발생 주기 등의 기준에 대한 점검을 통한 분석 기법
- 설문지 - 서비스 활용 목적에 따라 연계 필요 데이터 식별 및 연계 주기 등을 설문을 통해 분석하는 기법
- 델파이 기법 - 전문가의 경험적 지식을 통한 분석 기법
- 브레인스토밍 - 구성원들이 자연스럽게 제시된 아이디어 목록을 통해 분석

#### ▼ 참고 문서

- 코드 정의서
- 개체 정의서

데이터베이스 개념 모델링 단계에서 도출한 개체의 타입, 관련 속성, 식별자 등의 정보를 개괄적으로 명세화 한 정의서

- 테이블 정의서

논리 및 물리 모델링 과정에서 작성하는 설계 산출물로 테이블을 구성하는 컬럼들의 특성, 인덱스, 업무 규칙을 문서화

컬럼 특성 - 데이터 타입, 길이, 널 여부, 키 여부, 디폴트 값

- 응용 프로그램 구성도

애플리케이션의 메뉴 구성, 화면 설계, 데이터 발생 지점, 주기, 발생 패턴 등 확인

메뉴 구조도, 화면 설계서, 사용자 인터페이스 정의서

- 시스템 구성도

하드웨어, 소프트웨어, 네트워크 등 연계 대상 시스템 구성도

- 인터페이스(연계) 명세서

인터페이스 정의서에 작성한 인터페이스 ID 별로 송수신하는 데이터 타입, 길이 등 인터페이스 항목을 상세하게 작성

#### ▼ 구성 요소

- 인터페이스 ID
- 최대 처리 횟수

단위 시간당 처리될 수 있는 해당 인터페이스 최대 수행 건수

- 데이터 크기
- 시스템 정보
- 데이터 정보

#### ▼ 분석 절차

1. 시스템 현황 확인
2. 정의서 확인
3. 체크리스트 작성
4. 인터뷰 및 면담
5. 연계 요구사항 분석서 작성

#### ▼ 구성

- 송신 시스템과 수신 시스템으로 구성할 수 있으며, 연계 방식에 따라 중계 서버를 둘 수 있음
- 송신 시스템

연계할 데이터를 데이터베이스와 애플리케이션으로 부터 연계 테이블 또는 파일 형태로 생성하여 송신하는 시스템

- 수신 시스템

수신한 연계 테이블 또는 파일의 데이터를 수신 시스템에서 관리하는 데이터 형식에 맞게 변환하여 데이터베이스에 저장하거나 애플리케이션에서 활용할 수 있도록 제공

하는 시스템

- 중계 서버

송수신 시스템 사이에서 데이터를 송수신하고 연계 데이터의 송수신 현황을 모니터링 하는 시스템

## (2) 연계 데이터 식별 및 표준화

### 1) 연계 데이터 식별

- 송수신 시스템 데이터에 대한 상세 식별 정보

### 2) 연계 데이터 표준화

- 인터페이스 시스템 사이에서 교환되는 데이터는 규격화된 표준 형식을 정의하여 사용해야 함
- 데이터, 공통 코드 정보 누락 없이 식별하고 인터페이스 명세서 작성
- 인터페이스 데이터 공통부(인터페이스 표준 항목 포함), 개별부(업무 처리에 필요한 데이터 포함), 종료부(종료 표시)로 구성

### 3) 연계 데이터 식별 및 표준화 절차

1. 연계 범위 및 항목 정의 - 시스템 간의 연계 하려는 정보를 구체화하여 범위와 항목을 정의. 각 항목의 데이터 타입 및 길이, 코드화 여부 등 확인
2. 연계 코드 변환 및 매핑 - 연계 대상 범위 및 항목 식별 이후, 연계 정보 중 코드로 관리되는 항목을 변환. 코드로 관리되는 정보는 검색 조건으로 활용
3. 인터페이스(연계) 정의서 작성 - 정의된 연계 항목의 데이터 타입, 길이, 공통 코드 여부 등의 속성을 확인하고 연계 정의서 및 명세서 작성

## 2. 연계 메커니즘 구성

### (1) 연계 메커니즘

#### ▼ 개념

- 응용 소프트웨어와 연계 대상 모듈 간의 데이터 연계 시 요구 사항을 고려한 **연계 방법과 주기를 설계**하기 위한 메커니즘

#### ▼ 기능

- 데이터를 생성하여 전송하는 송신 시스템과 송신 데이터를 수신하여 DB에 반영하는 수신 시스템으로 구성
- 송신 시스템 - 운영 데이터베이스, 애플리케이션으로 부터 **연계 데이터를 연계 테이블 또는 파일로 생성**하여 송신
- 수신 시스템 - 송신 시스템으로부터 전송된 데이터를 **변환 처리 후** 데이터 베이스에 반영

## ▼ 연계 방식

### ▼ 직접 연계 방식

#### ▼ 분류

- DB 링크

수신 시스템에서 DB 링크를 생성하고 송신 시스템에서 해당 DB 링크를 직접 참조

EX) 테이블명@DBLink명

- DB 연결

수신 시스템의 WAS에서 송신 시스템 DB로 연결하는 DB 커넥션 풀을 생성하고 연계 프로그램에서 해당 DB 커넥션 풀 명을 이용

EX) 송신 시스템의 Data Source = DB Connection Pool 이름

- API/Open API

송신 시스템의 DB에서 데이터를 읽어서 제공하는 애플리케이션 프로그래밍 인터페이스 프로그램

- JDBC

송신 시스템의 DB와 연결

- 하이퍼링크

- 장점

연계 및 통합 구현이 단순, 용이

개발 소용 비용 및 기간이 짧음

중간 매개체가 없으므로 데이터 연계 처리 성능이 대체로 좋음

- 단점

시스템 간의 결합도가 높아 시스템 변경에 민감

암복호화 처리 불가

연계 및 통합 가능한 시스템 환경이 제한적

## ▼ 간접 연계 방식

### ▼ 분류

- 연계 솔루션 (EAI)

기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션들 간의 정보 전달, 연계, 통합을 가능하게 해주는 솔루션

- Web Service (ESB)

- 소켓

소켓을 생성하여 포트를 할당하고 클라이언트의 요청을 연결하여 통신

- 장점

서로 상이한 네트워크, 프로토콜 연계 및 통합 가능

인터페이스 변경 시에도 장애나 오류 없이 서비스 가능

보안이나 업무 처리 로직을 자유롭게 반영 가능

- 단점

연계 아키텍처 및 메커니즘이 복잡해 성능 저하 요소 존재

개발 및 적용을 위한 테스트 기간이 상대적 장기간 소요

## ▼ 수행 절차

1. 연계 데이터 추출 및 생성

2. 코드 매핑 및 데이터 변환 - 송신 → 수신, 수신 시스템의 코드로 매핑 및 변환, 데이터 타입 상이할 경우 변환 작업 수행

3. 연계 테이블 또는 파일 생성 - 데이터베이스의 테이블 또는 파일 형식으로 생성

4. 로그 기록 - 송수신 시스템에서 처리되는 과정에 대한 결과를 기록 및 관리

5. 연계 서버 또는 송수신 어댑터

연계 서버는 한 곳에만 설치

없는 경우 연계 데이터가 송수신 시스템의 운영 서버에 모듈을 구현하여 인터페이스 데이터를 전송 형식으로 변환, 송신 등의 역할 수행

6. 전송 단계

생성된 연계 데이터는 네트워크 환경에 따라 설정된 전송 형식으로 변환된 후 연계 서버에서 수신 시스템으로 전송.

전송 주기 결정 요소 - 추출 및 생성 방식, 연계 서버나 송수신 어댑터의 연계 주기, 수신 시스템의 연계 데이터를 운영 DB에 반영하는 방식

7. 운영 DB에 연계 데이터 반영

변환(매핑) 프로그램을 별도로 구현

## (2) 연계 장애 및 오류 처리 구현

### 1) 오류 유형

- 연계 시스템 오류

연계 서버의 실행 여부 및 송수신, 전송 형식 변환 등의 오류  
서버의 기능과 관련된 장애 및 오류

- 연계 프로그램 오류

데이터 추출을 위한 DB 접근 시 권한 불충분(송신)  
운영 DB에 반영하기 위한 DB 접근 권한 불충분(수신)  
데이터 변환 및 반영 시 예외 처리 미처리

- 연계 데이터 오류

연계 데이터가 유효하지 않은 값으로 인한 오류

### 2) 연계 장애 사례 및 처리 방안

- 연계 시스템 오류

연계 서버 미 구동

송수신 시스템의 IP 및 포트 접속 불가

→ 연계 서버의 로그를 확인하여 원인 분석 후 처리

- 연계 프로그램 오류

등록되지 않은 코드로 매핑 불가

등록된 데이터가 존재하지 않음

→ 송수신 연계 프로그램 기록 로그 확인 및 처리, 데이터 재전송 및 재반영

- 연계 데이터 오류

유효하지 않은 일자

→ 연계 프로그램 로그를 확인하여 데이터 보정 후 재전송

### 3. 내외부 연계 모듈 구현

#### (1) 연계 모듈 구현 환경 구성 및 개발

##### ▼ 연계 모듈 기능 구현

- 개발하고자 하는 응용 소프트웨어와 연계 모듈 간의 세부 설계서를 확인하여 일관되고 정형화 된 연계 기능을 구현

##### ▼ 방식

##### ▼ EAI/ESB 방식

- EAI

기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션들 간의 정보 전달, 연계, 통합을 가능하게 해 주는 솔루션

각 비즈니스 간 통합 및 연계성을 증대시켜 효율성을 높여 줄 수 있으며 각 시스템 간의 확장성을 높임

- ESB

기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션들 간을 하나의 시스템으로 관리 운영할 수 있도록 서비스 중심의 통합을 지향하는 아키텍처

버스를 중심으로 각각 프로토콜이 호환이 가능하도록 애플리케이션의 통합을 느슨한 결합 방식으로 지원

#### ▼ 절차

- i) 연계 DB 또는 계정 생성
- ii) 연계를 위한 테이블 생성 - 연계 테이블, 로그 테이블, 매핑 테이블, 오류 코드 테이블 등
- iii) 연계 응용 프로그램 구현 - 트리거 활용

#### ▼ 구현

- 추출 대상 테이블에서 삽입, 삭제, 갱신이 발생했을 때 로그를 기록하고 연계 테이블에 데이터 전달

```
-- 트리거 생성
CREATE OR REPLACE TRIGGER 프로그램명

-- 연계 데이터 추출할 테이블 각 행에 INSERT, UPDATE, DELETE 발생할 때마다 수행
AFTER INSERT
  OR UPDATE
  OR DELETE ON 테이블명
FOR EACH NOW

-- 선언부
-- 사용할 변수 선언
DECLARE AFTER_CODE VARCHAR2(5);

-- 트리거 시작
BEGIN

-- 데이터가 추가 된 경우
IF INSERTING THEN

-- 공통 코드 및 데이터 변환 처리
SELECT [변환 후 코드 값]
  INTO AFTER_CODE
  FROM [코드 매핑 테이블]
 WHERE [변환 전 코드 값] = :new.[코드 컬럼]

-- 진행 단계별 수행 결과에 대한 로깅 처리
-- 로그 테이블에 처리 결과 INSERT
INSERT INTO [로그 테이블명] ( [컬럼1], [컬럼2] .... )
VALUES ( ... );
```



```

-- 연계 테이블에 INSERT 처리
-- 변경 구분을 'I'로 설정
INSERT INTO [연계 테이블명]
( [컬럼1-처리일시], [컬럼2 - 변경 구분], ... [컬럼n])
VALUES (SYSDATE, 'I', AFTER_CODE, ... [:new.컬럼명]);

-- 데이터가 삭제된 경우 처리
ELSE DELETING THEN
(...)

-- 데이터가 수정된 경우
ELSE

-- 연계 데이터 중 코드 컬럼이 변경된 경우
-- 텍스트 컬럼이 변경된 경우
-- 연계 데이터가 아닌 컬럼이 변경된 경우
IF UPDATING(' [코드1컬럼] ')
OR UPDATING(' [코드2컬럼] ')
THEN
...
ELSEIF UPDATING(' [텍스트 컬럼] ')
THEN
...
END IF;
END IF;

-- 오류처리
EXCEPTION
WHEN NO_DATA_FOUND THEN (...)
WHEN DUP_VAL_ON_INDEX THEN (...)

-- 종료
END;

```

## ▼ 웹 서비스 방식

### ▼ 개념

- 네트워크에 분산된 정보를 서비스 형태로 개방하여 표준화된 방식으로 공유하는 기술로써 서비스 지향 아키텍처 개념을 실현하는 대표적 기술

### ▼ 유형

#### ▼ SOAP (Simple Object Access Protocol)

- HTTP, HTTPS, SMTP 등을 사용하여 XML 기반의 메시지를 네트워크 상태에서 교환하는 프로토콜
- HTTP 프로토콜 상에 SOAP Envelope, 헤더, 바디 등이 추가된 XML 문서로 기본적인 송수신은 HTTP로 수행

#### ▼ WSDL (Web Service Description Language)

- 웹 서비스명, 제공 위치, 메시지 포맷, 프로토콜 정보 등 웹 서비스에 대한 상세 정보를 기술한 파일로 XML 형식으로 구현

#### ▼ UDDI (Universal Description Discovery and Integration)

- 웹 서비스에 대한 정보인 WSDL을 등록하고 검색하기 위한 저장소로 공개적으로 접근 검색이 가능한 레지스트리
- 알려진 송수신 시스템의 통합 구현에서는 서비스의 공개 및 검색 과정이 필요 없으므로 UDDI를 구축 안함

#### ▼ 구현 절차

EAI, ESB와 달리 실제 전송이나 전송 이력에 대한 기록 및 모니터링 기능을 별도로 구현해야 함

##### 1) 송수신 파일 경로 및 파일명 정의

##### 2) 송신 연계 응용 프로그램 구현

연계 데이터를 추출하고 송신 파일을 XML 형식으로 생성하는 프로그램 구현. 주기적 실행 위해 크론 또는 잡 스케줄러에 등록

##### 3) 파일 전송

HTTP 기반 SOAP 사용하여 전송

SOAP 대신 REST 프로토콜 대체 가능

##### 4) 수신 DB 반영 서비스 호출 처리

연계 파일을 수신 위치로 전송 후 파일 내용을 수신 DB에 반영하는 서비스 실행

##### 5) 수신 연계 응용 프로그램 구현

연계 파일의 데이터를 수신 시스템의 운영 DB에 반영 및 처리 과정을 기록하는 내용으로 구현

#### ▼ 웹 서비스 방식 연계 모듈 구현

##### 1) 파일 전송 및 서비스 호출 처리 구현

## (2) 연계 테스트 및 검증

### ▼ 개념

- 송신 시스템과 수신 시스템을 연계하였을 경우 **데이터의 정합성과 데이터 전송 여부에 대한 테스트**
- 사전에 테스트 케이스를 작성하고, 작성된 테스트 케이스를 활용하여 연계 테스트를 수행한 후 수행 결과를 검증하는 방식

### ▼ 절차

#### 1) 연계 테스트 케이스 작성

- 단위 테스트 케이스

송수신 시스템 간 연계 데이터 정상 추출 여부, 데이터 형식 체크, 데이터 표준 준수 여부 등을 테스트할 수 있도록 작성

- 통합 테스트 케이스

송수신 시스템 간에 연계 시나리오를 구성하여 다양한 결과가 나올 수 있도록 테스트 케이스 작성

#### 2) 수행

- 연계 테스트 케이스의 시험 항목 및 처리 절차대로 실제 연계 응용 프로그램을 실행하며 테스트하고 결과 확인
- 송수신용 연계 응용 프로그램 각각 단위 테스트 수행

#### 3) 검증

- 동작 단계 별 오류 여부 체크, 오류 수정
- 송수신 연계 테이블을 통한 트랜잭션이 실패했을 경우 프로세스 구간 별로 예외 처리 방안 고려

### ▼ 사례

#### ▼ 송신 연계 테스트 검증 사례

- 연계 데이터 생성

선택 SQL, 프로그램 오류

데이터 객체 생성 오류

→ 사용자에게 오류 발생 시 알람이 발생되도록 프로그램 개선

→ 예외 케이스가 발생하지 않도록 프로그램 개선

- 연계 테이블에 입력

입력 SQL 오류

데이터 정합성 오류

→ 입력 실패 결과와 원인을 연계 테이블에 기록

→ 예외 케이스를 분석하여 발생하지 않도록 프로그램 개선

- 연계 데이터 전송

DB Connection (네트워크, 서버 설정 등) 오류

→ 통신 결과를 읽어서 실패 결과와 원인을 연계 테이블에 기록

→ DB Connection 재설정 후 다시 테스트 수행

#### ▼ 수신 연계 테스트 검증 사례

- 연계 데이터 읽음

데이터 선택 시 오류(SQL, 프로그램)

→ 수신 측 사용자에게 예외 발생 알람(원인 포함)

→ 연계 테이블에 예외 사항 기록

→ 재발되지 않도록 프로그램 개선

- 데이터 트랜잭션

데이터 트랜잭션 시 프로그램 논리상 오류

→ 사용자에게 예외 알람

→ 연계 테이블에 예외 사항 기록

→ 재발되지 않도록 프로그램 개선

- 처리 결과 응답

DB Connection(네트워크 서버 설정 등) 오류

→ 연계 테이블에 예외 사항 기록

→ DB Connection 재설정